Better EHR

Usability, workflow and cognitive support in electronic health records

Jiajie Zhang & Muhammad Walji, Editors

NCCD
The National Center for Cognitive Informatics and Decision Making in Healthcare

Better EHR

Usability, workflow & cognitive support in electronic health records

Jiajie Zhang and Muhammad Walji, Editors

NCCD
National Center for Cognitive Informatics & Decision Making in Healthcare

UTHealth | School of Biomedical Informatics
The University of Texas Health Science Center at Houston

ISBN: 978-0-692-26296-2

First edition, first printing: November 2014

This project is supported by Grant No. 10510592 for Patient-Centered Cognitive Support under the Strategic Health IT Advanced Research Projects (SHARP) from the Office of the National Coordinator for Health Information Technology.

Table of Contents

Acknowledgements

Project Advisory Committee

Sameer Baht	eClinicalWorks
Ralph Farr	The University of Texas Medical Branch at Galveston
James Ingram, MD	Greenway Medical Technologies
John Joe, MD, MPH	HealthTex International
David McCallie, MD	Cerner Corporation
W. Paul Nichol, MD	Veterans Affairs Office of Informatics and Analytics
Donald Norman, PhD	Nielsen Norman Group
Thomas Payne, MD	University of Washington Medical Center
Toby Samo, MD, FACP	Allscripts
Michael Shabot, MD	Memorial Hermann Healthcare System
Edward H. Shortliffe, MD, PhD	Arizona State University
Debora Simmons, PhD, RN	CHI St. Luke's Health System
Jack W. Smith, MD, PhD	Texas A&M Health Science Center
Michael Speer, MD	Baylor College of Medicine
James Spohrer, PhD	IBM Almaden Services Research
Laura B. Townsend	Louise H. Batz Patient Safety Foundation
Mellanie True Hills	StopAfib.org and American Foundation for Women's Health
Charlotte Weaver, RN, MSPH, PhD	Gentiva Health Services

Patricia B. Wise, RN, MS, MS, FHISS	Healthcare Information and Management Systems Society
David Woods, PhD	Center for Complexity in Human, Natural and Engineered Systems The Ohio State University

Federal Steering Committee

Joe Bormel, MD, MPH	Office of National Coordinator
Jodi G. Daniel, JD, MPH	Office of National Coordinator
Patricia Diggs	Office of National Coordinator
Jorge Ferrer, MD, MBA	US Department of Veterans Affairs
Doug Fridsma, MD, PhD	Office of National Coordinator
Charles P. Friedman, PhD	Office of National Coordinator
Amy L. Helwig, MD, MS	Office of National Coordinator
Kathy Kenyon, JD	Office of National Coordinator
Svetlana Lowry, PhD	National Institute of Standards & Technology
Alicia Morton, DNP, RN-BC	Office of National Coordinator
Matt Quinn	National Institute of Standards & Technology
Jacob Reider, MD	Office of National Coordinator
Wil Yu	Office of National Coordinator

Contributors

Ali Bahrami, PhD	Medico Systems
Jeffery L. Belden, MD	University of Missouri-Columbia
Natalie Benda	National Center for Human Factors in Healthcare, MedStar Health
Elmer V. Bernstam, MD, MSE	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Melissa Braxton	University of Washington
Keith Butler, PhD, MS	University of Washington
Trevor Cohen, MBChB, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics
J. Franck Diaz-Garelli	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Rollin (Terry) Fairbanks, MD, MS	National Center for Human Factors in Healthcare, MedStar Health
Amy Franklin, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Lyndsey Franklin, MS	University of Maryland
Yang Gong, MD, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics

Krisanne Graves, PhD, RN	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Robert A. Greenes, MD, PhD	Arizona State University
Mark Haselkorn, PhD	University of Washington
Peter J. Haug, MD, PhD	Intermountain Healthcare
Jorge R. Herskovic, MD, PhD	The University of Texas MD Anderson Cancer Center
A. Zachary Hettinger, MD, MS	National Center for Human Factors in Healthcare, MedStar Health
Lei Hua, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Todd R. Johnson, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Thomas Kannampallil, MS	University of Illinois at Chicago Department of Family Medicine
Richelle Koopman, MD, MS	University of Missouri-Columbia
Nathan Lowrance	University of Missouri-Columbia
Lawrence Lyon, MD	VA Puget Sound Healthcare Systems
Eliz Markowitz	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Allison B. McCoy, PhD	Tulane University School of Public Health and Tropical Medicine

Joi Moore, PhD	University of Missouri-Columbia
Jennifer Patel	Involution Studios
Vimla L. Patel, PhD, DSc	The New York Academy of Medicine
Catherine Plaisant, PhD	University of Maryland
C. Adam Probst, PhD	Baylor Scott & White Health
Raj Ratwani, PhD	National Center for Human Factors in Healthcare, MedStar Health
Deevakar Rogith, MBBS	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Konrad Schroder	University of Washington
Ben Shneiderman, PhD	University of Maryland
Dean F. Sittig, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Hsingyi Song	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Juhan Sonin	Involution Studios
Davide Sottara, PhD	Arizona State University
Sureyya Tarkan, PhD	University of Maryland
Harold Thimbleby, PhD	Swansea University
Muhammad F. Walji, PhD	The University of Texas Health Science Center at Houston School of Dentistry

Adam Wright, PhD	Brigham and Women's Hospital
Xinshuo Wu	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Yan Xiao, PhD	Baylor Scott & White Health
Jiajie Zhang, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics
Min Zhu, MD, PhD	The University of Texas Health Science Center at Houston School of Biomedical Informatics

Foreword

Jiajie Zhang, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Electronic Health Records (EHR) offer great potential to increase healthcare efficiency, improve patient safety, and reduce health costs. The adoption of EHRs among office-based physicians in the US has increased from 20% ten years ago to over 80% in 2014. Among acute care hospitals in US, the adoption rate today is approaching 100%.

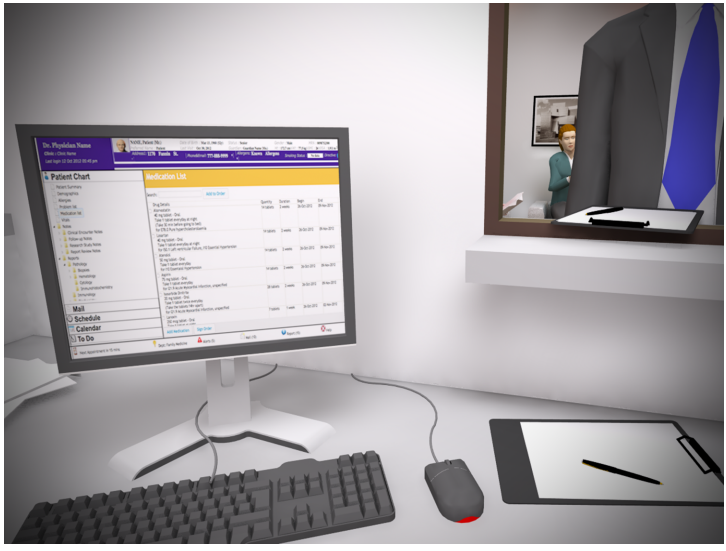
Although the rapid adoption of EHR is generating benefits for care providers and patients, usability, workflow, and cognitive support problems have surfaced. When an EHR is inappropriately designed, implemented, or used, problems can overweigh benefits.

Here's an imaginary example of the kinds of problems an EHR system with poor usability can cause. (Note: this example is a work of fiction. Any resemblance to real company names, persons, places, events or technologies is purely coincidental).

Imagine a patient arriving at his physician's office.

He checks in and is repeatedly asked his name, date of birth, and other identifying information while the receptionist tries to identify the correct record.

The patient is handed a clipboard and asked to complete forms that he has already completed and submitted via fax. He still has to complete the forms because the receptionist cannot find them in his record.



Moving to the exam room, he stops to have his height and weight checked in an area with other patients. His information is announced aloud as an assistant enters it into the EHR. During a phone call, the system logs her out due to inactivity. When she logs in again, she has forgotten the measurements and asks the patient to return for re-measure.

In the exam room, another assistant checks temperature, pulse, and blood pressure.

She struggles inputting information into the EHR, repeatedly asking the patient for drug allergies and medications. The patient gets frustrated because these questions were included on the forms already completed—now twice.

The physician enters the room, but has difficulty logging into the EHR. Then he struggles finding the patient's record. Because the EHR's workflow does not match the clinic's, he has difficulty finding the reason for today's visit. All the while, the physician primarily looks at the screen, not the patient.

After the examination, the patient is given a general brochure and a folder of blurry education materials photocopied once too many times.

As the physician talks to the patient, he is automatically logged out of the EHR system for inactivity.

The physician logs back on to write a prescription for the patient, acknowledging multiple inconsequential drug-drug interactions and drug-allergy interactions. Unfortunately, he misses an important interaction. The prescription is electronically sent to the pharmacy…

… with unfortunate results.

The usability, workflow, and cognitive support problems in this scenario are barriers to EHR meaningful use. Although many of these problems have been addressed by the EHR community, many others remain and prevent optimal use of EHRs by physicians. A worldwide 2014 KLAS survey of healthcare physicians shows usability is the number one criterion when choosing an EHR, with nearly 75% of the responders reporting usability more important than any other criteria [1].

This book is a subset of work from SHARPC, an Office of the National Coordinator for Health Information Technology funded project focusing on patient-centered cognitive support issues of EHRs. SHARPC is a sizable project ($15 million funding over a four-year period), yet it only touches some of the usability, workflow, and cognitive support issues of EHRs. We hope EHR designers, developers, implementers, users, patients, and policy makers will find this book informative and useful.

1: Cognitive Support for Health Information Technology

Jiajie Zhang, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Muhammad F. Walji, PhD
The University of Texas Health Science Center at Houston School of Dentistry

ABSTRACT

This chapter provides an overview of the SHARPC project—how SHARPC was started, conceptualized and funded, what goals it tried to achieve, and how it approached patient-centered cognitive support challenges. Research and products resulting from SHARPC are described in following chapters.

INTRODUCTION

The Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009 provided $19 billion to support a multi-pronged approach to increase nationwide adoption and meaningful use of electronic health records by 2014. As part of the HITECH Act, four centers were funded by the Office of the National Coordinator for Health Information Technology (ONC) as Strategic Health IT Advanced Research Projects (SHARP). The goal was to "address well-documented problems that impede the adoption of health IT" [2]. Security and health information technology problems (SHARPS) were addressed by the University of Illinois at Urbana-Champaign. The University of Texas Health Science at Houston (UTHealth) focused on patient-centered cognitive support (SHARPC) issues. Better health care application and network design (SMART) was studied by Harvard University. Work on secondary use of electronic health record (EHR) information (SHARPN) was undertaken by Mayo Clinic.

APPROACH

SHARPC studied usability, workflow, and cognitive support issues of health information technology (HIT). Work was performed through UTHealth's National Center for Cognitive Informatics and Decision Making in Healthcare (NCCD). NCCD is a collaborative center with key investigators from eight institutions and consultants, and advisors from other institutions, organizations and corporations across the country. NCCD's vision is to become a national resource providing strategic leadership in research and applications for patient-centered cognitive support in healthcare. It's mission:

1. Bring together a collaborative, interdisciplinary team of researchers across the nation with the highest level of expertise in patient-centered cognitive support research from biomedical and health informatics, cognitive science, computer science, clinical sciences, industrial and systems engineering, and health services research.
2. Conduct short-term research that addresses the urgent usability, workflow, and cognitive support issues of HIT as well as long-term, breakthrough research that can fundamentally remove the key cognitive barriers to HIT adoption and meaningful use.
3. Translate research findings to the real world through a cooperative program involving researchers, patients, providers, HIT vendors, and other stakeholders to maximize the benefits of HIT for care quality, efficiency, and safety.

SHARPC considered "patient-centered cognitive support" to be HIT specifically designed to support problem solving and decision making for the highest quality of care as measured by the Institute of Medicine's (IOM) six dimensions of quality (safe, effective, timely, efficient, equitable, and patient-centered) [3]. SHARPC's characterization of cognitive challenges for HIT adoption and meaningful use is shown in Figure 1.

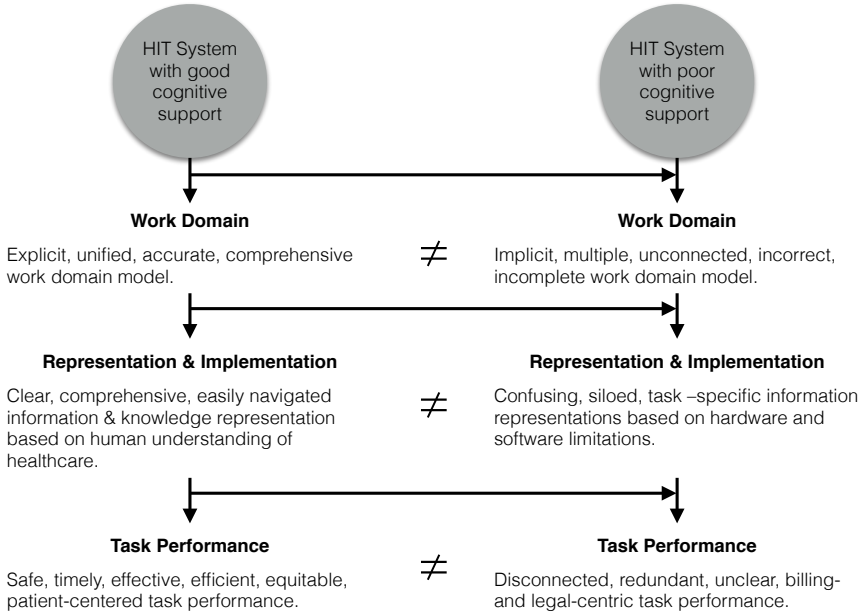


Figure 1. Cognitive challenges for Health IT. There are gaps at three levels between good HIT systems and poor HIT systems.

Cognitive challenges can be described as gaps between HIT systems that have good cognitive support and HIT systems with poor cognitive support. At the work domain level, HIT systems with good cognitive support have an explicit, unified, accurate, and comprehensive model that reflects the true ontology of the work domain, providing a clear understanding of the care problem that is independent of how systems are implemented. HIT systems with poor cognitive support typically suffer from models of the work domain that are implicit, multiple, unconnected, disparate, incomplete, and often inaccurate.

At the representation and implementation level, HIT systems with good cognitive support have clear, comprehensive, easy to navigate information and knowledge models optimized for human users. HIT systems with poor cognitive support have representations based on hardware and software features, which can make them confusing, siloed, task-specific, difficult to use and learn, and hard to navigate if these representations do not match human characteristics.

At the level of task performance, HIT systems with good cognitive support build in safe, timely, effective, efficient, equitable, patient-

centered task performance [4]. HIT systems with poor cognitive support often have disconnected, redundant, tedious, and unclear user models based on billing and legal requirements, which can interfere with task performance.

Cognitive Support Issues		
Observations	Consequences	Opportunities
Patient records are fragmented • Computer and paper records co-exist • Computer records divided among task-specific transaction processing systems • Users have to know where to look	• Synthesis depends on intra-team conversation • Problem recognition left to chance • Team members waste time getting information in the form they want to use	• Techniques to synthesize and summarize information about patient in and across systems with drill-downs for detail • Mechanisms to focus on a constellation of related factors
Clinical user interfaces mimic paper • Flow sheet is predominant display • Font size is challenging *User interfaces do not reflect human factors and safety design* • Improperly structured pull-down lists • Inconsistent use of location, symbol, and color	• Important information and trends are easily overlooked • Cognitive burden of absorbing information detracts from thinking about what the information means • Systems intended to reduce error but create new errors	• Design reflecting human and safety factors • Automatic capture and use of context • Techniques to represent and capture data at multiple levels of abstraction
Support for evidence-based medicine and computer-based advice is rare	• Lost opportunity to provide patient-specific decision support	• Peer to peer techniques for developing guidelines and decision support content • Mass customization techniques for practice guidelines

Cognitive Support Issues		
Observations	Consequences	Opportunities
High complexity and coordination requirements of care. • Within teams • Across teams and services within settings • Across settings • Non-transparent workflow • Clinical roles and responsibilities are not explicit	• Reactive care • Handoff errors • Redundant care • No clear thinking about overall workflows, process design, and efficiency and handoff errors • Unpredictable escalation and response	• Dynamically computable models to represent plan for care, workflow, & escalation • Scripting languages for decision and workflow support content • Uniform provider ID • Explicit team roles and escalation paths • Capabilities for context-aware efficient scheduling
Clinical users do not have a consistent understanding of the purpose of a system or the functionality of the user interface	• Inefficient workflow Incomplete or inaccurate data entry • Misinterpretation of information • System work-arounds	• Design system modules for use in production (operation) and simulation (training)
Data capture/data entry are commonly manual	• More time spent entering than using data • Loss of opportunity for decision support	• Redesign roles, process, and technology to capture data at the source as data are created

Table 1. Summary of NRC Committee's Observations & Opportunities for Patient-Centered Cognitive Support

	Projects
1	Work-Centered Design of Care Process Improvements in HIT
2A	Cognitive Foundations for Decision Making: Implications for Decision Support
2B	Modeling of Setting-Specific Factors to Enhance Clinical Decision Support Adaptation
3	Automated Model-based Clinical Summarization of Key Patient Data
4	Cognitive Information Design and Visualization: Enhancing Accessibility and Understanding of Patient Data

Table 2. Five projects to address patient-centered cognitive support issues

ONC-Identified Six Cognitive Challenges for Patient-Centered Cognitive Support	1	2A	2B	3	4
Creating models that support dynamic abstraction of clinical information	X	X	X	X	
Techniques for parsimonious information display that simplifies, while capturing essential features of a clinical decision problem	X			X	X
Understanding decision making under stress and time pressure, and its implications for cognitive support		X	X		
Communication to clinicians, addressing message content and delivery, that blends with workflow	X	X	X		X
Methods to support decisions that involve multiple stakeholders, taking into account their preferences and utilities	X				
Methods for minimizing and simplifying, when it is necessary, manual data input by clinicians	X				X

Table 3. ONC's six cognitive challenges and their mapping to SHARPC's five projects

These gaps result from multiple cognitive problems with HIT, as identified in the 2009 National Research Council [5] (Table 1). SHARPC was funded to address many of these cognitive problems. Under SHARPC, five projects were developed, each focusing on a set of short and long-term tools and methods to address major cognitive problems (Table 2). Each cognitive problem was covered by more than one project and each project covered more than one problem (Table 3). Collectively, SHARPC's collaborative, interwoven, and integrative projects delivered a suite of tools and methods to improve HIT cognitive support.

RESULTS

The tools and methods developed from SHARPC are described in following chapters as high level summaries. Detailed list of tools, methods, and other products can be found at SHARPC's website www.sharpc.org.

DISCUSSION

SHARPC elevated the awareness and importance of EHR usability and patient-centered cognitive support. ONC added Safety-Enhanced Design to the 2014 edition of "Health Information Technology: Standards, Implementation Specifications, and Certification Criteria for Electronic Health Record Technology" [6]. Test Procedure §170.314(g)(3) for Safety-Enhanced Design was developed to emphasize the importance EHR usability safety and includes eight use cases:

1. Computerized provider order entry (CPOE)
2. Drug and allergy interaction checks
3. Medication list
4. Medication allergy list
5. Clinical decision support
6. Electronic medication administration record (inpatient setting only)
7. Electronic prescribing (medication order entry), and
8. Clinical information reconciliation (patient problems, medications, and allergies)

Including safety-enhanced design in EHR certification is a major step toward improving EHR usability. However, much more needs to be done to improve the usability, safety, workflow, and other cognitive issues of EHRs. HIT usability is a key research and development topic requiring systematic and sustained efforts by vendors, users, patients, researchers, professional organizations, and federal agencies to achieve meaningful outcomes.

Foundations of Usability

While slow to be adopted in healthcare, usability principals are common in other industries, including computer hardware and software design. Computer usability is often described in vague terms, such as "convenience," "ease-of-use," and "user friendliness." A more precise definition is needed for EHRs, such as "how useful, usable and satisfying a system is for its intended users to accomplish goals in a work domain by performing certain sequences of tasks" [7].

Usefulness is a critical component of an EHR system, and it's a quality that can be objectively analyzed and measured. SHARPC developed the TURF EHR Usability Framework (Chapter 2) specifically for this purpose.

Another key issue of EHR usability is balance. There's an inherent challenge designing useful EHR systems that are both systematic and flexible. SHARPC studied tradeoffs when users adapt to EHR systems versus a system adapting to user needs. Chapter 3: The Systematic Yet Flexible Systems Analysis (SYFSA) proposes how to determine an appropriate mix.

2: TURF Unified Framework of EHR Usability

Jiajie Zhang, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Muhammad F. Walji, PhD
The University of Texas Health Science Center at Houston School of Dentistry

ABSTRACT

We present a unified framework for evaluating electronic health records system usability. TURF is a theory for describing, explaining and predicting usability differences; an objective method for defining, evaluating and measuring usability; a process for designing in good usability; and a potential principle for developing EHR usability guidelines and standards. TURF defines usability as how useful, usable, and satisfying a system is for intended users to accomplish goals in a work domain by performing sequences of tasks. TURF provides a set of measures for useful, usable, and satisfying dimensions of usability. TURF stands for Task, User, Representation, and Function, four components that determine usability of an EHR system. These components are described with theoretical descriptions and examples of how usability is measured in several case studies.

How TURF can be used to improve usability through redesign is also demonstrated. We conclude that usability can not only be defined scientifically under a coherent, unified framework, but also objectively and systematically measured.

INTRODUCTION

Electronic Health Records (EHR) systems have great potential to increase care quality, efficiency and safety through wide adoption and meaningful use [8-14], a major rationale behind the national HIT Initiative started by President Bush in 2004 and strengthened by President Obama in 2009. The $19 billion HITECH Act's goal under the American Recovery and Reinvestment Act is for every American's

medical records to be on computers by 2014. However, there are huge gaps between the status quo and the potential of EHR, primarily due to cognitive, financial, security/privacy, technological, social/cultural, and workforce challenges [15-18]. The cognitive challenge is mainly concerned with usability issues, which have not receive significant attention in the EHR community until recently [5, 18-23]. Unlike many other industries (e.g., aviation, nuclear power, automobile, consumer software, and consumer electronics) where usability is the norm in product design, the practice of usability in EHR has been sporadic, unsystematic, casual and shallow, partly due to lack of sufficient attention to usability and lack of EHR-specific usability frameworks and methods. Designing and implementing an EHR system is not so much an IT project as a human project about usability, workflow, patient safety and organizational change [15, 18, 21, 24-26]. To facilitate adoption and meaningful use of EHR, an EHR-specific usability framework is needed to increase efficiency and productivity, increase ease of use and ease of learning, increase user retention and satisfaction, decrease human errors, decrease development time and cost, and decrease support and training costs. We present here the initial form of a unified framework of EHR usability, TURF, for: 1) describing, explaining, and predicting usability differences; 2) defining, evaluating, and measuring usability objectively; and 3) designing in good usability. Once fully developed, TURF could also be used as a principle for developing EHR usability guidelines and standards.

Definition of usability

Under TURF, usability refers to how useful, usable and satisfying a system is for its intended users to accomplish goals in a work domain by performing certain sequences of tasks. Useful, usable, and satisfying are the three major dimensions of usability under TURF (see Table 1).

	Dimensions	Descriptions	Representative measures
Usability	Useful	A system is useful if it supports the work domain where the users accomplish the goals for their work, independent of how the system is implemented	• Across-model Domain Function Saturation: Percentage of domain functions in the EHR vs. all domain functions in the work domain • Within-model Domain Function Saturation: Percentage of domain functions over all functions (domain and non-domain) in the EHR

	Usable	A system is usable if it is easy to learn, easy to use, and error-tolerant.	• Learnability
			  • Number of trials to reach a certain performance level
			  • Number of items that need to be memorized
			  • Number of sequences of steps that need to be memorized
			• Efficiency
			  • Time on task
			  • Task steps
			  • Task Success
			  • Mental effort
			• Error Prevention and Recovery
			  • Error occurrence rate
			  • Error recovery rate

	Satisfying	A system is satisfying to use if the users have good subjective impression of how useful, usable, and likable the system is	• Various ratings through survey, interview, and other instruments

Table 1. Dimensions and measures of usability under TURF.

TURF's usability definition is based on the ISO definition (ISO 9241-11), but differs in significant ways. ISO defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." Under ISO's definition, effectiveness refers to the accuracy and completeness with which users achieve specified goals. Efficiency refers to the resources expended in relation to the accuracy and completeness with which users achieve goals, and satisfaction refers to comfort and acceptability of use. TURF and ISO definitions of usability differ with "effective" in ISO and "useful" in TURF, and "efficient" in ISO and "usable" in TURF.

Under TURF, "useful" refers to how well a system supports the work domain where users accomplish goals for their work independent of how the system is implemented. A system is fully useful if it includes domain, and only domain, functions essential for the work, independent of implementations. Full usefulness is an ideal situation; it is rarely achieved in real systems. Usefulness also changes with the change of the work domain, with development of new knowledge, and with availability of innovations in technology. Usefulness can be measured by the percentage of domain functions in the EHR over all domain functions (those in the system and those not) and the ratio of domain functions vs. non-domain functions in the system. More details about domain functions are described in Section 3.2.

How usable a system is can be measured by learnability, efficiency, and error tolerance. Under TURF, a system is usable if easy to learn, efficient to use, and error-tolerant. Learnability refers to ease of learning

and re-learning. This can be measured by examining how much time and effort are required to become a skilled performer for the task, such as the number of trials needed to reach a preset level of performance, number of items that need to be memorized, and the number of task steps to be memorized. Learnability usually correlates positively with efficiency, but it could be independent of efficiency and sometimes correlates negatively with efficiency (e.g., an interface optimized for ease of learning may not be optimized for efficiency). Efficiency refers to the effort required to accomplish a task. This is usually measured in terms of time on task, task steps, task success rate, mental effort, etc. Time on task refers to the time it takes to complete a task. Task steps refer to the number of steps (both mental, such as recalling a drug name from memory, and physical steps, such as clicking a button on the screen) needed to complete a task. Task success rate is the percentage of time a task can be successfully completed. Task success rate is referred to as the completion rate definition of usability. Under TURF, however, effectiveness, including task success rate, is considered a measure of efficiency because it is a measure of user performance, just like time on task. Mental effort, under TURF, is the amount of mental effort required for a task, such as the percentage of mental steps over all steps (physical and mental). Error prevention and recovery refer to the ability of a system to help users prevent and recover from errors. This can be measured by error frequency, recovery rate, and other measures. Under the ISO definition of usability, error is a measure of effectiveness. Under TURF error is a measure of efficiency for the same reason that task success rate is considered an efficiency measure under TURF.

Satisfaction under TURF is similar to satisfaction under ISO's definition of usability. In TURF, satisfaction refers to the subjective impression of how useful, usable and likable a system is to a user. This is typically measured through survey questions assessing an end user's perception or ratings of a system. Subjective assessment of user satisfaction is an important component of usability. But this aspect is often equated with all that usability is about, giving many people the wrong impression that usability is subjective, unreliable and useless for product improvement. TURF, as a unified framework, offers both objective and subjective measures of usability. The useful and usable aspects under TURF are objective, evidence-based, and systematic. Only

when both are considered is usability evidence-based. Satisfaction alone should never be used as the complete measure of EHR usability.

TURF considers usefulness a major dimension of usability because TURF takes a work-centered approach [27-31]. Usefulness is also often referred to as utility or functionality. Its importance in successful applications is long acknowledged. For example, Landauer argued successful applications should be not only usable, their functionality should also be useful [32]. Goransson and colleagues [33] compiled a list of applications that failed for lack of useful functionality, even though they were usable. If the functionality or utility of an application is not useful, whether it is usable or not is irrelevant. On the other hand, if functionality is chosen effectively and usable, then even poor user interfaces might be acceptable. Successful applications should be both useful *and* usable, and they should be considered together because they are not independent, as demonstrated by Butler et al. [27] who developed a work-centered framework on how to allocate functionality across machines and users. If a system does not have a desired function, users may have to find a workaround that could complicate usableness of the system. Thus, choice of functionality will not only determine how useful a system is, but also how usable [34]. For this reason, under TURF, usefulness (functionality or utility) is an integral component of usability.

TURF

The essence of usability is *representation effect*. Representation effect is the phenomenon that different representations of a common abstract structure (e.g., a work domain ontology, see Section 3.2.1 for details) can generate dramatically different representational efficiencies, task difficulties and behavioral outcomes [35-39]. Form of representation is so important that it often determines what information can be perceived, what processes are activated, what structures can be discovered, what errors are generated, and what strategies are learned and adopted [40].

