User Interaction with Computerized Provider Order Entry Systems: A Method for Quantitative Measurement of Cognitive Complexity

By

Denise C. Dinulescu, M.D.

A THESIS

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

Master of Science

April 2007

School of Medicine

Oregon Health & Science University

Certificate of Approval

This is to certify that the Master's Thesis of

Denise C. Dinulescu

"User Interaction with Computerized Provider Order Entry Systems: A Method for Quantitative Measurement of Cognitive Complexity"

Has been approved

[redacted]

Thesis Advisor – David Dorr, M.D., M.S.

[redacted]

Committee Member – Dean Sittig, Ph.D.

[redacted]

Committee Member – Holly Jimison, Ph.D.

[redacted]

Committee Member – Richard Gibson, M.D., Ph.D.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF GRAPHS

LIST OF FIGURES

Acknowledgements

I am very thankful and grateful to the following for providing support in my carrying out this study and for sharing with me the depth of their knowledge.

My thesis committee members, Drs. David Dorr, Dean Sittig, Holly Jimison, and Dick Gibson, who provided invaluable feedback and expertise to my thesis research.

Dr. Bret Fuller, who provided guidance and advice for the data analysis of this study. Dr. Blake Lesselroth, who gave me a crash course in using the VA computerized ordering system. My colleagues in the Medical Informatics program, Adam Wright, who introduced me to the Principal Components Analysis technique, and Hari Krishna Rekapalli and Ravi Teja Bhupatiraju, who provided support in writing the programs for automatic collection of data and computation of certain design metrics. Also, many thanks to Hari for his friendship, continuous support, and many stimulating discussions.

My sister, Dr. Daniela Dinulescu, who encouraged me throughout this pursuit and kindly tolerated my frequent preoccupation with it.

Abstract

BACKGROUND: Overly simplistic or complex user interfaces of Computerized Provider Order Entry (CPOE) systems may impede clinical efficiency and the decision making process by increasing cognitive load and diverting physicians' attention from the clinical task at hand. In order to design better healthcare user interfaces, there is an imperative need for developing new evaluation methodologies to quantify the cognitive complexity of CPOE user interfaces. OBJECTIVES: To propose a method for quantitative measurement of cognitive complexity of CPOE user interfaces; to validate the developed metrics by characterizing and comparing the cognitive complexity of three CPOE systems; to gather preliminary usability data that can be correlated to quantitative estimations of cognitive load of user interfaces. METHOD: A quantitative analysis of cognitive complexity of CPOE user interfaces was performed by computing thirteen design metrics. This analysis was followed by a usability order entry study involving a total of 30 experienced clinician users of the three CPOE systems investigated. Study participants were timed as they entered seven generic orders in each system; various ordering issues were identified. Participants also completed a short survey regarding their computer skills as well as their experience and satisfaction with using CPOE systems. RESULTS: The quantitative evaluation showed a measurable difference in cognitive complexity between the three CPOE user interfaces. Furthermore, the usability study established that participants spent more time and were less satisfied with using an overly complex CPOE system than a system with a lower cognitive complexity score.

Introduction

It is widely recognized that effective evaluation of clinical information systems, especially their user interfaces, is necessary in order to ensure that such systems are well designed to meet the day-to-day clinical needs of users. Conventional outcome-based evaluations are instrumental in providing information about how use of systems affects certain outcome measures, such as morbidity, mortality, cost of health care, and work efficiency, but do not provide feedback for improving the quality of clinical processes [1,2]. Other evaluation methods that are commonly used because of their convenience and richness of data collected are questionnaire-based surveys. Surveys have the disadvantage that are highly subjective and rely on participants' ability to recall and describe their interaction with the systems [2]. The same issues characterize qualitative methods of evaluation such as retrospective focus groups and interviews.

To better understand a system's ability to meet the needs of users, new methods of assessment of medical information systems have been developed. These novel approaches borrow ideas from multiple fields of research including cognitive science, computer science, systems engineering, and usability engineering. The field of **usability engineering** has emerged from "the integration of evaluation methods used in the study of human-computer interaction aimed at providing practical feedback into design of computer systems and user interfaces" [2]. Employing these new evaluation methods can give us a better understanding of how systems can be developed or redesigned to better aid the complex processes of medical reasoning and decision making.

This research study proposes to describe a quantitative approach to the summative evaluation of user interfaces of completed healthcare information systems based on usability testing. The method developed in this study can also be used in the formative iterative evaluation of systems to help developers design more usable and efficient medical software.

Background and Significance

Computerized Provider Order Entry (CPOE) systems are clinical information systems that allow clinicians to enter patient orders directly into a computer rather than on paper. The reasons behind implementing CPOE systems in hospitals are explained by the potential of such systems to reduce the number of preventable clinical errors, streamline clinicians' workflow, standardize the delivery of medical care in US hospitals, and ultimately increase patient satisfaction and decrease healthcare costs. Some of the key factors for successful implementation and acceptance of CPOE systems by clinicians are ease of use and efficiency of systems as well as the design of the user interfaces [3]. Clinical information systems interfaces are inherently complex because of the "multifaceted nature of clinical medicine" [4]. However, poorly designed and overly complex CPOE user interfaces may hinder the users interactions with the systems by placing unnecessary cognitive demands on the working memory of physicians, causing them to focus on problematic aspects of the system instead of higher-order reasoning processes involved in clinical decision making [5]. This diversion of mental focus and the workflow delay it entails may result in some physicians' showing resistance to entering patient orders electronically.

The distributed cognition theory, which gained recognition in human-computer interaction research in recent years, defines **human cognition** as a process of coordinating distributed internal representations stored in users' memory and external representations (e.g., screen layout), effectively constituting an indivisible information-processing system [6]. This paradigm, borrowed from cognitive science, helps explain how complex user interfaces, which serve as external representations for end users, may increase cognitive load making the learning process challenging and time-consuming. A user interface that is characterized by high mental load is thus said to be cognitively complex. A more formal definition of the concept of **cognitive complexity** in human-computer interaction is offered by Rauterberg: "The complexity of the user's mental model of the dialog system is given by the number of known dialog contexts on one hand, and by the number of known dialog operations on the other hand" [7]. In other words, a cognitively challenging system is characterized by the use of many objects or constructs that have multiple relationships to one another. System simplicity would be defined as the exact opposite - a system with few constructs that have a limited number of possible dialog operations.

User interface design metrics, also called usability metrics (Figure 1), can be employed to assess the cognitive complexity or simplicity of user interfaces. **Usability** has been defined by Preece et al. as the capacity of a system to allow users to carry out their tasks safely, effectively, efficiently, and in an enjoyable manner [8]. Constantine described three categories of usability metrics [9]:

- ***Preference metrics*** are subjective evaluations of users regarding the screen design, ease of use, and control of the system;
- ***Design metrics*** can objectively quantify various properties of user interfaces, such as screen information density and layout consistency, and can predict the usability of a completed system.

➢ ***Performance metrics***, such as task completion time and error rate, are objective measures that are usually used to drive the design process of new user interfaces. **Usability testing** refers to evaluation of information systems by employing performance metrics and it involves representative users directly interacting with systems.

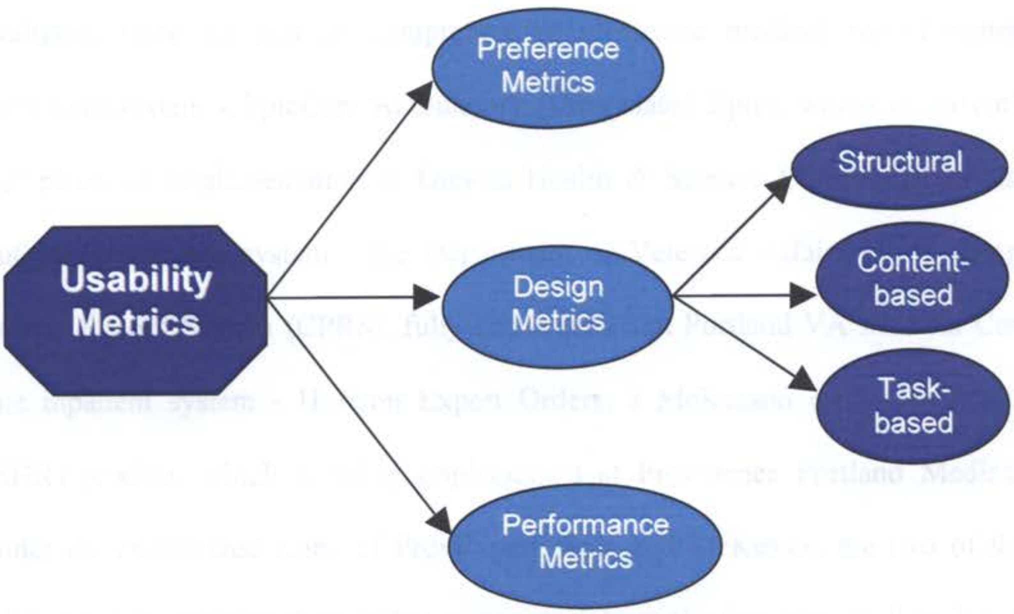


Figure 1 – Usability metrics employed to assess the user interface complexity

The design metric suite includes ***structural***, ***content-based***, and ***task-based*** metrics. **Structural metrics** characterize the static architecture of the interface – number of objects on the screen, configuration of objects, number of colors used, etc. – whereas **task-based metrics** depend on the task performed by users and include measures such as the number of screens in the path, consistency of screen layout within the path, and number of keystrokes and mouse operations performed by users. **Content-based metrics** relate to both the functionality of objects on the screen and their actual meaning for users.

A combination of preference, performance, and well-designed structural and task-based metrics can thus permit not only an objective evaluation of CPOE user interfaces but a comparison of different interface designs as well. By selecting and measuring a variety of design, preference, and performance measures, this research study proposed to investigate the user interfaces of one government and two commercial CPOE systems implemented at major hospitals in the Portland metropolitan area. The CPOE systems evaluated were all part of comprehensive electronic medical record systems: one outpatient system - EpicCare Ambulatory (abbreviated Epic), which is currently in the last phase of implementation at Oregon Health & Science University; one integrated outpatient/inpatient system - the Department of Veterans Affairs (VA) Computerized Patient Record System (CPRS), fully implemented at Portland VA Medical Center; and one inpatient system - Horizon Expert Orders, a McKesson electronic health record (EHR) product, which is being implemented at Providence Portland Medical Center under the customized name of ProvExpert. Epic and McKesson are two of the largest EHR vendors, making their software appropriate evaluation targets. The VA CPRS, a pioneer in CPOE systems, was selected because of its large user base and successful national-wide implementation.

This quantitative evaluation study proposes a series of design metrics that can be used by prospective software purchasers to objectively compare interfaces of various clinical information systems when deciding which system they should select for implementation in their healthcare organization (summative system evaluation). System developers can also employ these quantitative measures to compare different system design proposals and determine the most usable interface design with the optimum cognitive complexity. Furthermore, the metrics can be incorporated directly into the development environment to provide constant feedback to application programmers during the system development phase (formative iterative system evaluation).

A second study outcome consists of the findings of the comparative analysis of the three commercial CPOE user interfaces in terms of their cognitive complexity and usability. These results may indicate which interfaces are characterized by high cognitive load and how that correlates with task efficiency and system ease of use. User interfaces can be redesigned to achieve lower complexity and higher usability scores.

Previous Work

In the last decade, many studies concerned with characterizing the complexity of clinical information systems and assessing its impact on users' behavior have been published. In 2003, a paper by Horsky et al. presented a methodology developed within the theoretical framework of distributed cognition for analyzing the cognitive demands of CPOE user interfaces [5]. As mentioned earlier, distributed cognition is a novel approach that defines cognitive processes as being distributed between users (as internal representations) and computers (as external representations) as opposed to belonging exclusively to the human being. In other words, the information that is required to carry out a task using a computerized system can be either located as an interface object or users can bring it to the task as a piece of knowledge that they possess. This paradigm is especially relevant to the evaluation of clinical information systems because it explains how newly adopted medical technology, if cumbersome, may negatively impact clinical reasoning and diagnosis generation. At the same time, the distributed cognition theory can help us understand how we can improve properties of user interfaces, especially the graphical ones that are inherently rich in external representations, to minimize the cognitive load of users. More important than the availability of external representations is their availability in optimal configurations to facilitate the user-computer interaction and reduce errors.

