A COGNITIVE APPROACH TO UNDERSTANDING PHYSICIAN USE OF CPOE: A FIELD USABILITY EVALUATION OF COMMUNITY PHYSICIANS AND COMMERCIAL SYSTEMS

by

Kenneth P. Guappone, MD

A DISSERTATION

Presented to the Department of Medical Informatics and Clinical Epidemiology

and the Oregon Health & Science University

School of Medicine

in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

June 2008

School of Medicine
Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is certify that the Ph.D. dissertation of

Kenneth P. Guappone, MD

has been approved

Chair and Advisor, Joan S. Ash, PhD

Member, Dean F. Sittig, PhD

Member, Brian L. Hazlehurst, PhD

Member, Richard F. Gibson, MD, PhD

Member, Richard J. Appleyard, PhD

Table of Contents

Index of Tables.

Index of Figures

Acknowledgements

I would like to thank the National Library of Medicine for financial support throughout this research. Primary support came through my position as a Biomedical Informatics Fellow at Oregon Health & Science University, with the Biomedical Informatics Research Training Grant T15 LM007088. Other NLM grants that provided support were Unintended Consequences of Physician Order Entry, R01 LM006942-05 (PI: Joan Ash) and Clinical Decision Support in Community Hospitals: Barriers & Facilitators, R01 LM006942-07 (PI: Joan Ash.) Funding for research expenses was aided with an OHSU Department of Medical Informatics and Clinical Epidemiology Dissertation Research Award, June 2007.

There are a great many people who have helped me through this long process.

First, I would like to express my heartfelt thanks to my talented dissertation committee members for their invaluable help and support. I thank Joan Ash for her years of steady guidance, support and friendship that taught me the special value of qualitative research. I thank Dean Sittig for his understanding of clinical information systems, Dick Gibson for his knowledge of the real world implementation of these systems, Brian Hazlehurst for his help in maintaining scientific rigor and Richard Appleyard for his knowledge of usability evaluation.

I want to thank Andre Kushniruk and Elizabeth Borycki at the University of Victoria, British Columbia for their kindness and hospitality. It was they who advised me on the use of the portable usability laboratory and gave me the confidence to try it.

At each of my study sites, I received a great deal of help from very busy people who selflessly donated their time to help me. This research would have been impossible without them. At Providence Health System, I would like to thank Dick Gibson, for his help with my initial contacts, and Jim Carpenter, Alan Hendrickson and Stuart Morgan for their help with the details of the information systems, setting up my laboratory and recruiting subjects. At Legacy Health System, I thank Juan Millan for the initial introductions and Chad Cagnolatti and Todd Guenzburger for their time and effort with the mechanics of the research. At Peace Health, I thank Brian Churchill, John Haughom, Marcus Pierson and Deb Brown for their considerable help in recruiting subjects and setting up my testing.

There are many colleagues at OHSU who helped me. Kathryn Krages helped me with grant applications. Cody Curtis was always available with her expert editing advice. Richard Dykstra created CPOE and CDS annotated bibliographies at www.cpoe.org (with the help of Joshua Richardson) which were most helpful in researching the literature. I greatly thank all the members of the Physician Order Entry Team (POET) for their years of moral support, camaraderie and humor: Joan, Dean, Richard, Joshua, and Cody above and Emily Campbell, Carmit McMullen, and Caroline Wethern.

Lastly and most importantly, I thank my partner Gary Wilson for his constant encouragement and unflagging confidence and love.

Abstract

Objective:

Computerized Provider Order Entry (CPOE) is an important component of the electronic health record, but there has been some difficulty with user acceptance, and this is often due to poor computer interface usability which disrupts clinician workflow. This qualitative research employed usability engineering methods to study community hospital physicians using commercial CPOE systems while in a naturalistic context.

Methods:

Using Norman's Theory of Action as a framework for analysis, this study sought to understand any disparity between the users' mental models and the representations and behavior of the computer system as the basis for usability problems. Community-based physicians from three health systems were observed entering patient admission orders in real time on commercial CPOE systems. Using think-aloud methods, the users' words, video images and computer display activity were recorded. These data were analyzed from cognitive, human-computer interaction and grounded theory perspectives.

Results:

Numerous usability problems were uncovered with these observations. Users demonstrated cognitive problems when they were unable to specify the actions needed to enter orders and when they had difficulty perceiving and interpreting the display. They were noted to have problems locating items on the display and remembering details of their respective systems and they expended excessive time and effort entering data. The users occasionally were resigned to actions that were not initially intended, guessed at the next steps to take and often ignored items on the display. Specific CPOE interface issues that caused usability problems were inappropriate alerts and default values, overly long lists and a lack of synonyms. Order sets were generally a benefit to usability.

Conclusion:

Field usability testing should be considered in all institutions implementing CPOE or other clinical computer applications as a means of identifying problems important to the user so that one might improve the usability and acceptance of those systems.

"...in an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it"

Herbert Simon (1)

Chapter 1. Introduction

1.1 The problem

The amount of information that must be utilized when practicing medicine can be overwhelming. No practitioner can hope to keep up with the volume or the frequency of change in information from biomedical research. In an effort to solve that information overload problem, computerized clinical information systems have been created as tools to provide the clinician with accurate information at the time it is needed in the practice of medicine. However, the complexity of the computerized information tools may in itself be aggravating the poverty of attention that this overabundance of information creates.

Computerized clinical information systems in medicine are in a precarious position in their evolution. We are several years past the pronouncement of the need for computerization as a means for quality and safety improvement in health care, but the hopes for that technology have fallen far short of the expectations of many. The use of the computer to guide medical decisions as they are made, that is, computerized provider order entry (CPOE) with clinical decision support (CDS) may be the most problematic aspect of this phenomenon. CPOE has the potential for extracting the greatest benefit from the computerized patient record and has yet to scratch the surface of that potential. Indeed, not only has the widespread use of CPOE not occurred, there are already criticisms of the technology as a potential cause of clinical errors (2, 3). Even when implementations have been “successful”, there are still significant problems with end-user acceptance during and after implementation (4).

1.2 CPOE

The Institute of Medicine’s well-known proclamations were a call to arms for Medical Informatics. The IOM reports noted that shortcomings in medical care did not occur because of the mistakes of individuals, but because more systems-wide problems existed. This recognition that an excessive information load on physicians was at the heart of the problem with medical errors allowed the consideration of information technology as a potential solution (5, 6). The Institute reiterated this in July, 2006 recommending the use of computer technology as a means of prevention of medical errors (7). Over the past decade and a half, there have been many studies that have shown that CPOE can, in certain circumstances, play a part in improving some of the most significant problems that beleaguer the current healthcare delivery system. Studies have demonstrated that the use of CPOE can increase the utilization of accepted preventive care practices and compliance with organizational guidelines and formularies. It has also been shown that these technologies can decrease errors in medication ordering, decrease unnecessary ordering of diagnostic tests and decrease hospital costs (8). The US Congress has proposed legislation in support of health IT and the Department of Health and Human Services has steadily endorsed the same, indicating that there is significant support for this issue (9). One should note that most of the research on the impact of health information technology comes from observing its use in only academic centers. A recent systematic review of health care information technology demonstrated that fully 25% of the quality studies were based on only four academic institutions with internally developed systems. Published evidence of the information needed to make informed decisions about acquiring and implementing health information technology in community settings is nearly nonexistent. (10) Of the 6706 hospitals listed in the American Hospital Directory, only 293 (4.4%) are major teaching hospitals (Council of Teaching Hospital members) in which the majority of orders are entered by resident physicians. Of the remaining 96% that are community hospitals, some have a minority of orders written by residents, but 85% have orders written solely by attending physicians. Thus, the vast majority of care in the US occurs in the community setting. (11)

So although it would seem an ideal time to encourage CPOE implementations, the introduction of CPOE into the US healthcare system presently faces significant barriers. The proportion of computerized order entry use in US hospitals is small; estimates vary, but it is certainly less than ten percent (12). In those institutions where CPOE has been implemented many different problems have arisen that have hindered smooth introduction of the systems. Foremost among these problems are the difficulties that the physician users encounter when making the change from paper to computer (13, 14). Arguably, the primary barrier to acceptance on the part of the clinicians is that the use of the computer to enter orders is extremely disruptive to their daily workflow. (15) Physicians justly complain that CPOE makes previously simple procedures difficult to perform and adds an onerous amount of additional time to the work routine. This added time spent is especially burdensome for the typical community hospital physician whose compensation is generally fee-for-service. Clinical information systems are often difficult to understand and thus may not be used productively or effectively. In addition to the disruption of workflow, the difficulty understanding the information system makes some physicians fear that the quality of patient care may be affected. This is important because the success of an information system rests on the acceptance of the users, who, in the case of order entry, are primarily physicians (14, 16-18).

As stated, one of the main reasons for user frustration and lack of acceptance of CPOE is the difficulty clinicians have in understanding and using the technology. Indeed, usability is a primary reason that workflow is disrupted and consideration of these unique usability issues must be made at the time of software development. It is well known in software engineering that human-computer interaction concerns should be addressed iteratively in the software development life-cycle (19). Unfortunately, considerations of end-user tasks and preferences by health care software developers are often overlooked, resulting in dissatisfied users and abandoned systems (20). Research in understanding user interfaces and the evaluation of these systems is thus greatly needed in heath care informatics.

There is clearly room for improvement in the human-computer interaction of clinical information systems in order to improve user satisfaction. However, even with optimal usability, there can still be problems with successful use because the skills of the user must be considered as well. The computer skills of physicians are quite variable and often may not be up to the task at hand; moreover, physicians' understanding of medical informatics issues are wanting (21). A poor user interface can also be compounded with insufficient skill in an environment where the incentives to adoption are lacking to begin with (22, 23). User training has been repeatedly observed to be instrumental in improving user acceptance. But, there are difficulties inherent in training physicians and this is especially so in the community setting. Here one often sees a lower level of computer sophistication, the physicians spend less time in the institution where the training takes place, and there is often a disinclination to learn to use CPOE effectively at all (13, 22, 24). Thus, in addition to understanding usability to advise application development, it would seem that efforts to make efficient and successful training sessions are warranted as well.

The need to create computer systems from the standpoint of the user’s needs instead of merely exploring the functional capacity of the machine is an idea that is transforming informatics. (25) Understanding the processes that occur when the clinician is using computerized order entry is crucial to guiding vendor design prior to implementation, the correct processes of implementation, the training the users receive after that, system maintenance and support and even further research on the subject. There is a great deal of information in the computer science literature studying end-user characteristics and behavior and how they relate to technology acceptance. This research has been inadequately extended, however, to the domain of clinical medicine and the users of medical information technology. What follows is some background information on technology acceptance, usability engineering and methods and frameworks that aid in understanding usability.

Chapter 2. Background

2.1 The extent of current knowledge

Over the past few decades, the study of information science has progressively shifted much of its focus from the technical issues of hardware and software development to the social, organizational and cognitive behavior issues of the users of computer applications. It has been recognized, especially in the last decade, that successful adoption of new technology in health care systems depends on the understanding of the complexity of these human organizations and individual behavior. Introduction of information systems into medical organizations has a profound effect on them and it is destined for failure unless one attempts to understand these individuals and their organizations and the change that is imposed upon them with the implementation of the potentially disruptive technology (26). Research in medical informatics is increasingly making attempts to acknowledge the human factors that are involved in information system implementations. In addition to considering these issues at the time of the introduction of the technology into the organization, one must address issues such as workflow even earlier, at the stage of software development (27).

Thus, in biomedical informatics we are learning that organizational and behavioral issues are important to understand when implementing information systems, especially CPOE. The biomedical literature has not, however, truly begun to examine the antecedents to successful individual end-user acceptance of this technology. The study of human factors and usability is just beginning to be recognized in our field. (28) However, outside of healthcare IT study of these topics is extensive and is summarized in the next section.

2.1.1 Technology Acceptance

The research in computer and information science is rich in addressing user adoption and the diffusion of technological innovation. The past thirty years are replete with theoretical models of user acceptance with a great deal of empirical evidence to support the theories. Foremost among these is the Technology Acceptance Model of Davis, further refined by Davis and Venkatesh that describes the now well-accepted concepts of “perceived usefulness” and “perceived ease of use” as antecedents to the intention to use and thereupon the actual usage of a technology. Perceived ease-of-use is defined as "the degree to which a person believes that using a particular system would be free from effort" and perceived usefulness is "the degree to which a person believes that using a particular system would enhance his or her job performance.” (29) The extended version of the Technology Acceptance Model (TAM2) further elaborates upon antecedents to those concepts (30). The technology acceptance models are based on various “value expectancy” theories, first proposed by Vroom in 1963 and extended by Fishbein in the 1970s. These theories posit that behavior is a function of the belief (expectancy) in the likelihood that behavior will achieve some outcome and the degree to which that outcome has value. The constructs of perceived usefulness and perceived ease-of-use are well-documented and are antecedents to the *intention to use* from which the behavior of *use* follows. Furthermore, perceived ease-of-use appears to itself have an influence on usefulness, and thus is an extremely potent predictor of use (31).

These concepts are firmly established in the information technology literature. They have been repeatedly demonstrated empirically to be predictors of use. These however do not give us any information as to what specific attributes of computer systems will lead to perceptions of ease-of-use and usefulness. One would like to understand, before the fact, how to build systems that users will accept, no matter what characteristics the user has. The study of these antecedents to use suggest that with time, system-specific perceived ease of use comes to reflect objective usability (32). In other words, there are many user characteristics that have been studied that collectively influence the user's assessment of a given computer system as being easy to use. There are also, conversely, attributes of the computer system that can result in an assessment of "easy to use" by a user. In any user with a given collection of "antecedents" there are numerous system qualities that allow the user to ascribe it the quality of "ease of use." These system qualities refer to the concept of usability of the system.

Although this may seem self-evident, efforts employed at improving human-computer interaction lead to improved computer system use and acceptance. Understanding usability can only be achieved by observations of users working with the actual systems. Usability may be broadly defined as "the capacity of a system to allow users to carry out their tasks safely, efficiently and enjoyably." (33) The application of cognitive science to what is known about the design and construction of machines and technology is known as cognitive engineering. When applied more specifically to the human-computer interface it is generally referred to as usability engineering. (34) (35)

2.1.2 Usability engineering

The field of user interface or usability engineering seeks to understand those elements of computer design that facilitate the ease of use of computer applications. It is the study of human-computer interaction. Human-Computer Interaction (HCI) research was initially an effort to apply cognitive science methods to software development, when it began in the early 1980s. Since then the field has developed considerably into a field of highly active multidisciplinary research and industry interest. (25) (36) The process is strictly user based in that it asks the user what he needs to make the system more acceptable. Equally important is the understanding that the user may not know what he needs to facilitate use. To effectively use a computer system, it must

1) be easily learned,
2) be efficient to use to make the user more productive,
3) be easy to remember how to use once away from the system,
4) prevent the user from making errors and
5) create subjective user satisfaction – make it pleasant to use.

Because this information in so highly related to the user, these usability characteristics of a computer system cannot be understood without methods for specifically studying the user (34).

Research in human-computer interaction (HCI) has its roots not only in computer science but also in cognitive science and psychology. The field of HCI is concerned with the study of humans as they use a computer within a certain task domain and what factors within that context contribute to the efficient, productive and enjoyable use of computer tools. The field explores empirically the issues that matter to humans such as computer design issues. There are many methods of evaluation of these issues and one important method is the think aloud evaluation, which will be described in detail subsequently. HCI also takes analytic approaches by studying the knowledge requirements of tasks and the cognition required for the execution of those tasks. (37)

When this research was first applied to industry, this field of applied cognitive science became known as cognitive engineering. A subset of that field, usability engineering, addresses the interaction of humans and computers. There are three major ideas that became central to this applied research. The first was that of the *iterative development* of computer applications, such that design is cycled with progressive evaluation at various stages. The second idea was that the empiric (and iterative) evaluation of the applications specifically employs *user participation*. The third important notion was cost-effectiveness so that cycles of development need not actually employ writing computer programs with each design stage. So, the computer interface was thus separated from the software application, and consequently, models of human-computer interaction then needed to be developed. (38)

A need for usability engineering arose in the first place because, like computerized order entry in medicine, the introduction of computerization in the work environment often leads to dramatic changes in the work processes and places increased cognitive demands on the computer user. (39) Thus the computer as a tool creates the possibility of greatly improving the user's performance at his work tasks, but because of a complex interaction between user and computer there often is a decline in human performance. As such there is a need for studying the cognitive processes that occur when humans use computers.

2.1.3 Methods for studying usability. "Please think aloud!"

"The Think Aloud Method consists of asking people to think aloud while solving a problem and analyzing the resulting verbal protocols." (Someren 1994) (40)

Various methodologies are employed in usability engineering to elicit users' points of view, such as interviews, observations, surveys, scenarios and heuristic evaluation (34). One method that attempts to isolate and understand a user's cognitive processes while he is performing a task is protocol analysis or think-aloud methods. The term protocol analysis refers to the "protocol," or verbal description of what goes on in one's mind as a task is performed. (41) Protocol analyses are undertaken in real time and, if carried out in their natural environment, capture the context of events. Think-aloud techniques are a widely used method in usability engineering and also may be the most valuable. (42) The participant in this type of usability study continuously verbalizes about his thoughts while using the computer system. One gets a very direct understanding of the user's interpretations of the system as well as his misunderstandings (34). The theoretical underpinning of this method is that verbal behavior may be treated as any other kind of behavior. Observation of the performance of a task and its attendant behavior (immediate verbal reports) is a valid method of accessing an aspect of a user's cognition or information processing (41).

Thinking aloud was originally discussed in psychological research in the 1920s and 1930s and was referred to as the "introspection method." It was used in research on productive thinking by Duncker in 1945. The 1984 work of Ericsson and Simon in protocol analysis, however, is considered the seminal volume on the topic. They argue that the process of verbalizing concurrently with the performance of a task is psychologically indistinguishable from the cognitive processes that occur in working memory (short term memory). This research technique gives one access to mental behavior and insight into the subject's thinking. (43) Verbal reports have been shown to be no different from other modes of responding, such as button presses with fingers or nodding of the head. It represents the information that has come to our attention and simply making it overt with verbalization. Ericsson and Simon claim that "cognitive processes are not modified by the verbal reports, and that task-directed cognitive processes determine what information is heeded and verbalized." (41)

Within the theoretical framework, there are two forms of verbal reports that are reflections of cognitive processes: concurrent and retrospective verbal reports. The concurrent report is the most important reflection of the cognitive processes and is divided into two types, Level 1, "talk aloud" and Level 2, "think aloud." Level 1 is that verbal response that is automatic and direct such as verbalizing a sequence of numbers during a math problem. This response did not need be transformed linguistically before being verbalized. Level 2 requires some verbal encoding to transmit the information as communication, such as the description of an image or abstract concept. Level 3 verbalizations, which are retrospective, also require verbal encoding as the user reports on tasks immediately after they have been performed. This response involves some filtering or analysis such as responding to a specific question or heeding previous instructions on what to report. This requires some additional cognitive processes to access the reported information. Levels 1 and 2 are considered the most valuable, because they are direct reflections of that information that is heeded and stored in working memory. Level 3 verbalization requires accessing long term memory to analyze and report on the heeded information in working memory. Thus, it may represent information in addition to what has immediately been attended to. (41, 42)

Although the concept of protocol analysis or think aloud study is quite popular, it is not without its critics. The main concern is the premise that the verbal protocols are a direct reflection of the user's cognitive processes. The very idea that users are speaking aloud automatically gives them a second task to perform, in addition to what they are talking aloud about. (44) Critics argue that thinking aloud disturbs the cognitive process by talking, but also due to interpretation by the subject of the task. In addition the task will necessarily be slowed by the addition of talking. There are limits to the capacity of working memory and this technique may be particularly problematic when attempting high cognitive load tasks. (40, 43) Although verbal protocols may not be a completely faithful reproduction of cognitive processes, it may not matter for the purposes of most user testing. In usability testing, one hopes to understand human-computer interface issues that considers human cognition, but one does not necessarily need to capture details of mental processing to produce useful results. (45)

The significant popularity of the think-aloud technique has also led to some misuse of the methods that differs from Simon and Ericsson's original vision. In fact, it is sometimes felt to be synonymous with "usability testing," which is not the case. The method, as described above, often does not match what is incorrectly called thinking-aloud. The instructor must be minimally involved, for if significant interaction occurs between him and the test subject, the cognitive processes of that interaction are what will be recorded, rather than the information processing about the task at hand, so-called interference caused by non-task related processing. Often, the preponderance of data is of the Level 3 type, which potentially may not reflect the user's cognitive processes and could have undue influence of the instructor. Extreme care must be taken to adhere to the original concept of thinking-aloud. (42)

Thus "thinking aloud" is theorized to allow access to cognitive processes. This presupposes and necessitates some model of what those processes might be. Some examples of cognitive models will now be explained.

2.1.4 Models of cognition. What goes on in users' minds?

One of the earliest models from cognitive psychology that relates to humans using computers is the information processing model, developed in the 1970's by Lindsay and Norman. It simply describes human cognition as a series of stages that takes information as an input, processes it and produces an output. Information that is received is encoded in some manner, compared with other data already present in current memory stores, chooses the best course in which to proceed and executes an action. (46) This model has been modified many times and has been a useful framework with which to observe and describe user behavior as well as a tool for the prediction of user performance. (47) It is roughly depicted in Figure 1.

Ericsson and Simon's model also sees human cognition as information processing, "that a cognitive process can be seen as a sequence of internal states successively transformed by a series of information processes."(41) There is recognition of sensory stores as well as short (or working) and long term memories, each of which has different capacities and speed with which information can be accessed. External stimuli are appreciated by sensory organs and the process of perception places these in working memory for cognitive processing. That information may be further processed and stored in long term memory stores. The latter process is reversible in that long term memories are retrieved for cognitive processing in working memory. (41) This model is shown in Figure 2 with its relationship to verbal protocols.

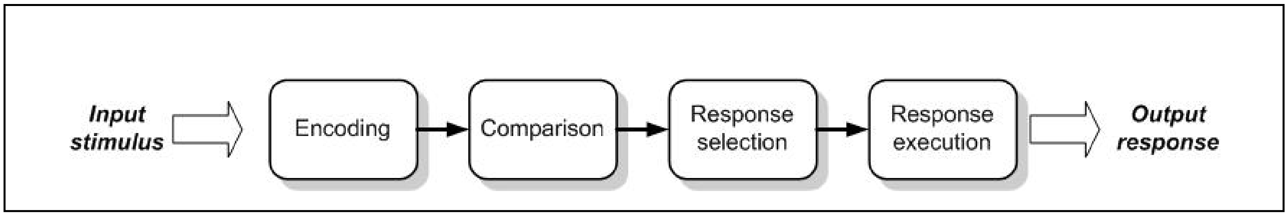


Figure 1. The Information Processing Model (46) Information from the exterior world is perceived and processed to produce an output response.

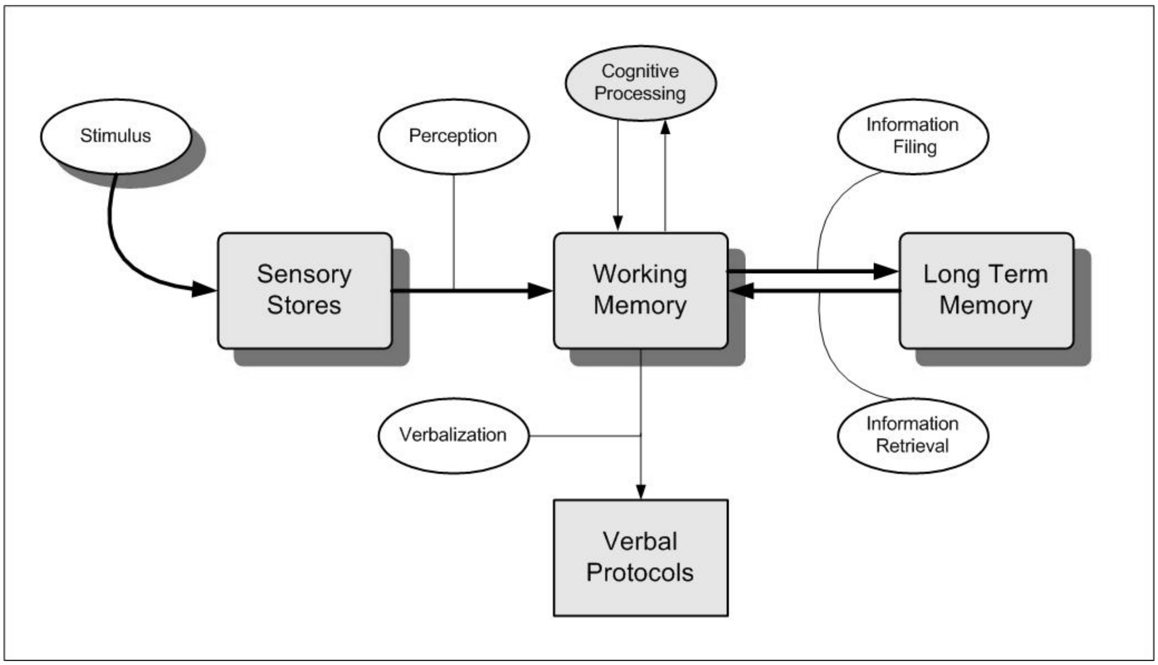


Figure 2. **Human cognition model: accessing with verbal protocols, modified from Norman, 1983. (48)**

External information is noted by sensory organs and placed briefly in sensory stores. The process of perception places that information into working memory. Information in working memory moves reversibly to and from long term memory. The verbal protocol is a manifestation of the working memory processes.

As potent and informative as the Ericsson and Simon cognition model may be, it was initially utilized mainly to understand very simple cognitive tasks and was not originally intended to apply to the use of sophisticated computer systems. While the model may still be apt for understanding these tasks, other cognitive models have been devised that are intended for use with information systems while still applicable to the think-aloud method.

One of the earlier models of cognitive structure related to human-computer interaction was that of Card, Moran and Newell. This classic model is similar to that of Ericsson and Simon in that it views the human mind also as an information-processing system. The first component of their framework is a characterization of this information processing and they called this the Model Human Processor. It is described as a set of memories and processes and a group of principles of operation. It was intended not as any statement of what occurs physiologically but to model and predict computer-user interaction. The human processor is divided into the perceptual system, the motor system and a central cognitive system that mediates between the other two. All of these have processors which have the attribute of Cycle Time (how fast it can access information). The cognitive processor also has memory which has the parameters of Storage Capacity, Decay Time (how long information remains in memory) and the Mode of information. (49). The second component of the Card et al. framework is a way of describing what users need to know in order to perform tasks and is known as GOMS, a reduction of computer tasks into its elementary actions. GOMS is an acronym for Goals, Operators, Methods and Selection.

The GOMS model has been used extensively and successfully in HCI research. It is actually a family of models that describes the knowledge necessary and the four components of skilled task performance: goals, operators, methods and selection. The goal is that higher level task that the user intends to accomplish. An operator is any action that is performed in order to reach that goal. The method is a specific sequence of operators that will bring about some action to achieve the goal. Of the various methods that are available, the user must apply some rule of selection. There are many variations of this model but all allow each action to be accurately observed and measured. The strength of this “keystroke level model” is its ability to predict the time it takes a skilled user to execute a task, based on the composite of actions of retrieving plans from long term memory, choosing among alternatives and executing the motor movements of keyboard and mouse. (49, 50)

While GOMS has a great deal of empiric substantiation and is highly predictive, it has some significant limitations. First, it assumes that cognitive tasks occur serially, which is often not the case. It applies to skilled users only, as it does not account for the time it takes to learn a system or recall how to use it after a period of nonuse. Lastly, it does not account for error, that is, its prediction value is based only on errorless performance. (50) As such, it is not helpful in studying novice users, performing parallel cognitive processes or in error-prone situations, which are precisely the user situations that may warrant HCI study. Specifically, it would not be appropriate for the purposes of this research about CPOE.

Hacker’s Action Theory is the first in a succession of theories that help to explain human activity in the context of goal directed behavior, describing actions that are human initiated as a series of stages that progress from a mental intention to a physical action of the human body. Hacker’s theory lists the main components of act, actions and operations. *Acts* are tasks that are higher order goals and they are motivated by intentions.

The acts are then realized through *actions*, which are components of the higher order act (sub-tasks.) Actions are the smallest unit of a cognitive or sensory-motor process, while still being oriented to that higher order goal. Lastly, *operations* are components of actions, but they are no longer specifically directed toward the original goal. (47, 51) They are "all-purpose" activities that may be used for any number of acts or actions.

Action theories that deconstruct user behavior have been developed and utilized by many researchers. While maintaining the concept of decomposing a task into smaller sub-tasks, Donald Norman takes a different perspective by analyzing the kinds of mental activities that occur in the control of action at the human-computer interface. He focuses on the user activities that occur at the gulf between the mental and the physical. The user must bridge the separation between the desired goal and the physical action of using a computer, thus the brain-machine interface. Bridging that gulf between the goal and the physical system is an "execution bridge" that implements some action. Then, the state of the physical world changes and that must be perceived and analyzed. The gulf must again be crossed in the other direction with an "evaluation bridge." See Figure 3. (35)

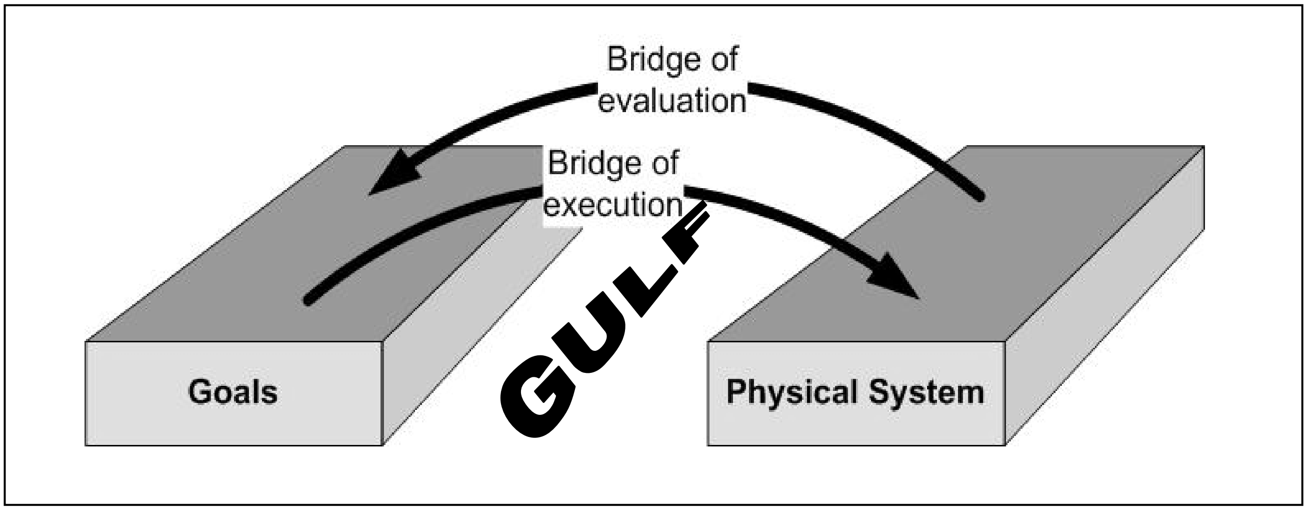


Figure 3. Bridging the gulfs of Execution and Evaluation, adapted from Norman, 1983. (35)
The user must bridge the gulf between himself and machine as a task action is executed. Then, changed in the state of the physical system must be evaluated and compared to the task goal.

The gulf of execution is bridged from the psychological side by forming intentions relevant to the goal and deciding upon an action sequence, and then executing those actions. From the system side there are characteristics of the interface built by the designer that must be evaluated. These characteristics change to a new state after a user has manipulated the system by the execution of any actions. The user bridges this gulf of evaluation by attempting to perceive the new state of the system, then interpreting that state. He then evaluates the system state by comparing it to the original goals and intentions. These tasks, Norman's seven "Stages of User Activity" are explained in Table 1 and shown in Figure 4.

If the new state satisfies the original goal, the action ceases. Otherwise, the user must determine what actions are necessary to bring the state closer to that goal, and form a new intention. The cycle then repeats.

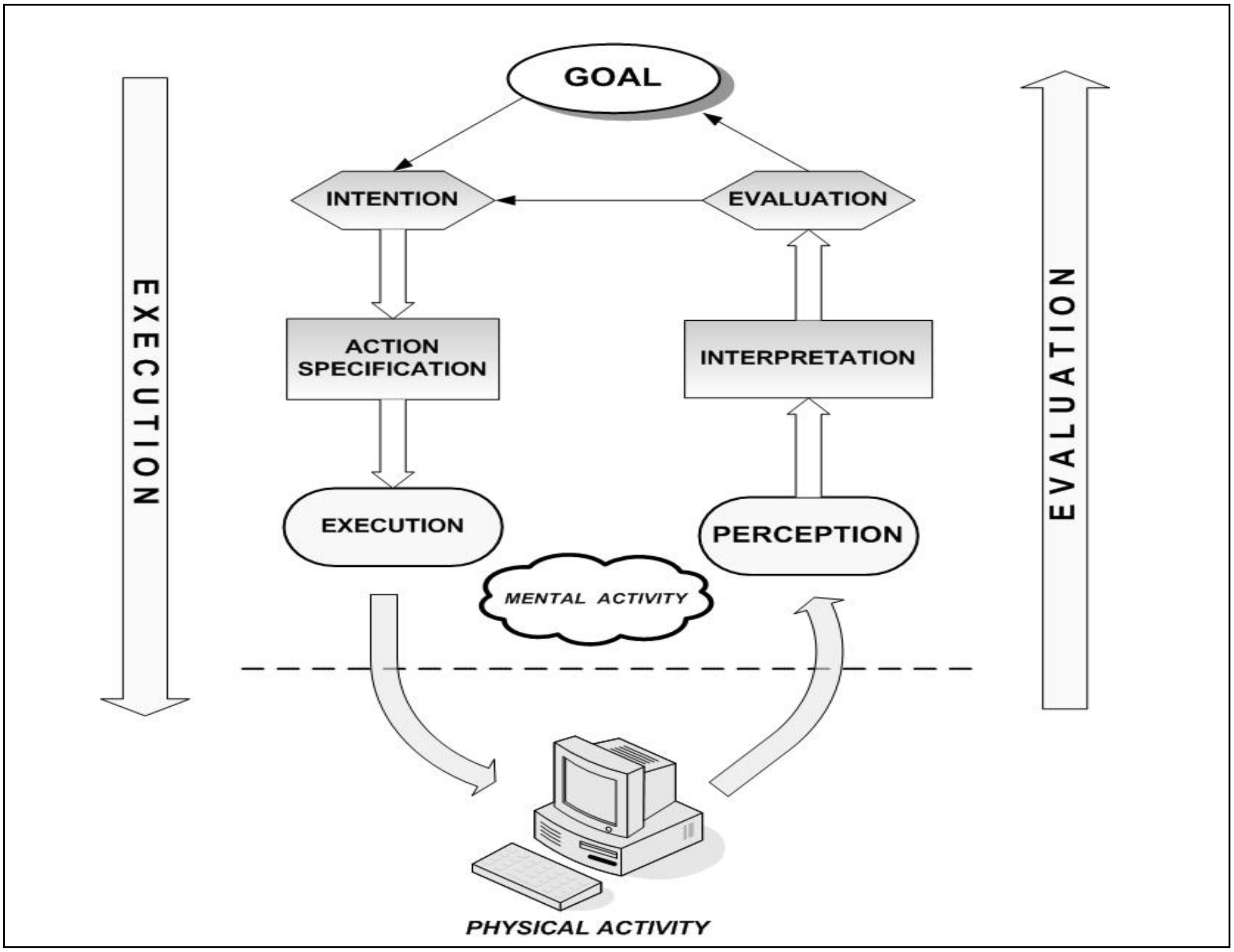


Figure 4. Seven Stages of User Activity involved in the performance of a task, modified from Norman, 1986 (35)

A benefit of Norman's model is that it directly addresses the issue of the interface between the mental and the physical. It also addresses the user's mental model of the system state based upon the external representations of the system. This framework is not quantitatively predictive, but does demonstrate how the cognitive consequences of particular software design on a user may be represented and understood. Users directly act upon some physical representation of an action and then can interpret and evaluate the end result of that action. (37) Tasks do not have to be done serially, nor is there any need to have any particular level of skill as errors may occur and be addressed.

Norman's cognitive model of user activity is valuable in understanding the use of computer systems by describing the intended tasks, the execution of actions and the evaluation of the system. For the model to be useful, one must be able to observe those behaviors during actual computer use. To interpret the behaviors observed during usability testing in terms of this model, some additional discussion in necessary. The following section attempts to explain what is actually observed in usability studies, and how they relate to underlying cognitive processes, and how these processes may be explained by the cognitive models.

1. Establishing the goal:	The state the user wishes to achieve.
2. Forming the intention:	The decision to act so as to achieve the goal.
3. Specifying the action sequence:	The user must map the psychological goals and intentions into the desired system state. He must determine what physical manipulations of the system are required to achieve that state.
4. Executing the action:	The user comes in contact with the interface mechanism of the physical system.
5. Perceiving the system state:	The interface display changes as a result of the physical manipulation and the user senses that.
6. Interpreting the state:	The user interprets the meaning of that perception, what the displays is communicating about the state of the system.
7. Evaluating the state:	The user compares this interpretation of the state to the expectations of what it should be, based upon the original goals and intentions. (35)

Table 1. Norman's Theory of Action.

2.1.5 Appreciating human cognition in action with usability engineering methods.

What inferences may be drawn from the data that are obtained in usability studies, specifically the think-aloud data? The verbal protocols that are generated from this method are theorized to depict representations of cognitive processes in working memory. By indirectly observing these cognitive processes, we can hope to understand what goes on in the mind of the user when he attempts to do work with a computer system as a tool.

2.1.5.1 Mental models.

When we humans use any technology, we form a mental idea of how that technology works. This is what is known as a mental model. For example, if one flips a wall switch up into the “on” position and the bulb in the ceiling lights, one needs to have some kind of understanding of how this process works to use it successfully. One understanding might be that flipping the switch allows the contacts to touch each other so that electricity may flow to the ceiling fixture. Another might be that turning the switch to “on” causes the bulb to light. Both of these mental models are really sufficient to effectively use a light switch, although the former is closer to what the designer of the technology envisioned. Another model might be that flipping the switch in an upward direction causes power to move upward toward the ceiling and light the ceiling fixture. This model would be less robust in certain other circumstances. If hoping to turn on a light fixture that is below the switch on wall, having this mental model would generate confusion.

Mental models are important to consider in the use of technology. There are four concepts to consider. The target system is the system the person is learning or using. The conceptual model is the model conceived by the designer of the system on how it works. The mental model is the user's conceptualization of the system. It may or may not be technically accurate, but it needs to be functionally accurate to be useful. The user's mental model may evolve with learning and persistent use. Lastly the scientist's conceptual model needs to be considered. That is the model the researcher has of the user's mental model. The closer the user's mental model maps to the designer's conceptual model of the target system, the more effective the use. These two may differ for two reasons. The user may not have a clear understanding of the system or the computer interface may poorly represent the designer's concept of how the system works. In either circumstance, the internal representations of the user poorly match the external representations of the physical system. Lastly, the closer the scientist's conceptual model obtained from usability studies maps to the user's mental model, the more useful the results are. (52)

2.1.5.2 Slips

It is important to understand that users' mental models are frequently incomplete. They may not be precise and the user may not feel certain about them. Mental models are "unstable" in that they often change or are forgotten with time. They do not have firm boundaries – similar devices or similar operations get confused with one another. They can even be "superstitious," that is, models and rules are created because they seem to work sometimes. This allows the user to feel more certain, which in turn reduces mental effort. The mental model reflects the user's beliefs about the target system and the researcher's conceptual model should contain the relevant parts of this belief system. (52) For this reason, an organized and reasonably objective model of the user's behavior is essential is capturing the essence of the user's mental model.

If the user was able to work with this computer tool flawlessly, there would be no problem at all, and no reason to study this situation in the first place. If the user's understanding of the computer system were complete there would be no difficulty in using it. When looking at the situation with a cognitive model, we are attempting to understand how the internal representations of the user differ from the external representations displayed by the computer system. We want to know what is going on in the mind of the user and the extent to which it differs from the original system designer's ideas of how the system works. Simply put, we want to understand how the user makes errors when using the system.

One way of recording and analyzing the user behavior would be to simply look at it from the Norman's theory of user action, as described above (Figures 3 and 4). One might attempt to classify user behavior based upon it is establishing a goal, forming an intention, etc. The Norman framework places heavy emphasis on studying errors. Within this framework, an error of intention (having the incorrect intention) is called a "mistake." That is not what one wishes to understand in usability studies, as a mistake is entirely related to the correct practice within the user's domain. However, an error in carrying out the intention is a "slip." (53) Understanding why users commit slips is directly related to why the physical behavior might differ from the user's original goal. A slip may be related to problems in any of stages of user activity in Norman's model.

Norman has done other research in studying slips. This is a slightly different framework from the stages of user action, but provides considerable insight into the mismatch between the user's internal and external representations of the technology system, and will be elaborated. However, in order to understand Norman's "classification of action slips," some further discussion of a few cognitive science concepts is necessary. I will briefly describe these.

2.1.5.3 Schemas, Activation and Triggering Conditions (The ATS system – Activation/Trigger/Schema system).

In order to study them, cognitive processes need to be depicted in some manageable way. The concept of the schema is one that is useful in cognitive science. The schema is a structure that is representative of a chunk of knowledge or a mental idea. A schema is an organized memory unit and it may represent a unit of perception, knowledge or motor activity. One important concept of schemas is that they have a hierarchical structure. Higher level schemas (parents) contain the smaller component parts of sub-schemas (children). When thinking of a schema in terms of memory, when we can recall a parent schema, we can also recall its children. For example, one schema would be a representation of a "house." Sub-schemas of "house" are attributes of that house, such as "contains rooms," "is a building," or "has a roof." If we can remember "house," we already know those children schemas and we do not have to specifically remember them.

If we didn't know what a house was, we could describe it in terms of some sub-schemas that we already know. (54)

Norman uses the concept of schemas in a less common way and applies them to motor skills and physical actions. The same hierarchical structure applies in these action schemas. Skilled actions need only to be specified at their higher level of memory representations. If I know how to perform the schema "drive a car," I already know how to execute the sub-schemas of "turn the steering wheel" and "depress the brake pedal." So if instructed to drive, I do not need to be told to use the children schemas. I don't even need to think about them, in the same way that if I was told to think about a house, I would not need to be told that a house had rooms or that people live in it. (53) Of note, the highest level of an action schema is the intention.

Other concepts related to schemas are activation levels and triggering conditions. These relate to how easily one can recall a schema. The activation of a schema is the probability and speed that one can access a knowledge structure in memory. The activation level is how easily ideas "come to mind" and is associated with the existing related schemas that are already in one's working memory. The idea of free association is due to the activation level of schema. If one hears the word "bread," one is much more likely to think of "butter" than "giraffe." "Butter" has a higher activation then "giraffe" after I say the word "bread." Certain schemas recently attended to raise the activation level of related schemas (54). That is why hints and mnemonic devices are helpful at improving recall.

Action schemas have activation values, but these alone are inadequate to specify which schema will be recovered from long term memory. In addition to other related thoughts that have recently been attended to, environmental conditions have to be such that they trigger the schema. When the current condition is sufficiently matched to the trigger condition, the action schema may occur. For example, when driving, I know how to "release the parking brake" but that action schema will not be triggered without the appropriate previous actions of "turn on the ignition" and "depress the brake pedal." So, both activation level and the goodness-of-match of the trigger conditions are necessary to initiate an action schema. A schema must be sufficiently activated to remember it and there must also be a satisfaction of the trigger condition.

Norman gives a wonderful illustration of this concept:

> Consider an example. When I drive home from the University, the intention to go home [parent] activates a host of relevant child schemas. These schemas then get triggered at appropriate times by satisfaction of their conditions by previous actions, by the environment, or by perceptions. I need not consider the details: I intend only that I should drive home. I can now do other tasks such as talk to a passenger, listen to the radio and think about things other than the driving. The normal schemas required for avoiding obstacles, maintaining speed, braking properly, and following the correct route all have been activated and all trigger themselves when appropriate conditions arise. Conscious attention to the task can vary, with the task itself demanding attention at critical action points. Suppose, however, that I wish to drive to the fish store, not to my home. Because the fish

> store route is almost identical to the route required to go home, it is specified as a deviation from the better-learned, more frequently used home route schema. For this purpose I must set up a new schema, one that is to be triggered at a critical location along the usual path. If the relevant schema for the deviation is not in a sufficiently active state at the critical time for its triggering, it is apt to be missed, and as a result, the more common home route followed: I find myself home, fishless. (53)

These concepts of schemas, activation levels and triggering conditions are important to understand as they relate to the idea of errors in cognition and the phenomenon of action slips, which will be described below.

2.1.5.4 Classification of Action Slips.

Again, the important information we want to glean from usability testing is the discrepancy between the intention of the user (the internal representation) and his actions. We are assuming that the user's intention is correct, but the problems arise when carrying out that intention and a slip occurs. Norman has devised a framework for classifying those slips which is applicable to the user action model (Table 1, Figure 3). He classifies these according to whether the slip is due to the intention of the action, the activation of the schema or due to faulty triggering conditions. (53) His original work related to the action of verbalization. I find many of his classifications overlap as well as being less apt for activity such as using technology, so I will condense them into to fewer categories as follows and summarized in Table 2.

Categorizing slips in actions can be helpful in understanding them. One needs to consider the source of the slip. The initial source could be in the formation of the intention. A mode error occurs if the user believes the system is in one mode, but it is really in another. This is an erroneous classification of the situation; such as if one has opened the wrong patient file. All activity will be subsequently “correct,” but errors obviously occur. Another error in the formation of the intention is a description error, in which the interface is so ambiguous or is inadequately specifies and action that one performs the wrong task. Here one actively looks for the correct mode, but misreads it because of the similarity of the action specification. (53)

Other slips may arise from faulty activation of an action schema – it may be unintentionally activated and the action occurs when not expected or it may fail to be activated and the action does not occur at all. In the case of unintentional activation, there are capture errors and data-driven activations. A capture error occurs when a familiar habit that is highly activated substitutes for the intended action. If you “pass too near” a common behavior, it may take over. Examples are walking into the kitchen to use the faucet and instead open the refrigerator, or hitting the “enter” button after entering text when one intends to hit the “tab” button, because the former action in both of these examples occurs more commonly. The capture error is due to “internal” activation. The other type of unintentional activation is an “external” one, the data-driven activation. In this case, there is an intrusion of external events and the intended action’s activation is altered. The arrival of new information just at the time of action can change it. An example would be taking out one’s credit card to order something on the telephone and punching in the card number, instead of the telephone number. (53)

Errors in formation of the intention		
	mode errors	
	description errors	
Faulty activation of schema		
	unintentional activation	
		Internal activation - capture errors
		external activation - data driven activation
	lack of activation	
		loss of intention
		misordering of components of action
Faulty triggering condition		

Table 2. Classification of action slips. Modified from Norman (53)

Rather than an unintentional activation, the schema may lose its activation. A loss of intention error occurs when one simply forgets the intention and the action never occurs. An example is walking into a room and forgetting why you went there. Another type like this is the misordering of action components. In this case the lower level children action schemas occur out of order or one is skipped, causing an error in the parent action. (53)

Lastly, slips may occur not due to an error in intention or activation, but because the environment is such that the action is triggered improperly, at the wrong time or not at all. If one is used to entering certain data in a computer system when a pop-op window asks you, a slip could occur if that prompt fails to fire. Often one makes mistakes when doing a task in a series. If one attends to a thought further along in the series it will trigger a different action than required at the time. (53)

2.1.6 Application of information obtained with usability engineering methods

The goal of a usability engineer is to create computer systems that fulfill the needs of the user. Like the broader concept of cognitive engineering, usability engineering considers the human cognitive factors that relate to the effectiveness, ease-of-use and satisfaction of the user when employing computer systems as tools. The above theoretical frameworks of human cognition may effectively constrain the point of view when observing the participants in usability testing. One may consider these frameworks such as Norman's Theory of action, users' mental models, schemas and action slips when analyzing the information obtained from usability testing processes.

