

Design of Electronic Medical Record User Interfaces: A Matrix-Based Method for Improving Usability

Kushtrim Kuqi, MS^{1,*}; Tim Eveleigh, DSc¹; Thomas Holzer, DSc¹;
Shahryar Sarkani, PhD¹; James E. Levin, MD, PhD² and
Rebecca S. Crowley, MD, MS³

¹*Department of Engineering Management and Systems Engineering, The George Washington University, Washington DC, USA*

²*Children's Hospital of Pittsburgh, University of Pittsburgh Medical Center, Pittsburgh, PA, USA*

³*Department of Biomedical Informatics, University of Pittsburgh School of Medicine, Pittsburgh, PA, USA*

Submitted December 2012. Accepted for publication April 2013.

ABSTRACT

This study examines a new approach of using the Design Structure Matrix (DSM) modeling technique to improve the design of Electronic Medical Record (EMR) user interfaces. The usability of an EMR medication dosage calculator used for placing orders in an academic hospital setting was investigated. The proposed method captures and analyzes the interactions between user interface elements of the EMR system and groups elements based on information exchange, spatial adjacency, and similarity to improve screen density and time-on-task. Medication dose adjustment task time was recorded for the existing and new designs using a cognitive simulation model that predicts user performance. We estimate that the design improvement could reduce time-on-task by saving an average of 21 hours of hospital physicians' time over the course of a month. The study suggests that the application of DSM can improve the usability of an EMR user interface.

Keywords: design structure matrix, electronic medical record, usability inspection, system modeling

1. INTRODUCTION

Healthcare provider dissatisfaction, loss of productivity, and medication related errors are often directly associated with usability issues present in current Electronic Medical Record (EMR) systems [1-3]. The International Organization for Standardization (ISO) 9241-11 standard defines usability as “*the effectiveness, efficiency, and satisfaction with which the users can accomplish their tasks in the intended context of product use*” [4].

*Corresponding author: Kushtrim Kuqi, Department of Engineering Management and Systems Engineering, The George Washington University, 1176 G Street, NW, Suite 101, Washington, DC 20052. Phone: (202) 994-7541. Fax: (202) 994-0245. E-mail: kkuqi@gwmail.gwu.edu. Other authors: eveleigh@gwu.edu; holzert@gwu.edu; g1usxs@gwu.edu; jim.levin@chp.edu; crowleys@upmc.edu.

Despite increased efforts by healthcare providers to use such systems, the latest report on office-based EMR physician adoption trends from the National Center for Health Statistics suggests that 62% of respondents claimed to be “somewhat satisfied,” “somewhat dissatisfied,” or “very dissatisfied” with their EMR product [5]. Existing EMRs are cumbersome to use and 30% of implementations fail, in part because they often cause productivity issues and do not fit well with the work physicians perform [2]. Loss of productivity caused by usability issues (such as providers having difficulties finding important information [2] in the EMR due to densely populated screens [3], and an increased number of steps to accomplish simple tasks) has diminished provider satisfaction with these systems. Additionally, medication-related errors caused by usability issues create a risk to patient safety. In 2006, approximately 25% of medication errors reported in the national MEDMARX preventable medication error database involved technology [1]. Of those, more than 80% resulted from the EMR *data entry* and *computerized provider order entry (CPOE)* functions [1]. CPOE, an EMR module designed to reduce error-prone handwritten and verbal orders by having providers enter them into a computer and then send them to the pharmacy [6], claims to reduce medication errors by 81% [7]. However, studies have suggested that poorly designed CPOE implementations [3] increase the risk of medication errors [7, 8].

EMRs are complex because of the variability of medical specialties, subspecialties and provider workflow [2]. As the complexity of the system increases, so does the need to support a multitude of complex functions that often bring forth user interface (UI) design challenges. A tempting design solution is to present as much information as possible to the user, which results in densely populated screens [1–3, 9]. Screen density can be defined as “*the number of items per degree of visual angle within a visually distinct group*” [10]. Density increases as the number of UI elements on the screen increases. Prior studies suggest that visual search time takes longer for densely populated screens [9–12]. User errors also increase with density [1]. A study found 22 types of medication error risks caused by EMR usability issues, such as densely populated screens and fragmentation issues [7]. The same study reported that 72% of clinical staff had difficulty in viewing all medications on one screen, and were often uncertain of the medication and doses being ordered or prescribed [7]. This is another reason, the significance it may have in reducing the risk of medication errors as well as improving user’s performance with the system, why reducing the screen density is important in designing EMR UIs.

The purpose of this study was to evaluate the application of systems engineering modeling approach to EMR usability and design. It introduces the new method of using the Design Structure Matrix (DSM) modeling technique to analyze and improve the design of EMR user interfaces. The aim of the study is to add to the existing portfolio of design practices for usable interfaces by means of a method that could simplify the EMR UI in terms of screen density and improve the user’s performance with the system. The proposed method includes a collection of screenshots for EMR UI design analysis and a walkthrough inspection, DSM modeling of the UI, and the evaluation of the existing and new designs using a predictive user performance model based on cognitive simulation.

1.1. Literature Review

The literature review on usability of healthcare information systems includes inspection and design improvement methods from cognitive psychology, ergonomics, and Human-Computer Interaction (HCI). Because the study focuses on the improvement of EMR UI design both in terms of complexity and the user's performance with the system, we cover only those aspects of usability evaluation and improvement methods. An overview of the DSM method used in this work is also provided.

1.1.1. Cognitive Walkthrough

Cognitive Walkthrough (CW) is a cost-effective HCI technique that has been used to evaluate the UIs of healthcare information systems [13, 14]. CW originates from the theory of learning by exploration [15], which is "an information processing model of human cognition" [14]. CW focuses on how easy it is for the user to accomplish a task, which covers the task analysis aspect of usability. This is helpful in identifying usability problems of EMR UIs that are typically associated with navigation type issues and users' attempt to locate the correct UI elements on the screen. The technique was developed to enable designers to find usability problems before user testing is possible [16]. CW is typically used with a group of 3–5 evaluators for the inspection process. The following needs to be defined before each inspection: who will use the system, what tasks will be analyzed, the action sequence for the tasks, and how the UI is defined [15, 17]. Evaluators answer a set of questions with the description of the user in mind [16]. Occasionally, the evaluator may unintentionally substitute user's behavior with his/her own experience with the system, a limitation that is often addressed by including more than one evaluator in the inspection process [16]. The method requires no prior knowledge of cognitive psychology, and requires no user testing. The results from the inspection process can be used to improve the design of EMR UIs.

1.1.2. Keystroke-Level Model

One of the most common metrics for measuring user's performance with the system is time-on-task or task completion time [18]. The Keystroke-Level Model (KLM) is an HCI technique used to predict a user's performance with the system, and the time it takes for a user to accomplish a task [19, 23]. Table 1 shows the physical-motor operators, a mental operator, and a system's response time operator used for counting keystrokes in KLM [19]. For simplicity, we have included only the operators that are most frequently used for measuring time-on-task.

The *P*-operator is an approximation of Fitts's Law, which states that pointing time from the starting point to a target element takes anywhere from 0.8 to 1.5 sec with 1.1 sec being the average [40]. Fitts's Law suggests that target elements in proximity to each other require less physical-motor actions and, therefore, can be reached faster than elements that are further apart [24, 40–42]. Hence, the basic principle of the law states that pointing time is a function of the distance to the target element and the size of the element [19, 24]. This is important information for an EMR usability study because it suggests that improving the spatial proximity between UI elements could reduce the time-on-task when pointing to a target element on the screen. KLM defines heuristic

Table 1. KLM description of operators [19, 25]

Operator	Abbrev.	Description	Duration in sec.
<i>Physical</i>	K	Keystroke or a button press	
		• Best typist	0.08
		• Average non-secretary typist	0.28
		• Worst typist	1.20
	P	Pointing with a mouse to a target element on a screen	1.10
	H	Homing hands to keyboard or mouse	0.40
<i>Mental</i>	M	Estimated mental operator for preparing to execute a physical action	1.35
<i>Response</i>	R	System's response time (<i>t</i>) measured in seconds (User's wait time until system responds)	<i>t</i>

rules for placing the *M*-operator [19, 25]. For example, rule one suggests that an *M*-operator be inserted in front of all *K* and *P*-operators (e.g., H + P + M + K: home hands to mouse + point with the mouse + estimated mental operation + pressing a key) [19]. The time-on-task (*T*) is then calculated based on the following equation [19]:

$$T = T_K + T_P + T_H + T_M + T_R \tag{1}$$

KLM is an effective, low-cost model for predicting time-on-task and an alternative to empirical usability testing methods that take time. This is essential for an EMR usability study where access to users is limited and costly. Occasionally, designers tend to make errors when placing the *M* operator necessary for the action sequence of the task, and doing all of the calculations by hand can be time consuming [25]. These limitations are addressed in other sophisticated tools and methods that predict user performance with more precision and that are still based on the underlying model of KLM [20-23, 25]. CogTool software (Section 2.2.3) used in this research work is one example [20-23].