Horsky's evaluation framework [5] involved a cognitive walkthrough of a CPOE system conducted by system experts who entered clinical orders in the system, followed by a usability study in which users were asked to place the same orders as the experts using the order entry application. The **cognitive walkthrough** is a powerful technique used in cognitive science that can be applied in medicine, in both laboratory and real-world clinical settings, to determine if a clinician with a certain level of system knowledge can successfully complete the sequence of actions required by a task that the system is designed to support. The walkthrough evaluation reveals the relative distribution of resources at every system state identifying the interface dimensions that place considerable demands on the internal resources of users, especially on less experienced users that lack a robust understanding of the system configuration. The usability experiment can identify both types of ordering errors - omission and commission - that users make partly because of inadequate configuration of resources in the interface. The results of this extensive study indicated that the user interface of the commercial CPOE system investigated had suboptimal configuration of external resources and required users to rely heavily on their memory. Consequently, each of the seven clinicians involved in the usability study made ordering errors that were partially due to the system complexity and low usability.

Several qualitative studies of usability testing and inspection of systems were performed by a research team lead by Kushniruk and Patel who have refined their methodological approaches to the evaluation of health information systems for more than a decade [1,2,10-13]. The researchers developed a portable usability laboratory consisting of audio and video recording equipment that could be taken to clinical settings to test users' interaction with computerized patient record systems in the 'thinking-aloud' approach (i.e., users verbalizing their thoughts while using the system). These evaluation studies demonstrated how usability engineering and cognitive task analysis approaches could be used to describe the dimensions of user interfaces (e.g., content and organization of information) that impede human-computer interaction and have an impact on users' cognitive processes, such as data collection, knowledge organization, and reasoning strategies. For example, a recent methodological review published by Kushniruk and Patel categorized the most frequent problems encountered by users while interacting with a clinical information system as "lack of consistency" in functionality of screen objects, "data entry blocked", "response time", "partial matches" in searches for medical terms, and "problem understanding system messages" [2]. Furthermore, the same study showed that 80% of the usability problems with a system can be identified by involving as few as 8-12 subjects [2].

A 2005 literature review by Despont-Gros et al. found that the number of studies related to users' interaction with clinical information systems has increased dramatically in the past 10 years. However, a large number of these studies focused on evaluation of preference and performance variables, such as user satisfaction, user acceptance of information systems, users' success in task performing, and impact of systems on users [14].

In order to design new healthcare interfaces or redesign the existing ones to achieve optimum complexity and usability, there is an imperative need to further evaluate the user interfaces of clinical information systems by using well-developed structural and task-based metrics and make the transition from current qualitative understandings of user interface complexity toward a quantitative representation of this critical interface property.

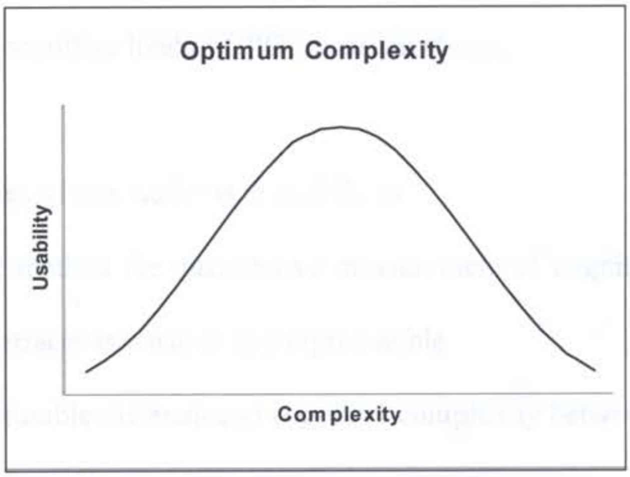
Very few studies related to quantitative evaluation of graphical user interfaces (GUI) have been published. Of these few studies, the most salient is Comber and Maltby's development of a method for quantitative measurement of the complexity of the GUI screen layout [15-18]. Their work was based on Bonsiepe's method for quantitative comparison of degree of order of two typographically designed pages [19]. Bonsiepe concluded that printed pages with better arrangement of components were judged by subjects as more attractive and easier to read.

Comber and Maltby were also inspired by Tullis' study concerned with evaluation of alphanumeric monochromatic screens [20]. Tullis investigated a wide variety of displays (a total of 52 screen formats) by employing six display measures and developed a prediction model to determine the relationship between the display measures and screen usability. Tullis concluded that simpler screen designs were more usable. Interestingly, Comber and Maltby, who compared the interface design of four different Microsoft Windows applications in terms of their usability, argued that users actually preferred screens of intermediate cognitive complexity than screens at either end of the complexity scale [15]. The results of this usability study are presented in Table 1. Although screen 3 had a mid-range value of complexity, 86% of the users who performed a task using this interface did not make any errors. In addition, the total time spent on screen 3 was the shortest and users rated this screen as the best design and the easiest to use.

USABILITY	SCREEN 1	SCREEN 2	SCREEN 3	SCREEN 4
Complexity	156	170	186	228
Error-free	36%	79%	86%	71%
Time	354	290	276	293
Rating	4	7	16	0

Table 1 – Comber and Maltby's summary of usability [15] (Table reproduced with permission)

The authors acknowledged, however, that the unexpected results could be due to the small number of study participants (a total of seven) and limited number of screen formats investigated. Despite these limitations of the study, Comber and Maltby concluded that in the case of GUI screens there was a trade off between usability and complexity, usability being a function of complexity (Graph 1). The highest usability can be achieved at an intermediate value of complexity, which is the optimum complexity of a user interface.



Graph 1 - Relationship between complexity and usability [15] (Graph reproduced with permission)

User interfaces characterized by low complexity do not have too much functionality, so they have a lower usability score. At the other extreme, interfaces that are too complex impede the user-system interaction, so they also show low usability.

Purpose of Evaluation and Research Hypotheses

The objective of this research was to complement the previous qualitative usability studies that identified features in the CPOE user interface that contributed to a greater cognitive load and resulted in greater effort on the part of users. The study focused on quantifying this cognitive load. The research aims were threefold:

- To propose a method for quantitative measurement of cognitive complexity of CPOE user interfaces.
- To validate the developed metrics by characterizing and comparing the cognitive complexity of three commercial CPOE systems.
- To gather preliminary usability data that can be correlated to quantitative estimations of cognitive load of CPOE user interfaces.

The research hypotheses of this study were as follows:

1) The developed method for quantitative measurement of cognitive complexity of CPOE user interfaces is feasible and reproducible.
2) There is a measurable difference in cognitive complexity between the three CPOE user interfaces.

3) Cognitive complexity of CPOE user interface affects users' performance and rating of CPOE ease of use. Specifically, clinicians will spend more time when entering orders into a more complex CPOE system than when placing the same orders into a system with a lower complexity score. Furthermore, clinicians will rate the overly complex system as more difficult to use.

Evaluation Method and Metrics

The research described in this paper extended the formal method for evaluating GUI interfaces developed by Comber and Maltby into a method for evaluating user interfaces of medical information technology. The quantitative evaluation of CPOE user interface consisted of two complementary approaches. First, the Principal Investigator performed a cognitive walkthrough of each CPOE system with the assistance of a physician highly skilled in computer ordering (David Dorr, M.D. for the Epic system, Blake Lesselroth, M.D. for CPRS, and Greg Sicard, M.D. for ProvExpert). Prior to this study, the Principal Investigator has attended training sessions to learn how to use each of the three computerized systems effectively and efficiently.

The cognitive walkthrough involved entering four representative orders on a test patient: a medication, laboratory test, imaging test, and procedure (see the orders marked with an asterisk in Appendix A). This analysis was intended to identify the most efficient, error-free sequence of steps for placing an order in each CPOE system. After the action sequence for completing each clinical task was identified, a quantitative evaluation of the CPOE user interfaces was performed by measuring various design metrics, both structural and task-based metrics. A detailed description of the metrics is provided below in the Design Metrics section. To demonstrate the measurement reproducibility, a succinct independent quantitative evaluation of the three user interfaces was conducted by a second investigator, Dean Sittig, Ph.D.

The next step of the system evaluation was a usability order entry study. The following section describes the process of recruitment of study participants.

Participant Recruitment

For the usability study, a total of 30 experienced clinician users of the CPOE systems (having more than 6 months of experience) implemented at Oregon Health & Science University (OHSU), Providence Portland Medical Center (PPMC), and Portland Veterans Affairs (VA) Medical Center were selected. The participant recruitment focus was on clinical residents with at least three months of experience with the CPOE system and who belonged to groups such as internal medicine and family medicine. Because Internal Medicine residents at OHSU performed clinical duties at the VA hospital as well and used the CPOE systems implemented at both hospitals routinely, these residents (n = 14) were recruited as subjects for both clinical sites. There was an additional Family Medicine resident recruited from OHSU. A number of 15 Internal Medicine residents from PPMC met the eligibility criteria and were also enrolled in the study.

The eligibility criteria intended to recruit both female and male participants who were in the age range of 25–50 years and were using their hospital's CPOE system to place at least 40% of the patient orders. The participant recruitment process first involved getting study approval from the Medical Director of the practice from which physicians were selected. The Medical Director provided a list of physicians' names. An electronic mail (e-mail) was then sent to these physicians to request their participation in the study.

The clinicians who were willing and able to participate were enrolled in the study. The Principal Investigator met with participants on an individual basis at an agreed time when computers were less used on clinical wards in order to not disrupt the practice's workflow. The study required about 15 minutes of participants' time. Participants were awarded a $5 Starbucks gift certificate at the conclusion of the study. Funding was provided from an OHSU discretionary account.

The Survey Instrument

Limited demographic data of participants was collected as part of the usability study – age, gender, medical specialty, and institution where residents practiced medicine. Participants were also asked to complete the following short survey regarding their computer skills as well as their experience and satisfaction with using CPOE systems.

1. How would you rate your computer skills on a scale of 1-10 (1 = never used, 10 = expert)?
2. What kind of tasks do you use computers for? Check all that apply: e-mail, reviewing patient results, entering patient notes, computerized provider order entry (CPOE), research, leisure activities.
3. How long have you been using the CPOE system implemented at this hospital?
4. How would you rate the ease of use of this CPOE system on a scale of 1-5 (1 = very difficult, 5 = very easy to use)?
5. Please estimate the percentage of clinical orders you enter into the CPOE system as opposed to on paper.

6. Do you have prior experience with any other CPOE systems? If your answer is yes, please specify the other CPOE systems you have used and rate their ease of use from 1 to 5 (1 = very difficult, 5 = very easy to use).

The study complied with the Investigative Review Board (IRB) requirements from OHSU, PPMC, and the VA Medical Center, and approval to proceed was obtained. At the time the collected data were analyzed, data were anonymous and aggregated.

Procedure

Residents recruited from the three clinical sites (OHSU, Providence, and the VA Medical Center) were given a list of seven representative patient orders (see Appendix A) and were asked to enter these orders in the CPOE system that they used on a test patient (no actual clinical data was viewed or sent). The seven clinical orders were carefully designed so that clinicians would be able to enter them in both outpatient and inpatient CPOE systems. Also, the selected orders matched closely the preconfigured orders available in each system.

The study participants were observed while entering and signing the orders. The time it took each participant to successfully submit these orders was recorded with a stop watch and the ordering issues encountered by residents were noted. Participants were not asked to think aloud while entering orders. The task completion time included the time spent by clinicians on backing out of any incorrect paths they chose. To ensure that the task completion time was not affected by an unacceptable response delay of the system, the CPOE system delay was also measured for random tasks.