Usability differences between two products for the same work domain, such as Arabic numerals vs. Roman numerals for calculation, or DOS vs. Windows operating systems for computer tasks, are prototypical examples of the representation effect. For EHR systems, whether one EHR has better usability than another for a display, a module, or the

entire system is also a representation effect. In Figure 1, usability of an EHR system is decomposed into two components: intrinsic complexity and extrinsic difficulty. Intrinsic complexity reflects the complexity of the work domain and is an indication of system usefulness. It also reflects the amount and complexity of work, independent of any procedures, activities, or implementations. Different work domains have different work domain ontologies which are associated with different levels of intrinsic complexities. Extrinsic difficulty reflects the difficulty when a specific representation or interface is used to perform a specific task and is an indication of system usableness. Extrinsic difficulty is mainly determined by formats of representations and workflows of tasks. Intrinsic complexity and extrinsic difficulty together reflect the usability of the system.

The next few sub-sections describe intrinsic complexity and extrinsic difficulty in terms of TURF's four components: Task, User, Representation, and Function, along with the results of several case studies.

It should be noted that EHR systems, like many other products, are used in real world settings that are interruption-laden, unpredictable, stressful, and involve many other factors such as organizational, social, physical, spatial, temporal, financial, and historical factors. All of these can contribute to the representation effect in various ways and should be considered in the design and evaluation of EHR usability. The focus of this paper, however, is only uninterrupted tasks performed by individual users.

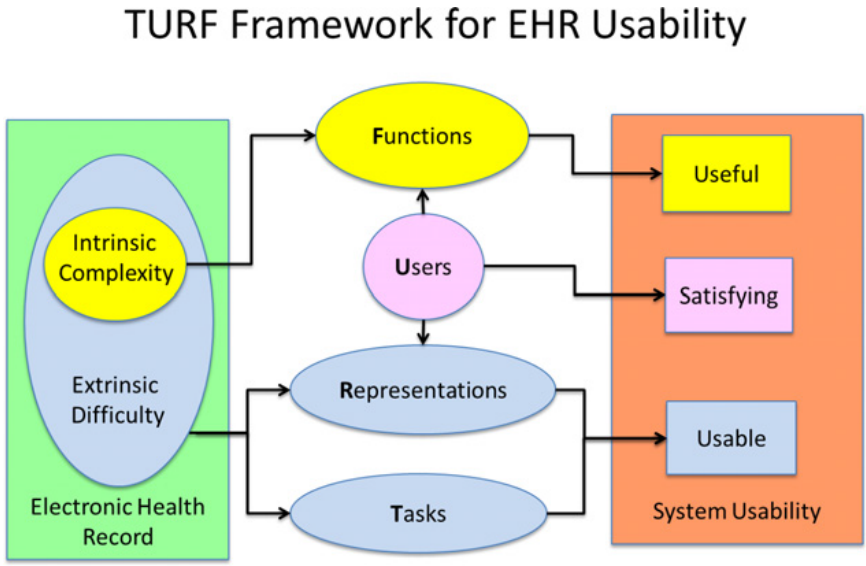


Figure 1. The TURF framework of EHR usability. See text for details.

TURF is an expansion of the UFuRT framework developed earlier in our research [28, 41, 42] and based on work-centered research [27, 28, 41, 43]. TURF is proposed as a framework for: 1) describing, explaining, and predicting usability differences in terms of the representation effect; 2) defining, evaluating, and measuring usability objectively; 3) designing built-in good usability; and 4) developing EHR usability guidelines and standards. We focus here on the first three aspects. We are in the process of developing a software application that implements a subset of TURF features to partially automate usability evaluation processes, measure usability along several metrics, and analyze usability and patient safety patterns. In the future, we plan to use TURF to develop EHR usability guidelines and standards.

3.1. User analysis

User analysis is the first step in applying TURF to the design and evaluation of usability, providing user information to conduct function, representation, and task analyses. User analysis is the process of identifying types of users and their characteristics. For EHR, types of users include physicians at various levels (e.g., attending, fellow, resident, medical student, etc.) and in various specialty areas (family practice, intensive care, dermatology, surgery, etc.), nurses with different specializations, medical technicians, medical staff, patients and family

members, and so on. User characteristics include experience and knowledge of EHR, knowledge of computers, education background, cognitive capacities and limitations, perceptual variations, age-related skills, cultural background, personality, etc. User analysis can help design systems that have the right knowledge and information structure that match its users. There are many established methods for user analysis (e.g., [44]), which we will not duplicate here.

3.2. Function analysis

3.2.1. Work domain ontology

Function analysis is the process of identifying a work domain's abstract structure: its ontology [27, 31]. Ontology is the basic structure of the work a system and its human users performs. It is an explicit, abstract, implementation-independent description of the work, describing essential requirements independent of technology systems, strategies, or work procedures. Work domain ontology describes the inherent complexity of the work, separates work context (physical, social, organizational, etc.) from the inherent nature of the work. It also supports identification of overhead activities non-essential for the work but introduced due to the way the system is implemented. Work domain ontology is inherent to the work's context, application technology, and cognitive mechanisms. If the system does not support the ontology of the work, the system will fail, regardless of a large collection of functions, fancy and cutting-edge features, and purely technical merits.

Work domain ontology has four components: goals, objects, operations, and constraints. Operations are performed on objects under constraints to achieve goals. Consider the following example: Dr. Townshend prescribes a 90-day supply of Metformin 500 mg tablets by mouth twice daily to patient John Joe, who is pre-diabetic with a glucose level of 110. In this example, the *goal* is "treating high glucose level in a pre-diabetic patient." The *operation* is "writing a medication prescription." The *objects* for this operation include the patient's name, doctor's name, diagnosis, medication name, dosage, frequency, duration, route, etc. *Constraints* include dependency relations between operation and objects (e.g., the operation "write a medication prescription" and objects "Metformin" and "500 mg"), between objects (e.g., "glucose level" and

"Metformin"), and between operations (e.g., "write a prescription" and "modify problem list").

Work domain ontology is usually a hierarchical structure based on operations with each operation having a set of sub-operations. For example, the operation, "maintain active medication list" has four sub-operations: record medication, modify medication, retrieve active medications, and retrieve medication history.

The word "function" in function analysis is based on the fact that operations in the work domain ontology specify the functionality (or utility) of the system. Identification of operations and their relations in the function hierarchy is the most important task for establishing a work domain ontology. For this discussion, a function is equivalent to an operation.

3.2.2. Functions as measures of usefulness

For EHR usability design and evaluation, one important task is to evaluate the functionality of the EHR system in the context of user-meaningful operations—those that can be carried out by users, or potentially built into the application through automation, or jointly by users and the application. We call a set of functions that are implemented in an EHR system the Designer Model. Identifying functions in the Designer Model is relatively unambiguous as the functions in an EHR system are defined as all user-actionable operations, such as clicking the "add medication" button, typing a medication name, etc. The set of functions that are wanted by users is called the User Model. Identifying functions in the User Model involves interviews and surveys. User Model ambiguities can be minimized through systematical application of ontology engineering methods and qualitative methodologies. The set of functions actually used in real activities by users is called the Activity Model. Functions in the Activity Model are typically identified through ethnography and extensive qualitative data analyses. For an ideal design with perfect functionality, all three models should be identical. However, discrepancies of functions across the three models are almost always present. This is the subject of a function analysis and offer opportunities for design improvement. One recent doctoral graduate in our lab developed a methodology for reducing function discrepancies across the

three models as part of her doctoral dissertation [45]. She described function discrepancy as seven areas in the Venn diagram of Figure 2.

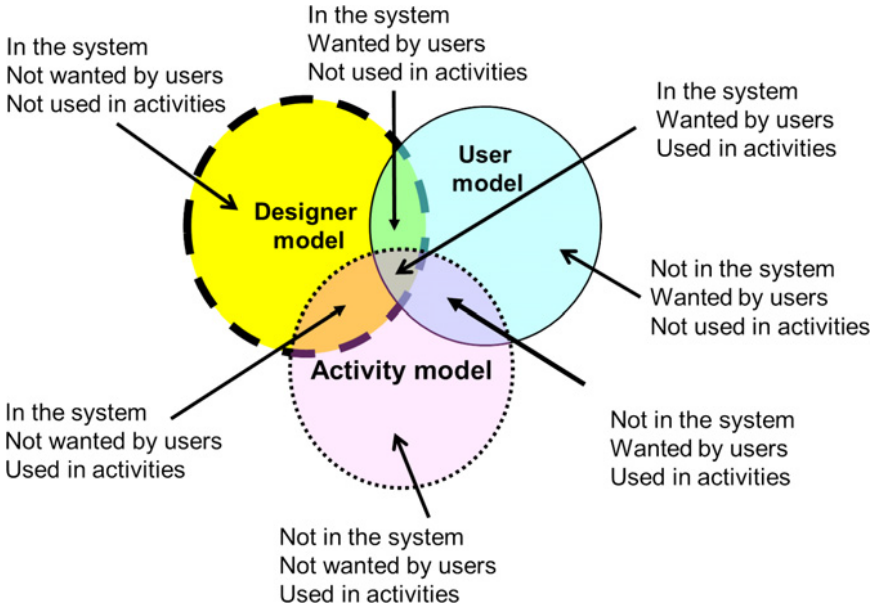


Figure 2. A conceptual model of function discrepancies [45].

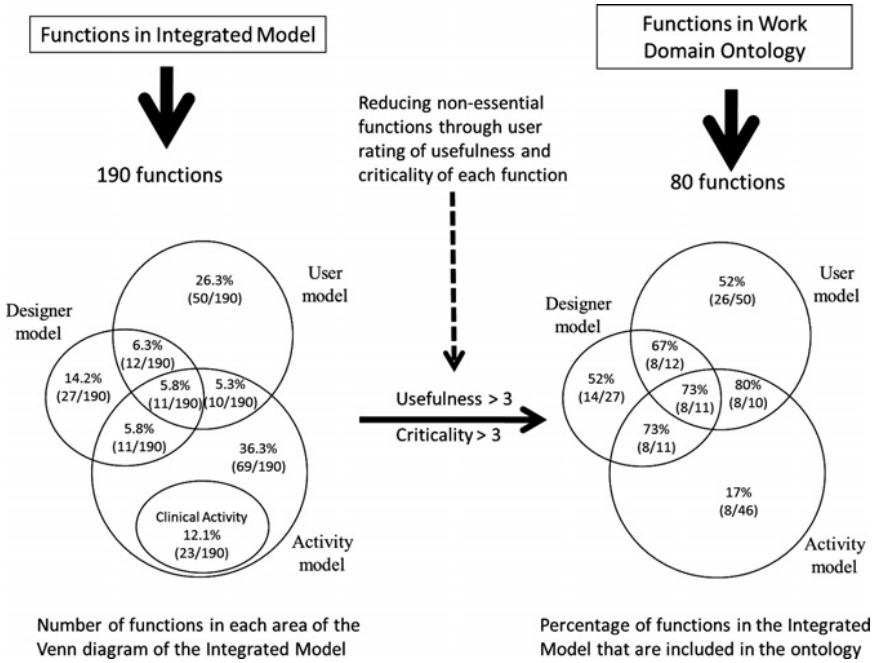


Figure 3. The left Venn diagram of an Electronic Dental Records system shows the number of functions in each area defined in Figure 2. The right Venn diagram shows the percentage of functions in each area on the left included in the work domain ontology (defined by the set of functions rated 3 or above for usefulness and criticality on a 1–5 Likert scale by users) [45].

The left side of Figure 3 shows the number of functions in each area of the Designer, User, and Activity Models of a small Electronic Dental Records (EDR) system. The Designer Model has 60 functions and was obtained through a complete system walkthrough. The User Model has 80 functions and was developed by conducting interviews and surveys with end users. The Activity Model has 97 functions and was developed by doing a field study involving many sessions of shadowing and observation (for details, see [45] of the end users in the clinics. The Activity Model includes 23 clinical functions (e.g., injecting medication) that were not directly relevant for the EDR. Functions in the three models were matched and merged into 190 functions in an Integrated Model (167, excluding the 23 clinical functions) and given in a survey to end users who rated each function on a 1-to-5 Likert scale for usefulness and criticality. Eighty functions received an average rating of 3 or above for both usefulness and criticality (see the right side of Figure 3), operationally defined as domain functions—functions in the work

domain ontology of the EDR. Functions with ratings below 3 were called overhead functions.

Figure 3 reveals some interesting points. First, 73% of functions in the system wanted by users and used in activities are included in the ontology. This indicates a function likely part of the ontology in all three models. Second, about half (52%) of the functions in the system, but not wanted by users and not used in activities, are included in the ontology. This means that some functions offered by the vendor are useful functions users are not aware of and do not use, but represent innovations by the vendor. On the other hand, the other half of functions in the same category are not considered useful and excluded from the ontology. The excluded functions are overhead and, therefore, not essential to the work domain, potentially adding to the intrinsic complexity of the system (see Figure 1). Third, 80% of the functions wanted by users and used in activities are included in the ontology. In addition, 52% of the functions wanted by users, but not in the system and not used in activities, are also included in the ontology. This means there are important domain functions wanted by users, but are not available in the system and should be added in future software updates. Fourth, only 17% of functions used in activities, but not in the system and not wanted by users, are included in the ontology. This means that most functions in this category are considered by users as inappropriate for inclusion in the system, at least for the time being. Most of the functions (about three quarters) in two or more models are included in the ontology, which means functions with cross-model agreement are likely to be the functions that are useful.

From these analyses, we can define three metrics for usefulness, one of the three dimensions of usability (see Table 1).

1. *Within-model domain function saturation*: This is the ratio of the number of functions in the Designer Model included in the ontology over the total number of functions in the Designer Model. For the EDR system in Figure 3, the ratio is 38/60 = 63%. This means 63% of the functions in the EDR are considered useful by users, and 37% are overhead functions not useful.

2. *Across-model domain function saturation*: This is the ratio of functions in the Designer Model included in the ontology over the total number

of functions in all three models (Designer, User, and Activity Models) included in the ontology. For the system in Figure 3, the ratio is 38/80 = 48%. This means the EDR system implemented about 48% of all domain functions considered useful by users.

3. *Across-model function saturation*: This is the ratio of all functions in the Designer Model over the total number of functions in all three models (Designer, User, and Activity Models). For the system in Figure 3, the ratio is 60/190 = 32%. This ratio means the EDR system implemented about 32% of all functions proposed by its designers, wanted by users, and used in activities. This ratio does not exclude non-domain (overhead) functions in the three models considered not useful by users. This ratio is similar to the second one, although is not as direct a measure of usefulness. The advantage of this third ratio is that it does not require the additional work of integrating the functions of all three models and conducting a survey among users to determine which functions should be included in the ontology.

There are a few points about these usefulness metrics that warrant further discussion. First, function saturations in the User and Activity models are empirical data collected from interviewing, surveying, and observing users. Second, whether a function is useful is determined by two ratings on 1 to 5 Likert scales by users: usefulness of the function and criticality of the function. The threshold for inclusion as a domain function in the work domain ontology in Chen's study [45] is the midpoint of 3 on the scale. This threshold can be adjusted to either exclude more functions, or include more functions into domain functions. In addition, the threshold could be based on either the usefulness or the criticality measure alone, or it could be based on additional measures depending on purpose of the evaluation.

3.2.3. Domain vs. overhead functions through expert review

In the last section we discussed the relationship of functions in the three models: functions available in an EHR system, functions wanted by users, and functions actually used in real activities. The method used to conduct the analysis described in the previous section is based on empirical data, usually requiring significant effort and resources. In the next section, we focus on the functions in the Designer Model only and

describe a relatively more efficient expert review method developed evaluating the Armed Forces Health Longitudinal Technology Application (AHLTA) EHR system [46].

The method started with identification of the hierarchy of the EHR system. System hierarchy was created by visually inspecting user interface items from top to bottom and left to right. Each interface item (label, field, drop-down menu etc.) was coded with a unique identifier, such as 2.3.1 for the first item on Level 3 of the third item on Level 2 of the second item on Level 1. AHLTA has six levels and almost two thousand items. The first three levels of the AHLTA system hierarchy are shown in Figure 4.

Each interface item was classified as an Object or Operation (i.e., function). An object was defined as an interface item on which no user actions could be performed. An operation was defined as an interface item on which a user action could be performed. Each operation was further classified as either a Domain Function or Overhead Function. A domain function was an operation inherent in and necessary for the work domain rather than dependent on artifacts or interfaces. An overhead function was an operation introduced to deal with specific implementations of user interface rather than the work domain. Figure 5 shows that among 1,996 interface items identified in the AHLTA hierarchy, 61% were Operations and 39% as Objects (kappa > 0.6 for inter-rater reliability between the two evaluators). Of 1,218 items classified as Operations, 76% were identified as Domain functions and 24% as Overhead functions (kappa > 0.6 for inter-rater reliability between the two evaluators).

From this study we can obtain usefulness metrics in a more efficient manner: percentage of domain functions in the Designer Model over all functions in the Designer Model through expert review. In Section 3.2.2, percentage was obtained through an empirical data collection process. From the AHLTA study, percentage was obtained through assessment by two expert evaluators. Although the process still requires significant effort, it is more efficient than a method using empirical data collection. From this expert review process, the usefulness metric for the AHLTA EHR as defined by the percentage of domain functions in the Designer Model over all functions in the Designer Model was 76%. Detailed

results are shown in Figure 6, which shows that most functions in the "summary" subsection are overhead functions and not useful, whereas most functions in the "readiness" subsection are useful domain functions.

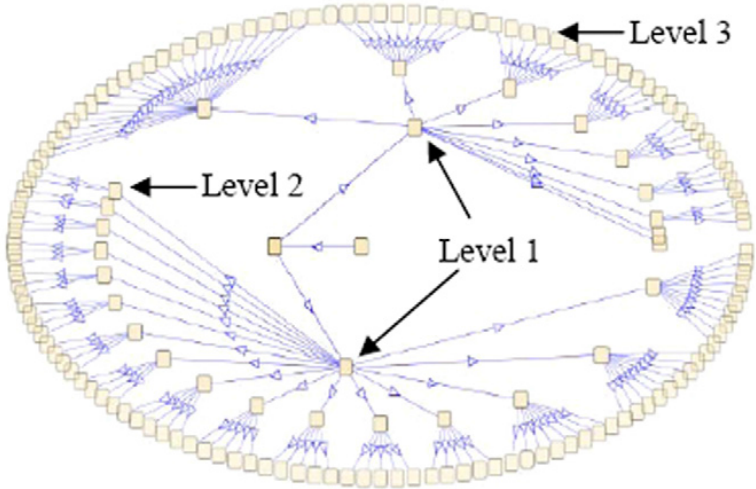


Figure 4. Visualization of the top three levels of the six-level hierarchy AHLTA user interface [46].

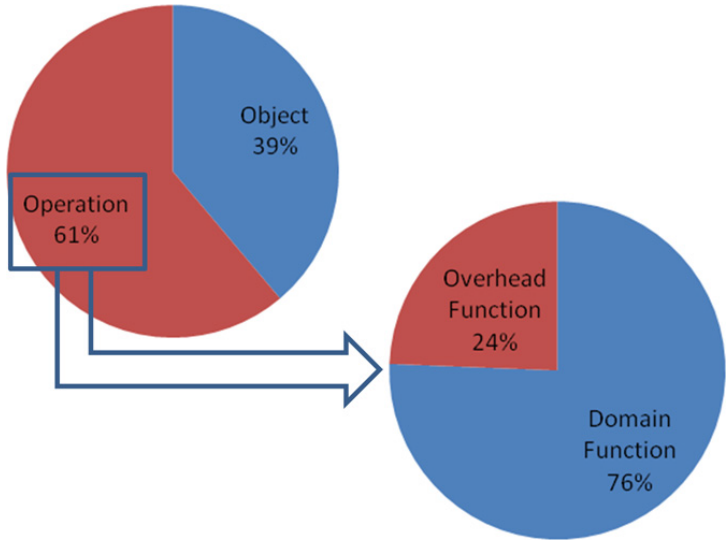


Figure 5. Among 1,996 interface items in the AHLTA EHR system, 39% were objects and 61% operations. Out of the 1,218 operations, 76% were domain functions and 24% overhead functions [46].

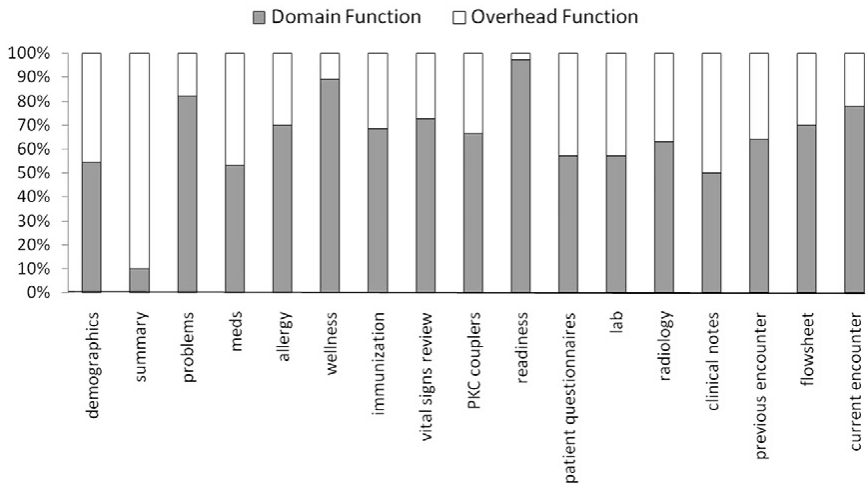


Figure 6. Percentage of domain vs. overhead functions in each of the subsections of the AHLTA patient record section [46].

3.3. Representation analysis

Representation analysis is the process of evaluating the appropriateness of representations for a given task performed by a specific type of user such that interaction between users and systems is in a direct interaction mode [47]. Representation analysis is based on the representation effect described in Section 3 [35, 36, 38, 39, 43]. Different representations of a common abstract structure can generate dramatically different representational efficiencies, task difficulties, and behavioral outcomes. A major type of representation analysis is a comparison of a representation with isomorphic representations of the same structure and determination of whether it is efficient for the task and the user. This is described in Section 3.3.1. Another representation analysis is based on the affordance of interface items, which is described in Section 3.3.2. Expert review of usability violations against well established principles includes various types of representation analyses, and described in Section 3.3.3. There are many other types of representation analyses, some of which are being developed and evaluated in our EHR Usability Lab at the National Center for Cognitive Informatics and Decision Making in Healthcare (NCCD).

3.3.1. Isomorphic representations

Identifying and generating isomorphic (functionally equivalent, but computationally different) representations is a major type of representation analysis. Work domain ontology is a common abstract structure that can be implemented many ways. For example, for the function "write medication prescription," it can be represented in a paper-and-pencil format, in a telephone call to the pharmacy, or a task on computer in an EHR. Each representation has different consequences for user performance. There is no best representation of a function for all tasks for all users. However, an efficient representation, or a set of efficient representations of a given function, can often be identified for a specific task for a specific user under specific constraints. In this section, we describe a previous study of relational information displays [38] to demonstrate how to use isomorphic representation as a representation analysis. Relational information displays are a significant category of displays in EHR systems.

Figure 7 shows the representation taxonomy of relational information displays, displays that represent relations such as tabular and graphic displays [38]. The taxonomy is a hierarchical structure. At the level of dimensionality, different relational information displays can have different numbers of dimensions, e.g., 2-D, 3-D, 4-D, etc. At the level of scale types, dimensions of a relational information display can have different scale types: ratio (R, such as length), interval (I, such as time), ordinal (O, such as ranking of movies by number of stars), and nominal (N, such as names of people) scales. At the level of dimensional representation, each scale type can be implemented by different physical dimensions. In Figure 7, for example, ratio scale is represented by length, distance, and angle; interval scale by position and orientation; ordinal scale by cell position; and nominal scale by shape, direction, texture, and position. With these physical dimensions, the scale combination R–R can be represented by length–length (Rectangle, Cross), length–angle (Coxcomb, Polar Plot), distance–distance (Line Graph, Cartesian Plot), and so on. The scale combination R–I can be represented by length–position (histogram), length–orientation (glyph, polygon), distance–position, and so on. The scale combination R–N can be represented by length–position (segmented and vertical bar charts), length–direction, angle–direction (pie chart), and so on. The scale combinations O–O–N

can be represented by CellPosition–CellPosition–shape (table, matrix), position–position–texture (network), and so on.

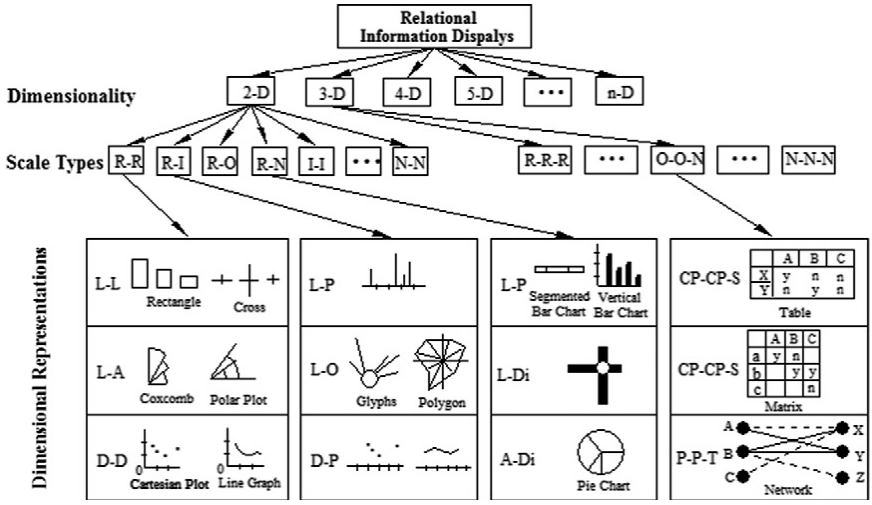


Figure 7. A representation taxonomy of relational information displays [38].

This taxonomy of relational information displays can be used for two types of representation analysis for EHR. The first is to analyze the dimensions of component displays (e.g., a flow sheet table in an EHR system) and evaluate whether each dimension in the display is appropriately represented according to the taxonomy. The second analysis is using the taxonomy to generate new designs. Once dimensions of data are given (e.g., various vital signs), isomorphic displays for the data can be systematically generated by using the taxonomy to match the scale types of the dimensions. Because displays in the taxonomy are optimized for user performance, displays with good usability can be generated for the design of the EHR.

Relational information displays are only part of EHR user interfaces. Other EHR user interfaces are more granular or more abstract than relational information displays. Developing a comprehensive library of EHR user interface representations along with mappings to tasks and users is an ongoing effort in the NCCD EHR Usability Lab.

3.3.2. Affordance of interface items

Affordance is a concept developed by Gibson [48, 49] in the study of visual perception. For user interfaces, affordance is a set of allowable

actions specified by the display coupled with knowledge of the user [50, 51] and indicates the ability to perform user actions. For example, a well-designed button on the display affords clicking. A hyperlink embedded in text without any visual cues (e.g., underlined blue text or a distinct color), even if it supports clicking, does not afford the action because the user cannot perceive it through its visual cues.

In our evaluation of the AHLTA interface, we determined the degree of affordance for each operation in a module. Two evaluators independently analyzed each operation and determined degree of affordance. Any discrepancies in ratings were resolved by consensus after further discussion. Operations were rated as follows:

1. *High affordance*: Operation can be perceived by using external cues in the interface.
2. *Medium affordance*: Operation can be perceived by external cues in the interface and internal knowledge of the application.
3. *Low affordance*: Operation can be perceived mainly by using internal knowledge of the application.

Table 2
Degrees of affordance in an AHLTA EHR module.

	# Of operations	Percentage
High Affordance	158	90
Medium Affordance	15	8
Low Affordance	3	2
Total	176	100

Table 2. Degrees of affordance in an AHLTA EHR module.

The results (Table 2) suggest operations in the AHLTA interface have a high degree of affordance and can be perceived using external cues. Only a few operations required internal memory, suggesting the interface items in AHLTA are well designed and that users can easily perceive what actions can be performed using the interface.

We plan to extend our representation analysis to classify degree of correct or incorrect mappings between AHLTA displays and specific

tasks. Ideally, information perceivable from a display should exactly match information required for the task—no more and no less. In other words, tasks assigned to a display should be the tasks afforded by the external representations of the display. Likewise, displays assigned to a task should be the displays whose external representations support the task [51].