1.1.3. ISO 9241-11 and NISTIR 7741

There are several empirical usability evaluation methods that involve testing with end-users; however, we limit the scope to the ISO 9241-11 and NISTIR 7741 standards due to their significance in designing usable EMR products. Safety-enhanced design criteria were recently added to the Stage 2 certification requirements of the meaningful use program introduced by the U.S. government as part of the Health Information Technology for Economic and Clinical Health Act for adopting and implementing a certified EMR technology [26]. The criteria requires EMR vendors to follow the User-Centered Design (UCD) guidelines documented in the ISO 9241-11 and NISTIR 7741 standards in designing EMR systems. The ISO 9241-11 standard defines what the

usability is vs. what it is not, which is often misinterpreted by practitioners. For example, usability is not user-acceptance testing, nor is it graphic or visual design, or market research [27]. This is important for any usability study, including this research work, where the focus is on improving the efficiency of the EMR relative to user performance. Developed by NIST, the NISTIR 7741 elaborates further on the UCD process while embracing the ISO 9241-11 definition of usability and by providing metrics for measuring effectiveness, efficiency, and user satisfaction [27].

The UCD process is founded on design guidelines from the fields of computer science, cognitive science, psychology, and HCI that are used for developing products with a focus on users [28]. There are several variations of UCD methodologies that predate NISTIR 7741, including the UCD version of ISO 9241-11 [4]. The one used in NISTIR 7741 describes a typical UCD process (e.g., documenting user requirements, usability testing) within the ISO 9241-11 definition of usability; however, it also references the information architecture aspect of the UI [27]. Information architecture is about organizing information on the UI so that it is easy for users to locate [18, 27]. As the complexity of the EMR UI increases, organizing information on the screen without the risk of densely populating the UI becomes a challenging task. Cluttering the EMR UI with elements increases visual search time, which impacts the user's performance with the system [1]. Although NISTIR 7741 includes the information architecture as part of the UCD methodology, it does not prescribe any usability technique or approach for organizing the elements on the UI.

The UCD approach involves high-cost usability and evaluation methods with end-users, making ISO 9241-11 and NISTIR 7741 costly to execute. Nonetheless, ISO 9241-11 and NISTIR 7741 are claimed as the standard for any future usability EMR work because of their inclusion in the safety-enhanced design criteria of the meaningful use program [26]. Therefore, we embrace their strict definition of usability in this work and introduce a new low-cost method that could enhance the existing portfolio of usability methods within the UCD approach and help in organizing UI elements with respect to screen density and, therefore, improve the user's performance with the system.

1.1.4. Design Structure Matrix

DSM is a matrix-based network-modeling technique used for developing and analyzing complex systems [29–33]. Donald Steward first introduced the method in 1981 as a systems analysis tool [30]. DSM enables visualization and modeling of the interactions among UI elements to understand the behavior they exhibit (e.g., related vs. unrelated elements). DSM is a square matrix stemming from the N^2 chart and graph theory square precedence matrices, showing system element inputs along the columns and outputs along the rows [29]. In DSM, the matrix entry (i, j) is noted with 0s and 1s or other numeric-based values, indicating the absence or presence of the interaction between UI elements.

Figure 1 shows an example of a binary DSM. In this example, the UI layer of the system comprises six elements, where each letter represents a UI element. In the pre-

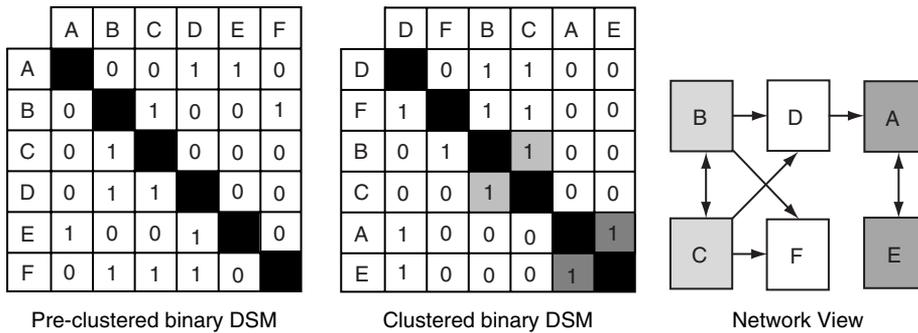


Figure 1. DSM examples (grey shading indicates clustered and highly interacting elements).

clustered version of the DSM, element A receives information from elements D and E. Additionally, element A sends information to element E. Therefore, elements A and E are said to be highly interacting system elements because they depend on each other for information and therefore may be grouped together. Next, a clustering algorithm is applied to the original matrix once all elements and their interactions are documented [31, 32]. The purpose is to group relatively similar elements into clusters based on their interaction types to help the analysis process for how to best organize elements on the UI. In Figure 1, the clustered binary-DSM version demonstrates that elements B/C and A/E interact highly with each other and cannot be decoupled because they rely on each other for information.

Elements outside the cluster provide additional information as to what their functions are in the UI. They often represent gaps in the interface design. For example, non-clustered elements may represent UI elements that are rarely used, if used at all; therefore, their removal may simplify the UI and improve visual search. Another representation of DSM is a network view, where edges (links or arrows) represent interactions, and nodes (squares) represent elements. Often, the network view of DSM is considered in conjunction with the matrix view for easy visualization and analysis of the system interfaces.

There are many benefits of DSM. The matrix-based format allows for easy visualization and analysis of interactions between the elements of complex interfaces. Analyzing the interactions between elements helps to understand how to best organize them in the EMR UI and improve information architecture. The DSM modeling process may also uncover information inside experts' heads, a knowledge that may exist beyond the use of available documentation [29]. As a systems engineering modeling technique, DSM has been found useful in many applications including financial systems, social systems, healthcare management, and defense [29, 43]. The lessons learned from different applications of DSM are applied in designing our research method for inspecting and improving EMR system usability.

1.2. Research Question and Hypothesis

Time-on-task is an effective measure of user performance [18]. Improving the time it takes for the user to complete a task improves the efficiency of the UI. Therefore, we define the *time-on-task* as the dependent variable and screen density as the independent variable. The research question developed for this study seeks to answer the following:

Has user performance of the existing EMR design been improved as part of the design improvement effort?

In addition to the research question this study seeks to answer, the following null hypothesis was developed based on the literature definition of usability and as measured by ISO 9241-11 [4, 27]:

H: Time-on-task is not different between the existing and new designs.

2. METHODS

2.1. Method Design

The foundation for the proposed method is derived from the literature review, the DSM approach to system modeling [29], and discussions with the physicians' representative at the hospital facility and a subject matter expert in usability studies for health information systems. Figure 2 illustrates the method steps developed and tested as part of this study. Each step produces an artifact that enables the usability analysis via modeling of UI element interactions for a given user task.

Step 1 - Identify: This step defines the scope of the system to be inspected. The system and sub-system modules (e.g., CPOE module) at the UI layer are identified first. Next, user workflows are identified for the given module(s), and individual user workflow tasks are documented.