Design Metrics

The design metrics used to quantify the cognitive complexity of CPOE user interface can be categorized into structural metrics that characterize the interface layout and task-based metrics that are based on aspects of the tasks carried out through the user interface. The CPOE **screen complexity (SC)** was measured by computing eight structural metrics. The overall **cognitive complexity (CC)** included the complexity of each screen with which the users interacted to complete a task and was quantified by computing five task-based metrics.

In this study, the evaluation of the CPOE user interfaces was based on well-tested design principles called usability heuristics, which were proposed by Nielsen in 1994 [21]. Key qualitative features of the user interface were quantified using structural and task-based metrics: esthetics, clutter, simplicity, redundancy, and consistency. Therefore, this study provided a quantitative instantiation of Nielsen's user interface heuristics (Table 2).

QUALITATIVE FEATURES	QUANTITATIVE METRICS
Esthetics	System Order Distribution Order Color Count
Clutter	Information Density
Simplicity	Screen Object Count Weighted Hidden Action Count
Redundancy	Shortcuts Count
Consistency	Modified Display Area % Screen Inconsistencies Count

Table 2 – Quantitative instantiation of Nielsen's usability heuristics

The following structural metrics were employed to characterize the screen complexity:

1. ***Screen Object Count*** represented the total number of actionable and information objects visible on the screen. An **actionable object** was defined, in the context of this research, as any selectable option on the screen, including free-text fields, with which the user could interact to cause a change in the system status. Examples of actionable objects included toolbar icons, radio buttons, free-text fields, vertical and horizontal scroll bars, tabs, etc. An **information object**, as opposed to the actionable object, could not be selected by a mouse click or keystroke but it provided the user with some clinical or system-related information. Some information objects were actually actionable objects that were not enabled for selection but they became selectable on subsequent screens. Examples of information objects were: patient's demographics, patient's vitals, chart headings, medication ordering instructions, etc.

2. ***Information Density*** was defined as the percentage of display area covered by actionable and information objects. A rectangle was drawn around each object and the density index was expressed as the ratio of the sum of rectangle areas to the total screen area.

3. ***Shortcuts Count*** represented the number of redundant actionable objects that were displayed on the screen to offer the user a shorter route to perform a task than the one usually found hidden in pull-down menus or sub-menus. Examples

of shortcuts included tabs, buttons, and toolbar icons that were associated or not with visible action label.

4. ***Screen Inconsistencies Count*** was defined as the number of screen objects that either had identical functionality but were displayed under different labels or invoked a different function but had the same label.

The previous four quantitative metrics were computed for this study to evaluate how much information, including the redundant information, was visible on each screen, how consistently that information was displayed to the user, and how tightly packed the screen characters were.

Entropy has been widely used as a quantitative measure of system disorder in areas such as thermodynamics and information theory. The next two entropy-based metrics, system order and distribution order, investigated the layout complexity of the user interface. System order measured the complexity with respect to the size (height and width) of display objects, whereas distribution order evaluated the complexity with respect to the objects' position (horizontal and vertical) on the screen. The more grouping and alignment a user interface had the lower its entropy was, hence the less complex the interface was.

5. ***System Order*** was determined by classifying screen objects in classes based on common heights and widths. A modified version of Shannon's entropy formula [22], based on the proportion of objects in each class, was used by Comber and Maltby in their studies [16-18] to arrive at a layout complexity figure. The same

formula was employed in this study to quantify the degree of order of the screen layout.

$$C = -N \sum_{i=1}^{n} p_i \log_2 p_i \quad \text{where:}$$

$$N = \sum_{i=1}^{n} n_i$$

$p_i = n_i / n$

C = complexity with respect to the size of the *i*th type element on the screen

N = number of objects in each class (widths or heights)

n = number of classes (number of unique widths or heights)

n_i = number of objects in the *i*th class

p_i = proportion of the *i*th class

6. ***Distribution Order*** was determined by classifying screen objects in classes based on horizontal position (measured from the top of the screen) and vertical position (measured from the side of the screen) and then applying the same modified version of Shannon's entropy formula.

$$C = -N \sum_{i=1}^{n} p_i \log_2 p_i \quad \text{where:}$$

$$N = \sum_{i=1}^{n} n_i$$

$p_i = n_i / n$

C = complexity with respect to the position of the *i*th type element on the screen

N = number of objects in each class (distance from the top or side of the screen)

n = number of classes (number of unique distances)

n_i = number of objects in the *i*th class

p_i = proportion of the *i*th class

The system and distribution order metrics identified four classes of objects for each screen and quantified the complexities of objects' size (height and width) and objects' alignment on the screen (horizontal position and vertical position). The sum of the four complexity figures constituted the **layout complexity** of each screen.

7. ***Weighted Hidden Action Count.*** This structural metric counted the total number of items hidden in pull-down menus and sub-menus, which could potentially be selected by the user to advance to the desired dialog state of the system. The count was weighted by the menu depth (i.e., weight 1 was applied to options from the pull-down, weight 2 to options from the cascading menu opened by selecting a choice from the pull-down menu, and so on).

8. ***Color Count*** constituted the number of colors employed to highlight certain background sections or to make key screen objects more prominent.

The following task-based metrics were computed to assess whether there was a significant difference in the overall cognitive complexity among the CPOE systems.

9. ***Dialog State Count*** was determined by counting the total number of screens in the path identified as the most efficient for entering a patient order in each system. A **dialog state** was defined as the current configuration of the external representations of the distributed resources model. The internal representations

were stored in and retrieved from the user's memory. The Dialog State Count metric can be viewed as an indicator of cognitive navigation effort.

10. ***Pop-up Window Count*** was the number of temporary windows that suddenly appeared when the user selected an actionable object with the mouse or pressed a special function key. Pop-up windows generally contained text-free fields and menus of commands and disappeared from the screen once one of the commands was selected. A special type of a pop-up window was the window that opened when the user selected an option from a pull-down menu.

11. ***Modified Display Area%*** was defined as the sum of percentages of screen area that was changed when transitioning from the main ordering screen to subsequent screens. This is a measure of consistency of screen layout. A well-designed system would have its external representations distributed equally between screens so that they place similar cognitive demands on users.

12. ***Keystroke/Mouse Click Count*** was calculated by counting the total number of user actions that were part of the command sequence needed to carry out each of the computerized tasks. It was assumed that a mouse click included the mouse operation of positioning the cursor on the screen.

13. ***Screen Complexity Sum*** was computed by adding the cognitive complexity scores of all screens in the path.

A summary description of structural and task-based metrics computed for this study is presented in Table 3.

	DESIGN METRICS	DESCRIPTION
STRUCTURAL METRICS	Screen Object Count	Total actionable and information objects visible on a screen
	Information Density	Percentage of display area covered by screen objects
	Shortcuts Count	Total redundant actionable objects that are routes for tasks shorter than the usual ones
	Screen Inconsistencies Count	Total screen objects with either identical functionality but different labels or different functionality but same label
	System Order Distribution Order	Complexity with respect to the size (height and width) of screen objects Complexity with respect to the position (horizontal and vertical) of screen objects
	Weighted Hidden Action Count	Total hidden potential actions in pull-down and cascading menus weighted by the menu depth
	Color Count	Total colors used on each screen
TASK-BASED METRICS	Dialog State Count	Total number of screens required to enter a clinical order
	Pop-up Window Count	Total number of temporary windows that suddenly appear when selecting an actionable object
	Modified Display Area %	Sum of percentages of screen area changed between the main ordering screen and subsequent screens
	Keystroke / Mouse Click Count	Total user actions that are part of the command sequence needed to enter an order
	Screen Complexity Sum	Sum of cognitive complexity scores of all screens in the path

Table 3 – Summary description of the structural and task-based metrics computed for this study

Collection of Design Metric Data

As described in the Evaluation Method section, the Principal Investigator performed a cognitive walkthrough of each CPOE system in order to compute the design metrics, both structural and task-based metrics. The cognitive walkthrough involved entering four representative clinical orders on a test patient - a medication, laboratory test, imaging test, and procedure. The clinical orders were as follows:

1. Nitroglycerin sublingual 0.4 mg subling q5min x 3 prn start today for chest pain
2. CBC with automated differential stat
3. Ultrasound abdomen complete routine, reason: periumbilical abdominal pain
4. EKG now, reason: chest pain

The screens that showed the signing of the orders were not included in the analysis. Design metrics were computed on a total of 64 CPOE screenshots (19 (30%) of Epic, 23 (36%) of ProvExpert, and 22 (34%) of CPRS), which were inserted into Microsoft Office PowerPoint slides. Screenshots were saved after free-text fields were populated (e.g., after the order name was typed in the order search box). Additionally, screenshots were scaled to the size of PowerPoint slides unless they were captures of pop-up windows, which were left at their original size (Figure 2).

A rectangle was drawn around each actionable (black rectangle) and information object (red rectangle). For computing the Modified Display Area % metric, another rectangle (blue) was drawn around the screen area that was changed in transition from the main screen to subsequent screens (Figure 3). Drawing rectangles was done as accurately and consistently as possible by zooming in on the screenshots at 200% magnification and setting spacing between grid lines at 1/24'' inches.

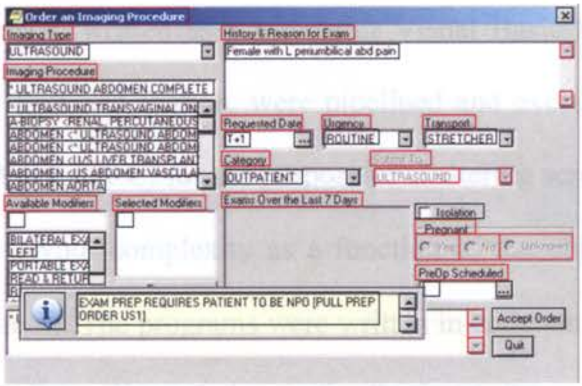


Figure 2 – Rectangles drawn on the screen capture of a CPRS pop-window

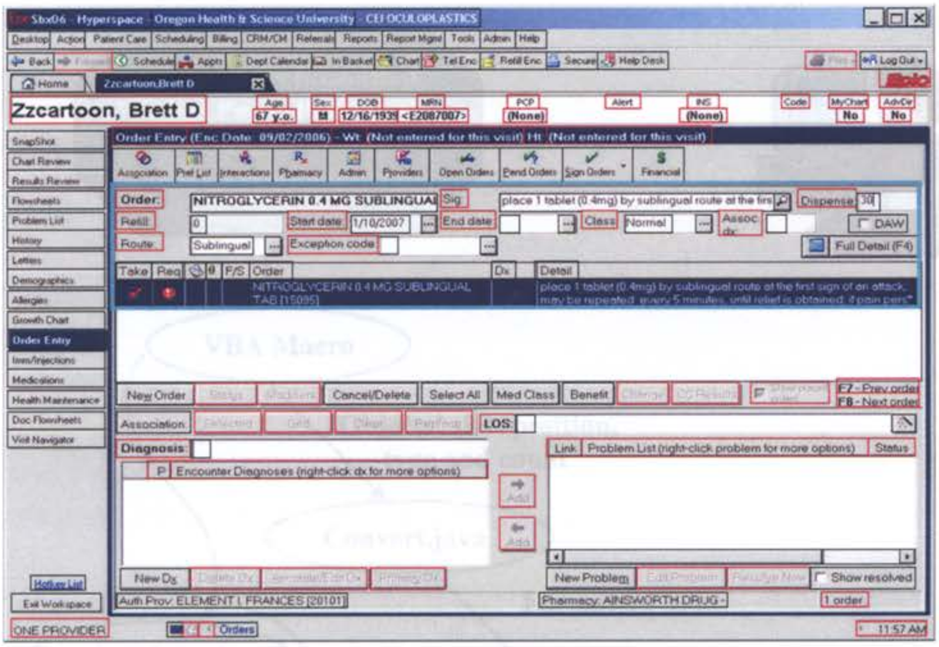


Figure 3 – Rectangles drawn on the main ordering screenshot of Epic

The drawing of the rectangles was done manually using a series of pre-defined rules (see Appendix B). However, the identification of the object type (actionable or information), object counting, determination of object size and position on the screen, and the computation of entropy-based measures (system order and distribution order) were done automatically by a set of programs using the information contained in the drawn rectangles.