3.3.3. Representation analysis through expert review of usability principles

Expert review of violations against well-established usability principles, often called heuristic evaluation [52-55], is a large portion of representation analysis. Heuristic evaluation is an easy-to-use, easy-to-learn, discount usability evaluation technique for identifying usability problems of a product in a timely manner with reasonable cost. The technique requires a few evaluators to independently apply a set of usability principles to a product, identify violations of principles, and assess severity of each violation. In an early project, we integrated, revised, and expanded the ten heuristics by Nielsen [54] and the eight golden rules by Shneiderman [56] to form 14 principles customized for the health domain [57]. We since applied these fourteen principles to a variety of healthcare domains [57-60].

The 14 principles:

1. [Consistency] Consistency and standards in design.
2. [Visibility] Visibility of system state.
3. [Match] Match between system and world.
4. [Minimalist] Minimalist design.
5. [Memory] Minimize memory load.
6. [Feedback] Informative feedback.
7. [Flexibility] Flexibility and customizability.
8. [Message] Good error messages.
9. [Error] Prevent use errors.
10. [Closure] Clear closure.
11. [Undo] Reversible actions.

12. [Language] Use users' language.

13. [Control] Users are in control.

14. [Document] Help and documentation.

The first six (Consistency, Visibility, Match, Minimalist, Memory, and Feedback) concern representation properties of user interfaces and are considered a type of representation analysis. Figure 8 shows the evaluation of the AHLTA EHR with the 14 principles. Evaluation was performed by three independent evaluators and results integrated into a master list of all violations. Then, each evaluator independently rated each violation for severity on a scale of 1 to 4 (1 = cosmetic; 2 = minor; 3 = major; 4 = catastrophic). Ratings were averaged as shown in Figure 9. Figure 10 shows detailed results of where violations occurred in the Health History module of the AHTLA. Violations were documented in detail and recommendations for generated.

Representation analysis through expert review of usability principles is an efficient method capable of a large range of usability violations. It usually generates informative results for users and designers. However, as it currently stands, it is not a well-organized, systematic method that can generate consistent and reliable results for comparison of different representations. An ongoing effort at the NCCD EHR Usability Lab is developing and validating a reliable, systematic, and operationalized process for a subset of usability principles relevant to representations.

3.4. Task analysis

Task analysis is loosely defined in the literature [61, 62]. For EHR usability, we define task analysis as the process of identifying steps needed to carry out an operation using a specific representation, relationships among these steps, and the nature of each step. Our definition of task analysis is based on the GOMS approach [63, 64]. An important point about cognitive task analysis is that steps include not only physical steps but also mental steps. By considering mental steps, we identify cognitive factors that make a task easy or difficult [43, 65]. Steps needed to carry out the same operation are different with different representations (e.g., using a bar chart vs. using a spreadsheet to find the highest glucose level of a patient over three years). An important objective of task analysis is finding which representation is best suited for each task, why it is better,

and how to generate a better representation. By performing task analyses for the same operation implemented in different user interfaces, we can compare user performance associated with different user interfaces in terms of time on task, number of steps and mental effort, all of which are metrics of efficiency for usability (see Table 1).

We conducted a series of task analyses for many EHR systems. In the following, we describe a task analysis study for the AHLTA EHR system [66]. We used the Keystroke Level Modeling (KLM) to estimate time on task, task steps, and mental effort for fourteen prototypical use cases. KLM is a well-established and validated method that estimates performance level by experts [63, 67]. Over one hundred research publications have shown performance levels generated by KLM are within 20% of expert performance through empirical studies [63, 68]. The 14 use cases, provided to us by expert AHLTA clinician users, were:

1. Enter HPI (History of Present illness)
2. Enter PMI (Present Medical Illness)
3. Document social history
4. Document family history
5. Enter vital signs
6. Enter order consult
7. Document coding of procedures
8. Entering the lab order
9. Document Instructions—Other Therapies
10. Order radiology study
11. Document comments in A/P diagnosis
12. Review coding of medical encounter
13. Document follow-up plan
14. Associate orders/medication/labs

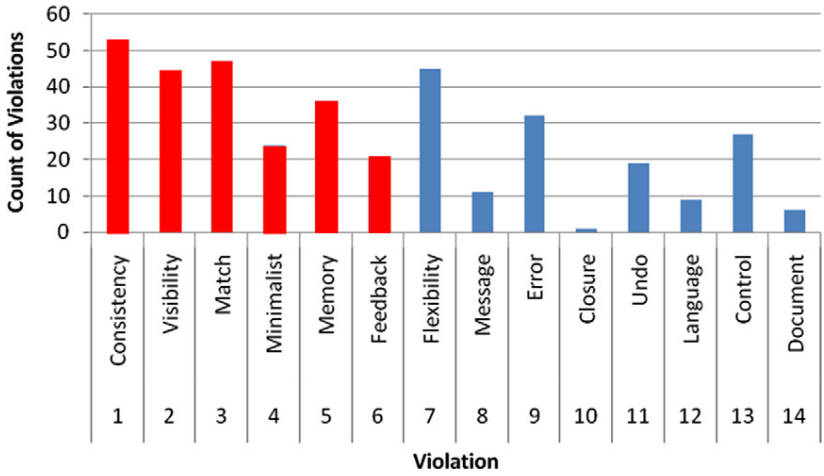


Figure 8. Usability principle Violations for the AHLTA EHR. The first six principles (consistency, visibility, match, minimalist, memory, and feedback) concern representation properties of user interfaces and are considered a type of representation analysis.

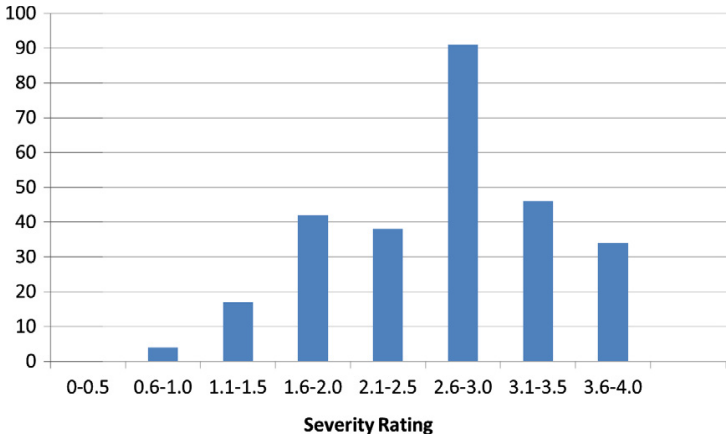


Figure 9. Violation severity ratings for the AHLTA EHR.

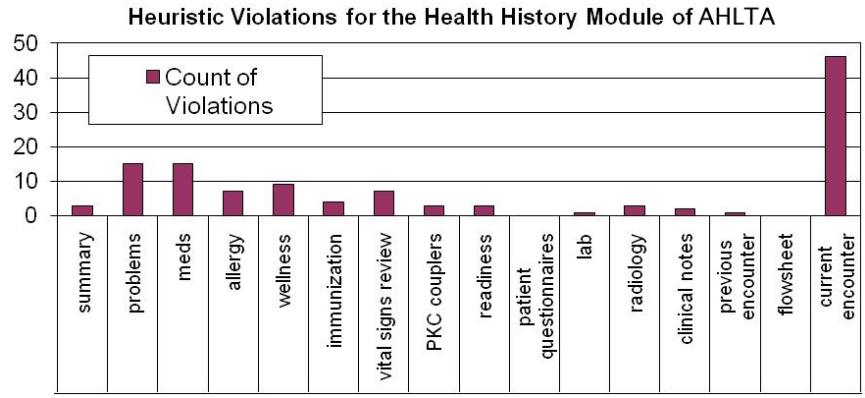


Figure 10. Violations of usability principles in the Health History module of the AHLTA EHR showing most of violations are in the current encounter section.

Figure 11 shows the KLM analysis of the 14 use cases. Each case was rated by two evaluators. Inter-rater reliabilities were good for all 14 use cases (kappa > 0.6 for all use cases). The number of steps varied from as few as 43 for Use Case 9 (Document Instructions – Other Therapies) to as many as 466 for Use Case 5 (Enter Vital Signs). Time on task shows similar patterns: 34 steps for Use Case 12 (Review Coding of Medical Encounter) and 389 steps for Use Case 5 (Enter Vital Signs). On average, 37% of task steps were mental and 50% of the time was spent on mental steps.

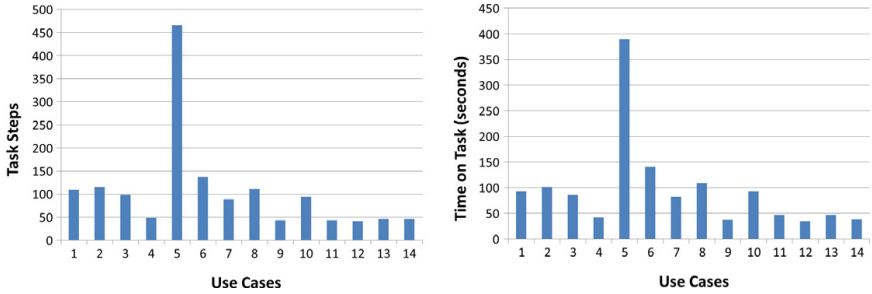


Figure 11. The left panel shows the number of task steps needed for each of the 14 use cases. The right panel shows time on task (from [66]).

In the AHLTA study, three metrics for efficiency measure of usability, time on task, task steps, and mental effort were estimated using KLM modeling (Table 1). These are expert performance levels following optimal paths of tasks, providing a set of benchmarks for EHR usability. Performance levels by actual users in real clinical environments will be

different from estimated expert performance levels with metrics collected through more effortful user testing.

Although KLM is an excellent method for estimating expert performance levels, it is not a straightforward or efficient process for controlling inter-rater reliabilities. To address this issue, we adopted the *CogTool* [68] method for usability evaluation of EHR systems. *CogTool* is based on KLM but incorporates the Act-R model of human cognition [69, 70]. *CogTool* increased the accuracy of KLM and has been reported to be within about 10% of empirical data [71]. In addition to better accuracy, *CogTool* does not require two evaluators to achieve significant inter-rater reliability because estimates of performance levels are carried by the model itself. Thus, *CogTool* provides more accurate, more reliable, and more objective estimates of expert performance levels on skilled tasks.

3.5. TURF in redesign of EHR user interface

TURF is not only a framework for evaluating the usability of existing EHRs, it is also a method for redesigning EHRs for better usability. In a small demonstration project [72], we applied TURF to evaluate the usability of a module of the OpenVista EHR for NIST Test Procedure §170.302(e): Maintain Active Medication Allergy List with three subtasks (Add, Modify, and Review Allergy). We performed user, function, representation, and task analyses; identified usability problems; developed new design mockups; and then compared the original product and a new design using KLM and function analysis. Figure 12 shows the results of the KLM task analysis: dramatic improvements to both time on task and task steps: 187 total steps in the original product to 79 total steps for a new design; 199 seconds for the original product to 82 seconds for the new design. The biggest improvement was for the Modify Allergy subtask, with improvement from 91 to 14 steps and 97 to 10 seconds. Function analysis showed similar patterns. Overhead functions reduced from 99 in the original design to 19 in the new design. Domain functions increased from 28 in the original to 53 in the new design.

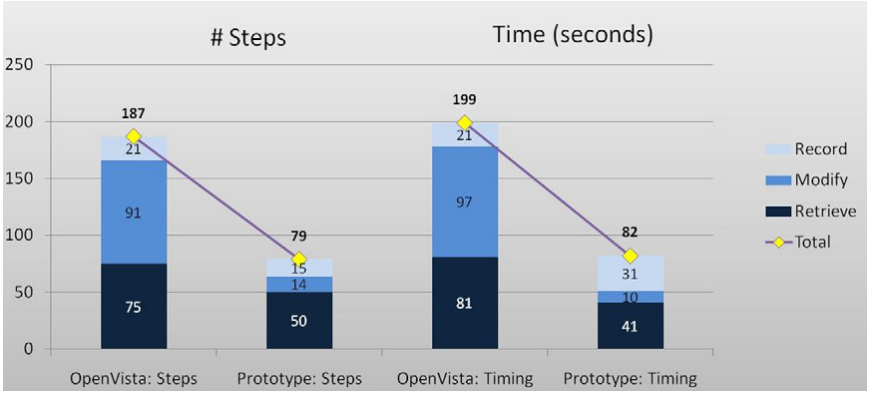


Figure 12. Improvements for time on task and task steps after the redesign of an OpenVista module: 187 total steps from the original product to 79 total steps for the new prototype design, and 199 seconds in the original product to 82 seconds in a new prototype design (from [72]).

3.6. Environmental factors and workflow for usability

So far we have presented TURF and case studies for idealized, uninterrupted EHR tasks by individual users. EHR systems, like many products, are used in real world settings typically interruption-laden, unpredictable, stressful, and involving many other factors, such as organizational, social, physical, spatial, temporal, financial, and historical influences. All of these can contribute to the representation effect in various ways and should always be considered in the design and evaluation of EHR usability.

For example, interruption and multitasking are routine in real clinical settings [73-75] and can cause medical errors [76]. A measure of an EHR's ability to handle interruptions and multitasking should be included as part of usability. Workflow across multiple people and artifacts is a major usability factor that we have not discussed under TURF; we only discussed task sequences within a task performed by an individual user. NCCD has developed a framework and software modeling tool for capturing, analyzing, and predicting workflow across team members in healthcare settings [77] (Chapter 11). The match between information flow and workflow is a key principle of usability for user tasks [78]. If the structure of an EHR does not match the workflow of clinical work, then its users have to perform additional overhead tasks to work around, or follow a sub-optimal workflow [79]. In the future, we

plan to expand the TURF framework to cover interruptions, workflow, team dynamics, and other socio-technical factors of usability.

Discussion and conclusion

TURF is a unified framework of EHR usability for: 1) describing, explaining, and predicting usability differences; 2) defining, evaluating, and measuring usability objectively; 3) designing built-in good usability; and 4), developing EHR usability guidelines and standards. We approached usability as a human performance issue in terms of the representation effect. Then we defined usability around the representation effect along three dimensions (useful, usable, and satisfying) and listed a set of representative measures for each dimension. Most of these are evidence-based, repeatable, and objective measures established over fifty years of research in cognitive psychology and human factors study. Unlike many approaches to usability, we consider usefulness an important component in addition to usableness and satisfaction dimensions. Usefulness is often more important than usableness for a product's success or failure.

Usability can not only be defined under a coherent, unified theoretical framework, it can be measured objectively and systematically. We presented a set of studies to demonstrate how EHR usability could be evaluated and measured in a scientific and systematic way. We also demonstrated how TURF can be used as a method to redesign products to improve usability. TURF's theory-based approach, systematical method, and operationalized process are essential tools for developing EHR usability guidelines.

Acknowledgment

This project was supported by Grant No. 10510592 for PatientCentered Cognitive Support under the Strategic Health IT Advanced Research Projects (SHARP) from the Office of the National Coordinator for Health Information Technology. We would like to thank our colleagues Vimla Patel, Keith Butler, Todd Johnson, Debora Simmons, Amy Franklin, and Ron Gimbel for many valuable discussions, our clinician collaborators Brent King, David Robinson, and Emeka Okafor for providing clinical expertise and access to clinical facilities on a

variety of projects, our former and current trainees Jung-Wei Chen, Min Zhu, Zhen Zhang, Himali Saitwal, Louis Lee, Yuanyuan Li, Craig Harrington, and many others for carrying out various studies cited in this paper, and lab members Krisanne Graves and Paul Wood for keeping everything together and on time. Nearly all of the cited studies in this paper were previously published in various journal and proceedings by our team members.

A version of this chapter originally appeared in the Journal of Biomedical Informatics, volume 44, Zhang, J. & Walji, M. F., TURF: Toward a unified framework of EHR usability, 1056-1067.

3: SYFSA Framework for Systematic Yet Flexible Systems Analysis

Todd R. Johnson, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Eliz Markowitz
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Elmer V. Bernstam, MD, MSE
The University of Texas Health Science Center at Houston School of Biomedical Informatics

The University of Texas Health Science Center at Houston Department of Internal Medicine

Jorge R. Herskovic, MD, PhD
The University of Texas MD Anderson Cancer Center

Harold Thimbleby, PhD
FIT Lab—Future Interaction Laboratory

ABSTRACT

Although technological or organizational systems that enforce systematic procedures and best practices can lead to improvements in quality, these systems must also be designed to allow users to adapt to the inherent uncertainty, complexity, and variations in healthcare. We present a framework called Systematic Yet Flexible Systems Analysis (SYFSA) that supports the design and analysis of Systematic Yet Flexible systems, whether organizational or technical, by formally considering tradeoffs between systematicity and flexibility. SYFSA is based on analyzing a task using three related problem spaces: the idealized space, the natural space, and the system space. The idealized space represents best practice—how a task is to be accomplished under ideal conditions. The natural space captures task actions and constraints on how the task is currently done. The system space specifies how a task is done in a redesigned system, including how it may deviate from the idealized space and how the system supports or enforces task constraints. The goal of the framework is to support the design of systems that allow graceful degradation from the idealized space to the natural space. We demonstrate the application

of SYFSA for the analysis of a simplified central line insertion task. We also describe several information theoretic measures of flexibility that can be used to compare alternative designs, measure how efficiently a system supports a given task, relative cognitive workload, and learnability.

INTRODUCTION

Efforts to improve healthcare quality have led to an increased push to develop and adopt systems that enforce or encourage consistent processes based on best practices and evidence-based medicine. These efforts follow similar successful practices in other safety-critical industries, such as aviation and nuclear power. Within healthcare, these efforts include clinical guidelines, structured documentation, standardized terminologies, decision support systems, checklists, and policies.

Although systems that enforce or encourage consistency can improve safety and efficiency, healthcare is filled with complexity, variations, and exceptions not easily captured by idealized processes. Systems too rigid to support deviations can lead to decreases in quality, caregiver resistance and creative workarounds that lessen the positive effects of best practices [80].

Hollnagel's efficiency-thoroughness tradeoff (ETTO) principle is an informal way to express the tradeoff between systematicity and flexibility [81]. Recognition of similar tradeoffs in other industries led to the design of *Systematic Yet Flexible* (SYF) systems [82] in which the system supports and sometimes enforces a systematic approach, while allowing flexibility. Thimbleby [83] argued that user interfaces are easier to use when they are "permissive" (i.e., giving users flexibility and, hence, lowering learning costs), but this is an informal treatment. Norman [84] emphasizes the role of design constraints and forcing functions in user interfaces, but not how to design the appropriate blend.

Although there are general design goals for SYF systems [82], there are no analytic frameworks that allow one to analyze tradeoffs and determine the appropriate blend of systematicity and flexibility. Without analytic frameworks, organizations (or system developers) will inevitably make arbitrary, sometimes sub-optimal, design choices. The usual response is to require iterative design, a period of repeated

implementation and evaluation to guide improved re-implementation of the procedures; essentially a "trial and error" design process.

We present here an analytic framework for designing SYF systems (organizational or technical) by formally considering tradeoffs between systematicity and flexibility. We propose that the ideal SYF system supports graceful degradation from idealized practices to those better fitting the situation at hand. The framework, which we call Systematic Yet Flexible Systems Analysis (SYFSA), is based on analyzing a task using three related problem spaces: the *idealized space*, the *natural space*, and the *system space*.

The idealized space represents the best and most efficient practice—how the task should best be accomplished assuming that only actions ultimately leading to a goal state are taken and that all logical task constraints are met (i.e., the least number of actions will be taken to achieve the goal). For example, the idealized space for choosing a medication includes a number of constraints, such as the medication is therapeutically appropriate, has the correct dose and route, is safe, is available for purchase in the form and dose prescribed and within the required timeframe, and is as economically efficient as possible.

The natural space captures the task actions and constraints on those actions imposed by the physical world. For example, if the natural space is a paper-based, handwritten prescription we see that it enforces almost none of the idealized constraint. It is too flexible. However, this flexibility allows a physician to use non-standard formulations and dosing regimens to better personalize care and easily prescribe new medications that may not yet be in more systematic information technology (IT) based ePrescribing systems.

The system space specifies how the task is done in a redesigned or newly designed system, including how it may deviate from the idealized space and how the system supports or enforces constraints in the idealized space. A system space for ePrescribing explicitly considers the constraints of the idealized prescribing space, supports known constraints, while recognizing the need to cope with the inevitable exceptions and variations that are common in healthcare.

SYFSA is a design and analysis framework, not a set of prescriptive guidelines or principles for producing SYF systems. Prescriptive guidelines give explicit design advice, but usually at a high level of abstraction that leaves considerable details underspecified. For instance, one of Perer and Shneiderman's guidelines for SYF systems that we discuss below is to allow the user to "See an overview of the sequential process of actions," [82] but the guideline does not help designers decide which of many possible sequences to highlight. In contrast, SYFSA's primary value as a design and analysis framework is to allow stakeholders to explore tradeoffs in systematicity and flexibility by making constraints (and lack of constraints) on actions and sequences of actions an explicit part of the design and evaluation process. SYFSA forces designers and others involved in the design or evaluation process to think about the constraints in each of the spaces and whether a specific system design supports those constraints. It is then up to the designer to use the results of the analysis to inform system design. Returning to Perer and Shneiderman's example, SYFSA can help designers decide which sequence of actions to highlight.

We also propose three quantitative, information-theoretic measures of task flexibility that allow designers to compare the flexibility of alternative system designs and how closely these designs match the idealized flexibility required to complete a task. These measures are motivated by an intuitive notion of flexibility, whereby a task that can be done by carrying out actions in any order has maximum flexibility and a task that can only be done with a specific sequence of actions has the least flexibility.

BACKGROUND

Flexibility characteristics

The concept of system or process flexibility has been explored for at least 30 years in a number of fields, including chemical process engineering [85], manufacturing design [86, 87] and more recently business process design and workflow automation systems [88-91]. A general consensus is that flexibility is a multidimensional concept, where relevant dimensions depend on the kind of process or system being analyzed and the analyst's goals. For example, Sethi and Sethi [86]

identified 11 different, but complementary, definitions of manufacturing flexibility, including production flexibility (the range of products a system can produce without need for major changes) and operation flexibility (the ability for a system to produce a product in different ways).

Despite the lack of a single, precise definition of flexibility, or even a fixed set of dimensions, there is general consensus that flexibility is the ability of a system to tolerate and adjust to variations in operating conditions. One common distinction is between short-term and long-term flexibility, where short-term flexibility is the ability to tolerate variations without changing the goal, whereas long-term flexibility is the ease with which a system can be changed to meet new goals. An example of short-term flexibility is the ability of an automotive manufacturing process to adjust to a part substitution. In contrast, long-term flexibility refers to the ease of changing the assembly line to manufacture a different vehicle.

There are often tradeoffs between different dimensions. For example, a multipurpose woodworking machine that acts as a router, planer, jointer, and table saw has a lot of functional flexibility, but because it takes time to convert from one function to another and can only perform one function at a time, a shop with a multipurpose machine loses scheduling flexibility over a shop with a dedicated machine for each function. In addition, dedicated machines often perform better (e.g., with more precision or speed) than multipurpose ones.

Some researchers argue the general definition of flexibility, with its emphasis on adapting to and tolerating variation, implies there are invariants meant to be maintained by flexible systems [92]. This implies a flexible system must be resistant to change in the same way that airplane wings must flex, but still return to their original positions. Some of the more formal definitions and approaches to measuring flexibility operationalize this concept by defining a range of operation a system must maintain in the face of variation. Flexibility is then the amount of variation that can be tolerated while maintaining operation in the desired range [85]. For example, a chemical process that works only when ambient temperature varies by no more than 5° is less flexible than one that works within a wider temperature range.

The multidimensional nature of flexibility means there are also different measures of flexibility. In a review of flexibility concepts and measures Gupta and Goyal [93] identified six different classes of measures and then further subdivided these into qualitative and quantitative measures. In chemical process design, researchers have developed a flexibility index—a single number that defines the maximum variation in a set of normalized variables that the system can tolerate while still producing the desired output.

Flexibility in healthcare

Healthcare system flexibility, including organizational and health information technology, is perhaps most similar to business process flexibility. Researchers exploring business process flexibility have discussed measures such as the number of possible initial states of a system, the number of reachable goal states, and the number of paths from some initial state to the goal states. Bider has applied mathematical systems theory to business processes [91]. However, research on business process flexibility is less mature than other domains, so the conceptual and analytical frameworks are not as well developed.

Like many industries, healthcare experienced a push to adopt and enforce consistent procedures based on best practices and evidence. While such systems can improve efficiency and safety, healthcare is complex and is not always amenable to idealized processes. Some health information systems are too rigid, leading to negative consequences, such as decreased quality, user resistance, and workarounds [17, 80, 94-97]. One study concluded many unintended consequences of clinical decision support systems (CDS) are attributable to insufficient flexibility [95]. An overly rigid system can cause medication errors by not allowing clinicians to enter atypical prescriptions [17].

On the other hand, there are also instances when errors can occur due to excessive flexibility. Consider the nurse who intended to program a pump to infuse 5 mcg/min, but accidentally selected a rate of 5 mcg/kg/min (equivalent to 350 mcg/min for a 70 kg patient). While an alert appeared, the flexible system allowed the nurse to simply bypass the warning [98].

Systematic Yet Flexible design

Perer and Shneiderman working in the context of exploratory data analysis systems proposed seven SYF design goals for systems that support exploratory data analysis [82]. The design goals enable users to: 1) see an overview of the sequential process of actions, 2) step through actions, 3) select actions in any order, 4) see completed and remaining actions, 5) annotate their actions, 6) share progress with other users, and 7) reapply past paths of exploration on new data. These design goals provide useful advice for tasks generally requiring a single sequence of actions, but they do not provide guidance on assessing task flexibility or tradeoffs among user interfaces that support different amounts of flexibility for the same task.

Cognitive Work Analysis

Cognitive Work Analysis (CWA) is a design and analysis framework created to develop systems that allow workers to flexibly adapt to unanticipated situations [99, 100]. It does this by using a number of methods to uncover the intrinsic constraints of a work domain at multiple hierarchical levels. Once constraints are visible, a designer can look for places where flexibility may be unnecessarily restricted. This gives workers flexibility to adapt to unanticipated situations. In addition, CWA emphasizes the development of information displays and controls that maximize a worker's situation awareness, readily understand an unexpected situation and respond appropriately.

Although CWA is designed to support flexible systems, it does not explicitly provide tools for analyzing tradeoffs in systematicity and flexibility. CWA emphasizes increasing flexibility to allow workers to adapt. We found only one paper that explicitly addressed flexibility in the context of CWA, but it focused on increasing flexibility [101]. It did, however, contain a brief comment that sometimes limiting flexibility can be beneficial because fewer choices can speed decision making. This was followed by a recommendation to develop interfaces presenting the most common strategy, while still allowing alternative strategies. This is the essence of an SYF system. Unlike CWA, SYFSA provides an explicit mechanism for understanding tradeoffs in flexibility and systematicity. However, CWA is highly complementary to SYFSA because it provides a number of methods and tools for uncovering, relating, and visualizing

intrinsic constraints in a work domain. A designer can use these constraints to develop the idealized and natural spaces.

Previous work on flexibility provides considerable insight on the nature of flexible systems, how to measure flexibility, and how to design user interfaces to support some kinds of flexible systems. Unfortunately, there are no clear operational definitions or measures for the kinds of flexibility that interests us in the context of healthcare. There is also no specific design process to help produce SYF systems and understand tradeoffs among alternative designs.

Types of flexibility

Based on our review, we differentiate among three types of flexibility: procedural, functional, and operational. *Procedural flexibility* is the number of ways to successfully complete a task and achieve a given goal. Procedural flexibility can result from multiple paths to a single goal state or multiple goal states each with one or more paths. *Functional flexibility* is the number of functions a system is designed to support. For example, an epinephrine auto-injector that delivers a single measured dose of only that drug has less functional flexibility than a programmable infusion pump that can deliver a variety of drugs at different rates and volumes. *Operational flexibility* is the amount of variation a system can tolerate while still allowing task completion. Variation is measured with respect to one or more variables and one or more tasks. For example, if the only task of interest is delivering a dose of epinephrine and available time to deliver the dose is the only variable used to measure variation, then the epinephrine auto-injector has greater operational flexibility than a programmable infusion pump because the auto-injector can deliver its dose under a wider range of available times. In contrast, if variation is measured by the range of patient-types (e.g., adult, pediatric, neonate, etc.) and conditions to be treated, then a programmable infusion pump has higher operational flexibility.