Step 2 - Inspect: A usability inspection method (e.g., CW) is chosen. A preliminary inspection against the UI is performed to ensure usability issues exist and merit design improvement. Items collected in Step 1 serve the purpose of developing inspection specific artifacts, such as a description of the users and module(s) being inspected, and user workflow tasks. This method requires screen prototypes or actual screens to be used for inspection. Only modules that demonstrate usability problems should be subject to further steps in the method.

Step 3 - Construct: Figure 3 illustrates the Construct step of the method. First, user workflow tasks are broken-down into individual steps. Each step is then mapped to the module UI elements (e.g., buttons, text-fields, pick-lists). Next, matrix entries are populated to indicate the absence or presence of the interactions between the UI elements identified for each task step. Importantly, Step 3 does not necessarily require the execution of Step 2 because the design improvement effort involves the process of analyzing UI design patterns that impact usability. The purpose of Step 2 is to use cost-

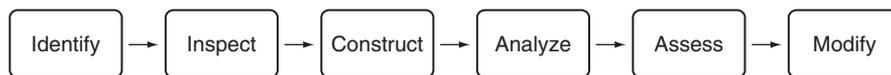


Figure 2. Research method.

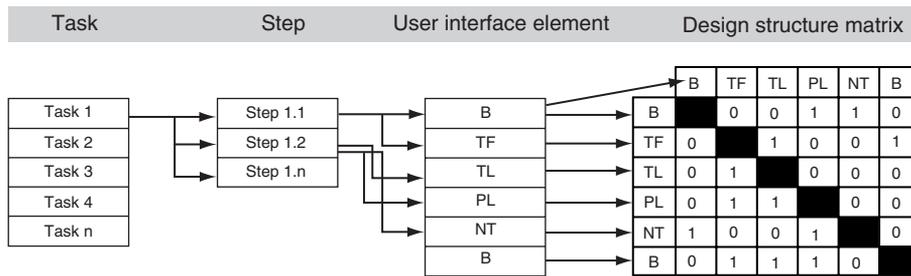


Figure 3. User workflow tasks, individual task steps, user interface elements, and the Design Structure Matrix. User interface elements include button (B), text-field (TF), text-list (TL), pick-list (PL), and non-interactive text field (NT).

effective, but well-established usability inspection methods before any major design improvement effort begins.

Step 4 - Analyze: The original DSM element entries and their interactions are rearranged using a clustering algorithm to organize related and unrelated UI elements into distinct groups based on their interaction types. Depending on algorithm results, iterating through the process of rearranging elements and their interactions may be required for an accurate representation of user tasks for the new design. This step also aims to identify UI elements that are rarely or never used.

Step 5 - Assess: A new design is produced based on results from Step 4. The purpose of Step 5 is to assess whether the new design could be expected to reduce screen density, improve time-on-task, and not change user functionality. The percentage change in screen density (PCSD) between the existing and new design is measured using the following formula:

$$PCSD = 100 \times \left(1 - \frac{\text{number of UI elements used in the new design}}{\text{total number of UI element in the existing design}} \right) \quad (2)$$

This formula is based on simple calculations for measuring system density [34] at the UI layer. The measure has been normalized to a range of 0 to 100 to be interpreted as percentage [45]. A low-cost method (e.g., predictive user performance simulation tools) is chosen for measuring time-on-task between the existing and the new designs. Alternatively, a high-cost method (e.g., empirical usability testing with end-users) could also measure time-on-task. Next, user functionality is validated on the new design to ensure that no functional loss occurred as a result of grouping UI elements into clusters based on their interactions.

Step 6 - Modify: The existing UI design is modified based on the design improvements.

2.2. Method Application

The study was conducted at an academic pediatric hospital located in Southwestern Pennsylvania. The hospital is a 296-bed facility that includes an emergency room and trauma center, cardiac, pediatric, and neonatal intensive care beds. The EMR system

inspected is implemented across the enterprise. System users include attending physicians, residents and fellows, nurses, nurse practitioners, and physician assistants.

To evaluate the application of systems engineering modeling approach to EMR usability and design (which is the purpose of this study), test our research method, and answer our research question and hypothesis, we used a combination of inspection (CW), modeling (DSM), and user performance predictive methods. This study involved interview sessions with the hospital's physician representative and a medical student familiar with the EMR system to: 1) define the scope of the system to be inspected (e.g., CPOE module), 2) identify user workflows and associated tasks for the chosen module, and 3) capture existing EMR screenshots for usability inspection. Additionally, we attended a three-hour EMR training session to become familiar with the system included in the study. The interviews and the collection of screenshots were carried out in a series of seven sessions spanning a period of two months in early summer 2012. All screenshots came from an EMR training system that used de-identified patient data. Because users were not involved in usability testing, Institutional Review Board approval was exempted.

Recognizing the importance of EMR usability in reducing the risk of medication errors, such as dosage errors [7], the EMR medication dosage calculator was chosen for usability inspection. Medication dosage calculators are used predominantly in pediatric medicine to determine the appropriate drug dose for young children when placing an order. The calculator is part of the EMR CPOE module used for sending orders to the pharmacy department. The scenario developed for the study involves a five-year-old female patient who checks in at the hospital with moderate pain and fever, and an acetaminophen order is placed. Six user tasks were developed, each representing one of the six ways a physician can adjust a medication dose using the calculator based on the patient's health condition:

- *Task 1: No Adjustment* – place 10 mg/kg acetaminophen order and verify updates to the medication dose based on patient's actual weight of 29.4 kg and targeted dose.
- *Task 2: Adjust Weight* – place 10 mg/kg acetaminophen order and adjust patient's weight. Verify updates to the medication dose based on the patient's adjusted weight of 32.2 kg and targeted dose.
- *Task 3: Reduce Dose to 75%* – place 10 mg/kg acetaminophen order and reduce dose to 75% of the targeted dose. Verify updates to the medication dose based on the reduced dose percentage.
- *Task 4: Adjust for Renal Failure* – place 10 mg/kg acetaminophen order and adjust for renal failure by applying the "Schwartz – full term" algorithm (used for estimating the creatinine clearance based on serum creatinine and body length). If creatinine clearance is less than 10 mL/min, reduce dose to 75%. Verify updates to the medication dose based on the creatinine clearance value and reduced dose percentage.
- *Task 5: Automatic Rounding* – place 10 mg/kg acetaminophen order and perform an automatic rounding. Verify updates to the medication dose based on the rounding rule.
- *Task 6: Standard Dose Rounding* – place 10 mg/kg acetaminophen order and round using a standard dose from the dose form template list (e.g., standard dose of 325 mg). Verify updates to the medication dose based on the standard dose rounding.

The six user tasks were used for the CW usability inspection artifacts. A total of 75 screenshots were captured for each task. Screenshots represent the 11 screens involved when placing a medication order including the single medication dosage calculator screen. The six user tasks and associated steps were mapped to individual UI elements (e.g., buttons, text-fields, pick-lists), and interactions between them were documented in the DSM. Cognitive simulation models that predict user performance were created using a free UI prototyping tool (Section 2.2.3), CogTool [20-23], where medication dose adjustment task completion time was captured in order to assess if the new design could be expected to improve performance. Last, any system errors caused by software defects that impact EMR usability were excluded from statistical analysis.

2.2.1. Cognitive Walkthrough Usability Inspection

A team of experienced cross-disciplinary HCI members (including the final author) who have performed similar CW evaluations together since 2003, conducted the cognitive walkthrough. The team consisted of five usability judges: two systems programmers, a project manager, a research analyst, and a professor of biomedical informatics. Evaluators had no prior experience with the EMR medication dosage calculator module being studied. The evaluation process took approximately three hours. Prior to executing the CW, an estimated 16 hours were spent on developing the CW inspection artifacts and preparing for the session. Each evaluator was provided with the following artifacts: user description of the EMR system, task steps description, and the sequence of user tasks. Because evaluators are required to answer a set of questions with the description of the user in mind [16], the session was kicked-off with an overview of the user. The following describes some of the 12 user attributes:

- The user is a physician, nurse practitioner, physician assistant, or pharmacist with sufficient knowledge of placing a medication order to understand UI terminology such as target dose, calculated dose, reduced dose, and standard dose.
- The user understands the basic actions available to them for adjusting weight, dosing based on percentage, dosing based on rounding rules, and dosing based on standard dose.
- The user understands that milligrams per kilogram (mg/kg) is the standard unit of measurement.
- The user understands that the calculated dose value is computed based on target dose and actual weight or adjusted weight.
- The user understands what serum creatinine means and how to adjust the medication dose for renal failure using the *Schwartz full-term* algorithm.