This set of programs, written as a Microsoft Visual Basic for Application (VBA) macro, as well as in Java and Python, were pipelined and executed in batch by a Perl script "invoker.pl" (Appendix C) for the purpose of clustering screen objects into classes and calculating their layout complexity as a function of the objects' relative size and screen position (Figure 4). The programs were written in collaboration with Hari Krishna Rekapalli and Ravi Teja Bhupatiraju, both Graduate Research Assistants in the Department of Medical Informatics & Clinical Epidemiology at OHSU.

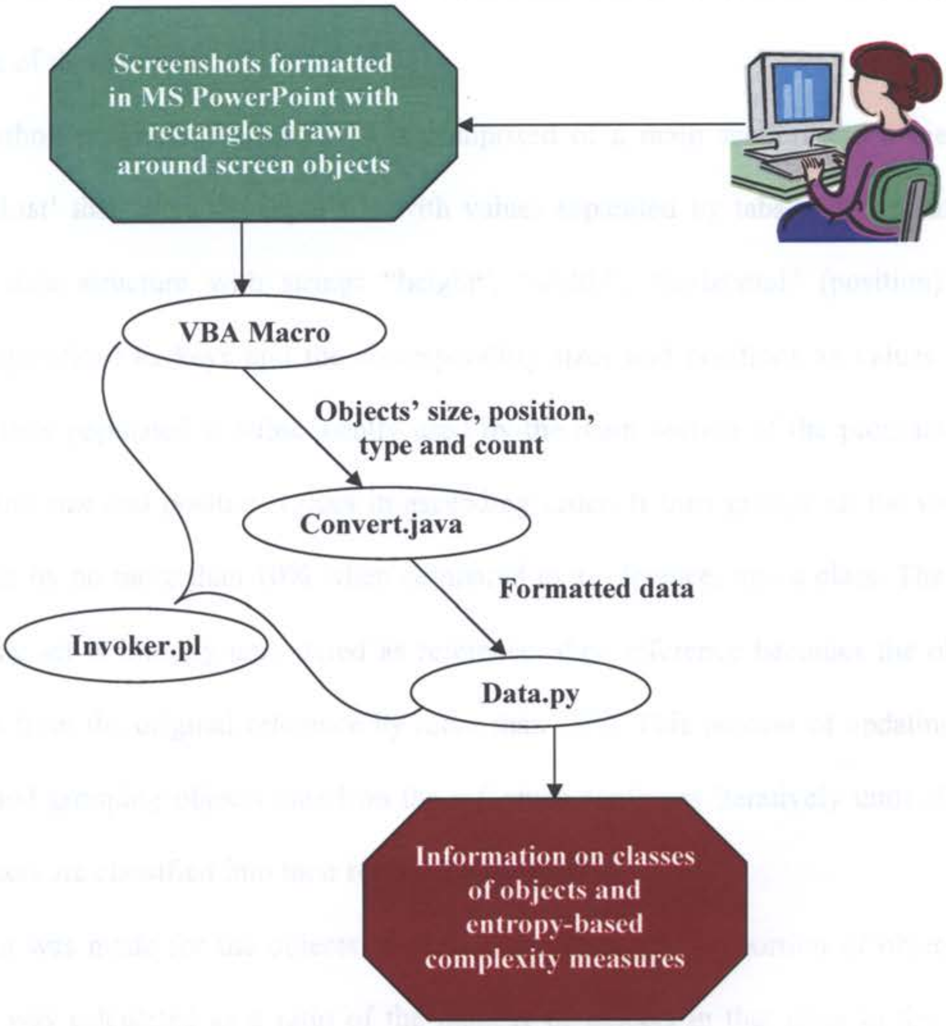


Figure 4 - Execution of sequential programs to determine the size and position of screen objects and compute the entropy-based complexity layout metrics

Microsoft PowerPoint slides that contained the rectangles delimiting the actionable and information objects were fed as input to a VBA macro (Appendix D). The macro automatically identified the size, position, type, and count of the screen objects using the OLE (Object Linking and Embedding) automation, which provided an infrastructure to access and manipulate Microsoft Office objects. The results of the VBA macro were converted by a Java program, "convert.java" (Appendix E) into a tab-separated format. The results were further processed by a Python program "data.py", which clustered screen objects into distinct classes and calculated the entropies for size and positional dimensions of these objects.

The Python program (Appendix F) is comprised of a main section and a method called 'getList' that takes the input file with values separated by tabs and populates a dictionary data structure with strings "height", "width", "horizontal" (position) and "vertical" (position) as keys and the corresponding sizes and positions as values. The dictionary thus populated is subsequently used by the main section of the program that first sorts the size and position values in ascending order. It then groups all the values, which differ by no more than 10% when compared to a reference, into a class. The first object in the set is initially considered as reference; then reference becomes the object that differs from the original reference by more than 10%. This process of updating the reference and grouping objects based on the reference continues iteratively until all the screen objects are classified into their respective categories.

A count was made for the objects in each class. Then, the proportion of objects in each class was calculated as a ratio of the number of objects in that class to the total number of screen objects. Entropy-based complexity measures, system order and distribution order, were calculated using the proportion of objects in each class.

Specifically, the complexity of size (height and width), as well as the complexity of position (horizontal and vertical) were calculated using the modified version of Shannon's entropy formula described in the Design Metrics section.

System Order = $\sum$ (Complexity of height, Complexity of width)

Distribution Order = $\sum$ (Complexity of horizontal position, Complexity of vertical position)

Layout Complexity = $\sum$ (System order, Distribution order)

A spreadsheet was used to compute the Information Density and Modified Display Area % metrics and to record the rest of the design metrics. Having the height and width of screen objects automatically fed into an Excel spreadsheet, information density was calculated as the ratio of the sum of rectangle areas to the total screen area. The sum of percentages of display area that changed between the main ordering screen and subsequent screens was also easily calculated using algebraic functions in Excel.

Data Analysis

Data analysis was conducted using version 13 of the Statistical Package for the Social Sciences (SPSS 13.0 for Windows). Data collected in the usability study involving experienced clinician users of the CPOE systems was processed using one-way analysis of variance (ANOVA). The ANOVA model tested for differences in the means of 11 outcome variables observed on the sample of participants grouped according to a single categorical variable (the CPOE system). The outcome variables were as follows:

- Demographic information about the study participants:

- Age
- Gender
- Specialty

➢ Computer experience of participants:

- Computer skills rated on a Likert-type scale ranging from 1 to 10 (1 = never used, 10 = expert)
- Number of tasks for which participants used computers (out of 6 tasks: email, reviewing patient results, entering patient orders, CPOE, research, leisure activities)

➢ Experience with using CPOE systems:

- Number of months using the current CPOE
- Percentage of orders submitted electronically (as opposed to on paper)
- Number of other CPOE systems used

➢ CPOE preference and performance metrics:

- Ease of use of current CPOE system rated on a scale of 1-5 (1= very difficult, 5 = very easy to use)
- Task completion time (the time to successfully enter seven generic orders in the CPOE system)
- CPOE response delay

When significant results were obtained from the overall F-test (ANOVA), multiple pairwise post hoc comparison tests (Bonferroni-adjusted comparisons) were further carried out to determine which particular group means differ. ANOVA could be used only if the assumption of homogeneity of within-group variances (assessed by the

Levene's test) was met. Because three of the above outcome variables – CPOE experience, percentage of orders submitted electronically, and task completion time – were not normally distributed within groups, a non-parametric alternative to one-way ANOVA given by the Kruskal-Wallis test was carried out. Unlike the analogous ANOVA, the Kruskal-Wallis test does not assume a normal population and tests for equality of population medians among groups. Kruskal-Wallis was followed by the non-parametric equivalent of the post hoc comparison technique, which is the Wilcoxon-Mann-Whitney test.

Structural metrics that quantified the screen complexity were analyzed using the Principal Component Analysis (PCA). The rationale behind selecting the PCA approach was to reduce the data complexity by summarizing a larger number of variables into a smaller number of factors and uncovering any patterns in the set of multivariate data. There were other reasons for employing PCA to analyze this data set:

- PCA did not assume a dependent variable was specified;
- it could be used for strongly correlated variables;
- it standardized the original variables that were on very different scales.

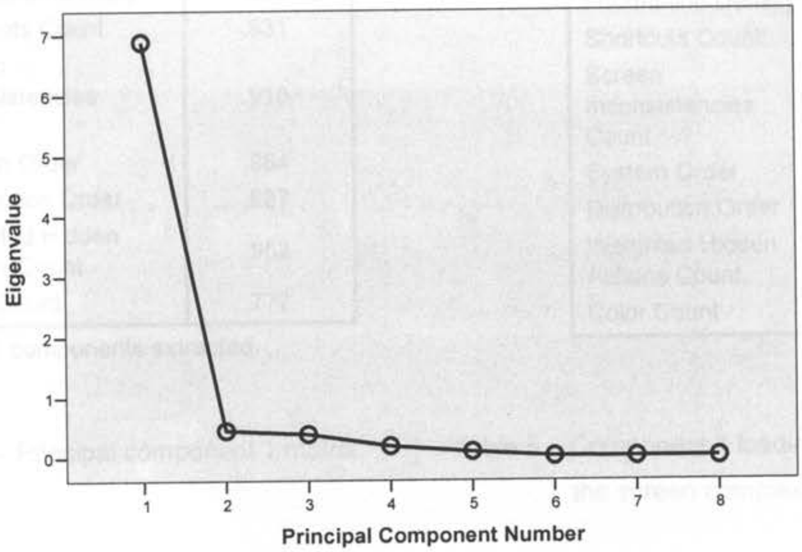
The way PCA works is by seeking a linear combination of variables such that the maximum variance is extracted from the correlated variables resulting in a smaller set of uncorrelated factors. By summarizing data in this manner, subsequent analyses can be greatly simplified. The derived variables (the principal components) that explain between 70 and 90% of the total variation of the original variables are retained, the rest being excluded from interpretation [23]. PCA component scores, which were calculated in SPSS using the factor loading (weight) of each structural metric, represented the weighted screen complexity scores.

Finally, the five task-based metrics that quantified the overall cognitive complexity of a CPOE system were analyzed carrying out a multivariate one-way analysis of variance (one-way MANOVA). This technique examined the differences in the cognitive complexity concept underlying the multiple dependent variables broken down by CPOE system type. As in the case of one-way analysis of variance, MANOVA was followed by a compilation of pairwise post hoc comparison tests (Bonferroni-adjusted comparisons) for each task-based metric.

Results

Quantitative Evaluation – Structural Metrics

The eight structural metrics that quantified the screen complexity were analyzed using the PCA approach. The main aim of this analysis was to identify interesting patterns of screen complexity among the different CPOE systems. A correlation matrix of the data outputted in SPSS showed that interrelations between structural metrics were substantial, most of the correlation coefficients being above 0.8. The color count was the least correlated with the rest of the metrics (correlation coefficient = 0.6). These results suggested that some simplification of the data using the PCA technique would be possible. Of the eight components extracted by PCA, the first principal component explained 86% of the total variance of the observed variables. The scree plot (Graph 2) illustrates how component 1 alone gives an adequate representation of the data, its eigenvalue of 6.9 being above the standard threshold of 1 [23], which helped to determine, along with the percentage of total variance explained, the number of components retained in analysis. Therefore, one-component solution was considered to be appropriate.