For example, in order to create a goal or decide if a goal has been reached, the system state must be obvious at all times. The user needs to know what alternatives are available from which to choose, when deciding which action specification to formulate. The image of the system should readily reflect the conceptual model of the designer, so that the user's mental model of the system can parallel the designer's model. The user needs the appropriate type and amount of feedback to "to bridge the gulf of evaluation." The computer should have visual clues as to the function of its objects to increase the activation of action schema. In order to proceed smoothly, trigger conditions need to be apparent to guide the user to the next action needed to be initiated. Errors must be avoided so steps must be implemented to avoid action slips. (55)

Thus, the information gleaned from usability testing under the focus of a cognitive science lens can lead to more effective and usable computer systems. The goal of usability engineering is to understand the human factors involved with computer use, such that software development may be guided. There are hundreds of ideas that have been obtained that have generalizability and have prompted authors to abstract these numerous guidelines into guidelines and “usability heuristics.” (34, 55) Table 3 lists Shneiderman’s “Eight Golden Rules of Interface Design” which include the ideas of interface consistency, the presence of adequate feedback, error prevention and correction and the reduction of cognitive overload. The generation of such a list would be the goal of all research in usability.

1. Strive for consistency.	Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.
2. Enable frequent users to use shortcuts.	As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.
3. Offer informative feedback.	For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.
4. Design dialog to yield closure.	Sequences of actions should be organized into groups with a beginning, middle and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.
5. Offer simple error handling.	As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.
6. Permit easy reversal of actions.	This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry or a complete group of actions.
7. Support internal locus of control.	Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
8. Reduce short-term memory load.	The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced and sufficient training time be allotted for codes, mnemonics and sequences of actions.

Table 3. Rules for Interface Design (Shneiderman, (55))

2.2 An approach to the problems of CPOE with usability engineering

Computerized Provider Order Entry has the potential to make significant changes in the quality of health care, but its potential benefits have yet to be fully realized. Introducing new technology into any work environment has a profound effect on many aspects of work flow, including the cognitive behavior of the user (56). Many of the problems of user acceptance of CPOE, including possible errors caused by these systems, are a result of the difficulty in cognitively adapting to change in complex physical, social and cultural environments. Newly adopted technologies tend to alter work habits and may result in new sources of error. (57) Effective use of information systems is not only beneficial to user satisfaction, but is important in reducing the errors the systems were intended to prevent. Therefore, it might be said that the lack of attention to the principles of human-computer interaction when designing clinical information systems may in itself be considered a safety hazard. (58)

New technologies in health care may place heavy cognitive demands on the user, especially those who have not formulated a robust mental model of the system. When clinicians interact with information technology, their workflow is dramatically disrupted and new sources of error may be introduced, especially with poorly designed order entry interfaces. Methods of usability engineering have an important role in understanding the cognitive issues that arise when clinicians interact with information systems. Attempts to characterize and understand the cognitive demands of the computerized order entry task are crucial to identify sources of error and to inform interface design and user training. (57)

The following is a quote from David Woods that nicely summarizes the needs for user-centered design and usability research in health care IT:

> Virtually every Human Factors practitioner and researcher when examining the typical human interface of computer information systems and computerized devices in health care is appalled. What we take for granted as the least common denominator in user centered design and testing of computer systems in other high risk industries (and even in commercial software development houses that produce desktop educational and games software) seems to be far too rare in medical devices and computer systems. The devices are too complex and require too much training to use given typical workload pressures. (59)

There has not been a great deal of research regarding usability engineering in healthcare informatics, but its importance is progressively being recognized. Understanding the cognitive aspect in the usability of health care information systems is likewise gaining importance. In fact, failure to recognize and understand the cognitive needs of the users and the subsequent interface problems that arise may be one reason for the lack of widespread adoption of clinical information systems (60). Although not a great deal of research in this field has been carried out, it is argued that healthcare computer interfaces can no longer be designed without actively considering the cognitive skills and limitations of the clinician user (61). There is a need to explore and develop methodologies for the assessment of medical systems and their users. This can advise both the design process and the educational needs of the user. The study of human-computer interaction and of cognitive science in health care workers both show promise as a suitable methodology for these evaluation methods. (28, 33, 62) Kushniruk and Patel have explored a method for usability testing in the field that employs usability engineering methods that are easily usable in a small scale study of this problem (63-65) and will be subsequently elaborated upon in Section 4.6.2.

Chapter 3. Purpose of the study

The purpose of this qualitative study will be to understand the individual cognitive processes that occur as physicians use computerized provider order entry systems. This is important because of the difficulty that physician users often experience due to the sub-optimal usability of these systems. The participants will be physicians with diverse computer skills who practice at community hospitals using in-patient commercially available computerized order entry systems. They will be studied using usability engineering methods to better understand the cognitive processes of users of these types of clinical information systems, in order to inform subsequent software interface development and user training programs.

3.1 Research questions

- What cognitive resources are needed and utilized when community-based physicians use commercial CPOE in-patient systems?

    - What is the relationship between difficulties utilizing cognitive resources and barriers to usability?

- Does any disparity between the user's mental model and the external representations and behavior of the computer system create barriers to successful use of that system?

- Can an understanding of this disparity be exploited to modify the user's cognitive resources or the system's external representation, through training and improved system design, to lead to more successful use of these clinical computer systems?

Chapter 4. Methods & Procedures

4.0 Introduction

The purpose of this research is to gain an understanding of the cognitive processes that physicians use and the usability problems they encounter when they enter medical orders on computerized provider order entry systems. This section will explain how the effort to gain that understanding is best approached with the methods of qualitative research. The section will discuss the recruitment of the participants, the setting of the observations and the task that was observed. I will then expand on data collection methods – the administration of a survey and the think-aloud procedures. This chapter will conclude with a discussion of the data analysis, including the various data coding strategies that were employed.

4.1 Research Strategy - the choice of qualitative research.

Understanding the impact of information technology on clinical medicine is best understood from a social, organizational and psychological perspective, as these issues arise at the intersection of humans and computers. It takes an ethnographic and user-centered approach to these issues to truly understand them. Qualitative research employs a variety of methods to elicit information that is subjective, tacit and human. Qualitative data may sacrifice true generalizability, but they provide richness of detail. Understanding how physicians interact with clinical information systems and the cognitive processes that occur can be most effectively addressed with qualitative methodologies.

This study employed data collection and analysis methods from cognitive science and usability engineering, attempting to observe physician end-users in naturalistic situations in an attempt to mimic their usual work processes as they interact with a computerized physician order entry system. The entire approach to this research is based on theoretical frameworks of cognitive science and usability engineering. The subject selection, the behavior observed and the data collection and analysis are built on established methods of qualitative research.

4.1.1 Role of the researcher

As the sole data collector and analyzer, it has been my role to interpret the actions and words of the users as they work with the clinical information system. As my participants were all physicians, my experience was important. I received my medical degree in 1981 and subsequently completed a residency in Internal Medicine; I am Board Certified in Internal Medicine and practiced as a primary care physician for seventeen years. This allows me useful insight into the thought processes of practicing physicians. In addition, my training in biomedical informatics and four years experience in ethnographic research of computerized order entry qualifies me to effectively collect and analyze the data. Paramount in qualitative research is, however, the concept of the reflexivity of the researcher, that is, maintaining an awareness of personal perspective and possible biases and being explicit about them in the research findings. I have done this through articulating prior assumptions, maintaining self-awareness during the analysis process and remaining aware of biases as I report the results.

4.2 The Setting

4.2.1 Locations

The settings for the research were all community-based hospitals. For this study, a community hospital is defined as having a medical staff that primarily consists of community based physicians, not faculty members or trainees of an academic center. None of my participants were resident physicians in training, although one of the hospital settings did utilize resident physicians. The reason that community hospitals were chosen is that most studies to date have been done in academic centers, but the vast majority of medical care occurs in the community setting. (10, 11) The culture and organization are quite different in those non-community settings, most importantly because academic hospitals have resident physicians, who are employed trainees. Resident physicians enter the majority of the orders and are hence primarily responsible for interacting with the computer information system. They are somewhat limited in their power to object to changes in their work environment, which is quite distinct from private physicians in non-academic centers, who have much more political and economic autonomy. In the latter setting, physician acceptance and buy-in are imperative to successfully implement an information system and more difficult to achieve than in the academic setting because they have more power to object to the hospital circumstances imposed upon them. It is therefore important to understand how to maximize that acceptance. Improving the usability of the clinical information systems will greatly aid the achievement of that goal (66); hence, this study of usability was undertaken among community physicians.

In addition, this study observed users of commercial systems only. Many of the academic centers have “home-grown” systems which have been developed over years and have been deeply customized to smoothly integrate with the local culture. That is not possible with community hospitals, which must employ “commercial-off-the-shelf” systems and thus are less customizable. Understanding the characteristics of commercial systems is therefore important if one wants to study the community setting.

4.2.2 Participants and systems

The participants were members of the medical staff of three community hospital systems who are active users of an in-patient computerized physician order entry system. The hospitals were as follows: Providence Portland Medical Center in Portland, OR; Legacy Salmon Creek Hospital, Vancouver, WA; and Sacred Heart Medical Center, Eugene, OR and St. Joseph Hospital of Bellingham, WA. The last two are members of the same hospital system and share the same computer system. Therefore, three different CPOE systems were studied. All users were remunerated with a cash honorarium of $75.00.

4.2.3 The Task

I studied the participants in person. The subjects completed a short survey that collected background experience and attitudes toward computer systems. I then observed the participants as they entered data into the computer. The central event observed was the input of “admission orders,” the set of orders that the physician enters soon after a patient is admitted to the hospital, and is necessary to start the inpatient care process. This task was chosen as it is relatively similar for every patient being admitted to a hospital. In addition, the time duration of the task is long enough to gather an adequate quantity of meaningful data.

The original intention was to choose participants based upon the results of the initial survey. This would allow me to choose equal numbers of two types of computer users, experienced and novice, from each institution. As will be explained in the Results chapter, this approach was abandoned due to difficulty in obtaining enough users in general and novice users in particular. All possible participants were included. The original goal was twenty users of each of the three CPOE systems.

4.3 Recruitment

I chose the participant volunteers differently in each hospital due to different circumstances of accessibility to the users. In all situations, I obtained letters of support from physician administrators prior to making contact with the participants. These administrators' titles were Vice President of Medical Informatics, Chief Medical Information Officer and Vice President of Clinical Quality and Patient Safety in their respective hospital systems.

To recruit participants from Hospital System 1, I obtained information about every person who had been trained on the CPOE system and the number of orders that had been placed. There were 54 persons who had been trained on the application who were non-resident physicians and recruitment letters were mailed to each of them. I sent follow-up emails one week later to all whose email addresses were available, nearly all 54. I also made sporadic reminder faxes and telephone calls. I approached a group of hospitalists, who were the most frequent users of the system, and made a personal presentation to them during two of their monthly meetings to solicit their participation in the study. The Hospitalist Medical Director and the project manager of the CPOE implementation team were key contacts in this process, and they facilitated participant recruitment.

The medical staff office provided contact information at Hospital System 2, which claimed universal use of the CPOE system. A written contract with the legal department was required that restricted the use of this information to recruitment for this study only. The entire list of the medical staff was limited to the clinician specialties that were at all likely to ever perform a hospital admission. I mailed 202 recruitment letters and followed up with email reminders where possible. I gave a presentation on two occasions at meetings of the Clinical Informatics Committee to encourage discussion of the project and to obtain the names of possible interested participants. Important contacts at this institution that aided in recruitment were the President of the Medical Staff, Clinical Systems Manager, Hospitalist Director and members of the CIS Support Team.

The participant selection at the two hospitals of Hospital System 3 was more focused due to considerably fewer users of the CPOE systems at these locations. All possible users of the system who could participate were contacted locally by the CPOE Project Lead and the Regional Vice President Clinical Information, respectively, who individually contacted the users to arrange participation.

4.4 Data Collection Procedures

4.4.0 Introduction

The goal of the data collection is to understand the cognitive issues that arise while using computerized provider order entry that affect the usability of the system. The following steps were undertaken. First I administered a brief quantitative survey of user skills and attitudes with respect to computers. The survey was a modification of a longer validated survey. I then observed the users in real life situations, using computers to practice medicine, and recorded those observations with three modalities: the capture of all activity on the computer screen, the audio recording of the user's words and a video recording of the entire scene.

4.4.1 Survey

It has long been recognized that when evaluating computer users there are apparent differences between those who are identified as "experts" compared to those who are "novices." The novice user, compared with the expert, may be less comfortable or experienced with computers in general, may spend less overall time with computers in the work or home environment, may be less skilled or have less experience using the computer system of interest or any combination of those traits. This categorization of the user has implications concerning training, work effectiveness and efficiency and the likelihood that errors will be committed. (34, 55, 67, 68) It was for these reasons that I was interested in knowing the experience and sophistication of the CPOE users as they undoubtedly would have qualitatively and perhaps quantitatively different issues with respect to the usability of the system. Though one might surmise that usage issues will differ between novice and expert users, it is not clear what those issues might be or what the implications those issues will have with respect to usability of CPOE systems.

4.4.1.1 Creating and administering the survey.

Therefore, to explore the relationship between the users' computer sophistication and usability, I administered to the participants a version of the survey, "Computers in Medical Care," (69) that I modified as explained below. This survey seeks information about the participant's medical background and computer experience. The purpose of seeking this information was to understand whether the degree of computer skills or experience of the user has an influence on the cognitive understanding of the task. Cork et al. created this validated instrument that is focused on users and measures physicians' use of, knowledge about, and attitudes toward computers. The original instrument by Cork has four scales: attributes of computer use, self-reported computer knowledge, computer feature demand and computer optimism. It captures, with some approximation, the Technology Acceptance Model issues of (desired) usefulness and ease of use. (69) However, not all these issues in the original survey are germane to this study; therefore I modified it in the following ways. An entire section of the survey speaking to future development was never validated and I deleted it. The scales of computer optimism and general computer knowledge are interesting, but were excluded in the interest of shortening the survey to make it more acceptable. The scale regarding feature demand was found to be two dimensional in the original validation study and could be explained by the factors of “demand for sophisticated features” and “demand for usability.” The latter factor is highly relevant, but cannot be separated from the former, thus this feature-demand scale was also not used. The remaining scale of computer use is highly relevant, and was retained. In the original authors’ validation study, principle component factor analysis demonstrated that three of the original ten questions asking about specific computer tasks had factor loadings less than the threshold and were deleted. The remaining seven questions were used and show a high correlation with the other three remaining questions in the survey. (69) In this remaining scale of computer use there are two questions that ask about experience and sophistication of computer use in general. It would be helpful to understand those attributes with the respect to the specific CPOE application under study. Therefore, I added two additional questions that mirror the original questions but refer specifically to the CPOE application of the users’ respective institutions. See Appendix I for the complete survey that was used. This survey is considerably shorter that the original Computers in Medical Survey, but strongly captures information about the participants’ previous and current use of computers, which may be very relevant to the users’ computer skill amount of computer training needed. It potentially would allow one to see if usability issues differ with the sophistication of the user. Lastly, the shortened survey with thirteen questions was much more acceptable to busy physicians than the original 89-question survey. This abbreviated survey can easily be completed in less than three minutes.

My original intention for this study was to administer this survey to all willing participants and then stratify them according to novice and expert status prior to participation in the central part of the research, the think aloud process. This was not possible and all participants were retained regardless of their survey results. The reasons for this will be explained in the Results section. In order to improve access to the survey (and aid in analysis) the survey was placed online (SurveyMonkey.com, Portland OR,). Some participants preferred to do this on paper and the survey was printed out and data were entered manually to the online server by me. The survey was administered before the observation of only a few users, the remainder were given soon after. A few users needed to be pursued and reminded repeatedly to complete the survey.

4.4.2 Task Performance

4.4.2.1 Think-Aloud Methods

The central body of data was obtained through a think-aloud procedure. Think-aloud techniques are a widely used method in usability engineering and also may be the most valuable method for uncovering problems.(42) A think-aloud procedure is a method of evaluating the usability of a computer system by directly observing the user as he interacts with the computer. The subject is instructed to verbalize about everything he sees, thinks about and any actions he performs with the computer and this is all recorded for analysis. One gets a very direct understanding of the user's interpretations of the system as well as his misunderstandings.(34) The theoretical foundation of this method is that verbal behavior may be treated as any other kind of behavior that is observed. Observation of the performance of a task and the accompanying verbalizations is a valid method of accessing an aspect of a user's cognition or information processing.(41)

With the aid of real time screen capture software (HyperCam, v2.13.01 Hyperionics Technology) and videography, I observed the participant and recorded him using a commercial CPOE application to enter a set of orders. The screen capture application records all actions on the computer screen as well as the verbalizations of the user. The video camera recorded the user at his location as he interacted with the computer. I studied physician users as they went about their work day and observed them entering a set of admission orders in the hospital. "Admission orders" is that set of orders a physician prescribes when a patient first enters the hospital. Because of fairly uniform US medical school training, most physicians traditionally enter a relatively standard set of admission orders, often remembered by the mnemonic, ADC VAAN DIMSL (which stands for admit, diagnosis, condition, vital signs, allergies, activity, nursing, diet, IVs, meds, studies, labs.) Thus using admission orders as the task, there was a reasonable similarity among users, and especially so if an admission order set is available.

4.4.2.2 Pilot tests

Prior to visiting the study sites, three pilot tests were performed to ensure a smooth procedure in the field. I obtained Internet access to a test session of one of the CPOE systems studied and enrolled three colleagues – two were physicians, all three were trained informaticists. I used a simulated list of admission orders. The pilot testers commented on the think-aloud training (which was ultimately abandoned, as described below) the comprehensibility of the instructions for the process, and the overall level of comfort with the process. Adjustments were made according to these recommendations.

4.4.2.3 Observations in the field

A completely naturalistic setting was the goal. Ideally this would entail observing a participant use the computer in the location in which he was accustomed. However, I was able to observe a user enter admission order in his completely customary setting in only five of the 28 sessions (three were of the same user.) There were many conditions that precluded this truly naturalistic setting for the observation sessions. First, I needed to be present in the hospital setting for days at a time awaiting the call of the participant. The local contacts were unable to procure a location with a networked computer for me to set up and remain that was in or about a nursing station, the usual place for entering admission orders. My “laboratory” was completely mobile, but setting up a tripod with video camera in a busy nursing station was felt to be too intrusive to some. The computer used for the study had to have screen capture software loaded on it. This could not be done hospital wide which limited the number of computers that could be used. For security reasons, I was not permitted access to the local area network in some locations. Lastly, it was simply much more practical and convenient to be set up and ready to record the user activity at a moment’s notice by remaining stationary in a single location in the hospital. Invariably that location allowed the physician to come to me within minutes of being ready to enter orders, and all users were willing to do so without hesitation. It is the nature (and a benefit) of computerized order entry system that that it may be used ubiquitously and the users were accustomed to doing this in various locations. Although not completely naturalistic in location, this arrangement uniquely allowed me to capture the user performing the task at precisely the point in time in the user's workflow when this would be naturalistically carried out. The setups were as follows:

With System 1, I was located in an IT resource room and used a designated network PC. The user logged in and used the application in the normal fashion. In System 2, I was not permitted access to a network computer. There, I was located in the medical library and I loaded the required software on a personal laptop and connected to the Internet through public wireless access. The users accessed the CPOE application through a web access application, as they would outside the hospital. The speed of this connection was identical to that of the in-house local area network. A mouse and keyboard were used to better simulate the form factor of the hospital desktop PCs to which they were accustomed. In System 3, the methods varied. As stated earlier, this system was observed at two different hospitals. At one hospital, the screen capture software was installed on local desktops in two different nursing stations. In the remaining location, I was loaned a network laptop (and used a standard keyboard and mouse) and installed the software; the CPOE application was accessed through a wireless network. The testing occurred in the medical library for three users and at the nursing station for two others.

In order to assure that the think aloud session proceeds optimally, one ideally should do some prior "training" or rehearsal of the process, to be certain the user understands what is expected. The process of thinking aloud is not completely natural, so brief rehearsal exercises are helpful. I prepared a script for the training, using some the exact language and exercises from Ericcson and Simon (41). I did meet with two physicians who had agreed to participate and trained them prior to the think-aloud session. However, the actual circumstances of this process did not permit the training for any of other users. They essentially were unwilling to meet with me twice. I attempted to do the full training immediately prior to the think-aloud session, but the participants were in the midst of a busy work day, and quickly expressed impatience. I chose to forgo the originally planned training session for the remainder of the users after considerable resistance had been encountered. Instead, I briefly introduced the participants to the process immediately prior to the think-aloud session.

The actual session consisted of the user at computer wearing a microphone headset. The screen capture software was started and the user began the order entry process in the usual manner, but speaking aloud about his activity and thoughts. Occasionally the audio and video screen capture rate needed to be adjusted, depending on the processing power of the computer. Meanwhile, a video camera (using MiniDV tapes) was aimed at the user, to capture his image and his use of the computer and any attendant artifacts. Space usually limited this to a side or rear view of the user.

The user then continued the order entry process in her usual manner. As the participant entered data, think aloud prompts were given. In attempting to be faithful to the Ericcson and Simon model, these were kept to a minimum so as to not interfere with the thought processes of the participant (leading to Level 3 verbalizations) but merely to urge the user to “keep talking.” As will be discussed in the results section, the users needed to be prompted to “keep talking” and “tell me what you are thinking” repeatedly, despite my intentions to remain as unobtrusive as possible. Mouse action, keyboard entries and the images of all screen activity were captured directly in real time, along with the users’ audible verbalizations of cognitive processes in a digital audio/video file, while simultaneously his image was captured on video. Verbalizations were transcribed verbatim and time stamped to be compared to the visual data.

4.5 Confidentiality

Institutional Review Board approval was obtained at OHSU as well as all three of the hospital systems. As screen capture would record images of actual patient charts, one system required additional approval of its Privacy Board, a non-IRB entity described in HIPAA.

To ensure the confidentiality of the participants, the voice and image recordings were encrypted (encoded) and burned onto a DVD. The original recording on the computer at the hospital site was deleted. The paper survey was identified only by a study identification number. The users’ words were transcribed, but all identifying information was eliminated. The DVD containing the voice and image recordings along with the paper survey has been kept in a locked container in a locked room. This must be retained for academic reasons, in case further review is necessary after the study is completed. A period of five years is considered adequate, after which these raw data will be destroyed.

Part of the recording is a screen shot of the physician's order page as the doctor is writing the orders, which may have a patient’s Protected Health Information and demonstrates the orders that the doctor is writing at that time. With video editing software, I censored patient identifying information that is on the screen, such as name, date, hospital name, hospital unit or room, location and any identification numbers.

4.6 Data Analysis Procedures

4.6.1 Introduction

This analysis was performed with respect to several theoretical and applied frameworks. First, as stated, I hoped to remain as faithful to Ericsson and Simon when collecting the data in the think-aloud sessions and as such, I hoped to carry over those authors' ideas when analyzing the data. However, that framework is applicable to the observation of any activity with no special consideration of computer use. Therefore, the use of Norman Human Action Cycle played a major part in the analysis scheme. The data were further analyzed with attention to established usability heuristics that were collected from the literature. Lastly the data were reviewed and analyzed in a more grounded fashion. This will all be elaborated below.

The data in this research are the results of the survey and observations of clinicians doing their daily work by using a computer. The survey data are a small portion of the total and were analyzed with descriptive statistical techniques. The bulk of the data analysis was applied to the observations which were analyzed qualitatively. The raw data consist of digital video files and their accompanying text transcripts. These were reviewed repeatedly in the analysis process.

The actual process of analysis consists of segmenting the data into usable chunks, and I will explain the details and rationale. Once conveniently divided, the data are then "coded" to organize them into similar categories for the purpose of better understanding the data. The coding of these data was done in three separate, though interrelated ways and will be explained in detail. But first I will explain the specific processes that were carried out in order to facilitate analysis.

4.6.2 Specific Analysis Processes

The idea for this method of data collection came from discussions with Andre Kushniruk, PhD and Elizabeth Borycki, PhD of the University of Victoria School of Health Information Science, Victoria, British Columbia. Kushniruk et al. have developed techniques working with a "portable usability laboratory" that is intended to apply usability engineering techniques intended for work in the field. (64, 65) I consulted with these researchers personally and they were kind enough to allow me to observe their approach and to provide counsel, which was extremely valuable in applying similar methods in my research.

The screen capture application saves the visual activity of the computer display and the users' audio verbalizations of the session as an .avi file and this was transferred from the computer used on site. The audio portion was transcribed verbatim by myself and saved as a rich text file. The digital video file from the video camera was captured on the computer and also saved as an .avi file. Video editing software (Adobe® Premiere Pro®, v2.0, Adobe Systems, Inc.) merged the two video files into a single file. It should be noted that these files were quite large. For example, a 15 minute session produced a screen capture file of nearly 90 MB, and a video file of 3.2 GB. (The size of the combined file was similar to the video file alone.) This combined file and the text document were burned onto a DVD. Files of this size makes these processes (video capture and rendering, and DVD recording) quite time consuming, as all steps lasted several times the duration of the real time process.

Video analysis software (Transana v2.1, Wisconsin Center for Education Research, Madison, WI) was used for qualitative analysis of these raw data. The text file and the combined video were exported to this software. The data were “segmented” into order tasks as explained below and the transcript was time-stamped to correspond to the video file with this analysis application. A spreadsheet was created cataloguing the segments by order task and their times and outlined the exact actions of the user in detail. The videos and transcripts were then reviewed together and coded with the cognitive, usability and grounded coding schemes described below, and then entered into the spreadsheet with notes commenting on cognitive issues. Each video session was reviewed repeated times to expound on these notes and write analysis memos. Lastly Transana was used to group the video segments by code. Reports were made of these coding groups with the corresponding audio transcripts and memos. From these reports, interpretations of the coding schemes were created to be discussed in the Results section.

4.6.3 Theoretical Assumptions

There are theoretical assumptions upon which protocol analysis is based which must be taken into account in the interpretation of these data. In Ericsson and Simon's landmark work in this area the basic assumptions are as follows (41):

1. The subject's behavior can be viewed as a search through a problem space, accumulating knowledge [in a step-by-step manner] about the problem situation…
2. Each step in the search involves the application of an operator, selected from a small set … to knowledge held by the subject in short term memory.
3. The verbalizations of the subject correspond to some part of the information … in short term memory, and usually to information that has recently been acquired.
4. The information [consists of] … inputs to the operators, new knowledge produced by operators, and symbols representing active goals and sub-goals that are driving the activity.

In summary, this theory considers the user's behavior as consecutive steps through a problem space. Each step is performed by selecting some available action that occurs to the user at that time. Each action carries the user along in the problem space toward his goal. The user verbalizes those actions as he performs them and the words spoken are indicative of active cognitive processes.

As stated earlier, though the cognitive underpinnings of Protocol Analysis are pertinent to this study, the procedural specifics of the original Ericsson and Simon work are less applicable in studying the use of computer information systems. The Ericsson and Simon model only accounts for the users' verbalizations and was not intended to be used to analyze action recorded on a video screen. To account for that, I chose instead to evaluate the user with respect to the Norman Theory of Action (Human Action Cycle) framework discussed in the Background section. This framework is specifically aimed at computer use. Moreover, although the Norman Theory's details correspond to Ericsson model, they are more explicitly described. For both of these reasons, it is extremely useful for the analysis of the observation of clinicians using computers. The model serves as the underlying framework of my methodology and analysis. The Norman model is depicted graphically in Figure 4 and details are reiterated below.

An attempt was made to analyze the participant's behavior considering Norman's seven stages of user activity. The order entry activity was observed in terms of two processes, the execution of action and the evaluation of the system state that results from that action. Specifically the "gulfs" are considered (See Figure 3.) The gulf of execution is the difference between the user's intentions and goals and the actions allowable in the system and the gulf of evaluation is the difference between the representations of the system and the user's original expectations. The activity was analyzed by looking out for the user's understanding of the system state, the action alternatives and his mental model based upon the system representations. (55)

4.6.4 Segmenting the data

The raw data of this method are verbal transcripts and video images. Once these data are obtained, one then seeks to encode this qualitative information for the purposes of interpretation. The initial step is called “segmentation,” whereby an attempt is made to divide the data into individual usable statements, sometime called assertions or propositions. In general, this may be done based upon time durations, at pauses or phrases in natural language, or “instances of a general process,” i.e., a specific task or subtask. (43) In this study, segmentation was based on “order tasks.” These are natural “instances of the general process” of entering admission orders. As described above, the mnemonic of “ADC VAAN DIMSL,” the “steps” of admission orders, is already naturally used by many physicians in this process. This set of tasks is performed when entering orders on paper as well as in the computer. Indeed, the interface of many order entry systems is designed around these tasks, most specifically “admission” order sets, which are most likely to be used in this situation.

I therefore segmented the data according to “order tasks.” This approach proved to be useful in two ways. First, the order tasks mapped naturally to the user’s actions and are identified by commonly used names. Second, the time duration of these tasks was of a length that was practical enough for technical purposes and contained enough data to make meaningful interpretations.

The actual process was as follows. When observing the audio and video recordings, the beginning and end of each order task was usually quite apparent. For example, as a user starts to order a laboratory test, this is evident by the screen activity and his verbalizations. This task continues until there are no further actions to carry out and the segment was marked. The beginning and the end of each of the usual order tasks was easy to delineate. For several reasons, though, sometimes the designation of the task segment was less clear-cut and resulted in somewhat arbitrary divisions. However, the central purpose of this segmentation is practical, and it did not matter if these segments themselves were not uniquely revealing or meaningful as well defined order tasks.

4.6.5 Coding the data

After segmentation, the actual coding of the data takes place. In qualitative research, *coding* is a fundamental process for analyzing data. Coding is initially a process for categorizing and organizing qualitative data. Developing some manageable classification scheme for huge volumes of data is the necessary first step in data analysis. Once the data are organized into categories, one then may describe the implications and details of these categories. (70) The actual process entails reviewing all of the data and applying label or "code" to portions of the data. I have already described how I divided my data into manageable portions or segments. Once all the data have been labeled with codes, one looks at the data grouped under each code. When many pieces of the qualitative data are grouped together based upon some shared characteristic, one may then start to understand or explain the data in a new light. As Strauss and Corbin define it, coding is "[t]he analytic process through which data are fractured, conceptualized and integrated to form theory." (71) The inductive process of qualitative research aims to discover new ideas and viewpoints and ultimately to assemble new theoretical concepts.

One may start with a predefined "codebook" or code manual. This "*a priori*" codebook is a list of codes based upon some preexisting ideas that may be derived from past research or assembled from a viewpoint with which one wants to examine the data. Conversely, one may use a more "grounded" approach. That is, one extemporaneously creates and applies codes to the data as they are analyzed that are descriptive, explanatory or suggest alternative meanings to the phenomenon observed. The term "grounded" comes from the term Grounded Theory, the original work of Glaser and Strauss and is a systematic and specific procedure for analyzing qualitative data to develop theory. (72) However, the phrase, "grounded theory" or analyzing data in a "grounded" manner, has come to be used as a general reference to many inductive methods of qualitative data analysis, whereby the ideas are generated solely from the data, without any pre-existing viewpoint. (70)

In this research, I coded the data in three ways. First, I analyzed the data with an *a priori* coding scheme based upon Norman's Theory of Action in an attempt to look at the data from a cognitive point of view. Next, I used another codebook that I developed based upon usability heuristics that were derived from the literature. By doing this, I hoped to understand the specific usability issues that were occurring in my observations of users. In some respects, this might correspond to considering the data in terms of "cognitive engineering" with the first coding scheme, and narrowing the focus to "usability engineering" in the second. Lastly, I coded the data with a grounded approach and to apply ideas that spontaneously occurred to me that the other two coding schemes did not satisfactorily explain. These three different coding processes were not done with three separate passes, but were performed simultaneously as the data were reviewed. They will now be explained in detail.

4.6.6 "Cognitive" coding

The original process of Ericsson and Simon suggests that an *a priori* coding scheme should be developed from a task analysis and a preliminary analysis of the verbal protocol.(41) I started an *a priori* code book based upon Norman's cognitive framework and it was generated as follows.

In my original idea for analyzing the data, I intended to understand the mismatch between the user's mental model and the designer's model of the computer system. I posited that, within each data segment, the enlightening information is the mismatches, or the segments in which the initial user goals go unmet. If the goal was not met, I hoped to identify which stage or stages of user action in the Norman model that the process went awry, allowing a mismatch between the user's internal representation of the goal and the external representation of the state of the computer system. Each segment was to be preliminarily coded according to the stage of user action that is the source of the problem. These codes are listed in Table 4.

	CODE NAME	Meaning of the code
GULF OF EXECUTION	Forming the **Intention**:	The decision to act so as to achieve the goal.
	Specifying the action sequence:	The user must map the psychological goals and intentions into the desired system state. He must determine what physical manipulations of the system are required to achieve that state.
	Executing the action:	The user comes in contact with the interface mechanism of the physical system.
GULF OF EVALUATION	**Perceiving** the system state:	The interface display changes as a result of the physical manipulation and the user senses that.
	Interpreting the state:	The user interprets the meaning of that perception, what the displays is communicating about the state of the system.
	Evaluating the state:	The user compares this interpretation of the state to the expectations of what it should be, based upon the original goals and intentions.

Table 4. Codes for stages of user activity; cognitive issues

However, when analyzing the data, I found that it was uncommon that the users' goals were not ultimately achieved. Slips and unmet goals were seldom identified, and the users also sometimes worked very quickly, smoothly and efficiently with no noticeable problems at all. However, although users' intentions were eventually carried out, users often experienced challenges. There were many problems observed and rather than focusing on unmet goals, I chose therefore to code those problem segments that demonstrated "cognitively interesting phenomena" that included irritations, struggles, backtracks, workarounds, forgetting as well the occasional slips and even errors that did occur. These phenomena often obtain their meanings as "cognitively interesting" because of their roles in advancing or hindering completion of the Action Cycle and so they were still coded with the above schema, matching the part of the User Action Cycle in which the phenomenon seemed to occur.

4.6.7 "Human Computer Interface" coding

In addition to identifying the stage of user action in which user problems ("cognitively interesting phenomena") occur, one should be able to identify a corresponding aspect of the human-computer interface (HCI) that may be implicated in this problem.

Past research in analysis of usability testing, in both the usability engineering literature (34, 55) and the medical informatics literature regarding clinical information systems, (28, 33, 55, 62, 64) has generated lists of interface elements that may be responsible for usability problems. This literature literally describes entire lists of heuristics, or "rules of thumb" to consider when designing user interfaces. There was much overlap among these items and I distilled them into a single collection.

I grouped this collection of usability heuristics into a "usability" coding scheme. I grouped the user interface heuristics into six categories: 1) *visual*, those aspects that are seen on the computer screen, 2) *navigation*, physical behavior used or needed to move around and locate items on a graphical-user interface, 3) *content*, information about the task or computer system, 4) *temporal*, items that are related to time, 5) *consistency*, the notion that similar tasks and actions should be represented similarly in more than one situation, and 6) *error*, the capacity to prevent and recover from errors. These codes are outlined in Table 5 with additional subcategories.

Category	Code	Subcategory
Visual	VIS	
	1	Graphics & text – understanding their meaning
	2	Visibility (colors, contrast, attention)
	3	Organization
Navigation	NAV	
	1	Moving around GUI (Graphical User Interface)
	2	Locating items on GUI
	3	Data entry
Content	CONT	
	1	System state/feedback
	2	Past actions
	3	Medical knowledge
	4	System knowledge
	5	Cognitive overload
Temporal	TEMP	
	1	Response time
	2	Chronological data entry
Consistency	SIST	
Errors	ERR	
	1	Recovery
	2	Help availability

Table 5. Coding scheme that applies a human-computer interface category responsible for the usability problem.

4.6.8 "Grounded" coding.

As stated, in addition to coding the segment with respect to the Norman model, I then applied these usability heuristic codes to each of the segments as appropriate. These two schemes of *a priori* codes were thus used to analyze the data. However, with analysis of the data, additional codes became apparent and were added to the list as warranted. This is standard practice in qualitative research in general (and protocol analysis specifically) (41), which emphasizes the iterative nature of research design. Expanding the codebook with entries that are more grounded in the data added depth and fidelity to the analysis. Some of these "open" or "grounded" codes are listed in Table 6.

CODE	Explanation
Mindless Entry	Data entry with less minimal forethought
Trust/dependence	Issue in which the user distrusts or overly trusts the system
Resign	The user give up and quits trying or settle for something short of the original goal
Led down the path	The user seems to be misled once a certain path is undertaken, or it appears the UI is leading the user.
Guessing the Action Specification	The user appears to be figuring out the course of action as he goes along. At time seems to be making things unexpectedly.
Work patterns	Different ways to do the same thing, workflow differences
Order set	Phenomena related to the order set
Frustration	User emotion
LTM	The user needs to rely on long term memory; he cannot specify an action for a task that has been done many times before.
Ignore	The user appears to ignore items on the display.
Synonyms	Phenomena related to the use of or need for synonyms in textual display
Tolerance for usability problems	The apparent willingness to tolerate repeated usability issues
Forgetting	The user seems to forget.
Gulf of Execution	Problems with the first three steps of the Norman model
Bad defaults	Problems with default values
More work	The computer creates more work for the user.
Bad alerts	Problems with alerts and reminders
Opt in/opt out	Issues of preselected items in order sets
testing artifact	Phenomena related to user testing and may not reflect real use
Selection of users	Issue related to user selection
The cognitive model	Thoughts on the model
Methods	Thoughts on my methodology
System factors/patterns/attributes	Issues that may be related to one system over another
The computer as a clinical tool	How the clinician uses the computer as a tool to practice medicine
CPOE use today	Issue specifically related to CPOE.

Table 6. "Grounded" coding scheme.

4.6.9 Beyond coding, further analysis leads to results

Coding is merely the process of applying labels to selections of raw data, but this alone does not produce "results." To do that, the data segments are grouped into collections according the codes that have been applied to them and considered within the context of one another. Based upon this analysis, the collections may be regrouped, clumped or split as the data are reviewed iteratively. Ultimately, when one considers the ideas of one collection that similarly reflect the data, one may discover new ideas. The groups are compared and contrasted and the analysis is meant to discover heretofore tacit or newly defined concepts.

In this research all segments were coded considering all three coding schemes. Some segments were coded with more than one code if they reflected more than a single coding concept. The collections of segments all labeled with the same code were considered as a group. Codes were changed or added to some segments if applicable. The Transana software aided in this analysis by generating reports of all the segments that were coded the same way. The groups were combined, split or rearranged with each iterative review of the data. Ultimately the collections of data were scrutinized, interpreted and developed into the Results presented below.

4.7 Strategies for Validating Findings

There were some inherent difficulties in maintaining reliability and validity in this research. As with all qualitative research methods, there are methods for maintaining reliability, or more accurately, trustworthiness. One of the most powerful strategies in qualitative research, using multiple observers, was not possible in this solo research endeavor. I reviewed my findings with colleagues and made informal reports to my site contacts and users to confirm my interpretation of the data. A complete audit trail of the data that were obtained was maintained at all times.

With respect to the validity of the findings of data analysis, think-aloud methods have their unique issues. Understanding the task and its context is an important source of removing the ambiguity of verbal data in order to make explicit the goals of the participant. My experience as a physician and a researcher of CPOE was invaluable here. Conversely, this experience may have been a liability as it may cause a tendency to interpret data as a confirmation of hypotheses and inferences about the user's intended ideas, but this was kept in mind.

4.8 Summary of methods

I observed the behavior of physicians as they use computerized order entry systems during the course of their workday. This behavior was recorded during think-aloud sessions in which they verbalized their thoughts and activities. The words they spoke, the actions on the computer display and the image of them working were the raw data and they were analyzed qualitatively. The data were coded with three coding schemes and iteratively analyzed for emergent themes. The results of that analysis are presented in the next section.

Chapter 5. Results

5.0 Introduction

The results of this data collection will be delineated in the following way. First, data relating to the users, the model and the methodology will be presented and secondly, the information obtained through the coding of the data from observation of users in the think aloud sessions will be described.

This second part will be presented in the context of the coding schemes. The three different coding processes will be the framework of the study results. The results of the observations will be presented through the frameworks of the cognitive codes, the usability heuristics codes and the “grounded” codes that surfaced during analysis. The interrelationships of these coding schemes will be expanded upon and the most important ideas that emerged will be discussed.

5.1 The Users

5.1.1 Selection of users

As discussed in the Methods section, user selection procedures were different in each hospital system. Prior to understanding the local differences of each hospital system, I intended to simply send out recruitment letters and emails and await the response. With time, I learned that more varied and creative methods needed to be employed to obtain the number of participants needed. I found that I needed every possible person who agreed to participate. The limits of participant willingness and the constraints of the user task that was chosen – admission order entry, a task whose timing cannot be predicted – made the selection of users and the observation sessions less flexible than originally expected.

5.1.1.1 Hospital System 1

In Hospital System 1, I obtained the most current user data through the CPOE implementation team. Three months prior to my study, 121 users had been trained on the CPOE system. Many of those users were Internal Medicine resident physicians training at that hospital as well as Family Medicine and Internal Medicine residents who rotate through that hospital from other training programs. I specifically did not want physicians in training as part of this study and those were excluded. The remaining 54 physicians were contacted by letter and email. From the system data provided by the implementation team, I found that the extent of use varied greatly in that group. For example, in the last month of collected data, the number of orders entered per user ranged from zero to 1702. Many of the 16 users having entered no orders were members of the faculty and deferred all order entry to the residents, and were thus not appropriate for study.

The initial response to my recruitment letter was tepid, so I chose another tack. The bulk of CPOE use in the hospital was done by a hospitalist group and I made a presentation to their monthly meeting. After I had received some agreements to participate, I secured a location in the hospital where I remained until receiving a call from a user that a hospital admission had occurred. This was not the most efficient use of time as the admissions are random events. Another group presentation as well as regular emails and pager text messages based upon the call schedule were finally productive of 11 user sessions observed, all of whom were hospitalists. This entailed 21 vigils at various shifts over a 36 day period. The time of day most productive of hospital admissions was between 12:00 noon and 9:00 p.m.

The CPOE system at this hospital was the Horizon Expert Orders™ (McKesson Corporation, San Francisco, CA). The system was initially implemented 18 months prior to this study as a limited pilot project on a Physiatry Rehabilitation unit. The system was extended to a limited number of medical floors 12 months prior to the study and the bulk of users were the hospitalist group employed by the hospital system and resident physicians. The hospital was using both electronic and paper orders concurrently and approximately 25% of all orders were entered electronically at the time of this study.

5.1.1.2 Hospital System 2

Taking lessons learned from this first system, I sought to streamline the process in observing Hospital System 2. I presented my study to a multidisciplinary Informatics Committee, six months and one month prior to my site visit there. Through the medical staff office I obtained contact information for the entire Medical Staff and based upon medical specialty, I focused my recruitment on those most likely to use order entry. I mailed 202 recruitment letters followed with a supplemental email in three quarters of the cases, those who has an email address. Further recruitment was aided by an implementation manager and the Computerized Provider Order Entry Medical Director, himself a hospitalist user. Flyers with pull tabs were placed at strategic places throughout the hospitals (none of the pull tabs were separated from the flyers; I assume these were ineffective). The four members of the CIS support and training team were helpful in introducing me to Medical Staff physicians throughout the hospital. A second round of follow-up emails was sent, focusing on certain persons and group practices based on the advice of the various contacts. I received an affirmative response from 16 persons. As with the previous site, I acquired an in-hospital location and remained there for 13 consecutive days, reminding potential users with emails and text pages based upon my positive responses and the call schedules of the adult and pediatric hospitalists. I was able to observe nine sessions of eight different users.

In this site, the CPOE system used was PowerOrders® (Cerner Corporation, Kansas City, MO). This hospital had a unique situation in that it was newly opened 23 months prior to my study and designed to be a “nearly paperless” hospital, with the goal of total computerized order entry. At the time of data collection, 79% of all orders were placed electronically, 11% were telephone and verbal orders and the remaining 10% were entered in writing.

5.1.1.3 Hospital System 3

The last CPOE system was observed at two hospitals within a multi-state hospital system. In both of these systems my local contacts directly communicated with and recruited users. The use of CPOE was limited in both of these hospitals, but remained my last possibility for observation of in-patient CPOE systems that was geographically feasible. In the first hospital, CPOE had been in place for approximately two years. Use of this application was limited to less than ten physicians, a group of neurologists and the four members of the Physiatry/Rehabilitation practice. Using CPOE for admission orders was limited to the latter group, of which only two are available for admissions at one time.

During a three day visit, I observed a single physician entering admission orders on three different occasions. In the second hospital in this system, CPOE had been implemented also for approximately two years. There had been twenty physicians trained in total over that period, but at the time of my study there were only six active users. Over a five day period, I observed five users entering admission orders. Thus in Hospital System 3, I observed eight sessions among six users.

In this last hospital system, the CPOE system was GE Carecast™ (Now GE Centricity® Enterprise, GE Healthcare, Chalfont St. Giles, UK). CPOE use was limited to the extent described above. Data regarding the proportion of all orders entered electronically at these last two sites were unavailable. Table 7 summarizes the number of users and sessions observed in all.

	Users	**Sessions**
CPOE System 1	11	11
CPOE System 2	8	9
CPOE System 3	6	8
Total	**25**	**28**

Table 7 Users and usability session observed (counts).

5.1.2 The user survey.

It was the original intent of this study to administer the "Computers in Medical Care (modified)" Survey prior to observation of CPOE use. As this instrument seeks to understand computer users' sophistication and experience with computers in general and with their respective CPOE application, I hoped to study the usability issues that arose comparing the users in either end of the survey results spectrum, i.e., the "novices" compared to the "experts." I intended to administer the survey to all potential volunteers and then proceed with the think aloud observations with samples of users of differing computer sophistication. As noted above, my users consisted of literally every possible volunteer I could find, so no purposive sampling (with respect to user sophistication) was possible. This causes obvious issues with the interpretation of the data to be expounded upon in the Discussion section.

5.1.2.1 User Demographics

Nonetheless the survey did yield some interesting results. Of the 25 users, 15 were male. Their ages ranged from 29 to 59 with a median age of 39. The users came from eight different medical specialties, the majority, 13, of whom were self-described as hospitalists. Five users described themselves as internists, though three of these were also in the hospitalist practices, bringing the total number of hospitalists to sixteen out of twenty five. The remainder of the specialties was as follows: Obstetrics-Gynecology (2), Critical Care Medicine (1), Nephrology (1), Radiology (1), General Surgery (1) and Physiatry/Rehabilitation Medicine (1). See Table 8. Thus, most of the users studied were from medical (as opposed to surgical) specialties.

In which area of medicine do you specialize?		
Answer Options	**Response Percent**	**Response Count**
Critical Care	4.0%	1
Hospitalist	52.0%	13
Internal Medicine	20.0%	5
Nephrology	4.0%	1
OB/GYN	8.0%	2
Physiatry, Rehabilitation	4.0%	1
Radiology	4.0%	1
Surgery, General	4.0%	1
Total	**100.0%**	**25**

Table 8. User specialties.

5.1.2.2 Computer Use by Participants

The users stated that in a typical week they used a computer, hands on, for 5 to 50 hours (median 25 hours.) When asked about prior computer training, 76% stated they had "none" or had "self-guided learning." Their self-described computer sophistication was rated by most (60%) as "neither sophisticated nor unsophisticated" and nearly all of the remainder rated themselves as "sophisticated." A single user called himself "very sophisticated." Thus everyday computer use was fairly frequent, but users rated themselves of average sophistication and had essentially no external general computer training. See Tables 9, 10 and 11.

The participants had used their own CPOE systems for 3 to 25 months (median 12). With respect to their own institution's CPOE system they rated their computer sophistication higher than with computers in general. Only 5% described themselves as "neither sophisticated nor unsophisticated," 64% as "sophisticated," and 16% as "very sophisticated." See Tables 9 and 12.

	In a typical week, how many hours do you use a computer, hands on?	**How much experience with your institution's CPOE system have you had (in months)?**
range	5 to 50	3 to 25
median	25	12
mean	25.4	14.5
SD	14.36	6.70

Table 9. User data.

What kind of training or experience with computers have you had?		
Answer Options	**Response Percent**	**Response Count**
Formal course(s) in computer science or related field	8.0%	2
Formal medical school training in computers	0.0%	0
Formal residency or fellowship training in computers	0.0%	0
Formal workshop or conference on computers	16.0%	4
Self-guided learning about computers	60.0%	15
None	16.0%	4

Table 10. User training and experience.