At this time, SYFSA addresses only procedural flexibility. Although this is a limitation of the current framework, we feel the focus on procedural flexibility is warranted for several reasons. First, procedural flexibility is an important component of system design that can affect both functional and operational flexibility. For instance, the high procedural flexibility of a programmable infusion pump allows it to

perform more functions (increased functional flexibility) under more conditions (increased operational flexibility) and do each function several different ways (procedural flexibility) than an epinephrine auto-injector. An analysis of procedural flexibility is, therefore, necessary for analyzing operational and functional flexibility.

Second, many best practices in healthcare are highly procedural. Attempts to improve practice or enforce best practices often take procedural forms. This is especially true of regulations, standard operating procedures, structured data entry, and Health IT forcing functions and interaction design. The motivation for this approach comes from decades of experience that shows the healthcare work domain is under-constrained and that even experienced workers often do not know or do not follow best practices. This has resulted in a well intentioned, but often ineffective reaction to erect barriers that force workers to do the "right" thing. As noted in our review, this can result in a system so inflexible that it prevents or hinders workers from delivering appropriate care, or leads workers to create workarounds that can jeopardize themselves or the institution, and even bring harm to patients. For example, estimating a required patient weight when there is no way to weigh the patient can lead to dosing errors.

In future work we plan to extend SYFSA to incorporate the other two types of flexibility.

A FRAMEWORK FOR SYSTEMATIC YET FLEXIBLE SYSTEMS ANALYSIS (SYFSA)

To illustrate our framework and how it can be used to design SYF systems, we consider a simplified procedure: central venous line insertion [102]. Central lines are used to establish reliable access to large (central) veins to deliver medications and fluids, draw blood for testing, and obtain measurements, such as central venous pressure. Once inserted, a central line remains in place for days or weeks. As a result, patients may develop central line infections that substantially increase morbidity and mortality. The chance of infection is reduced by following infection control guidelines during insertion and minimizing the number of days the central line stays in the body.

Our example is a simplified version of the insertion procedure and sacrifices realism for clarity. For example, hands are usually washed before putting on a sterile gown to avoid contaminating the gown. We consider only the following actions, listed in the approximate order, required to comply with best practices for infection control:

Sterilize site
Drape patient
Put hat on
Put mask on
Put gown on
Wash hands
Glove up (put gloves on)
Insert central line
Apply sterile dressing

Under ideal circumstances, a caregiver first prepares the patient by sterilizing the insertion site and then fully draping the patient. The caregiver inserting the central line must then put on a mask, hat, and gown. The gown prevents the donning of a mask and hat, so while the order of mask and hat does not matter, they must both come before donning a gown. Once the gown is on, the caregiver washes their hands and then puts on sterile gloves. Following this, they insert the central line and place a sterile dressing over the insertion site.

Following Newell and Simon [103], a problem space consists of a symbolic representation capable of capturing each problem state, a set of operators (information or physical processes that transform one state into another), an initial state, and one or more goal states. Just prior to setting up a new programmable infusion pump for a patient, the initial state is one in which the pump is turned off, whereas the goal state is one in which the pump is infusing the prescribed drug at the prescribed rate and volume. Infusion pump operators consist of the actions (such as the buttons on the front panel) available to install the drug administration set and program the pump.

In general, a problem space of a real world task may consist of hundreds, thousands or even millions of states and transitions between states (operator applications). Manual analysis is difficult or impossible. Thus, we implemented each space as a model in *Mathematica* [104] that

generates a finite state machine (FSM) containing every possible state and operator application. We then used the FSM to visualize the space and to calculate measures for each space, such as all possible paths between a pair of states, the number of states, different goal states, and so on. A *Mathematica* notebook containing the code for the examples presented here is available from the first author and may be used to develop new models. We do not describe the details of this approach here because it is one of many possible ways to automatically calculate the equations described below. The basic approach to generating and using FSMs for the analysis of user interaction is fully described by Thimbleby in a book [105] and several articles [106-108].

In the remainder of this section we walk through the specification and implications for each of the three spaces, beginning with the idealized space. Although we present the spaces sequentially, we expect the framework to be used in an iterative fashion. Part of the value of the framework is that it provides insight to better understand a task and how to design an SYF system to support that task.

The idealized space

The idealized space is best specified as a *work domain ontology* (WDO) for the task [27]. A WDO defines an explicit, abstract, implementation-independent description of a task by separating the task from the work context and technology used to accomplish the task. In other words, the WDO separates inherent constraints of the task from constraints due to system design. Rather than focusing on details of the current system, WDO highlights the fundamental nature of the work, thereby providing guidance for designing an appropriate system to support the work. WDO does not provide explicit methods for discovering and visualizing constraints, however, CWA (Section 2.4) provides a range of such methods and visualization tools.

A WDO is easy to express as a problem space. The WDO goal is specified as one or more goal state(s). Operations in the WDO are specified as problem space operators. Constraints are specified as sets of preconditions on the operators.

Assumptions

As with all models, a WDO is based on a variety of assumptions that set the scope of the model (i.e., which elements of the real world are considered relevant and which are not). When we specify the idealized space we must always specify our assumptions.

For the idealized central line insertion space, we assume a single caregiver will accomplish the entire task, that all required supplies are available, and that there is sufficient time to do the entire procedure according to best practices. We also assume the objects needed to follow the best practice and the caregiver are specified in the WDO (i.e., are inherent components of the abstract task).

Explicitly listing assumptions allows us to better assess the validity and scope of the idealized space and, subsequently, the results of the entire analysis. Berenholtz et al. [102] found lack of ready access to supplies a barrier to following the best practice for central line insertion. Part of their intervention for lowering central line infections was to develop a central line insertion cart, restocked on a regular basis. We assume all supplies are on hand to simplify our example, but in an actual design setting, making this assumption explicit would allow one or more of the stakeholders in the design process to question its validity, with the possibility of modifying the analysis.

State representation

To specify a problem space we must decide how to represent the system state. Abstractly, we think of state representation in terms of a set of state variables and a specific state as a specific assignment of values to each variable. In this example we use a simple Boolean representation of state components to record whether an action was done or not. For instance, if nothing has been done the components would all be false, thus:

```
centralLineInserted = False drape
Patient = False
glovesOn = False
gownOn = False
hatOn = False
maskOn = False
```

sterileDressing = False
sterilizedSite = False
washedHands = False

Here you can read "=" to mean "is." This representation captures the state of the system regardless of whether an element of the system state is visible or hidden. For instance, putting on gloves is a readily visible change to the system state. In contrast, washing hands is not.

There are many different ways to represent system state. We suggest including the minimum properties of the state needed to support the idealized problem space. One should model "relevant" features. As the model is analyzed, other significant features may be recognized and added to the model. We will discuss the importance of this advice below when we describe natural and system spaces.

Operators

We define operators using a set of logical preconditions on the state and how the operators change the state (Table 1).

Here we use the conventional symbols for logical NOT and ∧ for AND. The prime (') notation means the value after the operator has been applied to a state. For instance, the preconditions for the operator "Drape patient" state the operator can only be applied to states in which drapePatient is false and sterilizedSite true, and that after the operator is applied the state component drapePatient will be true. The prime notation allows this to be stated mathematically:

operator("drape patient") =¬ drapePatient ∧ sterilizedSite ∧ drapePatient'

As usual, any state component not mentioned is unchanged; if we wished we could have written operator("drape patient") =¬ drapePatient ∧ sterilizedSite ∧ drapePatient' ∧ maskOn = maskOn', which means the same thing, except redundantly says the state of the mask is unchanged.

Coincidentally in this example all operators only achieve setting the corresponding state component; thus "wash hands" implies washedHands', but in general many components might be affected. For example, if we tracked left and right hands separately, then the single "wash hands" would achieve *two* outcomes:

washedLeftHand' ∧ washedRightHand'

Note also that operators are formal problem space constructs that specify one or more task actions. In the central line insertion example each operator corresponds to a single task action, but in general, an operator can take parameters that define a set of task actions. For example, in an interface for selecting from among several patients we could define a Select(patient) operator, where patient is any patient shown on the screen. If 20 patients are shown on the screen, this single operator could be instantiated 20 times resulting in 20 different possible task actions.

Operator	Precondition	Postcondition
Sterilize site	¬sterilizedSite	sterilizedSite'
Drape patient	¬drapePatient ∧ sterilizedSite	drapePatient'
Put hat on	¬hatOn ∧ drapePatient	hatOn'
Put mask on	¬maskOn ∧ drapePatient	maskOn'
Put gown on	¬gownOn ∧ hatOn ∧ maskOn	gownOn'
Wash hands	¬washedHands ∧ gownOn	washedHands'
Glove up	¬glovesOn ∧ washedHands	glovesOn'
Apply sterile dressing	¬sterileDressing ∧ centralLineInserted	sterileDressing'
Insert central line	¬centralLineInserted ∧ glovesOn	centralLineInserted'

Table 1. Operator and conditions for the idealized central line insertion space.

Finally, we note that automated model checking can (and should) be used on specifications such as this. It is easy to check automatically that centralLineInserted always implies maskOn, even though this is never stated explicitly (and would be tedious and error-prone to try to say so for all relevant states).

Initial state

The initial state is one in which nothing has yet been done: all components are False.

Goal state

The goal state for this example is one in which all of the operators have been applied (equivalently, all of the actions have been done), and thus all the components are true:

centralLineInserted = True
drapePatient = True
glovesOn = True
gownOn = True
hatOn = True
maskOn = True
sterileDressing = True
sterilizedSite = True
washedHands = True

This is equivalent to the more concise logical statement:

centralLineInserted ∧ drapePatient ∧ glovesOn . . .

The goal state specifies only that all operators have been taken, not that they have been done in the correct order. There is no way to specify sequences of operators in terms of state properties alone. Instead, we constrain the sequence through the operator preconditions. Taken together, the initial state, goal state, operators, and operator preconditions, restrict the problem space to paths that reach the goal using an appropriate sequence of operators. However, we are not restricted to using this representation. Other representations may help us understand and explore the space from different perspectives. For example, we might choose to track whether the field is sterile or not and how actions affect whether or not a sterile field is created or maintained. We could then specify that some actions should only be done in a sterile field. Taking this further, we could choose to represent the urgency of the procedure and then modify the goal and operators to explicitly consider numerical time factors. Exploring alternative problem space formulations may inform system design.

Goal state

When a space is small, visualizing it can aid in understanding and pinpointing sources of flexibility and systematicity. From the idealized

space shown in Figure 1, we can see there is only one goal state and two different paths to it. The shortest path from the initial to the goal state is nine steps. There is clearly very little flexibility—one choice—in the idealized space.

The natural space

The natural space captures the task actions and constraints on those actions imposed by the physical world. For example, one natural constraint is that you cannot remove a surgical glove you have not put on. In contrast, you can wash your hands with surgical gloves on. In the natural space we also separate the primary goal from secondary goals. For instance, inserting the central line is the primary goal, while putting a sterile dressing on the insertion site is secondary.

Unlike the idealized space, the natural space need not be a WDO. Since the natural space is intended to reflect the real world, we can capture aspects that may affect task performance, such as non-task critical artifacts or cognitive limitations and assumptions. For instance, we might assume no clinicians will apply the sterile dressing prior to inserting a central line, even though there is nothing to physically prevent this.

When representing the state in the natural space, we must consider some state variables may be measurable and some may be hidden (or latent). Distinguishing between the two is a matter of perspective. In a typical automatic teller machine (ATM) the user has no visible indication of whether their ATM card is in the machine. However, this state variable is readily available to the ATM. When considering which variables are hidden vs. visible, we recommend taking the perspective of the human(s) part of the system. If the human cannot readily detect the value of a state variable, consider it hidden. In addition, assume that cognitive state variables are hidden. The former recognizes that the human in a system is likely to forget or distort values of state variables not readily observable in the environment. The latter recognizes that cognitive states are also likely to be forgotten or distorted. Both are likely to occur given the stress and interruptions present in many real world settings.

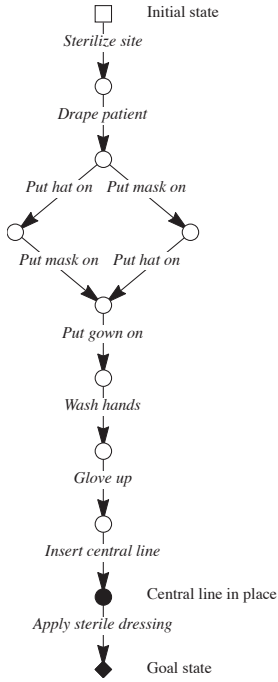


Figure 1. The idealized problem space. The initial state is square and the goal state is a diamond. The black circle is a state in which the central line is in place but the sterile dressing is not yet applied.

Assumptions

For the natural central line insertion space, our assumptions are similar to those of the idealized space. We assume a single care-giver will accomplish the entire task, that all necessary supplies are available, and that there is sufficient time to do the entire procedure. We also assume the artifacts needed to follow the best practice and the caregiver are part of the task model. In contrast to the idealized space, we define central line insertion as the primary goal. Creating and maintaining a sterile field are possible, but not required, because there are no natural constraints that enforce these requirements.

State representations

We use the same representation as the idealized space.

Operators

The operators for the natural space are identical to those of the idealized space, but the preconditions reflect hard constraints found in the task environment (Table 2). These are that the hat and mask cannot be put on after the gown is on and that the sterile dressing will not be put over the insertion site prior to inserting the central line. Preconditions also reflect our assumption that all other operators, except applying the sterile dressing, will not be done once the central line is in place.

Operator	Precondition	Postcondition
Sterilize site	¬sterilizedSite ∧ ¬centralLineInserted	sterilizedSite'
Drape patient	¬drapePatient ∧ ¬centralLineInserted	drapePatient'
Put hat on	¬hatOn ∧ gownOn ∧ ¬centralLineInserted	hatOn'
Put mask on	¬maskOn ∧ gownOn ∧ ¬centralLineInserted	maskOn'
Put gown on	¬gownOn ∧ ¬centralLineInserted	gownOn'
Wash hands	¬washedHands ∧ ¬centralLineInserted	washedHands'
Glove up	¬glovesOn ∧ ¬centralLineInserted	glovesOn'
Apply sterile dressing	¬sterileDressing ∧ ¬centralLineInserted	sterileDressing'
Insert central line	¬centralLineInserted	centralLineInserted'

Table 2. Operators and conditions for the natural central line insertion space.

Initial state

The initial state is the same as the idealized space.

Goal state

The goal states are any states in which the central line is in place. The goal is therefore a set of states.

Analysis of the natural space

The network diagram in Figure 2 shows the natural space is more complex and has considerably more flexibility than the idealized space. As with the idealized space, the initial state is shown as a square, goal states are black, and the goal state with all operators applied, although not necessarily in the right order, is shown as a black diamond. There are

many more goal states in the natural space because it recognizes a person may stop once they accomplish the primary (central line placement) goal.

The natural space has 384 states of which 256 are goal states. There are 13,004 paths that lead to a state in which the central line is inserted with the shortest being one step and the longest nine. Although there are 1,680 possible paths to the "ideal" goal state, only two of these paths contain the appropriate sequence of nine steps that reflect best practice.

Comparing the natural space to the idealized space, we can see the ideal sequence of actions is not enforced or encouraged by physical constraints. Some actions, such as washing hands or sterilizing the site, may leave no visible record, meaning the current system state is not visible. A lack of visibility of system state is a major usability problem that can lead to errors of omission (omitting a necessary step; e.g., not washing hands) and commission (including an unnecessary step; e.g., washing hands twice). Further, the system state contains insufficient information to allow an observer to detect the ideal goal state. The state variables in our problem space indicate only which actions were done, not the sequence of actions. However, the ideal goal depends, in part, on action order.

Because the sterile dressing is placed after the primary goal of central line insertion is achieved, there is a strong chance of post-completion errors [109], which are errors occurring when a person forgets to do an important task action that must be taken after they have accomplished the primary goal. Typical post-completion errors are forgetting to retrieve your ATM card after receiving cash from the machine or leaving an original document on a copier after making copies.

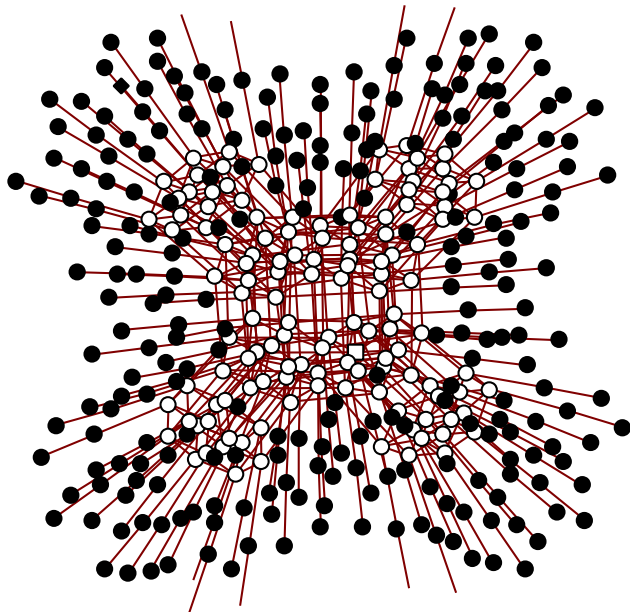


Figure 2. The natural central line insertion space. The initial state is the square in the lower right quadrant of the central image. The goal state in which all operators have been applied is the black diamond in the upper left corner. Black circles are states in which the central line has been placed. White circles are states where the central line has not been placed.

Taken together, characteristics of the natural space allow flexibility that makes idealized task performance less likely to be achieved (i.e., intuitively the task might be considered "error prone."). Below we use the comparison between these two spaces to consider a SYF system that encourages ideal performance, while supporting graceful degradation under unexpected or unusual conditions.

The system space

As noted above, stakeholders can use SYFSA to design a new system or to evaluate and possibly refine an existing system. For this demonstration of SYFSA, we base the system space on the existing intervention proposed and implemented by Berenholz et al., which has nearly eliminated central line-related bloodstream infections in multiple institutions [102, 110]. Although the intervention was widely reported to consist of a simple checklist, it actually has five components: 1) educating staff on best practices and the intervention; 2) creating a central line insertion cart to ensure easy access to all supplies needed to comply with the best practice; 3) asking daily whether the central line could be

removed; 4) a checklist to ensure adherence to best practices; and 5) empowering nurses to stop the procedure if guidelines were not followed during non-emergency situations. Here, we are concerned only with the elements of the intervention that directly affect central line placement.

These interventions lead to a system that addresses several of the characteristics, assumptions, and problems noted in our idealized and natural spaces. The supply cart supports our idealized space assumption that all supplies will be available at the start of the procedure. The checklist, external monitoring by a nurse, and nurse's power to stop the procedure encourages and enforces the ideal practice. The checklist itself increases visibility of system state and externalizes knowledge of the ideal action sequence. Taken together, these factors provide and encourage systematicity. At the same time, the system provides flexibility by allowing the provider to deviate from the best practice in situations where the central line must be inserted emergently.

The resulting system space is a combination of the graphs from the natural (Figure 2) and idealized spaces (Figure 1) with a new root state that switches between the two original root states depending on whether there is an emergency. Switching to the natural space relaxes the action constraints imposed by the idealized space and allows the provider to accept a goal that trades off the chance of an infection with the need to quickly insert the line.

SYFSA provides a means of qualitatively analyzing tradeoffs in systematicity and flexibility during organizational or information system design. The explicit descriptions of each of the three spaces (in terms of initial state, goal state(s), operators and their preconditions) force stakeholders to explicitly describe their assumptions and understanding of each of the spaces. By making these descriptions explicit, stakeholders can share, debate, and refine each space. This allows stakeholders to determine whether each space adequately models best practice (idealized space), the current system (natural space) and the new or redesigned system (the system space). Comparing descriptions of these spaces can reveal tradeoffs or potential opportunities to iteratively refine each space to better address stakeholder needs.

In the next section we consider information theoretic measures for qualitatively comparing the flexibility of different system designs and how closely they match the flexibility required to complete a task.

INFORMATION-THEORETIC MEASURES OF PROCEDURAL FLEXIBILITY

As noted earlier, there are different measures of flexibility. Here, we propose flexibility measures that capture our intuitive notion of procedural flexibility and allow us to compare the flexibility of different SYF system designs with respect to one or more tasks. We distinguish between *inherent task flexibility* and *system flexibility*. The former is the amount of flexibility required to do a task, whereas the latter is the amount of flexibility in a system designed to support the task. For instance, if the task is to deliver a single dose of epinephrine, the *inherent task flexibility* is low and best met by designing a device, such as an epininephrine auto-injector, that has similarly low *system flexibility*. System flexibility often differs from task flexibility because a particular system may admit actions incorrect or irrelevant to completing a task, or may not allow actions actually needed to complete the task. Thus, a system may support more or less flexibility than is inherent in the task. When a system design allows more flexibility than is inherent in the task, it allows actions that may lead to errors or inefficiencies. In contrast, when a system design supports less flexibility, it may be impossible to complete the task.

To derive appropriate measures of flexibility, we start by considering the extreme end points of task flexibility: no flexibility and complete flexibility. We propose that if there is only a single correct way to complete a task, then that task has 0% flexibility; whereas if any possible sequence of task actions completes a task, then that task has 100% flexibility. Between these limits, flexibility should increase monotonically (that is, if there are more ways of accomplishing the task, flexibility should not decrease).

To explore this concept, consider the following three simple tasks:

> *Any-object:* Table A has ten objects and Table B is empty. The goal is to place any one object from Table A onto Table B.

All-objects: Table A has ten objects and Table B is empty. The goal is to place all ten on Table B.

Sort-objects: Table A has ten numbered objects. The goal is to move all ten objects in increasing order to Table B (i.e., object 1, 2, 3 ..., object 10).

In our central line example, Table A might be the central line supply cart and Table B the sterile field.

By our intuitive definition of flexibility, Sort-objects is the least flexible of the three tasks. But, which of the other two is the most flexible? If we define flexibility as the number of paths to the goal, then All-objects with 10! = 3,628,800 paths is clearly more flexible than Any-object with only 10 paths. But intuitively, it seems Any-object is equally, if not more flexible than All-objects because Any-object allows any choice of action, and just one choice is needed. In contrast, although All-objects allows any sequence of actions to lead to the goal, each choice constrains the actions that follow, which intuitively would seem to decrease flexibility. In fact, a system space that allowed a person to move an object from Table B back to Table A would be overly flexible for the All-objects task. Thus, the number of paths in a space can have more to do with the size of the space, rather than constraints on actions.

Instead of using the number of paths to the goal to define flexibility, we can use the average amount of information needed to choose an action per non-terminal state (whether those states lead to a goal or non-goal terminal state). In information theory [111], the amount of information (measured in bits) in a choice between n equally likely actions is $\log_2(n)$, so the total information required to perform a sequence of actions is the sum of the information for each decision along the path. Suppose that there are n non-terminal states S_i, and these states have a corresponding number of equally probable actions a_i (in terminal states there are no actions). Then the average bits per non-terminal state F is given by:

$$F = \frac{\sum_{i=1}^{n} \log_2(a_i)}{n} \quad (1)$$

We can convert F to an indicative flexibility score. Of the many possibilities, here we define a percentage so it is conveniently measured as a number increasing to 100:

$$F = \frac{100F}{F+1}$$

(2)

Eq. (2) approaches 100% as F increases. In addition, because of the definition of F in Eq. (1) together with Eq. (2), a space where every state has a single action has 0% flexibility, whereas a binary tree (in which all non-terminal states have two actions) has 50%.

Table 3 shows the flexibility of the three simple tasks described above. Consistent with our intuitive notion of flexibility, Sort-objects has zero flexibility, Any-object has the most flexibility, whereas All-objects has less flexibility because each action further constrains the remaining available actions. Although any possible action leads to completion of Any-object, it falls short of our intuitive 100% flexibility measure because the information theoretic measure considers the number of choices at each step. As a result, the flexibility of Any-object will approach 100% as the number of objects increases.

Space	*F*	% Flexibility
Any-object	3.32	76.86
All-objects	0.51	33.64
Sort-objects	0	0

Table 3. Flexibility of three simple tasks using bits per state (Eqs. (1) and (2)).

Space	*F*	% Flexibility
Idealized	0.1	9.1
Natural	0.94	48.5
System	0.91	47.6

Table 4. Flexibility of three central line insertion spaces using bits per state (Eqs. (1) and (2)).

Table 4 shows the flexibility of the three types of spaces for central line insertion. As expected, the idealized space has the least flexibility,

whereas the natural and system spaces have considerably more, with the system space being nearly as flexible as the natural space. The small difference in flexibility between the natural and system spaces is misleading, because the more flexible path through the system space can only be taken in emergency situations—situations that are less likely to occur than non-emergent situations. The general problem is that Eq. (1) assumes all states have an equal chance of being visited, which is false because of structural properties of the space, (e.g., the top state is always visited) and because actions from any single state may be chosen with differing probabilities. The problem is easily corrected by computing the average amount of information based on the probability of each action in each state. If a non-terminal state S_i has a_i actions and those actions have probabilities $p_1^i ; \ldots ; p_{a_i}^i$, then a choice of action at S_i conveys an average number of bits given by:

$$B_i = \sum_{j=1}^{a_i} p_j^i \log_2 \left(\frac{1}{p_j^i} \right)$$

(3)

This results in a version of Eq. (1) that considers the probability of actions:

$$F = \frac{\sum_{i=1}^{n} B_i}{n}$$

(4)

However, this equation alone does not consider how action probabilities affect the likelihood of reaching future states. In the central line insertion space, Eq. (1) assumes that emergency and non-emergency situations are equally likely, resulting in 1 bit for the initial state. If we instead assume that an emergency occurs, say, 10% of the time, Eq. (3) reduces the required bits for the initial state from 1 to 0.469. Given the number of states in the space, however, and assuming that actions for all subsequent states are equally likely, this decrease for the initial state has very little effect on overall space flexibility (47.63–47.58%).

In general, it is important to consider the probabilities of actions in a SYFSA analysis, because SYF systems support graceful degradation by

making common actions and action sequences easy and uncommon ones possible. For example, in a user interface, common actions may be made more salient and/or faster to select than less common actions. This provides for graceful degradation in the face of unanticipated events.

To account for the probabilistic effects of actions on future states, we need to weight the average bits per state, B_i, by the probability of reaching each state. If there are n non-terminal states and these states have probabilities $s_i, \ldots, s_n$, then the weighted average bits per non-terminal state is given by:

$$F = \frac{\sum_{i=1}^{n} s_i B_i}{\sum_{i=1}^{n} s_i}$$

(5)

Space	Non-probabilistic (Eq. (1))		Weighted probabilistic (Eq. (5))	
	F	%*F*	*F*	%*F*
Idealized	0.1	9.1	0.11	10
Natural	0.94	48.5	1.86	65.0
System	0.91	47.6	0.78	43.8

Table 5. Comparison of the flexibility of three central line insertion spaces using non probabilistic (Eq. (1)) vs. probabilistic (Eq. (5)) flexibility measures.

Space	Average bits per patch (Eq. (6))
Idealized	1.00
Natural	9.62
System	6.31

Table 6. Average bits per path for the three central line insertion spaces using Eq. (6).

Because the probabilities of the non-terminal states need not sum to one, weights are normalized by dividing by their sum. Table 5 compares the percent flexibility of the three central line insertion spaces using the non-weighted, non-probabilistic F from Eq. (1) to that of Eq. (5). The weighted measure for the idealized space shows very little difference. However, there are larger differences in the measures for the natural and

system spaces. The natural space nearly doubles the required number of bits per state, reflecting that earlier states have both higher probabilities of being reached and a larger number of possible actions. The system space mean bits per state decreases from 0.91 to 0.78, reflecting the lack of flexibility in the idealized path. More importantly, under Eq. (5), the system space is now less flexible than the natural space (43.8% vs. 65%), as compared to their difference under Eq. (1) (47.6% vs. 48.5%).

Another useful information-theoretic measure for comparing spaces is the average information per path. This measure tells us, on average, how much information a person must convey in a particular space.

The total information conveyed by a single path is equivalent to the information content as measured by the probability of following the path (i.e., choosing a sequence of actions that result in taking the path to the goal). For instance, the probability of a path that has 6 states and 5 edges, where each edge has a probability of 0.5, is 0.5^5. The sum of the information conveyed by each of the 5 decisions is $5\log_2(1/0.5) = 5$, which is equal to the log of the probability of the path: $\log_2(0.5^5)$. Thus, the average information over all paths $P_1, \ldots, P_n$ with probabilities $p_1, \ldots, p_n$ is given by:

$$P^{avg} = \sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right)$$

(6)

This measure is sensitive to the size and complexity of a space, in that spaces that are deeper and have more choices per decision will naturally have greater average information per path. As noted in the previous section, it is often useful to compare the average information of specific paths, such as the correct paths in both the idealized space and the natural space. Table 6 shows the average bits per path for the three central line insertion spaces. The difference between the natural and system spaces results from the fact that the first state of the system space is an equally likely choice between an emergency situation, which leads to the natural space (requiring 9.62 bits), and a non-emergency, which leads to the idealized space (requiring only 1 bit).