Graph 2 – Scree plot for structural metrics

The principal component 1 matrix (Table 4) illustrates a high positive correlation with each of the structural metrics and is simply a weighted average of the variables, thus providing a measure of the overall screen complexity. The second table (Table 5) presents the factor 1 loadings used by SPSS to automatically compute the component scores, which represent the weighted screen complexities. Specifically, the screen complexity was computed as the sum of the products of the standardized scores of structural metrics and the corresponding component loadings. The screen complexity scores were used to further compute the Screen Complexity Sum metric, a task-based measure.

Component Matrix[a]

	Component
	1
Screen Object Count	.971
Information Density	.882
Shortcuts Count	.931
Screen Inconsistencies Count	.918
System Order	.984
Distribution Order	.987
Weighted Hidden Actions Count	.952
Color Count	.777

a. 1 components extracted.

Table 4 – Principal component 1 matrix

Component Score Coefficient Matrix

	Component
	1
Screen Object Count	.141
Information Density	.128
Shortcuts Count	.135
Screen Inconsistencies Count	.133
System Order	.143
Distribution Order	.143
Weighted Hidden Actions Count	.138
Color Count	.113

Table 5 – Component 1 loadings used to calculate the screen complexity scores

Table 6 shows the median differences in structural metrics among CPOE systems. The results are presented by screen type. There were two major types of screens: main full-size screens that look very similar with change of display area of approximately 15-20% and pop-up windows that were 100% different from the main screens. Thus, main screens included the first ordering screen and the similar screens in the path. ProvExpert did not have any pop-up windows. The color shading helps to signify the metric differences. For the main screens, Epic scored the highest in every structural metric. Furthermore, some measurements in Epic were twice or even three times higher than in ProvExpert (e.g., number of objects, information density, shortcuts count, entropy-based metrics). ProvExpert achieved either the lowest scores or similar to the CPRS scores. Lower scores of entropy-based metrics, system order and distribution order, mean that the screen objects were grouped by size and aligned on the screen, horizontally and vertically. The more grouping and alignment a system had the less complex it was.

	MAIN SCREENS			POP-UP WINDOWS	
MAIN SCREENS	Epic	ProvExpert	CPRS	Epic	CPRS
Screen Object Count	155	64	100	38.5	54.5
Information Density	34.4	17.7	17.8	9.2	18
Shortcuts Count	34	2	22.5	2	0
Screen Inconsistencies Count	5	0	0	0	0
System Order	892.8	321.8	540.7	170.8	234.7
Distribution Order	1281.1	466.9	689.3	236.1	329.5
Weighted Hidden Actions Count	257	5	139	4	0
Color Count	9	7	7	4.5	5

Table 6 – Median differences in structural metrics among CPOE main screens and pop-up windows

In the case of pop-up windows, most of the differences were not significant (non-parametric ANOVA). Exceptions were information density, which was twice higher in CPRS (expected result since CPRS is made up of pop-up windows in a proportion of 64% as opposed to 42% in Epic), shortcuts count and hidden action count (which were both 0 for CPRS).

To assess the reliability of the metrics, a second investigator, Dean Sittig, Ph.D. performed a succinct independent quantitative evaluation of the main ordering screen of each CPOE system. Dr. Sittig's results (Table 7 - 2nd evaluation column) were very similar to the 1st evaluation results obtained by the Principal Investigator (metric differences between 0% and 4.1%), except in the case of the system order metric that turned out to be more difficult to reproduce.

	STRUCTURAL METRIC	1ST EVALUATION	2ND EVALUATION	METRIC DIFFERENCE
EPIC	Screen Object Count	153	155	1.3%
	Information Density	33.93	34.27	1%
	System Order	876.60	963.31	9.9%
	Distribution Order	1265.53	1313.03	3.8%
PROVEXPERT	Screen Object Count	62	61	1.6%
	Information Density	17.44	16.98	2.6%
	System Order	313.28	394.04	25.8%
	Distribution Order	443.94	425.87	4.1%
CPRS	Screen Object Count	94	94	0%
	Information Density	17.08	16.74	2%
	System Order	526.08	635	20.7%
	Distribution Order	657.54	659.26	0.3%

Table 7 – Selected structural metric differences between two independent CPOE evaluations

However, differences in system order measurements did not change the fact that Epic had the highest complexity with respect to the size (height and width) of screen objects and ProvExpert the lowest system order, thus having the best grouping of objects based on their size.

Quantitative Evaluation – Task-based Metrics

Task-based metrics were computed to assess whether there was a significant difference in the overall cognitive complexity among the CPOE systems. Levene's test was computed to test the homogeneity of variances assumption [23] and it was found that variances were identical across different systems. As opposed to ANOVA that requires only the homogeneity of variances assumption to be met, in multivariate designs, because there are multiple dependent measures, it is also required that their intercorrelations (covariances) are homogeneous across the cells of the design [23]. Box's test of equality of covariance matrices could not be computed for the task-based metrics because of the low number of data points (12 in total, 4 computations of each metric per system). However, MANOVA is fairly robust to violations of the homogeneity of covariances assumption [23]. Normality for the data set was therefore assumed relying on the robustness of one particular multivariate test, Pillai's trace test, which also has the highest statistical power [23]. As shown in Table 8, four commonly used multivariate tests were computed and all of them were highly significant (p-value < 0.001) meaning that the set of five metrics that measured the cognitive complexity of a CPOE system was affected by the type of system.

Effect		Value	F	Hypothesis df	Error df	Sig.
Intercept	Pillai's Trace	.980	49.585[a]	5.000	5.000	.000
	Wilks' Lambda	.020	49.585[a]	5.000	5.000	.000
	Hotelling's Trace	49.585	49.585[a]	5.000	5.000	.000
	Roy's Largest Root	49.585	49.585[a]	5.000	5.000	.000
CPOE	Pillai's Trace	1.861	16.049	10.000	12.000	.000
	Wilks' Lambda	.001	40.021[a]	10.000	10.000	.000
	Hotelling's Trace	232.136	92.854	10.000	8.000	.000
	Roy's Largest Root	225.713	270.856[b]	5.000	6.000	.000

a. Exact statistic

b. The statistic is an upper bound on F that yields a lower bound on the significance level.

Table 8 – Multivariate tests computed for task-based metrics

Because the overall multivariate test was significant, univariate one-way ANOVAs and pairwise Bonferroni-adjusted comparison tests were examined for each metric to determine if mean values on each outcome variable were also different among systems. Only two task-based metrics – the number of pop-up windows and the sum of screen complexities – were statistically significant at the 0.05 level. The Modified Display Area % metric, although not significant ($p = 0.058$) showed an interesting trend. ProvExpert, the system with the lowest screen complexity score, showed intermediate consistency between the CPOE main screen and subsequent screens, whereas Epic, the most complex system, proved to be the most consistent in the user interface design (see Table 9). CPRS also had a high percentage of screen area that was different between screens. These findings can be explained by examining the number of dialog states necessary to accomplish a task in each system. By increasing the number of dialog states, the percentage of modified area increases as well, especially if all the screens in the path besides the main screen are pop-up windows (as in the case of CPRS). The percentage of area that changed when transitioning from the CPRS main ordering screen to each pop-up window was 100%. However, a clinician interacting with a user interface characterized by lower design consistency may actually prefer such an interface because it requires less manipulation (lower number of keystrokes and mouse clicks) and it takes less time (shorter task completion time) to complete a task. In the usability study, users of CPRS rated their system's ease of use the highest. This suggests that the above assumption may be valid.

	EPIC (E)	PROVEXPERT (P)	CPRS (C)	SIG. (BETWEEN GROUPS)
Dialog State Count	4.8	5.8	5.5	NS
Pop-up Window Count	2	0	3.5	E vs. P (p = 0.003) E vs. C (p = 0.015) P vs. C (p = 0.001)
Modified Display Area %	227.4	313.4	355.7	NS (p = 0.058)
Keystroke / Mouse Click Count	29	23.3	15.8	NS
Screen Complexity Sum	3.7	-2.1	-1.6	E vs. P (p = 0.001) E vs. C (p = 0.001) P vs. C (p = 0.001)

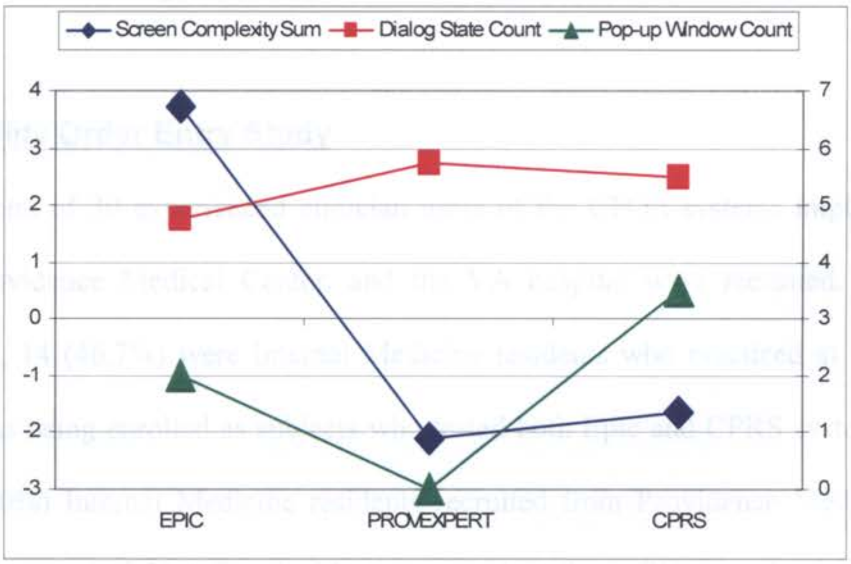
Table 9 – Mean differences in task-based metrics among CPOE systems

Table 9 Legend:

NS = the mean difference was non-significant at the 0.05 level

E = Epic, P = ProvExpert, C = CPRS

Graph 3 shows a comparison of the mean values of three task-based metrics, Screen Complexity Sum, Dialog State Count, and Pop-up Window Count across systems.



Graph 3 – Mean comparison of screen complexity, dialog state, and pop-up window count

This graph aims to answer the following questions frequently asked by designers of clinical information systems:

- Should a system require a user to go through multiple screens of low cognitive complexity or fewer screens that are more complex in order to accomplish a certain task?
- Should the screens following the main ordering screen be mostly pop-up windows or full-size screens that display information unnecessary to the task at hand?

Graph 3 suggests that systems with lower overall screen complexity scores (ProvExpert and CPRS) had a higher number of simpler screens, whereas Epic, which was more complex had fewer but more cognitively challenging screens. This graph also shows that the number of pop-up windows versus the total number of screens in the path did not matter in achieving a lower complexity figure. (ProvExpert system had zero pop-up windows and the lowest screen complexity whereas CPRS, also a less complex system, had the highest mean of pop-up window count, 3.5.)

The Usability Order Entry Study

A total of 30 experienced clinician users of the CPOE systems implemented at OHSU, Providence Medical Center, and the VA hospital were recruited. Of the 30 participants, 14 (46.7%) were Internal Medicine residents who practiced at OHSU and the VA, thus being enrolled as subjects who tested both Epic and CPRS systems. There were 15 (50%) Internal Medicine residents recruited from Providence Medical Center and one additional (3.3%) Family Medicine resident from OHSU (who used only the Epic system). Table 10 presents the summary of the survey data: the mean age of study participants, their gender and specialty distributions, rating of their computer skills, the median CPOE experience and percentage of orders submitted electronically (as opposed to on paper), and mean number of other CPOE systems used.