On the whole, how sophisticated a computer user are you?		
Answer Options	**Response Percent**	**Response Count**
Very sophisticated	4.0%	1
Sophisticated	36.0%	9
Neither sophisticated nor unsophisticated	60.0%	15
Unsophisticated	0.0%	0
Very unsophisticated	0.0%	0

Table 11. User general computer sophistication

With respect to your institution's CPOE system (order entry system), how sophisticated a computer user are you?		
Answer Options	**Response Percent**	**Response Count**
Very sophisticated	16.0%	4
Sophisticated	64.0%	16
Neither sophisticated nor unsophisticated	20.0%	5
Unsophisticated	0.0%	0
Very unsophisticated	0.0%	0

Table 12. User sophistication with respective CPOE system.

Lastly, the users were asked about the extent to which they used a computer for specific tasks, rated from “Never (1)” to “Always (5)”. The users most frequently rated themselves mid-way in the scale, as “sometimes,” when using a computer to communicate with colleagues, obtaining advice on diagnoses and therapies, professional writing, and preparing presentations. They generally “often” used a computer for searching the medical literature. They less frequently used a computer for teaching or performing statistical analysis. See Table 13.

To what extent do you personally use a computer for each of the following tasks?						
	Rating / answer option					
	1	2	3	4	5	
Task	Never perform this task	Perform this task, but never use a computer	Sometimes use a computer	Often use a computer	Always use a computer	Rating Mean (1-5)
Communicating with others	0	0	8	15	2	3.76
Obtaining advice on diagnosis or treatment	0	1	5	16	3	3.84
Writing	5	0	6	13	1	3.2
Preparing presentations	5	0	6	8	6	3.4
Performing statistical analysis	15	1	3	6	0	2
Searching the medical literature	0	0	1	15	9	4.32
Teaching students and residents	8	2	7	8	0	2.6

Table 13. How the participants use a computer. Participant counts are given under each rating/answer option. Average rating for each task is given in far right column.

5.1.3 Subjective User attributes

Some observation of user behavior is not strictly related to the activity of entering orders. In addition, comments are made while thinking out loud not directly related to the activity at hand. This should be considered as important as any other data.

5.1.3.1 Computer skills & styles.

Despite a relatively uniform self report of computer sophistication there were some striking differences observed. Some users are clearly much faster than others. Some are intimately aware of the nuances and possibilities of the application, moving from one task to another rapidly and flawlessly; others, while equally experienced, approach the task tentatively, as though some actions are being done for the first time. In the applications where data could be entered by keyboard or mouse, the entire spectrum of each modality, used alone and in combination, was seen. Despite the availability of the order set, some users chose to eschew those and preferred to enter each order individually.

5.1.2.1.1 The order of order entry.

It is naturally expected that different users would approach the same task differently, but the specific choices the users made compared to one another was interesting. One difference that was frequently commented on was entering the different components of the order set in a different sequence. Users often divided the orders into categories of "routine" and the remainder. For example, after first entering orders specific to the current admission, one user submitted those and started over. Here is a quotation:

"Then I always go back in and enter the orders again with all the stuff that I consider kind of 'fluff' and part of general orders."

Those which fell into the "routine" or "general" category varied among users, but often were the more mundane orders of admission status, activity, diet and the like. Routine "prn" medications (those that were used with every hospitalization) were lumped together, distinct from those medications and tests that were specific to the present hospitalization. The users made a point that they preferred to do one category before the others, explaining that they wanted to get one "out of the way" ("eating dessert first"). Even when many admission order sets are arranged in the order of the ADC VAAN

DIMSL sequence, many users still varied this, such as entering “admit” (A) first, last or mid-between. While most seemed to do this because of unexplained personal preference, others did this to assure the most important orders were entered for certain, either out of fear of forgetting or technical problems. Here is a quote from a user:

“And then, usually what I do also, before I get into the order sets, is put in the patient’s routine medications, especially if someone has a really long list. … But because my experience has been, if I go into the order set and try to do it at the end, if I get kicked out, then I lose all that work. And that part’s the most time-consuming part, putting in a long list of meds.”

Thus the users had specific ideas about which orders should be entered first and which should be grouped together. This has implications for interface design.

5.1.3.2 Compensation and user interest in computers.

In an effort to attract more users and as fair remuneration of the valuable time of a health professional, I compensated each user $75.00. When literally handing them this payment, I noticed that most acted hesitant or surprised and many attempted to refuse it. While this may be due to cultural behavior, I suspect that the compensation was little motivation for them to volunteer to be part of this study. My sense is that these persons had a genuine interest in the intellectual pursuit of this research as well as an affinity for computers. I do not think that I would have attracted a greater response had the compensation offer been larger. My point is this: I feel that I observed nearly all possible participants from my recruitment attempts. Moreover, as the survey results suggest, the current users of CPOE in these settings are those who are interested in and enjoy computers and even perhaps informatics.

5.1.3.3 Clinical, not computer orientation.

Despite a possible affinity for computers, even while in the midst of a think-aloud usability testing process, the users persisted in addressing the task from a clinical, not a technical point of view. Although given instruction to verbalize their observations, thoughts and actions, the users commonly described the process in clinical terms. Instead of selecting a tab or typing in a word, they spoke of wanting to order a medication needing to check a lab test, or being concerned about a complication.

Many users spoke as though they were "presenting" a case to a colleague instead of describing the activity of using a computer. An example of this clinical orientation is given. The user was "thinking aloud" and it is apparent what he was thinking about:

"I am entering orders on a lady who presented to the ER with respiratory failure. She actually presented in extremis, was intubated in the emergency room. Post intubation, her exam revealed diffuse wheezing in all lung fields. She was hypotensive, and her white blood cell count was 13,000…"

While this may affect the quality of the data in a think-aloud usability study, this additionally demonstrates the strong clinical orientation of the users of these applications. They are using the computer as simply another tool for practicing medicine. This has implications to the extent that this user perspective differs from the software designer, the local implementation team or the training staff, who might consider the process more from an informatics standpoint than from the user's more clinical point of view.

5.2 Methodological Issues

5.2.1 Purposive sampling.

As mentioned above, the selection of participants was not done in the purposive manner intended by the original study design. My ability to make any purposive choices about the participants was significantly limited. This was due to a combination of problems: 1) a scarcity of users (in absolute numbers, little variety of computer sophistication and the number of clinicians that enter admission orders), 2) the limited willingness of users to participate and 3) the problematic timing issues specific to the task that I hoped to observe (I needed to be present in the hospital when the one of those willing to participate was available for admissions and remain there until the essentially random event of a new hospital admission occurred). My original study design was also based to an extent on a misconception of the particular user distributions in these individual institutions. I was of the understanding that there were more users available to study. Given all that, my observation of 28 sessions of 25 users was by all accounts representative of the current CPOE usage in these particular health care systems as related to the task of entering hospital admission orders. Indeed, in one of the systems I observed quite nearly every clinician who enters admission orders electronically.

Lastly, aside from the selection issues, the number of participants employed was adequate to uncover a significant number of usability concerns, compared to standard usability testing (73). Typically, less than 10 users is an adequate number. There is no conspicuous reason why that standard would not be valid for an application in this domain, but that is not certain and will be discussed.

5.2.2 Technical issues.

Technical problems were few. I needed to interrupt two of the early sessions to adjust the recording speed of the screen capture application when it displayed an error message. This happened within seconds of starting; the session was restarted; and the problem did not recur. In System 2, the system crashed once and the user could not continue. In the interest of patient safety, she left to continue her work elsewhere. That occurred early in the session and the recording was not included in the data.

In Hospital System 2, although I was using an Internet-based access to the CPOE system, not the typical in-house access, the users all agreed that the connection experience was identical to that of the LAN.

In System 3, during the three sessions in which I installed the screen capture software on local machines, there was a significant reduction in computer response time. The IT staff was consulted and the problem could not be remediated. The user clearly noted it and remarked about the problem freely. Any issues related to this UI problem are not included in these results.

5.2.2.1 Clinician as data analyst.

Qualitative research assumes an intimate relationship between the researcher and the data. Far from the isolated and controlled perspective of quantitative study, the qualitative researcher himself is a "tool" in the data gathering and analysis processes. In this endeavor, my experience as a clinician proved to be valuable, even essential. I transcribed the audio recordings myself. This enabled an understanding of nuances of the verbalizations as well as an opportunity to organize the data from the outset that facilitated subsequent analysis. This advantage was minor in comparison to the task of interpreting the screen capture data. It seems necessary to be not just a clinician, but a physician to understand what these users were doing as they entered electronic orders. As will be described below, the users' "think aloud" verbalizations were less prolific than desired for optimal interpretation of their thoughts and action. Much of this had to be inferred by careful, detailed and repetitive analysis of the video recordings. Although, some activity was initially confusing, I ultimately felt completely confident in understanding exactly what action they were performing and what their intentions were with these verbalizations. I rarely observed a single instance where it was not completely clear to me what the goals or intentions were or what user activity was occurring. I do not think this would have been possible without an understanding of the underlying clinical circumstances. This speaks to the role of physicians in the evaluation and design of these systems, to be elaborated on in the Discussion section.

5.2.3 The Norman Model.

Given the wide acceptance of the utility of the Norman Human Action Cycle as a model for understanding human-computer interaction, it is not surprising that it was highly applicable in this domain of clinical information systems. The Model was used as a framework for one of the coding schemes, and the users' observed actions seemed to inherently fit into the model. In fact, I was unable to fail to map a single instance to the model. This may not be a surprising discovery, given the flexible nature of the model and the empirical observations of its validity in a wide variety of human-computer interaction research (74, 75). The extent to which it is useful in understanding issues of usability of CPOE is a major point of this research and will subsequently be discussed at length.

To briefly review the model (Figure 4), the user:

- Forms an *intention* to achieve some goal.
- He mentally plans his *action specification*, the physical manipulations of the computer that will achieve that intention, and
- He actually *executes* that action by touching and manipulating the computer.

As a result of that, the system state changes, as reflected on the display.

- The user *perceives* a change in the system,
- *Interprets* the meaning of that change, and
- *Evaluates* whether the initial goal/intention has been met.

As assumption of this model is that the user starts with a goal, a state he wishes to achieve and proceeds through this cycle in an effort to fulfill that goal. My observation with this computer application is that the "entrance" into that cycle may be different, or at least the interpretation of that cycle may be so. When it is the display that gives the user suggestions on how to proceed, such as in an order set or any number of clinical decision support features, it sometimes becomes unclear what the intention is or where it is initiated. Consider the following example:

A user is reviewing a series of check boxes on an admission order set:

- ☐ Daily weights
- ☐ Intake & output
- ☐ Oximetry

From the transcript as the user scans these checkboxes:

> *"Then, I don't need to weigh this guy at all. I's and O's are not that important. Oh, I'll stick them in anyway. Give the nurses something to do. I don't think we need oximetry. He's only 38 years old."*

It appears from the user's words that he is initiating his goals from what he reads on the screen. The user is starting with a "perception" of the check boxes, he "interprets" its meaning and "evaluates" whether he wants to do it or not. This is different from an intention initiating the sequence. The user does not want to order "daily weights" on this patient. Is the original intention "Do the correct admission orders" or "Don't order daily weights"? This is illustrated in Figure 5.

		#1	#2
Intention:	Does the cycle start here?	Do the correct admit orders (“no weights” are one of them, but I wasn’t thinking of that specifically.)	Don’t order daily weights
Action Specification:		scroll down and look at what’s there	don’t check the box and go to next task
Execution:		mouse at scroll bar	scroll to next check box
Perception:	Or does the cycle start here?	**see unchecked “weight” check box**	
Interpretation:		it is unchecked, so that means it will not get done	
Evaluation:		It is as I want it, unchecked	

Figure 5. Two versions of Norman’s Human Action Cycle, showing two different ways in which “Do not order daily weights” may be mapped to the model.

Does the cycle begin with intention, or does it begin with perception? It appears that order sets in particular (or decision support in general) may alter the human action cycle process, by having the interface propose goals, and the user must decided whether the goal is desirable or not. The implication of this will discussed later.

5.2.4 The Think Aloud Method.

The value of any data collection method is related to how precisely it is carried out and perhaps how faithfully it remains to any theoretical underpinnings. In the think-aloud method, as described in the original work of Simon & Ericcson, the data obtained are intended to correspond to the cognitive processes that occur in working memory (41). In order to assure that the most useful data are accessed (Level 1 and 2 verbalizations, see 2.1.3), the instructor must be as unobtrusive as possible so as to not interpose any cognition that is required for instructor-user communication with the thought processes of using the system. Training the user prior to the session relates to a more productive process, but, as described above, I was unable to carry out extensive training sessions. Nevertheless, as the instructor, I attempted to remain as unobtrusive as possible, limiting my prompting to the recommended phrases of "keep talking" or "tell me what you are thinking." In the absence of training, this "laissez-faire" approach may have not been the most prudent.

The essence of the data collected with this method is in the verbalizations of the users. As a rule, these users simply did not speak up as much as would be necessary to understand their cognitive processes by verbalization alone. One hopes to have a near constant stream of verbalization throughout the session, but this was not nearly the case with many of these users. As the study progressed, I decided to be more involved and prompt the users more vigorously, but it seemed to help little.

Certainly the lack of pre-session training could be implicated in the taciturn behavior of the participants, but perhaps it may be related to the nature of the clinician computer user. In addition to the paucity of verbalizations, the nature of the speech was decidedly clinical, as mentioned above. What I noticed in these physician users is that often the think-aloud verbalizations are less about computer interaction and more about clinical thought processes. They spoke about why they are making the choices they are making as a physician, not as a computer user. They rarely mentioned clicks, keystrokes or what they were visualizing on the screen. They might say, "I want an adult order set," but they are typing in the letters "a-d-u-…" In a perfect "think-aloud world" they would saying "I am typing in the letters ‘a-d-u-…’” They spoke in terms of actions of what they are doing to complete their work, most often with a clinical bent. They described actions such as “I need to check those labs again in the morning,” rather than explaining their interactions with the computer. Despite my exhortations that they explain what they are doing on the computer, they mostly continued heedless of my requests. It appeared they almost “forgot” they were using a computer. One user specifically denied she was not talking out loud. In one segment, while she checked off orders for activity, bed rest, IV fluids, vital signs, daily weight, intake & output, a Foley catheter, nasogastric tube and restraints, her only verbalization was:

“OK, that gets me into critical care, so we’ve got check-boxes which I’m going to go ahead and check. Basic items.”

She then remained silent as she proceeded to check off all those orders, so I urged to tell me what she was doing. To which she replied:

“I’m not actually doing anything. Scrolling down.”

Observation of the video data demonstrated she was “doing” quite a bit.

Interestingly, my three pilot users much more successfully verbalized computer activity than did the actual users. This may be that those in the pilot sessions were all trained informaticians and they thought more about the computer. This orientation of the clinician toward the clinical workflow as opposed to the computer workflow may have implications in interface design and user training. It certainly has implications in this type of usability testing with physician users and the possible importance of pre-session training.

On one hand, this reduced verbalizing may have been a problem, inhibiting my ability to understand their cognitive processes. However, I feel confident, based on the scrutinization of the non-verbalized screen-capture data, that I was able to get a clear understanding of their actions. The intentions always seemed clear to me. Perhaps when verbalizing less, they were more completely immersed in computer usage and thus were more closely mimicking their "true" field behavior. This may actually allow a more accurate understanding of their order entry behavior.

5.3 Results from the Initial Analysis of Think Aloud Data

The final video edits of the 28 user think-aloud sessions resulted in four hours, 20 minutes and 4 seconds of video data for analysis. These recordings were transcribed and produced 28,580 words of text. The first step of analysis was a process of dividing these lengthy sessions into manageable "chunks," the process of "segmentation" discussed in the Methods section. While mostly the purpose of segmentation is purely practical, the reduction of unwieldy amounts of data into pieces that can be addressed more easily and understandably, the process also produced some other interesting findings.

5.3.1 Segmentation

5.3.1.1 User and systems differences affecting segmentation

As discussed in the Methods section, the intention of segmentation is to parse these large volumes of data into usable "statements." I chose to divide the data according to individual orders. The overall task for these users was to enter the list of all the admission orders, so dividing them into individual orders seemed an obvious and practical choice. The complexity and duration of the task of entering a single order was manageable enough to handle as a single "proposition" or "assertion" such that it could be coded in a reasonably succinct way. That is to say, there was enough substance to each order to comment on it, but there were not so long or complex that they could not be assigned brief codes that explained them.

The segments roughly corresponded to a single electronic order each. Most of the time it was clear how to segment the data, as the order generally stood out as a discrete task. The tasks were relatively easy to categorize as one of "ADC VAAN DIMSL" steps, plus a few others. This was not always the case, however. Sometimes the order was so brief (a second or two) that it was technically difficult to capture as a single segment, but also was there not much meaningful data in such a small fragment. Therefore, sometimes more than one order was grouped into a single segment. Often users would talk about one task while they were physically performing another, making it impossible to isolate the audio and video of a single task. Other times tasks were overlapped and interleaved; one task was suspended to perform another and then the user completed the first.

The interface of the respective CPOE systems also made a difference. An order may be accomplished with a single click or with a series of steps. In one system, many orders had to be done "twice"; the order was entered with one action and later on a confirmation screen popped up and needed to be addressed to complete the order.

Thus, there was some difficulty in segmenting the data into chunks that represented a single order, though this was in a small minority of cases. If this was not possible or even if I performed the segmentation somewhat arbitrarily at times, it is not of great importance to the integrity of the analysis. The process of segmentation is primarily for utilitarian purposes, to make the data manageable.

5.3.1.2 Quantifying and labeling the segments

The quantitative data regarding the user sessions and their segments are as follows. In total there were 559 segments. The average duration of each entire user session was 9:19 (median 8:12 with a range of 1:54 - 21:42, standard deviation of 4:50) and was composed of an average 20.0 segments per session (median 18.5, ranging from 6 - 38, standard deviation 7.6) As indicated by the range and standard deviations there was a great deal of variability in both the duration of the sessions and the number of segments per session. The reason for this was mainly that the tasks were different between users – some simply has more complicated admission orders than others. There was a great variability in the number of segments and the duration of the session among the users of different systems. This seemed to be related less to the system itself than the type of user, specifically the sessions of the internal medical specialties lasted longer and had more orders (segments.)

In System I, the sessions averaged 22.2 segments and lasted on average 11:24; in System 2 they averaged 22.3 segments and lasted 9:20; whereas in System 3 there were only 14.3 segments on average lasting only 6:20. This variability was not as apparent when comparing only the medical specialties. I included the following as medical specialties:

hospitalist, internal medicine, critical care and nephrology, as the types of admission were similar among each. The numbers of segments for the medical admissions in the three systems were 22.2, 23.5 and 19.0 and their durations were 11:24, 10:05, and 10:18, respectively. Note that System 1 had only medical specialties, hence the numbers do not change after this correction. These descriptive statistics are offered, but inferential statistics were not calculated as they would not be meaningful. In addition, nearly all of the users were medical specialists. Therefore, no comparison is implied between the medical and non-medical specialties. They are shown in Table 14.

Initially, all 559 segments were named as an “order task.” In total there were 593 order task labels, thus some segments were labeled with more than one task name. I have stated that the primary purpose of segmentation was supposed to be utilitarian, so labeling with more than one name was excessive and unnecessary, but not detrimental to the analysis. There were simply times when no one order task code was most obvious or some segments didn’t quite match any one code. Thus, rather than lose information, I chose to “overcode” the data. The order task that was by far the most common was “medication.” This is not surprising, as one would enter an order for medications much more frequently than, for example, “diet.” Describing the frequency of the order task names may be of minor importance. However, the relatively frequency of the ten most used order tasks is given for illustrative purposes in Table 15.

System	no. of users	average no. of segments per session	average duration per session (mm:ss)
ALL	28	20.0	9:19
system 1	11	22.2	11:24
system 2	9	22.3	9:20
system 3	8	14.3	6:20
All medical	21	22.4	10:48
system 1 medical	11	22.2	11:24
system 2 medical	8	23.5	10:05
system 3 medical	2	19.0	10:18

Table 14. The differences between the session duration and number of segments per session among the computer systems and medical specialties.

Order task	count
medication	135
lab test	83
review	52
RN orders/intervention	40
admit	29
diet	27
referral	27
sign orders	26
order set	25
activity	24

Table 15. The frequency with which the segments named with order tasks.

5.4 Coding the Think-Aloud Sessions.

Having completed the segmentation of the user session, I coded each segment using the three coding schemes described in the Methods section. There were two *a priori* coding schemes, the "cognitive" codes that described the stages of the Normal model, the Human Action Cycle, and the "Human Computer Interaction" (HCI) codes, a compilation of established usability heuristics gleaned from the literature. Finally, the segments were tagged with "grounded" codes, those concepts that arose during the analysis of the data.

As above, there were 559 segments in all. The goal of analysis was to look for usability issues that arose, within the framework of the *a priori* coding schemes and any additional instances that were noted. As was mentioned in the Methods section, much of the time, the users' interaction with the computer system posed no problems or usability issues. This behavior is not the focus of this research and was not characterized. It was the "cognitively interesting phenomena" to which I chose to apply the coding schemes, those usability problems that interfered with the efficient use of the system. These irritations, stumbles, struggles, workarounds, cognitive slips and errors are the essence of usability problems and were analyzed as described below.

Out of 559 segments, 249 (44%) were labeled with one of the coding schemes, as such were considered "cognitively interesting phenomena." There were 44 codes that were applied in all. Quantifying these qualitative data may be inappropriate and of unclear significance. However, the most frequently used codes were applied much more extensively than those least used. A list of the most frequently used codes is shown in Table 16. It may be significant that the most frequent codes were in the "cognitive" and "HCI" schemes, though this may be because these two were created *a priori*.

The results presented thus far have been those obtained before any data were coded. I have presented findings related to the users, the user survey and issues related to the methodology. In a way, these are preliminary to what follows. The remainder of this chapter describes the bulk of the results, those that were obtained after analysis of the data from the think aloud session. These remaining findings are the essential essence of this research.

Name of code	"type" of code	Count
ACTION SPECIFICATION – Specifying the action sequence	cognitive	36
NAVIGATION 2 – Locating items of the GUI	HCI	33
PERCEPTION – Perceiving the system state	cognitive	29
INTERPRETATION – Interpreting the system state	cognitive	25
CONTENT 4 - Having knowledge of the system	HCI	20
NAVIGATION 3 – Data entry issues	HCI	19
MORE WORK – The system creates more work	grounded	19
TRUST – trust in the computer system	grounded	19
BAD ALERTS – problems with alerts	grounded	15
GUESSING – inventing action specification de novo	grounded	15
IGNORE – paying no attention to changes in system state	grounded	15
RESIGN – Settling for less than intended action	grounded	15
TOLERANCE - problems common to all/most users	grounded	15

Table 16. Frequency with which certain codes were applied to the cognitively significant data segment. See Tables 4, 5 and 6.

5.5 Results of Analysis of the Think-Aloud Data with Three Different coding schemes.

To review, the raw data of this research are digital audio/video and text files. The video images are recordings of the computer display as a clinician uses a CPOE application to take care of patient. Embedded into that display is an image of the user taken from a video camera. The audio portions are recordings of the verbalizations of the user as he performed the CPOE task. The text files are transcriptions of those verbalizations. With the data analysis software, the text documents are temporally mapped to the images so they may be read at the same time as one hears the audio portion and watches the corresponding visual display. The results of this study arise from observing the video images, hearing the verbalizations and reading the transcriptions in an organized manner.

As described, the data were separated into usable chunks and labeled with codes. After organizing the hundreds of data segments into groups labeled with a single code, each group of data labeled with the same code is reviewed independently and repeatedly. Ideas about user behavior arose through the review of these observations and words. Repeated review suggested combining some groups, dividing others, deleting some segments from the group or adding more codes to other segments. After time, concepts about this phenomenon of human computer interaction arose and are articulated in the remainder of this Results section. The results of this research are lists of ideas that became apparent after analyzing the qualitative data in this way.

These ideas that emerged were named with headings that describe the phenomena. As it turns out, some of the results have the same names as the original codes. This may be confusing. Remember that the codes are merely labels that one applies to the raw data. Codes are not results, most especially the *a priori* codes; they are merely tools that are used to arrive at the results. When grouping these many concepts for the purpose of communicating these results, sometimes the label that is applied happens to be the same as one of the codes. There may be no better word to describe them. This is specifically true in the section on the “Cognitive” codes as I wanted to expound on each of the six steps of the Norman theory by describing the data that relates to them.

The results are listed in three groups. Each group is the product of the data analysis using one of the three coding schemes. The goal of this method was to look at the data from more than one perspective. By analyzing in this manner, every data segment may have been labeled with one or more codes from any or all of the three coding schemes. Thus there was some overlap. A phenomenon that was observed at one point in time may have several aspects to it that may be considered and therefore described in more than one way.

Outside a research endeavor such as this, software developers use think-aloud methods to test users as they work with computer applications. The “results” of that testing are simply the lists of very specific interface issues that occur in that specific application such that they may be corrected or improved. My goals in this research are broader than that. I hope to express some ideas that have more generalizability or at least transferability to other settings. That is why these observations were done with more than one CPOE system and are being analyzed from different perspectives.

By using multiple coding schemes I explored this type of computer use from the perspective of the user’s cognitive processes that are needed and utilized as well as any identifiable interface issues that may be implicated. Then, by additionally analyzing these with a grounded approach I can show how these two other perspectives explain each other and arrive at some more general concepts. Therefore, there may be similar concepts described in more than one of the following sections. Indeed, some specific examples of user behavior are presented more than once. The point is to look at this behavior from different viewpoint with the intention of more fully understanding and explaining it.

Lastly, there are some difficulties that arise when trying to communicate images, actions and sounds with text alone. Some of video images are proverbially "worth a thousand words" but cannot be reproduced in this medium. In that this document is textual only, the actions that were observed will be described verbally and they will be accompanied by user verbatim quotations and occasionally a still image taken from the video recording. The quotations are especially potent and poignant representation of the users' cognitive processes and the experience of using a computer in this setting.

5.6 Results of Analysis Using the "Cognitive" Coding Scheme.

5.6.0 Introduction

Don Norman's model for human-computer interaction, "The Human Action Cycle," is a central framework for the research. As mentioned, the activity of the CPOE user was easily and suitably mapped to the six stages of this model. While this model seems applicable, it remains to be seen whether it is useful in providing an explanatory or insightful scheme for the understanding usability issues related to CPOE.

The terms defining those six stages were used to code the data and the result of that coding is presented here. To review, of the 559 segments, only 249 were felt to be interesting enough to apply codes. The model is used frequently (in other research) because it aptly describes computer use, therefore I could have coded all user activity but that would be utterly unhelpful. I therefore selected the segments in which usability problems occurred (cognitively interesting phenomena), and to the extent that one of the stages of the Human Action Cycle seemed to illustrate or explain that problem, it was coded accordingly.

Using the model in this way, the emphasis is therefore on the cognitive aspects of CPOE use. I have used the steps of the theoretic cycle to organize the usability issues encountered among the users. Of the 249 segments that were coded in all, 83 were coded with these cognitive codes, and 110 of these codes were used in all. Some of the cognitive codes were applied much more frequently than others; *action specification* (the code applied more than any other), *perception* and *interpretation* were felt to characterize most of the segments. For interest, Table 17 shows the frequency of the coding with this scheme. The results of this analysis will be described in the sequence of the steps of the Human Action Cycle.

Cognitive code	Number of segments coded by that term
Action specification	36
Perception	29
Interpretation	25
Evaluation	7
Intention	7
Execution	6

Table 17. Frequency of use of the cognitive codes.

5.6.1 Intention

This is the first step of the cycle. While similar to “goal” this term has a slightly different implication. The goal is the state the user wishes to achieve. Once the user makes the decision to move forward and use the computer to achieve that goal, he forms an intention. It may be a subtle difference, but it captures the concept of volition to do what is necessary to achieve the goal, more than simply the desire.

In this analysis, problems occurring related to intention were few and less important than others. This makes sense, as the user, a clinician wanting to enter an order, can easily form an intention with or without a computer. However, according to the model, human-computer interaction is cyclical, and the user must ultimately decide whether his intention is met. The degree to which this is so forms the basis for his next intention. If the intention is not met, he must change his course of action in order to achieve it. If the intention is met, he must form a new intention. Occasionally, problems occur with forming this amended or new intention based on occurrences in other places in the action cycle.

5.6.4.1 Intentions may be altered.

The manner or the order in which items are presented on the interface may influence the intention of the user. One user was ordering “prn” pain medications and was presented with the most potent medication first, which he dismissed as being more potent than needed. After, progressing and seeing the lesser, perhaps less effective meds, he changed his mind and ordered the potent one. Thus the way the options are organized may have an effect on the user’s intention. The user’s level of understanding of computer processes may also alter intention. Users often entered free text even when the same option in structured data was available and visible to the user. Assuming the user saw that option, users often chose to enter data with free text anyway. Perhaps understanding the technical difference between the two kinds of data would alter that behavior.

Lastly, it should be mentioned that intentions may be blocked appropriately. This effect on intention occurred frequently. This is not a usability problem, but a valuable process provided by clinical decision support. Prevention of drug interactions or drug allergies thwarts the user's original intention and is a deliberate goal of the system.

5.6.4.2 Intentions may be thwarted.

This may occur when there is a limit of possible options to choose from and the ideal option is not there. One user was deciding when to draw a blood test. The options included "stat," "routine," "in the am" and "off blood in lab." The user chose the last option, but didn't know if a specimen was available in the lab. She stated,

"*And, 'in lab if specimen is available'; yep, that's what I want, but I do want it drawn if it's not there. OK, I'm gonna go with 'in lab'."*

There was no way to account for the possibility of not having a specimen available and would require a callback. Any "if-then" statement should be followed with an "else." Another user was forced to claim a patient was allergic to a drug class (which may have been a false statement), when the individual drug in that was not listed. Yet another user wanted to delete one of the two "code status orders," to avoid redundancy. She didn't know which of the two slightly differently worded orders was the correct one to retain. She kept both.

Lack of enough available options or information can stop the user cold. When it is unclear if the intention has been met, the user is stuck. Without adequate information, the user may not know whether the cycle is complete. He doesn't know whether to proceed with the next goal or to modify the current one. In all, however, intention issues were a minor component of usability problems and when encountered, seemed secondary to other cognitive issues.

5.6.2 Specifying the action

Action specification is the second step in the Human Action Cycle. In this step, the user must map the psychological goals and intentions into the desired system state. He does this by determining what manipulations of the system are required to achieve that state. Simply put, he must figure out what actions must be taken with the computer system to achieve his intention from the first step. This requires knowledge of the specific computer system to understand what actions, entering text and mouse clicks on the screen, for example, will achieve his goals. At the very least he must have general understanding of the standard conventions of a graphical user interface or of the workings of similar programs to plan the actions he is about to undertake (assuming those conventions and workings apply to the current system.) This is a critical step in the model, as he must decide what actions to undertake before he can go any further in achieving his goals.

When observing the behavior of the users, many of the problems of usability occurred in this step. As it turns out, usability issues were labeled with this code more than any other (See Table 17.) Difficulty in this step proved to be a source of severe problems for the users, so severe that some actually resulted in the user altering or even deferring the original intention.

5.6.2.1 The user is unsure of which action to specify.

This problem seems like the most generic of all usability troubles, and could be another way of saying that the user has trouble using the system. This occurs in numerous situations such as when the user cannot find what he is looking for. This occurs if the options available are organized poorly, the terms used are confusing to the user or if the options are not easily noticed by the user. Sometimes the option desired simply does not exist within the system. The user may be limited by too few choices (the exact order desired is not listed) or even if there are too many choices. A quote that exemplifies some of those issues is this one:

"Now this one's going to be strength dose 10 instead of 5, and it's going to be instead of Q day, it's going to be Q evenings. I end up creating a 5 in the morning and 10 in the evening, by doing that. And then again, [I have trouble] finding 'evening' again in here. There's about 30 options. For frequency 'every 2 hours' up to 'every 72 hours'. And it's hard to find … here it is. Probably number 27 out of 40 choices."

5.6.2.2 The user can't determine any action to specify.

This seems to be one of the more frustrating usability issues – the user simply doesn't know how to start. The user has a goal in mind and the intention to fulfill that goal, but is stopped in his tracks, unable to proceed any further. Some examples follow. In the first, a frequent user of the system (eleven months experience) was stopped within seconds of initiating the session. He could not find an admission order set, which he needed to start the session. The following were his very first verbalizations:

"OK, all right, so... I do admission orders, [CPOE application]. Bring those up. And, um, care sets. I wonder if there are... any… now where are the admission orders? Wait a second, they're around here somewhere. This is not the form that I am used to. Well, I think there's something wrong here."

Another user (with 24 months experience) was stymied when hoping to insert free text comments:

"And let me see, find in there, so under the social work, I want to put a comment. I'm not sure it gives me a, allows me to make a comment. OK. So, here's a problem right here I'm not even sure I know what to do."

In both instances the users thought they were performing tasks they had done many times before, but they were simply unable to proceed. Other times users simply were unable to proceed because they have not been trained to do that task:

"I want some. Oh, how am I going to do that? I want some potassium in it. But I'm going to go back here for a second. I want some potassium in it, but I don't think I can put it there because I've never done it that way."

Sometimes, the limitation of the computer (actually any free text order even if written on paper) prevented the user from completing the task:

"I don't want her getting overdosed. I'm gonna give a nursing instruction, to watch this rather than ordering… OK, so instruction to nursing, "Please call MD, if patient seems over-sedated or, boy that's too subjective. Let's see, how do I best? Truthfully this is almost something I'd call to the floor and talk to the nurse and say what I really want is that I'm guessing on her Klonopin dose and she's starting on her methadone again, she hasn't had it in a while and I just don't want her to be over-sedated and I don't want her withdrawing either, and if you get a sense of one way or the other, please call me right away, don't hesitate. …Now, mind you, I'm a little bit uncomfortable with how vague that is. The response to that will be very variable depending on the nurse, and the time of day, and everything."

Although this was an example of a complex action that perhaps no interface could solve, occasionally the users cannot complete very basic actions. Other examples were instances in which the user was unable to find any dose frequency instructions (because she looked under "priority" not "frequency") or was not able to initiate order entry as he was unable to locate the "Enter Orders" icon.

Users were very frustrated at not being able to do a task they were certain they had done before. Here a user spent a great deal of time trying to order a bedside commode:

"… let's give him … a bedside commode. Hmmm, that's a long wait for a toilet. [Scanning…] Discharge….suction…. May stand…. Yeah, but where's the one that…. Aha! No… that's discharge equipment. That's not very helpful. Let's try Nursing. Maybe it will be under that. Common nursing orders … Tons of stuff to look at. None of which on first, initial read-through is actually what I want. Now, once again I am getting slowed down by something silly. I don't know how to find a bedside commode in here. I've done it before, so I know it's here."

Occasionally the user did not want to initiate a task out of concern for committing an error. They were fearful of the system and they didn't want to try a relatively novel action:

"There's supposed to be a way that you hit control and highlight them all, but I have found when I try to do that it doesn't always work, … for changing them from STAT to regular or routine. I'm not going to touch it right now because I don't want to screw it up."

When a user didn't know offhand how to specify an action in the system, he often resorted to trial and error. This may be a satisfactory approach if one is very familiar with the system as this may easily lead to the correct action specification. But in general, guessing the action specification is a potential for problems. Here are some examples:

"Strict Ins and Outs… Oh yeah! Hey, that was what I was looking for before and I don't think I found it. So, let's try this. Shoot! There we go."

"So the methadone, I'm gonna do per pharmacy, uhh. Let me see what comes up with methadone. I bet you the pharmacist protocol is one of the things here. Patient enrolled in methadone clinic. Pharmacist. Yayyy! Exactly what I was looking for. [Verbally scrolling down list of meds] Let's see, do I pick it individually? What happens if I push here? Oh, I have to click on them, OK."

"I hit Enter and nothing happened so I must come down here and click somewhere. All right!"

5.6.2.3 The action specified leads to further problems.

Many orders take several steps before the specific desired order is reached. For example, in any menu, the options are arranged hierarchically from the general to the specific. At times, once a user has committed himself to a certain preliminary action specification, en route to the desired action, he can become mired in the process as he has started down a path that will never produce his desired action. Although the user knows he has not found the order he wants, it is difficult to see that the problem is due to previous actions and as such is unwilling to "back up" to an earlier stage and continues foundering an interface location that will never lead to his desired action.

One user had admitted a patient with alcoholism and he had concerns that the patient would undergo alcohol withdrawal seizures. He wanted to order "alcohol precautions."

To start this he typed in “precautions.” This produced a list of precautions, but “alcohol precautions” was not among them. He did note “seizure precautions” which he selected but this was not related to alcohol withdrawal seizures. This led to a series of steps that ultimately failed to find “alcohol precautions.” He gave up and decided to try again later. As it turned out, “alcohol precautions” was to be found under a protocol named “CIWA” which he later located.

Another user wanted to order a series of three cardiac enzyme tests on a patient. The same test was to be drawn every eight hours for a total of three tests. The correct action specification would be to select a “priority” of “routine,” “frequency” of “q8hours” and “duration” of “3.” The user made an initial mistake of ordering the “priority” of “early am” (when he wanted the first of three to be obtained) and a “duration” of “1” (he only wanted one test to be drawn in the early am.) With that he became very confused and tried various actions to correct the problem. Nothing worked. Ultimately he cancelled everything, started over and got it correct the second time.

Both of these usability issues are due to different interface problems, but in both cases, once the user incorrectly started down the wrong path, he was misled and was unable to change his course of action to solve the dilemma.

5.6.2.4 Specifying the action on the computer is more difficult than the corresponding action on paper.

Due to many different interface issues, the users cannot complete their action specification. In these examples, the users lamented that their actions would be simpler to carry out written on paper orders. The quotations demonstrate the users' frustration.

In this first example, the system did not allow the user to adjust a medication dosage based upon a physiologic parameter:

"So what I would love is a way to make this thing automatically titrate up, Q one hour. So if his heart rate's still greater than 120 in one hour, they'll automatically go to 10 mg after they've checked a blood pressure and made sure he wasn't down in the basement somewhere. Now, I wonder if I can just do this under infusion instructions. Let's find out. I'll get a call from the nurses if they don't like it. Um "… for goal heart rate of uh, less than 100. To do this, check BP…" Arrrgh! OK this just isn't going to happen automatically easily. Let's skip it and I'll check his heart rate later. See, if I was writing this by hand I could write out exactly what I wanted and it would …. Ughhh, it feels frustrating."

In this example, the user cannot locate the specific radiology exam he is looking for in the manner it is displayed:

"Now for the big one, we've got to get the MRI, to look for small bowel ischemia.

Oh boy, this could be a real pain. Oh boy, oooh, look at that! This might work…. Unenhanced? See, I don't want unenhanced. Oops, oh that's good, it actually bumped me back to what I was working on last time, rather than going to some random place. Umm, well, shit! Is this what I want? See, this is really frustrating me. I'm getting to the point where I might just abandon this and just write an order in the chart. I want an arteriogram, or an angiogram of her celiac plexus and SMA, and I just don't trust this thing quite honestly. And I think I'm going to write it in the chart. So… that's exactly what I'm going to do."

5.6.2.5 The role of synonyms in action specification.

Another issue that gave the user problems with initiating or completing an action specification is related to the use of synonyms. Many orders are described in multiple ways and users often do not utilize the same terms when referring to an order. If the user seeks an order by a name that is not listed in the system, he will not be able to access what he is looking for. One user wanted to order a cardiac diet. After selecting "most common diet orders," the "AHA diet" (American Heart Association) is the fifth listed. "AHA" is a common enough term, but the user had "cardiac" in mind, an equally common term, and simply did not see it. While look directly at the term "AHA," she says:

"So let's see General Diet. Wait, I want. Let me see a minute. She's not diabetic. I want like a cardiac diet. Where is that?"

She proceeds to type in “cardiac diet” in free text to search for her goal and is presented with a list of 37 diets, none of which is called “cardiac.”

Another user wanted to admit a patient to the Intensive Care Unit. She is a hospitalist and does this frequently. She wanted to locate the “ICU order set,” something she apparently does frequently. She could not find it:

“New orders and I’m going to use the ICU Order set, which should be under Adult ICU, unless they moved it again. … Maybe it’s under ICU. Ah, or critical care. We don’t have a lot of synonyms built in yet so you have to know exactly what to call something to get your orders to come up.”

This was a recurrent problem with these users. They had trouble remembering the proper label for even those orders that are used frequently, even when seeking them with commonly used expressions. This has many implications in interface design.

5.6.2.6 The relationship of order sets to action specification.

Thus far I have given examples of instances in which a user has difficulty with specifying the action to be undertaken. One situation stood out numerous times as being a very effective tool in facilitating action specification and that is the concept of the order set. The order set is a grouping a related orders that assists the user by having commonly used orders together in one place making workflow simpler and serves as a reminder checklist to aid the user cognitively. It was apparent how important the order set was to the Action

Cycle step of Action Specification. The issue of Order Sets will be elaborated later in section 5.8.9.

5.6.2.7 Other codes that were associated with action specification.

Considering computer use with the Human Action Cycle model, it is hopefully apparent that struggles with specifying the action the user must perform to enter the order are a significant impediment to usability. The above exposition described circumstances in which the users experienced issues with that crucial step in the order entry process. The examples demonstrate some situations in which experienced users simply cannot perform the tasks they need to. These segments were thus coded with the term “action specification,” but they were also coded with other terms. There were certain codes that seemed to be more commonly associated with “action specification.” Noting the associated codes provides some explanation of this phenomenon. Under the “cognitive” codes, “perception” and “interpretation” were also often seen. The associated “Interface” codes were “system knowledge” (familiarity with the content of the system) and the “navigation” codes, that is, issues with moving around the graphical user interface, locating items and entering data. The other associated codes were “guessing the action specification”, “synonyms” (both mentioned above), as well as “long term memory” and “resignation.” This all will be elaborated upon subsequently.

5.6.3 Execution

In this third step of the Human Action Cycle, the user comes in contact with the interface mechanism of the physical system. Here the user carries out the actions that were specified mentally in the previous step. The user touches the computer, typing in text and using the mouse. From a cognitive perspective, this physical action is initiated when working memory summons from long term memory the knowledge of how to manipulate the computer combined with the desired action specification and transmits this to the neuromuscular system and the user touches the computer to bring about the desired action.

This step of the Action Cycle was not a great source of usability problems in this study. This is understandable as there are a limited number of allowable actions, and they are, for the most part, the least taxing in this cycle of cognitive, sensory and motor events. The user is either typing in text or activating an interface element by use of the mouse, in which he locates the item on the display with the mouse cursor and clicks. To the extent that the user does not know where to type, what to type or which interface element to click, this is a problem of cognitively specifying the action (the previous step), not of carrying it out physically (this current step of execution).

There were certain interface design issues that made the execution of some specified actions problematic. This fell in to the category of causing the user more work than was desired.

One common problem that created more work to execute is the problem of bad default values. For example, all hospitals have a standard time to draw routine labs in the morning. A common request for a time to obtain a test would be the morning following the day the lab test is ordered. All labs have a "priority" value and a request such as this would be named something like "early a.m." or "tomorrow a.m." In addition to priority the user must also enter the date and time to order the test. One would think that a priority value of "early a.m." would naturally default to a time on the following day and it often does not. The user must additionally enter those values. There were many examples of bad default values that forced the user to execute additional actions.

Another user wanted to enter a time to draw a lab test and clicked arrows on the interface to change it to many hours from the current time. One click of the arrow only changed the time by 15 minutes and it took many clicks to reach the desired time. In this case, the user did not know she could simply "select" the time with the mouse and type in the new desired time, which would have been much faster. This was a lack of system knowledge of allowable actions on the part of the user that caused this problem.

One unique but egregious interface problem caused a user a great deal of extra work. When entering post-operative orders, the user wanted to discontinue all the previous orders, but the system only allowed him to select a maximum of twelve and then click a "D/C" button. Worse yet, nurses were able to select as many as desired and complete the task in a single action. Here is the transcript:

"OK, we're going to put in orders for a primary C-section that we just did and let's see here, it's that one. And I go to order profile and it comes up and it has all my other orders, all the old orders on there, and it's cleaner if those are all discontinued before adding orders, so… I check the ones I want to discontinue. The nurses can check them all and D/C them all at once. I can D/C twelve at a time, which is a real pain in the ass."

Thus, the execution step of the Human Action Cycle did not account for a significant part of usability issues. The extent to which is occurs happens in the context of bad default values and other interface problems that cause the user more work. These categories of problems are discussed in another section on grounded codes.

5.6.4 Perception

After the user comes in physical contact with computer system, the actions that are executed on the machine create some changes in the system state. In this next step, the user perceives any changes in the system state as the interface display changes due to any previous physical manipulation. The user needs to see changes on the computer screen to continue through the Action Cycle in pursuit of completion of the original goal. The extent and accuracy of that perception is a strong factor is creating usability issues.

The data that were tagged with the Perception code were those in which it appeared the usability issue was related to the lack of or a difficulty perceiving some item on the computer screen. It is impossible to know for certain what a user is (or is not) perceiving and this can only be inferred by the verbalizations and the action on the computer screen.

While this is true for all the data gathered in this study, the cognitive action of Perception was most problematic and sometimes particularly difficult to separate from Interpretation. For example, while it may seem apparent that the user did not heed some text or icon, one cannot know whether that was because he never perceived it, or because the user perceived it and interpreted the information as something that did not warrant any attention or comment. It seems clear according to the Norman model that changes in the computer system must first be perceived and then interpreted, but under the circumstances of this observation it was often difficult to discern between the two. For the most part the phenomenon of Perception-related usability issues was due to trouble locating or seeing items.

5.6.4.1 Difficulty locating items in a list.

After some data are entered into the system, the user anticipates the next step in the ordering process. If the user wants to order a drug or lab test, for example, some actions are performed and he is presented with a list of items, including his desired order. The nearer the desired item is to the front of the list, the easier it is to perceive. The user has a reasonable expectation that common items should be listed first and this is often not the case. When there are a great many items in a list, it is difficult to find what one is looking for. Often, the user is presented with lists of over 30 items. One user could not find a common diet order among a list 37 items. I cannot present quotes from users when they did not see something, but here are some example quotes from users searching through complicated interfaces:

"… Put a "Q" after the "proton[ix]" and it brings me up to the quick orders. It should. Curious. The quick order should show up at the top of the entry, but here it's number 4."

"So, if I can find it, in the morning. It's a long way, there's too many options here to click. Here's one waaaay down here, Q AM!"

"...instead of Q day, it's going to be Q evenings. … And then again, finding evening again in here. There's about 30 options. For frequency every 2 hours up to every 72 hours. And it's hard to find, here it is. Probably number 27 out of 40 choices. So that was all one drug! Kind of much harder than pencil and paper on that one."

"So, let's see. It's got a lot of different choices with lots of different components, so you have to kind of read through each line to select what you want in terms of the sodium and the potassium and the vitamins and all that kind of stuff. It can be a little bit difficult."

"...do I want a CK? Is that on here? I use this form a lot and I still don't know what's on it. 'Cause there's so much. But, there never seems to be… there's at least one that's not. Oh, well it IS there, now that I said that."

5.6.4.2 Poor visibility.

In order for a user to see a screen item it must be sufficiently visible and relatively distinct from the background or other items to discern its presence.

A common usability requirement is that the users should be aware of the system state; they need to know what is going on. During slow response time, due to processing or network delays, the user often did not know what was happening. One user identified a problem, the system used a non-standard, and inconspicuous way of conveying information:

"And it's just making me wait. It's slow. It doesn't give you an hourglass. It gives you a TINY little note in the bottom left-hand corner that says "signing orders."

Often it is hard to see the word you are looking for if it is not obviously displayed. One user wanted to start his session for entering "Pre-admission orders." The display he was presented with is as seen in Figure 6. The operative phrase "Pre-admit" was not in a prominent place to the dismay of the user as follows:

"And so we're going to look for pre-admit. Having the word 'enter' usually is not an easy; it should be 'pre-admit orders' like admission orders.' "

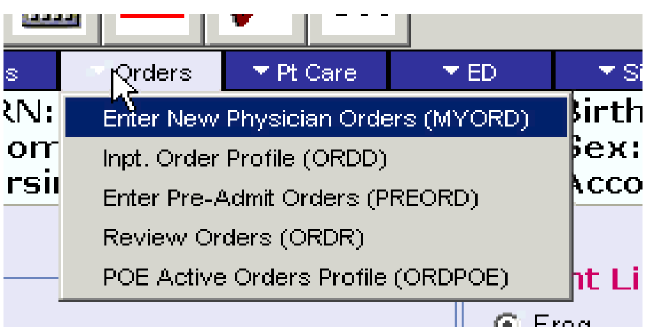


Figure 6. Interface menu to find pre-admission orders.