Evaluators were then presented with the task-action matrix as shown in Table 2, and the sequence of tasks that incorporated screenshots from “before” and “after” each step. The group worked collectively to answer four questions for each step with a “Yes” or “No” as defined by Wharton et al. [15, 17]:

Table 2. Task-action matrix

Actions	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Add medication to the list	✓	✓	✓	✓	✓	✓
Choose order type	✓	✓	✓	✓	✓	✓
Search for medications	✓	✓	✓	✓	✓	✓
Choose from search results	✓	✓	✓	✓	✓	✓
Enter dose information	✓	✓	✓	✓	✓	✓
Choose dose unit	✓	✓	✓	✓	✓	✓
Verify target dose, final dose, actual weight, and height	✓					
Adjust weight to 32.3 kg		✓				
Verify target dose, calculated dose, reduced dose, final dose, actual weight, adjusted weight, and adjustment		✓				
Reduce dose to 75%			✓			
Verify target dose, calculated dose, reduced dose, and final dose			✓			
Apply <i>Schwartz full-term</i> algorithm				✓		
Verify creatinine clearance level and if < 10 mL/min, reduce dose to 75%				✓		
Verify target dose, calculated dose, reduced dose, and final dose				✓		
Perform an automatic rounding					✓	
Verify target dose, calculated dose, reduce dose, final dose and rounding rule					✓	
Round using standard dose information from the <i>Dose Form</i> list						✓
Verify standard dose information						✓
Reference <i>Standard Dose Reference</i> for dose information and verify standard dose						✓
Verify target dose, calculated dose, reduced dose, final dose, and standard dose						✓
Apply standard dose						✓
Apply dose	✓	✓	✓	✓	✓	
Sign order	✓	✓	✓	✓	✓	✓
Verify medication list	✓	✓	✓	✓	✓	✓

Note: steps marked with '✓' are associated with individual tasks

1. *Will the user be trying to achieve the right effect?*
2. *Will the user be able to notice that the correct action is available?*
3. *Will the user associate the correct action with the effect they are trying to achieve?*
4. *If the correct action is performed, will the user see that progress is being made toward execution of their tasks?*

If any of the four questions was answered with a “No,” the entire step was considered to have failed [35]. Otherwise, the step was considered to have succeeded. During the usability inspection process, a scribe collected failed and succeeded steps for each user task.

2.2.2. DSM Modeling

Each of the six user tasks and associated steps were mapped to individual medication dosage calculator UI elements in the numeric-based DSM as part of the interview process with the hospital’s physician representative. The basis used for the mapping was determined by the function of the UI element in support of the action sequence of the task. UI elements are any of the following: a button (B), a text-list (TL), a pick-list (PL), a text-field (TF), and a non-interactive text-field (NT), i.e., text fields that display computed information based on inputs from other text-fields.

The interaction type parameters of the UI elements were defined as *information*, *similarity*, and *spatial adjacency*. For example, the patient *Actual Weight* text-field is populated automatically from the *Actual Weight Source* pick-list, which comes from other parts of the EMR system (e.g., patient encounter information). The *Actual Weight Source* pick-list contains up to five records of the patient weight. In the DSM, the *information* parameter represents the information signal that populates the *Actual Weight* text-field from the *Actual Weight Source* pick-list. The *spatial adjacency* parameter represents the proximity between a pair of UI elements. The frequency of use during task execution is what determines the adjacency between a pair of UI elements. For example, *Actual Weight* text-field and the *Actual Weight Source* pick-list are in spatial adjacency of each other because the latter frequently populates the former with information during task execution. The spatial adjacency between UI elements is also determined by the *similarity* parameter. The *similarity* parameter represents relatedness in type between a pair of UI elements. For example, two text-fields are said to be similar because they are of the same element type. The ISO 9241-11 design guidelines suggest that similar elements should be grouped together where possible in order to improve visual search [4].

Table 3 shows the numeric-based values assigned to each interaction type parameter in the pre-clustered DSM. Values for all applicable interaction types are summed up for every EMR UI element (e.g., *similarity + spatial adjacency*, *spatial adjacency + information*). The clustering algorithm determines the final interaction strength score and uses that information to group elements by calculating the distance and similarity between every one of them in DSM. Odd numbers were assigned to numerical values of interaction type parameters to prevent false positive outcomes during cluster analysis.

Table 3. UI element interaction type parameter values

Interaction Type	Value
No Interaction	0
Similarity	1
Spatial Adjacency	3
Information	5

An estimated 20 person-hours were spent in constructing the first DSM model for the first user task. The level of effort required to construct a DSM model can be estimated using the following equation [34]:

$$T = 0.02 \times N^2 \quad (3)$$

where N represents the number of UI elements. Other DSM models were constructed by reusing most of the DSM entries from the first model and making necessary adjustments to represent the remaining five user tasks. All DSM models were constructed and analyzed using the Cambridge Advanced Modeller (CAM) tool [36]. CAM is a free-for-research tool that uses a cluster algorithm for modeling and analyzing interactions in complex systems.

2.2.3. Predictive User Performance Model

Medication dose adjustment task completion times for the existing and new designs were predicted using the CogTool software [20–23]. The purpose was to measure whether changes to the existing design could be expected to improve the user's performance with the EMR system. Additionally, CogTool was used to create existing and new designs of the medication dosage calculator. Screenshots captured at the hospital facility were analyzed for developing the existing designs. Wireframes were developed for the new designs.

CogTool predicts the performance of a skilled user based on cognitive simulations. It is based on the underlying model of KLM (Section 1.1.2) and the ACT-R cognitive architecture [20–23, 37]. The ACT-R is used to simulate the human behavior and performance via additional physical and mental operators not available in the original KLM [23]. Each frame in CogTool represents a single UI screen of the design prototype. UI elements, such as buttons, text-fields, text-lists, pick-lists and non-interactive text fields are represented as widgets. Transitions from one frame to another represent a user action recorded in the design script. Design scripts include statements that are automatically created by the CogTool ACT-R architecture, and those explicitly added by the authors, such as *think*, *wait*, and *look-at*. Statements in the design script are mapped to the KLM operators. For example, the *wait* statements represent the system response R operator in KLM. The *wait* statements are transitions that occur when moving from one frame to another, similar to a system response time. We assumed 1.0 second for the *wait* statements. The *look-at* statements are added if the user

needs to shift the visual attention to a new UI element on the screen in order to take the next action [23]. The *look-at* statements also locate the spatial position of UI elements, thus allowing CogTool to obey Fitts's Law with precision by calculating the distance and time it takes to point with a mouse to a target element on the screen [23]. Therefore, CogTool does not have to use the 1.1 seconds estimated to predict the time it takes to point to a target element on the screen as used in KLM. We added the *look-at* statements for each widget of the existing and new designs in addition to other statements generated by CogTool predictions when measuring time-on-task.

CogTool provides more accurate predictions of user's performance with the system by using additional cognitive operators such as eye movements and memory retrievals that are not available in KLM. CogTool is reliable within +/- 10% of the observed user performance times [21].