		EPIC (E)	PROVEXPERT (P)	CPRS (C)	SIG. (BETWEEN GROUPS)
Demographic	Age	30.4	31.3	30.6	NS
	Gender	9 M (60%) 6 F (40%)	8 M (53.3%) 7 F (46.7%)	9 M (64.3%) 5 F (35.7%)	NS
	Specialty	14 IM (93.3%) 1 FM (6.7%)	15 IM (100%)	14 IM (100%)	NS
Computer experience	Computer skills (1 = never used, 10 = expert)	7.5	7.2	7.5	NS
	No. of computer tasks (out of 6)	5.8	5.3	5.8	E vs. P (p = 0.014) P vs. C (p = 0.019)
CPOE experience	CPOE experience (mos.)*	6	7	19.5	E vs. C (p = 0.001) P vs. C (p = 0.001)
	% orders submitted electronically*	100	80	100	E vs. P (p = 0.001) P vs. C (p = 0.001)
	No. of other CPOE systems used	1.4	0.7	1.3	E vs. P (p = 0.008) P vs. C (p = 0.039)

Table 10 – Summary of survey data

Table 10 Legend:

IM = Internal Medicine, FM = Family Medicine

NS = non-significant difference at the 0.05 level

E = Epic, P = ProvExpert, C = CPRS

* = median values reported instead of mean values (variables greatly skewed)

One-way ANOVA carried out for age, gender, specialty, computer skills, and response delay showed no significant difference between the three CPOE systems. However, there were significant differences among systems for all the other outcome variables tested using either parametric or non-parametric methods as described in the previous section. The VA residents used CPRS for a median of 19.5 months as opposed to the OHSU residents who had only 6 months of experience with Epic ($p = 0.001$), and the Providence residents who used ProvExpert for 7 months ($p = 0.001$). It is important to note that a one-month experience entering orders in an outpatient system does not equal a one-month experience entering orders in an inpatient CPOE because the number of orders submitted in the inpatient setting is much higher. There was also a significant difference in the experience with using other CPOE systems than the current one, the Providence residents being less familiar with additional clinical information systems. Furthermore, the Providence residents tended to use computers to accomplish a lower number of tasks (5.3 tasks as opposed to 5.8 tasks for OHSU/VA residents). Because at Providence Medical Center residents were still allowed to submit certain clinical orders on paper, the percentage of orders entered electronically was significantly lower (80%) than at OHSU and the VA (100% at both clinical sites).

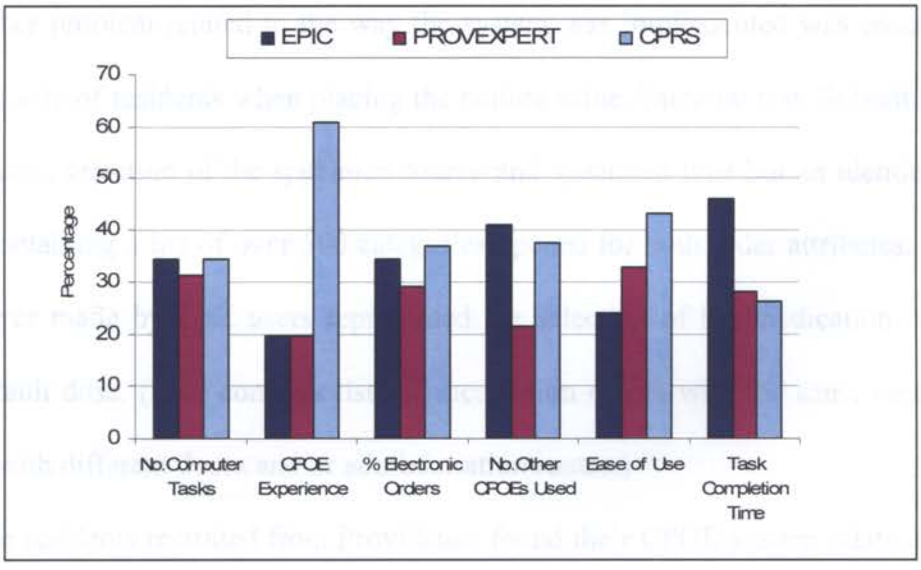
The most important findings of the usability study were related to the users' preference and performance metrics. The preference metric was a subjective evaluation of users regarding the ease of use of the CPOE systems rated on a 5 point scale, with 1 being very difficult to use and 5, very easy. Table 11 presents the results of these metrics: mean ease of use of current CPOE, median task completion time, and mean response delay.

PREFERENCE & PERFORMANCE METRICS	EPIC (E)	PROVEXPERT (P)	CPRS (C)	SIG. (BETWEEN GROUPS)
Ease of use (1 = very difficult, 5 = very easy)	2.4	3.3	4.3	E vs. P ($p = 0.023$) E vs. C ($p = 0.001$) P vs. C ($p = 0.007$)
Task completion time (min.)*	5.3	3.3	3.3	E vs. P ($p = 0.001$) E vs. C ($p = 0.001$)
Response delay (sec.)	1.1	1.3	1.2	NS

Table 11 – Summary of preference and performance metrics

OHSU/VA residents rated Epic somewhat difficult to use (2.4 on average) and CPRS somewhat easy (average score of 4.3), whereas Providence residents considered ProvExpert a system of intermediate ease of use (mean of 3.3). In the case of this metric, each of the three pairwise comparison tests had a highly significant p-value. Interestingly, the performance metric task completion time was also significantly different for Epic (median of 5.3 minutes) versus ProvExpert (3.3 minutes) and CPRS (3.3 minutes). It is important to emphasize that the task completion time was not affected by the system response delay, which was shown to be similar among the three CPOE systems. System response delay was the amount of time that elapsed between a user's mouse click or keystroke and the appearance of the new screen as the result of the user action. Response delay was measured randomly for each CPOE system during the order entry study.

The bar graph below (Graph 4) presents a graphical representation of the significant differences found in the users' CPOE experience, preference, and performance metrics between Epic, ProvExpert, and CPRS.



Graph 4 – Significant differences in outcome variables among CPOE systems

Qualitative Analysis of User Actions

Participants in the usability study were observed while submitting orders into CPOE systems and ordering errors as well as system-related issues were recorded. Overall, Epic users experienced more difficulty placing orders electronically. Some Epic users neglected to change test priority from the default selection *routine* to *urgent*. Others expressed confusion as to which priority option to choose when entering a STAT test, *extreme emergency* or *urgent*. Epic users also commented on the lack of commonly used aliases for certain medications or tests (e.g., *ntg* for nitroglycerin, *abdominal us* for ultrasound abdomen complete test). All 15 Epic users who were enrolled in the study spent considerable time (approximately 40 seconds) trying to find the ultrasound abdomen complete test among Preference List matches, which were displayed on more than three scrolling screens. Residents searched for various keywords, such as us, u/s, ultrasound, abd, abdominal ultrasound, but they were able to locate the imaging test on the long list of order matches only after typing the exact name of the order. This issue was the reason why some residents made the error of selecting the incorrect ultrasound test. Another problem related to the way the system was implemented was encountered by the majority of residents when placing the culture urine, bacterial test. Submitting this order required selection of the specimen source and specimen type but an identical pop-window containing a list of over 500 categories opened for both order attributes. A final type of error made by Epic users represented the selection of the medication with the wrong default dose. (Epic contains lists of medication orders with the same medication name but with different doses and/or administration routes.)

The residents recruited from Providence found their CPOE system relatively easy to use. However, these residents made an ordering error similar to the one made by their

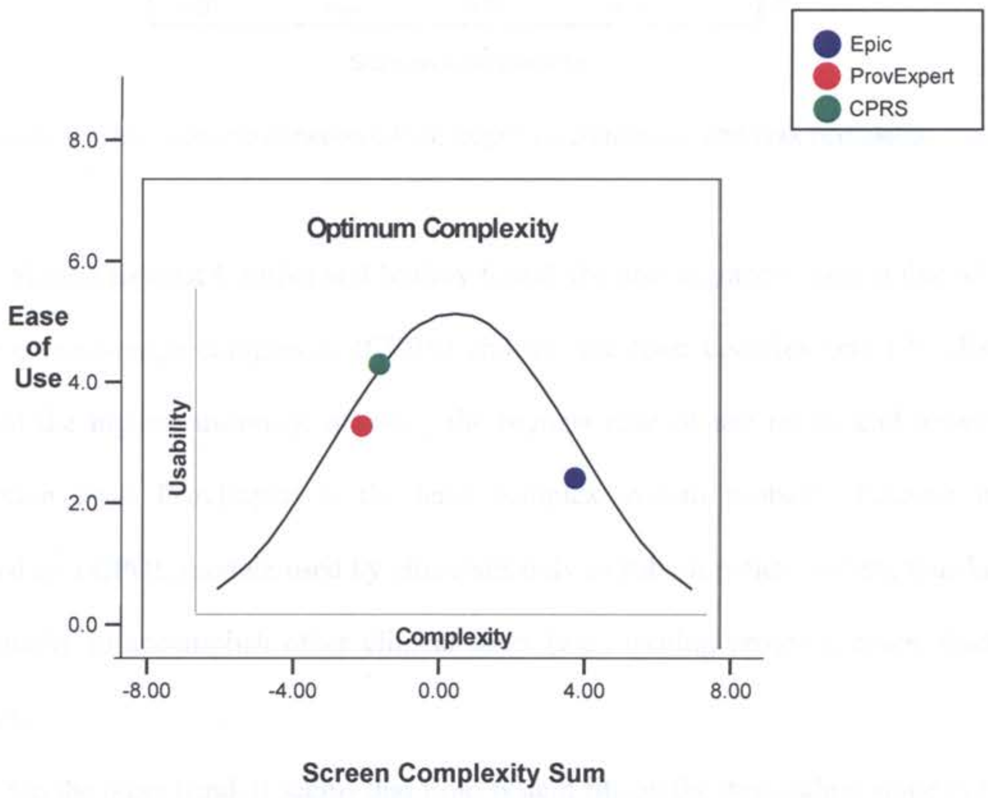
OHSU counterparts – they repeatedly neglected to modify default options for two order parameters: test priority and duration of medication order. Most of the Providence residents realized the error before submitting the order but they made another mistake by not entering the duration of medication in the appropriate text field. (They used the Comments field instead.)

The order submission by CPRS users was not error-free either. Instead of completing the test requested date, some VA residents modified the test priority level because the two order attributes had adjacent free-text fields. Residents also showed some difficulty of finding medication orders under various categories on the extensive medication menu. Therefore, most residents preferred to ignore this categorized menu and use the ALL MEDICATIONS search engine instead.

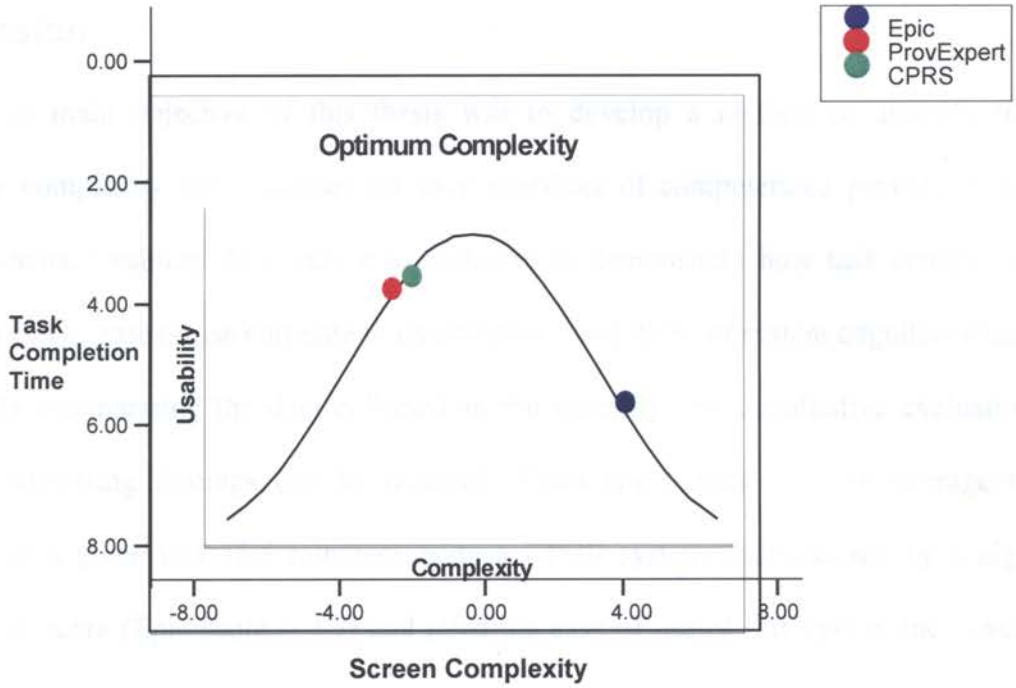
In summary, the CPOE user interface content and layout may affect the clinicians' performance. Epic users in particular seemed to have more difficulty entering orders in their system and spent more time to accomplish the task.