We can use a similar measure to quantify how efficiently the natural space supports the best practice by comparing the amount of information a clinician requires to do the best practice in the natural space versus the idealized space. In the idealized space there are two equivalent paths of nine non-terminal states. Eight of the nine states permit a single action, whereas one state has two possible actions. This means that a person need only convey one bit of information to correctly perform the task in the idealized state. In contrast, the natural space has the same two paths, but because of the lack of natural constraints on possible actions, seven of the non-terminal states allow more than one action. The initial state has eight possible actions, the second state seven, and so on, with each correct action eliminating one possible action until the final two non-terminal states admit a single action each (with zero bits of information). Assuming actions are equally probable, this makes the total bits in either correct path:

$$\sum_{i=2}^{8} \log_2(i) = 15.2992$$

(7)

Since the idealized space requires only 1 bit, the efficiency of the natural space for supporting the best practice is only 100 * 1/15.2992 = 6.5%.

According to the Hick-Hyman law, the time to make a decision is proportional to the amount of information in the available choices [112, 113]. As a result, the information theoretic analysis of a system provides a prediction of cognitive load and relative task times (e.g., a task that requires more information is likely to take longer than a task that requires less information). In addition, through practice a person can automate a consistent sequence of task actions, resulting in fast, nearly subconscious behavior. This means a person must acquire through practice over 15 bits of information to fully automate the idealized task in the natural space, but only 1 bit in the idealized space. We can use this kind of analysis to compare the learnability of different spaces for different system designs.

DISCUSSION AND FUTURE WORK

While the current approach is clear and rigorous, there are a number of limitations to SYFSA that should be noted and that could provide inspiration for further work. SYFSA as described here is designed to analyze systems that support a single task. However, many systems (such as an infusion pump) must support more than one task. In SYFSA such systems are modeled by expanding the spaces so that they admit all tasks, and then separately analyzing each task. For example, a programmable infusion pump supports many different volumes and rates of delivery, so the idealized space must include operators that can be applied to achieve each possible (and allowable) combination of volume and rate. Each task, such as the task of starting at a state where the rate is 0 and then moving to a state where the rate is 125, can then be analyzed using the equations described above. To analyze the entire system, a designer must analyze each task separately. It is up to the designer to decide how to aggregate the results of each task analysis. For instance, the designer could produce a single flexibility measure using a weighted average of each task's flexibility, where the weights are the expected frequency of each task.

SYFSA does not provide designers guidance on how to determine which tasks should be included in an analysis, so it is important for the designer to use other work-centered or user-centered methodologies to determine which tasks a system should support. In addition, any system designed to support multiple tasks necessarily requires additional procedural flexibility because the user has more possible actions to take at each step. This flexibility can lead to errors and inefficiencies for any one of those tasks. For example, a programmable infusion pump must provide actions that allow a user to enter different volumes and rates of delivery, but since the device does not know what the user wants to enter it cannot completely constrain the user's behavior for the specific task at hand. Designers of infusion pumps have dealt with this problem by including dose error reduction systems, wherein the user must first specify a drug and concentration prior to programming the pump. Once the pump knows the drug, the pump can enforce additional, drug-specific constraints on rate and volume. Developing a work domain ontology to inform the idealized space (as we suggest in Section 3.1) can help designers better explore intrinsic task constraints. In any design for

supporting multiple tasks, common user-centered design principles recommend providing error reversal, or undo, functionality to traverse back through prior choices, to change them or to review them. For example, if a clinician accidentally sets an infusion pump to 100 mcg/min instead of 10 mcg/min, it should be possible to clear or re-enter the infusion rate. This reflects an increase in flexibility over the idealized space (which assumes a perfect user), but is an appropriate trade-off given the realities of the natural space in which even highly trained users can make mistakes.

Design frameworks such as SYFSA are often difficult to validate. They tend to be used or abandoned based on whether designers find them useful and easy to use. Any evaluations are often qualitative in nature, consisting of case studies and arguments that outline strengths and weaknesses. However, some aspects of SYFSA may be empirically testable. SYFSA assumes that systems that are too flexible relative to the task (the idealized space) will be harder to learn and use, as will systems that support too little flexibility. Building on several existing laws and cognitive results, we also believe that SYFSA can predict relative efficiency, cognitive load, and learnability. However, we have not yet empirically evaluated these claims.

Another challenge is that many real world tasks and systems can have dozens or hundreds of possible actions leading to thousands or even hundreds of thousands of states in each problem space. There are at least three solutions to this problem. The first is to generate and analyze the spaces computationally as we have done for the examples here. Thimbleby describes these techniques in detail and they are also demonstrated in the *Mathematica* code available from the first author [105]. The second is to reduce the complexity of the spaces by selecting an appropriate level of abstraction. For example, in the central line examples we did not model the detailed cognitive steps required to determine the best location to insert the central line, nor all of the physical steps involved in the process, such as opening equipment packages. As with any modeling approach, selecting the right level of abstraction is challenging and remains part art and part science. The third solution is to separately analyze subparts of a complex system. For instance, we analyzed an infusion pump by analyzing the number entry

tasks (for specifying rate and volume) separately from the other tasks involved with the pump (e.g., entering various data entry modes, pausing the infusion, responding to an alarm, etc.). In practice, it is often necessary to use a combination of these approaches to tame the complexity of real world tasks.

Finally, the measures described in this paper characterize procedural flexibility only, not functional or operational flexibility. These other forms of flexibility are also important for health information and organizational systems, and will require extensions to SYFSA.

CONCLUSIONS

SYFSA is a systematic approach to analyzing and designing SYF systems. By explicitly representing three spaces, the idealized space, the natural space, and the system space, designers and domain experts can examine assumptions behind task analysis and system design, and possible tradeoffs between systematicity and flexibility. By making assumptions and constraints on actions explicit, the framework provides a means for designing novel systems that better support constraints inherent in a task, but not in the natural environment. In addition, the quantitative information theoretic flexibility measures allow analysts to compare different spaces and system designs in terms of relative efficiency for supporting a task, cognitive workload, and learnability.

ACKNOWLEDGMENTS

This work was supported in part by Grant 10510592 for PatientCentered Cognitive Support under the Strategic Health IT Advanced Research Projects Program (SHARP) from the Office of the National Coordinator for Health Information Technology; NCRR Grants 3UL1RR024148 (NCATS UL1 TR000371), UL1RR033173 ^ 1RC1RR028254; HRSA Grant D1BRH20410; EPSRC Grants EP/G059063, EP/K004549, and EP/F020031. E. Markowitz was supported by a training fellowship from the AHRQ Training Program of the WM Keck Center for Interdisciplinary Bioscience Training of the Gulf Coast Consortia (AHRQ Grant T32 HS017586). The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors.

A version of this chapter originally appeared in the Journal of Biomedical Informatics, volume 46, Johnson, T. R., Markowitz, E. Bernstam, E. V., Herskovic J. R. & Thimbleby, H., SYFSA: A framework for Systematic Yet Flexible Systems Analysis, 665-675.

EHR Usability Assessment

Healthcare providers often complain of EHR systems that are clunky, difficult to use, and—ironically—hindering instead of facilitating patient care. SHARPC developed the Rapid Usability Assessment (RUA) method to identify EHR usability problems through expert reviews and task modeling (Chapter 4). RUA was applied to five commercial EHR systems for common meaningful use tasks. Analyses demonstrated a number of usability problems and lengthy task completion times.

SHARPC researchers then interviewed 11 EHR companies about their product development processes (Chapter 5) and user-centered design (UCD) capabilities. Vendors ranged from small ($300,000 yearly revenue) to large (over $1 billion a year). Understanding and use of UCD varied widely. Some vendors had well-developed UCD processes, infrastructure and usability personnel. Some had only basic UCD capabilities, and others had misconceptions about UCD. Nearly all vendors identified short development timelines as a barrier to embracing UCD.

Usability assessment is not a straightforward process. There are many ways to measure overall usability, including pen and paper analyses, and spreadsheets. SHARPC developed the *Turf* usability tool suite (Chapter 6) to semi-automate the process of usability assessment, centralize data storage, and produce simple yet sophisticated statistical analyses. *Turf* can be an important tool in designing user-centered EHRs.

The US Office of the National Coordinator for Health Information Technology now requires vendors conduct summative user testing and provide evidence of UCD for their product to be certified for meaningful use. SHARPC created resources to help vendors meet 2014 EHR certification requirements, including a summary of Safety Enhanced Design requirements (Chapter 7), free downloadable use cases for summative user testing (Chapter 8), and education and training materials on usability, human factors and UCD (Chapter 9).

An EHR usability experience survey was developed and deployed at 11 acute care facilities (Chapter 10). Results indicated important opportunities for improving EHR usability.

4: Rapid Usability Assessment of Commercial EHRs

Muhammad F. Walji, PhD
The University of Texas Health Science Center at Houston School of Dentistry

Amy Franklin, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Thomas Kannampallil, MD
University of Illinois at Chicago Department of Family Medicine

Zhen Zhang, MD, MPH, MSc
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Krisanne Graves PhD, RN, CPHQ
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Yuanyuan Li, MD, MS
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Craig Harrington, LMSW
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Debora Simmons, PhD, RN
CHI St. Luke's Health

Jiajie Zhang, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

ABSTRACT

A laboratory-based, analytical usability process called Rapid Usability Assessment (RUA) was used to inspect and evaluate five commercial electronic health records (EHR) systems to identify usability challenges and estimate the efficiency in performing routine meaningful use related tasks. RUA consisted of three stages: 1) selection of meaningful use objectives, 2) use of a modeling tool to predict task completion times as an indicator of productivity, and 3) identifying usability challenges through expert review. Time taken to complete tasks varied across the twelve meaningful use case scenarios. Clinical summary (M=338 seconds), computerized provider order entry (CPOE) (M=326

seconds), and Medication List (161 seconds) test procedures had the highest mean task completion times. Expert review detected 1,135 usability problems within five EHRs. CPOE (N=58) and Clinical Summary (N=53) had the highest mean number of usability problems per EHR. Memory, Feedback + Error, and Match were the most frequently violated usability heuristics. Time for experts to perform meaningful use-related tasks in existing EHRs were high. These times are predictive of errors in routine performance and would likely be higher in actual clinical practice. Users face numerous usability problems as they use systems in real-world clinical practice. Poor usability is a critical challenge limiting the adoption and safe use of EHRs. Performance times can be used as benchmarks to measure and compare EHR systems along the efficiency dimension of usability. Our results suggest an urgent need to improve the usability of existing EHRs.

INTRODUCTION

The American Recovery and Reinvestment Act (ARRA) of 2009 provided significant financial incentives for healthcare providers to adopt and meaningfully use electronic health records (EHR) systems [114]. However, the promise of health information technology (HIT) to transform healthcare practice is often limited by its usability [23]. Within the framework of the ARRA, the Office of the National Coordinator funded the National Center for Cognitive Informatics and Decision Making in Healthcare as one of the four Strategic Health IT Advanced Research Programs. The establishment of this Center was based on the recognition that EHR usability was a significant barrier to achieving the goal of nationwide use by 2014 [115].

Usability refers to how useful, usable and satisfying a system is for the intended users to accomplish goals in a work domain by performing certain sequences of tasks [7]. In spite of recent reports of devastating consequences from poor EHR usability [16, 17], significantly more attention is directed towards the financial and technical aspects of EHR than its usability and integration into the clinical work environment [23, 116]. We conducted extensive evaluation and usability analyses of EHR systems, developing meaningful metrics for assessing EHR usability.

We studied five commercial EHR systems and identified core usability challenges and optimal efficiency based on a Rapid Usability Assessment (RUA) protocol. Results, we believe, provide diagnostic information for developers to improve the usability of EHR systems.

APPROACH

Methodology was based on our prior health IT usability research [18, 20, 21, 42, 46, 66, 117-120]. RUA combines model-based and inspection-based approaches to EHR evaluation. Laboratory methods are complementary to time-intensive field observation and user testing, making it a flexible way to develop feedback for iterative design. The approach generated a repository of EHR usability challenges and issues useful for future development efforts.

RUA uses the TURF framework [7]. TURF stands for Task, User, Representation, and Function, core analyses that can assess usefulness, usableness and satisfaction of an EHR. TURF provides a mechanism to evaluate an overall system based on intrinsic complexity and extrinsic difficulty. Intrinsic complexity refers to the complexity of work needed to be done in a domain, independent of technology implementations or procedures. Functional analysis can be used to assess intrinsic complexity, thereby determining the usefulness of a system [46]. Extrinsic difficulty reflects difficulties a user experiences with a specific representation or interface to perform a task, which is an indication of system usableness. Representation and task analyses can be used to determine extrinsic difficulty.

RUA focuses on the extrinsic difficulty component of the TURF framework, assessed by task and representation analyses. The current version of the RUA protocol uses multiple methods including model- and expert-based evaluations. RUA results can be followed up by more resource-intensive and targeted user-based usability techniques.

We used RUA to evaluate EHR usability in three stages: 1) selection of meaningful use objectives, 2) predicting performance by task completion time as measured by *CogTool* [71], and 3) identifying usability challenges through expert review.

Selecting meaningful use objectives

Health providers must use a certified EHR containing meaningful use-related functionality to receive incentive payments. The National Institute of Standards and Technology (NIST) developed meaningful use test procedures, which are publicly available for EHR certification bodies [121]. Tests contain specific instructions and sample data to determine if an EHR system has met a meaningful use objective. For example, the "Maintain up-to-date problem list" test states an EHR system should "enable the user to electronically record, modify, and retrieve a patient's problem list over multiple encounters." The test specifies data must be recorded in a structured format using either ICD-9-CM or SNOMED-CT. Specific problems to be recorded, modified and retrieved are also specified.

Twelve clinical tasks associated with the first 15 core objectives of meaningful use in ambulatory care settings were selected for evaluation. Five EHR products were assessed to determine if test procedures could be completed.

Predicting performance using task completion times

After selecting a standardized set of tests for EHR evaluation, we studied the impact of the system on performance. We used completion times as our primary outcome measure. Completion time is one of the most widely reported barriers for EHR adoption due to its direct effect on productivity [66, 122, 123]. Although numerous researchers have reported both positive and negative performance outcomes due to EHR use [124], there is surprisingly limited conclusive evidence for EHR's impact on clinical workflow. Part of the reason is a lack of reliable evaluation data.

Our approach was to predict an expert's routine task completion times using a modeling tool, then use these results as performance benchmarks for laboratory evaluations. The Goals, Operators, Methods and Selection (GOMS) technique [63, 125] is a cognitively grounded approach that has been used for many years to predict task completion times. We used a generalized task analysis method derived from GOMS known as KLM (Keystroke Level Model). KLM is the simplest GOMS technique [63], using a pre-specified set of keystroke and mouse

operators with associated timings [126]. KLM predicts the time it takes for an expert (skilled in the domain in which the task is considered) to execute keyboard and mouse inputs along with the associated cognitive overheads (e.g., thinking time or time taken to visually acquire objects on the screen). For example, to enter a patient name and click a submit button: 1) move cursor to the name text box using the mouse, 2) type the patient's name using the keyboard, then 3) move the mouse to the "Submit" button and left-click. The model also incorporates "think" operators, i.e., time spent to mentally prepare for an action or a set of related actions (e.g., "think" before moving mouse to the name text box). Parameterized KLM values have been prescribed. For example, double-clicking a mouse button takes 0.4 seconds. KLM has been used in a variety of fields to compute task-based performance measures, including visualization [127], cell-phone menu browsing [128], handheld devices [71], evaluating devices for the disabled [129], email organization [130], and in-vehicle information systems [131]. We used KLM to determine routine clinical task completion times for one EHR system [66].

We used *CogTool* [132], a software application based on an enhanced version of KLM incorporating the more detailed ACT-R framework [133], for analysis of selected NIST meaningful use test procedures. This provided a theoretically grounded, analytical approach for predicting comparative performance across potentially different screen paths. By using a standardized sequence of tasks (e.g., recall information, modify fields, and record new data), we computed the time taken for the same task across multiple EHRs. For example, we could consider the time to complete an e-prescription for a given set of data independent of system interface design (e.g., popups, drop-downs, searching, using text to complete drug name selection, etc.).

Identifying usability challenges through expert review

An EHR may use design elements unfamiliar to users, display information in unexpected ways, or fail to provide appropriate feedback to ascertain if intended actions have been completed. An efficient way of identifying usability problems is through a process called "expert review," in which usability specialists determine if a user interface conforms to established usability principles. Expert reviews are cost effective, can be conducted rapidly, and have been found effective in identifying gross

usability problems [134]. These methods are complementary and, used together, can provide evidence of task performance time and impediments limiting user performance.

Heuristic evaluation, a type of expert review, was initially proposed by Nielsen [135] and modified for use in clinical settings [57]. In a heuristic evaluation, a small set of evaluators examine an interface and review its compliance with recognized usability principles (or "heuristics"). Heuristic evaluation has been successfully applied to health IT, including practice management [136], computerized provider order entry (CPOE) [137], telemedicine [60], and medical devices [57, 58]. It has also been successfully used for predicting usability issues that impact end user involvement [59, 138, 139].

We developed a modified heuristic evaluation process. Two evaluators independently inspected an EHR in the context of a specific clinical task and focused only on interface items needed to accomplish the task. Each evaluator was guided by seven heuristic design principles (Table 1) based on human cognitive capacities and limitations [19] and derived from previous research [7, 57]. For example, the heuristic principle of consistency encourages interface designers to set and adhere to patterns in layouts, vocabularies, color or other features that allow users to immediately feel familiar with their interface and to avoid distracting or confusing users with unnecessary variations. A violation of the consistency heuristic would be alternating placement of the cancel and save buttons across screens.

Expert reviews have the potential to identify a range of usability problems. Some issues may have life-threatening consequences, while others may impact only the cosmetic appearance of the system. Each heuristic violation was ranked using a four-point scale, where 1 was a cosmetic issue, 2 a minor usability violation, 3 a major usability violation, and 4 a catastrophic usability violation. Ratings consider the proportion of users who may experience the usability problem, the impact of the problem on patient safety or performance, and whether the issue is a problem only during initial encounters or could persistently disturb users.

Heuristic	Description
Consistency	Does the user have to cope with different ways of presenting information and navigating through the product?
Visibility	Can the user readily determine what the state of the system is? That is, does the system provide feedback about where the user is located in the product hierarchy and what the system is doing when actions are occurring?
Match	How well does the system model the real-world processes it is designed to support, which enables users to be able to leverage their existing understanding of the sequencing of activities?
Memory	All users have limitations to the amount of information they can keep in their active memories. The more information that must be recalled to use the product, the easier it is for users to forget what to do next, to lose track of information, or to make mistakes.
Feedback and Error	Does the product provide the user with feedback about actions it is performing? If errors can occur, what is done to prevent them, to minimize their impact, and to inform the user in a meaningful way what has happened?
Undo	Does the product enable the user to undo or reverse changes or actions that have occurred? Undo not only provides a safeguard; it encourages users to explore alternatives.
Document	How well does the product educate the user on how to use the product or how to solve resolve uncertainty about operating the product? How accessible is help and how relevant is it?

Table 1. Seven heuristics used as part of the Rapid Usability Assessment.

RESULTS

Table 2 provides a summary of our findings. Only six of the 12 use cases (Problem list, E-prescribing, Demographics, Vital Signs, Smoking Status and Body Mass Index (BMI)) could be performed in all five systems. Only two systems had full functionality for all selected test procedures at time of testing.

	Task Performance Time in Seconds						Number of Usability Problems (Average Severity Score)					
	A	B	C	D	E	**Mean**	A	B	C	D	E	**Mean**
Clinical Summary	~	**357**	**383**	**274**	~	**338**	~	**94 (2.1)**	**56 (2.7)**	**9 (2.3)**	~	**53 (2.3)**
CPOE	~	**352**	**330**	**295**	**328**	**326**	~	**78 (2.1)**	**48 (2.7)**	**66 (2.6)**	**41 (2.4)**	**58 (2.4)**
Record		250	220	189	196	214						
Modify		83	88	85	108	91						
Retrieve		19	22	21	24	21						
Medication List	~	~	**207**	**128**	**149**	**161**	~	~	**12 (3.1)**	**33 (2.4)**	**18 (2.7)**	**21 (2.6)**
Record			155	82	98	112						

	Task Performance Time in Seconds						Number of Usability Problems (Average Severity Score)					
	A	B	C	D	E	**Mean**	A	B	C	D	E	**Mean**
Modify			33	26	37	32						
Retrieve Active			9	9	6	8						
Retrieve All			10	11	8	10						
Problem List	**155**	**116**	**124**	**150**	**146**	**138**	**31 (3.0)**	**20 (2.2)**	**11 (2.6)**	**24 (2.4)**	**17 (2.6)**	**21 (2.6)**
Record	114	84	70	95	99	92						
Modify	21	13	27	32	38	26						
Retrieve Active	8	7	12	8	4	8						
Retrieve All	12	12	15	15	5	12						
Medication Allergies	**~**	**~**	**142**	**109**	**113**	**121**	**~**	**~**	**13 (2.7)**	**21 (2.6)**	**17 (2.3)**	**17 (2.5)**
Record			88	68	74	77						
Modify			33	23	22	26						
Retrieve Active			10	8	8	9						
Retrieve All			11	10	9	10						
E-prescribing	**65**	**78**	**71**	**67**	**45**	**65**	**43 (2.6)**	**22 (2.0)**	**18 (2.9)**	**28 (2.4)**	**19 (2.6)**	**26 (2.5)**
Demo-graphics	**38**	**66**	**57**	**76**	**75**	**62**	**12 (2.8)**	**21 (2.4)**	**11 (2.4)**	**14 (2.7)**	**28 (2.3)**	**17 (2.5)**
Record	18	27	26	37	32	28						
Modify	16	31	25	35	37	29						
Retrieve	4	8	6	4	6	6						
Vital Signs	**48**	**48**	**57**	**56**	**63**	**54**	**15 (2.6)**	**12 (2.3)**	**13 (2.7)**	**13 (2.6)**	**28 (2.3)**	**16 (2.4)**
Record	23	23	27	26	31	26						
Modify	18	19	22	24	24	21						
Retrieve	7	6	8	6	8	7						
E-copy	**~**	**~**	**54**	**23**	**-**	**39**	**~**	**~**	**13 (2.5)**	**55 (2.4)**	**~**	**34 (2.4)**
Smoking Status	**33**	**31**	**46**	**25**	**47**	**36**	**16 (2.7)**	**8 (1.9)**	**3 (2.5)**	**10 (3.1)**	**15 (2.2)**	**10 (2.5)**
Record	12	12	16	10	16	13						

	Task Performance Time in Seconds						Number of Usability Problems (Average Severity Score)					
	A	B	C	D	E	**Mean**	A	B	C	D	E	**Mean**
Modify	13	12	15	10	20	14						
Retrieve	8	7	15	5	11	9						
Growth Chart	**~**	**26**	**30**	**34**	**25**	**29**	**~**	**19 (2.3)**	**11 (2.8)**	**16 (2.0)**	**20 (2.5)**	**17 (2.3)**
Body Mass Index	**15**	**12**	**19**	**17**	**17**	**16**	**9 (2.8)**	**9 (1.9)**	**5 (2.7)**	**8 (2.2)**	**12 (2.3)**	**9 (2.3)**

Table 2: Summary of results from the Rapid Usability Assessment of 5 EHRs (Products A-E). ~ Use case not evaluated due to missing functionality. Note: Numbers have been rounded up to the nearest whole number.

Time on task – completion time

To understand the impact on clinical workflow, we computed the total task time per meaningful use case using *CogTool*. Looking across the subset of use cases we found that, on average, the BMI use case took the least time (*M*=16 seconds), while Clinical Summary" took the longest (*M*=338 seconds). If we consider task completion time as an indirect measure of task complexity, given the number of steps involved in the task, these results are not surprising. What may be surprising was the consistency across systems for the total completion time.

Expert review

Expert review useful in identifying challenges users face interacting with a system. In total, 1,135 usability problems were identified across the five EHRs. On average, Clinical Summary had the highest number of usability problems and BMI the fewest. Mean severity ratings were highest (2.6) for Problem List and Medication List. Growth Chart had the lowest mean severity rating (2.3).

Each usability problem was categorized into one or more heuristic violations. As shown in Figure 1, Memory, Feedback + Error, and Match heuristics were the most frequently violated. Figure 1 also provides examples of specific heuristic violations found in our analysis, along with an example of an alternate design showing adherence to the heuristic principle.

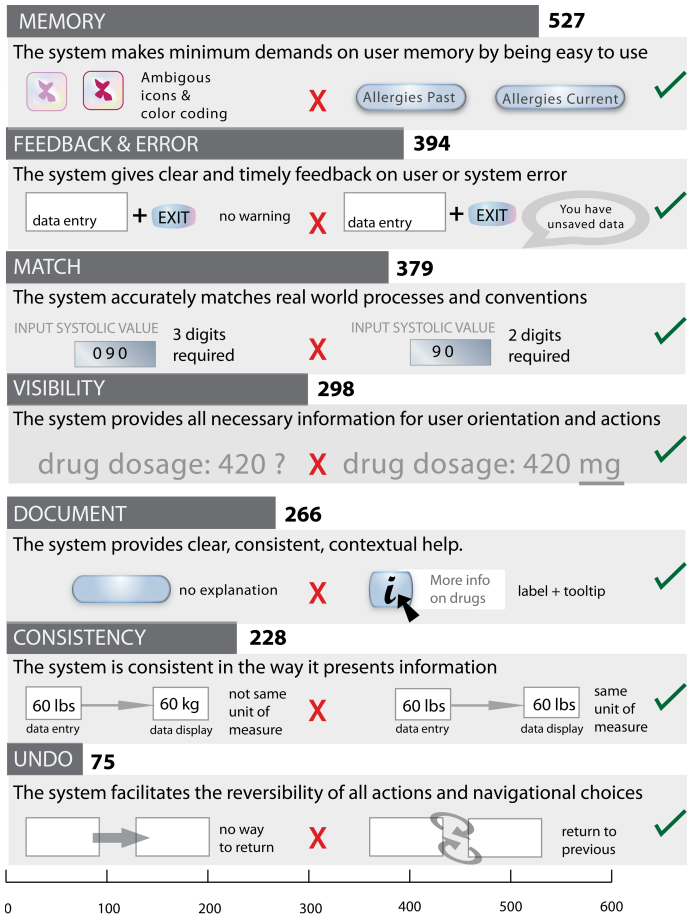


Figure 1.

DISCUSSION

We evaluated the usability of five commercial EHR products for their potential impact on clinical environments in terms of task completion time and adherence to good usability design principles. Our findings are consistent with concerns expressed in a Institute of Medicine (IOM) report on health IT and patient safety citing "poor interface design, poor workflow and complex data interfaces" as serious threats to patient safety in clinical environments [140].

We found some tasks took significantly longer to complete (e.g., clinical summary and CPOE). Increased time reflected increased complexity of the task and potential risk.

While our evaluation was performed on systems that had already been developed, it was directed at identifying efficiency benchmarks for comparing usability across systems. To the best of our knowledge, such benchmarks do not currently exist in the field of academic or commercial EHR developers/researchers. Our predictions of task performance times are of particular importance when viewed in the context of a real-life clinical encounter. For example, a typical patient visit that involves: 1) recording demographics, vital signs, smoking status, 2) calculating BMI, and 3) recording problem list, medication list, medication allergy list, and a set of orders through CPOE would take 11 minutes and 19 seconds on average. This represents the optimal performance time interacting with the EHR (assuming no user errors), and does not include the time spent examining or interacting with a patient.

Through expert reviews we discovered tasks that were more complex and took longer had the highest number of usability problems. However, even shorter tasks had a number of usability challenges. Some issues had potential for causing patient safety-related events. Of particular concern were interfaces requiring high memory load or mismatches between clinical workflow, internal representation and system design.

More user-based testing is needed to identify specific challenges users face and directions for interface improvement. Our approach and results are complementary to the recently released National Institute of Standards and Technology (NIST) EHR Usability Protocol, which focuses on ensuring the design of safe EHRs through expert reviews and summative user testing [141].