3. RESULTS

The CW inspection results are summarized in Table 4. Answers from the four questions outlined in Section 2.2.1 were aggregated into overall percentages of failed and succeeded steps for each of the six user tasks analyzed. The usability judges found, on average, about 23% of the steps to have UI design flaws that a clinical user may experience while completing any of the six tasks. Failures were associated with navigation type problems and the evaluators' attempt to locate the correct UI elements on the EMR screen. For example, to adjust medication dose for a patient with renal failure (Task 4), evaluators had to first locate the *Creatinine Clearance* text-field and apply the *Schwartz full-term* algorithm from a pick-list. The algorithm pick-list was

Table 4. CW failed/succeeded steps of the medication dosage calculator

User Tasks	Number of Steps	Failed Steps	Succeeded Steps
		<i>Freq. (%)</i>	<i>Freq. (%)</i>
<i>Task 1:</i> order 10 mg/kg acetaminophen (No Adjustment)	10	2 (20%)	8 (80%)
<i>Task 2:</i> order 10 mg/kg acetaminophen (Adjust Weight)	11	2 (18%)	9 (82%)
<i>Task 3:</i> order 10 mg/kg acetaminophen (Reduce Dose to 75%)	11	2 (18%)	9 (82%)
<i>Task 4:</i> order 10 mg/kg acetaminophen (Adjust for Renal Failure creatinine clearance < 10 mL/min)	12	3 (25%)	9 (75%)
<i>Task 5:</i> order 10 mg/kg acetaminophen (Automatic Rounding)	11	2 (18%)	9 (82%)
<i>Task 6:</i> order 10 mg/kg acetaminophen (Standard Dose Rounding)	13	5 (38%)	8 (62%)

grouped together with other UI elements giving the impression that it was not intended to calculate the creatinine clearance level. Task 6 showed the highest number of design errors, where 38% of the steps failed. Failures associated with Task 6 included a lack of feedback mechanism when using commit buttons. For example, the task did not progress when the *Apply Standard Dose* button was used. Additionally, the *Apply Standard Dose* button was mistaken for the *Apply Dose* button because they are in spatial adjacency of each other in the existing design. The *Apply Standard Dose* button is used only for adjusting medication dose based on a set of standard dose templates (Task 6); hence the path to accomplish a task was often interrupted. This cost-effective usability inspection effort on the EMR medication dosage calculator revealed UI design flaws of sufficient degree to merit design improvement. As a result, we constructed the DSM model out of the existing design for further usability analysis and interface design improvement.

3.1. DSM Models

The DSM model represents all 32 UI elements available in the medication dosage calculator screen. As called for in Step 3 of the research method, we mapped individual task steps to UI elements, and used that information to populate the matrix entries with the interaction strength score assigned to each element. Because each of the six tasks differ from each other in that they require specific steps from start to finish, we constructed six DSM models, one for each task. We designed task-specific DSM models (task-specific UI screens) given the patient condition as opposed to DSM models that account for task overlap, so that the user is presented with the right level of functionality for accomplishing the task. For example, if the physician needs to adjust medication dose for a patient with a renal failure condition (Task 4), the EMR system would automatically launch the medication dosage calculator with all required UI elements for successful completion of Task 4.

For simplicity, only the matrix view of the first DSM model is exhibited in Figure 4. Additionally, we categorized UI elements within and outside the formed clusters and sub-clusters into two distinct classes: used UI elements (in grey) and unused UI elements (in white). The used UI elements represent elements that support the functionality required for the user to complete a task. The unused UI elements represent elements not required for the user to complete a task. Grouping of the elements is determined by the output of the DSM cluster algorithm based on the element interaction strength score. Elements with interaction strength scores of five and higher (≥ 5) belong to the group of used UI elements, whereas those with scores of less than five (< 5) belong to the group of unused UI elements. An element had to at least have a property of the *information* type parameter in order to qualify; i.e., an element is required to send or receive information within the pair during task completion. Used UI elements with scores higher than five have other properties that determine their placement in the UI. For example, an element with an interaction strength score of nine, *information* (5) + *spatial adjacency* (3) + *similarity* (1) = 9, is considered to exchange information within the pair, be in spatial adjacency with the interacting element, and be of the same element type. This information was also used to organize elements in the new EMR design.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Date of Birth (PL)	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gender (PL)	2	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Height (TF)	3	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
Height source (PL)	4	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
Adjusted weight (TF)	5	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0
Adjustment type (PL)	6	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0
Creatinine clearance (TF)	7	0	0	0	0	0	0	3	4	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0
Creatinine clearance alg. (PL)	8	0	0	0	0	0	0	3	0	4	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
Body surface area (TF)	9	0	0	0	0	0	4	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Body surface area alg. (PL)	10	0	0	0	0	0	0	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Target dose (TF)	11	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Target dose units (PL)	12	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Calculated dose (NT)	13	0	0	0	0	0	0	0	0	0	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reduced dose (NT)	14	0	0	0	0	0	0	0	0	0	5	9	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reduced dose percentage (PL)	15	0	0	0	0	0	0	0	0	0	0	0	3	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Final dose (TF)	16	0	0	0	0	0	0	0	0	0	5	5	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
Final dose units (NT)	17	0	0	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
Actual weight (TF)	18	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Actual weight source (PL)	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Serum creatinine (TF)	20	0	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
Serum creatinine source (PL)	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0
Apply dose (B)	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	4	0	0	0	0	0	0	0
Cancel (B)	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0
Formulae (B)	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
Standard dose reference (B)	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0
Apply standard dose (B)	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	0	0
Dose Forms: APAP (TL)	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Route (NT)	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
Standard dose (TF)	29	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	
Standard dose units (NT)	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	
Adjust reason (PL)	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	4
Rounding rule (PL)	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	4	0	0

Figure 4. DSM of Model I shows used UI elements (in grey) and unused UI elements (in white). The matrix entry (i, j) noted with values of 0s and ≥ 3 indicates the absence or presence of interactions between UI elements.

3.1.1. DSM Model I

The analysis of the first DSM model generated a total of eight clusters, three sub-clusters, and five sub-sub-clusters as shown in Figure 4. Elements within the main cluster (matrix rows 11–23) represent UI elements required for a physician to place a 10 mg/kg acetaminophen order and verify medication dose based on patient’s actual weight (Task 1). Elements included within the six sub-sub-cluster of the main cluster also represent used UI elements required for successful completion of Task 1. These sub-sub-clusters are *Target Dose* (rows 11/12), *Reduced Dose* (rows 14/15), *Final Dose* (rows 16/17), *Actual Weight* (rows 18/19), and *Serum Creatinine* (rows 20/21). The three sub-clusters within the main cluster (rows 11–17 and 18–21) and *Commit Buttons* (rows 22/23) demonstrate how used UI elements were organized in the new design. As an example, a conceptual wireframe of the first DSM model (matrix rows 11–23) is shown in Figure 5. Matrix elements 18/19 (*Actual Weight* and *Actual Weight Source*) and 20/21 (*Serum Creatinine* and *Serum Creatinine Source*) are further split into their own clusters. All elements within their clusters have interaction strength scores of eight (*information* (5) + *spatial adjacency* (3)). Therefore, matrix elements 18/19 and 20/21 exchange information between each other during task execution. Because of their frequency of use during task completion, they are also placed in spatial adjacency of each other. Although matrix elements 18/19 and 20/21 are in spatial adjacency of each other, they are not similar in

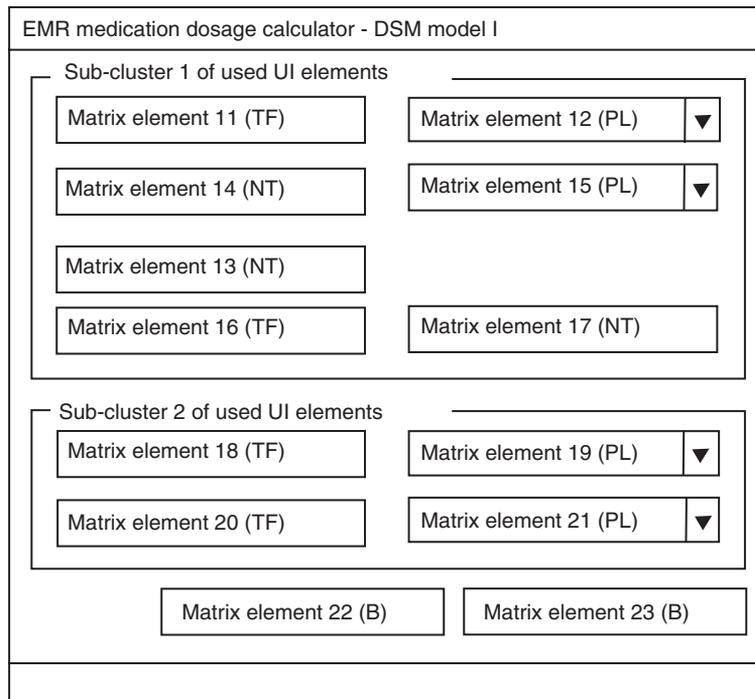


Figure 5. A conceptual wireframe of the first DSM model.

type and therefore are horizontally aligned on the UI. This is not the case for matrix elements 18/20 (text-fields) and 19/21 (pick-lists). They are vertically aligned in the UI because of their interaction strength score of four (*spatial adjacency* (3) + *similarity* (1)). Following the ISO 9241-11 design guidelines [4], we have grouped similar elements together (matrix elements 18/20 and 19/21) and have kept matrix elements 18/19 and 20/21 adjacent to each other due to their frequency of use.