Summary of Results

In the Previous Work section of this thesis, Graph 1, which was originally published by Comber and Maltby in one of their papers concerned with investigation of complexity of Windows applications' interface design [17], illustrated the trade-off between usability and complexity of GUI screens. Comber and Maltby believed that the highest usability could be achieved at an intermediate value of complexity, which was called optimum complexity of a user interface. User interfaces characterized by very low complexity did not have enough functionality, thus achieving a low usability score. At the other extreme, interfaces that were too complex showed low usability as well. Graphs 5 and 6 represent two superimposed graphs, one being the graph created by the two researchers and the other being a SPSS scatterplot made using the thesis data and in which the Screen Complexity Sum metric (on the X axis) was plotted against the ease of use on the Y axis (Graph 5) and task completion time (Graph 6, in which the Y axis is in reverse order, the shortest task completion time corresponding to the highest usability). The three CPOE dots fit relatively well on the bell-shape complexity-usability curve.



Graph 5 – Relationship between CPOE cognitive complexity and ease of use



Graph 6 – Relationship between CPOE cognitive complexity and task completion time

Similar to what Comber and Maltby found, the above graphs suggest that a CPOE system of mid-range complexity (CPRS) and not the least complex one (ProvExpert), achieved the highest usability, meaning the highest ease of use rating and lowest task completion time. ProvExpert is the least complex system probably because it was designed as a CPOE module used by clinicians only to submit patient orders, thus lacking functionality to accomplish other clinical tasks (e.g., writing progress notes, discharge notes, etc.).

On the other hand, it seems that Epic system fits on the descending slope of the complexity-usability curve, meaning that its overly complex user interface places unnecessary cognitive demands on users' memory, which makes the system harder and more time-consuming to manipulate.

Discussion

The main objective of this thesis was to develop a method to quantify the cognitive complexity that characterizes user interfaces of computerized provider order entry systems. Usability data were also collected to demonstrate how task completion time and CPOE ease of use correlate to quantitative estimations of system cognitive load.

By summarizing the data collected in the usability and quantitative evaluation studies, interesting findings can be revealed. Users spent more time on average to accomplish a given task (5.5 minutes) using a CPOE system characterized by a high complexity score (Epic score = 3.7) and rated the ease of use of this system the lowest (2.4) on a 5-point scale, with 1 being very difficult and 5 very easy to use. On the other hand, CPOE systems of similarly low complexity (complexity scores for ProvExpert and CPRS being -2.1 and -1.6, respectively) were subjectively evaluated by users as relatively easy (ProvExpert ease of use of 3.3) or very easy to use (CPRS rating of 4.3). The mean values of task completion time for ProvExpert and CPRS were very similar as well (3.3 and 3.1, respectively). These results illustrate that cognitive complexity of CPOE user interface may affect users' performance and rating of CPOE ease of use. Clinicians spent more time when submitting orders into a more complex CPOE system than when placing the same orders into a system with a lower complexity score. Furthermore, clinicians rated the overly complex system as more difficult to use.

The study participants did not differ in age, gender, specialty, or computer skills. However, residents recruited from Providence Medical Center had a more limited experience with using computers in general and CPOE systems in particular, regarding not only the number of systems used in the past but also the percentage of patients orders submitted in their current system. Moreover, Providence data included two outliers representing two residents who were clearly opposed to using both computers and clinical information systems. For this reasons, it was expected that Providence residents would have an increased task completion time and decreased ease of use rating. That was not the case as ProvExpert achieved usability figures similar to the CPRS scores, which were the best among the three systems. Therefore, it can be argued that the VA residents' extensive CPOE experience (24 months on average) was not the only factor that contributed to achieving the best task completion time and the best rating of system ease of use. Another factor might be the design of user interface that proved to have low cognitive complexity.

The high complexity of the Epic user interface could be explained by examining the high values of each of the eight structural metrics computed. For example, the information density index for Epic was exactly twice as much as the index for the other two systems. Information density is considered in the literature to be one of the main factors that affect the interaction between user and computer interface. Danchak proposed that display loading should not exceed 25 percent [24]. This proposed guideline was based on his analysis of cathode-ray tube (CRT) displays that were judged by users as well-designed and that showed an information density of 15 percent. On the other hand, NASA proposed for the design of the Spacelab module display of the Space Shuttle that the information density not to exceed 60 percent of the total available character area [25]. Modern clinical information systems have mostly graphic user interfaces, which are inherently complex as they reflect the complexity of clinical data and of the medical field in general. For this reason, it was expected that CPOE user interfaces would show a screen density level falling between the two guidelines presented - that is, within the range of 25-60 percent. It turned out that an information density index of approximately

18 percent (ProvExpert index = 17.7, CPRS index = 17.8) would make a CPOE system have low complexity and high usability.

Another distinctive feature of Epic was the high number of shortcuts displayed. Shortcuts are generally considered very useful for experienced users who appreciate a flexible system that allows them to accomplish a task in more than one way. However, shortcuts offered for too many tasks and more than one shortcut for the same task contribute to a high screen object count and may negatively affect the users' interaction with the system.

The number of hidden actions in pull-down and cascading menus in Epic was also significantly higher (count = 257) when compared to ProvExpert and CPRS (counts of 5 and 3, respectively). Most of these hidden menus contained extensive lists of options that may have placed additional cognitive demands on the users' memory. Furthermore, selection of an item in the pull-down menu required multiple user actions (click the menu, scan the item list, and select item) and thus more time than selection of a visible screen object.

A number of screen colors higher than 7 proved to contribute to high screen complexity. The count of user interface inconsistencies, either inconsistencies in object labels or functionalities, was not particularly high in Epic (only 5 inconsistencies could be detected) but it was higher than in the other two systems that actually did not have any screen inconsistencies. Finally, the degree of order of the Epic screen layout was almost three times lower than that of ProvExpert or CPRS layout. Despite its high screen object count, Epic's layout complexity could have been reduced by using fewer fonts, grouping objects by size (width and height), and aligning them better on the screen (horizontally or vertically).

Of the task-based metrics computed for this study, only the keystroke/mouse click count was relatively high in Epic. Regarding the number of dialog states, Epic system was designed in such manner that in order to accomplish a task, users needed to interact with only a few screens but very complex ones. It turned out that to achieve low system complexity it was necessary to reverse the relationship between the screen count and screen complexity. However, it did not matter whether the sequence of screens included multiple pop-up windows or none at all. As discussed in the Results section, these findings are particularly important for developers of clinical information systems.

Limitations of the Study

Due to time constraints, the scope of this quantitative study was limited to evaluation of three CPOE systems. Also, computation of content-based metrics was beyond the scope of the study. The results of this study can be used for further research to develop a regression model with the complexity metrics as its independent parameters and a usability measure, for example, time it takes to complete entering orders in the system, as the dependent variable. This regression model can more accurately identify the metrics that are important in predicting performance of a CPOE system. To increase its external validity and generalizability, the model needs to be extended so that more information systems are included in evaluation.

This research involved comparison of cognitive complexities of three relatively different CPOE systems – one outpatient, one inpatient, and one integrated outpatient/inpatient – that were in different phases of implementation, one of the systems being fully implemented for several years, which had an effect on the content developed in the system and its layout. Also, the study groups were different in:

- the level of system use – more orders submitted in an inpatient CPOE system;
- experience with the currently used CPOE – VA residents had extensive CPRS training;
- experience with computers and other CPOE systems – Providence residents were less experienced.

Another limitation of the study was that task-based metrics were computed for the sequence of screens obtained by entering four representative orders in each system. Due to the low number of data points for these metrics, the Principal Components Analysis approach could not be employed to compute an overall system complexity figure. Only complexity scores for each CPOE screen were calculated. A more comprehensive quantitative evaluation of the task-related cognitive complexity may result in additional interesting findings.

Future Research

There are many interesting research opportunities involving the computation and use of the design metrics developed in this pilot study. Further research needs to be conducted to provide additional empirical validation of the design metrics. Thus, more clinical information systems that have matching characteristics must be evaluated to increase external validity and generalizability of the study. For the proposed method of quantifying cognitive complexity of user interfaces to be practically applied to an extended system evaluation, design metrics need to be incorporated directly into the development environment to be computed automatically. This way, system designers will be able to receive objective on-the-fly feedback about the complexity and usability of the user interface and modify it accordingly to achieve the optimal balance between the two outcome variables. Furthermore, an evaluation module added to the development environment will allow computation of a usability index from the complexity-usability relationship illustrated by this study. A usability index along with a cognitive complexity score can be used to characterize the user interface of a clinical information system influencing the system selection of prospective software purchasers.

It would also be interesting to conduct a comparative study that includes both novice and experienced CPOE users to investigate whether these two groups of users perceive the screen display differently. The hypothesis needed to be tested would be that expert users group and recall screen objects according to their functionality paying attention only to objects needed for the task at hand, whereas novice users notice all the screen objects exhibiting difficulty selecting the right action on the screen. This experiment can employ various content-based metrics (e.g., similarity metrics currently used to annotate images) to quantify the functionality of screen objects and their actual meaning for users. Additional research is also needed to investigate whether the users' process of learning the user interface navigation is affected by the system complexity. Such study may answer the question "What is the optimum system complexity that enables users to learn how to use a system effectively?" Achieving optimum complexity may mean that aligning objects on the screen to reduce the display disorder is not as important for the learning process as grouping objects based on their functionality.

Conclusions

The significance of this pilot study is that it proposed a series of design metrics for quantifying the cognitive complexity of CPOE user interfaces. Results showed that clinicians spent more time submitting orders into a more cognitively challenging CPOE system. Also, users rated the most complex system as more difficult to use. The study had two main goals, which were met: to show a measurable difference in cognitive complexity among CPOE user interfaces using the proposed design metrics; and to demonstrate that the computation of the metrics was feasible and reproducible. However, of the thirteen design metrics there was one structural metric, system order, which turned out to be more difficult to reproduce. The complexity-usability relationship described in the literature for the graphical user interfaces appeared to be valid for the CPOE systems evaluated. A system of mid-range complexity achieved the highest usability. Future research needs to be conducted to demonstrate the generalizability of this pilot study.

References

1. Kushniruk AW, Patel VL, Cimino JJ. Usability testing in medical informatics: cognitive approaches to evaluation of information systems and user interfaces. Proc AMIA Symp 1997:218-22.
2. Kushniruk AW, Patel VL. Cognitive and usability engineering methods for the evaluation of clinical information systems. J Biomed Inform 2004;37(1):56-76.
3. Sittig DF, Stead WW. Computer-based physician order entry: the state of the art. J Am Med Inform Assoc 1994;1(2):108-23.
4. Horsky J, Kaufman DR, Patel VL. The cognitive complexity of a provider order entry interface. Proc AMIA Symp 2003:294-8.
5. Horsky J, Kaufman DR, Oppenheim MI, Patel VL. A framework for analyzing the cognitive complexity of computer-assisted clinical ordering. J Biomed Inform 2003;36(1-2):4-22.
6. Wright PC, Fields RE, Harrison MD. Analyzing human-computer interaction as distributed cognition: the resources model. Human-Comput Interact 2000;15(1): 1-41.
7. Rauterberg M. A method of a quantitative measurement of cognitive complexity. In: Van der Veer G, Tauber M, Bagnara S, Antalovits M, editors. Human-computer interaction: tasks and organisation. Roma: CUD-Publ.; 1992. p. 295-307.
8. Preece J, Rogers Y, Sharp H, Benyon D, Holland S, Carey T. Human-computer interaction. New York: Addison-Wesley Publishing Company; 1994.
9. Constantine LL. Visual coherence and usability: a cohesion metric for assessing the quality of dialogue and screen designs. In: Proceedings of the 6th Australian

Conference on Human-Computer Interaction (OZCHI 1996). Hamilton, New Zeeland: IEEE Computer Society Press; 1996. p. 115-21.