Many times the users simply did not see what they were looking for. I noted users looking for items that were on the screen, but they were missed. The users often would repeat the same search strategies over again, as they recognized they may have missed the item they were looking for on the first go around. They intermittently found it on the subsequent look. In one system several of the users missed the “activity” order in an admission order set. When signing the orders, an alert was fired that the activity order was missed, not allowing the signature until the missing order was entered. One user acknowledged the alert as though he was expecting it, but claimed that he did not miss submitting an activity order that was clearly there on video review. His quote:

“Doesn’t allow signature, that’s going to be his activity level. Which, yep, activity, it always does that. Never gives you the activity order to sign, but it always tells you [that] you haven’t done it at the very end.”

Some items may not be seen when they are not expected. One user was presented with two alerts at once, a duplicate drug alert and an allergy alert. He was prescribing a combination medication, one component of which was already ordered in a different medication. The system fired a duplicate drug alert, which he acknowledged and overrode. Immediately after that an allergy alert fired and he quickly overrode that without comment. His verbalizations seemed to indicate he assumed it was still related to duplicate drug administration, which he dismissed as not meaningful, and not a new patient safely warning. Perhaps pairing alerts together suggests to the user they are the same problem or of similar importance and if one is dismissed, the other may not even be attended to.

5.6.4.3 Ignoring items.

The example of ignoring alerts is one of many instances in which the user seemed to neglect an item presented to him on the user interface. One never knows what a user does not perceive, but when something "pops up" in the foreground, such as an alert or reminder, and the user makes no comment at all, it is an interesting phenomenon. It was impossible to know whether the user did not perceive the item on the screen, or whether he interpreted it as something not worth mentioning. That one would make the decision to keep silent about an obvious attention-getter, in the setting of "think aloud" session, is a noteworthy event. This "ignore phenomenon" will be discussed later.

In summary, the users appeared to have difficulty perceiving items on the interface. Too many items, or being too far down a long list gave the users problems. The users cannot "see" things that are not visually or semantically prominent. The other cognitive codes associated with the segments that were deemed Perception problems were Action Specification and Interpretation. The HCI codes that arose noticeably with this group were "locating items on the GUI" and the "visibility issues of problems with color, contrast and attention."

To repeat, it was sometimes difficult to distinguish between the user's perception of the interface representation and his interpretation of its meaning. It is not obvious to the observer when the user did not see something. However, when the user did not clearly verbalize anything about the activity, I made the inference that it was an issue of perception. There were several segments that were coded with both Perception and Interpretation. Some of those that seemed to be more a problem of the latter are described in the following section.

5.6.5 Interpretation.

In this next step of the Human Action Cycle, after the user has perceived the changes in the system state, he must try to understand it. The user interprets the meaning of his perception, what the display is communicating about the state of the system. When the user has difficulty in this step of cognition, the user is confused about the state of the system. Misunderstanding what the user has perceived on the display will likely lead to an inaccurate evaluation of whether his intentions have been satisfied. In some circumstances, the user is unaware of the misinterpretation and at other times, he may openly verbalize his confusion.

5.6.5.1 Lack of understanding of the meaning of graphics and text.

As discussed under Action Specification, users have many ways to name items and it is important that the system utilize multiple synonyms for concepts so that the users readily understand. As when the action could not be specified when the desired term was not in the system, the user cannot understand text that represents an idea with a phrase he was not expecting or looking for. As mentioned earlier, a user simply could not find “cardiac” diet although it was clearly displayed as “AHA diet” and presumably perceived. A screenshot of that display is shown in Figure 7 and shows that the term "AHA diet" is prominent. It is unlikely that the user did not perceive this.

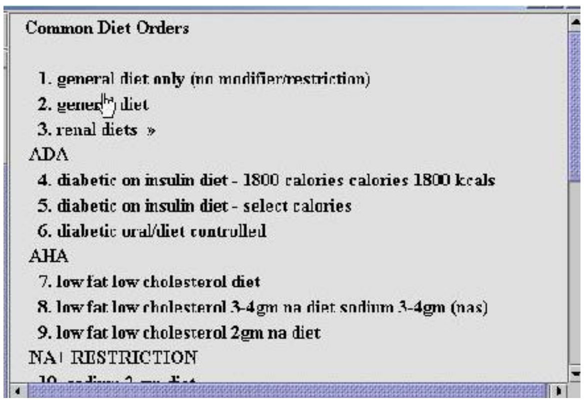


Figure 7. Menu of diet orders.

The users sometimes were confused even after the system alerted them to an error. One user, while entering an order for a drug, became confused about the fields in which to enter data. He accidentally entered "20" as the name of the drug and "mg" as the dose. The system alerted him that the latter was an invalid input for "dose." He saw the alert and remarked:

"That was an error message regarding the free text dosing of the measurement."

He seemed to acknowledge what the message was telling him but it did not occur to him that it was accurate and he did not correct it.

In another example, the user admitted confusion but proceeded anyway.

“Notify Provider… Huh?? Wow, they really … OK, they want me to notify provider if there’s ST elevation or new left bundle branch block. I think they do that anyway, but I’ll check it. Notify provider for complications of myocardial ischemia. Sure, that sounds reasonable. Whatever that is.”

5.6.5.2 Lack of system knowledge or state.

This is similar to the first situation, but here the user describes a more global confusion that does not seem due to any single display item. One user “got lost” at the outset of his session, likely at a point in time that he has experienced often, as follows:

“Ok I’m going start putting in my orders. I always pretty much do orders the same way- I always go to hospitalist admission orders. And I go through the choices here. And then I go back and do everything that’s not here. I’m clicking on …. Actually, I’m not. Ahh, it looked a little different than it usually does for some reason.”

A moment later he gave this explanation:

“I’ve been off for a couple weeks, so I’m not quite up to as quick as I was.”

One user was waiting for a response from the computer. She first explained it as a slow response time, and then suggested that it was devising some alerts to present to her.

All the while the computer displayed “READY” in small text in the bottom left corner. Here the user was confused at a pause in the action because she was misinterpreting the silence as something else she was accustomed to occurring.

“OK. So, everything looks good. I’m going to sign. And often times it pops up with drug interactions of various sorts which are usually not significant, but occasionally are. It’s thinking. It’s still thinking. Actually, maybe there are no drug interactions. That’s a surprise!”

5.6.5.3 The timing and delivery of information.

One user wanted to order an exercise treadmill test. He was given the information that if he had been ordering a nuclear medicine test, it would include this test he was ordering. He was not ordering the nuclear medicine test in the first place, so he was confused as to what the message meant:

“Yes this guy wants a stress test, in the morning, and now it says ‘cardiac treadmill stress test.’ He doesn’t need a nuke med test, but it says under here, ‘An EKG Cardiac Treadmill test is included in the nuclear medicine with Treadmill stress’ What? OK, so I gotta, I will not order the nuclear medicine. He doesn’t need that.”

The order in which information is presented can lead to different interpretations, even clinical ones. One user was admitting a patient for chest pain. In a “chest pain” order set, he was presented with a list of “prn” medications appropriate to that diagnosis. He saw “morphine” and dismissed it as being too potent. After ordering several other meds, he came upon “nitroglycerin” and then reconsidered and decided to order morphine after all. He said:

“It’s OK to give him nitro. Oh, I guess I can give him morphine if he breaks through the nitro. I think it’s back up here somewhere. OK. OK. They put the morphine before the nitro, so, yeah, after nitro, you can give him morphine. That’s fine. With me. I hope they do an EKG before that...”

There is a definite clinical sequence to orders, especially “prn” medications. The clinician’s goal is to choose a medication first that relieves the symptoms but has the least potential for harm. If the clinical goal is not reached, a “second line” drug is chosen. Here the user dismissed the morphine as being excessive, even though it is very commonly prescribed in this scenario. Cognitively, it didn’t seem appropriate seeing it out of the sequence he is accustomed to.

5.6.5.4 Warnings and alerts.

The “pop-up” alerts that are seen in CPOE are known to be commonly troublesome (76-78) and here in this study, they were conspicuous culprits leading to the user’s difficulty in interpreting the display that is perceived. Many times the user may not even know what the alert is suggesting. Even a frequently encountered alert can be misinterpreted. In one system, if the user “signs” the orders, then continues to enter more orders, any subsequent “signing” causes an alert to fire, telling the user that changes have been made. As follows:

“And, let’s sign those. ‘ Changes to orders have been made.’ These suggest like these suggest additions.”

That was an incorrect assessment of an alert that the user likely sees every session. In an example of entering free text, mentioned above, the user ignored an alert to his erroneous date entry as he interpreted the message as something it was not. In another example,

“So now I’m going to sign all these, and it’s giving me the fact that this CT order is pending, and I’m going to say ‘continue signing anyway.’”,

That was not an alert about a pending CT, but the above alert that new orders have been entered in the same session.

Misinterpretation of alerts seemed to be common. This leads to users frequently ignoring alerts and seems to be related to the “ignore” issues mentioned that will be further elaborated later.

Thus the user may have difficulty interpreting the display, leading to usability problems. This step of the Action Cycle is sometimes difficult to differentiate from the related Perception step and leads to problems with subsequent Action Specification. The segments coded here were commonly associated with those two codes. In addition the “HCI” codes that were associated were problems understanding system state and feedback, lack of system knowledge, difficulty locating items on the graphical user interface and difficulty with the understanding of graphics and text. Other associations were bad alerts, problems with synonyms and ignoring items on the display. This now leads to the last step of the Human Action Cycle, the Evaluation of the system state.

5.6.6 Evaluation.

The final step in this Cycle is evaluation. Once the user has interpreted the system state he has perceived on the display, he compares this interpretation to the expectations he has, based on the original goals and intentions from the first step. If he evaluates that his intention has been met, he decides upon a new one. If the intention has not been met satisfactorily, he creates a new action specification. Either way the cycle continues. (See Figure 4.)

The step of Evaluation is very much tied to Interpretation and hence Perception. Conceptually, one can abstractly differentiate these processes, but when observing users it becomes difficult to separate them. The extent to which one may evaluate whether an intention has been satisfied is dependent on the interpretation of one's perception. Thus all three steps in the "gulf of evaluation" (See Figures 3 and 4.) seem to be interrelated.

5.6.6.1 Evaluation is dependent upon interpretation.

In the case that the interpretation of the computer display is incomplete or in error, the evaluation of that information will be incomplete or in error as well. The lack of understanding leads to a false evaluation, and this certainly creates a possibility for error. As per the examples above, if a user misunderstands an alert that has been presented, he cannot possibly heed the warning. Any benefit that decision support might have provided in reducing error is lost. If the degree of confusion is so great that that the user does not merely misunderstand his perception, but makes no interpretation at all, no evaluation is possible and the intention cannot be fulfilled. This may lead the user to stop in his tracks and proceed no further. When this was observed, rather than trying another tack, the user occasionally conceded his inability to carry out the intention and decided against doing it altogether. This is described below as “resignation.”

5.6.6.2 Evaluation that a new action is needed.

Even when Interpretation is adequate, the user may change the action anyway to reach the original or a similar intention. For example, if, after the user tries one or more actions and this leads to a dead end or little or no response from the system, he simply tries something else. Users often entered repeated terms until one was recognized by the system. Such as:

“Start with just admitting. OK. Admit. To PC …. OK. To PCU. Hmmmm? I’ll say Med-Surg then.”

Other times, the users fully understood and correctly interpreted the display, but simply did not trust the information. That potentially makes it impossible to evaluate whether the intention has been reached. Often alerts are dismissed as they are not deemed accurate, based upon knowledge about the patient that contradicts the alert, or even an overall distrust of the system, as in:

"He has this allergy … [but] I think we'll just go ahead and give him Ativan if he needs to sleep. There shouldn't be any problem with that. I … well. This is one of the problems with [this system]. The allergy indicators are a little overzealous."

A lack of understanding or agreement with issues outside the computer system, such as hospital policy may also alter one's evaluation of information presented. On user routinely removed the default values that declared him the admitting physician and the attending physician, which he was, but he chose to take the effort to change that and uncheck boxes.

"I'm the admitting doc. And the attending. Most people know that already, so where that attending physician goes to I have yet to find out. I don't think it helps anybody so why bother? Yeah, it's obvious I'm admitting and everybody knows [here] I'm the attending so this bit of information somewhere, nobody ever looks at it again, and so it's superfluous, so I figure why add more to the process?"

The only other code that appeared to be associated with Evaluation was Interpretation. Given that the two concepts are so related, this seems understandable.

5.6.7 Summary of "Cognitive" codes

Using Norman Theory of Human Action to analyze and describe the experience of these CPOE users, some of the steps were more prominently represented in the segments where usability problems occurred. As shown in Table 17 above the codes that appeared most frequently were Action Specification, Perception and Interpretation. Table 18 below shows the counts of when each of the cognitive codes co-occurred in the same segment, and these three codes predominate. Thus, these observations indicate the user interface issues arise as the user attempts to formulate which actions to take, as well as when he tries to perceive and interpret any changes that appear on the display. Under the above descriptions of the findings, there also appear to be other codes that cluster with the "cognitive codes." In the next section, I will present the findings of cognitively significant segments that exemplify issues related to usability heuristics.

	INT	AS	EXE	PER	TERP	EVAL
INT	-	2	0	2	2	0
AS	2	-	2	**5**	**6**	2
EXE	0	2	-	2	2	1
PER	2	**5**	2	-	**6**	1
TERP	2	**6**	2	**6**	-	4
EVAL	0	2	1	1	4	-

Table 18. Numbers indicate the number of time the codes from each corresponding row and column co-occurred within the same segment. (Abbreviations: INTention, Action Specification, EXEcution, PERception, inTERPretation and EVALuation.

5.7 Results of Analysis Using the "Usability Heuristics/HCI" Coding Scheme

5.7.0 Introduction

In the previous section the data were analyzed with respect to users' cognition, using the framework of Norman's Human Action Cycle. In this next section, the data are presented in another way. Here are results of coding the data using the *a priori* list of codes that was assembled according to established usability heuristics common in the literature (see Table 5). These were compiled using lists of heuristics found in general texts of user interface design as well as publications relating to both usability and medical informatics. Of the 249 data segments that were analyzed, 130 were felt to contain elements that demonstrated usability issues that could be characterized by the usability codes. There were 169 "usability codes" used in all. There is considerable overlap between this coding scheme and the previous cognitive coding scheme. That is, many of the segments that were assigned cognitive codes were also coded using this usability schema. This was done in a deliberate attempt to explain and expand upon how specific cognitive issues may relate to these Human-Computer Interaction phenomena and the converse. The following ten categories of usability issues were discovered and will be explained (Table 19). The extent to which they relate to the cognitive concepts of the Human Action Cycle and other phenomena will also be presented.

1	Locating items of the GUI
2	Lack of knowledge of system content
3	Time and effort of data entry
4	Consistency
5	The meaning of text and graphics
6	Navigating – Moving around the GUI
7	System state
8	Cognitive Overload
9	System Response Time
10	Visibility and attention

Table 19. Human-Computer Interaction Issues, Results of Analysis Using the "Usability Heuristics/HCI" Coding Scheme

It should be noted that not all of these usability issues occurred with similar frequencies. Two thirds of all the segments that were felt to contain usability issues were contained in the first three categories, "Locating items," "Knowledge of system content," and "Time and effort of data entry." Quantifying these data is problematic for many reasons, of course, but the difference is striking enough to potentially have some significance.

5.7.1 Locating Items on the Graphical User Interface

This was the most common problem the users encountered. The process of "writing" orders on paper compared to "entering" orders in a computer system is different in many respects. One of the purported advantages of computer order entry is that the user does not have to initialize the idea of which order to write next, but is able to choose from a pre-constructed list. (This was discussed in section 5.2.3, questioning where the user "entered" into the Human Action Cycle.) The user is not writing orders so much as picking from a list. One user stated:

"Trying to look and choose what I'm going to give her …Trying to think of what I'm going to give her at the same time I look through my choices."

Therefore, the task often changes from writing down words to finding and choosing items in the graphical user interface. Even if text is entered by typing into a blank box, it must correspond to a constrained list that the system recognizes, and the user must "find" the right word to enter. This category was by far most common usability issue. The users spent a great deal of energy attempting to "find" what they were looking for. The frustration of this problem was expressed by this user:

I'm going to keep looking to see if there's like a, well… see this drives me crazy, I never am sure which imaging study to order. If I were writing it, I would just write what I want. But now I worry that not the right thing's going to get [ordered].

5.7.1.1 The user has expectations of where to find items.

The users appear to have expectations about the location of certain orders. These expectations arise either out of conventions of medical practice or from the observation that items are organized similarly throughout the interface. This occasionally is not the case and is a source of usability problems.

One user had admitted his patient to a telemetry (monitored) bed. The whole point of a cardiac monitor is to continuously observe the patient's cardiac electrical activity and react quickly to problems with the appropriate medications. This user had the expectation that the appropriate medications would be located conveniently near the order set for monitoring. This user was actually involved in the creation of this order set, so he was doubly surprised not to find the orders where he expected:

"Oh yes, he needs all the telemetry…meds so we can get all the atropine type stuff in there, which is not on that, and we need to probably put a link to that on … orders, telemetry orders. What I want from here is all the meds which we were going to, I thought we were going to put that up at the top."

Two other users had difficulty ordering precautionary nursing instructions. One user was certain "Fall Precautions" was located with other nursing instructions under "activity" but couldn't locate them:

"…activities. Trying to find Fall Precautions. Can't remember where that is."

Another user wanted to find alcoholic seizure precautions (as discussed in the section under action specification.) This was an item he had ordered before. He clicked on "precautions," then "seizure precautions" on a sub-menu, but those referring to alcohol precautions were not located there. They were to be found under some alcohol withdrawal protocol in a different location altogether. His strategy for looking for "alcohol precautions" under "precautions" seemed logical and reasonable. A cross reference at that location would have been helpful.

Users naturally have expectations that similar orders should be grouped together. This user thought that fluid restrictions should be located under dietary orders:

"Diet thing is always difficult. It's hard to tell… how to order what I need. Do I …? Oh, let's start here. I want him on a 2 gram sodium diet. Now somewhere I'm hoping there's going to be a way to get a fluid restriction out of all of this stuff. I'm not sure how to make that happen."

When testing for a serum cortisol, a physician very often orders it for the first thing in the morning. This user thought to look on an "A.M. Laboratory" form. The idea of the "form," analogous to its paper counterpart, is to organize together items that are often ordered together. The following quote is from a user who wants to order an "a.m. cortisol" and demonstrates the utility of the lab form in locating items, but also the frustration when an item is not in an expected location.

"OK. AM labs. Get an AM cortisol. Use the AM lab form. It's definitely the easiest way to order these labs. Is there cortisol on here? I'm not sure if it's one of the labs on here or not. Actually, I don't see it … so ….I'll just cancel that. And try to order it the conventional way. Cortisol AM. Listed as AM. You'd think then, that all this would just be geared so that it would automatically be done in the AM. Now I'm trying to remember how to order AM labs, so I think if I do "Tomorrow 0400 in AM," no "AM draw" that should do it. We don't do these that often anymore because of the I-forms."

This concept of placing items that are frequently ordered together in the same location on the interface is a basic component of CPOE interfaces. In addition to the forms mentioned, the clustering of like orders underlies the popular idea of the order set. This concept is extremely valuable to usability and will be discussed at length below.

5.7.1.2 Specific Causes of difficulty locating items. (Lists. Order sets. Synonyms. Scrolling.)

As discussed under "Perception," a user cannot find things when the menu lists get too long. One of the most valuable interface aspects of CPOE with clinical decision support is the constraint of like items on a menu. However, when the pick list contains many items, it becomes arduous search for them. At times, this becomes more difficult that typing in name of order. Here is an example:

"...instead of Q day, it's going to be Q evenings. … And then again, finding evening again in here. There's about 30 options. For frequency every 2 hours up to every 72 hours. And it's hard to find, here it is. Probably number 27 out of 40 choices. So that was all one drug! Kind of much harder than pencil and paper on that one."

Users expect to find the items they want in these menu lists or in order sets. The default values listed are supposed to be those most frequently ordered. If the desired item is not on the default list, the user must change his tack and try again. Here, a user expected to find the dose of a medication under a short-cut named "quick orders":

"Let me try this one more time... Let's see what comes up under klonopin... Oh darn it. Oops. Go back. …Let's see here real quick. 1 mg. You know what I want? I want 1 mg. QID, that's what I want. So you know what I'm gonna do? I don't want to be in quick orders. Go back one more. Will that get me back?"

Another user wanted to order a standard ("PA & lateral") chest x-ray for the following morning. While using a general hospital admission order set, he found that the following options were available under "chest x-ray":

1. STAT - portable chest x-ray
2. ASAP - portable chest x-ray
3. AM - portable chest x-ray
4. STAT - PA & lateral chest x-ray
5. ASAP - PA & lateral chest x-ray

But there was no allowance for the standard type of x-ray for the "AM." That would not be an uncommon order, but it is unclear why that was left off the list. The user had to make a mental note to do this later, when finished with the order set:

"Once again, going through this form. She needs a repeat X-ray in the morning. Which I am not seeing exactly what I want here, so I'll remember to do that …"

There are other difficulties that users have in locating items. The problem with the lack of synonyms for the names of order entry items again comes up here. When looking down a list of items the user will not be able to locate his desired order if it is named differently from what he is expecting. Likewise when entering an order by typing its name in a text box and submitting the entry, the computer system must recognize that word among its list of synonyms. If the synonym entered is not recognized, the user has essentially failed to “locate” the desired order.

Also, there were many instances in which the user had difficulty finding items because he needed to scroll to find them. If a list of items is too large to all be seen in a single window, the user must scroll to find them. While it is understandable that not all lists of items can or should be contained all within one window, a point of focus outside the visible area makes for poor usability. A noticeable issue arose in one system that listed the recently entered items on the left of the window after they are entered on the right. The item most recently entered was not visible for the most part unless the user scrolled to that area to visualize it.

5.7.1.3 Getting lost in the Interface.

In the following examples, the users appear to be confused to the point of feeling lost. Here the user was trying to find a therapy referral in an order set:

“Going down the hospitalist admit form again. I can never remember if the OT and PT are under Consultants or if they have their own choice.”

This user kept trying different locations to order fluid restrictions:

“Now I think this is where I get to find "Strict Ins and O's” There we go, Ins and O's, monitor. Hmmmh. It's probably not exactly what I am looking for. So I'm going to go out and see if I can find it under the CHF… Congestive heart failure treatment orders. Looking promising.”

Lastly, this user searches all over and gives up by deciding to transfer the responsibility of enacting this order to the nursing staff:

“And then I couldn't find Fall Precautions before, in the Hospitalist Admit. So, let's see if I can find it. I click on Precautions. And… I don't know… weight bearing? …. Now, I clicked on the Weight Bearing to see if it was there, but it doesn't look like it is there so… I'm going to do "oops" to get out of there. Uhhh… what am I going to do? I'm going to click on "activity" and see if I can find anything there. See where's my choice… Nope. I know I've done this before. So uh … So, I'll try… I'll type in "fall" and see what that gets me. No, it doesn't get me anywhere. So. I'll do default. I'll type in Nursing Order. "Please make sure that patient is on fall precautions." They'll figure out what that is.”

There were many examples of users having difficulty locating items on the GUI. The category was the most commonly encountered one when looking at the data from the standpoint of usability and HCI. This category was associated with the cognitive codes of “action specification” and “perception.” Within the other HCI categories, the problem of “cognitive overload” co-occurred with this issue as well, to be discussed below.

5.7.2 Lack of Knowledge of System Content

The second most frequent problem occurred when the user simply did not have enough understanding of the system to continue without problems. It may seem obvious that users must have some prior knowledge to use a computer system. It may be unreasonable to expect a physician to use a computerized order entry system without having specialized training in how that system works. However, there are many other applications, outside medicine mostly, in which any person who has a reasonable degree of familiarity with computer systems can pick up on how to use the system with little to no prior training. The so-called “user-friendly” interface is described as such because its manner of use is either intuitively obvious or employs standard interface conventions that are known by all. A good user interface can guide the user along without the need for a great deal of prior knowledge. To the extent the user is required to have specific, perhaps arcane knowledge of his CPOE system, usability problems occur. This problem related to the cognitive issue of specifying the action sequence. Cognitively speaking, before one can specify which action to take next, one must be aware of and consider the range of possibilities. The knowledge of these possibilities is essentially the “knowledge of system content” herein described. One must access one’s long term memory and if unsuccessful, this may lead to guessing the action specification. These two points will be discussed later. This is related to many other phenomena as well, and very often leads to the previous issue of difficulty locating items on the GUI.

One must have a general knowledge of the way the system works. Certainly one must know how to start. The users studied were all generally experienced users and yet they experienced problems negotiating even fundamental activities with the interface. This is the opening few seconds of one user session (this person had been a user of this system for 22 months and rated himself as sophisticated) and it appeared he does not even know how to get started as he could not find the admission order sets:

"And, ummm, care sets. I wonder if there are... any… now where are the admission orders? Wait a second, they're around here somewhere. This is not the form that I am used to. Well, I think there's something wrong here."

Here a user seemed to have forgotten some fundamental details:

"We want a lipase level and an amylase level, but I think that's going to be stat. Let's see if it is or not. … All right, we will make this every morning if we can… Let me see. "Early AM draw" There is no space on here for every AM. So we'll just do "early AM draw." Oh wait a minute, "frequency," Q AM, let's see, "early AM draw"… "Q 24 hours". OK."

In this example the user could not find a very common instruction, "every AM" and looks in the wrong place. Instead of the under "frequency," he looks under "priority." This is very basic functionality, and users very commonly behaved as though they were novice users. Certainly anyone may forget something. But misunderstanding basic system functions on such a frequent basis suggests a confusing interface.

At one site, when one orders a test to be performed "in the a.m." (the following morning), the system defaults the scheduled date and time quite naturally to the next day after it is ordered and the time that “a.m. labs” normally get performed. Inexplicably, this does not happen when cardiac tests, such as an electrocardiogram, are ordered. The users seemed to know this and adjust their behavior accordingly as in this quote:

“And I'll do an EKG for the AM, and this is another one of those where I mark "early AM" and it doesn't change the date and time, so I have to go in and change the date and time to tomorrow, which it should default.”

Here is another example in which the user must have esoteric knowledge:

“And then in the comment line under standards of care, type in what we are going to do, which is a knee aspiration.”

Here the user knows she must enter the name of the procedure under “standard of care,” which is certainly not intuitive.

There were numerous examples of this problem. Conversely, there were many examples in which users seemed to have this detailed knowledge and they performed effortlessly. There was one user in particular who was especially expert who stood out to this observer, in that he did not share many of the same struggles. Interestingly, he was a clinician whose part-time duties lay with the information system implementation team.

5.7.3 Time and effort of data entry

The third most frequent observation of usability issues was related to data entry. This phenomenon reflected the extra work that is caused by entering data on a computer (compared to paper). The segments in which the users indicated that the process was too time consuming or complex were the ones that were accompanied by the most emotional frustration. This result is associated the findings of “more work” and “bad defaults” to be discussed in the next section.

5.7.3.1 Poor use of defaults.

The poor use of default values in an interface was a major cause of this problem. Default values are meant to be time savers, but when an existing default value needs to be changed every session, or when a default value is lacking for an entry entered every session, the user becomes frustrated. Users reported problems of entries that defaulted to unhelpful or nonsensical values. One set of post-operative orders defaulted the timing of the removal of a urinary catheter to 24 hours after the order was written when the optimal time would be the following morning. Unless caught, this could cause a several hour delay and undue patient discomfort.

One order set for an ICU admission had a list of over 25 medications, but users only needed one of them and took some time searching for one desired item. Again, this is an example of long lists causing problems.

As mentioned under action specification, complicated processes that seemed easy to explain on paper became convoluted and arduous on paper. A repeated example involved adjusting ("titrating") a medication based upon a physiologic parameter. If an automated form existed for that particular problem, this could be accomplished. Otherwise, these orders were often impossible without long free-text comments to help nursing and pharmacy comprehend the order. Even then, occasionally users gave up and decided to postpone the order entry until after the session when it could be done verbally or on paper.

Here a user has trouble entering a time to obtain a lab test:

"And that's something that I don't understand happens. If you do the time to draw and you go to the collection dated time, it doesn't automatically go to today's date so you can move it up and down…"

A list of default dosages in order sets can be a great benefit, assuming that of the items listed is desired. A very experienced user took over two minutes to enter one medication when he wanted to prescribe 5 mg. in the morning and 10 mg. in the evening. This example also shows problems with finding items on long lists:

"It's one in the morning and two in the evening, but there's no straight set for that, so I got to fiddle. I probably would pick a 5 mg., no they don't even have a 5 BID, but I'm probably gonna do… so I have to pick the 5 mg TID oxybutynin and instead I have to change the standard to 5 BID instead of TID. No, I think I'll do 5 in the morning and then 10 at night. That's what I'll do. So, if I can find it, in the morning. It's a long way, there's

too many options here to click. Here’s one waaay down here, Q AM and then I end up hitting the drug again. This time I’m going to do 10 in the evening. So here’s a 10 size. Here’s a 5 size. I’m going to push 5 once a day. I’m going to have to change that again ...”

5.7.3.2 Issues of text entry.

Entering data by typing takes time even for the skilled typist, and this was often exacerbated by difficult text editing processes seen on some systems. One system did not allow use of the “backspace” key when correcting errors. The entire entry had to be selected with the mouse, deleted and typed again:

“I'm trying to add a "g" and spell correctly. But I have to delete everything…”

Similarly, users may need to change a default date and time by highlighting the entire default value and retyping, rather than changing only the one or two digits that were off.

“And that's going to be timed. Can I make comments about the program? This is the one part of the program that I really do not like because it's hard to enter… it's a little clunky as far as entering the labs and so make it 19, the timing, placing the timing is not the easiest.”

One user summed it up:

“Even if you're good at typing, typing takes time!”

Some users did not understand the difference between free text and structured data. One user routinely entered the duration of a medication in a comments box. She found the name of the drug and its dose in list boxes, but ignored the "duration" field and typed that in a free text box named "order comments." It would have been faster to do it the correct way.

5.7.3.3 System issues.

One system had a feature that all the users lamented. One often needed to enter orders "twice." The order sets have a list of possible orders that are checked off with check boxes, and then one clicks a "process orders" button. Then, some of those orders need confirmation. One at a time, each order will pop up in an individual window. The window states "Place an order for X" or "Place an X order." Why exactly some orders are split up into two steps and some are not, seemed to baffle the users:

"And, I have to wait and order the diet a second time which is irritating. And pre-admit orders which I didn't want now I get to tell them I don't want them a second time which is irritating. And then I get to order a social service consult for the second time, which is very irritating. Why they select those three or four that come back another time, I'll leave to the IT experts?"

5.7.4 Consistency

An important aspect for interface usability is consistency. Ideally, users should not have to wonder whether words, situations or actions mean the same thing every time they are encountered. In addition to uniformity in appearance, users expect uniformity in behavior of the interface. This is essential to making the user interface easy to learn and remember. Words on the screen should do what they say they do, and they should not mean more than one thing. Actions should work the same way every time. Cognitively, users see what they expect to see and if the interface is inconsistent, problems occur.

5.7.4.1 Words should have consistent meaning.

In one instance a user want to order a sub-cutaneous injection preparation of a medication, enoxaparin. In a menu of six items, the third on the list was, “enoxaparin inj. subq,” which would seem to be the ideal place to find this item. For some reason it was to be found under “enoxaparin → quick orders.” The user notes:

“He needs sub Q lovenox… and that's actually in the quick orders, not the sub Q, interestingly enough.”

There were circumstances where names for different items were labeled identically. In the same screen “BNP” meant both “basic nutrition panel” and “beta-type natriuretic peptide.”

5.7.4.2 Actions should behave consistently.

If the user interacts with the interface by clicking somewhere, he expects an identical response to an identical action. Users had the expectation that hitting the “Enter” key would “place” an order. This was effective in some circumstances, but at other times they needed to click a “Place Order” button.

“I hit Enter and nothing happened so I must come down here and click somewhere. All right. Place orders.”

As mentioned above, one system automatically changed the date and time of a lab order, when the user wanted it “in the a.m.,” as it should. The exception to this was cardiac labs, for which the date and time needed to be changed manually. Fortunately, the users were accustomed to this anomaly and all seemed to catch it and accommodated accordingly.

In another system, one user needed to uncheck all the undesired medications in one order set, whereas there was a shortcut to do this in others.

“De-select the ones that we don't want. It would seem to me that this should be a one click operation, but unfortunately now we'll go through this whole process and revisit each one individually.”

The concept of “selecting” versus “unselecting” was commonly seen. The advantage to an order set it that similar orders – those that are frequently ordered together – are located in the same place to save time. In a given clinical situation there are often lists of medications or laboratory tests are commonly used. In some order sets, the user scans the list and selects those items that are desired. In others, many (but not all) are already selected will be ordered unless the user chooses to remove a checkmark (deselect) from the item. This was not consistently employed within and among the various systems and seems to be a potential for error. This “opt-in/opt-out” phenomenon will be discussed below.

Users have expectations. Once they become accustomed to a user interface, they expect it to work similarly each time. Here are more examples. In the first, the user was actually waiting for an alert to fire and it never does. In the second, the user felt she can’t order the dose of a medication that she wants, as the systems had changed from the way she remembers it.

“I already see one thing is missing; I should have had the medication reconciliation thing show up immediately, but that hasn't shown up. It often times shows up the next day.”

“Oh, Zofran quick orders. You know, there used to be a 4. I think they changed this just recently. I think they changed this, because there used to be 4 mg every 6 hours. And I suppose that's only bad if other people use that too, but we're going to have to go find it.”

Lastly, one user wanted to order physical therapy, occupational therapy and social service referrals. In each, he wanted to add a text comment that would inform the clinician of his needs. There must have been more than one way to do this, because in the past, a blank box in which to enter comments was presented to him. During the observation session, that did not occur and he spent four minutes looking for this. He never found it and had to start all over. The second time, he approached it as an individual order, not as a nested order in the admission order set.

5.7.5 The meaning of text and graphics

[Note: this is also a subsection of "Interpretation," 5.6.5.1, but is applicable here as well.] When a user encounters text or an object on the graphical user interface, it must convey meaning to him in order for him to interact with it. In this setting, the interface items can have semantic significance to the user as one who interacts with computers in general as well as a clinician. That is, in order to communicate effectively to the user, the display items must follow some pattern that is a computer interface convention or was learned in the system training, but also it needs to convey a clinical medical idea that has some prior meaning to the user. This idea corresponds very closely to the concept of Interpretation in the cognitive sense, and segments were often coded similarly. A user notes something on the interface, but it needs to make sense for him to interact with it effectively.

In one example as mentioned above, one user had trouble understanding how to start the order set he wanted. He was looking for "pre-admit orders," but it was listed under "enter pre-admit orders". A small difference admittedly, but he noted the problem.

In general, the closer the interface reflects the normal workflow of medical practice, the more usable it is. The use of graphics that are metaphors for clinical experiences help the user. As in Figure 8, the use of "forms" is very helpful, as this mirrors a process that might be done clinically. It is a metaphor for something that is used in one's usual workflow. In this case, similar labs are grouped together into the categories that are natural to the clinician. The users seemed to address these forms with little difficulty, and as such, did not mention them.

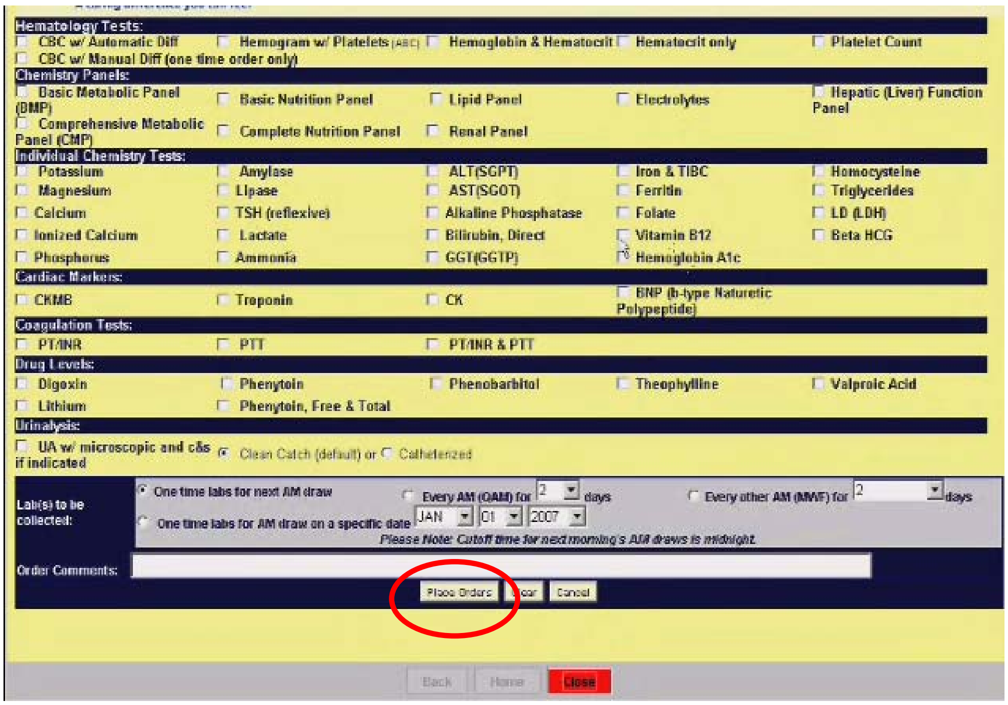


Figure 8. A laboratory "form" is a helpful metaphor. "Place orders" is highlighted.

As in figure 8, the use of the terms "place order" or "accept orders" are very clear to the user. Conversely the use of buttons with terms like "submit," "enter" or "end" seemed to cause confusion.

The users had difficulty discerning the difference in terms if they were not semantically comparable. For example, a user wanted to order an X-ray of the abdomen. He wanted a plain X-ray, one view. He was presented with the two options of "abdomen 1 view" and "abdomen complete." He verbalized that he wanted "1 view, complete" but he ordered the second one. It is not clear whether he ordered the test he wanted. The names of the radiology tests were not ontologically similar, hence confusing. It would have been easier to distinguish the two if they had been listed as "1 view" and "3 views" or "simple" and "complete." Another user wanted to order a bilateral duplex carotid study. He was presented with the options of "carotid duplex limited/unilateral" and "carotid duplex." The latter is the bilateral test, the default, but the user was confused,

"I want a carotid duplex, bilateral, where is that?"

Presumably seeing "unilateral," he expected to see "bilateral" as well.

Here, the issue of synonyms is also related. If some clinical concept is named differently from what a user is familiar with, it will have no meaning. As in all language, there is more than one way to say something. There are, however, widely recognized terms, abbreviations and acronyms in clinical medicine. Some are so common, the jargon used by all physicians, as to be the favored term and it is essential that the interface contain that term, at least. One user complained after a session about not being able to find items under what he considers extremely standard terms that are used by physicians everywhere ("every doctor in the country".) His example was urinalysis. It is not listed as "urinalysis" or the extremely common "UA" or "U/A", but as "U Urinalysis." He said, "This is stuff you learn in med school."

5.7.6 Navigating – Moving around the GUI

Users think about interacting with a computer in a dynamic way. They conceive the activity as *motion.* The users here employed many metaphors about moving in and around and through some space: "How do I get out of here?" "I gotta get back." "Let's see if I can go there." This is similar to "Locating Items on the GUI," but has a difference in that it is more like traversing, maneuvering, trying to make your way through the system as a space.

While on this trek, the users do their work, entering text, clicking icons, looking for the things they need. This code seemed to co-occur with several others: Locating Items, Data Entry, Action Specification, Perception and Interpretation. It was also associated with "guessing," to be mentioned below, as the users needed to figure how to "move around," "go back" or "get outta here." To the extent that smooth motion was impeded, usability issues occurred. As mentioned elsewhere, occasionally users became "lost."

5.7.6.1 The language of motion.

The following quotations show how the users navigate through the interface and use the language of motion. The first user clicks on icons as a way to travel:

"Again, I see that "COMMENTS" and I click several times just to get through. I always hit the "END" to get back to my main admission order set."

Guessing the way is often involved:

"Can I click through this? Look at that, I can! OK, excellent."

This user described a convoluted passage through the interface space:

"OK, so now I have to go, in order to get back a screen, I actually have to go to the home screen and go forward again. Um, actually I probably didn't have to do that for this particular one, but in other, if I wanted to go to the note section, because of the way I have things ordered, I'll show you. So it goes in this particular order. So if I'm in the [Order Entry], I can't go back to the clinical notes, I have to get back to the home, and go forward to, which that is pretty time consuming. OK, what was I doing?"

5.7.6.2 Getting lost.

If a user is not in the location she thinks she is, she may get "lost." Here a user failed to click on an icon that would take her to the next screen. Because of that she was not at the place in the interface she thought she was, and she started entering orders in the wrong place.

"Oh shoot. …. I actually guess it was where I wanted it in the first place. OK. I always forget to click on Comments and I'll start typing my next order into the Comment screen, and I end up with this crazy comment on some other order. Have to go back and fix it."

5.7.6.3 The hierarchical structure.

A hierarchical structure of menus is a common and useful computer interface convention. Because all the information in an interface cannot be placed on a single screen, an organized system of menus and submenus exists. The higher order headings are on an initial screen. Selecting one of those takes one to a lower level on a different screen, frame or window. The user needs to be aware of "where" he is in that structure.

Here the users used language of "up and down" or "back and forth" and they traverse the hierarchy in one of two directions.

"And to get out of here, back to our list."

"We'll click back into our order set"

This user, for some reason, did not like "moving backwards":

"Again, I don't like going backwards here, but I'll do it."

"Let's go END; I actually always don't like clicking out of that"

"OK. End. I don't like clicking out of these again, like I just said before"

5.7.7 System state

[Note: this is also a subsection of "Interpretation," 5.6.5.2, but is applicable here as well.] In the cognitive model above, the state of the computer changes once the user interacts physically with the computer. In the model, the entire "gulf of evaluation" (See Figures 3 and 4) is related to system state in that the user must perceive, interpret and evaluate the any change in the state of the computer with each cycle of the Human Action model. The concept thus overlaps with other codes; in fact, it is described above as a subsection of Interpretation. Here, the term is used more in terms of getting adequate feedback as to what it going on with the computer. Often, the user was not aware of what the computer was "doing." Typically this occurred while activity appeared to be "quiet" and the user was waiting for something to happen.

5.7.7.1 Nothing is happening.

If there is a delay in response time, the user must wait until system has completed its operations. The users were used to this and made many remarks about waiting for the computer while it is "thinking." Throughout this the users are often not given any indication as to what it happening and sometimes are not aware when the system is ready for further input from the user. One user was aware that the usual convention of the "hourglass" was not used in her system, but that a message was given on the bottom of the screen. Others did not seem to notice this. One user was waiting and expecting a drug interaction alert to appear. She waited for 45 seconds, while the system was indicating "ready" in a location she did not notice:

"It's thinking. It's still thinking. …Actually, maybe there are no drug interactions. That's a surprise."

5.7.7.2 Unsure whether the system "knows"

Other than not knowing what is going on when the system is quiet, the user needs confirmation that data input will be carried out. If a user places an order, he must have some assurance that this has been done and the system "heard" him. As mentioned above, in one system the user placed the order on the right-hand side of the display and it appeared among the list of active order on the left. However the list of active orders was larger than the display and one had to scroll to see the whole list. Part of the list remained in view, but often, at the point on the list where the new orders were being place remained out of view unless the user scrolled to view it.

Another user expressed dismay at not being sure whether the system would carry out an order at the desired time:

"OK now, this is a little confusing to me. I'm hoping. Next am draw. It's 11:34. Theoretically that's going to go in for next morning. I can be surprised by this because of working the night shift, sometimes."

The user needs confirmation and feedback from a computer system to maintain the confidence of "being sure."

5.7.8 Cognitive Overload

One of the greatest rationales for using information technology in health care in the first place is that there is too much information needed to function without external information resources. However, when the computer system itself presents an excess of information at one time, its efficient use is impaired. In this study, users were occasionally overwhelmed by the contents on the screen. They could not see the trees for the forest. This was also noted under a subsection of “Perception” and constitutes to a large extent “Locating Items on the GUI.” The following are more examples of users expressing frustration at having to wade through a large group of items trying to find something they all ready have in mind. In addition to frustration, one often heard the user expressing concern about committing errors:

“Is that [lab test] on here? I use this form a lot and I still don't know what's on it. `Cause there's so much. But, there never seems to be… there's at least one that's not. Oh, well it IS there, now that I said that.”

“Now he needs AM labs…Looking at the AM labs form, as usual there is a ton of junk there and it all looks pretty similar. I've used this form a million times and I still like have to carefully read it, to make sure I am getting what I want.”

“I find it very hard to stay organized, because I am clicking all over the place and doing a bunch of different things. It's not as nice and neat as it is when I write it down, and it's easy for me to forget what I am trying to do, as I am trying to make something else work.”

“Common nursing orders ….. Tons of stuff to look at. None of which on first, initial read-through is actually what I want. Now, once again I am getting slowed down by something silly.”

As already discussed, the presence of long lists in drop down menus is a recurrent complaint. This is another one:

“Instead of Q AM, I got to make it BID, and in the 30 or 40 options I find BID and click it.”

Associated with the idea of cognitive overload is the idea that the system itself is too complicated to learn. This was mentioned as “Lack of Knowledge of System Content.” Is there simply too much to learn to operate these systems as they now exist? Although a significant related issue, it cannot be determined with this methodology.

5.7.9 System Response Time

It may be a truism to say that all computer users want the system to run faster and delays are universally frustrating. In this domain, in which CPOE is expected to replace another way of entering orders, delays in workflow are a particular impediment to user acceptance. In two published editorials about “Ten Commandments” of systems implementation in the clinical informatics literature, the exhortation that “Speed is everything” was first in one and first and tenth in the other! (79, 80) This is an important usability issue.

The following are a smattering of the more polite utterances of irritation when waiting for the computer:

"This processing time always, ohhhhhhh!"

"…and in a minute it will come up and we are waiting, waiting, waiting and there we finally have our screen"

"Processing. Processing. Irritating. Irritating."

"Ah, c'mon. c'mon, c'mon."

There is no way to know this, but I sometimes had the impression that some of this frustration was "testing artifact." That is, perhaps some users were behaving more dramatically as they were being observed. That does not, however, gainsay the idea that some users often want the computer to work faster than it does.

5.7.10 Visibility and attention

For a user to find or attend to something on a display screen, it must be visible or stand out distinctly from other items. This HCI phenomenon is tightly linked to the cognitive concept of "Perception" discussed above. Like "Perception," it is hard to know when something is not seen, as the user cannot comment on something unseen. If they comment about an item, they must have seen it. I made a fair number of inferences about this phenomenon, but even then one can only speculate why something was not seen.

5.7.10.1 Visibility.

One example in which it was apparent that a user could not see an item occurred while looking for a lab test. When selecting from various options on a laboratory “form,” the user needed to decide between “tox panel 1” and “tox panel 2”:

“Urine drug screen. Urine drug screen. I don't think that's on here. Oh, I'm scrolling around. OK. Wait. Tox panel? Urine. Tox panel 2. Hmmm. I don't know the difference between the two. OK at this point I would default to 2, thinking it will catch more.”

What the user did not see was that when she hovered the pointer over those items, the contents of the panel were spelled out in a title bar, as seen in Figure 9. That box changed from “Drug level” to the details apparent there, but the user did not see that.