EMR UI elements that fell outside the main cluster are identified for further analysis (matrix rows 1-10 and 24-32). These elements have interaction strength scores of less than five (< 5) because they do not exchange information with any of the used UI elements, and therefore are not required for the physician to complete a task. For example, the *Standard Dose* cluster elements (rows 29/30) are required when a physician adjusts medication dose based on a predefined set of standard dose ranges. An example would be Task 6, which is not part of DSM Model I. *Patient Height* cluster elements (rows 3/4) are not required or validated at the time of medication dose adjustment for Task 1. Unless the physician is adjusting medication dose based on the body surface area, the data source for the patient height comes from outside of the medication dosage calculator, and is validated at the time of patient admission to the hospital. *Adjusted Weight* (rows 5/6) is required if the physician chooses to adjust the medication dose by overwriting the patient's actual weight (e.g., Task 2). *Creatinine*

Clearance (rows 7/8) is required when adjusting medication dose for patients with renal failure condition (e.g., Task 4). It is noteworthy that the *Serum Creatinine* sub-sub-cluster elements (rows 20/21) typically used in conjunction with those of the *Creatinine Clearance* cluster (rows 7/8) were identified as used UI elements. Unlike the *Creatinine Clearance* cluster, the *Serum Creatinine* sub-sub-cluster serves as a reminder to physicians that a patient may have a kidney disease, and therefore the medication should be adjusted following the renal failure guidelines (e.g., Task 4). *Body Surface Area* cluster elements (rows 9/10) are required only if the physician is adjusting for specific drugs other than acetaminophen. Other elements placed outside the main cluster, for example patient *Date of Birth* and *Gender* (rows 1/2), were also categorized as unused UI elements. The data for the *Date of Birth* and *Gender* elements come from other parts of the EMR system. Both elements exist on the main patient encounter screen, which is visible to the physician at the time of the medication dosage calculator launch. Therefore, in this example, the presence of the *Date of Birth* and *Gender* elements further increases screen density and duplicates the information already available and visible on the main EMR system UI.

Removing the unused UI elements from the EMR medication dosage calculator and organizing the used elements into groups based on their spatial adjacency and similarity reduce screen density and simplifies the UI. Nielsen suggested that an interface should not contain elements that are irrelevant or rarely used because any units of information impacts visibility [9]. Findings from the analysis of the first DSM model suggest that the new medication dosage calculator design should include only 13 of the 32 total UI elements. The 13 UI elements provide the required functionality for the user to successfully execute Task 1. The new design reduces the UI screen density by 59%, which may lower the user cognitive load and improve visual search.

3.1.2. DSM Model II through V

The same approach for DSM Model I was taken to organize and group used and unused UI elements for DSM Models II through V. Model II produced a total of eight clusters, three sub-clusters, and five sub-sub-clusters. A total of 16 elements were identified as used UI elements, reducing screen density in the new design by 50%, or 16 fewer elements. DSM model III produced a total of eight clusters, three sub-clusters and five sub-sub-clusters. UI screen density was reduced by 56%, or 18 fewer elements. Model IV produced a total of seven clusters, two sub-clusters and seven sub-sub-clusters. UI screen density of the new design was reduced by 50%, or 16 fewer elements. DSM Model V produced a total of eight clusters, two sub-clusters, and five sub-sub-clusters, and UI screen density was reduced by 53%, or 17 fewer elements.

3.1.3. DSM Model VI

This model exposed system errors caused by software defects that impact EMR usability. Seven UI elements, although present in the existing design, did not interact with the other elements of the medication dosage calculator. For example, a physician is not able to place the 10 mg/kg acetaminophen order and round the medication dose using the standard dose form from a template list, because the latter does not generate any information when invoked by the end-user. Figure 6 shows the network view of the

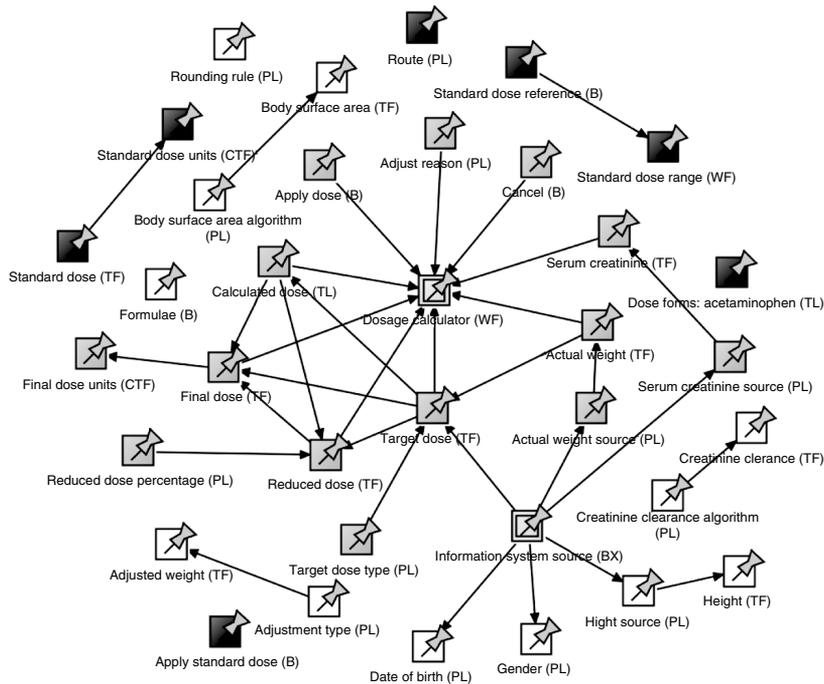


Figure 6. Model VI (network view) shows used UI elements (in grey), unused UI elements (in white), and system defects (in black).

DSM model produced. Color-coded in black, these elements are considered system defects because they hinder successful completion of Task 6. As noted earlier, the CW usability inspection findings showed the highest number of design flaws for Task 6 where 38% of the steps failed. Failures were associated with the lack of feedback mechanism, as the task did not progress when any of the seven elements were used. Because CW cannot uncover software defects, the root cause of the high percentage of failed steps for Task 6 emerged during the construction and analysis of DSM Model VI.

This work focuses on usability improvements but not utility. Utility represents the absence or presence of a function without considering if it is usable or not [27]. Therefore, findings produced by DSM Model VI were excluded from further evaluation and statistical analysis.

3.2. Validation of New Designs

A total of 110 frames were created in CogTool to statistically validate design results produced by this research. The first 55 frames were developed using screenshots captured from the existing design representing user tasks 1–5. The second 55 frames were developed using the DSM results that represent the new design for user tasks 1–5. The new design involved the removal of the unused UI elements from the existing screens. The basis for the new layout was the output of the DSM cluster analysis that

grouped used UI elements based on the interaction types of information exchange, spatial adjacency, and similarity. We kept the widget sizes similar to those of the existing design; therefore, all similar UI elements were considered to be of the same heights and widths. For example, in Figure 5, the widget sizes are similar to those of the existing medication dosage calculator design. Following the ISO 9241-11 design guidelines [4], we right-aligned commit elements (e.g., *Apply Dose* and *Cancel* buttons) and left-aligned other elements, all within their own distinct groups. We followed HCI design guidelines for the actual spacing between adjacent UI elements (e.g., four dialog units for elements similar in type and seven for those dissimilar in type) [44, 45].