Future Direction

Despite several advantages for the RUA approach, limitations exist relating to the comprehensiveness of our findings. First, we used expert user performance for computing task completion times. This is likely to differ from actual task performance in clinical settings. Nevertheless, KLM predicts an ideal measure of time required to complete certain routine tasks. We plan user testing in field settings to validate and further explore task completion times. Second, we used only two usability evaluation methods in our RUA protocol. For usability evaluation to be comprehensive, additional user-centered methods such as field studies or

laboratory-based experiments (as outlined in the NIST EHR Usability Evaluation Protocol) would be useful capturing a greater number of usability problems and triangulating findings. Third, we did not use all the NIST defined meaningful use-related tasks for our assessment. However, the RUA can easily be expanded for other tasks as they are defined by NIST or others as part of future stages of meaningful use.

IMPLICATIONS

Combining a predictive model of time required to complete a task with a more subjective, expert-based measure of usability violations provides significant flexibility understanding the structure of NIST meaningful use tasks. Our results show lack of efficiency in completing certain tasks in conjunction with incompatible user interfaces provides evidence for the "threats to patient safety" highlighted in the IOM report [140].

5: EHR Vendor Usability Practices

Raj Ratwani, PhD
National Center for Human Factors in Healthcare, MedStar Health

Rollin (Terry) Fairbanks, MD MS
National Center for Human Factors in Healthcare, MedStar Health

Zachary Hettinger, MD MS
National Center for Human Factors in Healthcare, MedStar Health

Natalie Benda
National Center for Human Factors in Healthcare, MedStar Health

ABSTRACT

A team of healthcare human factor experts interviewed 11 electronic health record (EHR) system vendors to better understand their implementation of user-centered design (UCD). The team learned both application practices and challenges. Interviews revealed three UCD implementation categories: rigorous UCD processes in place, basic UCD processes employed, and no UCD. Challenges that vendors faced in each category are described, including a lack of UCD understanding, problems integrating UCD into software development, and difficulty recruiting participants for usability studies. Results provide insight on the current state of UCD in EHR products and ideas that could improve usability.

INTRODUCTION

Electronic health records (EHR) systems have the potential to dramatically improve healthcare efficiency, quality and safety. To reach this goal, systems must be designed, developed, and implemented with a focus on usability and safe use [142]. This suggests vendors employ user-centered design (UCD) during product development and that healthcare users appropriately implement the system.

The Office of the National Coordinator (ONC) for Health Information Technology requires EHR vendors certify their products' usability and safety [6]. ONC safety-enhanced design requirements

specify vendors employ a UCD design process, conduct testing and report test results of at least eight UCD functions. While ONC does not describe a specific UCD design and development process, requirements reference the International Standards Organization (ISO) and National Institute of Standards and Technology (NIST) standards [143]. Despite these requirements, however, EHR usability remains problematic [144, 145].

Our team studied vendor UCD processes and researched new ideas to help vendors improve UCD [7]. The EHR marketplace is diverse. Some vendors have less than ten employees and offer only ambulatory care products, others employ more than five thousand and market a suite of products. We conducted semi-structured interviews with a diverse representation of EHR vendors to learn about their implementation of UCD. Insight could help create better UCD design methodology and development tools. A clear understanding of vendor UCD processes and challenges can also assist policymakers, regulatory agencies and others dedicated to improving EHR systems.

APPROACH

A research team of human factors, clinician/human factors, and clinician/informatics experts visited eleven EHR vendors and conducted semi-structured interviews about their UCD processes. "Process" was defined as any series of actions that iteratively incorporated user feedback throughout the design and development of an EHR system. Some vendors developed their own UCD processes while others followed published processes, such as ISO or NIST guidelines.

Vendor recruitment. Eleven vendors based on market position and type of knowledge that might be gained were recruited for a representative sample (Table 1). Vendors received no compensation and were ensured anonymity.

Vendor	Estimated Revenue	Estimated Employees	Estimated Usability Size
Vendor 1	$1 billion+	6,000+	15 people
Vendor 2	$100 million+	2,200	30+
Vendor 3	$100 million+	650	NA
Vendor 4	$100 million+	2,000	30+
Vendor 5	$40 million	500	1-5
Vendor 6	$20 million	250	1-5
Vendor 7	$20 million	150	NA
Vendor 8	$10 million	60	NA
Vendor 9	$300,000	10	NA
Vendor 10	$1 billion+	6,000+	30+
Vendor 11	$1 billion+	6,000+	NA
Range	**$300,00 - $1 billion**	**10-6,000+**	**0-30+**

Table 1. Demographics of vendors visited.

Interview process. Semi-structured interviews were conducted at each vendor site with business analysts, product managers, software developers and user experience experts. Five visits were full day and six half day. Interviews were generally held with one to three employees at a time. At least two members from the research team travelled to each site and interviewed together to allow documentation redundancy. Dr. Ratwani was present at all vendor visits accompanied by either Dr. Fairbanks or Dr. Hettinger to ensure both human factors and clinical expertise.

The research team asked open-ended questions about UCD processes being employed, challenges the vendor faced practicing UCD, and questioned what could facilitate their UCD process. The research team asked follow-up questions to extract detailed information.

Data collection and analysis. Researchers documented responses during the semi-structured interviews with notes integrated by a research assistant immediately after the interviews. Once all of interviews were complete the research team identified emerging themes that characterized UCD processes employed by vendors.

RESULTS

Vendors generally fell into one of three UCD implementation categories:

Well-developed UCD: These vendors had a refined UCD process, including infrastructure and the expertise to study user requirements, an iterative design process, formative and summative testing. Importantly, these vendors developed efficient means of integrating design within the rigorous software development schedules common to the industry, such as maintaining a a network of test participants and remote testing capabilities. Vendors typically employed an extensive usability staff.

Basic UCD: These vendors understood the importance of UCD and were working toward developing and refining UCD processes to meet their needs. These vendors typically employed few usability experts and faced resource constraints making it difficult to develop a rigorous UCD process.

Misconceptions of UCD: These vendors did not have a UCD process in place and generally misunderstood the concept, in many cases believing that responding to user feature requests or complaints constituted UCD. These vendors generally did not have human factors/usability experts on staff. Leadership often held little appreciation for usability.

About a third of our vendor sample fell equally into each category. We found an apparent relationship in overall vendor size and category, however, given our small sample, this may not be statistically significant.

Challenges to practicing UCD

Vendors in each category identified challenges faced integrating UCD into their development cycle.

Well-developed UCD: A common theme was difficulty conducting detailed studies of subspecialty workflows. Investment required to conduct studies on the large number of medical subspecialties was considered significant, so vendors tended to focus on the largest markets. Vendors also described challenges getting users to share problems associated with using their EHR product, feedback critical to vendor improvements.

Basic UCD: Vendors in this category required additional resources and knowledge to efficiently and effectively employ UCD processes. In particular, these vendors face challenges recruiting participants for usability studies, had difficulty developing detailed use case scenarios to test product, and required assistance learning how to efficiently integrate UCD into software development.

Misconceptions of UCD: These vendors lacked understanding of UCD processes, the importance of UCD in product development, and the need of UCD for patient safety. A successful business case to leadership for UCD investment had not been made.

Nearly all vendors participated in the study identified rigorous development timelines as a significant challenge to practicing UCD. Meeting the summative testing requirements for EHR certification was consistently described as challenging and resource intensive.

DISCUSSION

Characterization of EHR vendor UCD practices and challenges provides unique insight into EHR product development, information that could help researchers and policymakers target their efforts to the specific needs of vendors. Our team identified several ways to facilitate EHR vendor UCD processes.

Facilitating Vendor UCD

Increase knowledge on conducting usability studies: Many Basic UCD vendors had specific questions about the UCD process itself. For example, they were unsure of the number participants required for summative testing, the amount of training to provide participants, where to conduct studies, and the experience/expertise that participants should have. Providing vendor resources that clearly and concisely describe this information may lead to dramatic improvements in their application of UCD.

Improve participant testing: It is impossible to practice UCD without participants for usability studies and focus groups. Several vendors described recruiting challenges. While some vendors were able to rely on their installed user base, finding participants that have not been exposed

to the vendor's product was a significant challenge. Methods should be developed to assist vendors recruit participants.

Develop use case scenarios: Several vendors had difficult recruiting experts to develop contextually rich use case scenarios to test product. Creating a set of use cases that could be leveraged by any vendor could save vendor resources and perhaps raise product quality.

Policy implications

Safety-enhanced design (SED) is a 2014 certification criteria for EHR. Many vendors expressed disagreement with SED mandate of summative testing. In particular, vendors with well-developed UCD processes felt the requirement forced them to dedicate a large staff to conducting tests with few, if any, benefits. Because these vendors had a rigorous UCD process in place, they believed usability issues were uncovered by the UCD process before summative testing and, therefore, testing to discover UCD problems to be redundant. Vendors said not requiring summative tests could free staff for other aspects of product development.

Required summative testing, however, may be effective forcing vendors with misconceptions of UCD to dedicate usability resources. It may be beneficial to consider a policy that embraces the diversity of UCD vendor capabilities, for example, offering vendors the option of demonstrating rigorous UCD processes *or* summative testing. Policymakers may want to explore whether modifications to SED certification could better serve the vendor and user communities.

IMPLICATIONS

Improvements in EHR usability cannot be achieved by focusing solely on vendors; a holistic approach with all health information technology stakeholders—vendors, healthcare providers, policymakers and patients—is required. Still, identifying vendor UCD practices and challenges has practical value improving the usability of EHRs. Understanding current UCD processes and the specific challenges vendors face can help researchers create new tools to facilitate UCD, in particular those in No UCD and Basic UCD categories. Appreciating

vendor community diversity and UCD practices also provides policymakers with greater context to inform decision making.

6: Turf Usability Tool Suite

Min Zhu, MD, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Deevakar Rogith, MBBS
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Amy Franklin, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Muhammad F. Walji, PhD
The University of Texas Health Science Center at Houston School of Dentistry

Jiajie Zhang, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

ABSTRACT

Usability is a major barrier to electronic health record (EHR) adoption and meaningful use. Traditional usability assessments include pen and paper, and stand-alone recordings. We developed *Turf*, software to evaluate, document, and improve EHR usability in an all-in-one product. *Turf* can be used in both usability evaluation and the testing and design of EHR systems. Intended to support lab and real world environments, *Turf* streamlines conducting and reporting formative and summative assessments, including user testing and heuristic evaluation.

INTRODUCTION

Poor usability is a barrier to adopting electronic health records (EHR) [21, 146]. Despite significant research on the importance of EHR usability in patient safety and quality of care, EHR systems' ease of use and learnability are problematic [147]. Usability testing can identify problems during product development, but assessment can be laborious. Conventional usability testing is often word processor, spreadsheet, or pen and paper-based.

An EHR system's required tasks, its users' needs, how it represents health information, and its functional capabilities (TURF) determine its usability. We developed a unified framework for evaluating electronic health records system usability called TURF [7] (Chapter 2). TURF is: 1) a theory for describing, explaining, and predicting usability differences, 2)

a method for objectively defining, evaluating, and measuring usability, and 3) a set of usability measures. The Office of the National Coordinator for Health IT requires vendors adhere to safety-enhanced design (SED) [148]. SED mandates EHR vendors document their user-centered design (UCD) processes and conduct summative usability testing. Legislation also requires test results be reported in a "common industry format." Based on the TURF framework, we developed *Turf*, an all-in-one software tool to help EHR vendors evaluate, document, and improve the usability of their products.

APPROACH

Turf was designed and developed using a work-centered approach. Functional requirements were based on extensive experience conducting usability assessments, SED, and national guidelines and protocols published by the National Institute of Standards and Technology (NIST):

- (NISTIR 7741) NIST Guide to the Processes Approach for Improving the Usability of Electronic Health Records
- (NISTIR 7742) Customized Common Industry Format Template for Electronic Health Record Usability Testing
- (NISTIR 7804) Technical Evaluation, Testing and Validation of the Usability of Electronic Health Records
- (NISTIR 7865) A Human Factors Guide to Enhance EHR Usability of Critical User Interactions when Supporting Pediatric Patient Care

Turf architecture

Because many clinical applications require a Microsoft Windows operating system, we based development on Microsoft Windows 8. Our major programming language was C#. Microsoft coding standards were adopted, Microsoft Visual Studio 2012® used as our integrated development environment, and Microsoft User Interface Design Guidelines for Windows presentation foundation (WPF).

Turf uses a model–view–controller design pattern. We divided the application into interconnected components to separate the internal representation of information from how information is presented to users. The central component consists of application data, business rules,

logic and functions. A view can be generated of any information, such as a chart or diagram. All *Turf* view layers in were coded in WPF. The controller accepts input and converts it to commands for the model or a view. *Turf* architecture was updated after each development cycle.

Development, testing and refinement

Scrum is a common approach to agile software design and was adopted to guide iterative, incremental development. SCRUM enabled us to test high-quality software on a regular basis and adapt new versions based on usability assessments, which were early and throughout the development lifecycle. Feedback obtained was critical to improving each *Turf* iteration.

Regression testing was used to find defects after major code changes. Test cases based on previous user stories were created. Users re-ran prior sets of test-cases and checked whether previously fixed faults re-emerged. Depth of testing depended on development phase and risks associated with new features.

PRODUCTS

Three versions of *Turf* have been released:

- *Turf 1.0* (client-server version) focused on core data capture capability for usability assessments including image and screenshot capture, video, and keystroke data. Data were stored in a server at NCCD.
- *Turf 2.0* (stand-alone version) provided data analytics and a customizable moderation function to help moderate usability studies. Based on user feedback, we found vendors more comfortable storing usability data locally rather than on a remote server.
- *Turf 3.0* (stand-alone version) included enhanced analytic features and supported a semi-automated processes to generate a standard Common Industry Format (CIF) report.

Turf 3.0 contains the three principle modules:

1. User testing: Whether in formative or summative stages of software development, *Turf* allows developers to assess user experience through user testing. *Turf* streamlines testing by pushing participants through tasks using customizable moderation in a process called Autoflow.

2. Heuristic evaluation: Usability evaluators can use *Turf* to capture images, video and annotate problems. Evaluators can also audit problems, rate severity, and print reports.

3. Analytics: Statistical methods are included to process raw data collected during user testing and heuristic evaluations to facilitate CIF reports. Methods include calculation of mean values, counts, descriptive analyses, and inferential statistics.

Turf can be downloaded at https://sbmi.uth.edu/nccd/turf/.

User testing module

Data capture and visualization

Turf can record audio, webcam video, screenshots, and keystroke data. User interface information about a specific control can also be captured as an to identify unique task paths for later analysis. Task paths can be annotated using predefined coding, which includes "typo," "wrong mouse click," and "wrong mouse movement" events.

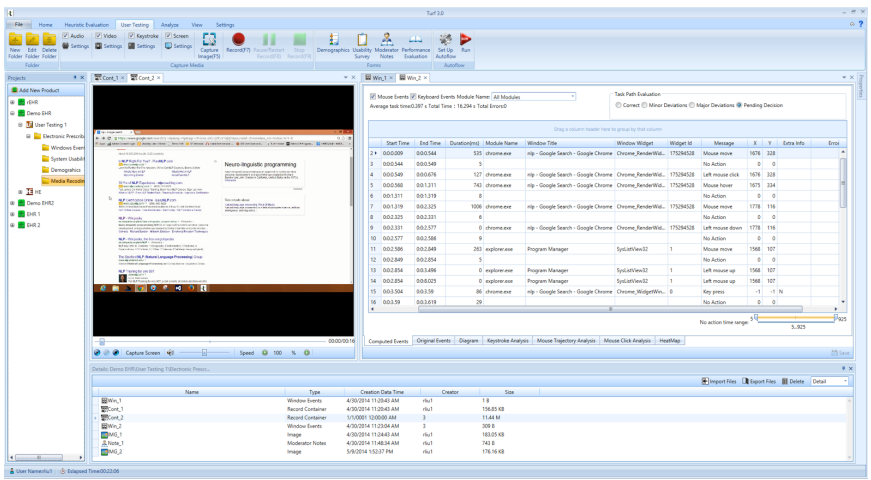


Figure 1: Data capture in the User Testing Module. The left panel presents the screen video and webcam in a picture-in-picture mode. The right panel presents keystroke data.

Autoflow configuration and execution

Managing user testing while maintaining data quality in a standardized process can be challenging. *Turf's* Autoflow allows moderators to customize routine testing steps with actions such as, "read PDF", "record participant ID," "play training video," "record screen

interaction," and "record audio." After a moderator creates an Autoflow process, sequential steps can be reused for multiple participants.

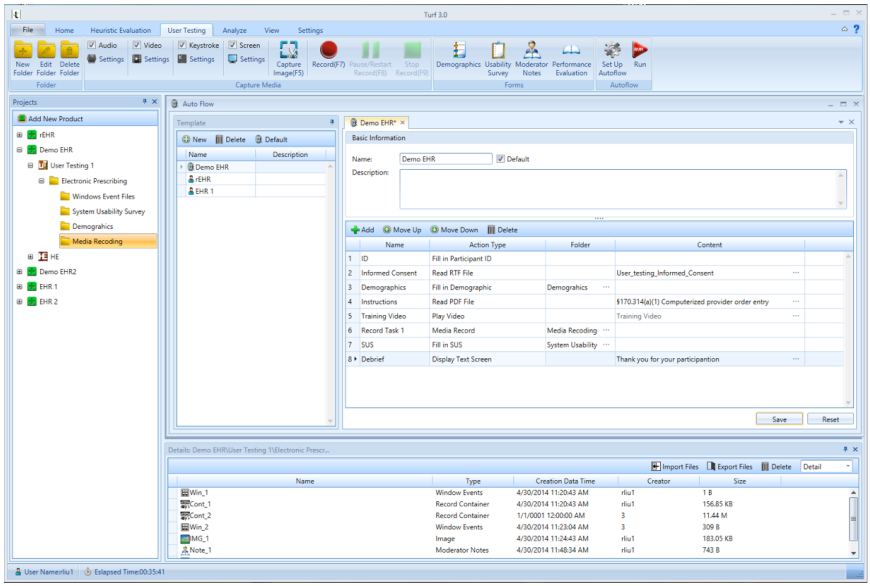


Figure 2: Autoflow setup screen used to streamline moderation of user testing.

Heuristic evaluation module

Highlight problems area on the screen and document problems

Evaluators can capture and annotate screenshots with record usability problems through the Heuristic Evaluation Module. Screenshot areas can be highlighted using a colored rectangle and problems annotated with customizable templates.

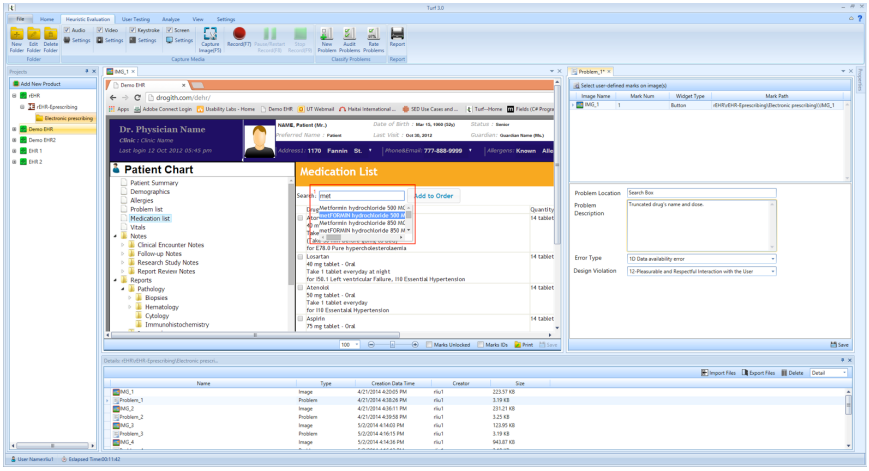


Figure 3: Heuristic Evaluation Module showing an EHR screenshot on the left with marks highlighting problem areas. The right panel shows a template that can be used to document and categorize a usability issue.

Review problem

Turf includes a Review function to ensure the precision and quality of problem descriptions. Review allows multiple evaluators to systematically audit marks and their associated problems.

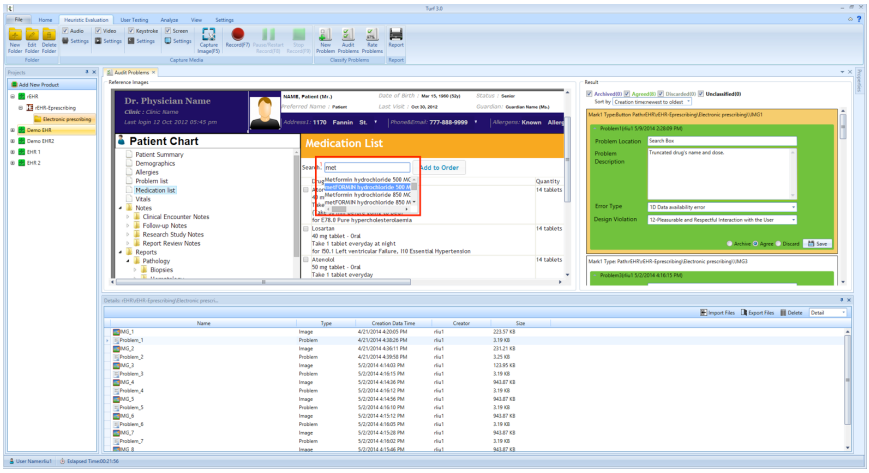
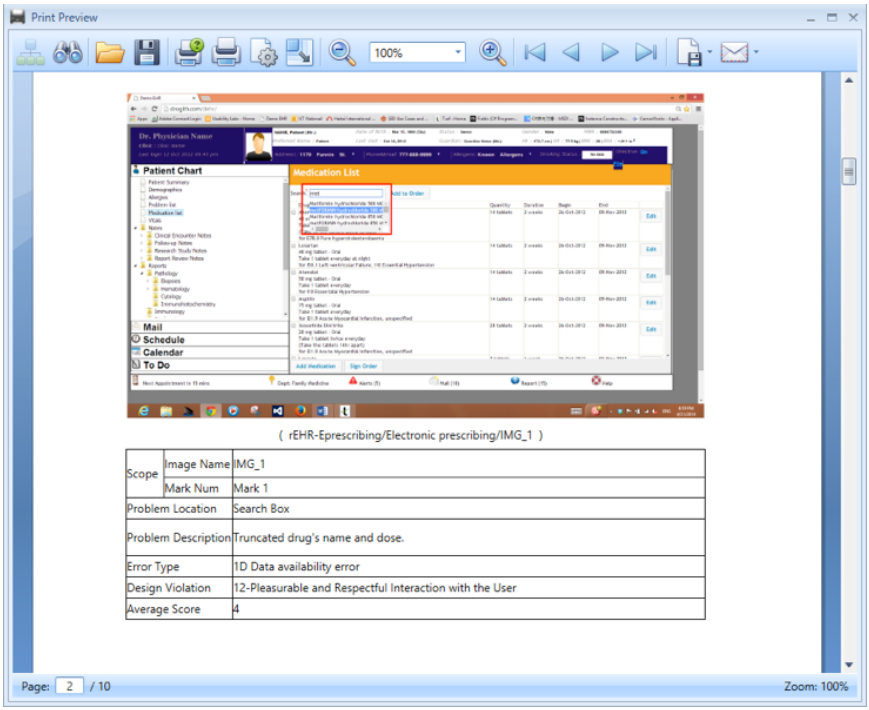


Figure 4: The auditing interface in the Turf *Heuristic Evaluation Module.*

Heuristic Evaluation Report

Evaluators can generate a report customized to show screenshots and associated problems.



(rEHR-Eprescribing/Electronic prescribing/IMG_1)

Scope	Image Name	IMG_1
	Mark Num	Mark 1
Problem Location		Search Box
Problem Description		Truncated drug's name and dose.
Error Type		1D Data availability error
Design Violation		12-Pleasurable and Respectful Interaction with the User
Average Score		4

Figure 5: Example of a Turf *Heuristic Evaluation Report.*

Analytics module

Descriptive analyses, including calculation of means, medians, sums, and counts are supported in *Turf*. Inferential analyses are also included, such as t-tests, u-tests, linear regression, and correlation.

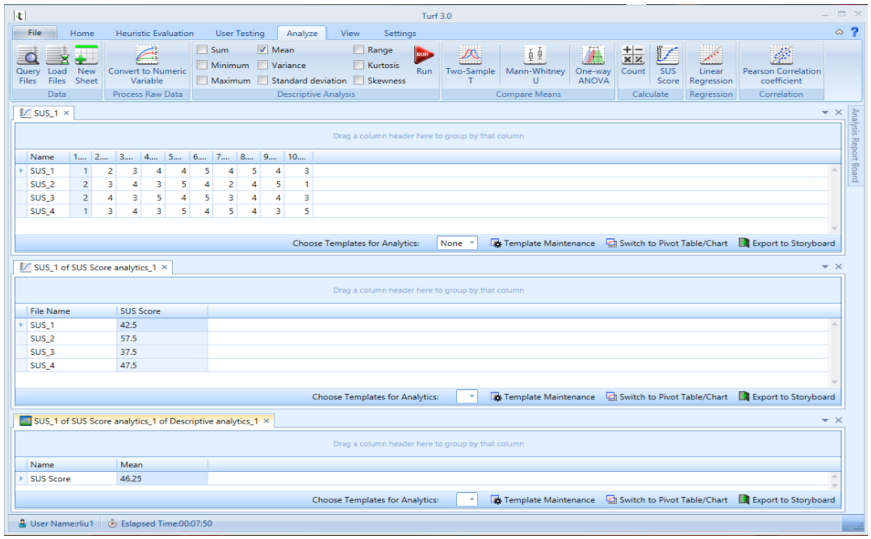


Figure 6: Data collected in Turf *can be sorted, converted, grouped and analyzed using statistic methods in the Analytics Module.*

Common Industry Format (CIF) report generation

Statistical results can be exported as raw data or inserted into the built-in CIF template. CIF reports can be edited and saved as a PDF or RTF file.

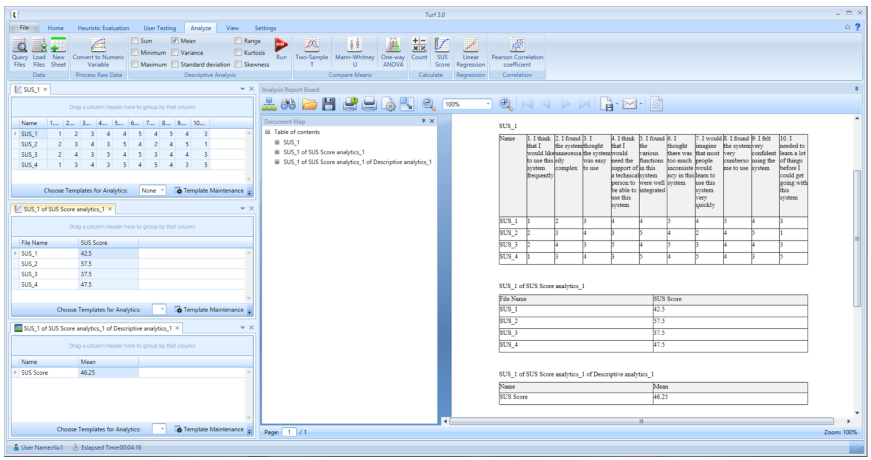


Figure 7: Turf *CIF report. Users can review data on the left panel and export data to the CIF template on the right.*

DISCUSSION

Turf is a software tool to evaluate an EHR system's usability and document problems. Two of the most common usability methods are supported: heuristic evaluation and user testing. The Heuristic Evaluation Module evaluators compare a user interface against established usability standards. *Turf's* built-in tools replaces simple screenshots and paper-based methods, helping expert evaluators identify actionable design issues. *Turf* also captures video and still images, including mobile devices. Reviewers can annotate images, score the severity of a usability violation, compare and adjudicate results with other reviewers.

The User Testing Module can be used for formative or summative assessment, with testing templates and user data capture. *Turf* semi-automates user testing metrics, such as path deviation and task time. *Turf's* Autoflow function streamlines the testing process by guiding participants through customized protocols using personalized moderation. *Turf* also captures audio/video recordings, screens, and mouse/keystroke events.

When assessments are complete, *Turf* supports data analysis with descriptive and inferential statistics. *Turf* also has flexible reporting capabilities including generation of CIF usability. More information about *Turf* is available at https://sbmi.uth.edu/nccd/turf/.