10. Patel VL, Kushniruk AW, Yang S, Yale JF. Impact of a computer-based patient record system on data collection, knowledge organization and reasoning. J Am Med Inform Assoc 2000;7(6):569-85.

11. Kushniruk AW, Kaufman DR, Patel VL, Levesque Y, Lottin P. Assessment of a computerized patient record system: a cognitive approach to evaluating an emerging medical technology. MD Comput 1996;13(5):406-15.

12. Kushniruk AW, Patel VL, Cimino JJ, Barrows RA. Cognitive evaluation of the user interface and vocabulary of an outpatient information system. Proc AMIA Symp 1996:22-6.

13. Kaufman DR, Patel VL, Hilliman C, Morin PC, Pevzner J, Weinstock RS, et al. Usability in the real world: assessing medical information technologies in patients' homes. J Biomed Inform 2003;36(1-2):45-60.

14. Despont-Gross C, Mueller H, Lovis C. Evaluating user interactions with clinical information systems: a model based on human-computer interaction models. J Biomed Inform 2005;38(3):244-55.

15. Comber T, Maltby JR. Evaluating usability of screen designs with layout complexity. In: Proceedings of the 5th Australian Conference on Human-Computer Interaction (OZCHI 1995). Wollongong, Australia; 1995. p. 175-9.

16. Comber T, Maltby JR. A formal method for evaluating GUI screens. In: Proceedings of the 5th Australasian Conference on Information Systems (ACIS 1994). Melbourne, Australia; 1994. p. Group F.

17. Comber T, Maltby JR. Investigating layout complexity. In: Proceedings of the 2nd International Workshop on Computer-Aided Design of User Interfaces (CADUI 1996). Namur, Belgium: Presses Universitaires de Namur; 1996. p. 209-29.

18. Comber T, Maltby JR. Screen complexity and user design preferences in Windows applications. In: Proceedings of the 4th Australian Conference on Human-Computer Interaction (OZCHI 1994). Melbourne, Australia; 1994. p. 133-7.

19. Bonsiepe G. A method of quantifying order in typographic design. J Typograph Res 1968;2(3):203-20.

20. Tullis TS. A system for evaluating screen formats: research and application. In: Hartson HR, Hix D, editors. Advances in human-computer interaction. Volume 2. Norwood, NJ: Ablex; 1988. p. 214-86.

21. Nielsen J. Usability engineering. Boston: AP Professional; 1994.

22. Shannon CE, Weaver W. The mathematical theory of communication. Urbana, IL: University of Illinois Press; 1949.

23. Landau S, Everitt BS. A handbook of statistical analyses using SPSS. London: Chapman & Hall/CRC; 2004.

24. Danchak MM. CRT displays for power plants. Instrumentation Technology 1976;23(10):29-36.

25. NASA. Spacelab display design and command usage guidelines (Report MSFC-PROC-711A). Huntsville, AL: George C. Marshall Space Flight Center; 1980.

Appendix A: Representative Clinical Orders

MEDICATIONS:

1. Nitroglycerin sublingual 0.4 mg subling q5min x 3 prn start today for chest pain *
2. Miconazole 2% topical cream topical bid x 10 days start today
3. Ciprofloxacin 500 mg po bid x 10 days start today

LABORATORY TESTS:

4. CBC with automated differential stat *
5. Culture urine, bacterial routine

IMAGING TEST:

6. Ultrasound abdomen complete routine, reason: periumbilical abdominal pain *

PROCEDURE:

7. EKG now, reason: chest pain *

Note: The orders marked with an asterisk were used to conduct the cognitive walkthrough of each CPOE system and perform the computation of design metrics.

Appendix B: Rules for Drawing Rectangles around Screen Objects and Computing Design Metrics

- The entire name of a clinical order, including the order code, was considered one object (specifically, actionable object if it could be selected for further editing).
- Where a selectable (actionable) area of a tab, toolbar options, or menu bar had characters or characters with icons, the entire area was selected.
- Every screen option that could be selected by keystroke or mouse click, one or double click, and left or right click, was considered an actionable object.
- Actionable objects that were not enabled on a screen were considered information objects on that particular screen but once they became active (available for selection), they were counted as actionable objects.
- Although free-text fields were counted as actionable objects, they were not included in the calculation of information density if they were left blank. Therefore, a rectangle with the same height as the height of the free-text field and with a width small enough to not considerably affect the computation of the information density metric was drawn on top of each blank free-text field in order to be included in the automatic detection of actionable objects (see explanation of object automatic detection in the next section.)
- Each sentence (delimited by colon or semicolon) included in ordering instructions was considered one information object.
- A screen label and the data value associated with the label were considered two different information objects. For example, "Provider: Smith, John M." included 2

information objects, one was the label "Provider" and the other, the full name of the provider.

- As a general rule, the extent to which a rectangle was drawn around a string of words on the screen was determined based on semantics. For example, "enter comments for this order (optional)" was considered one information object because each word of the string taken separately could not provide the user with the same meaning as the entire string.
- To compute the Weighted Hidden Action Count, only the actionable options from pull-down and cascading menus were counted.
- A particular type of shortcuts in a CPOE user interface was considered the right-click menu frequently used by clinicians to edit, discontinue, renew or sign selected clinical orders.
- To compute the Modified Display Area % metric, the percentage of area that changed when transitioning from the main ordering screen to a pop-up window was assumed 100%. The estimation of this change was done by careful visual inspection of the two screens.

Appendix C: Perl Script for Invocation and Batch Processing of Programs

```
#!/usr/bin/perl

$a = system("java convert");
$b = system("python data.py");
#$c = system("del VBAnotconverted.txt");
#$d = system("del final_input.txt");

if (($a > -1) and ($b > -1) and ($c > -1) and ($d > -1))
{
	print "Please check your results in final_output.xls.\n";
}
else
{
	print "$a $b $c\n";
}
```

Appendix D: VBA Macro for Accessing and Manipulation of Microsoft Office PowerPoint Objects

```
Sub Test()

    Dim TX, TY As Integer

    Dim cur_slide As Slide
    Set cur_slide = Application.ActivePresentation.Slides(76)

    Dim c1, c2 As Integer
    Dim s As Integer

    hFile = CurDir & "\VBAnotconverted.txt"
    fnum = FreeFile()
    Open hFile For Output As fnum

    Write #fnum, ; "Height Width Horizontal Vertical";
    Write #fnum,
    For s = 1 To cur_slide.Shapes.Count
      Dim cur_shape As Shape
      Set cur_shape = cur_slide.Shapes.Item(s)
      With cur_shape
        If cur_shape.Type = 1 Then

          If .Line.ForeColor = 0 Then
           c1 = c1 + 1
             Write #fnum, 0; Round((.Height / 72), 2); Round((.Width / 72), 2);
             Round((.Left / 72), 2); Round((.Top / 72), 2)
          Else
           c2 = c2 + 1
```

```
            Write #fnum, 1; Round((.Height / 72), 2); Round((.Width / 72), 2);
            Round((.Left / 72), 2); Round((.Top / 72), 2)
          End If
        End If
      End With
    Next s

    Write #fnum, "Black-objects"; c1
    Write #fnum, "Other-Color-objects"; c2

    Close #fnum
End Sub
```

Appendix E: Java Code for Data Formatting

```
import java.util.*;
import java.io.*;

class convert
{
        public static void main(String[] args)
        {
        try
        {
    BufferedReader br = new BufferedReader(new FileReader("VBAnotconverted.txt"));
      BufferedWriter bw = new BufferedWriter(new FileWriter("final_input.txt"));
                StringTokenizer str = null;
                String t = null;

                //bw.write("Height\tWidth\tHorizontal\tVertical");
                //bw.newLine();

                while( ( t = br.readLine() )!= null)
                {
                        if(t.startsWith("0"))
                        {
                                str = new StringTokenizer(t,",");
                                str.nextToken();
                                while(str.hasMoreTokens())
                                {
                                        bw.write(str.nextToken()+"\t");
                                }
                                bw.newLine();
                        }
```

```
                else if(t.startsWith("1"))
                {
                        str = new StringTokenizer(t,",");
                        str.nextToken();
                        while(str.hasMoreTokens())
                        {
                                bw.write(str.nextToken()+"\t");
                        }
                        bw.newLine();
                }
                else if( (t.startsWith("\"B")) || (t.startsWith("\"O")))
                {
                        bw.newLine();
                        str = new StringTokenizer(t,",");
                        bw.write(str.nextToken()+":");
                        bw.write(str.nextToken());
                        bw.newLine();
                }
        }
        br.close();
        bw.close();

    } catch(Exception e) { System.out.print(e); }

    } //end of main
} //end of class
```

Appendix F: Python Program for Computing Entropy-based Metrics

```
import sys
import glob
import os.path
import math

def getList(fileName):

      file = open(fileName,'r')
    height = []
    width=  []
    horizontal = []
    vertical  = []
      for line in file:
            fields = line.split()
            if len(fields) == 4:
                  height.append((float(fields[0])))
                  width.append((float(fields[1])))
                  horizontal.append((float(fields[2])))
                  vertical.append((float(fields[3])))

      file.close()
         Dx    =    {"Height":height,    "Width":width,    "Horizontal":horizontal,
         "Vertical":vertical}
      return Dx

myDx = getList("final_input.txt")
out = open("final_output.xls", "w")
file = open("final_input.txt",'r')
```

```
for line in file:
        if (line.startswith("\"B") or line.startswith("\"O")):
                out.write("%s\n"%line)
file.close()

for tag in myDx.keys():

   hlist = myDx[tag]
   sortedList = []
   for el in hlist:
        sortedList.append(el)

   sortedList.sort()
   classesDx = {}
   listDx = {}
   className = 1
   number1 = sortedList[0]
   classStatsDx = {}

   for number2 in sortedList :
        if ( number2 > (number1 * 1.1) ) :
             className+=1
             number1 = number2
        classesDx.setdefault(className,[]).append(number2)
        listDx[number2] = className

   out.write("%s \t Class Name\n" %tag)
   for el in hlist:
        className = listDx[el]
        out.write("%.2f \t %d \n" %(el,className))

   for className in classesDx.keys():
```

```
        scores = classesDx[className]
        probability = float(len(scores))/float(len(sortedList))
        classStatsDx[className] = probability

    entropy = 0
    netProb = 0.0
    netNumOfObj = 0
        out.write("\nClass Name \t Number of Objects in each Class \t Probability \t
        p*log(p)\n")

    for className in classStatsDx.keys():
        probability = classStatsDx[className]
        prob = probability
        netProb += probability
        numOfObjects = len(classesDx[className])
        netNumOfObj += numOfObjects
        ent1 = probability * math.log(probability,2) * (-1.0)
        entropy+=  ent1
        out.write("Class %d \t %d \t %.2f \t %.2f\n" % (className,numOfObjects,prob,(-
1*ent1)))

    out.write("Totals \t %d \t %.2f \t %.2f\n\n" %(netNumOfObj,netProb,(-1*entropy)))
    entropy *= float(len(sortedList))
    out.write("Entropy = %.2f\n\n\n" % entropy)

out.close()
```