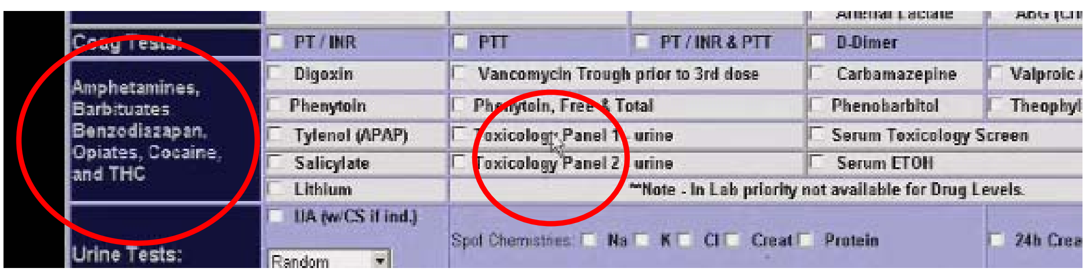


Figure 9. As the user placed the mouse pointer over the lab test, the contents of that panel were illustrated on the left.

Though none of the users mentioned it verbally, I noted how the use of different lettering makes items more or less difficult to visualize. Specifically the use of upper and lower case can make a great deal of difference. It appeared variable, even within the same system. This was most apparent when displaying medication dosages. These were mostly written in lower case, but occasionally all in upper case. Figure 10 demonstrate the difference between two types. In the left-hand example, the medication prescriptions are written with a liberal use of upper case letters and may be easier to read than the example on the right. This is similar to the user of tall man lettering (81).

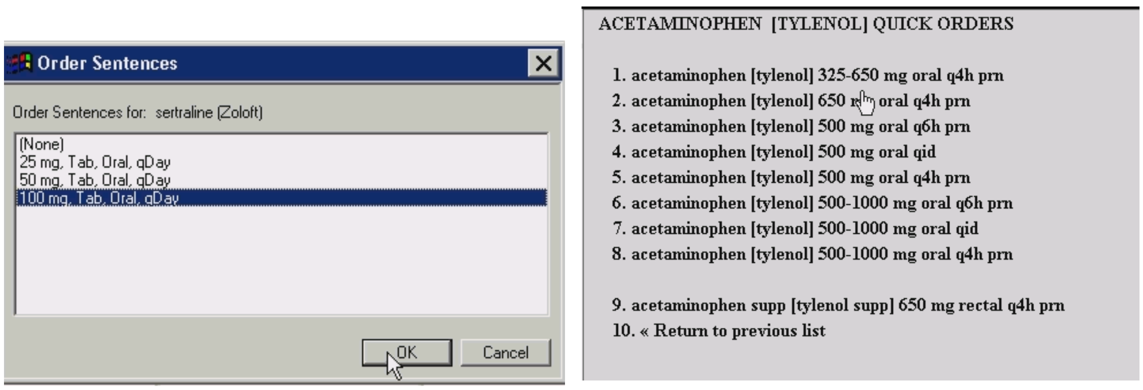


Figure 10. The use of upper and lower case lettering makes a difference in legibility.

When discussing the matter of visibility, there is a common occurrence that may be related to this. Users very frequently dismiss alerts very quickly. I have discussed that they ignore alerts. The alerts appear on the display very briefly and the user often briskly overrides them with a mouse click without as much as a mention. The speed with which they dismiss these items certainly suggests they are not reading them carefully. This may be a situation where something is not seen, but only because it is ignored. The fact that these alerts get ignored is related to poor usability and will be further discussed below.

5.7.10.2 Organization.

The organization of items on the GUI makes a great deal of difference how easily they may be seen. This has been described many times thus far in this paper. Well laid out order forms made it easier to find items. Users repeatedly commented that long lists of items are difficult to negotiate; they simply cannot see an item they are seeking.

Figure 11 shows two example of a list of orders in two different order sets. These examples come from the same system in the same hospital. The left-hand item is from “ICU Routine Orders” and the right-hand screenshot is called “Nephrologist Admit Orders.” The right side example separates the different kinds of orders by headings such as “VITAL SIGNS” or “NURSING” and seems to be a much easier way to find the orders needed. This could not be discerned from the verbalizations of the users, however.

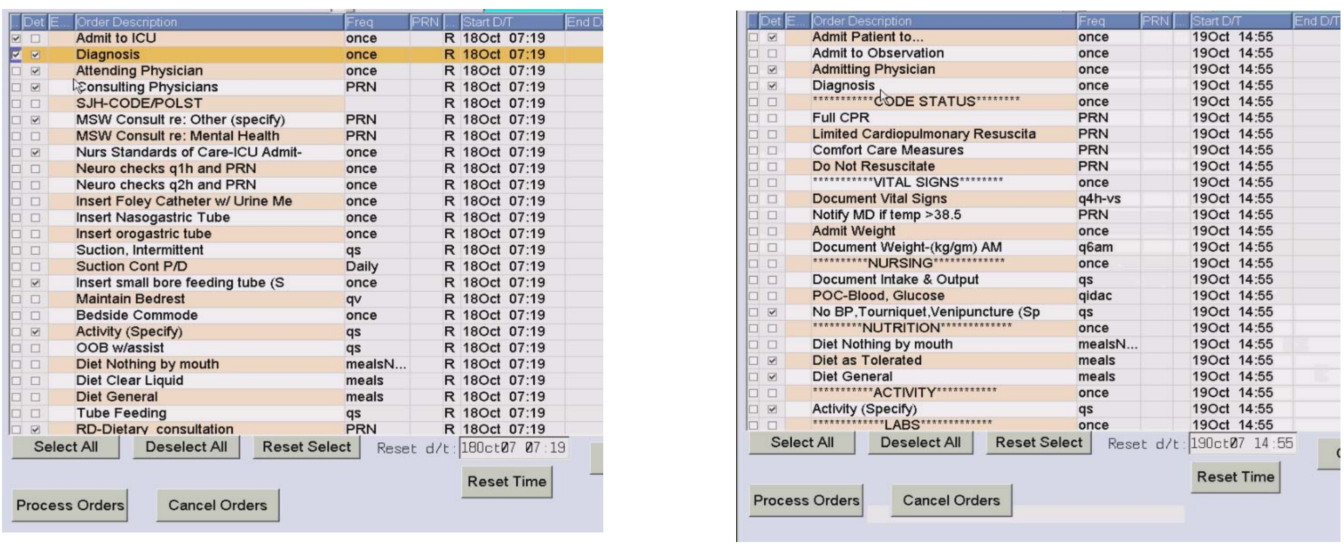


Figure 11. Organizing long lists with subheadings improved visibility.

5.7.11 Summary of results from HCI codes.

The above findings were produced by analyzing the raw data using a coding scheme that was derived from a list a heuristics that was compiled from the literature. This analysis produced 10 themes (see Table 19).The most commonly noted usability issues related to problems locating items on the graphical user interface, a lack of knowledge of the computer system itself and the time and effort of data entry. Many of the individual problems were similar to those seen when analyzing the data from the viewpoint of the cognitive model of computer use in the previous section, but this analysis also produced some unique insights into usability issues. In the following section I will present results from an analysis of the data from grounded coding. The grounded or open codes (71) are ideas that arose from review of the data without any pre-existing framework. They were created and refined throughout the analysis process. The results of that third analysis are shown in the fourteen themes described below.

5.8 The "Grounded Codes" – Cross-Cutting Themes. Results of Analysis Using a Grounded Approach.

5.8.0 Introduction

The other two coding schemes looked at the raw data from specific points of view. The first examined the data with respect to the users' cognition with the framework of the

Norman cognitive model. The second perspective looked at the user data in so far as they violated some heuristic of usability. Both of these perspectives noted some interesting usability issues. In some cases the concerns that were raised overlapped between the two coding structures. Again, those two coding schemes were created *a priori*, the first are the steps of Norman's Human Action Cycle and the other were generated from known usability heuristics.

While each of these coding schemes provided a unique way of approaching the data, I also analyzed the data in a more grounded way. During the analysis phase, data were coded with ideas that were not a part of either pre-existing schemes, but emerged from the data "spontaneously." That is, they are answers to the question, "What is going on here?" Looking at it that way, some cross-cutting themes arose.

Themes from Grounded Coding	
1	More work
2	Trust
3	Resignation
4	The problem with bad alerts
5	Guessing the action specification
6	The Ignore Phenomenon
7	Tolerance of Recurrent Problems
8	Unhelpful defaults
9	Use of the Order set
10	Issues related to long term memory
11	Opt in or opt out
12	Mindless data entry
13	Misleading the User
14	Synonyms

Table 20. Results from grounded coding

Because the results of the two previous sections were created by employing the Norman cognitive model and usability heuristics, these potentially may be applicable to any computer system. They were further analyzed with the additional grounded codes to express some meaning upon which one might make actionable recommendations. The themes that are presented in this section arose from these data specifically; hence they may be more relevant to this type of setting, the use of commercial CPOE systems by community physicians, or CPOE systems in general.

Of the 249 data segments that were coded, 159 were coded with this last category and these codes were used 227 times. The resultant themes are listed in Table 20 and each will be explained more fully below.

The order in which the themes are listed in this table relates to the number of data segments that were felt to be representative of this category.

5.8.1 More Work

At many times during the observations, it appeared that interface issues frequently caused additional and unnecessary work for the user. It has long been a concern that CPOE simply created more work for the user. This concern is one of the chief barriers to user acceptance (15, 22, 82). This finding is not especially unexpected, but it is included in these results as it was an extremely common observation. Of all the segments that were coded with these post hoc codes, this category occurred most frequently.

In many respects, this is a similar phenomenon to “Time and effort of data entry” above and there was some overlap in the coding. Here are a few examples. One user noted that orders placed in the emergency department immediately prior to admission that have a “stat” will occasionally be carried over to the admission orders and she must change all of them to “routine status.” It does not happen every time, but she is on the lookout for it. Her quote is:

“And that's one thing that happens that I should probably comment on, that if the patient hasn't, if something happens in the ER where all the meds get ordered STAT, and if they haven't been, if their status hasn't changed by the time they reach the floor, often times these are all still STAT and I have to go back and change them all, but we'll see what it is when I go to sign it.”

Another user complained that in order to write orders for mechanical ventilation on an ICU patient, she had to look in another application to find the current settings. She had to jot those down on paper first, and then return to the order entry application to record the order electronically:

“In this particular case, I am going to expand the quick view real quick to get her current ventilator settings, because I am going to need to reproduce ventilator orders, within the system and I am going to get that off of some the charting from RT. And I'm just scribbling that down on my paper because there's not any way to convert it from the

Quick View to the order set, because the order set locks you out of data extraction."

There were many instances in which the users felt they had to duplicate efforts and these following quotes demonstrate the users' frustration:

"Delete those two. H & P consult in the chart and the diagnosis has already been captured somewhere else. This seems very redundant. I don't know why we even did this, but…"

"De-select the ones that we don't want. It would seem to me that this should be a one click operation, but unfortunately now we'll go through this whole process and revisit each one individually."

"And I'm processing these orders and even though I have already deselected diet this pops up. Nope, there it, guess they want me to order it again as tolerated. Small waste of time but nevertheless."

"Social services, yep, this has been with us for a couple years, I get to select social worker twice which is a waste of time."

Excessive alerts and warnings as well as unhelpful default values were a recurrent problem and are both described in detail below. The users also commented on them with respect to how much time they wasted for them. One system in particular seemed to take a great deal of time to initialize an “admission” order set. This seemed to be the case over several different specialties. Here is a quotation:

“A real long way to just open an order set. Kind of defeats the purpose of an order set saving time.”

Lastly, the following quotation exemplifies one user’s annoyance with work duplication, order sets and his desire for personal customization to save time.

The thing I think is frustrating is we select the orders once and then we go through individually and click them all off again which is redundant. It seems like if you're a physician writing orders, you write them once and you don't… that should be sufficient. Also, it's annoying, you'd think, since these are all canned, we could have a one-click to say, "Please do my routine orders" and maybe accept number nine or something but it isn't. The way it's set up now we have to go through each order individually. So it's tedious - it should be a one-click kind of deal. It's actually much slower than it was on paper.

5.8.2 Trust

The users exhibited varying degree of trust in the computer system. A user has to feel reasonably sure that the action he thinks he is carrying out is actually occurring. (This was discussed under “system state,” 5.7.7.2.) Users repeatedly were leery about the accuracy of information presented to them and at other times they were skittish that some input would ultimately have an unexpected or untoward effect. Users were skeptical about certain computer functions working consistently or that the system would malfunction and cause them to lose data:

“There's supposed to be a way that you hit control and highlight them all, but I have found when I try to do that it doesn't work.”

“But because my experience has been, if I go into the order set and try to do it at the end, if I get kicked out then I lose all that work.”

The users took extra steps to make sure that their orders were carried out:

“I'm going to change the "routine" to "as soon as possible" so hopefully they'll do that tonight. Actually I'll write "please do it tonight" to make sure. "Please do tonight.” Click.”

Here a user hoped that a little additional courtesy would do the trick:

“I'm going to ask the nurses to get the PCP records. I'm going to type "nursing.” Pull up Instructions to Nursing. "Please!” I type in "please get records from PCP.” And I see that got on through.”

Alerts and reminders were a common problem that raised issues of users’ trust. The users’ had to weigh in their mind whether the information presented in the pop-up was reliable, accurate or applicable to the current situation. Here in this instance, the user got an allergy alert for a drug that is related to Ativan. At first he did not trust the alert and thought it was permissible to administer the Ativan anyway. He then hesitated, lamented the untrustworthy alerts in general, but then ultimately heeded the alert and did not order the medication:

"I think we'll just go ahead and give him Ativan if he needs to sleep. There shouldn't be any problem with that. I … well… This is one of the problems with [this system]. The allergy indicators are a little overzealous. I think I'm going to hold off on the sleeper altogether. If he needs it then I'll think about that later."

Then he changed his mind moments later and decided to prescribe the drug after all. There is clearly some ambivalence about the system:

"I do want him to have Ativan in case he is withdrawing. I don't see he'll have a problem with that, so I'm going to say benefit outweighs the risk. OK. We'll give it IV if need be. Override the warning. This is kind of nuisance, having to go through all this."

Sometimes the users are surprised when things work out as expected. Here a user acted surprised that "Tylenol 3" (an extremely common drug) appeared in a list for which she searched with the phrase "Tylenol" (nearly the exact name in its entirety, such that if Tylenol 3 was there, that is highly likely to pick it up). Does this suggest an overall lack of trust in how this system returns the desired orders?

"Tylenol 3. See if I can pull that up. Tylenol… Number 3! Excellent!"

Although a lack of trust of the system was common, occasionally users exhibited what might be considered an overreliance on the system. One user ordered two narcotics first and then questioned whether any allergies existed:

"I usually order a low dose of Vicodin for elderly patients, for PRN, and then a low dose of morphine, for PRN. I don't think he has any allergies."

She then waited for an alert to appear in order to make the determination whether or not this patient is allergic to the medications just ordered. After that was not forthcoming, she stated:

"Nope. And then I'll order hydromorphone or Fentanyl if they have allergies to morphine."

One would think that if she endeavored to understand whether allergies existed, she would simply look at an allergies list, either in the system or in her notes. She waited and said "Nope" when nothing happened, thus confirming no allergies to the ordered drug. In this last example, the user showed a great deal of reliance of the system. He treated it as a trustworthy guide:

“I pretty much trust the hospitalist admit form, 'cause I've done this several times now, so I just go through this and fill it out. I'm clicking on Vital Signs. I'm kind of reading these as I go to make sure I don't miss anything.”

“Clicking on the Med Notification form. I've done that a million times so I'm just accepting it.”

“I need to stick with this form or I get lost at times. It's helpful to walk me through.”

5.8.3 Resignation

This describes circumstances in which the user encounters enough difficulty in proceeding with the desired action that he stops what he is doing. The user then resorts to changing the intention somewhat, often doing “less” than was intended or giving up altogether and simply forgoes the task.

5.8.3.1 Resorting to an easier course of action.

In some circumstances, the users encountered difficulty and decided to change course. In this example the user was ordering an echocardiogram and the application requested that she enter a reason for ordering the test. This is standard practice and gives the technician and the cardiologist interpreter enough information to perform the correct test and understand the findings. Here the user was trying to do the right thing by entering a reason in the comment section, but the UI prevented her from doing so, because her comment contained more characters than allowed by the system. She could have rewritten it with fewer characters but she just gave up in frustration. Thus she gave up and simply omitted the task writing a reason for a test:

“OK, Decompensated CHF. Check cardiac ….. UGGHH. All right, never mind. They just don't get to know what I am looking for. OK. So much for the echo.”

The “comments” section is often the “workaround of choice.” There is typically a location in an order entry system called “comments” or “miscellaneous” that allows the entry of free text. It is reserved for information that is not already listed somewhere else. Because free text is entered, instead of structured text that a computer can understand, its use is discouraged. However, often this user settled for the comments section because it is easier:

“She wants a new PCP. Can I put that in the computer? "Please get her a new PCP" "ACM referral" is what that is. ACM referral, although really it's discharge … referral for discharge planning. Enter. OK. I'll actually put something in the comments if it's here.”

In another example of resorting to the comments section, this user was desperately trying to order a bedside commode for a patient. (This example was discussed under Action Specification.) The user tried twice unsuccessfully to perform this task over a period of four to five minutes, and on her third try she used the comment section. This has the effect of passing work on to someone else.

“Uhhhhhh, Alright so, I'll do my fall back. I'm not finding bedside commode anywhere. Tried there and I just can't find it. Soo… I will do instruction to nursing and just ask them

please provide bedside commode, There we go. The computer's a little frustrating but the nurse's are pretty smart. They will figure that out."

5.8.3.2 Altering the course of action.

In some cases when the user encounters a roadblock, she changes the order slightly to something different than originally intended which might be easier to carry out. One user wanted to admit a patient to the Progressive care unit, but could not for some reason and admitted him to a medical-surgical unit:

"OK. Admit. To PC …. OK. To PCU. Hmmmm? [changes mind when PCU didn't seem to work.] I'll say Med-Surg.

Here is a case in which the interface guided which diet order the physician chose:

"We're just going to go with general diet <u>only</u>. Because <u>this</u> one is going to make me click lots of things, I know that. <u>This</u> one is not going to make me click as many things."

Another user changed the lab test she wanted to order because it was easier. She intended to order a serum sodium test, but it was not contained in the "common lab form" area in which she was looking, so she ordered a panel of six tests that includes sodium instead (five of those tests may have been unnecessary). She could have left the "form" and simply ordered a sodium level on its own. She gave up because she had spent a fair amount of time (45 seconds) looking for the sodium test that was not there, so she settled for the basic metabolic panel.

"I want to check her sodium this evening as well. I'm going to Common Labs form to order a serum sodium, which I am looking for. I guess I'll just order a basic panel."

5.8.3.3 Settling for less than accurate information.

Occasionally the user settles for less than accurate input. Here is an example of a user reviewing the list of the patient's allergies. An allergy list is certainly an important issue to establish as accurate or not. In this example, the doctor felt the information was inaccurate, that the meds listed as having allergies were only intolerances, not true allergies. She stated that she did not know how to make the changes to the allergy list. She did not try, however, and was resigned to have the record in error.

"So allergies. I think they're in here. Let me look and see if they're in here. Um, yeah. Now, I've never tried actually deleting allergies. She says she's only allergic to penicillin and these are probably intolerances. I'll just leave them there."

Here a user changed a medical decision, in this case the dose of a medication to avoid the aggravation of changing from a default value:

"OK, out of sheer laziness `cause I'm on this page, I'd almost go to 100 BID, seems kind of high. I'm going to do 100 a day. I'm in a hurry and I'm defaulting to it because it's right in front of me. I don't think it's that big of a difference. 100 a day versus 50 BID, so I'm just gonna go for it and be done with it."

In another case, the user made a mistake in data entry and left it incorrect rather than bother with fixing it. He was ordering “PRN” medications for a patient on a monitored bed. Immediately noting and addressing cardiac electrical irregularities is the point of continuous cardiac monitoring. The user must enter parameters against which the nurse will judge whether to administer medications. First, the user was prescribing nitroglycerin sub-lingual tablets. The interface stated:

“nitroglycerin sublingual 0.4 subling q 5 min prn chest pain if sbp > ____.”

And user entered “8” in the blank space for SBP (systolic blood pressure) greater than 8. That is, of course, far too low a number (a systolic blood pressure is generally greater than 100) and is incorrect. He was anticipating the next entry:

“Lidocaine bolus 50-100 mg iv prn for sustained symptomatic v tach or > _____ pvc’s in succession.”

Here is value of “8” was appropriate. The patient was to receive lidocaine if the monitor shows greater than eight PVC’s (premature ventricular contractions) in a row. Here is the monologue:

“Let's go ahead and give him all the cardiac PRNs

[he writes: if SBP > 8]

And I think that's oops. I need to go back.

[He meant to put 8 in the next one, >8 PVCs]

And I got myself into a pickle here but let's … I entered a number that I thought was in the lidocaine field but it wasn't. So I think I'm actually just going to ignore that."

He chose to settle for the inaccurate data, rather than correct it.

5.8.3.4 Giving up altogether.

Occasionally the user simply forgoes the order and doesn't try any more. Some of these examples have been given earlier, but they are appropriate here as well. In this first one, the user tried to order a medication drip based upon vital signs and simply could not figure out how to do it:

"OK this just isn't going to happen automatically easily. Let's skip it and I'll check his heart rate later."

Here, a user wanted to order a social services referral and give the appropriate accompanying information, but he was stymied:

"And let me see, find in there, so under the social work, I want to put a comment. I'm not sure it gives me a, allows me to make a comment. OK. So, here's a problem right here I'm not even sure I know what to do. So, I will have to just go and talk to the social worker myself."

Lastly, this user abandoned the computer completely and decided to use the paper chart to enter the order:

"I just don't trust this thing quite honestly. And I think I'm going to write it in the chart. So… that's exactly what I'm going to do."

5.8.4 The problem with alerts and reminders

It should come as no surprise that observations of users of CPOE would exhibit usability problems related to alerts. Alerts are an important part of automated clinical decision support and are typically an unsolicited message presented to the user in response to some recent action of the user (83). There are numerous references to this problem throughout clinical informatics literature (84-86) By far the most common issue is that these fire too commonly to be useful and are more irritating than helpful. The finding that the CPOE users in this study were dismayed by alerts might be as astounding as noting that email users dislike spam.

However, this is a common problem and this examination would be remiss without discussing the usability problems of alerts. As is well known, users dismiss these alerts at alarmingly high rates (87, 88), and these observations did not deviate from the norm. Except for the alert that was begrudgingly heeded in the above section in "trust" (and then ultimately dismissed), there was not a single instance in these recordings in which the user did not override an alert. The disregard of the pop-up alerts sometimes occurred in such an offhand way that this behavior was regarded as the "ignore phenomenon," to be discussed below.

Here is an example in which the user hoped an alert that asks for a justification for the use of an intravenous medication would not fire, though she feared it would. She was transiently relieved that the alert did not present itself, only to be disappointed:

"IV, uh, IV use only, that's right. 400 mg dose. OK. Good, it's not asking me to tell them why I'm using it IV. No. Yep, it is going to ask me that. How nice! She's nauseated! I know what I am doing, OK?! Comment? No, no comment!"

In another segment, mentioned previously in a different vein, a user waited for nearly a minute for an alert to appear that never came. The computer system was idle, but the users are so accustomed to these admonishments, they come to be expected, even if they are not forthcoming:

"OK. So, everything looks good. I'm going to sign. And often times it pops up with drug interactions of various sorts which are usually not significant, but occasionally are. It's thinking. It's still thinking. Actually, maybe there are no drug interactions. That's a surprise!"

There were observations of some particularly egregious unnecessary alerts. In attempting to update the medical record about a user's allergies, one user clicked "no known allergies," an item listed under "allergies." The system warned him that he could not enter an allergy because the patient was listed as having "no known allergies."

There are many drug interaction alerts that the users seemed to routinely experience and they breezily ignored them. A decision support system will alert the user if duplicate orders are entered. In one user's post-partum order set duplicate order alerts fired five times for items that were in the order sets and already preselected by default. The user did not select any of these preselected items, he merely clicked the "submit" icon and a string of alerts fired. Among these were alerts for oxytocin administered by two different routes, three topical corticosteroid preparations, a topical anesthetic and the anesthetic the patient received in delivery. These were all standard orders; that is why they were in an order set in the first place. These alerts thus fire every time the clinician enters post-partum orders as they were listed by default in the order set.

There were other examples in which the "default" behavior of the system was to fire an alert every time, because the order set contained an item guaranteed to trigger an alert. Using the same drug with two different routes, an extremely common and sensible thing to do, was commonly implicated. As more than one route of administration was desired, one drug must be ordered as two separate orders, but this fired an alert.

In a single instance, I observed a user possibly erroneously dismiss an alert. The clinician was entering an order for a patient with respiratory problems. The patient was previously prescribed and was already receiving an albuterol inhaler. The user wanted to order another inhaler that contained a combination of medications, including albuterol, thus a duplicate order alert fired. The user stated:

"I'm entering the Combivent order and there's already an order for albuterol 10 puffs as needed. So, I'm going to place that Combivent order. And I'm aware that there's two medications that are ordered."

However, the alert also stated the patient was allergic to salmeterol, a medication related to albuterol. Because it was on the same warning message as the duplicate med alert, the user did not notice it. (The patient was already taking the drug without problem, so the allergy alert was likely incorrect anyway.)

Some important issues about alert messages arose. There are numerous specific alerts that all users override. Some alerts are fired by "default" because of pre-selected items in an order set. Duplicate drug alerts are often in error. And, because so many alerts are ignored, users may not perceive an alert if it is paired with another he is likely to dismiss.

5.8.5 Guessing the action specification

To review, the term "action specification" is taken from the Norman cognitive model and refers to the actions that user has decided to undertake with the computer system to fulfill the desired intention. The user specifies in her own mind what steps to take in order to accomplish what she wants to do next. As mentioned in the above discussion of Action Specification, sometimes the user simply cannot determine an action to specify. The user then has two options, change to another intention and try to specify the action steps for that, or persist with the current intention and take a stab out how to carry it out. Knowing the conventions of the computer system, the knowledgeable user may well be able to make some very educated estimations of how to proceed and likely will ultimately be successful. However, there are times when the users are confused and appear to be just making it up as they go along.

This phenomenon overlaps with other themes. It is discussed under Action Specification, as expected. It also appears commonly with three other codes, “Locating items of the GUI,” “Moving around the GUI,” and “time and effort of data entry.” All three of these are related to navigating the user interface.

5.8.5.1 The language of guessing.

There are many different kinds of tasks that the user guessed at. There did not seem to be any underlying pattern to these tasks, other than the users’ conjectures. Here are a few of the samples of the user trying to move forward while confused:

“Let's see, do I pick it individually? What happens if I push here?”

“Can I click through this?”

“So, let's try this. Shoot!”

“Now, I wonder if I can just do this under infusion instructions.”

“... and see if we'll get anything…”

“So allergies. I think they're in here. Let me look and see if they're in here.”

Here are more examples of users trying one action and then, if unsuccessful, trying another:

“And then I want them to check orthostatics here and I'm going to figure out. So I don't want that. And I want maybe "other,” I'm going to see if it's in there. Oh, well, I guess I can't do that.”

“Let's see what happens if I write "stool studies" Let's see if it gives me options for things … that will go faster. No. Well, maybe it's under here. OK, the ones that I want are…?”

The following language suggests navigation and getting lost and guessing one’s way:

“…we're going to write "instruction to nursing" and hope I can get back to the other form in a minute”

“And then… return to … is that want I want? No, yeah, I guess so.”

In addition to guessing the computer’s actions, the user was confronted with medical decisions to speculate. These were typically policy issues with which the system forced the user to comply”

“I'm going to say "fair" condition. I never know what those mean really.”

In this last example, the criteria were listed on the screen. The user did not see those and needed to guess at the correct answer:

“I don't know what class she's in, let's see, how about we'll do a 2 because I don't even remember what the rules are for those but she seems like a 2.”

Should a doctor be trying to order something by guessing? In addition to the potential for making errors, the process will likely be time consuming if one guesses wrong. Although it would seem preferable to know exactly which step to take next, if one reasonably knows the system, the user likely will make some good estimation of which actions to take.

5.8.6 The “Ignore” Phenomenon

Users appear to ignore many items that appear before them on the display. This phenomenon occurred quite frequently and it was essentially always related to alerts and reminders. This has been referred to above. I could only infer its occurrence; people do not tell others they are ignoring something. If one mentions it they cannot be ignoring it. However, there were many times in which an item seemed so obvious to my observation that the user most likely must have been making some attempt to refrain from noticing it. With respect to alerts in pop-up windows, the user must click on them to remove them from the display. I am framing this as “ignoring” as the user does this so quickly that it seems impossible to have read the message that appeared.

This phenomenon was discussed above in the cognitive themes of “Perception” and “Interpretation.” Again, one can never truly know what a user does not perceive. If never perceived, the user is not ignoring. However, if a warning suddenly appears in the foreground in a pop-up window and the user takes steps to remove it, one may rightly suppose that he perceived it, but interpreted it as so insignificant as to make no mention of it.

The users ignore alerts without mentioning them. I find this has some significance in this context as it is “think aloud” exercise in which they are instructed to comment on everything that they see. Does the fact that they are not mentioning them in a think-aloud study suggest they do not notice them at all? Probably not, because as stated, they have to dismiss them with some override action. In addition, as discussed in “Subjective User Attributes” (5.2.4) in the above Results and in the Discussion section to follow, the users, in general, did not “think aloud” in an ideal way. They failed to verbalize about much of their activity. However, the users do verbalize most of what they are doing and the fact that they very commonly make no mention of these items speaks to how little attention or significance they give them.

With respect to the alerts and reminders, the users override them and get them out of their field of vision so quickly it looks like they are swatting away gnats. This was observed numerous times. It is impossible to capture an activity with a quotation that is not verbalized. There are a few instances of note, however. As above under “Bad alerts,” when two alerts are listed together a user may ignore one and miss the other entirely. In another example the user entered a term that was invalid in syntax. The system alerted him to that and asked if he want to correct it. He clicked “NO.” A few seconds later, the system asked him again. This time he clicked “YES” but made no further changes to his order. He agreed and disagreed with the same alert. It seems he did not read the item or did not understand the request. In any event, it seems apparent he ignored it.

One quotation demonstrates the user describing his habit of ignoring an alert:

"And then I do to his medications. And there's a little drug interaction that they're telling me about between pro-clo-, between compazine and potassium, which I hardly ever read, but here it is. I usually blow right by it and keep going."

5.8.7 Tolerance of recurrent usability problems

Throughout these sessions the users remarked about problems they note with the interface that detracts from its usability. Remarkably there are many times that they make no reference to some obvious detriments to ease of use. Whether they note the problem or not, they seem interestingly tolerant of some usability problems. The users dismiss unhelpful or intrusive warnings and alerts without breaking their stride.

In one system, a warning popped up at the beginning of every session. This happened after the first order was entered only and was never seen again. It said,

"This device does not have the default location defined"

This message showed up every time and everyone seemed to be somewhat aware of it. Most ignored it and clicked the window closed without comment. Only one user read it aloud, and said, *"That's OK."* I asked users after the session and no one knew what it referred to, but it had been presented for a long time. It appeared to be innocuous and it was simply tolerated.

It seemed that the users grew to expect this sort of thing. As discussed under “Alerts and reminders” above, some users anticipate delays and are surprised when they do not appear:

“Actually, maybe there are no drug interactions. That’s a surprise.”

One user was willing to chance a potential system crash, while undergoing the observation session.

“*I'm going to try something different this time. I'm going to try "chest pain" as a set. This blew up on me my first time I did it, three months ago and froze and I couldn't get it out, couldn't get out of it, so I never did it again and I heard maybe they fixed it in the last 90 days, so I'm going to do chest pain order set.”*

During many sessions the users complained about problems that occur every time, such as in their favorite order set. They sometimes have given up trying to raise the issue with the information technology department and simply tolerated it:

“Social services, yep, this has been with us for a couple years, I get to select social worker twice which is a waste of time.”

Duplicate entry, bad defaults, clumsy order sets area all a nuisance to the user, but they persevered. This may be a phenomenon of the type of users tested here, as alluded to, and will be reviewed in the Discussion section.

Lastly, some items that appeared every session were actually appreciated. It seemed then it was not the presence of pop-ups per se that the users object to. For example, in one system, after the patient was selected and the order entry was initiated the following appeared:

"You are about to enter orders on [patient name]"

One user was glad to see that reminder to help him prevent errors:

"And we are off. OK. So [patient name]. Make sure we have the right patient. Always a good idea to double check you are actually entering orders on the right patient."

5.8.8 Default Values

One of the most valuable features of medical order entry applications is the presence of pre-constructed lists of items from which to choose. As discussed in "Locating Items of the GUI," much of the process of entering orders is essentially choosing items already on the screen. The user may save a great deal of time by instantly clicking on items rather than typing out the term. This can be much faster than writing on paper. The presence of default values also constrains the options from which a user may choose. To the extent that this limits or directs a user's choice to accurate values, this may reduce the risk of medical errors. Indeed, the presence of default values from which to choose is one of the central mechanisms through which CPOE may promote improvements in quality and safety (89-91).

Of course, the default terms used most frequently must be present in the interface to have any value. When they are not present, usability is made worse. Conversely, the presence of numerous inappropriate or rarely used default values clutters up the interface. Lastly, occasionally there is a single value that is the most apt for certain circumstances. If this value is absent or is inappropriate and must be changed nearly every time, the advantage of these default terms is lost. Problems with usability that clustered into this category were either due to the lack of proper default values or the presence of unhelpful defaults.

5.8.8.1 Lack of needed default values.

The users complained about having to input data when it was required in every instance of entering a certain order. This seems to be the ideal situation in which to have a default value assigned.

This first excerpt demonstrates a common problem. When ordering a test, one must enter a "priority" to indicate when this test is to be obtained. For example, if the test is needed immediately, "stat" is the designation. In the following case, the user selected the preconfigured priority of "early am" but the requested date and time do not automatically reflect that and must also be entered:

"I also want to order an echocardiogram for the morning. So that's early AM. This is something that's bothersome actually. When I order an echo or certain x-rays and I hit early AM, it doesn't default the time and date to tomorrow morning, so I have to go change that as well and that should be, it would be nice if that was automatic."

Another user complained about having to change a configuration with every session to a format that gave her the information she needed. She would have hoped the configuration could be customized to the role of the user and she clearly states the reasons:

"Let me see what they gave him in the ER. That's why I always like to leave this, when I'm doing orders, on "All orders 5 days back", because then I can see what's already been given…One thing that would be nice for physicians is to automatically… this med list is defaulted to start at time, time now, usually, or shortly before now, and for we docs who often don't need to see things time forward, we need to see things time back, so it would be nice if it defaulted to that. But I can easily change that, it's just time consuming."

In the following, the user ordered an exercise treadmill test. It is required for that test that the patient takes nothing by mouth (NPO) after midnight if the test is done the following morning. The user felt that should have been done automatically after he ordered the treadmill test as it is necessary for the test:

"I don't see NPO after midnight for his stress test. That should have been… Let me see "as of". "As of" tells me … what … else is in there. And where's the diet. Oh, it doesn't say. Well, I'm going to have to put in "diet" and find "NPO after midnight" for his stress test."

In another segment, the user had to enter an order for "Nursing Standard of Care, ICU Admit" for a patient he was admitting to the intensive care unit. This is something that all patients in this unit receive and seems redundant to have to enter it separately. This was never required when using paper orders, so it appears that it is merely creating extra work for the physician. More importantly, if this is required of every patient in every ICU admission, it seems the ideal situation to have a default value in place in the interest of patient safety.

5.8.8.2 Presence of bad default values.

In the previous section lack of apt default values was demonstrated to be an inconvenience. The presence of erroneous values is likewise cumbersome as they have to be changed every time. Moreover, if missed and left unaltered, there is risk of committing an error.

In the following example, the user wanted to order a set of cardiac enzyme tests to be obtained at three specific times. As the quotation indicates, there is a "chest pain order set" that is available but doesn't work correctly. There is also a problem when selecting the "routine" option. The user must enter the times manually and the interface reasonably requires the entry of a date and time when a laboratory test must be obtained. There is a blank box accompanied by "up" and "down" arrows (See Figure 12), but if an "up" arrow is clicked, tomorrow's date appears and yesterday appears with a press of the "down" arrow. The user had some issues with this and simply hoped that there would be a default to the current day:

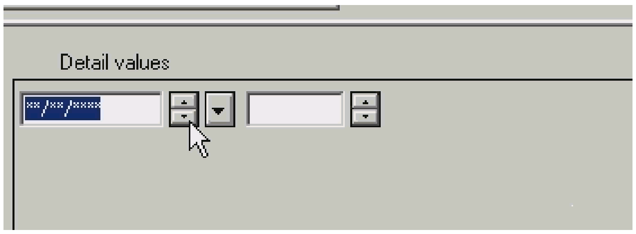


Figure 12. Text box for entering date and time.

"And the next thing that I want to do… I used to order the chest pain order set which had the CK Q 8hours times three, but I found that the patient would also get, would often get, like, a CK [at] 3 o'clock and get morning labs drawn at 5 and that would be 2 lab draws within 2 hours which means that the patient would get 2 needle sticks so I actually put the time in myself… I do a "time to draw" because when I do a "routine" for some reason it doesn't get done.

And that's something that I don't understand happens. If you do the time to draw and you go to the collection dated time, it doesn't automatically go to today's date so you can move it up and down… So here if I did a time to draw, and I go to put the time and date in, if I hit either of the of the arrows, it either goes above or below today's date when I first start and it would be nice to have it default to today's date. Just a suggestion."

One would never need to do a timed lab in the past. At least defaulting to today's date would seem reasonable.

In the following the user wants to prescribe a medication. The default value for the frequency with which it is administered is “Q day” or once daily. The drug would never be given in that way. The user notes that and complains that she must change this every time:

“OK, so I'm going to order promethazine in the low dose and then Zofran as a backup and this frequency is Q day which is stupid! So I do Q6. I always change that.”

In the following example, the default value in an order set is always incorrect. The lab test will get drawn at an inopportune time routinely unless the user makes a change. Worse yet, there is an accommodation to get the lab “in the morning” by selecting an “M”, but this doesn’t work and the date and time have to additionally be entered manually, despite the “M” value that was already entered:

“Now, we need to fix this because this is supposed to be a default to a morning lab draw tomorrow morning, but the way it's working out is it assigns an obscene time, like 10 o'clock tonight or one in the morning, so I just ignore this one. And I go here and order my lab again…. The way this is set up, if I put an "M," that designates it for a morning draw, routine morning draw, put tomorrow's date there. You got me, is today the 18th or the 17th? Today's the … there you go.”

Next is an example in which a post-operative order for a clear liquid diet routinely starts 24 hours hence, withholding food from a patient for an entire day. The user wants it to start immediately and must make the adjustments.

“It defaults to the same time I am entering the orders tomorrow and for some reason it defaults "clear liquids" to start tomorrow instead of starting today. So I have it start today.”

Here is an example of order sets that are inconsistent. For some reason, all labs are pre-selected by default, but none of the meds are. The user desires only some of the items to be selected by default. This is a change from the way the interface used to be:

“And then, on the old screens, this is sort of a new system here, they had some default check marks. In going to this system they have gone to no check marks and so it's kind of a pain in the butt, but you have to check everything that you want. And I have asked them, and they say that it is possible, to get a personalized check mark system here.”

The issue of preselecting certain items in a check list will be expanded below in “opt in/opt out.”

5.8.9 Use of the order set.

Osheroff et al. (83) define an order set as

> a pre-defined and approved group of orders related to a particular clinical condition or stage of care. Often, the order set consists of both diagnostic and therapeutic orders. The goals in creating order sets are to standardize care, increase compliance with best practices and facilitate the order entry practice.

An order set's capacity for facilitating the order entry process is a great benefit to usability. It is simply easier to have a group of orders which are frequently entered consecutively all together in the same location. This avoids the more prolonged chore of searching for each order individually. The order set may be the singularly the most valuable element that contributes to the usability of CPOE (92-94).

This was especially so in this study. The task that was chosen to be observed, the entry of admission orders, universally involved using an order set. On all the systems observed there were "admission" order sets. These were categorized by the user specialty (e.g. hospitalist), the location in the hospital (e.g. cardiac care unit or rehabilitation unit) or the patient's clinical condition or procedure (e.g. rule out myocardial infarction or post-partum.) For the most part these were a boon to usability by virtue of their convenience and capacity for streamlining the workflow.

There were many segments that were tagged with the code "order set," mostly because they were noted to effect a benefit to usability. However, they are not something the users commented on except for the occasional brief reading of the name of the order set on the display. Thus there is really no way beyond description to demonstrate this phenomenon in writing. There are some issues worth discussing nevertheless.

5.8.9.1 Groupings of orders.

Order sets themselves may be "nested" within other order sets. In a larger list of orders there are selections that themselves a smaller group of orders. The different order sets are hierarchically arranged. The ultimate "child" of this hierarchy is the order itself and selecting it places an order. However, the order itself may be a singular datum or a collection, a miniature order set, if you will.

These small groupings vary within and between systems. For example, an order for "Atenolol 50 mg. to be taken orally every day" may be ordered in some situations with a single click of the mouse upon a display item or may be ordered sequentially as the name of drug, dose, route and frequency in four separate steps that are driven by successive prompts. It is essentially a spectrum of order types from a group of information all at once or ordering each most atomic part individually ("à la carte"). While the individual steps did not have a name, the groupings were sometimes called "order sentences" or “quick orders.” (Although definitely specific to one system, the term “order sentence” may also be a more generic term for this phenomenon) (95).

While ordering a cluster all at once is obviously quicker and easier than using several individual steps, the two ends of this convenience spectrum had their own liabilities. The “express” manner of entering orders might be preferable but this assumed that each one of the details of that order sentence is desired. Slight variations of the default value may be wanted. For example, most drugs may be administered at various dosages, routes and frequencies depending upon the needs of the clinical situation. If the order intended is not listed among the “one-click” orders, the user must make accommodations. It seemed that altering a pre-defined order arrangement tended to be much more time-consuming than ordering each component “from scratch.” An example of this was given in the section above, “time and effort of data entry.”

Thus, any benefit of a pre-defined order or order set may only be appreciated if the desired order is present. In these observations, many deviations from the set menu were fraught with difficulty to the user. If a needed order was outside a grouping, the user needed to take extra steps. Consequently, default values are again of great importance to usability to the extent the ones chosen for an order set are the ones desired. The configuration of the order set must balance the need for including the most commonly desired items against including too many items that worsens usability.

There are some examples of these in the observation data. This first user spent some time looking for doses of acetaminophen and oxycodone within an order set. He needed to change course when he could not find what he wanted for his particular patient and prescribe the medication in an interface location outside that grouping. One size did not fit all:

"Going to PRN medication. Going to give her Tylenol around the clock. See what I can find…. Don't really have exactly what I am looking for in Quick Orders. Want to give her oxycodone, too. In a very small dose and they don't have that on the Quick Orders, what I want to give her. OK. Considering their "considerations," but I don't really like those either. And I'm going to type in oxycodone, which I couldn't find before on the quick orders. I'm skimming down, trying to find the one that is NOT a quick order. I'm going to put in a different dose; I'm going to do 2.5 mg to 5 mg. because she's a teeny 95 year old woman."

Lastly, one user expressed dismay that orders for mechanical ventilation were not in an order set for the intensive care unit, something that is commonly used there:

"And then unfortunately, despite that being a critical care order set, it would appear that the ventilation orders are not built into that."

It appears the decision of which order to include and exclude from an order set are crucial to its usability.

Another issue related to orders sets is their potential to affect user behavior. While this is the raison d'être of the order set, the possibility exists that it may influence user behavior in unexpected or undesired ways. This is discussed below under "Mindless Entry" and "Misleading the User."

5.8.10 Issues related to Long Term Memory

Users need to access long term memory constantly, of course. Remembering how to use a mouse or the name of a patient requires long term memory. This observation refers to usability problems that arise when users cannot remember something they have already learned. Certainly users cannot remember everything. But if an action specification is not immediately remembered, a highly usable interface can guide the user by providing starting points and "walk him through" until he finds his way. A basic psychological principle of good interface design is that it is easier to recognize something that it is to recall it (96). An interface can provide clues to raise the activation level (see 2.1.5.3) of a long term memory so that it might more easily be recovered. This concept was mentioned above under "Action Specification" and "Knowledge of System Content."

One of the more frustrating situations for a computer user is attempting to perform an action that she knows she has done on a previous occasion but cannot remember how to repeat the process. The example of the user who could not find the order for a bedside commode has been described twice above, but it is a very dramatic example of this problem. Several times throughout the session, she repeated her effort to order the cursed commode only to postpone it after failing:

“Let's me try it one more time, I guess.”

She spent a large proportion of the fifteen minute session in that futile attempt. Another user had difficulty right from the start and could not find the admission order set. This was an action he had done many times before; it is how to start the session. He would not be able to proceed at all without that action. This was not a trivial detail. His memory of the interface was such that he claimed he had not seen that display before.

“And, ummm, care sets. I wonder if there are... any… now where are the admission orders? Wait a second, they're around here somewhere. This is not the form that I am used to.”

Users variously could not remember items they used commonly, not just esoteric items. Equally surprising, however, were the times in which users remembered items that were not at all intuitive. For example, in one system a “timed” lab order was entered by writing “4/12 8am timed” to obtain the test on April 12 at 8:00 a.m. This is an action that must be memorized in order to carry out.

“Guessing the action specification” enters into this. If something cannot be remembered, being a good guesser helps. This user demonstrated that:

"Yes, I was going to order some optic drops for his conjunctivitis, so …. I choose some eryth-ro-my-cin ointment, type in "erythromycin oint'", see if we'll get anything… I haven't put this order in before… There is it, "eye ointment.""

One user worked intermittently in a hospitalist capacity for her internal medicine group. She and her partners rotated this duty for a week-long shift, once every four weeks. She remarked that it was difficult to remember the system after an absence. She stated:

"It's tough to maintain your skill level."

She starts her shift on Monday and

"…by Thursday, you get good at it again, and then your week is over."

There are many interface details that can serve to help the user when memory fails. Good organization of forms, the generous use of synonyms and well laid-out menus all guide the user. Successive prompts that naturally mimic thought processes or workflow serve the user so he does not have to remember every detail on the system. The principle of recognition rather than recall is crucial to the usable interface design.

5.8.11 Opt In/Opt Out

As stated, the order set is an important asset to usability in computerized order entry. The user is allowed to select the desired order from an organized list. In some circumstances orders may already be pre-selected and the user submits them en masse with a single click of the mouse. In this latter arrangement the list of orders is accompanied by check boxes. The list of all possible orders is displayed and the user may “check off” the ones he wants. Some of these orders are already checked by default and will be ordered when the user submits the entire group. If something is not desired, it must be deselected by removing the checkmark. Thus the user may decide to “opt in” by checking an unselected item or by doing nothing to a preselected item. “Opting out” is the opposite process, doing nothing to an unselected item or unchecking a preselected one. This may lead to some usability issues. It should be noted that two of the three systems observed had this feature. In the third, there were no preselected items.

The issue of the problem of long lists of order possibilities has been mentioned before. Users often complained about being able to find a desired item amid a large group. If unable to find an item to order, nothing gets ordered. In the case of preselected items, however, if an undesired item is not seen and thus not removed, it will get ordered erroneously. This interface design is intended to facilitate order entry. It certainly can save a great deal of time by allowing the user to enter any number of orders with a single stroke. This has the potential of easily submitting unintended orders unless the user knows the list well or scrutinizes the list to be certain in includes nothing undesirable. An excessively long list makes this more difficult.

It is difficult to give an example that explains this much further, but the following is submitted as an illustration of another aspect of the problem. For a hospitalized patient, the “code status” is the designation that one assigns the patient to instruct the caregiver in the appropriate course of action in the case of a life-threatening emergency. A “full code” means all possible measures are undertaken. In an order set in one system there is a heading for “code status.” There are three checkbox options listed under that heading:

- ☐ Do Not Resuscitate
- ☐ Comfort Care Measures
- ☐ Limited Cardiopulmonary Resuscita [sic]

If the user does nothing here (“opts out”), the user is a “full code.” In both the paper and computer record, if no code status is given, “full code” is, of course, the default designation. However, it can be and often is written down in an order to make certain. In this example, it seems that the “full code” option ought to be listed; the way it stands, the only way to order this is by doing nothing. One may think that decisions about crucial medical issues such as life sustaining treatment should not be made so passively. Orders that are already pre-selected should thus be carefully considered. This may be more a patient safely issue than specifically a usability problem, but it is an important consideration in interface design.