Table 5 summarizes medication dose adjustment task completion time (time-on-task) captured for the existing and new UI designs using the CogTool predictive user performance model described in Section 2.2.3. An Anderson-Darling normality test at $\alpha = 0.05$ was performed first to determine that data generated from measuring time-on-task are from a normally distributed set. P-values of the existing (0.391) and new (0.575) designs are greater than the chosen alpha; therefore, by definition, normality holds. Additionally, the variances for the existing and new designs were tested to be homoscedastic. An unpaired two-tailed t-test was performed to test the null hypothesis defined in Section 1.2 at a 5% significance level. Table 6 summarizes the results of the hypothesis testing while comparing mean completion time of all user tasks for the existing and new designs.

Given that the P-value is less than chosen alpha, we reject the null hypothesis and conclude that time-on-task is statistically different between the existing and the new designs. Table 5 shows that the new design reduces time-on-task and could be expected to improve user’s performance with the system. For example, in 2012, the number of acetaminophen orders placed at the hospital pharmacy exceeded 112,000. Assuming that the difference between the existing and new design mean task completion time (8.24 sec) represents any of the five developed tasks a physician would perform when placing an acetaminophen order, the suggested EMR design improvements could have saved on average 21 hours of hospital physicians’ time over one month. Provided that

Table 5. Medication dose adjustments task completion time (in seconds)

Tasks	Existing Design	New Design
<i>Task 1:</i> order 10 mg/kg acetaminophen (No Adjustment)	34.43	25.78
<i>Task 2:</i> order 10 mg/kg acetaminophen (Adjust Weight)	39.53	31.85
<i>Task 3:</i> order 10 mg/kg acetaminophen (Reduce Dose to 75%)	38.54	29.97
<i>Task 4:</i> order 10 mg/kg acetaminophen (Adjust for RenalFailure creatinine clearance < 10 mL/min)	42.09	34.25
<i>Task 5:</i> order 10 mg/kg acetaminophen (Automatic Rounding)	38.51	30.04

Table 6. Hypothesis testing of medication dose adjustment task completion time

Descriptive Statistics	Existing Design	New Design
Mean (μ)	38.62 sec	30.37 sec
Variance (σ^2)	7.61	9.64
Standard Deviation (SD)	2.76	3.11
Sample size (n)	5	5
Degrees of Freedom (df)		8
Test Statistics (t)		4.43
Critical Value (5%)		2.31
P-value		0.00218 (< 0.01)

physician-patient consultation time at the hospital facility is approximately 18 minutes, physicians would have collectively seen additional 75 patients over the course of a year.

4. DISCUSSION

Usability has often been cited as a major factor hindering successful adoption of EMR systems [1–3, 38, 39]. User interface design challenges are a reflection of the complexity that EMRs expose due to the variability in medical specialties and provider workflow [2]. Design solutions have often resulted in complex interfaces with densely populated screens that impact user's performance with the system and introduce errors [1–3, 9, 38].

This work is significant in that it introduces the application of systems engineering modeling approach to EMR usability and design in understanding how UI elements influence one another in designing complex interfaces. The interactions between UI elements necessary for understanding the information architecture of the system are not specifically covered by the existing usability evaluation and design methods. Therefore, we developed and tested a six-step matrix-based method based on the systems engineering modeling aspect of UI inspection and design. In the present case study, we examined the usability of an EMR medication dosage calculator at an academic pediatric hospital, and demonstrated that standard inspection methods from HCI could detect design flaws before any major design improvement effort begins. Using the DSM modeling technique, we organized UI elements in the new design based on the interaction types of information exchange, spatial adjacency, and similarity. We removed rarely or never used UI elements that were not interacting with other elements during task completion. As a result, the new design reduced the UI screen density by an average of 54%. Reducing screen density lowers the user cognitive load, and is expected to improve visual search, to limit acceptance of erroneous data while fatigued, and to improve user's performance with the system. The cognitive simulations that predict users performance demonstrated that the mean time-on-task was reduced by 21% from the existing design to the new design (research question), which could save, on average, 21 hours of hospital physician's time in one month. Organizing UI elements based on task analysis and interaction types is also expected to improve the information

architecture of the system. It is noteworthy that use of the DSM modeling technique also revealed system errors caused by software defects that usability inspection methods do not cover. Although software defects are not usability issues, and therefore were excluded from further analysis, they still may impact EMR usability.

4.1. Limitations and Future Direction

This study addresses the efficiency aspect of usability as defined by the ISO 9241-11 standard [4]. Due to the limited availability of physician resources at the study hospital, the effectiveness and user satisfaction aspects of usability were not captured. The sample size of the study represents only a finite set of user tasks for adjusting medication dose in the EMR system. Additionally, any unintended consequences that may have occurred as a result of design changes were not studied. In this work, we considered similar UI elements to be of the same heights and widths in the DSM. Additional interaction type parameter values can be used to define different sizes of similar UI elements. The predicted user performance model is reliable within +/- 10% of the observed user performance times [21]. Therefore, a future study may consider capturing time-on-task with actual users in order to capture the error rate and the path that users take to achieve a given task, which satisfies the effectiveness aspect of usability as defined by ISO 9241-11 [4]. The proposed research method is one of the first to study interactions between UI elements to improve EMR usability, therefore correlation studies between this method and existing ones would be useful to assess the value added by using this approach.

5. CONCLUSION

Acknowledging the role of usability in reducing medication adverse drug events, and improving user performance, this study evaluated the systems engineering modeling aspect of UI inspection and design. We introduced a novel method that incorporates usability and DSM modeling techniques to improve EMR system usability by organizing elements on the UI based on their interactions, spatial adjacency, and similarity. Our research method was tested through usability inspection of an EMR medication dosage calculator. It was shown that DSM could be applied to an EMR module with known usability issues, resulting in improvements of predicted user performance based on cognitive simulations. Statistical analysis suggested significant differences between the existing and new designs as a result of the EMR design improvement effort. The proposed method is also expected to improve the information architecture of the system, and to be extended to inspect the usability of medical devices and to improve their design. This study also provided guidance for future direction, such as capturing the effectiveness and user satisfaction aspects of usability as defined by the ISO 9241-11 standard [4].

ACKNOWLEDGEMENTS

The authors thank Joaquin Espinoza Goodman for his assistance in collecting data for EMR inspection. The authors also thank members of the Department of Biomedical Informatics CW team at the University of Pittsburgh, including Elizabeth Legowski, Olga Medvedeva, Eugene Tseytlin, and Girish Chavan.

CONFLICT OF INTEREST

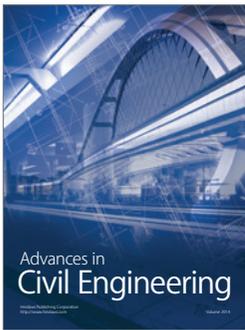
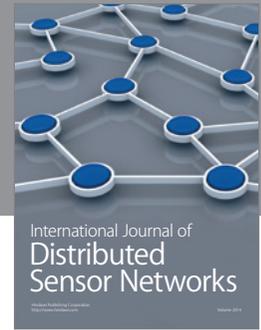
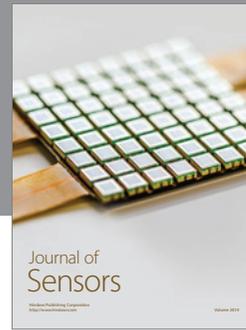
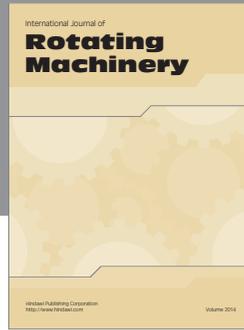
The authors indicate no potential conflicts of interest.