7: Safety-enhanced & User-centered Design

Amy Franklin, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Abstract

The US Centers for Medicare & Medicaid Services provide financial incentives to health providers for meaningful use of certified electronic health records (EHR) systems. Safety-enhanced design (SED) is a 2014 certification criteria and requires EHR vendors to:

1. Use a formal, user-centered design (UCD) process during EHR system development, and
2. Perform summative usability testing for portions of their EHR product.

We detail efforts to help EHR vendors better understand UCD, including disseminating:

- Requirements to meet SED certification, and
- International standards that meet SED legislation.

Introduction

The US Centers for Medicare and Medicaid (CMS) provide financial incentives to health providers using certified electronic health records (EHR) systems. To receive EHR incentive payments, a provider must show meaningful use of their EHR and the EHR vendor must certify their EHR product meets certification requirements for safety-enhanced design (SED) [149]. The Office of the National Coordinator (ONC) describes SED as a design process to reduce design-based errors within EHR interfaces, thereby improving the quality and safety of EHR systems [141]. Integral to this approach is the concept of user-center design (UCD). SED certification requires:

1. Documentation of the UCD process used during EHR development, and

2. Summative usability testing of their EHR, including:
    a. Computerized Provider Order Entry System (CPOE)
    b. Drug-drug, drug-allergy interaction checks
    c. Medication list
    d. Medication allergy list
    e. Clinical decision support
    f. Electronic medication administration record
    g. Electronic prescribing
    h. Clinical information reconciliation

Guidelines and protocols published by the National Institute of Standards and Technology (NIST) and International Organization for Standardization (ISO) have been suggested as possible routes for meeting SED certification requirements.

User-Centered Design

UCD is "an approach to designing a product or service in which the end user is placed in the center of the process" [150]. The intent is to build systems that accommodate users rather than forcing users to adapt. ISO describes six key principles of user-centered design [151]:

1. Design is based upon an explicit understanding of users, tasks and environments.
2. Users are involved throughout design and development.
3. Design is driven and refined by user-centered evaluation.
4. The process is iterative.
5. The design addresses the whole user experience.
6. The design team includes multidisciplinary skills and perspectives.

NIST, ISO, and ONC SED criteria are not prescriptive. ONC set SED requirements without limiting vendors to specific UCD processes or summative testing procedures. To fulfill certification requirements, vendors must only submit documentation specifying the UCD process(es) used. This allows for significant flexibility in achieving SED. Written

reports, documentation of procedures or providing acknowledgement of having followed ISO standards are a means of meeting requirements.

Approach

We participated in conversations with several EHR vendors over months regarding their summative testing plans. We also commented on UCD protocols, analyzed issues and offered reporting ideas. Discussions varied by vendor understanding of UCD and capabilities for summative testing. We also summarized ISO standards to assist vendors document relevant UCD practices.

Results

ISO standards provide one way to document UCD approaches during EHR system design. The standards are content rich and conceptually dense, making it difficult for some vendors to determine which standards apply and how to select between them (see Chapter 5). We provided a synopsis of ISO standards relevant to UCD and SED (Table 1).

Standard	Title
ISO 9241-11	Ergonomics Requirements For Office Work With Visual Display Terminals — Part 11: Guidance On Usability
ISO 9241-210	Ergonomics of Human-System Interaction— Part 210: Human-Centred Design for Interactive Systems
ISO/TR 16982	Ergonomics of human-system interaction — Usability methods supporting human-centred design
IEC/ISO 62366	Medical devices — Application of usability engineering to medical devices
ISO/IEC 25062	Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports

Table 1. ISO standards applicable to UCD and SED [151-155].

Each summary described an ISO standard's scope, definitions, rationale and was cross-referenced to related standards. References were also made to other UCD methods or principles. Summaries were 4-6 pages long, compared to typical 50-page ISO documents. Table 2 describes the scope of summarized standards.

Standard	Title	Provides	Does not provide
ISO 9241-11	Ergonomics Requirements For Office Work With Visual Display Terminals— Part 11: Guidance On Usability	Defines usability, describes benefits of integrating usability into the design process, provide guidances on how to specify the requirements needed measure the usability of products, how to evaluate usability.	Comprehensive coverage of ergonomic design objectives or human-centered design processes
ISO 9241-210 [ISO 9241-210 updates and replaces ISO 13407:1999.]	Ergonomics of Human-System Interaction— Part 210: Human-Centred Design for Interactive Systems	Provides information about human-centered design available to managers of hardware, software, and redesign processes in order for them to help identify and plan human-centered design activities.	Detailed human-centered methods nor does it detail project management.
ISO/TR 16982	Ergonomics of human-system interaction — Usability methods supporting human-centered design	A resource for project managers to better understand the methods of usability testing so that they can make more informed decisions about how to support human-centered design as described in ISO 13407 (later revised in ISO 9421:210).	
IEC/ISO 62366	Medical devices — Application of usability engineering to medical devices	A process for a manufacturer to analyze, specify, design, verify and validate usability, as it relates to safety of a medical device.	It does not address risk and errors associated with atypical use.
ISO/IEC 25062	Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports	Provides the common industry format (CIF) for reporting the results of usability testing related to software. The scope of ISO/IEC 25062 is it NISTIR 7742, conforms to ISO/IEC 25062, and was written specifically for reporting on EHR systems and have been made freely available.	

Table 2. Scope of ISO standards applicable to UCD and SED (available at https://sbmi.uth.edu/nccd/SED/).

Linking to ISO

Referencing appropriate ISO standards requires not only understanding the standards, but mapping them to appropriate stages of EHR design and development. We provided linkages to ISO standards based on components of the UCD process (Figures 1 and 2).

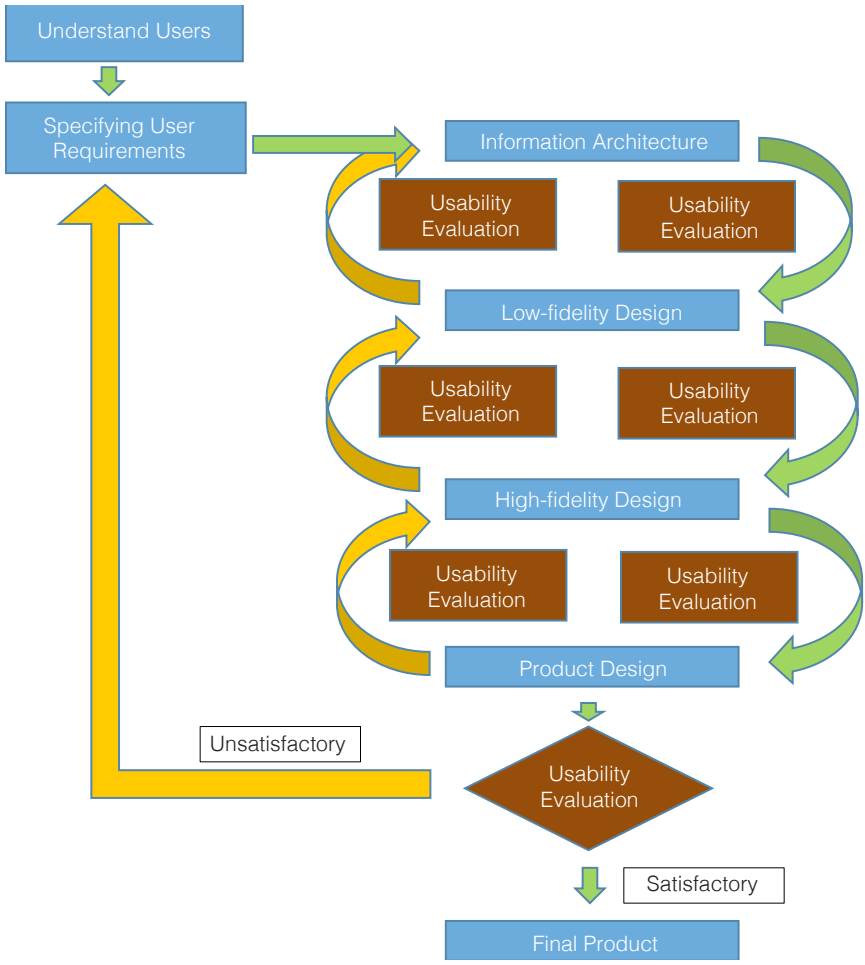


Figure 1. The UCD process.

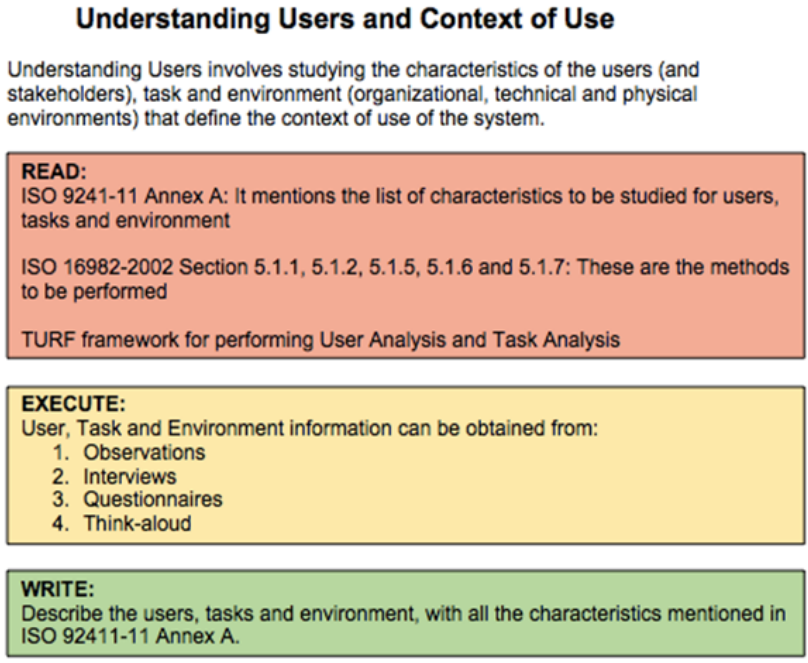


Figure 2. ISO standards mapped to the first step of the UCD process.

Discussion

SED certification encompasses a range of UCD methods and processes for certification. While flexibility benefits many within the EHR vendor community, for others the lack of a specified process is problematic. Publicly available reports on EHR usability are available through ONC's Certified Health IT Product List [156], however, few summative tests or UCD experience reports are available. General references for following standard UCD procedures can be found [157], but finding practical references for vendors implementing UCD is difficult. Growing the literature on EHR UCD implementation is needed.

Suggested Reading

Health information technology: standards, implementation specifications, and certification criteria for electronic health record technology, 2014 edition; revisions to the permanent certification program for health

information technology. Final rule. (2012). Fed Regist, 77(171), 54163-54292.

Schumacher, R. M., & Lowry, S. Z. (2010). NIST Guide to the Processes Approach for Improving the Usability of Electronic Health Records. (NISTIR 7741). Gaithersburg, MD: National Institute of Standards and Technology. Retrieved from http://www.nist.gov/itl/hit/upload/Guide_Final_Publication_Version.pdf.

8: Use Cases

Amy Franklin, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Abstract

New electronic health records system safety-enhanced design requirements underscore the need for validated, contextually-rich user test scenarios. We developed assessment tools to measure system usability and identify safety risks, and detailed methods of generating these materials.

Introduction

Information technology (IT) usability has implications beyond adoption or ease of use. Testing real users in actual tasks is one of the most effective means of assessing the safety of an electronic health records (EHR) system for patient care. The Office of the National Coordinator (ONC) 2014 EHR safety-enhanced design (SED) certification criteria require user testing and reporting of specific tasks, including e-prescribing, medication list maintenance and clinical information reconciliation [141]. To assess EHR procedures, activities such as entering a prescription are embedded into scenarios or descriptions of hypothetical work. While the National Institute of Standards and Technology (NIST) Test Procedure for §170.314(g)(3) Safety-enhanced design [141] is a valuable reference for system evaluation, greater specification of roles, workflows, and EHR system specific task descriptions are necessary to replicate real world engagement. EHR test protocols should reproduce a system's user interface, rely on users' knowledge, simulate users' thought processes while performing tasks (their "mental model"), support inferences users make interacting with the system, and produce valid output.

Use cases describe "a system's behavior as it responds to a request. Each use case is represented as a sequence of simply steps, beginning with a user's goal and ending when that goal is fulfilled" [150].

ONC and NIST use cases describe functional requirement for SED testing [141]. Broad datasets are provided, but without specifying workflows. For each test procedure, definitions capture certification criteria for recording, modifying and retrieving information. For example, a system must "enable a user to electronically record, modify, and retrieve a patient's vital signs including, at a minimum, the height, weight, and blood pressure." NIST provides sample values, such as height 66 inches, weight 61.2 kg and blood pressure 120/80 are given, which may be insufficient for testing. To create a use case with appropriate context for more complicated cases, however (such as entering a prescription and utilizing decision support), more data entry fields are needed. To generate comprehensive use cases, data must be enriched to detail roles, business processes, goal(s), and events that might occur achieving a goal [158].

Approach

We created an organizational framework and workflow for each required 2014 SED test method [159], such as medication reconciliation, computerized provider order entry, etc. Our use cases include task-required data and their clinical context (e.g., user role, purpose, workflow, etc.).

Our use cases were informed by observing and interviewing clinicians, and verified by clinical collaborators. Synthetic patient data, including historical information, were generated by mining two large patient datasets (including 100,000 patients in one sample) from major metropolitan areas. We used real world data to replicate test participant (i.e., doctor and nurse) experiences in common patient interactions. For example, we found the average length medication list has three drugs. We selected the most frequent conditions, medications and labs for inclusion in our test database and developed advanced scenarios for testing less typical cases, such as long medical histories or lengthy drug regimens.

Output

We envisioned two users of each scenario: a clinical test participant and a usability expert. Participant instructions, expectations and guidance to identifiable outcomes were embedded within a realistic narrative, providing motivation for prescribing drugs or ordering lab tests without

prescriptive direction. Narratives provided patient name, age, chief complaint, and course of immediate action.

Example: medication reconciliation. Consideration of a patient's age, gender and overall health are needed to push this summative test from data entry to real world simulation.

> Julie, a 62-year old female, has come to your clinic today for followup of her hypertension, which you diagnosed six months ago. During that visit, Julie also mentioned taking 20 mg Lipitor. In that Julie is a relatively new patient to your practice, you are concerned that her medication list is not up-to-date.
>
> *Your first task is to review the medications Julie is currently taking to ensure they are complete and correct.*
>
> After locating the medication list and reviewing it, you ask Julie to verify if she is still taking 20 mg Lipitor. However, this time Julie has the bottle with her and she notices that it states 40 mg.
>
> *Your second task is to correct her medication list.*
>
> After making the Lipitor correction, you ask Julie if she is taking any other medications. She reports she is taking Centrum Silver for Women, ibuprofen for regular aches and pain, and Claritin for allergies.
>
> *Your third task is to enter these drugs in her medication list.*

Task requirements were then given in two formats: 1) a bulleted list and 2) a tabular form of data entry components (Table 1).

> 1. Navigate to Julie's full medication list and verbally state the name and strength of the first medication on Julie's medication list.
> 2. Navigate to a place where you can update drug information and change the Atorvastatin dosage from 20 mg to 40 mg using as much

information as necessary from Table 1, and then verbally state when you believe you have successfully completed this step.

3. Navigate to a place where you can add a medication to her medication list and add Centrum Silver for Women, ibuprofen, and Claritin using as much information as necessary from Table 2. Verbally state when you believe you have successfully completed this task.

	Drug Information
Drug Name	Lipitor
Generic Name	Atorvastatin
Strength	40 mg
Dose	Once daily at night
Duration	30 days
Form	Tablets
Route	Oral
Dispense Amount	30
Brand Necessary	No
Refills	2 refills
Reason	Hypercholesterolemia

Table 1. Tabular form of data entry components.

We included data in both formats to reinforce required tasks and limit memory demands placed on test participants (i.e., recall task details). SED certification requirements include reporting task time. Requiring test participants to shift through documents to retrieve information adds burden, increases risks of error and interference across multiple tasks in a single assessment.

Additional details were added to the table if required for complete medication entry. Duration, dose, route and reasons were often required fields. Our instructions requested navigation to key points within the

system without directing participants to certain locations or providing keystroke level details of task completion.

For usability experts, we provided moderator guides with step-by-step support for each component. We recommended NIST 7804 for procedures [141] and detailed other methods of data collection for system set-up, scripts for participant instruction, task start and end times, and scoring guides.

Required reported results for each task include success/failure, time to completion, and path deviation/error. We developed moderator guides with clear start and end points within each task to support recording of measures. For example:

> 1. Navigate to Julie's full medication list START POINT and verbally state the name and strength of the first medication on Julie's medication list.
>
> *Task Success/Failure, Time A*
>
> 2. Navigate to a place where you can update drug information and change the Atorvastatin dosage from 20 mg to 40 mg, using as much information as necessary from Table 1, and then verbally state when you believe you have successfully completed this step.
>
> *Task Success/Failure, Time B*
>
> 3. Navigate to a place where you can add a medication to her medication list and add Centrum Silver for Women, ibuprofen, and Claritin using as much information as necessary from Table 2. Verbally state when you believe you have successfully completed this task.
>
> *Task Success/Failure. Use case complete, Time C.*
>
> *TIME A+B+C = Total Task Time*
>
> *Three potential points of failure (locate, modify & record)*

For path deviation/error, we delineated task goals. Moderator materials for each test included optimal paths for completing subtasks. This improved deviation/error recognition and provide participant support.

> Path Deviation
>
> Navigate to Julie's full medication list *and verbally* state the name and strength of the first medication on Julie's medication list.

Optimal paths allowed us to guide test participants when problems arose. Example:

> Home
>
> - Find Patient
> - Enter Patient's Name
> - Click on Patient's Name from list
> - Click on Medication List tab (on left side navigation panel
> - Click on Current Medication List
> - Locate first medication

We created scenarios for nine SED use cases: Medication List, Computerized provider order entry (CPOE) LAB and CPOE IMAGE, Drug-Drug Interaction, Drug-Allergy Interaction, Medication Allergy List, e-prescribing, CDS reminders and CDS system inference. We also developed advanced scenarios for pediatric cases in which drug, age and/or weight trigger decision support notices, and more complicated decision support triggers regarding drug-pregnancy contraindications and serious harmful impacts.

DISCUSSION

2014 ONC SED certification is an opportunity for safer EHR systems through usability testing. Gaps in resulting reports have sparked discussion about further ways to enhance the role of usability in SED. Proposed 2015 Rule Making includes changes to usability requirements: formative testing, structure of testing (roles, number of participants, and

maybe more standard testing procedures.) We believe greater structure in test scenarios will enhance evaluation of EHR safety considerations and enrich the discussion of shared difficulties across systems.

9: Education & Training

Amy Franklin, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

ABSTRACT

Educating and training health information vendors on usability may lead to better HIT and electronic records systems. We describe efforts to develop usability education materials, including an introductory course and a compilation of training resources.

INTRODUCTION

Usability is key to patient-centered cognitive support. A 2009 report by the National Research Council found serious gaps in health information technology, concluding patient-centered cognitive support should play a central role [5]. The importance of HIT usability and patient-centered cognitive support is evidenced by the Office of the National Coordinator (ONC) for Health Information Technology's requirement that EHR vendors certify their products' usability and safety [6]. Electronic health records (EHR) vendors use, understanding or appreciation of key usability principals, however, varies (see Chapter 5). Usability education and training offers a potential solution.

Usability education can take many forms, from in-person didactics to online courseware. EHR vendors may have an in-house group devoted to user-centered design and employ rigorous safety-enhanced design (SED) practices from product conception to delivery. Other vendors may have only a few members dedicated to UCD and SED. Still others may have no in-house expertise.

APPROACH

EHR vendor usability use and methods were surveyed and analyzed (Chapter 5) and vendors potentially benefiting from usability education identified. We conducted a search of existing courses, books, and public or private educational usability training services. Resources primarily targeted website development and physical product creation, appearing

insufficient to apply usability to EHR systems. We devised a ten-part series of lectures and developed hands-on opportunities to learn usability principals.

RESULTS

University-based programs

The Human Factors and Ergonomics Society (hfes.org) lists schools that grant bachelors, masters, or PhDs in human factor-related fields or with a concentration in related areas (engineering psychology, human factors, industrial and operations engineering, etc.). Certificate programs are a less time intensive means of learning and degrees. We researched certificate programs and found nine universities offering certificates in human factors, user-centered design, or human computer interaction. Length and content varied by program, from three -day to semester-based courses. While no program focused on EHR, many offered courses on measuring user experience and task analysis (Table 1).

Course	Summary	Organization	Duration	Location
User Experience (UX) Certificate	• Field Methods and User Research • Designing the User Experience • Human Factors and the User Experience • Designing for Accessibility • Measuring Emotional Engagement • Managing a User Centered Development Process	Bentley University	nNAa	Online l Onsite
UX Boot Camp	• Elements of the user experience • User research and market segmentation • Design Implementation and innovation • Assessment and measurement • Process improvement and success metrics	Bentley University	5 days	Waltha, MA
Usability Evaluation Techniques	• Contextual inquiries • Focus groups • Heuristic (or expert) reviews • One-on-one user testing	Michigan State University	NA	Ann Arbor, MI
User-Centered Design	• Best practices related to the entire technology development life-cycle • Initial technology design • User analysis • Development issues • Evaluation of user performance	Michigan State University	NA	Ann Arbor, MI
Certified User Experience Professional (CUEP) Workshop	• Usability test design and execution • Lifetime access to Texas Tech's usability faculty and literature	Texas Tech University	3 days	Lubbock, TX
The Human Computer Interaction Certificate	• Understand emerging human computer interface technologies • Understand human cognitional behavioral methods and usability techniques • Articulate societal and ethical issue related to human computer interaction • Overview of the latest human computer interaction research in multiple disciplines	Iowa State University	12 hours	Ames, IA

Course	Summary	Organization	Duration	Location
Human Factors and HCI	• Human factors • Human-computer interaction • Interface design • Usability and evaluation • Basic principles of human factors • The importance or ergonomics and anthropometrics • Discuss the role and importance of human factors and human computer interaction • Describe some basic principles of user centered interface design • Outline elements of usability testing and systems evaluation	Pennsylvania State University	3 hours	Univ. Park, PA
Human-Computer Interaction Design	• Interaction design • Professionally interface design • Strategic design planning	Indiana University	36 hours	Blooming -ton, IN
Human-Computer Interaction	• Introduction to human-computer interaction, design, evaluation and implementation of interactive computing systems for human use • Communication between people and computers • Capabilities of people to use omputers • Concerns that arise in designing and building interfaces • Design trade-offs • The process of specification, design and implemenation of use interfaces	Illinois Institute of Technology School of Applied Technology	NA	Chicago, IL

Table 1. University certificate programs related to usability.

We also researched private organizations offering education on usability and user experience. Sixteen companies offered education on topics such as user-centered analysis, interaction design, and basic user experience training. Formats varied from one hour to multi-day programs (Table 2).

Course	Summary	Organization	Duration	Location
User-Centered Analysis	User profiling, data gathering, task analysis, transitioning to design	Human Factors International	3 days	Multiple Worldwide
Practical Usability Testing	Design, analyze, present test results, refine facilitation technique, remote testing, comparison tests, how to use server logs for usability	Human Factors International	2 days	Multiple Worldwide
Web & Application Design	Implement user-centered requirements into usable designs, navigation, presentation, content, interaction, and how website and application design strategies differ and overlap	Human Factors International	3 days	Multiple Worldwide
Putting Research into Practice	Summaries of literature in human-computer interaction, psychology, computer science, technology, usability engineering, practical implications of research, how to conduct user-centered design, and how to apply exploratory research in real-world applications	Human Factors International	2 days	Multiple Worldwide
User Interface Principles Every Designer Must Know	Origins of HCI, designing better interfaces, input devices, interaction style, universal design, complex interface features, and HCI methods	Nielsen Norman Group	1 day	Multiple USA Locations; Berlin, Germany
Usability in Practice: 3-Day Camp	Usability foundations, choosing research methods, team strategies, measuring usability, determining scope of study, setting up user studies, recording observations and data, facilitation techniques, tracking usability findings, analyzing findings, and reports for usability studies	Nielsen Norman Group	3 days	San Francisco, CA Oct 6-8, 2013
UX Basic Training	The business value of UX design, foundation of UX, understand the purpose and roles of UX professionals, as well as the stages of organizational UX maturity	Nielsen Norman Group	1 day	Multiple Worldwide
Interaction Design: 3-day Course	Principles of interaction design, Information Theory, Fitt's Law, and increasing the power and visibility of HCI and HCI groups	Nielsen Norman Group	1 day	San Francisco, CA Oct 6-8, 2013
Usability Testing Training	Design and plan usability tests, run testing sessions, and create templates	Webcredible	1 day	London, UK

Course	Summary	Organization	Duration	Location
Agile UX: Incorporate Usability into Agile Development	Implementing Agile and User-centered design, methodology pluses and minuses, and how to incorporate usability into Agile processes	Experience Dynamics	1 hour	Online
Usability Testing Skills Refresher	Usability testing process and types of testing, seven golden rules of usability testing, usability testing as an art and science, professional testing skills, as well as tips and techniques	Experience Dynamics	1 hour	Portland, OR
Usability Testing Metrics	Types of usability testing, observable dta, measuring subjective stuff, usability metrics, and tools for usability testing	Experience Dynamics	1 hour	Online
Usability Testing Methods Training	Planning your test, setting usability metrics, reporting on test data, analyzing results, and communicating results	Experience Dynamics	6 hours	Portland, OR
A Practical Guide to Usability Testing	Defining usability, writing a test plan, bias-free test moderation, how to classify behavioural observations, using Binomial Confidence Interval and z-scores to account for variation, measuring satisfaction with the SUS survey and the Microsoft Desirability Toolkit	UserFocus	1 day	On Demand
UXLabs Certified Usability Testing Professional (UCUTP)	Basics behind usability testing, understand Mobile Usability testing, and demonstrating the ability to conduct an effective usability test	UX Labs	6 weeks	NA
UXLabs Certified Usability Professional (UCUP)	Explain usability principles, methods and guide, conduct heuristic evaluations on sites to identify usability issues, conduct usability test sessions to identify usability issues	UX Labs	12 weeks	NA
UXLabs Certified Usability Analyst (UCUA)	Implement user research with various methodologies, implement competitive analysis using market research and SWOT, conduct heuristic evaluations to identify usability issues, design usability focused websites, web apps, SaaS applications	UX Labs	8 weeks	na

Course	Summary	Organization	Duration	Location
Usability Testing	Types of usability tests, how to integrate usability testing into your development process, cross-functional usability team, analyze users, tasks and context of use, how to design a test	ClickStart	1 day	Online
Usability Testing Training Course	How to design, plan and faciliate usability test sessions, how to analyze test results and effectively communicate results, logistics, technologies and ethics of usability testing	PeakUsability	4 hours	Multiple Locations in Australia
UX Professional Training Program	Web usability courses, custom seminars and workshops	Akendi Intentional Experiences	5 days	Multiple Locations in Canada
Certified User Experience Designer	Intro to UX design and experience thinking, information architecture design, mobile user experience design, GUI interaction design, and visual design for user experiences	Akendi Intentional Experiences	5 days	Multiple Locations in Canada
Certified User Experience Researcher	Intro to UX design and experience thinking, user experience research, information architecture research, and usability testing	Akendi Intentional Experiences	5 days	Multiple Locations in Canada
Certified User Experience Specialist	Intro to UX design and experience thinking, UX Research, Information Architecture, Usability testing, GUI Interaction Design	Akendi Intentional Experiences	9 days	Multiple Locations in Canada
Usability Testing Course	What is usability testing, creating an effective test plan, facilitating test sessions and analyze data, usability testing methods, and usability testing preparation	Akendi Intentional Experiences	2 days	Multiple Locations in Canada
Human Factors of Medical Devices	Human factors validation testing, HR/Usability report, case study 1:Planning and conducting human factors for Auto-Injector Drug Delivery Device, and application of Human Factors in Medical Device Design	Association for the Advancement of Medical Instrumentation	3 days	Arlington, VA

Course	Summary	Organization	Duration	Location
Conducting a Validation Usability Test	Determine the right time to conduct a validation usability test, ensure that usability testing focuses on use-safety and usability, contribute to usability test plans, gain the maximum benefit from observing usability tests, analyze the root causes of use errors, as well as close calls and operational difficulties	UL Knowledge Services	1 day	Chicago, IL
Human Factors Engineering in Medical Device Development	Understand human factors engineering expectations set by IEC 60601-1-6 and IEC 62366, understand the resources required to implement a comprehensive human factors program, plan human factors engineering programs that are properly scaled to the medical device in development, ensure that a design history file contains the necessary human factors engineering end-products	UL Knowledge Services	1 day	San Jose, CA
Medical Device Usability and IEC 62366	Achieve compliance with IEC 62366:2007, use design guidance in ANSI/AAMI HE75:2009, select and apply usability techniques, develop testable requirements for usability, manage the risk of use-error, develop a usability engineering process, document usability activities in the usability engineering file and HFE report	Bergo	2 days	London, UK
Training in Medical Device Usability	Achieve compliance with IEC 62366:2007, use design guidance in ANSI/AAMI HE75:2009, select and apply usability techniques, develop testable requirements for usability, manage the risk of use-error, develop a usability engineering process, document usability activities in the usability engineering file and HFE report	Bergo	2 days	Nivå, Denmark
Human Factors 101	Key human factors concepts and principles, human limitations and cognitive biases, user-centered design and evaluation methods, practical application of human factors to healthcare	Healthcare Human Factors	NA	Toronto, Canada

Course	Summary	Organization	Duration	Location
Human-Centered Design for Social Innovation	Introduction to concepts of human-centered design, use the design process to create innovative, effective, sustainable solutions	Acumen	5 weeks	Online
Human-Centered Design	Benefits of human-centered design, experience diagramming, heuristic review, affinity clustering, importance/ difficulty matrix, rough and ready prototyping	LUMA Institute	2 days	Multiple Locations
Error Prevention in Complex Care	Introduction to Human Factors, Cognition, errors, and sub-optimal behavior, Tools to manage error in emergency situations	Atrainability	1 day	UK
Advanced Clinical Trainer Skills	Course can be tailored; contact vendor for details	Atrainability	1 day	NA

Table 2. Private organizations providing usability training.