Not surprisingly many users found the preselected checklist a great advantage. This timesaving measure is an extremely attractive feature in an application that has so many other undesirable inconveniences. This arrangement may well be very appropriate for some circumstances in which the same items are ordered for nearly every patient. Certain post-operative orders may fall into this category. One user lamented that he was required to make any selections and hoped it could be more personally customized:

"De-select the ones that we don't want. It would seem to me that this should be a one click operation, but unfortunately now we'll go through this whole process and revisit each one individually."

5.8.12 Mindless Data Entry

This and the next section, "Misleading the User," share some similarity. They both are related to the intentions of the user as he acts upon the computer system. In this first section, "mindless" data entry refers to the idea the user may interact with the computer with less than the usual or needed amount of forethought. As the user gets engaged with the computer and swept up with the flow of activity, his attention to detail may wane. To achieve his goals, the user must decide to act toward that goal and form his intention. Deciding what actions will achieve those intentions takes some thought as well. If the actions are ultimately executed with less than the required amount of thought into forming the intention or specifying the action, problems may occur. These results reflect the extent to which the user interface influences this decreased attention.

There are some actions that the user must perform quite often. For example, the user must "submit" every order or part of an order in some way. A common computer convention is the user of the "enter" key to submit data that has just been typed into a text box. In one system, there are some defaults at the end of the order process that may be unneeded. For example, there are opportunities to enter additional comments about the order, but if there are none, the user clicks "enter." The users were required to click one, two or more times to get back the starting place to start writing another order. The user was often observed to “tap-tap-tap” an indiscriminate number of times. Here are examples:

“I tap several times any time I enter an order that has any comment sections to get through them all as quickly as possible.”

“No comment. Again, I see that "comments" and I click several times just to get through.”

“Skip that. I need PT and OT to see her, because she is weak after all. OK here we go with our multiple enters, just to get that going. OT. Here we go, same thing…”

It seems that pressing keys without thinking is a potential source of error. Other times, the user interface “enticed” or even forced the user to commit to some action, even if the user was not sure what to do. The presence of order sets may be implicated here. If information is requested that the user cannot supply, he sometimes enters a less than thoughtful response. This is very similar to “guessing the action specification” and the following is an example of both:

“I don't know what class she's in, let's see, how about we'll do a 2 because I don't even remember what the rules are for those but she seems like a 2.”

In another example the user was skimming items in an order set and appeared to enter items that he may never have thought of if they were not contained in the list. If the order set serves to remind the user if things that may have been forgotten, that is beneficial. It is another matter, however, if the interface encourages the user to enter items frivolously. Here are examples of the latter:

“Notify Provider… Huh, wow, they really … OK they want me to notify provider if there's ST elevation or new left bundle branch block. I think they do that anyway, but I'll check it.”

“Notify provider for complications of myocardial ischemia. Sure, that sounds reasonable. Whatever that is.”

Lastly, the user may feel persuaded to enter “something, anything” if he cannot understand what to do next. The following was observed as a user was trying to enter information about the patient’s condition and code status.

“And then we scroll down some more. And then, she's, OK, so condition, we talked about this. So I know what her code status is. I'm going to say "fair" condition. I never know what those mean really. Let's see. Ok. I don't do these ones usually. I'm going to do this one because she said she would not want to be coded but then she also said it would be OK to intubate her for a respiratory reason so I need to look at this form for a second and figure out where the heck I write that in. Um. So. This says "don't do these things."”

Well, yeah, OK. And well she's decided against these. I'm going to put it in the Comments, how about that? That sounds like a good idea…"

Here the user truly wanted to impart information that the patient has related to her, but there was a mismatch between the user's intentions and the allowable actions. The user could not enter very critical information in the form given. As this was about the patient's code status, it was obviously very important to be unambiguous. The user knew the patient's desires because of some explicit conversation. Getting this information into the record proved difficult and she appeared to place it in the only way she could figure out.

5.8.13 Misleading the User

Although similar to the previous section, this idea of "misleading the user" refers to a phenomenon in which, once the user has selected one course of action, it was difficult *cognitively* to change course. The user was led down the garden path, so to speak, and this generated usability problems. Once a user has made a cognitive commitment to follow one course of action, the interface often makes it difficult to back out of that commitment and try a different tack.

One example, which has been used before in this report, was a user who wanted to order "alcohol precautions" on a patient who had the potential for alcohol withdrawal. He attempted to find "alcohol precautions" by searching for them with the phrase "precautions." This seemed a perfectly reasonable first step. As it turns out, precautions to prevent alcohol withdrawal were not to be found there, and it took the user a great deal of time to discover that. Once he got the idea to pursue this order by looking under "precautions," he was committed cognitively and it did not occur to him to stop altogether and try something else. The precautions he was looking for were listed under a particular alcohol withdrawal protocol named CIWA (Clinical Institute Withdrawal Assessment) which he ultimately deduced. He paid for his initial misstep, however, with several minutes of wasted time. Picking the wrong initial path constrained his options.

In another example, the user wanted to order an abdominal x-ray. In the "nephrologist" order set, under an entry titled "Imaging," there were only two tests, "Abdomen Complete US" and "Abdomen without Contrast CT." His intention is to order a plain abdominal x-ray, which was not listed here and he mistakenly checked the ultrasound box. He caught his error however and unchecked the box:

"Imaging, I do want to get a plain film, so I'm going to… oh, but I don't want the ultrasound, so I'll have to go to a different screen."

As a nephrologist, he undoubtedly used this order set frequently, yet he did not initially discern or remember that only two imaging tests were available. He followed along the direction the order set was "leading him" in and he nearly ordered the wrong test.

As stated, this issue is similar to "mindless entry" in that the interface occasionally seems to be leading the user. The users often seemed to be choosing items to order that they may not have, if not listed in an order set. This may be an ideal circumstance, if the system was suggesting items recommended by evidence. But occasionally the user seemed to have a casual, perhaps careless approach to ordering:

"Laxatives? Probably laxative of choice sounds reasonable.
OK, a little slow. Antacid of choice? Why not?"

"Pulse Oximetry? Well? "Intermittent with vitals", why don't we do that?"

5.8.14 Synonyms

The significance of synonyms to usability has been referred to a number of times in these results. Users found synonyms important in specifying which action to undertake, interpreting the display, locating items on the interface, understanding the meaning of text and graphics and improving the access of long term memory. Most of the segments that are examples of this concept have all been shown in other sections. Users could not find the names of diet orders, imaging tests or order sets because they were listed by synonyms the users did not recognize. Here are two more examples. In this first, the user prefers the term "day" to "24 hours":

"Amylase, we'll do the same thing. " Early AM draw." Frequency, "Q24 hours," it would be nice if it would just say "Q Day"."

In one last example, a user wanted to order an antibiotic and searched for it by typing in the brand name "Levaquin." The system could not find this and he then typed in "levoflox [sic]" and the system found the generic name for the drug, levofloxacin. Here the system did not allow for the entry of a perfectly acceptable term for a medication (There is only a single brand name for this in the US.).

The language of medicine is an integral part of its everyday practice. Physicians in training quickly learn the "lingo" and there are commonly accepted terms, phrases, acronyms and abbreviations that every physician knows. Certainly including both the brand and generic names of medications seems reasonable as this terminology that is used extremely commonly.

As long as there is the use of text in an order entry interface, the user cannot be expected to retrieve the terms he needs or enter terms the system recognizes without an adequate number of synonymous and commonly used medical terms.

5.9 Summary of Results.

I have presented the results of observing 28 sessions with community physicians using three different CPOE systems with a think-aloud methodology. The goal of this research was to understand the usability of these clinical systems and uncover impediments that occur. By analyzing the data from different perspectives I intended to explain any usability difficulties that arose while developing a larger understanding of the phenomenon. The three different coding schemes that were employed each produced different kinds of results, but they shared many interrelated findings and seemed to complement and embellish each other. The outcome was a body of results that considered this computer use from the viewpoints of the users' cognitive processes, the human-computer interface and a humanistic consideration of the clinician users.

The characteristics of the users enrolled in the study were relatively uniform. They primarily practiced in internal medicine specialties and the majority of them were hospitalists. They were average or above in self-rated computer sophistication and experience. They were noted to approach the CPOE systems with different strategies, but a common characteristic was treating the use of the computer as primarily a clinical rather than technical process.

When looking at the data from a cognitive framework, it was noted that the Human Action Cycle model fit this domain of computer use very successfully. In this view, usability issues arose predominantly when the clinicians attempted to determine the actions the computer needed to carry out the user's intended task. The users were often unsure which steps to undertake. Also, the users commonly exhibited problems with perceiving and interpreting items on the computer display. These cognitive issues seemed directly related to interface issues in general and those that are specific to these types of systems.

When analyzing the observations from a viewpoint of Human Computer Interaction, different aspects of usability issues became apparent. Primarily, the users had considerable difficulty locating the items on the interface and they lacked some specific knowledge of the system essential to enter orders effectively. They expended excessive amounts of time and energy inputting information into the computer. These seemed to be related to the above cognitive problems that were uncovered.

Lastly, when analyzing the data with a more open approach, more insight into the phenomenon was uncovered, yet many issues were related to the findings exposed with the two previous viewpoints. Overall the users felt these systems created more work for them, they had difficulty trusting the system and they often felt resigned to a course of action other that originally intended due to constraints imposed upon them by the system. Surprisingly, the users often ignore much of what was offered to them by the system and they managed to complete their work many times only by speculating the appropriate steps to take. Issues specific to CPOE that were important to usability were automated alerts, default values, order sets, synonyms and the ability to opt in or out of an order.

In the following Discussion section, these ideas will be expanded. I will look at the appropriateness and usefulness of this cognitive model as an approach to analyzing and understanding CPOE systems. I will explore the specific usability issues that were uncovered here and examine them with the goal of creating more generalizable and actionable knowledge that may benefit future designers and implementers of these systems. Lastly, I will review the utility of employing field usability testing outside the research setting as a routine component in the implementation of clinical information systems to promote a more user-centered approach and improve the acceptance and effectiveness of these systems at the local level.

Chapter 6. Discussion

6.0 Introduction

In this chapter, I will expound on the findings presented in the Results chapter. I assert that the cognitive model of user action is a useful approach in the evaluation and understanding of CPOE systems. I will elaborate on the significance of the usability issues that were uncovered with the goal of generating ideas that may be transferable to other institutions that struggle with similar issues. And I will consider the idea of using a simplified version of this field evaluation method in non-research situations as a tool for uncovering and rectifying usability problems.

At this time, it is informative to review the research questions and evaluate how these were addressed by this study. I will briefly answer these questions and then will elaborate. The central research question was:

What cognitive resources are needed and utilized when community-based physicians use commercial CPOE in-patient systems?

The Norman Model frames this question nicely. The user requires cognitive resources that allow him to recognize and outline the actions necessary to complete his goal. He also must perceive any changes in the state of the system and interpret them in a meaningful way. This was addressed in the Results (5.6 especially) and is expanded below in section 6.1.3. The sub-questions were:

What is the relationship between difficulties utilizing cognitive resources and barriers to usability?

By analyzing the data with a cognitive perspective as well as with usability heuristics, it is apparent there are many circumstances in which cognitive struggles are created by the interface and present a barrier to further interaction with the interface. This was discussed extensively in the Results section and below, especially in 6.2.2.

Does any disparity between the user's mental model and the external representations and behavior of the computer system create barriers to successful use of that system?

A prominent point in this thesis has been the emphasis of the "gulfs of execution and evaluation" of the cognitive model and how they are central to the understanding of user-computer interface problems. Understanding this issue with respect to the Norman Theory of Action outlines the problem of the disparity between the user's mental model and the computer interface design.

Can an understanding of this disparity be exploited to modify the user's cognitive resources or the system's external representation, through training and improved system design, to lead to more successful use of these clinical computer systems?

Yes. This analysis has repeatedly and thoroughly demonstrated the types of interface problems where the disparity/gulfs occur. The Norman model outlines that the way to narrow these gulfs is to “move” the system and the user closer together. Using the information that comes to light in this evaluation may improve the design and implementation of the system as well as focusing and improving user training.

6.1 Understanding CPOE use with Norman’s Theory of Action

6.1.1 Cognitive science, computer use and clinical information systems.

Before discussing the value of the Norman model in this setting, I will briefly outline the rationale for using a cognitive model and the consideration of cognitive science in general when discussing human computer use. I will summarize the importance of cognitive science to this field, the underlying theoretical frameworks, the methods that have been developed within these frameworks and their application to medical informatics.

The ground-breaking work of Card, Moran and Newell and of Lindsay and Norman from the 1970’s and 1980’s established the model of human cognition as an information processing system (see Figure 1) and as a useful way to understand the “user-computer interaction." They looked at the Model Human Processor as three interacting subsystems consisting of the perceptual system, the motor system and the cognitive system.(46, 49) They explicitly analyzed the knowledge people need to perform work with computers. This work subsequently developed into a large body of research that confirmed and advanced the value of this model. (50)

It is relatively easy to understand computers as information processors. Considering humans as information processors as well greatly advanced the understanding of human computer interaction. This approach works effectively by considering the interaction of humans and computers in terms of two highly complex information processors effectively conducting a dialogue. When using a computer to do a task, the user cannot rely solely on his knowledge of the domain; he has to know the tool as well; but this is a tool that is another information processor and the user must consider how it behaves. The study of human computer interaction seeks to understand what it is about the factors in this interactive process that leads to productive and efficient use of computer tools. (37)

However, the Human Information Processor model has gaps. It falls short when applied to such things as non-skilled users, user fatigue, work environments, user acceptability, ease of use and the social and organizational aspects of computer use. (50) Therefore, one needs to place computer use in context. The individual nature of the user and his environment are extremely important and thus we expand our consideration beyond the processor model to the ideas of usability and "user-centeredness."

Consideration of the user is the central concern of “usability engineering” which takes into account the human information processor and user in context. (97, 98) The need for usability engineering approaches occurs because computerization dramatically changes the work environment and this makes changes in the cognitive demands that are placed upon the worker. The user has an intermediary in his work, another cognitive processor which changes his role to a manager of sorts, supervising this resource. (39) Usability engineering makes use of evaluations like think-aloud methods to understand the human-computer relationship.

The consideration of usability engineering with respect to medical information systems is a newer idea. But as we recognize that health care providers will be increasingly using computers, we must also understand that effective and productive computer use directly affects user acceptance and the larger issues of health care quality and safety. Both are related, because a tool that is easier to use will allow one to perform one’s work effectively and free of errors. (58, 99)

In the late 1990s, Patel and others argued for the consideration of a prominent place for cognitive science in medical informatics. They argued that the theory and the methods of cognitive science can provide an effective way of approaching issues of processing information, usability and user training. (100, 101) At about that time Patel, Kushniruk and others advanced this idea further by using usability engineering methods to evaluate health care information systems as a means to improve the usability of interface design. (28, 61, 62, 102) These ideas were advanced by using cognitive and usability methods such as think-aloud techniques, simulations and video recording to evaluate clinical information system interfaces (33, 63, 65, 103). Employing the cognitive evaluation of clinicians using CPOE specifically, Pelayo et al. performed a cognitive task analysis and noted that information gathering, selection and interpretation were the most critical cognitive functions needed to support the use of CPOE systems. (104)

The key to the relationship of humans and computers lies in understanding the interaction between them. Within the last few years, few would argue against the substantial importance of a human-centered and cognitive science approach in the analysis, design and evaluation of health information systems.(105) Thus, approaching human computer interaction from a cognitive perspective has a well established justification in the literature and this extends to its use in the evaluation of medical information systems.

6.1.2 Use of the Norman Theory of Human Action for the Evaluation of CPOE

Principles of cognitive science and usability engineering have thus been shown to exhibit significant utility when studying computer usage in general and this should extend to the use of clinical information systems. This study employed a cognitive theory that has broad appeal (96, 106-110), but how exactly does the Norman Action Cycle fit into the picture of cognitive science and computer use? Is it applicable to the evaluation of CPOE systems?

Norman developed his Stages of Action model "to understand the fundamental principles behind human action" that are relevant to the human-computer interface design. The basic problem he hoped to illuminate was the observation that user's psychological variables differ from a computer system's physical variables. Thus "there is a discrepancy between the person's *psychologically* expressed goal and the *physical* controls and variables of the task" [emphasis Norman's]. The central thrust of this theory is the understanding of these discrepancies, the so-called "Gulfs of Execution and Evaluation" (see Figures 3 and 4.) To further quote Norman:

> The user of the system starts off with goals expressed in psychological terms. The system, however, presents its current state in physical terms. Goals and system state differ significantly in form and content, creating the Gulfs that need to be bridged if the system can be used. The Gulfs can be bridged by starting in either direction. The designer can bridge the Gulfs by starting on the system side and moving closer to the person by constructing the input and output characteristics of the interface so as to make better matches to the psychological needs of the user. The user can bridge the Gulfs by creating plans, action sequences and interpretations that move the normal description of goals and intentions closer to the description required by the system.

So the gulfs between the system and the user are essentially the requirements for system design. The only way to bridge these gulfs is to move the user and the system closer to each other in interface design. However, changing one aspect of the action cycle can have effects on others. Altering the physical interface to more closely match the psychological intentions of the user constrains the possibilities of that interface. For example, menus may be considered as a provision of information that assists the user in intention and action specification, but they can make execution more difficult. (98) The model cannot hope to provide the answers to good interface design, but merely to explore the possibilities in the context of the user's psychological viewpoint. Like all complex devices, there are fundamental difficulties in understanding and using computers. This model's advantage is that it helps to understand the issues that are involved and expose the possible choices in interface design. It demonstrates when choices exist and shows what the tradeoffs are, as an improvement in one aspect usually leads to a deficit somewhere else. (98)

In this study which examined physicians using CPOE, this model did exactly that. First, with these data, which consisted of observations of users thinking aloud, the model seemed to fit extremely well. In the user activity observed, essentially any segment could easily be explained in terms of this model and readily mapped to a corresponding step in the model. (As stated in 5.2.3, it was not always clear where the user entered into the cycle, but this is accounted for in the model and was noted by Norman) (35). User activity also reliably followed the sequence of action steps outlines by the model. Secondly, when analyzing the data under this lens, an abundant amount of interface issues was revealed, confirming that this is a useful approach to considering the human-computer interface and uncovering usability concerns. Different interface problems were seen to relate to specific steps in the model. User problems were observed and they could be explained in terms of the Human Action cycle and known specific usability heuristics equally well and sometimes interchangeably. Lastly, there was a strong interrelationship between the results obtained using the multiple coding schemes. The findings from the Norman theoretical coding, those that looked at human-computer interaction rules and those revealed by the grounded analysis frequently overlapped. This strongly suggests verisimilitude in the assertion that this theory accurately reflects the experience of the user, at least under the circumstances of this type of observation.

Kurt Lewin said "There is nothing so practical as a good theory." (111) A theory is only useful insofar as it is explanatory and predictive. In the following section I will demonstrate that the Norman model is explanatory in that it helps us understand what is going on cognitively for the users when they interact with a computer. The predictive aspects of the theory are reflected in the tight "fit" its "action steps" have with the observed user behavior. That really is not surprising, in that this theoretical viewpoint of user-computer interaction has been demonstrated amply before, and there is no reason to suspect it would not also fit in this domain of clinical information systems.

However, there may be some additional predictive aspects as well. When certain conditions exist in the interface, the users were observed to behave in a way that might be foreseeable considering this cognitive model. While true prediction is not possible in this qualitative analysis, where the endpoints are subjectively defined, we can consider that the outcomes may be anticipated by considering the model. When certain interface issues exist, one may see associated cognitive problems. Conversely, when users exhibit certain cognitive struggles, one may anticipate that they experience problems with certain kinds of interface issues. There is a two-way relationship between cognition and the interface as described by this model. That is the thrust of Norman's theory – that the "gulfs" between the psychological (user) and the physical (system) are the heart of user problems. The way to bridge these gulfs is by changing either the user or the system.

Most importantly, one must keep in mind that this is qualitative research. Conclusions about observed behavior apply only to the users observed. Definite inferences beyond the selected participants are not possible. However, this examination attempts to understand and explain the behavior thoroughly, and like all human behavior there may be some transferability to other users that should be considered.

6.1.3 Considering the data from the perspective of the Norman model

When observing community physicians as they use commercial CPOE, how can this theory inform us about human computer interaction? The results above in Section 5.6 were presented by looking at aspects of the data segments that corresponded to each "step" of the model. For example, if a segment of data seemed to suggest an issue with "intention," it was coded as such and all similarly coded segments were considered as a group to understand how "intention" operated among these users.

Looking solely at one step of the Human Action Cycle may be problematic. It may be inappropriate to isolate any action to a single step, as cognitive behavior is a continuous fluid process and steps overlap and interleave. In a sense, all six steps can be considered to occur with any segment of observation. But if one is going to use this model at all to help understand a phenomenon, it is more helpful and informative to consider the interface with respect to each step, as one may uncover processes that are less evident when looking at behavior without attempting this degree of perspicacity. This will be demonstrated in the following sections which will elaborate how the Norman Theory of Action was useful in understanding the observations of CPOE users.

6.1.3.1 Intention

For the users observed, the prime intention is always clinical. According to the theory, however, these psychological intentions must be expressed within the constraints of the physical computer system. "Downstream" intentions, therefore, are affected by what the user perceives, interprets and evaluates about the physical state of the computer.

In this study, the interface played some part in altering and thwarting intentions. This was mostly the case when the interface left the user with few or limited options on how to progress. A source of great difficulty occurred in which conditional situations were set up and none of the conditions were available. For example, if there are "if-then" items on an interface, there must be an "else" to follow, if none of the "if" conditions are met. In other observations, the sequence or manner of order presentation altered intentions. The timing and sequence of the orders must be delivered to the user in the order he expects it according to customary clinical practice or his intentions may need to change.

6.1.3.2 Specifying the Action

After a user has a clarified his intention, he must specify which actions to carry out on the computer to achieve that intention. A great number of data segments were considered to be associated with this cognitive step. Problems with this step of cognition might rightly be defined as “the user doesn’t know what to do.” That may simply be a restatement of the obvious, that most of the trouble with computer applications occurs, in general, when users don’t know what to do. However, the results of the observations were more nuanced than that. This step in the Action Cycle was critical, as difficulty here sometimes resulted in altering or deferring the original intention.

Indeed, sometimes the users did not know what to do – at all! They were stopped in their tracks with no idea how to proceed. This is a devastating position for a user. This often occurred at the outset of a session and simple labels identifying where to start would have been helpful. Another situation that left the user “frozen,” was when he was trying to specify complex tasks, such as dynamic orders that are dependent on physiologic parameters or the results of lab tests. If the user is to ever hope to enter these types of orders electronically, considerable work needs to be applied to understanding these tasks and specifying the array of possible actions in some manner on the interface.

The users had difficulty specifying which actions to take when there were too few or too many choices. The users were often led on a “wild goose chase” if the initial actions they chose started them down the wrong path, as they could not subsequently recover. Problems with action specification caused the huge problem of difficulty locating items on the interface if it was poorly organized, terms were unclear or confusing, items were not noticeable enough or if the intended choice simply did not exist. Users were occasionally hesitant to venture into novel action specification for fear of making errors. This step was associated with many sorts of dilemmas of data entry and was often the result of cognitive overload. The presence (or absence) of well thought out synonyms, order classification schemes and order sets played a large part in affecting how a user outlined his actions.

Considering design of the interface with this cognitive step in mind seems to be crucial in creating a useful, flexible and intuitive system. In the observations in this study, this step of the cognitive process involved in order entry may have been associated with the most problems of usability. An interface condition that gave the user difficulty with action specification resulted in a range of user behavior, from guessing what steps to take, ignoring items, entering orders with not enough forethought, to simply giving up and foregoing the intended order.

6.1.3.3 Execution

Considering the interface from this step of the Human Action Cycle was less revealing than the previous one, but a few issues stood out. This step refers to the physical action applied to the computer and this amounts to issues with typing text and clicking the mouse over interface items. Anecdotally, it is often stated that the physicians' typing skills are wanting and a barrier to computer use. The literature has little to say, but older studies suggest that physicians have considerable computer anxiety related to poor typing skills. (112, 113) This problem, while still present, may be less significant in more recent studies. (114) In any event, typing may simply demand more cognitive attention than writing and should be considered in interface design. Users in this study did state that typing text was time-consuming for them.

If not typing, the user must manipulate "widgets," or elements of a graphical user interface that display an information arrangement changeable by the user, such as drop down menus or check boxes. Conditions that made the execution of these devices more difficult were evident, such as when long lists were present or numerous clicks were needed to change a display item. Observation of the users' problems with execution suggests that actions with widgets should be simple and limited to conventional and readily identifiable ones.

6.1.3.4 Perception

After the user has executed an action, the system display will change and that change must be perceived. Many of the observations in this study demonstrated situations in which the user could not perceive an item on the display, which resulted in problems using the computer. Primarily, the users had trouble when the information they needed to proceed was not easily visible to them on the display they were looking at or when they could not locate information and had to go hunting for it.

Long lists of items in order sets, drop down menus and forms were a significant problem with usability in this study. Users repeatedly complained about them. (The definition of a "long" list is subjective, but there are guidelines. Many authors suggest a list box, for example, should have at least three and no more than eight items and all should be visible without scrolling (96). This was greatly exceeded in numerous observations in this study.) In addition to the length, it appeared that items early on in the list were the most likely to be seen or at least the front of the list seemed to be where users looked for frequently used items. The items that were located on a complex display were difficult for the users to see if the organization of the display was unclear or differed from what was expected from a medical or computer system convention. Similarly users could not "see" items that were labeled with terms different from what they expected. In phrases, the first word was the one they noted first.

Users ignored items frequently. This was surprising and not entirely understood. It seemed odd when users appeared to be looking at "very obvious" items on the interface, but made no mention of them. Whether they never perceived what the item was attempting to communicate or simply did not care to comment was not clear to me. By their verbalizations, it did seem that users occasionally mistakenly dismissed pop up alerts as they incorrectly assumed what the alert message was communicating. They did not look at them thoroughly enough; it appeared they determined the item did not warrant enough attention to scrutinize. Although interesting, this phenomenon of "ignoring" was not satisfactorily explained in these observations.

6.1.3.5 Interpretation

After a user perceives some change to the display, he must interpret that information as something that is meaningful. A great number of usability issues were observed in association with user interpretation of information, second only to action specification in frequency.

Users cannot interpret language they do not understand. The need for the use of multiple synonyms to express ideas cannot be overstated, as users cannot understand terms that are not meaningful to them. It is apparent from these observations that users of these systems think in medical concepts and the language must correspond to that. There were other misinterpretations. There are technical reasons why certain behavior is favored over others by system designers (such as the avoidance of using free text) and the users did not seem to understand how this will affect patient care. Users similarly could not understand what was happening when the computer was "quiet." The user must be kept apprised of the system state at all times.

Beyond the specific language, the meaning of information is affected by the timing and manner of its delivery. It was apparent if information was delivered out of sequence of medical conventions or in an unusual context, this altered the users' interpretation of that information. Also, the users had a need to know "where" they were when following hierarchical menu paths and the presence of “breadcrumbs” or other interface navigation clues would have helped considerably. Lastly, the sheer frequency of pop-up alerts and reminders led the users to interpret them as universally unimportant.

6.1.3.6 Evaluation

After the user notices and interprets interface information, he must evaluate if his original intentions have been met. As stated in the Results, this step is tightly coupled to Perception and Interpretation. The user’s evaluation of his work is dependent on his understanding of what is going on thus far. Situations in which the user was not clear when an order was completed were most relevant to this cognitive step. Without that clarification, the user could not evaluate whether his intentions had been met and he became stuck.

6.1.3.7 Summary and recommendations

The observations in this study demonstrated that consideration of the user’s cognition is an enlightening and useful exercise in uncovering usability issues. These observations strongly suggest that, in the development or evaluation of user interfaces, allowing for the cognitive activities of the user significantly affects the usability of the computer system. Because every interaction of the user and the computer involves each step of this cognitive model, and because these study results show the cognitive processes that are tightly coupled to usability problems, it is not only worthwhile to consider the cognition of the user, but outlining the steps of the Norman model is a convenient device to employ when addressing human computer interaction. One need only quickly review the six steps of intention, action specification, execution, perception, interpretation and evaluation to get a thorough review of user cognition when addressing every addition or change to the user interface. The model is concise yet thorough. Table 21 shows a short list of usability issues that were raised when thinking about the steps of the Action Cycle. Developers often employ lists of guidelines and usability heuristics when building an interface. Consideration of these six steps is another tool to ensure a more usable interface.

Rule for CPOE interface based on cognitive evaluation	Step in the model that suggested the usability problem
For all "if then" statement, there must be an "else"	Intention
The sequence of order presentation matters	Intention, Interpretation
Tell the user where to start	Action Specification
Templates for complex, dynamic orders such as drips or sliding scales are needed	Action Specification
Avoid too many choices in lists; long lists are deadly to usability	Action Specification, Perception
Allow the user enough choices to do the work necessary	Action Specification
Avoid navigation dead ends	Action Specification
Be generous with synonyms	Action Specification
The thoughtful creation of order sets is essential	Action Specification
Cross reference order items under various headings	Action Specification
Limit the amount of text that must be entered	Execution
Use well known and simple widgets	Execution
Users see the first item on a list easily and deem it most important	Perception
Organize lists and forms according to medical conventions	Perception
Address and correct items that are frequently ignored	Perception
All language must be meaningful to a clinician	Interpretation
Let the user know what the computer is doing	Interpretation
Makes alerts meaningful and infrequent	Interpretation
The user must know when an order is complete	Evaluation

Table 21. A short list of interface usability issues uncovered by considering the Norman cognitive model.

6.2 Usability issues uncovered in this study.

This section will discuss the important findings in this research. I will describe some of the attributes of the physician users who were observed and what those characteristics might suggest more generally about community physicians who use commercial computer systems. I will outline the important usability problems that were observed in the testing session and some of the common behaviors of CPOE users that may have been a result of the usability difficulties. I will then enumerate some of the usability issues that were specific to these and possibly other CPOE systems. Lastly, I will characterize some contrasting characteristics of the three systems and will consider whether some usability issues predominated in a certain type of CPOE system.

6.2.1 Who were the CPOE users in this study?

When considering the use of commercially available CPOE systems among community-based physicians in general, it would be interesting to understand the attributes of the typical users, if there is such an entity as a “typical” user. There are no studies that answer (or have even asked) that question. In a 2006 systematic review of Health IT use in the US, of the 257 studies selected, only nine studies examined commercial clinical information systems. The number of institutions that were community-based (non-academic) is not stated but it is unlikely there are many community hospitals among the 248 institutions that have internally developed systems.(10)

With the methods employed in this study I did not obtain a sample of users that was representative of any particular group. In each institution, however, I did observe a large proportion of the (non-resident) physicians who wrote in-patient medical admission orders. Indeed, in one hospital, I observed everyone who fit that description. These physicians shared some similarities; most salient was the preponderance of hospitalists. The past decade has seen a large growth in that profession (115) that has corresponded temporally with the growth in clinical information systems. In the US, there may be parallels in the expansion in the number of these hospital-based generalists, the growth of healthcare IT and the desire for the improvement in healthcare quality and patient safety.(116)

The users were mostly members of internal medicine specialties. The complexity of the medical admission may be greater than for other disciplines. The obstetric and rehabilitation tasks observed were much more straightforward and uniform among patients so that there were fewer decisions and much of the work consisted of reviewing preselected items. In the surgical realm, many post-operative order sets are quite similar between patients and thus may be more of a “one-click” process. Order sets tend to be used more often in less complex clinical situations.(117) Therefore, the phenomena observed apply predominantly to medical specialties and may not apply to a more diverse range of physician users.

Based upon the survey results, the clinicians used computers regularly in their work lives for clinical information seeking in addition to using CPOE. They were experienced in the use of their institution’s CPOE system and were fairly sophisticated in the use of computers in general. For the most part, they were all self-taught, which may suggest an inherent interest in using computers. Indeed, the users in this study may have been the most enthusiastic and interested users available, such that their skills may be more advanced than average. In two of the three systems observed, these users were the "early adopters" and the users of the third system of "100%" use, were self-described as more willing to use the system than most. (One user in this third group cautioned me against "selection bias" when observing CPOE users, as those clinicians who cannot or will not use the system "don't come here." They go to neighboring hospitals or only admit to the hospitalist service.)

Thus, I may have observed users that were "the best and the brightest." In Diffusion of Innovations theory, the "early adopters" of any technology innovation are described in the "ideal type" as innovative, venturesome, deliberate and those that have the respect of their peers. (118) This has implications for interface design and use. Nielsen describes three dimensions on which users' experience differs: Novice to Expert User of the System, Minimal to Extensive Computer Experience and Ignorant to Knowledgeable about the Domain. (34). These users were all on one end of the spectrum – expert, experienced and certainly knowledgeable of their domain. This may affect the degree of transferability of these study results. The expert user may employ interface features that the novice does not. However, the system features observed in this study were those that were relatively basic to what is required to enter admission orders on any patient by any physician. Though the users' attributes may be less generalizable, the basic admission task is one that must be performed by many users, at least among medical specialties.

Some of the more subjective observations were interesting. It was striking how the users differed in their approach to a relatively similar list of tasks. Users adhered to the "ADC VAAN DIMSL" mnemonic inherent in the order sets, but they did not remain faithful to that sequence. Many clearly expressed a preference for performing the "routine" or "fluff" orders (those that are common to many patients) separate in time from those orders that were very specific to the current admission. The temporal sequence varied but their desire to keep the "wheat from the chaff" was common. This has implications in the design of order set interfaces.

Most conspicuous was the "clinical" orientation of the users. Despite having greater than average computer sophistication and interest and being observed under rather contrived circumstances they knew were related to the study of computer use, the participants did not maintain a "computer" orientation. (All of the participants knew that I am a clinician, and perhaps some of this orientation may have been due to that consideration.) Much of their think-aloud verbalizations did not express the cognitive issues related to the interface but were mostly concerned with the clinical task before them. In that this was a real-time observation of physicians caring for patients, ethically this may be reassuring. This also has implications for informatics. The three pilot users were all trained informaticists, and their verbalizations were much more oriented toward the interface. This may imply that users do not think like the interface designers and system implementers. This must always be considered, most especially when the informatician wonders why the users "just don't get it." The clinician user is primarily, and thankfully, a clinician.

6.2.2 The Human-Computer Interaction issues observed in this study

This study observed physician users entering the admission orders for hospital patients on 28 occasions. The goal of these observations was the identification and understanding of usability issues that arose, if any. Although I was deliberately looking for usability issues and therefore my threshold for characterizing an activity as such may have been low, it is nonetheless informative to note that 44% of the 559 data segments observed were felt to contain some usability phenomenon that was "cognitively interesting." In addition, the analysis of these data from three different perspectives discovered very similar usability problems and this provides some validation of these results. I think it can clearly be argued that the community physicians using commercial CPOE systems in this evaluation encountered more than a few issues related to poor usability.

In general terms the users had the most trouble in six major problem categories. To maximize usability, the users must at all times:

1. Know what to do next
2. See on the display what they need to see
3. Understand what they see
4. Find the item they are looking for
5. Enter data without excessive effort
6. Know how the computer system works

These will now be described in more detail below. Some specific interface issues may be pertinent in more than one of these categories.

6.2.2.1 The user must know what to do next.

At the very least, the user must know how to start. None of the participants observed in this study were novice users, yet there were examples of users having trouble initiating the order entry process. Some interface item that unambiguously tells the user where to start ("Start here→") is needed in every system. Once started, the user must have enough options available to him. A thorough inventory of all the possibilities must be considered and available to the user. Short of that, some way to account for "other" unusual items must be incorporated. If an order activity can be performed on paper, there must be a corresponding way to perform it on the computer. This definitely does not mean the interface must mimic paper, but it must allow some analogous action. This is especially true if the action is complex, such as those dynamic orders based upon patient data, or any action that is conditional upon some pre-existing state (e.g. sliding scale insulin orders). The user must be able to repeat an activity she has done in the past. This may be supported by a robust collection of synonyms and cross-referencing of terms. If there is a change to the design of the interface, the user must be given some cross-referencing information to prevent him from futilely attempting an action that cannot be carried out. The system must be stable enough that the user does not hesitate to try novel actions for fear of crashing the system or causing medical errors. Once started, the system must allow a fluid and intuitive sequence of actions that make it clear to the user where they are in that sequence and what is the next obvious step. Lastly, the user must know when he is finished entering an order.

6.2.2.2 The user must be able to see on the display what he needs to see.

Certainly all display text must be of adequate font size using easily visible, pleasing and consistent colors. Headings need to be obviously visible and distinguishable from other text. Long lists of items are to be avoided. In this study, excessively long lists of text items were a recurrent and particularly troubling usability issue. No user should have to laboriously rummage through lists of thirty to fifty items, especially when there are usually only a handful of commonly used items. Users can find something more easily in the beginning of a list, especially the first item. If the list of options is numerous, the judicious use of hierarchical submenus is easier on the user, with a list of commonly used items displayed at first and containing links to submenus of more esoteric items. Table 22 shows an example of commonly used medication dosage frequencies that would be infinitely more usable than one drop-down menu that was observed, which listed every possible time frequency of drug administration and went on for several dozen items.

If a long list is necessary, such as in an order set, it must be organized with headings and adequate white space. Users see what they expect to see, and there are many reasons users develop these expectations, some are related to computer conventions, some relate to graphical arrangement preferences due to inherent cognition and some relate to the domain. Users need items to be named with the terms that they expect them to be named; otherwise they cannot “see” them. Lastly, to avoid a user from ignoring something, he first has to see it. This means an alert or warning must be visible and clearly identified as to its contents.

ONCE PER DAY..............................	(QD)
TWICE PER DAY...........................	(BID)
THREE TIMES PER DAY................	(TID)
FOUR TIMES PER DAY..................	(QID)
FOUR TIMES PER DAY WITH MEALS & AT BED	(QID, AC & HS)
EVERY 24 HOURS	(Q 24H)
EVERY 12 HOURS.........................	(Q 12H)
EVERY 8 HOURS	(Q8H)
EVERY 6 HOURS	(Q6H)
OTHER (link to a submenu)	

Table 22. Example of a reasonable list of medication frequencies

6.2.2.3 The user must understand what he sees.

All text and graphics may have different semantic significances to different computer users and there are many reasons for this. This difference in a user's background, experience and knowledge may affect the meaning of words. In the users observed here and more generally in the medical domain, one may infer somewhat more uniformity of experience among the users. There often is, however, a great deal of difference in the meaning of interface items between the medical user and the non-clinical interface designer. The order, organization and most especially the terms of an interface must be designed with respect to clinical orientation. The grouping and the sequence of interface items and their labels must be vetted with clinicians. There are "universally" understood clinical terms that every doctor has learned since early training and some local conventions as well and these must be contained in the interface.

In addition to the naming of items, the timing and manner of information delivery are important to the meaning of interface items. Again, user expectation is important. Physician users are accustomed to a certain sequence of orders and expect some orders to be associated with others. As was observed in this study, the order in which medications were listed (in terms of therapeutic potency, for example) affects the decisions the clinician made. One considers the next items to order often in reference to what has just been ordered. When attempting to select an item from a list, users became confused if sequentially arranged items were not ontologically comparable (for every "left," should be a "right" not an "other.") It is imperative to understand these conventions to make the interface more comprehensible.

The use of an inadequate number of synonyms greatly affects the user's comprehension and was a repeated source of usability problems in this study. All clinicians have a preconceived idea of what a medical item is called but not all users have the same preference for their favorite term. Although there are likely some items that all users name identically there are a great number of different preferences for synonymous terms. The designer and implementer must attempt to collect the list of preferred terms and make all of them available as soon as possible. Users first look for an item by the term to which they are most accustomed, and the lack of that preferred word results in a lot of hunting, as discussed next.

6.2.2.4 The user must find what he is looking for.

As was discussed in the Results section (5.7.1), "writing" orders on a paper order sheet is in many respects quite different from "entering" orders with a computer interface. In the former circumstance, the user formulates an intention and expresses that intention in written language. Granted there may be certain conventions in that language, but the task does not cognitively differ much from standard written language. The process is significantly different when entering orders on a computer. Most often, the user must "find" the items he wants to enter rather than write them. The task is changed with a computer from expressing the intended actions in words to choosing those items from the interface. Most often the user must pick an item from a list, but he has to find it first. Even if the process is initiated by searching for a term entered in a text box, the system cannot locate a term that is not present in its controlled vocabulary and the user must be certain to "find" the right words to submit. This was a huge usability problem in this study and users expended a great deal of effort pursuing their desired item. There are many explanations for this.

The organization of interface items played a large role in the difficulty or ease of locating them. Psychological principles of interface design apply here: users find things where they expect to find them; it is easier to perceive a structured layout; and recognition is easier than recall. (96) The understanding of a clinical orientation is important here as well. The order of terms in a list, the grouping of terms and consistent organization allow the user to more easily find items. It is difficult to pick items out of long lists and this applies to selecting desired items and deselecting an undesired item from among a long list of preselected orders. Terms must have meaning to the user or they cannot be found; an adequate variety of synonyms again are crucial. Words should have consistent meaning, however. It is desirable to have more than one word for a single concept, but there cannot be more than one meaning for a single word. Also, to find an item he is working with, the user should have the area of the interface in use available to him without having to scroll to it.

Users truly do consider their interaction with the interface as a process of moving through a physical space; it is a location, a place. They “go there” and “back out of here.” Therefore, steps need to be taken to prevent the user from “getting lost.” A hierarchical arrangement of orders and menus is commonly employed and the users in this study negotiated them easily. However, the more levels of menus encountered, the more easily the user became confused. A system of “breadcrumbs” will greatly help the user to know “where he is” at all times, so that he can maneuver easily from one location to another, without expending any cognitive resources trying to keep track of where he is. Once the user is oriented and has found the desired item, he must execute some action of order entry, discussed next.

6.2.2.5 The user should be able to easily enter data.

The central task of CPOE is, of course, “order entry.” In the systems observed, the actions that the user must execute are either typing text from a keyboard or clicking on a GUI item with the mouse, as is the case with essentially all CPOE systems. The users observed often related problems with text entry. As noted in 6.1.3.3, physician typing skills may be comparable to other computer users, but does typing, in general, require more cognitive resources than handwriting (hence worsening usability)? Skilled transcriptionists enter data much faster than writing with no increase in cognitive load, but they have little comprehension of the text entered and this does not relate to typing novel ideas, such as occurred with these users.(119) Brown noted that even "two finger typists" enter data faster than handwriting, but this was also a study of transcription. (120) There is no information comparing typing and handwriting in tasks comparable to order entry, but it is reasonable to assume that it is at least as fast for those with "two-finger" competency or greater and consumes no more cognitive energy. (Newer technology like handwriting recognition may have advantages, however (121).) Nonetheless, the users in this study perceived problems with typing. It would seem that minimizing the amount of text that is required to be entered would be desirable, avoiding the need for extensive text entry. This would be achieved in part by having a thoughtful availability of likely clinical options (clicking on a term in an interface is faster than typing) to avoid resorting to typing in the "Comments" section. Robust searching functionality and auto-complete would be helpful in this regard, as well. At the very least, if text must be entered, a capable text editor is needed, with standard function such as backspace or select and delete; this was lacking in some of the systems observed.

The easiest way to enter data is to do little or nothing. In this regard we find the value of short-cuts such as order sets, pre-selected orders and default values. Default values were observed, paradoxically, to be a source of increased effort of data entry. Many times preselected default values were mostly undesired and the users spent a great deal of energy deselecting these "time-savers." Some preselected items triggered alert warnings by default. A default value that causes unnecessary work for the user every time should obviously be avoided. Conversely, there was often a lack of default values for data that must be entered at every session. There are some values that, if entered by the system by default, would reduce the workload on the user, such as the current date or orders that are always ordered together.

The appearance of unnecessary alerts increased the user's work load and interfered with the primary task of order entry. Decision support guidance is one of the values of CPOE, but unnecessary admonishments are never helpful. Alerts were essentially universally dismissed in this study, suggesting an extreme need for reworking those conditional warnings. There were also some particular system designs that appeared to increase the work of data entry. Most prominent were arrangements that insisted the user duplicate his efforts by ordering and then confirming some orders a second time. No matter how usable the design, the user needs certain knowledge to interact with a computer system and not all of it can be naturally intuitive; some knowledge must be acquired and that will be addressed in the next section.

6.2.2.6 The user must understand how the computer works.

It may be unrealistic to propose that systems that enable a task as complex as CPOE ought to be usable without training. In the foreseeable future it is unlikely there will be "walk-up-and-use" CPOE systems. Short of that, the user needs training. The observations in this study suggested that the users would have benefited from having reinforced knowledge of how the system works. There is knowledge that is specific to the individual system and there is a more general knowledge of computers that is important to know. Users may benefit from a rudimentary understanding of computer science, such as understanding how a computer interprets data. For example many users did not seem to understand how free text differed from structured data and appeared to feel they were interchangeable. They did not seem to appreciate the consequences of entering free text to the integrity of the data or the effect it has on other clinicians. Some users did not know what was happening during pauses when the computer was quiet, when an order process was finished or why certain entries did not retrieve any results. Perhaps some basic understanding of computers would narrow the gulfs of execution and evaluation and make the system more usable.

In addition to general computer knowledge, the user should have command of basic information specific to the system in use. The user should certainly know how to initiate the process. There are conventions to order writing that should be reinforced. For example in the "order sentence" of prescribing a medication all orders might contain a drug name, a dose with a number and a unit, a frequency, duration and priority. If the user did not know the label of each component, he became confused as to what data belonged in a given field. The users observed were the most experienced and sophisticated computer users in their respective institutions, yet they often seemed to lack some basic information that would improve the usability of their interaction with the system. Implementers should note where training needs to be emphasized and subsequently reinforced. Measures that target training to trouble spots would improve the efficiency of the efforts. Training is, of course, not a substitute for a naturally intuitive interface.

Table 23 outlines a list of CPOE interface imperatives gleaned from the observation of usability difficulties in the users in this study. The table contains no accompanying solutions to these requirements, but those have been suggested throughout the text.

1	The user should know where to start.
2	Actions previously performed on paper should have analogous but not identical computer functions.
3	The user needs enough options to do her work.
4	The user should be able to order something that she has done before.
5	The user should not have to fear trying something novel.
6	Step by step flow should be obvious.
7	The user should know when she is finished.
8	Text and images should be large and visible.
9	Lists should have headings and white space.
10	Lists should be short.
11	Frequently used items should be at the beginning of a list, submenus for less frequently used items.
12	Items on forms and lists should be grouped and organized clinically.
13	Warnings should be clear as to what they are warning.
14	Items should be labeled with clinical language, the universal and local slang must be included.
15	The order in which items are listed should correspond to the user sequence of work.
16	Items in a list should have ontological consistency.
17	Inventory the synonyms of the users and they should be employed generously.
18	The meaning of labels and actions should be consistent.
19	Avoid the need to scroll to the active area of the interface.
20	Acknowledge the metaphor of location and motion when navigating the interface and give the user clues to avoid getting lost.
21	Avoid the need for long text entry.
22	Text editing function is needed.
23	Use defaults when an actions and information are entered frequently or always.
24	Avoid default for actions done rarely.
25	Avoid the need for duplicate effort on the same order.
26	Avoid frequently dismissed alerts.
27	The user would benefit from basic knowledge of computers.
28	Inventory the knowledge gaps in the user and retrain, most especially the basics.

Table 23. CPOE interface imperatives gleaned from the observation of usability difficulties in the users in this study.

6.2.3 Observations of the behavior of CPOE users.

The previous section discussed interface properties that were observed to create usability problems for the users. The data in this study were analyzed from different perspectives and as a result, in addition to observing problems with the interface, problematic behaviors of the users were also observed. These interesting and perhaps surprising behaviors were manifestations of and secondary to the effects a poorly designed interface had upon the user. It may be definitional to state that poor usability is troublesome for any user and it would be easy to claim it causes him dismay. However, the following are more concerning behaviors and responses that are consequences of struggling with an unwieldy computer. These actions are concerning, especially because the task entails risks to the safety of a patient.