REFERENCES

- [1] Healthcare Information and Management Systems Society. Defining and testing EMR usability: principles and proposed methods of EMR usability evaluation and rating. 2009. http://www.himss.org/files/HIMSSorg/content/files/HIMSS_DefiningandTestingEMRUsability.pdf. Accessed Feb 25, 2013.
- [2] Smelcer JB, Jacobs-Miller H, Kantrovich L. Usability of electronic medical records. *Journal of Usability Studies*, 2009, 4(2):70–84.
- [3] Sengstack P. CPOE configuration to reduce medication errors: a literature review on the safety of CPOE systems and design recommendations. *The Journal of Healthcare Information Management*, 2010, 24(4):26–34.
- [4] International Organization for Standardization. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) – Guidance on usability. 1998. www.it.uu.se/edu/course/homepage/acsd/vt09/ISO9241part11.pdf. Accessed Feb 20, 2013.
- [5] Jamoom E, Beatty MS, Bercovitz A, Woodwell D, Palso K, Rechtsteiner E. *Physician adoption of electronic health record systems: United States, 2011*. NCHS data brief, no 98. Hyattsville, MD: National Center for Health Statistics. 2012.
- [6] Pham JC, Aswani MS, Rosen M, Lee HW, Huddle M, Weeks K, Pronovost PJ. Reducing medical errors and adverse events. *Annual Review of Medicine*, 2012, 63:447–463.
- [7] Koppel R, Metlay JP, Cohen A, Abaluck B, Localio AR, Kimmel SE, Strom BL. Role of computerized physician order entry systems in facilitating medication errors. *The Journal of the American Medical Association*, 2005, 293(10):1197–1203.
- [8] Walsh KE, Adams WG, Bauchner H, Vinci RJ, Chessare JB, Cooper MR, Hebert PM, Schainker EG, Landrigan CP. Medication errors related to computerized order entry for children. *Pediatrics*, 2006, 118(5):1872–1879.
- [9] Nielsen J. *Usability Engineering*, Academic Press, San Diego, CA, 1993.
- [10] Department of Health and Human Services, Digital Communications Division. Page Layout. <http://www.usability.gov/pdfs/chapter6.pdf>. Accessed Feb 10, 2013.
- [11] Halverson T, Hornof AJ. Local density guides visual search: sparse groups are first and faster. Conf Proc of Human Factors and Ergonomics Society. 2004, 1–5.
- [12] Tullis TS. The formatting of alphanumeric displays: a review and analysis. *Human Factors*, 1983, 25(6):657–682.
- [13] Kushniruk AW, Patel VL. Cognitive and usability engineering methods for the evaluation of clinical information systems. *Journal of Biomedical Informatics*, 2004, 37:56–76.
- [14] Liu Y, Osvalder AL, Dahlman S. Exploring user background settings in cognitive walkthrough evaluation of medical prototype interfaces: a case study. *International Journal of Industrial Ergonomics*, 2005, 35:379–390.
- [15] Wharton C, Rieman J, Lewis C, Polson P. The cognitive walkthrough method: A practitioner’s guide, in: Nielsen J, Mack RL (eds). *Usability Inspection Methods*, John Wiley & Sons, New York, NY, 1994, 106–139.
- [16] Hertzum M, Jacobsen NE. The evaluator effect: a chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction*, 2003, 15(1):183–204.
- [17] Sears A, Hess DJ. Cognitive walkthroughs: Understanding the effect of task-description detail on evaluator performance. *International Journal of Human-Computer Interaction*, 1999, 11(3):185–200.
- [18] Tullis T, Albert B. *Measuring the User Experience*, Morgan Kaufman, Burlington, MA, 2008.
- [19] Card SK, Moran TP, Newell A. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 1980, 23(7):396–410.

- [20] Teo L, John BE, Blackmon MH. CogTool explorer: a model of goal-directed user exploration that considers information layout. *Conf Proc of Human Factors in Computing Systems*. 2012, 2479-2488.
- [21] John BE, Utesch BS. Experiences with collaborative, distributed predictive human performance modeling. *Conf Proc of Human Factors in Computing Systems*. 2012, 437-452.
- [22] Bellamy R, John BE, Richards J, Thomas J. Using CogTool to model programming tasks. *Conf Proc of Evaluation and Usability of Programming Languages and Tools*. 2010, 1-6.
- [23] John BE, Prevas K, Salvucci DD, Koedinger K. Predictive human performance made easy. *Conf Proc of Human Factors in Computing Systems*. 2004, 455-462.
- [24] Fitts PM. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 1954, 47:381-391.
- [25] Hudson SE, John BE, Knudsen K, Byrne MD. A tool for creating predictive performance models from user interface demonstrations. *Conf Proc of the ACM Symposium on User Interface and Technology*. 1999, 93-102.
- [26] Department of Health and Human Services. Health Information Technology: Standards, Implementation Specifications, and Certification Criteria for Electronic Health Record Technology, 2014 Edition; Final Rule. (2012). <http://www.gpo.gov/fdsys/pkg/FR-2012-03-07/html/2012-4430.htm>. Accessed Feb 23, 2013.
- [27] Lowry SZ, Schumacher RM. NIST guide to the process approach for improving the usability of electronic health records. NIST Interagency/Internal Report (NISTIR) – 7741. 2010.
- [28] Johnson CM, Johnson TR, Zhang J. A user-centered framework for redesigning healthcare interfaces. *Journal of Biomedical Informatics*, 2005, 38:75-87.
- [29] Eppinger SD, Browning TR. *Design Structure Matrix Methods and Applications*, MIT Press, Cambridge, MA, 2012.
- [30] Steward DV. The design structure system: a method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 1981, 28(3):71-74.
- [31] Browning TR. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management*, 2001, 48(3):292-306.
- [32] Yassine A, Braha D. Complex concurrent engineering and the design structure matrix method. *Concurrent Engineering: Research and Applications*, 2003, 11(3):165-176.
- [33] Danilovic M. Bring your suppliers into your projects – Managing the design of work packages in product development. *Journal of Purchasing & Supply Management*, 2006, 12:246-257.
- [34] Suh ES, Weck O. Xerox digital printing technology infusion, in: Eppinger SD, Browning TR (eds). *Design Structure Matrix Methods and Applications*, MIT Press, Cambridge, MA, 2012, 43-48.
- [35] Nielsen J. Heuristic evaluation, in: Nielsen J, Mack RL (eds). *Usability Inspection Methods*, John Wiley & Sons, New York, NY, 1994, 25-61.
- [36] Wynn DC, Wyatt DF, Nair SMT, Clarkson PJ. An introduction to the Cambridge advanced modeler. *Conf Proc of the International Conference on Modelling and Management of Engineering Processes*. 2010, 1-4.
- [37] Lovett MC, Reder LM, Lebiere C. Modeling working memory in a unified architecture: an ACT-R perspective, in Miyake A, Shah P (eds). *Models of Working Memory*, Cambridge University Press, New York, NY, 1999, 135-182.
- [38] Hillestad R, Bigelow J, Bower A, Girosi F, Meili R, Scoville R, Taylor R. Can electronic medical record systems transform health care? Potential health benefits, savings, and costs, *Health Affairs*. 2005, 24(2):1103-1117.
- [39] Kaplan B, Harris-Salamone KD. Health IT success and failure: Recommendations from literature and an AMIA Workshop. *Journal of the American Medical Informatics Association*, 2009, 16:291-299.
- [40] MacKenzie SI. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 1992, 7:91-139.

- [41] Seow SC. Information theoretic models of HCI: a comparison of the Hick-Hyman law and Fitts' law. *Human-Computer Interaction*, 2005, 20:315–352.
- [42] Ding W, Lin X. *Information Architecture: the Design and Integration of Information Spaces*, Morgan & Claypool, 2009.
- [43] Kuqi K, Eveleigh T, Holzer T, Sarkani S. Using design structure matrix for improving electronic medical record usability. *Conf Proc of IEEE International Systems Conference*. 2012, 1–6.
- [44] Galitz WO. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*, 2nd ed., John Wiley & Sons, Canada, 2002.
- [45] Constantine LL, Lockwood LA. *Software for User: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison Wesley, Reading, MA, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