6.2.3.1 Mistrust

The user needs to feel sure that what he believes to be happening with the computer system is actually happening. He needs assurance that his intended actions will be reliably carried out. There were many instances in the observation of the CPOE users in which they did not trust the information they received from the system or the certainty of the outcome of entering information. The users mistrusted the accuracy of some information, most often alerts, which they considered overzealous or irrelevant. Allergy warnings about drugs in the same class were by and large felt to be inaccurate. Users mentioned their leeriness about certain functionality that has worked erratically in the past or has caused the user to lose work. At times, the users “hoped” that an order would go through and reported that outcomes of certain order entry actions occurred inconsistently in the past. The users complained they did not receive enough confirmation that the order was entered satisfactorily and often added little extras, like miscellaneous text comments and double checks, to hedge their bets and help assure an order would be carried out. At times, the users expressed amazement when certain operations, even simple ones, were carried out successfully or when untoward effects did not occur.

One can never know if this mistrust is “warranted” or this is a manifestation of a clinician’s naturally compulsive behavior. But it is certain that these users have had some disappointing experiences in the past that justifies this skepticism. To improve the users trust, many steps may be taken. Certainly, making the technical fixes to avoid crashes and be certain all functions are working properly is minimal. Turning off the spurious alerts will prevent the distrust that comes from hearing the computer repeatedly crying wolf. Trust needs to be earned, and repeated engagement with a robust and reliable system will engender confidence in the system.

6.2.3.2 Resignation

The user needs to be able to carry out his intention to its completion. One of the more surprising observations was the number of times a user changed or abandoned his intention after battling an antagonistic interface. One should be certain that changes to a physician’s order entry intentions would arise exclusively for clinical reasons. In this study the users settled for entering an order that was not completely satisfactory when, after a great deal of effort, they could not accomplish their original intention. At times, they gave up and simply did not order anything. Users took the path of least resistance if the task at hand was overly difficult to finish or simply impossible to complete. This often resulted in entering free text somewhere which foists the responsibility of the order completion upon someone else. Conditions that foster resignation are, of course, the genesis of workarounds, those usually unsatisfactory and often brittle bypass solutions. Short of forgoing the order altogether, the workaround of choice seemed to be the use of free text comments in a “miscellaneous” section.

It should be noted that the users did not give up easily. They often tried very hard to “do the right thing” but conceded defeat after multiple valiant attempts. Users avoided actions that are too time consuming, especially when their successfully completion begins to appear less likely. They certainly avoided actions that have caused them pain in the past. These generally enthusiastic users naturally gave up when things became too difficult. One would surmise that less motivated users would have less patience. In one session the user was observed to quit trying to enter an order and he claimed he was going to enter the order on paper instead. Users will quit and use paper if they feel they must, perhaps permanently if the struggles are recurrent.

6.2.3.3 Guessing

Cognitively, the user forms an intention and outlines the action he must undertake to carry out that intention. The specification of these actions is done purposefully. Based upon his knowledge of the system he considers his psychological goals and tries to achieve them with physical manipulations of the system. If the user cannot determine how to do that, and he does not give up, he might be given to hazarding a guess. In that the activity under consideration here is a physician entering medical orders, a hazard is indeed potentially the result of that conjecture.

Like resignation, guessing an action specification might be a user's recourse when he does not know what else to do. And the users did not resort to taking stabs without some initial effort. The users guessed when they became "lost" in the interface, when they could not decide which option to pick (either because none was satisfactory or there were too many to consider) or often when the system required them to enter arcane "policy" orders that they either did not understand or would have no way of knowing. Occasionally the users chose to guess rather than quit. The notion of doing "something" was deemed preferable to doing nothing.

6.2.3.4 Ignoring the Interface

Not everything on a computer display is relevant to every user at all times. However, if some interface item is important to completion of a task, hopefully the user will not ignore it. As discussed in sections 5.6.4.3 and 5.8.6, the observer can never know for sure when a user is ignoring an item on the display. If a user appears to ignore an item, one of the following is occurring. The user may:

1. Perceive the item, adequately evaluate it and decide to dismiss it.
2. Perceive the item and cursorily evaluate it as something (or similar to something) that is irrelevant and dismiss it.
3. Perceive the item, but barely consider the item and dismiss it without any evaluation.
4. Not perceive the item.

I would suggest that #3 is definitely "ignoring" and #2 may be, as both are cases in which the user dismisses something without giving it enough attention to thoroughly evaluate it.

Even given that definition, the observer never knows for certain when a user perceives something if he does not verbalize perceiving it. However, if a user must perform an action to dismiss the item, such as clicking a button, and does that so quickly that he is unlikely to have evaluated it, one might infer he is ignoring it.

There are some situations where it makes sense to ignore something that is irrelevant to the task at hand. To the extent that users may be ignoring items of importance, a problem of usability exists. As stated before, the observation of a user ignoring an interface item in the setting of a think-aloud session where he has been instructed to be more mindful than usual, suggests how automatically some items are ignored. Items ignored were mostly decision support alerts.

In other studies of CPOE users in general, ignoring alerts and reminders is common. Users routinely dismiss alerts for drug allergies or interactions from 50-95% of the time. The reason given is mostly that the alerts were incorrect or irrelevant, and fortunately this behavior is not associated with adverse events, suggesting it is appropriate. (84, 122-125) In this literature, however, the "ignoring" implies the user actually acknowledges the alert and chooses mindfully not to heed it. The phenomenon I am discussing is an observation of a lack of mindfulness. This inattention seemed to be related solely to the alerts and reminders. This would suggest that the users may be so accustomed to alerts having little or no significance, that they become apathetic to alerts in general. The abundance of unnecessary alerts should, of course, be addressed. However, the creation of apathy toward decision support guidance is equally concerning and should also be addressed. In other words, if an interface item need not be heeded, it should not be there. If it warrants attention, it should be designed to get the user's attention.

6.2.3.5 Mindlessness

Unlike guessing, in which the user knows what he is doing, but is unsure, mindlessness suggests the user is not paying enough attention. The previous issue that discussed ignoring also includes the concept of mindlessness. There have been many studies in the psychology literature about "mindless" information processing. Mindlessness is defined as the rigid use of information while not being aware of any novel aspects. On the other hand, mindfulness is marked by making active distinctions, creativity and attention to details. It is the ability to create new categories in perception and interpretation. When information is presented conditionally (this *could be* an X) rather than unconditionally (this *is* an X), one tends to think more creatively and mindfully. One is also more mindful with unfamiliar over familiar information. Langer noted that "Mindlessness is based on the past, whereas mindfulness is based on the present."(126)

In this study, mindless behavior took many forms. Users repeatedly tapped the "enter" key an indiscriminate number of times to finish an order. Alerts and text details were ignored. Users uncritically accepted some information presented to them. As mentioned in the Results section, when reminded to verbalize more, one user claimed she was not doing anything about which she might verbalize, the contradictory screen capture data notwithstanding.

Mindless behavior becomes more likely when tasks become more routine. Automatic behavior lends itself to mindless behavior. Ironically, creating information systems that are easier to use and tailored to the individual may promote mindlessness; this may be a tradeoff for efficiency and occasionally at odds with a usable interface. (127) Although it may be difficult to prevent, it should be minimized. Making interface items conditional and novel may promote creative and mindful user behavior. “Just in time mindfulness” may be a way to promote this. Providing contextual explanations and letting the user know why things are occurring would promote more mindful user behavior.

6.2.3.6 Being misled

If a user is not being mindful enough, he is liable to be misled. Like the previously discussed behavior, rigid behavior based on previous exposure to information may discourage creative thinking and promote users to make premature conclusions about the information they are interpreting. Users in this study were observed to make cognitive commitments based on interface representations that led them down pathways that would not allow them to achieve their intentions. They became stuck down “blind alleys” and could not recognize they were never going to reach their goal the way they were pursuing it, and hence they did not stop and change their strategy to a more effective one.

There is a concept in the psycholinguistics literature known as a “garden path” sentence or “garden path” phenomenon. This describes the action in which listeners or readers parse sentence structure based only upon partially revealed information. They become “primed” by their initial linguistic information and thus have difficulty comprehending the overall meaning of language even after additional information is revealed. For example, first read only the first five words of the sentence, "While Anna dressed the baby spit up on the bed." Listeners often "recover" after reading the whole sentence, however. (128)

This study observed users who were also "led down the garden path" but this was not related to parsing language. Users pursued an avenue while attempting to find an item in the interface. The (garden) paths in this interface consist of progressively arranged hierarchical menus. If the user traversed the hierarchy and found himself in a level that simply did not contain the item he was searching for, the users often did not think to stop and "back up" but kept futilely searching for an order task that was not located where they were looking. The users appeared to commit cognitively to one path and rigidly remained with their originally strategy even though it was ineffective. Perhaps it did not occur to the users that other alternate strategies may have existed. Somewhat like mindlessness, they rigidly used information and were not aware of novel strategies.

This phenomenon could be avoided by having multiple ways to perform similar tasks. Keeping the user aware of his location in the interface may allow him to be less likely to rigidly commit to one path and to regroup and try another strategy. Generous collections of synonyms that classify task as well as cross-referencing may minimize this problem as well.

Another way users may possibly have been misled is by the temptation of numerous choices in an order set (a "kid in a candy store" phenomenon.) Although this was rare, I surmised users may have ordered an item predominantly because it was available and easily ordered. Certainly the point of the order set is to remind the use of items that may be forgotten. The verbalizations I noted, however, suggested the more spontaneous and perhaps capricious behavior that the presence of pre-existing items may promote.

6.2.3.7 Tolerance of bad interface design

While the previous behaviors listed in this section might be considered maladaptive, the last one is more of a coping mechanism. Every user of every kind of computer system learns to live with less than wonderful interface design. Users complained about many usability issues through the think-aloud process and this was the basis of many of the results presented. Remarkably, however, they often did not mention problems. Technical difficulties, inscrutable alert messages and clumsy data entry mechanisms were occasionally treated in a matter of fact, perhaps fatalistic way. There were references made (on and off the record) about frequent requests to fix problems that have not been completed. Users implied they stopped asking about some problems when no changes were forthcoming. The CPOE interface is exceedingly complex and the resources needed to keep up with maintenance are enormous. Also, some problems may have been irremediable due to software limitations or undiscoverable bugs.

Untoward User Behavior	Possible avenues for resolution
MISTRUST	• Allow the ability to drill down on information in progressive detail to check veracity and rationale • Unobtrusive confirmation of actions • Maintain vigilance of system problems and repair
RESIGNATION	• create several ways to perform tasks • context sensitive, unobtrusive and progressively detailed help • Monitor patterns of usage or user testing to find trouble spots.
GUESSING	• unambiguous labels and directions • same as resignation
IGNORING	• meaningful, accurate and prioritized alerts • label alerts clearly, use verbs for choices of user action
MINDLESSNESS	• run reports to monitor missing data • feedback to users to see benefit of every action • allow the user to skip unneeded fields
BEING MISLED	• multiple ways to perform same task • allow error recovery • robust collection of synonyms and cross referencing and classification • keep user apprised of his location in the interface
TOLERANCE FOR PROBLEMS	• allow easy means to report problems • address them promptly • communication with users about problems, fixes and changes

Table 24. Untoward user behaviors as a result of poorly usable interfaces and possible solutions.

6.2.4 Interface issues specific to CPOE systems.

In the previous two sections, I discussed human-computer interface issues that were shown to create problems for the users in this study and some user behaviors that result from the usability problems. This section will discuss some interface issues that are very specific to computerized order entry. Based on the observations of usability in this study, I feel these five issues are ones that must be thoroughly considered and addressed by designers and implementers of CPOE systems. They are:

1. synonyms
2. alerts
3. defaults
4. preselected orders
5. order sets

The last three are similar. They are related to the tradeoffs one must face when deciding whether to have data already entered into the system or expect the user to enter information de novo. They explore the comparison of information recognition versus recall and ease-of-use versus flexibility.

6.2.4.1 Synonyms

One of the primary challenges of medical informatics is the representation of clinical information that is usable to both the clinician and the computer system. Computers must recognize concepts as unique and unambiguous, and the approach that has been deemed most effective is to encode these concepts according to standardized terms from a preexisting controlled vocabulary.(129) Clinicians, however, must express their ideas in the vocabulary they are used to and to the extent that it may not match the controlled vocabularies, usability issues arise. Thus, part of the usability of clinical systems depends on the underlying vocabulary and how it is displayed on the interface. Underlying controlled vocabularies must therefore support synonymy, the presence of multiple terms to express single concepts. (130)

Spackman recognized multiple categories of controlled vocabularies and described the "interface terminology" that must include lexical variants and synonyms. (131) The interface terminology is defined as one which is designed to support human-computer interaction. Clinical interface terminologies must therefore be designed to represent the variety of colloquial phrases used in medical practice and require a rich synonymy to allow the nuance with which users can express themselves. Rich synonymy increases usability by improving expressivity and accuracy. (130)

In this study, there were many observations of usability problems directly related to a paucity of synonyms and these have been mentioned repeatedly throughout this paper. Without the presence of synonymous terms for the same concept, the user has difficulty when search terms were not recognized by the system, when they could not "see" the concept they were looking for when listed by another name and they did not know where to look for an order when listed with a name they were not readily accustomed to using. This applied to medications, lab tests, diets, nursing orders and even the names of units in the hospital. Users can remember a term that has a higher level of activation in their memory, the term that each user most commonly associates with an underlying concept.

One of the ironies and challenges of computer use is the disparity in the needs of linguistic representations between the user and the computer. (132) The computer's need for unique and unambiguous terms is sharply contrasted with the human practice of expressing ourselves in complex and nuanced ways that vary with context. Therefore it is essential that interface designers understand how important synonyms are to users.

Efforts should be undertaken to gather an inventory of synonyms employed by the system users and map them to single concepts. Perhaps these could be done automatically by collecting terms that users spontaneously enter or provide an easy way in the interface for users to submit alternative terms that occur to them as missing but desirable. At the very least, all interfaces should have some basic synonyms like the brand and generic names of medications or the fundamental argot and idioms that every physician has used since the first year of medical school.

6.2.4.2 Alerts and Reminders

Automatic alerts, unsolicited, intrusive and context-related messages to clinicians, are fundamental types of electronic clinical decision support interventions. They are known to be of benefit to patient care by promoting improved adherence to guidelines, decreased costs and decreased errors. (10, 133-135) Because of these potential benefits, alerting systems have been widely implemented, but invariably users feel they fire excessively. Because they apply the same threshold to all patients in all situations, this results in poor specificity and physicians develop what is commonly known as “alert fatigue.” (84, 136) This phenomenon has been studied fairly extensively with relatively uniform findings: physicians override the alerts at fairly high rates (approaching 90%). A review paper by van der Sijs noted the most common reasons clinicians list for dismissing the alerts are when the alert is not serious, not relevant, or if they are presented repeatedly. (137) Taylor found the most common type of alert that is overridden is the toxicity alert – allergic reaction potential, medication intolerance and therapeutic duplication. Alerts that are the most heeded are those that alert the user to the patient’s age or health condition or notice of medication interactions. (124) When clinicians were asked the reason for overriding alerts, Hseih found physicians stating that they were aware of the problem and will monitor it, the patient didn't have the allergy indicated or the patient was already taking the medication to which he was purportedly allergic and was doing fine. (138) In one study, 22% of clinicians admitted to overriding alerts without even checking the reason for the alert. (137)

Other studies have found that even though alerts are overridden at very high rates, clinicians still felt the concept of automatic alerts was a good idea. They would be more acceptable to physicians if they were interrupted only by alerts of critical or high severity or if they were provided with alternatives or information rather than a simple negative warning. (136, 139) Based upon ideas physicians give for overriding alerts van der Sijs lists 23 recommendations for improving acceptance and Hseih outlined sevens ways to reduce the dismissal of allergy alerts. (137, 138)

The observations in this study are in alignment with the literature on this topic. As discussed in Section 5.8.4, the override rates of these users approached 100%. They implied from their verbalizations that they felt the alerts were related to more minor concerns or were simply irrelevant. Repeat alerts were particularly unpleasant to the user, especially those for duplicate drugs when the duplication was the same medication but merely different routes of administration. The excessively sensitive threshold for firing these alerts led to user irritation but it seemed to also result in user apathy. It is human nature to pay less attention to alarms which are predominantly false. It is also appropriate behavior. If a user incorrectly dismisses the rare alert that actually is significant, this cannot be attributed to user negligence. The system is culpable because it interrupts the user with meaningless warnings nine out of every ten times.

Elimination of unnecessary and excessive alerts is important to the usability of CPOE systems. As mentioned above, there are recommendations in the literature. As a general rule of thumb, most authors and implementers recommend limiting alerts to those that are the most significant to patient safety. To understand which alerts clinicians feel are insignificant, one must repeatedly monitor and evaluate alert firing and override rates. (83) Other recommendations to consider would be the elimination of duplicate drug alerts that fire for different routes of administration or similar drug classes. Limit alert firing rates to only once per drug per user session or user encounter. Allow clinicians with the appropriate authority an easy ability to alter allergy lists. Lastly, further cognitive research needs to be done on the optimal interface approach to alerting users that is minimally interruptive and gives the user adequate and prioritized information from which to make a decision.

6.2.4.3 Default values

Default values are integral to the CPOE interface. A fundamental psychological principle underlying interface usability is that it is easier to recognize something that is there on the display than to think of it and enter it oneself. And it is simply faster to click on the display than to type in text. The only way decision support systems can ever impart evidenced based guidelines is to supply default values as suggestions. Default values are a large proportion of the essential knowledge content of CPOE and CDS systems. (91)

However, this study also found that default values to be a source of usability problems. (See Section 5.8.8.) The only useful default value is the one that the user desires to enter. If it is not present, the user must start looking for it without success, especially if he assumes that because the value is commonly used or is essential, it is very likely present in the system. Certainly not every item conceived by the user can be present in the system. However, it is a reasonable expectation for items commonly used in the domain and the particular context to be available. Common lab tests, procedures and drugs and doses are expected to be present in the system. Certain "corollary" items that must accompany an associated order in all circumstances should be available as a default. For example, if the patient receives "standard nursing care" every time he is "admitted to ICU" that order should be entered by the system by default. If a test is ordered "tomorrow morning," the user should not have to enter that date and time somewhere else (or change it from conflicting ones.)

Like alert messages, the presence of inappropriate default values detracted from any inherent value. The users in this study spent a great deal of effort removing preselected default values they did not want. Some users explicitly remarked that they were required to deselected a value "every" time. It seems clear that careful attention to including commonly used default values is important to usability. The more commonly an order is used, the more accessible it should be. The designer should also keep in mind items that are associated with each other in the mind of the user. This information can be obtained by studying user activity reports or evaluations such as this. As discussed, many default values take the form of “preselected” items, and this warrants special discussion as follows.

6.2.4.4 Preselected orders

Default values are a broad concept and may comprise all knowledge content already displayed on the computer screen. Just because a value is present on the display by default does not necessarily mean it will be entered as an order. In some circumstances however, orders are already preselected from a list, usually a list of check box widgets, and will be ordered whenever the user completes his session with the click of a “submit” or “enter order” button. That last step usually occurs after a list of orders has been preliminarily entered and is a confirmation step. It is common to all orders, but the user need not make any other effort other than this final action to enter the preselected items.

Users find this very convenient and desirable at times. Certain circumstances, such as routine procedures usually performed on a population of very similar patients, allow a collection of identical orders. The template is the same for every patient, or nearly so. (One example would be post-partum orders after a routine delivery of a healthy mother with no medical problems. Indeed, at one institution I met with an obstetrician in response to my recruiting letter. He was very interested in the project, but felt that his efforts would be uninteresting in that his orders often consisted of a “one-click” process that takes him only a few seconds.)

However, not all preselected orders may be desired. Users were frequently observed to spend a nontrivial amount of time reviewing a list of items in order to de-select some that were not wanted. This seems to defeat the purpose of any convenience pre-selecting orders may bring about. In addition, the users needed to carefully pore over the preselected orders by searching up and down the list. This raises the concern that one of the predestined but undesired orders could be missed. This would permit a procedure to be carried out or a drug to be given that is at best, not wanted and wasteful or at worst, harmful to the patient.

It seems that the designer and implementer should cautiously scrutinize the choices for preselected orders. These may be limited to the unique circumstances when the procedure and the patient attributes are uniform. This may be difficult to achieve for complicated patient admissions. Generally speaking, complicated medical admissions may not qualify. At the most, perhaps this could be limited to orders with the least potential for harm, such as vital signs or activity, but even this may not be safe in very ill patients. One could consider no preselected items for medications or procedures. Certainly, to improve usability and assure patient safety, the lists of pre-selected orders should not be too long, similar to cognitive concerns about locating items in long lists in the interface mentioned elsewhere in this paper (5.6.4.1; 5.7.1.2; 5.7.3; 6.1.3.2; 6.4.3.4.)

6.2.4.5 Order sets

If irksome alerts, burdensome defaults and an insufficient supply of synonyms are notorious banes of usability, the order set is generally just the opposite. The order set, or a pre-defined grouping of orders, is another manifestation of pre-configured medical knowledge from which the user may choose from the interface display. This study was designed to observe users verbalize their thoughts about computer use, but I primarily focused on understanding usability issues (problems) in my analysis. Topics that were not as troublesome were generally mentioned less and coded less.

The users did mention order sets (one system's users additionally referred to them as "care sets") frequently. Searching the transcriptions for the phrases, "order set," "care set" and "set," 18 of 28 users mentioned them and none in a negative light. Some users had complimentary things to say:

"It gets me a care set up so it makes it easier."

"Now I really like to use the hospitalist order set for this. I'm used to it and it contains pretty much everything I need."

"I've arranged for some of my orders to be in an order set. We created those after much pain and turmoil."

"…which is the pre-admission order set and it's already been… put in the computer so that's nice."

"I can pull up different sets at lightning speed."

Users generally like order sets because they are fast, convenient and free of errors. They are often ergonomically organized to match the user's customary workflow (as when they take the format of ADC VAAN DIMSL or "admit, diagnosis, condition, vital signs, allergies, activity, nursing, diet, IVs, meds, studies, labs".) and they readily permit the user to be aware of and follow standards and guidelines. They follow the interface psychological principle that describes recognition as easier than recall.

Order sets' configuration is not uniform however. In this study they varied within and between systems. Payne defined three types of "order configuration entities" – order dialogs, preconfigured order sets and order sets. The *order dialog* is the least configured of the three. In this, the user enters a single order by completing appropriate fields from pick lists or by entering text. This method is the most flexible, but the most time consuming. The *preconfigured (quick) order sets* (also known as an order sentence) are total orders that can be selected with a mouse click for submission or further editing. Here, some or all of the order dialog box fields have default values and the user only reviews, or minimally edits, and signs the quick order. The term *order set* encompasses a larger group of orders linked together that can be invoked to generate many orders quickly. When an order set is launched, all of its components are selected and prepared for submission. This is the fastest way to enter orders. (92) All three of these shortcuts can be organized into larger order grouping menus that serve a specific purpose. Payne called these *order menus*, but these are what are commonly referred to as order sets. The purpose of these collections may for a specific diagnosis or symptom, procedure or hospital unit. All of these types of pre-configured orders were observed, but some systems employed some styles more than others. This description of the spectrum of flexibility and convenience of pre-configured orders within the larger groupings was noted in Section 5.8.9.1.

There are several significant points to consider about order sets. They are convenient and fast. Users generally like them, they are used frequently and they are a major attractive feature of CPOE. Order sets afford the users the ability to follow guidelines and practice more standardized medicine. (92-94) They may have some drawbacks, however. The more pre-configured an order set is, the less flexible it is. The “order dialogs” allow much more flexibility but they are slower to use. Users have to find order sets in the first place and this may worsen usability. The point of practice guidelines is to alter clinician’s behavior. But, the extent to which the user relies on the interface in opposition to his own personal judgment may be a challenging concern. Users in this study sometimes appeared to be choosing selections from menus somewhat facilely. Lastly, configuration and maintenance of these systems by designers and implementers consumes a great deal of resources. (92)

In the same paper, Payne also made recommendations in the use of order sets. Here are three that are worth consideration in terms of the effect of order sets upon usability (92):

1. The more order sets that are present, the more difficult it is to find what is needed. One should create a clinically oriented and clear hierarchy of menus to locate these.

2. One should design an ordering system that allows the clinician to follow a clinical pathway (by condition or purpose), to order with guidance and to order complex orders quickly.
3. Balance the use of pre-configured items with patient safety.

6.2.4.6 Summary of CPOE specific interface issues.

I have listed five items that seem to be important to consider in CPOE interface design to optimize usability – synonyms, alerts, default values, preselected orders and order sets. Some of these represent a spectrum of features which may present a tradeoff of benefits at either end of that spectrum. For example, different "flavors" of orders sets trade off speed versus flexibility. The next section will describe attributes of each of the three systems studied and discuss the benefits and liabilities of these various characteristics.

6.2.5 Comparison of systems.

In this section, I will attempt to compare and contrast the CPOE systems observed based upon some salient usability issues. I will attempt to succinctly characterize each system as a "type" based on certain functionality. I will then outline how I compared the raw data from the each system's observation think aloud sessions and the limitations of such an analysis. Lastly, I will discuss the ideas generated from those comparisons.

6.2.5.1 Comparison of the features of the three CPOE systems observed.

In this study, I observed physicians using three commercially available CPOE systems. There are distinct differences in the interfaces of these systems, but there is no established vocabulary that characterizes CPOE systems in terms of these differences. To that end I will initially describe the systems in general terms. I will attempt to characterize the human-computer interface experience of entering medical orders in a way that captures the style or type of each system uniquely. Payne's description of order set types discussed in Section 6.2.4.5 goes a long way in describing the spectrum of possibilities of the order entry process and this typology will be considered.

One of the main differences between these systems was the number of steps taken to enter an order. This ranged from a single click to a step-by-step process in which each component or field in the order "sentence" had to be entered individually. The former is a "pre-configured" or "quick" order set and the latter is the other end of the spectrum, the order dialog or menu- and prompt-driven style. A single step is obviously faster and easier than multiple steps. The single step is only useful, however, if every detail of that order is exactly how the user wants it. A deviation or modification of a complex order sentence was generally somewhat difficult for the user. Thus the speed of the "quick" order process is offset by its relative inflexibility. On the other hand, modification of any component of a multiple-step order creates no increased difficulty, as that is what is required in the first place. Generally speaking this multiple step type of order entry is more flexible.

The three systems differed with respect to this consideration. The System 1 interface generally employed a multi-step process. System 2 almost exclusively used orders that were already pre-arranged in an order sentence. System 3 was similar to System 2 except there was often an additional step required that asked the user details about the order sentence. I will elaborate somewhat for clarification.

System 1 used a simple display that contained four panes. See Figure 13. The left side of the display listed the active orders and the right side was split into three panes. Generally, one of the three panes listed a menu and a second pane listed detailed choices as a response to a selection from the menu. One could select an item from any pane with a mouse-click as a means of entering a choice or to bring up more information about it in another window. A third pane allowed text entry, as a means of selecting the number or letter of an item choice in another pane, or for entering free text to search or complete textual information. In addition, orders could be entered from pop-up forms that contained collections of related orders. Other than these forms and some “quick orders,” most orders required multiple steps to input each component of the order sentence.

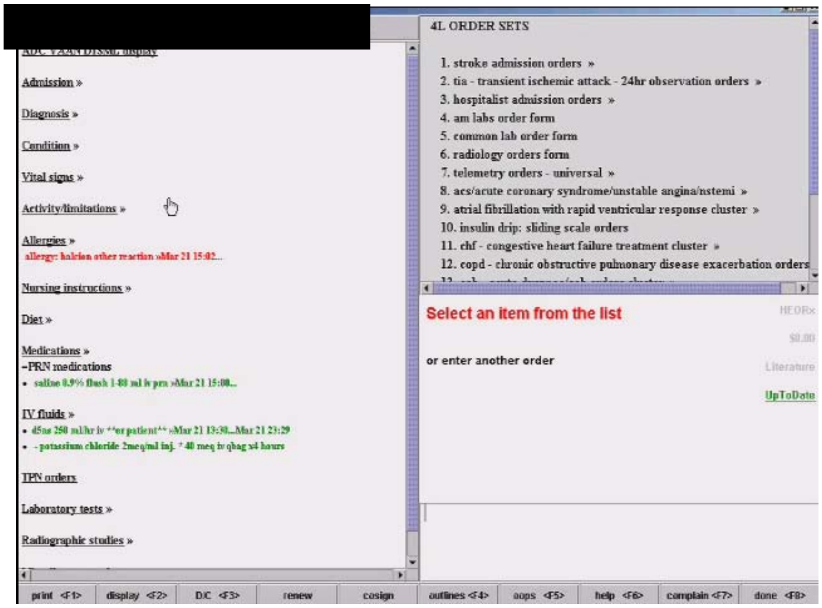


Figure 13. Screenshot of System 1

System 2 made more use of pre-configured order sentences. A typical display is seen in Figure 14. In this system, the user merely checks off an item from a list in the order set in the upper half of the display. The details of the order that is highlighted are already pre-configured and displayed in the lower half. In this example the user has ordered a “Comprehensive Metabolic Panel” by clicking in the check box next to that item. As seen in “Details” in the lower part of the display, this lab test is to be collected in an “Early AM Draw”, date and time listed and only performed once (“Frequency”). Any modification to that can be done in the lower right pane, but is limited only to what is listed in the menu; no free text was possible. Although not always required, many times these changes were difficult, as the user had to find them from a long list or the choices were not exactly what was needed.

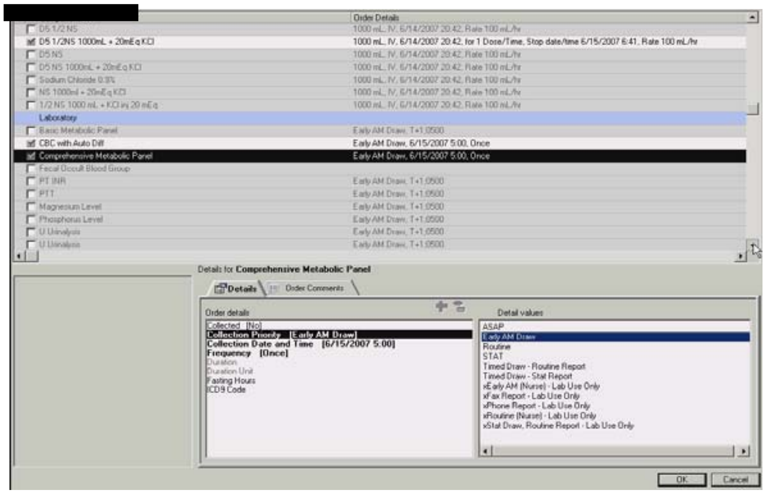
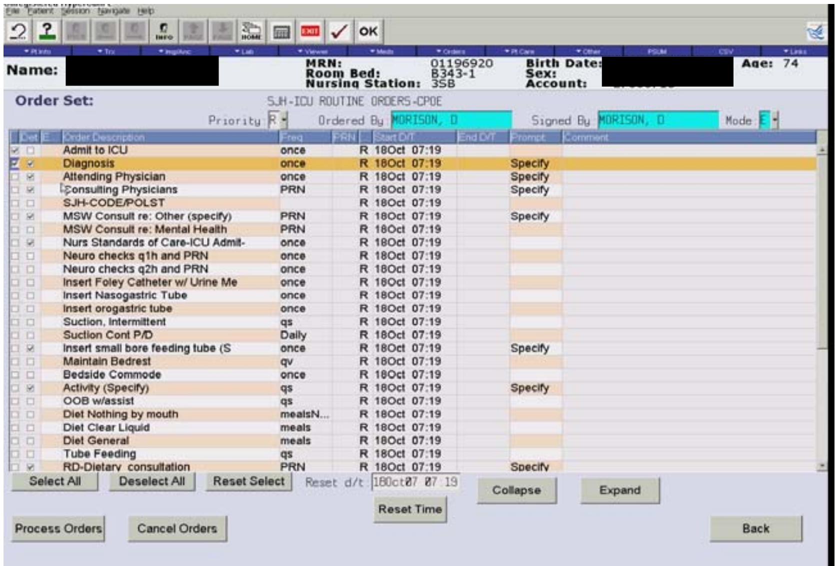


Figure 14. Screenshot of System 2

System 3 was similar to System 2. As seen in the upper half of Figure 15, orders are clicked by selecting a check box. However, after all the orders have been checked and a "process order" button is clicked, a series of pop-up windows appears as in the lower half of that figure, that asks for details about some of the previous orders. It was not at all clear to me while observing or analyzing the data, nor to the users whom I questioned, which orders would "reappear" in the subsequent pop-up for clarification and which would not. Often these clarifications were time consuming for the same reasons as in System 2.

Thus, these systems range in their speed of entry of complete order sentences and in the ability to flexibly modify any pre-configured order sentences. System 1 used mostly non-pre-configured orders that had to be entered in a stepwise fashion, but the detail could be changed readily. The Systems 2 and 3 had mostly pre-configured order sentences which were faster to enter, but deviations from these were somewhat unwieldy, and especially so in System 3, as the users frequently verbalized.



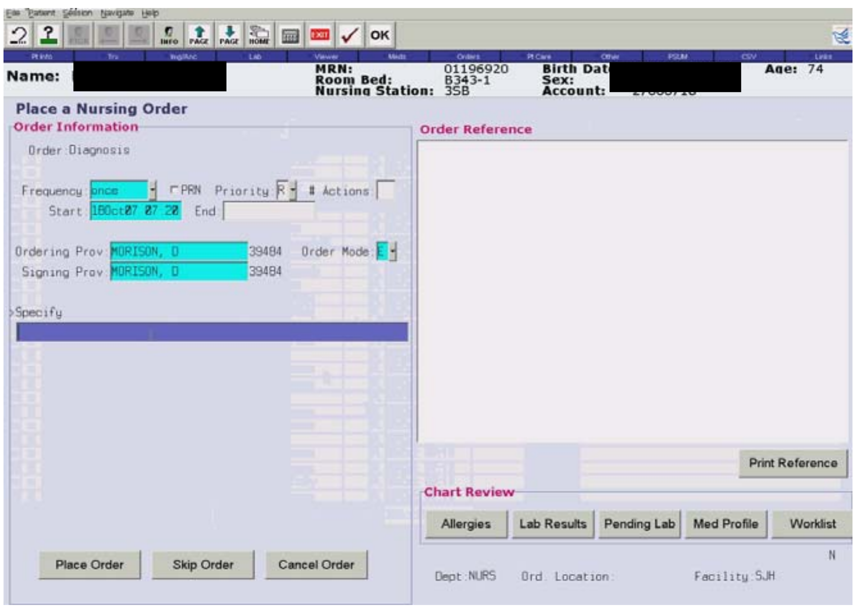


Figure 15. Screenshot of System 3. Upper panel shows an order set and the lower panel a confirmation window of the "diagnosis" order.

6.2.5.2 An analysis of the codes that predominated in the data analysis of each system

The goal of this analysis is to consider if certain "types" of order entry interfaces have any usability advantages, at least as observed in the 28 users of this study. To briefly review, the data were studied qualitatively by first dividing all the audiovisual data into discrete segments, each of which roughly corresponded to a single order. Each segment was labeled by a "cognitive," "usability" or other code based on my subjective observation of the segment. By comparing the frequency with which I applied certain codes to interface characteristics and user behavior in each of the systems, this might reveal differences in specific areas of usability issues among the three systems.

This quantitative approach is problematic, however. Prior to this point, this study has been designed with the assumption that the unit of analysis has been the data segment, not the computer system. There was no attempt at randomization or other purposeful participant selection that would readily allow inferences to be made that may apply outside these observations. There were rather different types of physicians and admission orders observed in the systems (e.g. Systems 1 and 2 were primarily medical/hospitalist admissions.) The sessions differed in duration and complexity. Also, these observations of computer user interaction may reflect more the local configuration of the systems than the "underlying" systems. Lastly, a single particularly troublesome user experience could generate many codes applied to those segments and could skew the data.

Given those qualifications, I chose to look at some of the more frequently applied codes and determine how often they occurred in each system per user. I will simply compare these raw frequencies without any attempt to describe these comparisons statistically. This is because the labels are subjectively applied to the data segments, even though every attempt was made to be consistent throughout the analysis. That being said, a great deal of data was reviewed – 28 users and over 500 order segments. Some very striking differences or trends among the systems may have some significance.

I reviewed the reports and counted how often certain codes appeared among the three systems, correcting for the number of users observed per system. I chose to compare only the most frequently used codes (arbitrarily chosen as 10 or more, see Table 16) as some codes were applied only a handful of times. In addition, some codes really only apply to the local configuration, such as those that reflect the use of alerts or default values, and those were not included. Lastly, because each segment was also labeled with the order type, I analyzed whether any of those specific order tasks were related to a predominance of usability issues, as evidenced by the frequency of coding. Unfortunately, only "medication" orders appeared frequently enough to consider any difference in the usability coding among the systems. (When searching the segments for order type AND (Boolean) any other codes, the frequency was very small except for the most common order type, medications.)

The codes that appeared most frequently were (see Sections 5.6.0, 5.7.0 and 5.8.0 to review the detailed meaning of these codes):

1. Action Specification
2. Interpretation
3. Perception
4. Lack of knowledge of system content
5. Cognitive overload
6. Navigating – moving around the graphical user interface
7. Locating Items on the Graphical User Interface
8. Time and effort of data entry
9. More Work
10. Trust
11. Resignation
12. Guessing the action specification.

In the next section, the systems will be compared with respect to the frequency with which the above types of usability issues appeared.

6.2.5.3 Usability differences between the systems

The goal of the analysis described in this section was to determine if there are any differences in usability among the CPOE systems observed. As discussed in the previous section, the "type" of order entry functionality differs between the systems. Considering those differences, if the type of order entry related to usability issues, one would hypothesize that System 1 would differ from Systems 2 and 3 and the latter two would be more similar.

The three systems' primary differences in functionality may be summarized as follows:

System 1 – order dialog requiring multiple steps (slower/more flexible)

System 2 – order sentence, requiring a single step (faster/less flexible)

System 3 – similar to System 2, but with an additional follow up confirmation step

One might therefore further postulate the usability issues that relate to the time and effort of entering orders might differ among the systems. These relate to the "execution" arm of the Norman model (deciding what to do and then doing it). These are roughly items like "action specification," "system knowledge," "moving around the GUI," "effort of data entry" and "more work." The data are presented in Table 25. This table shows the counts of how many times a segment was tagged with a particular code in each system, divided by the number of users observed. Figures 16, 17 and 18 show the data graphically.

Although Systems 2 and 3 are similar to each other in design, the users of Systems 1 and 2 were more similar, in that they were nearly all medical hospitalists. The medical admissions observed in Systems 1 and 2 were generally more complex than those in System 3 and this may affect the observations unrelated to the design of the systems.

Code	**Number of segments coded per user**		
	System 1	System 2	System 3
action specification	1.73	1.44	0.50
interpretation	1.00	1.44	0.13
perception	0.91	1.33	0.88
lack of system knowledge	0.64	1.22	0.25
cognitive overload	0.45	0.33	0.13
moving around GUI	0.82	0.33	0.13
locating data	2.18	1.11	0.00
effort of data entry	0.73	0.44	2.38
more work	0.18	0.56	1.63
trust	0.91	1.00	0.00
resignation	1.27	0.22	0.13
guessing	1.00	0.56	0.00

Table 25. Counts of segments to which the code was applied per user in each system.

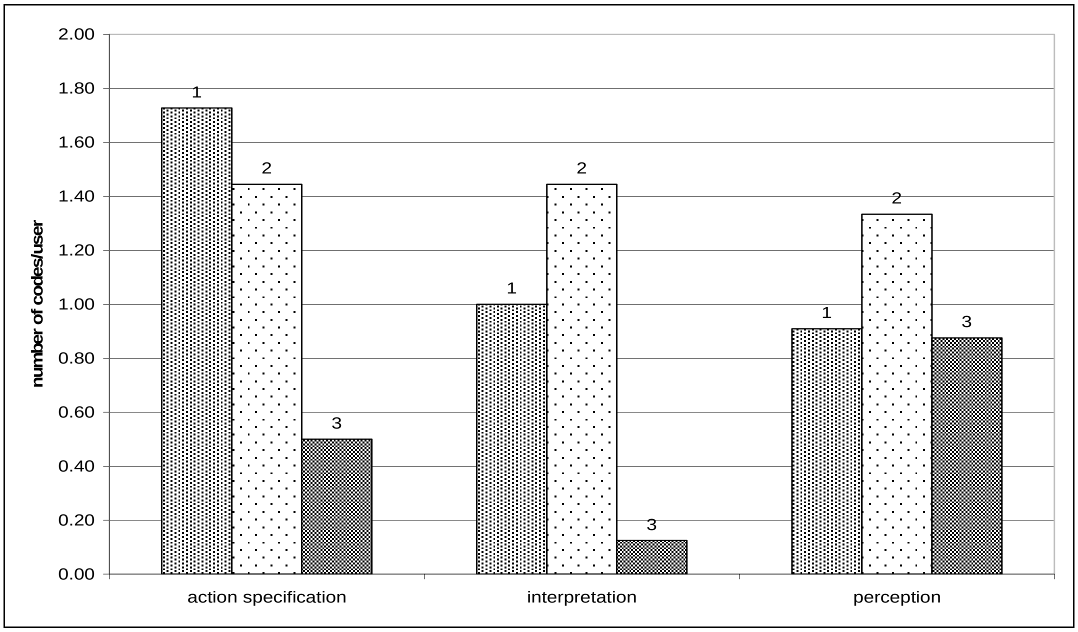


Figure 16. Counts of "cognitive" codes per user noted in each system. The numbers above the bars indicated System 1, 2 or 3.

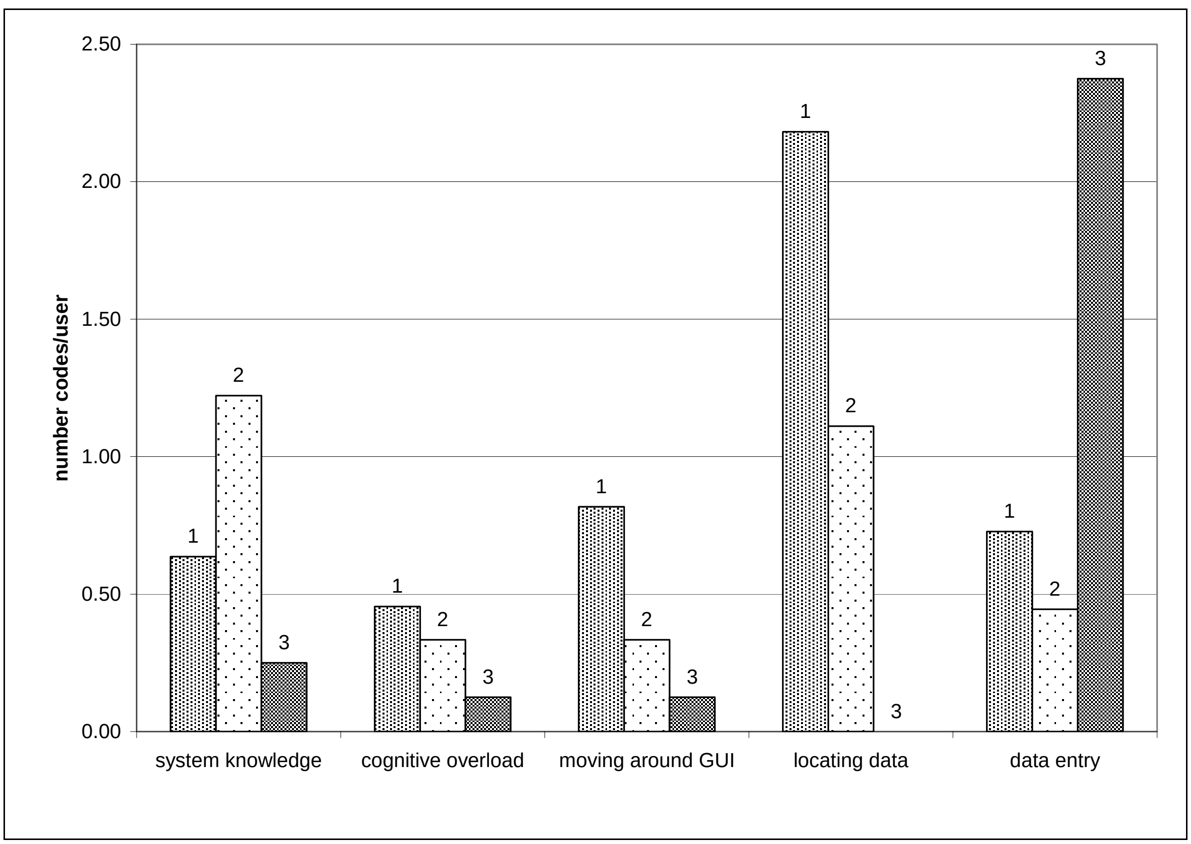


Figure 17. Counts of "usability" codes per user noted in each system. The numbers above the bars indicated System 1, 2 or 3.

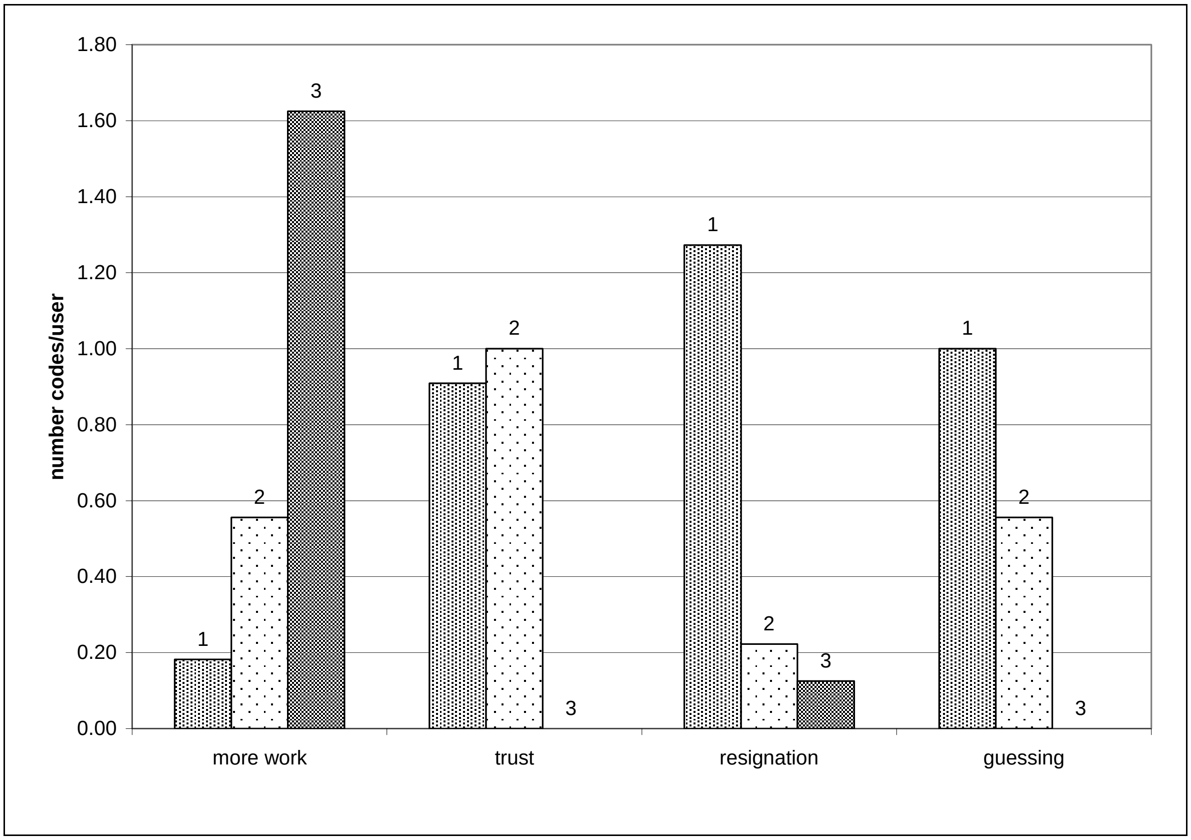


Figure 18. Counts of "grounded" codes per user noted in each system. The numbers above the bars indicated System 1, 2 or 3.

As stated above the data were then evaluated considering only a single type of task, entering an order for a medication. These ratios (code counts per system per user) were no different than those seen when considering all the tasks and will not be discussed.

Because of the qualitative nature of the data, statistical analysis of these quantitative results will not be helpful and may be misleading. A statistically significant difference in the code counts per system per user likely will have little meaning because of all the factors outlined in the previous section, 6.2.5.2. I do not consider a small difference to carry any significance. However, perhaps a several-fold difference in the presence of certain codes between systems has some relevance and only the most striking differences are discussed.

Although Systems 2 and 3 were more like each other in design compared to System 1, these differences were not consistently represented in the data. This difference between these two “types” was represented only by the segments that were coded for “difficulty locating data,” “guessing” and “resignation”; these were far more common in System 1. Segments that were coded for “effort of data entry” and “more work” predominated in System 3. System 3, however, has far fewer issues in “action specification” and “trust” than Systems 1 and 2. System 2 did not appear to stand alone in any respects compared to the other two. The other codes did not appear to show much difference between the systems.

Given the considerable limitation of this quantification some tentative generalizations may be made. In the system (1) that had mostly multiple-step order dialogs, the users were observed more to have difficulty locating data, guessed what to do next and gave up without reaching their goal. Contradictory to this, the users of a system (3) with mostly single-step order sentences were observed to be requiring more effort to complete their work, but the users had fewer issues in trusting their system and specifying which action to do next. This differed from the other single-step system (2). Recall that System 3 required an additional confirmation step that may explain its difference from System 2.

In summary, the multiple step system of prompt-driven order dialogs generally seemed to cause the users more effort. It is not clear how much this difference was offset by the increased flexibility of that type of system, but in general, fewer and more direct actions appeared to improve usability. Improvements in ease of use that the fewer-step order sentence style of order enter may confer were offset by secondary confirmation steps that required more work. Naturally the fewer actions required for admission orders, such as routine post-operative or post-partum patients (which can be accommodated with order sets), the more easily the user's order entering experience is facilitated.

Understanding the usability issues of a Computerized Order Entry system cannot be predicted in a straightforward way. There may not be one kind of system that is "best" and the attributes of the various types of systems may all possess their individual benefits and liabilities. In addition, the context of the individual user and the local configuration of the system have a lot to do with usability. It is clear, however, that numerous specific usability issues were discovered during these observations. Many of these issues may be universal, but these data may not extend to all systems and all users in all circumstances. It is possible, however, to perform these usability observations in any institution and the issues specific to each location may be elucidated. The following section discusses how a modified version of this study methodology might be used anywhere.

6.3 The significance and utility of this field usability evaluation method

This section argues that this type of usability testing outside the research setting is an important component in the implementation of CPOE systems. It is relatively easy to utilize and can provide valuable information about user issues that cannot otherwise be acquired. Observation of technology use in context generates a unique and important understanding of user behavior that is required to create successful interface design.

6.3.1 Technology required for field usability evaluation

A formal usability laboratory is not required for this field usability testing. The equipment, software and location required are relatively simple and inexpensive. One merely needs a networked computer with the CPOE application, screen capture software, a microphone headset and a means to transfer the video file to another computer for analysis. In this study, I also video recorded the user to observe any activity not recorded on the screen. This would, of course, require the inclusion of a video camera and video editing software. In this study, the concomitant use of the video camera data added little additional information from which to understand usability problems. The video editing processes are also quite time consuming. Therefore, for the purposes of employing this method in a clinical setting, these could easily be omitted with little loss of valuable information. This "portable usability" laboratory type of usability testing is what has been referred to as a "discount" usability engineering method. It has been argued that this relatively quick, easy and inexpensive evaluation of interface design has the "most bang for the buck." (33, 140)

6.3.2 What can an individual organization learn from testing in the field?

Harrison et al. stressed the importance of evaluation of health care technology in the context of the user's work experience and his social and organizational environment. This analysis emphasized examining how clinicians actually use the health information technology ("HIT-in-use"), rather than studying the use that was planned by designers or managers. "Examining HIT-in-use provides opportunities for understanding and responding to user experiences and emerging needs, for example by changing the pace of implementation or reconfiguring IT properties." (141) This statement also suggests that observing users in a laboratory under artificial conditions may miss some valuable information about the user's interaction with the clinical information system.

As demonstrated in this study, watching users do their actual work live was a very effective way of uncovering specific usability problems within each locally configured system. This type of testing uncovered problems that never would have been seen in a laboratory using fabricated scenarios or a list of prescribed tasks. The users verbalized issues with which they are quite familiar but had never spoken of before. They do not note these issues in any permanent way, so the individuals often forget to mention them when they are asked about them later. They may recall them in a general sense, but not the specific details. This method documents and preserves that memory.

This method can also be a very effective way of understanding clinician workflow in a "microscopic" way. Task analysis can watch users move about and interact. Cognitive walkthroughs by experts can outline the steps of a process. But given the numerous permutations of steps possible in any given task, every physician does each task seemingly in a unique way. Observing a user's individual approach is a way to document this behavior with the goal of potentially streamlining the task for all users. For example, as discussed in Section 5.1.2.1.1, users demonstrated idiosyncratic ways of navigating order sets. This level of detail could never be discovered or understood without these direct observations.

6.3.3 Suggestions for performing field testing and analyzing data

IT departments could easily make this user testing a routine part of implementing any clinical application. This could be performed before the roll out of a new application or a revised version. Testing users periodically and regularly could identify usability issues that arise with the initial use of an application and those that develop later as users become more experienced. The information that is obtained from this testing may serve to advise a reconfiguration of the system or to refer to developers for future upgrades. In addition, this can identify areas of user problems that can be improved with further training. The results can advise the training of new users but also can be helpful for refresher courses for individuals or the entire staff depending on the prevalence of the user misunderstanding.

Based upon the results of this study, this usability testing would be most beneficial if employed while observing physicians during the course of their workday as they are using the computer while actually taking care of patients. No users in this study felt the testing impeded their ability to perform their duties. Because one is observing actual patient data, precautions to safeguard its confidentiality should be rigorously enforced and the files should be deleted as soon as usability issues are tabulated. Under the Privacy Rule of the Health Insurance Portability and Accountability Act, the use of the Protected Health Information is permitted by a covered entity without a patient's authorization, for the purposes Health Care Operations. (142)

This usability testing does not need to be performed on all users. There is evidence that a large proportion of usability problems can be identified with a small number of users, usually ten or less, as noted in Figure 19. This usability testing perhaps could be required as a part of initial or ongoing training. Alternatively, one could compensate the users for their time.

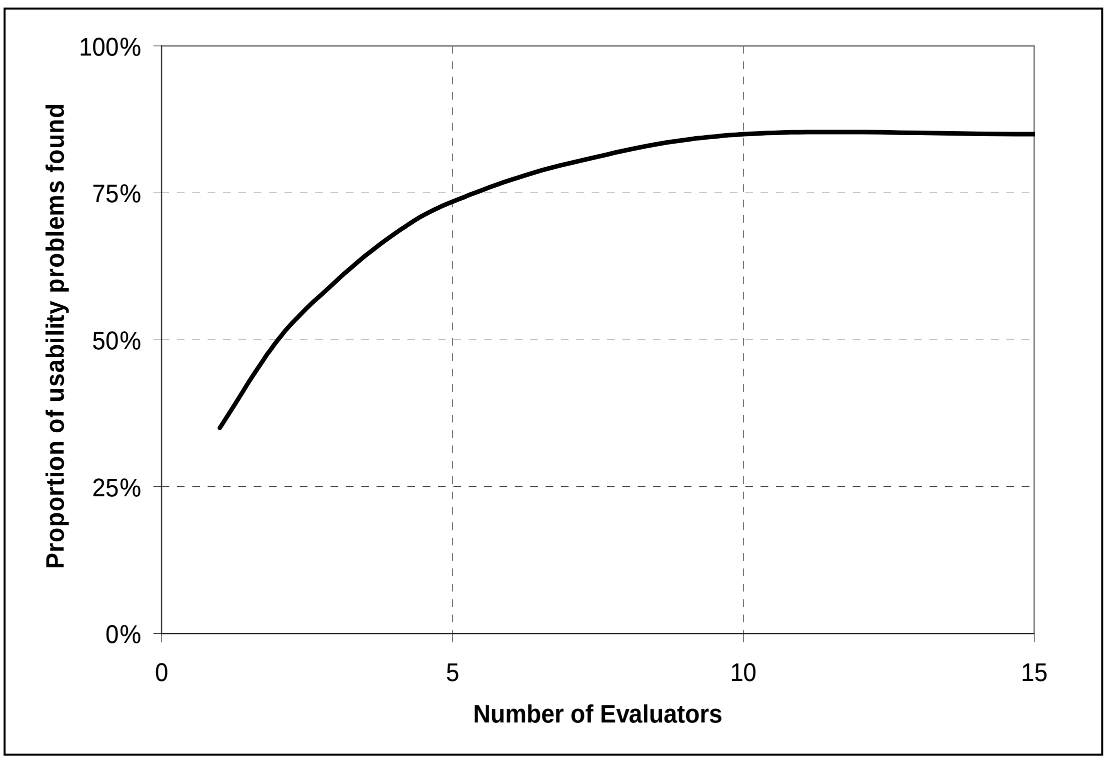


Figure 19. Usability problems as a function of the number of users tested. This figure shows the average results of six studies discussed by Nielsen. (143)

The analysis of the data in this study was thorough, rigorous and time consuming as the goal was a broad understanding of this phenomenon. For the purposes of institutional testing the analysis process could be greatly streamlined. Transcription of the verbalizations need not be done. This process and that of segmentation and time-stamping were arduous, but were required to utilize the video analysis software. This software served to facilitate the analysis of coding and this effort would be beyond the needs of a local implementer. Likewise, the detailed documentation of every user action is unnecessary. All that needs to be performed is an observation of the video file while listening to the users' verbalizations. From this, the analyst could merely enumerate the specific usability issues and decide what steps may mitigate them. The many months of analysis realistically could be reduced to hours or days. By reviewing the video file, the analyst may quite easily create a list of usability problems to be addressed.

Although the steps needed to correct the usability issues may appear obvious, one could also employ some of the analysis strategies described in this study. For example, one could consider an issue individually and review it in terms of the Norman model. Before making a decision about any changes to the interface one might ask the following questions:

1. What is the user's intention?
2. Can he specify the action?
3. Can he execute it?
4. Can he perceive changes to the interface?
5. Can he interpret it?
6. Can he evaluate if his intention has been met?

Alternatively, considering the analysis of the data in Section 6.2.2, one might ask the following regarding any point in which usability troubles are noted or interface choices need to be decided:

1. Does the user know what to do next?
2. Can the user see on the display what he needs to see?
3. Can the user understand what he sees?
4. Can the user find the item he is looking for?
5. Can the user enter data without excessive effort?
6. Does the user know how the computer system works in this situation?

Usability testing in any institution implementing clinical information technology applications should be routine. It entails relatively little effort and the information obtained is valuable. These efforts promote a user-centered approach to implementation to achieve the goals of improved user acceptance and clinical effectiveness.

6.4 Limitations

6.4.1 The users.

One problem with the design of this research is the lack of purposive participant selection. My original intention was to recruit a spectrum of CPOE users and administer the modified Computers in Medical Care survey. Based upon the results of that inquiry, I hoped to observe users with a range of computer sophistication and experience, from novice to expert. Instead, I was limited to observing the only users who were willing and available. In general these users were the most experienced in the use of their individual

CPOE systems and probably of above average enthusiasm about clinical information technology among their colleagues. The reasons for this varied between hospitals, but the most prominent explanation was that there was simply a limited number of non-resident physician users who entered computerized admission orders. In some situations there were no inexperienced users who routinely performed the task required for this study. In addition, I did have discussions with a few of the less experienced and less skilled computer users, but they were reticent and declined to volunteer. I suspect there were others who felt similarly. The task itself was problematic because it occurred relatively infrequently among the potential participants in the limited time I was available to gather data.

Thus, due to this limited variation in user attributes, the observations may not represent the experience of a wider range of users. However, even though the users were not heterogeneous in some attributes, they may have represented the typical physician that uses computerized order entry the most in these institutions. Indeed, in one hospital I observed essentially all the physicians who were using CPOE. This may reflect a selection bias in the participants of this study, but it also may represent the limited penetration of CPOE among community hospital-based physicians. If the users in this type of setting are themselves generally homogeneous, these participants may actually be a representative cross-section. While this precludes, of course, any observations or commentary about novice users, it is instructive in that the usability problems are common even among experienced and enthusiastic users. Lastly, the observation of 28 users interacting for hours with their respective computer systems generated a large amount of data. While these findings cannot be probabilistically applied elsewhere, there is an extended amount of user behavior that was observed and represents a significant body of information.

As with any qualitative data obtained by observing participants, this can only truly be representative of the users observed. True statements of generalization cannot be made. However, it would also be a mistake to consider these observations isolated incidents. Many of the same usability problems were experienced by multiple users. There was no obvious disconfirming evidence of the usability issues. That does not mean there is not any, but the lack of any obvious contradictions means something. Likewise the existence of corroboration and similarities among users and between systems is meaningful.

In summary, ideally one wants to select users for purposeful reasons that align with the study design. Contrary to the initial intentions of the study design, the selection of participants in this study amounted to a "convenience sample" and thus may lack some methodological rigor. (144) However, the users observed were certainly representative of the users in their respective institutions and the observations yielded a considerable amount of data that demonstrated a wide variety of usability issues.

6.4.2 The observations

A major goal of this research was to observer users "in the field" in circumstances that mimicked as much as possible the users' usual work experience. The observation of users entering orders on actual patients in real time went far to create a naturalistic environment. Although the task itself was as realistic as possible, there were some factors that create artificiality. In nearly all circumstances, the users were observed at a workstation that differed from where the users were accustomed to working. The interaction with the computer was the same as usual but the environment was unique.

The users were clearly aware this was "research" and that they were being observed. While this cannot completely be avoided, the users often tended to regard me as someone who was somehow connected with their local IT department who was there to "fix" the system. This mindset did not seem to suppress any comment, but it may have interfered with understanding the users' cognitive processes as they expended cognitive efforts having a dialogue with an observer. In general, the relative artificiality of the testing process cannot be avoided as it is inherent in this type of observation.

Ethnographic research requires knowledge of the domain under study. As a physician, I am already "immersed" in the culture and have a thorough understanding of it. This position was invaluable in analyzing the data. It enabled me to infer intention rather easily by understanding the clinical task at hand. Indeed, I feel a non-clinician would not be able to ascertain the nuances of user behavior as much of it is implicit and unvoiced. To the extent that familiarly with the culture of the users interfered with objectivity of the observations, I cannot know, but this is a possible limitation. Also, as a solo researcher, I lacked the ability to corroborate my impressions of the observations with other researchers analyzing the data who might have a different and complementary viewpoint.

6.4.3 The methodology

As was discussed in the Background section 2.1.3, there are concerns of limitations in think-aloud observation methods. Insofar as gaining an understanding of the cognitive processes of the user, there is concern that the actual process of talking aloud during a task that is usually done in silence adds an additional cognitive load on the user, and thus may not reflect what goes on for the user when customarily performing his work. In addition, critics of this research have suggested that researchers worsen the cognitive load by intervening with the participants to assure that crucial information is obtained.

In the observations conducted in this study, I attempted to remain as unobtrusive as possible, with the goal of remaining faithful to the theoretical basis of the method and mindful of the critics by limiting the cognitive interruptions. Although talking out loud may alter a user's cognitive processes, it provides the only access to the user's thoughts. My relative paucity of interaction may have had another untoward effect. The user's often verbalized very little despite my urgings to "keep talking," one of the "allowable" spoken interventions of the observer. These concerns are paradoxical, having the simultaneous concern that the user was verbalizing at all, but not enough.

There exist two extremes in the approach to this method. On one hand, the researcher attempts to maintain maximum rigor in usability testing by limiting the observer intervention. The other extreme is the "quick and dirty" testing that makes no attempt to employ any methodological soundness as long as some results are obtained. There is an optimal middle ground here. One needs to realize that the manner in which usability testing is performed has a great influence on the data that are collected. However, in some circumstances, it may be crucial to collect some specific information and the observer may make an effort to intervene somewhat. (45)

To the extent that the verbalizations recorded in this method do not accurately reflect the cognitive processes of the user, there are limitations on any declarations about CPOE users' cognition. On the other hand, many usability issues were observed, the degree to which they are cognitive representations notwithstanding. The user behavior was recorded as observed, and the less I interacted with the user, the more this behavior was "unsolicited," the rationale behind limited observer-user interactions. The goal of this research was not to strictly understand detailed mental processing, but rather to use a cognitive perspective with which to view the user behavior. To that end, these observations were successful.

Chapter 7. Summary

The aim of this research was to understand usability issues as they pertain to commercially available computerized provide order entry (CPOE) systems in community hospitals. CPOE is felt to be an important factor in improving the safety and quality of our health care system, but clinician acceptance of this technology has been lackluster. One important reason for clinicians' lack of enthusiasm about CPOE is that it is often difficult to use as it increases users' work and may have other unintended consequences. If one hopes to improve the user experience with this technology, one needs to clearly understand the issues related to human-computer interaction. One method of studying the interface of humans and computers is the think-aloud study. Using this method, one observes and records a user as he performs a task at the computer while verbalizing his thoughts about this process. This method seeks to understand the cognitive processes of the user as well as highlight usability problems that occur.

A framework for this study was Norman's Theory of User Action. This model studies the interface between the user's mental processes and the physical aspects of the computer technology. There are two major processes that are described. In the "execution" process, the user must decide upon his goals and generate ideas about how to carry out those goals by physically manipulating the computer, which results in a change in the state of the computer. The second part is the "evaluation" process, wherein the user must perceive any changes in the computer display that occur, interpret them and decide if the original goal has been met. The extent to which there are “gulfs” between the execution and evaluation results in problems with usability.

The subjects in this study were physician users of three different commercial CPOE systems in community hospitals in Oregon and Washington. The task observed was the entry of hospital admission orders on an actual patient in real time. The users “thought aloud” as they entered the orders and their verbalizations were audio recorded. Screen capture software recorded all action on the computer display and a video camera recorded the image of the user working with the computer. The verbalizations were transcribed verbatim and synchronized with the video recordings of the computer display and the user. These data were divided into usable segments according to individual order tasks and each segment was coded qualitatively according to three coding schemes. The first scheme captured the points of the Norman model above. The second scheme looked at known usability rules from the literature. The data were coded in a third way using a grounded approach that arose from review of the data. All segments were grouped according to their codes and reviewed repeatedly.

In all, 25 users were studied for 28 different order entry sessions. Most users were hospitalists or internists and all were of above average computer sophistication and experience. Despite the constraints of the usability testing the users’ orientation toward computer use remained decidedly clinical. As a result, the users tended to be less verbal than hoped, although detailed analysis allowed thorough understanding of their intentions and actions. The Norman model proved to easily describe the actions of the users as they interacted with the computer in terms of their mental processes.

A great number of usability problems were noted; about half of all segments demonstrated at least minor usability issues. In analyzing the data from a cognitive point of view, I found that the users had most difficulty specifying which action to undertake next as well as perceiving and interpreting information from the computer display. Looking at usability rules, I discovered that the users had difficulty locating items on the display, lacked an adequate understanding of the workings of their respective systems and expended an undue amount of time and effort entering data. Further analysis demonstrated examples of mistrust of the computer, giving up on intended actions and guessing what to do next. Specific CPOE interface items that caused the most problems were alerts and reminders, poor default values, long lists of items and the lack of adequate synonyms. Orders sets, on the other hand were a significant asset to usability.

This study demonstrated that community physician users of commercial CPOE systems experienced a number of usability problems in the course of their work. Observing these users from numerous perspectives allows the understanding that the disparity between the users' understanding of these systems and the way they were designed creates problems in usability. The testing of users in the field is an effective way of uncovering these problems and is a feasible method to be used by any institution hoping to improve the CPOE user's experience.

References.

1. Simon, H. Designing Organizations for an Information-Rich World. In: Greenberger, M, editor. Computers, Communication, and the Public Interest. Baltimore, MD: The Johns Hopkins Press; 1971. p. 40-41.
2. Han Y, CJ, Venkataraman S, Clark R, Watson R, Nguyen T, et al. Unexpected increased mortality after implementation of a commercially sold computerized physician order entry system. Pediatrics 2005; 116(6):1506-12.
3. Koppel, R, Metlay, J. P., Cohen, A., Abaluck, B., Localio, A. R., Kimmel, S. E. and Strom, B. L. Role of computerized physician order entry systems in facilitating medication errors. JAMA 2005;293(10):1197-1203.
4. Berger, RG, Kichak, JP. Computerized physician order entry: helpful or harmful?[see comment]. Journal of the American Medical Informatics Association 2004;11(2):100-3.
5. Institute of Medicine. Crossing the Quality Chasm: A New Health System for the 21st Century. Washington, DC: National Academy Press; 2000.
6. Institute of Medicine. To Err Is Human: Building a Safer Health System. Washington, DC: National Academy Press; 1999.
7. Institute of Medicine. Preventing Medication Errors. Washington, DC: National Academy Press; 2006.
8. Kuperman, GJ, Gibson, RF. Computer physician order entry: benefits, costs, and issues. Annals of Internal Medicine 2003;139(1):31-9.
9. Ferman, JH. Health information technology: tentative federal and congressional developments. Healthcare Executive 2006;21(1):40.
10. Chaudhry, B, Wang, J, Wu, S, Maglione, M, Mojica, W, Roth, E, Morton, S, Shekelle, P. Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. Annals of Internal Medicine 2006;144:742-752.
11. American Hospital Directory Inc. Updated June 22, 2006 [cited August 23, 2006]; Available from: http://www.ahd.com/
12. Ash, JS, Sittig, DF, Seshadri, V, Dykstra, RH, Carpenter, JD, Stavri, PZ. Adding Insight: A Qualitative Cross-Site Study of Physician Order Entry. International Journal of Medical Informatics 2005;In Press.
13. Poon, EG, Blumenthal, D, Jaggi, T, Honour, MM, Bates, DW, Kaushal, R. Overcoming the barriers to the implementing computerized physician order entry systems in US hospitals: perspectives from senior management. AMIA 2003;Annual Symposium Proceedings/AMIA Symposium.:975.
14. Ash, JS, Fournier, L, Stavri, PZ, Dykstra, R. Principles for a successful computerized physician order entry implementation. AMIA Annual Symposium Proceedings 2003:36-40.
15. Campbell, EM, Sittig, DF, Ash, JS, Guappone, KP, Dykstra, RH. Types of Unintended Consequences Related to Computerized Provider Order Entry. J Am Med Inform Assoc 2006;13(5):547-556.
16. Ash, JS, Gorman, PN, Lavelle, M, Payne, TH, Massaro, TA, Frantz, GL, Lyman, J. A Cross-site Qualitative Study of Physician Order Entry. Journal of the American Medical Informatics Association 2003;10:188-200.

17. Gainer, A, Pancheri, K, Zhang, J. Improving the human computer interface design for a physician order entry system. AMIA Annual Symposium Proceedings 2003;/AMIA Symposium.:847.
18. Windle, J, Van-Milligan, G, Duffy, S, McClay, J, Campbell, J. Web-based physician order entry: an open source solution with broad physician involvement. AMIA 2003;Annual Symposium Proceedings/AMIA Symposium.:724-7.
19. Sommerville, E. Software Engineering. 6th ed. Harlow, England: Addison-Wesley Pearson Education; 2001.
20. Johnson, CM, Johnson, TR, Zhang, J. A user-centered framework for redesigning health care interfaces. Journal of Biomedical Informatics 2005;38(1):75.
21. Devitt, N, Murphy, J. A survey of the information management and technology training needs of doctors in an acute NHS trust in the United Kingdom. Health Information & Libraries Journal 2004;21(3):164-72.
22. Poon, EG, Blumenthal, D, Jaggi, T, Honour, MM, Bates, DW, Kaushal, R. Overcoming barriers to adopting and implementing computerized physician order entry systems in U.S. hospitals. Health Affairs 2004;23(4):184-90.
23. Lacher, D, Nelson, E, Bylsma, W, Spena, R. Computer Use and Needs of Internists: A Survey of Members of the American College of Physicians–American Society of Internal Medicine. Proceedings of the AMIA Symposium 2000:453-6.
24. Ferren, AL. Gaining MD buy-in: physician order entry. Journal of Healthcare Information Management 2002;16(2):66-70.
25. Shneiderman, B. Leonardo's laptop: human needs and the new computing technologies. Cambridge, MA: The MIT Press; 2002.
26. Lorenzi, NM, Riley, RT, Blyth, AJ, Southon, G, Dixon, BJ. Antecedents of the people and organizational aspects of medical informatics: review of the literature. Journal of the American Medical Informatics Association 1997;4(2):79-93.
27. Kaplan, B. Addressing organizational issues into the evaluation of medical systems. Journal of the American Medical Informatics Association 1997;4:94-101.
28. Kushniruk, AW, Patel, VL, Cimino, J. Usability Testing in Medical Informatics: Cognitive Approaches to Evaluation of Information Systems. AMIA Annual Symposium Proceedings 1997:218-222.
29. Davis, FD. Perceived usefulness, perceived ease of use and user acceptance of information technology. MIS Quarterly 1989;September:319-340.
30. Venkatesh, V, Davis, F. A theoretical extension of the technology acceptance model: four longitudinal field studies. Management Science 2000;46(2):186-204.
31. Venkatesh, V, Davis, FD. A model of the antecedents of perceived ease of use: development and test. Decision Sciences 1996;27(3):451-481.
32. Venkatesh, V. Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model. Information Systems Research 2000;11(4):342-366.
33. Kushniruk, AW, Patel, VL. Cognitive and usability engineering methods for the evaluation of clinical information systems. Journal Of Biomedical Informatics 2004;37(1):56-76.
34. Nielsen, J. Usability Engineering. San Diego, CA: Academic Press; 1993.

35. Norman, D. Cognitive engineering. In: Norman, D, Draper, S, editors. User centered system design. Hillsdale, NJ: Lawrence Erlbaum Associates; 1986. p. 31-61.
36. Carroll, J. Toward a multidisciplinary science of human-computer interaction. In: Carroll, J, editor. HCI Models, Theories and Frameworks. San Francisco, CA: Elsevier Science; 2003.
37. Barnard, P. The contribution of applied cognitive psychology in design of humanly useful and usable systems. In: Baecker, R, Grudin, J, Buxton, W, Greenberg, S, editors. Readings in Human Computer Interaction: Toward the Year 2000. San Mateo, CA: Morgan-Kaufmann; 1995. p. 640-658.
38. Carroll, JM. Human-computer interaction: psychology as a science of design. Annual Review of Psychology 1997;48(1):61-83.
39. Woods, D, Roth, E. Cognitive systems engineering. In: Helander, M, editor. Handbook of Human-Computer Engineering. North Holland: Elsevier Science Publishers; 1988. p. 3-42.
40. van Someren, W, Barnard, F, Sandberg, A. The think aloud method. A practical guide to modeling cognitive processes. London: Harcourt Brace & Co.; 1994.
41. Ericsson, K, Simon, H. Protocol Analysis: verbal reports as data. Cambridge, MA: MIT Press; 1993.
42. Boren, M, Ramey, J. Thinking Aloud: Reconciling Theory and Practice. IEEE Transactions on Professional Communication 2000;43(3):261-278.
43. Nielsen, J, Clemmensen, T, Yssing, C. Getting access to what goes on in people heads?- reflections on the think aloud technique. In: Proceedings of the second Nordic conference on Human-computer interaction; 2002 October 19-23, 2002; Aarhus, Denmark; 2002.
44. Jordan, P. An Introduction to usability. Bristol, PA: Taylor and Francis; 1998.
45. Ramey, J, Boren, M, Cuddihy, E, Dumas, J, Guan, Z, van den Haak, M, De Jong, M. Does think aloud work? how do we know. In: CHI '06 Conference on Human Factors in Computing Systems; 2006; Montreal, Quebec, Canada; 2006. p. 45-48.
46. Lindsay, P, Norman, D. Human information processing: an introduction to psychology. 2nd ed. New York: Academic Press; 1977.
47. Giacoppo, A. The role of theory in HCI. Choosing human-computer interaction appropriate research methods 2001 [cited December 20, 2006]; Available from: http://www.otal.umd.edu/hci-rm/theory.html
48. Jaspers, M, Steen, T, van den Bos, C. The think aloud method: a guide to the user interface. Methods of Information in Medicine 2004;73:781-795.
49. Card, S, Moran, T, Newell, A. The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates; 1983.
50. Olson, J, Olson, G. The growth of cognitive modeling in human-computer interaction since GOMS. Human-Computer Interaction 1990;5:221-265.
51. Hacker, W. Activity: a fruitful concept in industrial psychology. In: Frese, M, Sabini, M, editors. Goal directed behavior: the concept of action psychology. Hillsdale, NJ: Lawrence Erlbaum Associates; 1985. p. 262-283.
52. Norman, D. Some observations on mental models. In: Gentner, D, Stevens, A, editors. Mental Models. Hillsdale, NJ: Lawrence Erlbaum Associates; 1983. p. 7-14.

53. Norman, DA. Categorization of action slips. Psychological Review 1981;88(1):1-15.
54. Anderson, J. Cognitive Psychology and Its Implications, 6 Ed. New York: Worth Publishers; 2005.
55. Shneiderman, B. Designing the user interface: strategies for effective human-computer interaction, 2nd ed. Reading, MA: Addison-Wesley; 1992.
56. Patel, VL, Kushniruk, AW, Yang, S, Yale, J-F. Impact of a Computer-based Patient Record System on Data Collection, Knowledge Organization, and Reasoning. J Am Med Inform Assoc 2000;7(6):569-585.
57. Horsky, J, Kaufman, DR, Oppenheim, MI, Patel, VL. A framework for analyzing the cognitive complexity of computer-assisted clinical ordering. Journal of Biomedical Informatics 2003;36(1-2):4-22.
58. Horsky, J, Zhang, J, Patel, VL. To err is not entirely human: Complex technology and user cognition. Journal of Biomedical Informatics 2005;38(4):264.
59. Woods, D. Behind Human Error: Human Factors Research to Improve Patient Safety. Testimony at the National Summit on Medical Errors and Patient Safety Research. Quality Interagency Coordination Task Force (QuIC) September 2000 [cited August 11, 2006]; Available from: http://www.quic.gov/summit/testimony.htm
60. Johnson, CM, Turley, J. The significance of cognitive modeling in building healthcare interfaces. International Journal of Medical Informatics 2006;75:163-172.
61. Patel, VL, Kushniruk, AW. Interface design for health care environments: the role of cognitive science. Proceedings of the AMIA Symposium 1998:121-5.
62. Kushniruk, AW, Patel, VL, Cimino, JJ, Barrows, R. Cognitive evaluation of the user interface and vocabulary of an outpatient information systems. In: Proceedings of the AMIA Symposium; 1996; Washington, DC; 1996. p. 22-26.
63. Kushniruk, AW. Evaluation in the design of health information systems: application of approaches emerging from usability engineering. Computers in Biology and Medicine 2002;32(3):141-149.
64. Kushniruk, AW, Triola, M, Borycki, E, Stein, B, Kannry, J. Technology induced error and usability: the relationship between usability problems and prescription errors when using a handheld application. International Journal of Medical Informatics 2005;74:514-526.
65. Borycki, E, Kushniruk, AW. Identifying and preventing technology-induced error using simulations: application of usability engineering techniques. Heathcare Quarterly 2005;8(Special issue):99-105.
66. Li, Q, Douglas, S, Hundt, AS, Carayon, P. A heuristic evaluation of a computerized provider order entry (CPOE) technology. In: Pikaar, R, Koningsveld, E, Settels, P, editors. Proceedings of the IEA2006 Congress; 2006; Maastricht, The Netherlands: Elsevier.; 2006.
67. Card, S, Moran, T. The keystroke-level model for user-performance time with interactive systems. Communications of the ACM 1980;23(7):396-410.
68. Prumper, J, Frese, M, Zapf, D, Brodbeck, F. Errors in computerized office work: differences between novice and expert users. SIGCHI Bulletin April 1991;23(2):63-66.

69. Cork, RD, Detmer, WM, Friedman, CP. Development and Initial Validation of an Instrument to Measure Physicians' Use of, Knowledge about, and Attitudes Toward Computers. J Am Med Inform Assoc 1998;5(2):164-176.
70. Patton, M. Qualitative Research and Evaluation Methods. 3rd ed. Thousand Oaks, CA: Sage Publications; 2002.
71. Strauss, A, Corbin, J. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. 2nd ed. Thousand Oaks, CA: Sage Publications; 1998.
72. Glaser, B, Strauss, A. The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine Publishing; 1967.
73. Bevan N, BC, Cockton G, Nielsen J, Spool J, Wixon D. Panel: "The 'magic number 5':is it enough for web testing?" In: CHI 2003 extended abstracts; 2003; Ft Lauderdale, FL; 2003.
74. Hackos, J, Redish, J. User and Task Analysis for Interface Design. New York: Wiley; 1998.
75. Newman, W, Lamming, M. Interactive System Design. New York: Addison-Wesley; 1995.
76. Bates, DW, Cohen, M, Leape, LL, Overhage, JM, Shabot, MM, Sheridan, T. Reducing the Frequency of Errors in Medicine Using Information Technology. J Am Med Inform Assoc 2001;8(4):299-308.
77. Krall, M, Sittig, D. Clinician's assessments of outpatient electronic medical record alert and reminder usability and usefulness requirements. Proc AMIA Symp 2002:400-4.
78. Nebeker, JR, Hoffman, JM, Weir, CR, Bennett, CL, Hurdle, JF. High Rates of Adverse Drug Events in a Highly Computerized Hospital. Arch Intern Med 2005;165(10):1111-1116.
79. Bates, DW, Kuperman, GJ, Wang, S, Gandhi, T, Kittler, A, Volk, L, Spurr, C, Khorasani, R, Tanasijevic, M, Middleton, B. Ten Commandments for Effective Clinical Decision Support: Making the Practice of Evidence-based Medicine a Reality. J Am Med Inform Assoc 2003;10(6):523-530.
80. Shabot, M. Ten commandments for implementing clinical information systems. Proc (Bayl Univ Med Cent);17(3):265-269.
81. Filik, R, Purdy, K, Gale, A, Gerrett, D. Drug name confusion: evaluating the effectiveness of capital ("Tall Man") letters using eye movement data. Social Science & Medicine 2004;59(12):2597-601.
82. Ash, JS, Stavri, PZ, Kuperman, GJ. A consensus statement on considerations for a successful CPOE implementation. Journal of the American Medical Informatics Association 2003;10(3):229-34.
83. Osheroff, J, Pifer, E, Teich, J, Sittig, DF, Jenders, R. Improving Outcomes with Clinical Decision Support: An Implementer's Guide. Chicago: Thomson; 2005.
84. Shah, NR, Seger, AC, Seger, DL, Fiskio, JM, Kuperman, GJ, Blumenfeld, B, Recklet, EG, Bates, DW, Gandhi, TK. Improving Acceptance of Computerized Prescribing Alerts in Ambulatory Care. J Am Med Inform Assoc 2006;13(1):5-11.
85. Abookire, S, Teich, J, Sandige, H, Paterno, M, Martin, M, Kuperman, G, Bates, D. Improving allergy alerting in a computerized physician order entry system. Proceedings / AMIA Annual Symposium. 2000:2-6.

86. Hsieh, T, Kuperman, G, Jaggi, T, Hojnowski-Diaz, P, Fiskio, J, Williams, D, Bates, D, Gandhi, T. Characteristics and consequences of drug allergy alert overrides in a computerized physician order entry system.[see comment]. Journal of the American Medical Informatics Association 2004;11(6):482-91.
87. Payne, T, Nichol, W, Hoey P, Savarino, J. Characteristics and override rates of order checks in a practitioner order entry system. Proceedings / AMIA. Annual Symposium 2002:602-6.
88. Weingart, SN, Toth, M, Sands, DZ, Aronson, MD, Davis, RB, Phillips, RS. Physicians' Decisions to Override Computerized Drug Alerts in Primary Care. Arch Intern Med 2003;163(21):2625-2631.
89. Newby, D, Fryer, J, Henry, D. Effect of computerised prescribing on use of antibiotics. Medical Journal of Australia 2003;178(5):210-3.
90. Chertow, G, Lee, J, Kuperman, G, Burdick, E, Horsky, J, Seger, D, Lee, R, Mekala, A, Song, J, Komaroff, A, Bates, D. Guided medication dosing for inpatients with renal insufficiency. JAMA 2001;286(22):2839-44.
91. McDonald CJ, Overhage JM, Tierney WM, Dexter PR, Martin DK. The Regenstrief Medical Record System: a quarter century experience. International Journal of Medical Informatics 1999;54(3):225-53.
92. Payne, TH, Hoey, PJ, Nichol, P, Lovis, C. Preparation and Use of Preconstructed Orders, Order Sets, and Order Menus in a Computerized Provider Order Entry System. J Am Med Inform Assoc 2003;10(4):322-329.
93. Bobb, AM, Payne, TH, Gross, PA. Viewpoint: Controversies Surrounding Use of Order Sets for Clinical Decision Support in Computerized Provider Order Entry. J Am Med Inform Assoc 2007;14(1):41-47.
94. Ahmad, A, Teater, P, Bentley, TD, Kuehn, L, Kumar, RR, Thomas, A, Mekhjian, HS. Key Attributes of a Successful Physician Order Entry System Implementation in a Multi-hospital Environment. J Am Med Inform Assoc 2002;9(1):16-24.
95. Sittig, DF. personal communication 2008.
96. Stone, D, Jarrett, C, Woodroffe, M, Minocha, S. User Interface Design and Evaluation. San Francisco: Morgan Kaufmann; 2005.
97. Rasmussen, J. Information processing and human-machine interaction: an approach to cognitive engineering. New York: North-Holland; 1986.
98. Norman, D, Draper, S. User centered system design: new perspectives on human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum; 1986.
99. Jaspers, MWM, Steen, T. Cognitive engineering in interface design. Studies in Health Technology and Informatics 2002;90:123-7.
100. Patel, VL, Kaufman, DR. Medical informatics and the science of cognition. Journal Of The American Medical Informatics Association 1998;5(6):493-502.
101. Patel, VL, Kaufman, DR, Arocha, JF, Kushniruk, AW. Bridging Theory and Practice: cognitive science and medical informatics. In: Proceedings. Greenes, R, editor. MEDINFO '95; 1995; Vancouver, BC, Canada; 1995. p. 1278-82.
102. Kushniruk, AW, Patel, VL. Cognitive computer-based video analysis: its application in assessing the usability of medical systems. Proceeding of the Eighth World Conference on Medical Informatics 1995:1566-1599.

103. Johnson, CM, Turley, JP. The significance of cognitive modeling in building healthcare interfaces. International Journal of Medical Informatics 2006;75(2):163.
104. Pelayo, S, Leroy, N, Guerlinger, S, Degoulet, P, Meaux, JJ, Beuscart-Zephir, MC. Cognitive analysis of physicians' medication ordering activity. Stud Health Technol Inform 2005;116:929-34.
105. Zhang, J. Guest Editorial. Human-centered computing in health information systems. Part 1: Analysis and design and Part 2: Evaluation. Journal of Biomedical Informatics 2005;38:1-3, 173-175.
106. Andre, T. Determining the Effectiveness of the Usability Problem Inspector: A Theory-Based Model and Tool for Finding Usability Problems. A Dissertation thesis. In: Virginia Polytechnic Institute and State University; 2003.
107. Cuomo, D, Bowen, C. Understanding usability issues addressed by three user-system interface evaluation techniques. Interacting with Computers 1994;6(1):86-108.
108. Lim, K, Benbasat, I, Todd, P. An experimental investigation of the interactive effects of interface style, instructions, and task familiarity on user performance. ACM Transactions on Computer-Human Interaction 1996;3(1):1-37.
109. Rizzo, A, Marchigiani, E, Andreadis, A. The AVANTI project: prototyping and evaluation with a cognitive walkthrough based on the Norman's model of action. In: Symposium on Designing Interactive Systems; Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques; 1997; Amsterdam; 1997. p. 305-309.
110. Smith, A, Dunckley, L. Prototype evaluation and redesign: structuring the design space through contextual techniques. Interacting with Computers 2002;14(6):821.
111. Lewin, K. Field Theory in Social Science. New York: Harper Collins; 1951.
112. Brown, SH, Coney, RD. Changes in physicians' computer anxiety and attitudes related to clinical information system use. J Am Med Inform Assoc 1994;1(5):381-394.
113. O'Dell, D, Tape, T, Campbell, J. Increasing physician acceptance and use of the computerized ambulatory medical record. In: Frisse, M, editor. Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care. American Medical Informatics Association; 1992; Baltimore; 1992. p. 848-52.
114. Hier, DB, Rothschild, A, LeMaistre, A, Keeler, J. Differing faculty and housestaff acceptance of an electronic health record. International Journal of Medical Informatics 2005;74(7-8):657.
115. Wachter, RM. Hospitalists in the United States -- Mission Accomplished or Work in Progress? N Engl J Med 2004;350(19):1935-1936.
116. Clancy, C. AHRQ: Present and future activities impacting hospital medicine. Journal of Hospital Medicine 2006;1(4):253-256.
117. McAlearney, AS, Chisolm, D, Veneris, S, Rich, D, Kelleher, K. Utilization of evidence-based computerized order sets in pediatrics. International Journal of Medical Informatics 2006;75(7):501.
118. Rogers, E. Diffusion of Innovations. 5th ed. New York: Free Press; 2003.
119. Salthouse, TA. Perceptual, Cognitive, and Motoric Aspects of Transcription Typing. Psychological Bulletin 1986 May;99(3):303-319.

120. Brown, C. Comparison of typing and handwriting in "2 finger typists". Proceeding of the Human Factors Society 1988:381-5.
121. Anthony, L, Yang, J, Koedinger, K. Towared the application of a handwriting interface for mathematics learning. In: IEEE International Conference of Multimedia and Expo; 2006; Toronto; 2006.
122. Eslami, S, Abu-Hanna, A, de Keizer, NF. Evaluation of Outpatient Computerized Physician Medication Order Entry Systems: A Systematic Review. J Am Med Inform Assoc 2007;14(4):400-406.
123. Kuperman, GJ, Bobb, A, Payne, TH, Avery, AJ, Gandhi, TK, Burns, G, Classen, DC, Bates, DW. Medication-related Clinical Decision Support in Computerized Provider Order Entry Systems: A Review. J Am Med Inform Assoc 2007;14(1):29-40.
124. Taylor, L, Tamblyn, R. Reasons for physician non-adherence to electronic drug alerts. Medinfo 2004;11(Pt 2):1101-5.
125. Nies, J, Colombet, I, Degoulet, P, Durieux, P. Determinants of success for computerized clinical decision support systems integrated in CPOE systems: a systematic review. AMIA 2006;Annual Symposium Proceedings/AMIA Symposium.:594-8.
126. Langer, E, Piper, A. The Prevention of Mindlessness. Journal of Personality and Social Psychology 1987;53(2):280-287.
127. Butler, B, Gray, P. Reliability, Mindfulness and Information Systems. MIS Quarterly 2006;30(2):1-14.
128. Ferreira, F, Christianson, K, Hollingworth, A. Misinterpretations of garden-path sentences: implications for models of sentence processing and reanalysis. Journal of Psycholinguistic Research 2001;30(1):3-20.
129. Cumin, J. Review Paper. Coding Systems in Health Care. Methods of Information in Medicine 1996;35:273-284.
130. Rosenbloom, ST, Miller, RA, Johnson, KB, Elkin, PL, Brown, SH. Interface Terminologies: Facilitating Direct Entry of Clinical Data into Electronic Health Record Systems. J Am Med Inform Assoc 2006;13(3):277-288.
131. Spackman, K, Campbell, K, Cote, R. SNOMED RT: a reference terminology for health care. Proceedings of the AMIA Annual Symposium 1997:640-4.
132. Rector, A. Clinical Terminology: Why Is it so Hard? Methods of Information in Medicine 1999;38(4):239-252.
133. Bates, DW, Gawande, AA. Improving safety with information technology. New England Journal of Medicine 2003;348(25):2526-34.
134. Garg, AX, Adhikari, NK, McDonald, H, Rosas-Arellano, MP, Devereaux, PJ, Beyene, J, Sam, J, Haynes, RB. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review.[see comment]. JAMA 2005;293(10):1223-38.
135. Kaushal, R, Shojania, KG, Bates, DW. Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review.[see comment]. Archives of Internal Medicine 2003;163(12):1409-16.
136. Miyo, K, Nittami, Y, Kitagawa, Y, K, O. Development of case-based medication alerting and recommender system: a new approach to prevention for medication error. Medinfo 2007;12(Pt 2):871-4.

137. van der Sijs, H, Aarts, J, Vulto, A, Berg, M. Overriding of Drug Safety Alerts in Computerized Physician Order Entry. J Am Med Inform Assoc 2006;13(2):138-147.
138. Hsieh, TC, Kuperman, GJ, Jaggi, T, Hojnowski-Diaz, P, Fiskio, J, Williams, DH, Bates, DW, Gandhi, TK. Characteristics and Consequences of Drug Allergy Alert Overrides in a Computerized Physician Order Entry System. J Am Med Inform Assoc 2004;11(6):482-491.
139. Ko, Y, Abarca, J, Malone, DC, Dare, DC, Geraets, D, Houranieh, A, Jones, WN, Nichol, WP, Schepers, GP, Wilhardt, M. Practitioners' Views on Computerized Drug-Drug Interaction Alerts in the VA System. J Am Med Inform Assoc 2007;14(1):56-64.
140. Nielsen, J. Heuristic evaluation. In: Nielsen, J, Mack, R, editors. Usability Inspection Methods: John Wiley & Sons, New York, NY; 1994.
141. Harrison, MI, Koppel, R, Bar-Lev, S. Unintended Consequences of Information Technologies in Health Care An Interactive Sociotechnical Analysis. J Am Med Inform Assoc 2007;14(5):542-549.
142. HIPAA Administrative Simplification Regulation Text.CFR 164.502, 164.506.February 16, 2006
143. Nielsen, J. Finding usability problems through heuristic evaluation. In: Proceedings of the ACM CHI '92 Conference; 1992; Monterey, CA; 1992. p. 373-380.
144. Marshall, M. Sampling for qualitative research. Family Practice 1996;16(6):522-525.

9. Appendices

9.1 Appendix 1. Modified Computers in Medical Care Survey

I. Demographics

a. Your name: ____________________________

b. Your age: ______________

c. Your gender: ☐ female ☐ male

d. In which area of medicine do you currently specialize (please check only one)?

☐ Cardiology	☐ Hospitalist	☐ Ophthalmology	☐ Surgery, general
☐ Cardiothoracic surgery	☐ Infectious disease	☐ Orthopedics	☐ Urology
☐ Critical care	☐ Nephrology	☐ Otolaryngology	☐ Vascular surgery
☐ Emergency medicine	☐ Neurology	☐ Pulmonary medicine	☐ Other__________
☐ Endocrinology	☐ Neurosurgery	☐ Radiology	
☐ Gastroenterology	☐ OB/GYN	☐ Radiation oncology	
☐ General internal medicine	☐ Oncology	☐ Rheumatology	

II. Computer Experience

a. In a typical week, how many hours do you use a computer hands-on? ____________ hours

b. What kind of training or experience with computers have you had?

☐ Formal course(s) in computer science or related field
☐ Formal medical school training in computers
☐ Formal residency or fellowship training in computers
☐ Formal workshops or conference on computers
☐ Self-guided learning about computers
☐ None

c. On the whole, how sophisticated a computer user are you?

☐ Very sophisticated
☐ Sophisticated
☐ Neither sophisticated nor unsophisticated

☐ Unsophisticated
☐ Very unsophisticated

d. What much experience with your institution's CPOE system (order entry system) have you had?

___________________months

e. With respect to our institution's CPOE system (order entry system) how sophisticated a computer user are you?

☐ Very sophisticated
☐ Sophisticated
☐ Neither sophisticated nor unsophisticated
☐ Unsophisticated
☐ Very unsophisticated

f. To what extent do you personally use a computer for each of the following tasks? Please circle your answer.

1. Never perform this task
2. Perform this task, but never use a computer
3. Sometimes use a computer
4. Often use a computer
5. Always use a computer

Communicating with colleagues	1	2	3	4	5
Obtaining advice on a specific patient's diagnosis or therapy	1	2	3	4	5
Writing (e.g. grants, research papers, articles, teaching material, correspondence)	1	2	3	4	5
Preparing presentations or slides	1	2	3	4	5
Performing statistical analysis on clinical or research data	1	2	3	4	5
Searching the medical literature	1	2	3	4	5
Teaching students and residents	1	2	3	4	5