Government resources

We found usability information available at usability.gov and training from the US Federal Aviation Administration (Table 3). The British government offers the NHS Institute for Innovation and Improvement's website on implementing human factors in healthcare. The website also provides reports and training on situation awareness and other healthcare topics. The Agency for Health Care Research and Quality provides a range of materials including SAFER guides that, while not providing training in usability, have checklists for safety and usability.

Course	Summary	Source	Duration
GSA First Fridays Usability Testing Program	How to find and fix usability problems on government websites and applications	HowTo.gov	1 hour
Usability Testing: The First Fridays "Discount" Method	"Do-it-yourself" usability testing, plan for a test, recruit, test participants, develop test scenarios	HowTo.gov	1 hour
Usability Testing and Debriefing Best Practices	Conduct a simple usability test and debriefing session, identify the most serious usability problems participants, create a list of solutions that can be implemented in 30 days, tips for defriefing session success	HowTo.gov	3 hours
Designing a Better Customer Survey	How surveys can add value to your research, types of questions in surveys, best practices for writing survey questions, effective strategies for testing surveys	HowTo.gov	1 hour
Usability Testing	Basic review of usability testing, evaluating web sites, using inspection evaluation results cautiousl	usability.gov	1 hour
FAA Human Factors Awareness Web Course	Students should have an appreciation and awareness of the fundamentals of human factors, its methods, and the importance of having Human Factors specialists participate in FAA projects, course develop to introduce FAA personnel with backgrounds in various disciplines to the sciene and methods of Human Factors, course is tailored for Integrated Product Team (IPT) members to foster an understanding of the role and contribution of Human Factors in FAA system development	Federal Aviation Administraton	7 hours
Implementing Human Factors in Healthcare	Produced by the safety first campaign, situation, background, assessment recommendation is a structured method for communicating critical information that requires immediate attention and action contributing to effective escalation and increased patient safety	NHS Institute for Innovation and Improvement	NA
Human Factors: Training Competence	The ability to undertake responsibilities to perform activities to a recognized standard on a regular basis, linked to key responsibilities, establish and maintain competency for safety-related work	Health and Safety Executive UK	NA

Table 3. Government sponsored usability and human factors resources.

Massive Open Online Courses

We identified online education available through massively open online courses (MOOC) (Table 4). Udemy, Coursera, and the Health Informatics Forum all have variants of usability or user experience classes. Content includes user testing, prototyping, and usability evaluation. The Health Informatics Forum has a module on EHR usability within its usability and human factors online course.

Course	Summary	Organization
User Experience: The Ultimate Guide to Usability	• Section 1: Introduction • Section 2: Going where the action is: Understanding users in context • Section 3: how to get niche quick • Section 4: What can a London bus teach us about usability • Section 5: Beyond "easy to use": Measuring the user experience • Section 6: Site structure and navigation: Finding is the new doing • Section 7: Simple rules for designing simple pages • Section 8: "And I have the data to prove it": How to access a web site (usability evaluation) • Section 9: What next? Putting your knowledge into practice	Udemy
Human-Computer Interaction	• Lecture 1: Introduction • Lecture 2: Needfinding • Lecture 3: Rapid Prototyping • Lecture 4: Heuristic Evaluation • Lecture 5: Direct Manipulation and Representations • Lecture 6: Visual Design and Information Design • Lecture 7: Designing experiments	Coursera
Usability and Human Factors	• NA	Health Informatics Forum
Installation and Maintenance of Health IT Systems	• Unit 1: Elements of a Typical EHR System • Unit 2: System Selection – Software and Certification • Unit 3: System Selection – Functional and Technical Requirements • Unit 4: Structured Systems Analysis and Design • Unit 5: Software Development Life Cycle • Unit 6: System Security Procedures and Standards • Unit 7: System Interfaces and Integration • Unit 8: Troubleshooting; Maintenance and Upgrades; Interaction with Vendors, Developers, and Users • Unit 9: Creating Fault Tolerant Systems, Backups, and Decommissioning • Unit 10: Developing a Test Strategy and Test Plan • Unit 11: Pilot Testing and Full Scale Deployment	Health Informatics Forum
Health Management Information Systems	• Unit 1: What is Health Informatics? • Unit 2: Health Information Systems Overview • Unit 3: Electronic Health Records • Unit 4: Computerized Provider Order Entry • Unit 5: Clinical Decision Support Systems • Unit 6: Patient Monitoring Systems • Unit 7: Medical Imaging Systems • Unit 8: Consumer Health Informatics • Unit 9: Administrative, Billing, and Financial Systems	Health Informatics Forum

Table 4. MOOCs on usability and Human Factors.

SHARPC tutorials

We created our a series of short courses taught by NCCD faculty (https://sbmi.uth.edu/nccd/SED/tutorials/). Tutorials provide broad self-paced overviews to EHR usability, including principles of good design and methods of assessment. Topics areas are: Fundamentals of EHR usability, Safety-enhanced design, user-centered design, heuristic evaluation, user testing, other usability methods, usability tools such as TURF and references to other resources. Course materials consist of video lectures, hands-on assignments with rubrics, videos and photos of user testing setups, and access to an example poor EHR system (Figure 1). Participants are given opportunities to work through problems and can compare their responses with posted answers. Following videos describing heuristic evaluation, students are shown screenshots from the EHR test system. Students are then asked to locate and describe design violations following a set of heuristics. Assignments include calculating task completion time, noting path deviations, and analyzing responses to standardized measures such as the system usability scale. While each topic is stand alone, information contained in each series builds on previous modules.

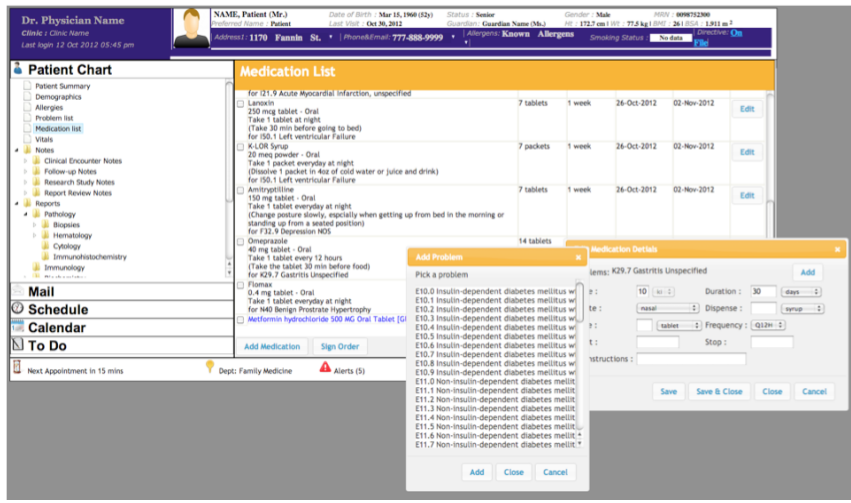


Figure 1. Demonstration of a poor EHR system.

Future Direction

We plan to expand our tutorials to encompass additional usability methods and exercises to practice usability testing of an interactive EHR system.

10: Usability Experience Survey

Yan Xiao, PhD
Baylor Scott & White Health

C. Adam Probst, PhD
Baylor Scott & White Health

ABSTRACT

We developed the Baylor EHR UX Survey, an instrument to comprehensively assess the user experience (UX) of an electronic health records (EHR) system. EHR UX is a function of many factors, including technical infrastructure, system integration, optimization, user training and support, and administrative policy. Ongoing UX assessment can help prioritize scarce improvement resources. Our instrument was piloted in six hospitals, then deployed in 11 acute care facilities. Survey responses were received from 1,301 nurses, 202 physicians, and 228 pharmacists. Combined with other user comments, survey results reinforced the multifaceted nature of EHR UX, revealed opportunities for healthcare organizations to improve UX, and highlighted EHR vendors' role in improvement efforts. All are now available for all organizations looking to improve their own EHR system.

INTRODUCTION

A majority of hospitals in the US have adopted electronic health records (EHR) [160]. Studies indicate significant EHR usability challenges (e.g., negative feedback from nurses, concern over patient safety and work efficiency [161], which pressure hospitals to improve user experience (UX). Unfortunately, improvement is often constrained by healthcare organization resources and limited ability to redesign EHR user interfaces. Usability data can help organizations focus on areas for improvement. Systematic EHR surveys can also complement other feedback, such as data from operations (e.g., downtime, screen update delays and user responses to clinical alerts), observation and time studies, suggestion boxes and unsolicited user comments.

UX is an essential concept in user-centered design, encompassing a user's perception of and response to using a product, system or service (ISO 9241-210 [151]. Surveys have been used to assess technology acceptance, functionality preferences and user satisfaction [162]. No survey has been published, however, assessing aspects of EHR UX over which hospitals have direct control, such as infrastructure, user support and training.

We developed a survey instrument that provides information about an EHR system as implemented within an organization. We also developed a process to use the survey data to guide continuous improvement and underpin collaborative opportunities with software vendors. Our instrument was successfully deployed in Baylor Scott & White Health (North Texas Division), a large integrated healthcare system, and piloted by other organizations [163]. Results have guided healthcare organization decisions on user training, functionality development, documentation policies, and infrastructure improvement.

Approach

We formed a multidisciplinary team of experts in patient safety, safety culture, human factors, survey design, statistics, nurse informatics, medical informatics, pharmacy informatics and hospital administration. The team reviewed the literature on user satisfaction surveys, user experience, EHRs and computerized clinical documentation. Five tools were identified, such as those by Edsall and colleagues [164] and Otieno and colleagues [165]. We also consulted professional organizations, including the Health Information and Management System Society (HIMSS) for unpublished surveys.

EHR UX should be viewed as a socio-technical system—socio, in that healthcare organizations have the power to directly influence facility aspects of EHR UX, technical in that users can only indirectly impact EHR software design. For example, installation of patient room computers used by nurses to document and administer medications may be installed in inaccessible locations (e.g., behind a couch) or with inadequate space (e.g., in a closet too narrow for using a mouse pad) (Figure 1). Facility redesign can alleviate these problems. EHR vendors' software design, however, determines intrinsic functionality and user

interface, which are technical aspects healthcare organizations have limited ability and resources to change (Figure 2).

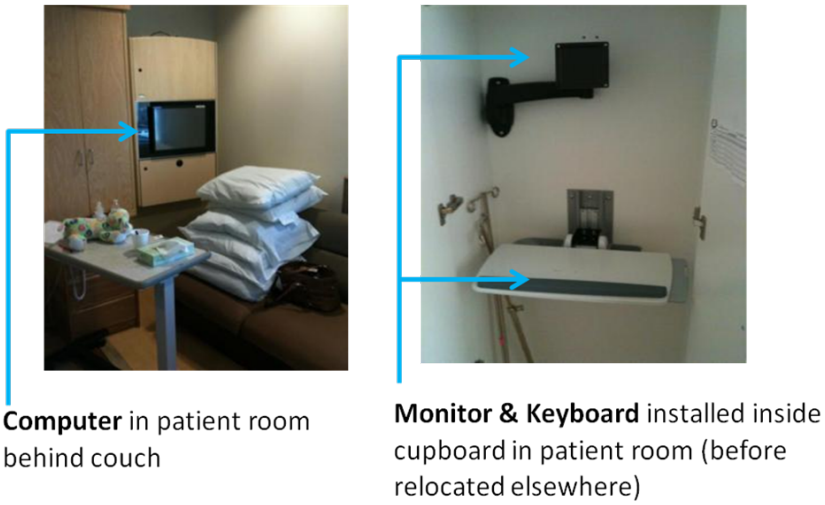


Figure 1. Left: computer in patient room behind couch. Right: Monitor and keyboard installed inside patient room closet.

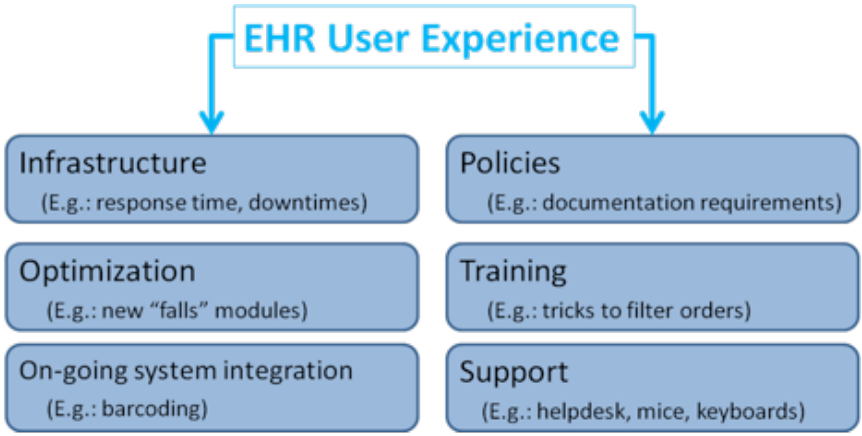


Figure 2.

Versions of the Baylor EHR UX Survey were tailored to nurses, physicians and pharmacists. The multidisciplinary team chose five survey domains to capture UX data:

- *Training and competency:* The degree to which users receive information (including help within the program) that prepare them to use their EHR as a novice and, with time, as an experienced user.
- *Usability:* The degree to which a user can find where to enter information into their EHR for documentation and retrieve data for clinical assessment. Usability also includes the degree to which users are able to easily navigate, view and edit medical information.
- *Usefulness:* The degree to which user tasks are enabled or prevented compared to electronic tools existing prior to EHR deployment or traditional paper-based environments.
- *Infrastructure:* The degree to which users can easily access their EHR, (i.e., sufficient locations where the EHR is available and fully functioning). Infrastructure also refers to the degree users are free from delays transitioning from one screen to another through normal software navigation.
- *End user support:* The degree to which users are provided solutions to problems encountered when their EHR system is not functioning as intended (slow performance, freezing, printing problems, hardware malfunction, etc). This is distinct from gaps in training that could improve users' ability to use the EHR more effectively.

Nursing and pharmacy versions of the survey instrument were deployed in six Baylor Scott & White Health hospitals in 2011 to assess internal consistency. In 2013, the instrument was deployed in 11 Baylor Scott & White Health hospitals for the three targeted user groups: nurses, physicians, and pharmacists.

RESULTS

In the 2013 survey, email invitations were sent to randomly selected EHR users at the 11 Baylor Scott & White Health facilities. The system automatically sent reminders and managed replies. A total of 1,301 nurses, 202 physicians and 228 pharmacists responded. The response rate for nurses ranged between 31% to 48% for individual facilities (overall rate 37%). Pharmacist response rate ranged from 9% to 76%. We were not able to assess physician response rates. Large numbers of free-text comments were also received. Nurses also provided 81 pages of

comments. We judged the survey a success based on responses compared with other surveys carried out in the organization.

Since nurses were surveyed in 2011 and 2013, it was possible to assess changes in user experience. The nursing version of the Baylor EHR User Experience Survey demonstrated good internal consistency (Cronbach's alpha=0.72-0.76) based on the 2011 survey from 606 nurses (response rate 36%) (Table 1). The instrument was published on the HIMSS website for non-profit usage [166]. Greatest gains were in usability, improved infrastructure, and enhanced support. Content layout and ability to correct errors did not improve.

Free-text comments highlighted specific areas for EHR improvement, including medication administration records and referral management. Text also included comments on changes made to the EHR and about learning to use the EHR.

Workgroups for the three targeted users (nurses, physicians, and pharmacists) were formed to study survey results and prioritize adding new or modifying existing functions. The survey, thus, became a platform for communicating healthcare organization EHR needs to their EHR vendor. Nursing policies were examined to understand discrepancies highlighted by survey results. User-centered design principles were introduced to assess readiness of new modules prior to going live.

DISCUSSION

Implementation of an EHR system is complex and challenging. Costs of initial acquisition, change management, user training, continual improvement and on-going maintenance are high [167]. It is important to describe how survey results will be used early in the process and to include all stakeholders early in the solution, including administration officials. We found involving executive leadership important to the success of our survey.

As much as vendors should be held responsible for delivering highly usable products, healthcare organizations hold many keys to a positive user experience. EHRs, unlike isolated software applications or consumer products, are part of a complex system of technology, professionals, policies and other interacting components. Refresher training, for

example, was identified in the survey as an important need for users who have initial experience and would like to learn techniques that can improve their productivity. Hardware maintenance is a constant challenge given the number of devices deployed among dispersed geographical areas in a typical hospital. Documentation may be challenged by user interfaces, unclear expectations, variance in practices, and suboptimal access to computers. A survey is an excellent tool to systematically sample a large number of users, especially at multi-hospital healthcare organizations. Taking a socio-technical perspective allows assessing user experience from multiple perspectives (user and vendor), building a foundation upon which improvements can be made.

Among all tools, EHRs consume the most clinicians' time [168]. Measuring UX and improving UX should be a top priority in all healthcare organizations. As more organizations measure their own user experience, it will be instructive to compare results. Vendors should be invited to analyze survey results and work closely prioritizing their future software improvement efforts.

Suggested Reading

Staggers, N., Rodney, M., Alafaireet, P., Backman, C., Bochinski, J., Schumacher, R. M., & Xiao, Y. (2011). Promoting Usability in Health Organizations: Initial Steps and Progress Toward a Healthcare Usability Maturity Model. https://www.himss.org/files/HIMSSorg/content/files/HIMSS_Promoting_Usability_in_Health_Org.pdf

Acknowledgement

The authors writing for the Baylor EHR User Experience Survey team included Donna Montgomery, BSN, MBA, RN-BC; Lindsey M. Philpot, MPH; Sunni A. Barnes, PhD; Jan Compton, BSN, MSHA, RN; Donald Kennerly, MD, PhD; Jeff Kerr, MD; Joseph Schneider, MD; Lynn Finck, RN; David Watson, BS, RPh. The authors thank contributions from Drs. Linda Harrington, Jiajie Zhang, and Muhammad Walji, as well as nurses, physicians, and pharmacists. Results of the nursing version of the survey originally appeared in the Journal of Nursing Administration.

EHR Design

A significant barrier to usability is an EHR interface that does not meet the unique needs of its healthcare users. Synchronizing the flow of information with clinical workflow is key. SHARPC developed Modeling & Analysis Tools for Healthcare, called MATH, that capture existing information and workflows and generate simulations of improvements (Chapter 11).

Chapter 12 reports the compilation of 300 general-purpose design principles directly applicable for EHR systems. Safety Enhanced Design Briefs (Chapter 13) are single-page design principles covering user interface issues, such as effective use of color and electronic prescribing. Detailed design guidelines can be found in Chapter 14: Inspired EHR - Designing for Clinicians, co-funded between SHARPC and the California Healthcare Foundation.

SHARPC researchers translated theory into practice in TwinList, a novel means of medication reconciliation EHR (Chapter 15). Chapter 16 tackles medical order management user interface challenges and proposes tabular display design guidelines that can enhance patient safety.

11: Advances in Workflow Modeling for Health IT

Modeling & analysis toolsuite for healthcare (MATH)

Keith Butler, PhD
University of Washington

Ali Bahrami, PhD
Medico Systems

Konrad Schroder
University of Washington

Melissa Braxton
University of Washington

Lawrence Lyon, MD
VA Puget Sound Healthcare Systems

Mark Haselkorn, PhD
University of Washington

ABSTRACT

Synchronizing the flow of health information and the workflow of clinical care is a key principle for successful health information technology (HIT) systems. When the flow of information matches better workflow, significant gains in quality and efficiency can be achieved. When information flow contradicts, it can rearrange clinical workflow by accident rather than by design. We introduce the Modeling and Analysis Toolsuite for Healthcare (MATH), a method with powerful modeling and analysis tools to make measurable improvements to clinical workflow a predictable, integral part of HIT systems. We illustrate how MATH can analyze how HIT should be applied by designing HIT functionality on the basis of evidence of beneficial impact. We demonstrate the feasibility of conducting formative evaluations on workflow models to predict HIT impact and present validation data on predictions from summative tests conducted on an alpha version of a new system. These capabilities allow

a new evidence-based approach to HIT in which healthcare leaders decide and plan the appropriate role of computing for their clinics.

INTRODUCTION

One of the most important goals of health information technology (HIT) is to improve the way clinicians can perform healthcare [169, 170]. Based on recent analyses, however, the goal is not being achieved [170, 171]. Two of the most important challenges to HIT are problems with workflow and usability [170, 172].

Synchronizing information flow and workflow is a key principle for HIT effectiveness, quality and usability [7]. HIT users are faced with a dilemma when information flow does not support an appropriate workflow of care: either compensate and perform unplanned overhead tasks by modifying their information environment or follow a sub-optimal workflow to conform to the way their HIT applications provide information [79]. Unplanned overhead is more than just extra work that interrupts clinical care; it can disrupt users' cognition, inhibit coordination among team members, and even obscure their understanding of tasks [37, 39]. It is a form of usability problem that places ease-of-use in opposition to patient safety [173, 174]. A well-designed HIT application with good usability will make the routine performance of safe, efficient and effective care procedures the easiest course of action.

When HIT design decisions are not directly related to the values of better care, the design can be dominated by issues of technical feasibility, schedule or cost. The resulting applications can have the unfortunate effect of rearranging clinical workflow by accident rather than by design [78, 175]. Conversely, understandable evidence about HIT's measurable benefits to care will result in increased adoption and productive.

To close the gap between the flow of health information and the workflow of clinical care we developed a comprehensive new design method called MATH (Modeling & Analysis Tools for Healthcare) to integrate three fundamental elements that are currently disjoint: workflow models of how clinical care is actually performed and the areas where improvement is needed, options for how HIT should improve

workflow in measurable ways, and software specifications for rapid implementation.

MATH is supported by a suite of tools.

- *MATHflow* for capturing, analyzing and integrating workflow and information flow,
- An information dictionary created while modeling workflow with *MATHflow,* and
- *MATHsim* for discrete-event simulations and formative evaluation of HIT options.

MATH is an evidence-based approach that makes measurable, predictable workflow improvement integral to HIT design.

Cognitive science principles

The term "information system" is something of a misnomer because work is executed not only by computers, but also by the cognitive and manual procedures of human users. Further, HIT applications typically play a support role as they are used in a clinical workflow with many important manual tasks of care and administration.

The integration of manual tasks and computer-performed tasks is critically important for HIT effectiveness and usability [175]. Information resources, like many other types of resources, constrain the way clinicians can use them to perform care. Research from cognitive science [37, 39] and software design [27, 78, 176] consistently demonstrates that content, organization and representation of information inherently impose powerful constraints on the way users are able to perform their tasks. The constraints are widespread and powerful. They affect users' procedures and even their cognitive strategies for performing tasks [39]. Cain and Haque described how HIT implicitly imposes workflows upon nursing [177]. White and Miers argue that the software of information systems embody a model of workflow, whether or not that workflow was understood and planned [178]. Further, unplanned tasks may be added to the workflow in an accidental manner when users have to deal with design-induced errors [173, 179] caused when properties of information do not match the needs of a task.

HIT developers may be reluctant to accept responsibility for constraining the way clinicians work with their applications, but taking a neutral design stance is mistaken. Information resources, like many other types of resource, constrain the way they can be used to perform work. Constraints on information-dependent work are inherent to complex HIT applications. The only question is whether the workflow impact of an HIT application will be understood and planned as part of systems design, or that its impact will be accidental and discovered after deployment.

APPROACH

MATH capitalizes on two established software standards: the Business Process Modeling Notation (BPMN) standard for modeling workflow [178], and the class and state diagrams of the Unified Modeling Language (UML) to model HIT information architecture [180, 181]. Originally popularized for web design, information architecture is a powerful, non-visible dimension of usability for both mobile and client platforms. Information architecture is, therefore, a key part of software design, defining an application's body of content, and how it is organized for end-users [182].

The traditional way to reduce undesirable HIT impact on workflow was to make conservative, incremental improvements to existing information systems. This cautious approach, however, can easily fail to exploit the full potential of HIT or achieve benefits to justify significant costs. Our use of standards develops well-defined models of workflow and of information architecture. As the paired models are being developed we can understand the implications of design decisions for one on the other, so their respective designs can converge. MATH's high-level design goal is for workflow and HIT to function as a pair of well-matched, complementary components to improve the performance of care in a predictable manner.

MATHflow

The core capability is MATHflow, a visual diagramming tool that captures the existing workflow and information resources in such a manner to reveal how care should be improved with HIT. There is growing interest in workflow models as a tool to analyze and design HIT

[183-188]. Carayon, et al., argued that HIT necessarily involves workflow, but little is known about how to integrate the two [184]. MATHflow allows us to analyze how an HIT application will impact workflow by replacing, augmenting, or complementing important manual activities of care, then evaluating the resulting model. MATHflow models can also explicitly represent the overhead tasks of using HIT so we can identify and minimize them.

MATHflow is independent of any specific EHR. It is based on the Object Management Group's recent standard for Business Process Modeling Notation [178]. The standard for BPMN has been widely adopted for software requirements to support manual work by teams of people. Our use of standards, therefor enables interoperability with other tools, such as workflow engines, external simulation and analytic tools.

Innovation for information modeling

MATHflow's integration of workflow modeling, physical resources, and the use of information resources is different than of workflow diagrams. One of MATHflow's major innovations is to increase the expressiveness of BPMN 2.0 by implementing a small set of modeling constructs that increase modeling power while simplifying the models. The extensions provide more flexibility and yet easier modeling of information resources, including HIT systems. BPMN comes with three types of Information Artifacts: Document, Annotation and Group. MATHflow has extended the set with three types of Information Resources:

1. *Person*, such as a patient or a colleague providing information,
2. *Information System*, such as an EHR or other HIT application, and
3. *Part*, such as equipment that provides EKG, blood pressure, etc.

This extended set reflects the richness of information resource types commonly used in a clinical environment that must be explicitly documented in order to understand the complete flow of important information. MATHflow distinguishes the access properties of different types of information resources. For example, the contents of a Document can only be used by clinicians in the immediate vicinity. In contrast, the

contents of an Information System can be shared electronically by a clinician in any location with a suitable device and access privileges.

MATHflow builds an information dictionary that models information resources as objects and their contents as attributes, creating a record of the information requirements of a workflow. The tasks in MATHflow have an editor for entering the information requirements in terms of the information attribute that is needed and the resource that provides it, which are then recorded automatically in the dictionary.

Another innovative extension of BPMN are complex decision gates. MATHflow's information dictionary tracks the relationships among tasks, information resources, the information attributes they contain, and any specific values for attributes. These data structures allow values of attributes that are set in one part of a workflow to be used in the decision logic of gates in other parts of a workflow. Complex decision gates allow much more realistic and complex information flows to be modeled than conventional workflow tools.

MATHflow can also model relationships between information resources. MATHflow has integrated UML class diagrams with BPMN in order to model user definable information resource artifacts and their relationships. In essence, researchers can model any HIT system to analyze how it will impact workflow as an information system.

Innovation for formative evaluation and analysis

MATHflow's new capabilities to integrate information modeling with workflow allow the MATH method to make trade-off analyses between better information resources for less need of physical resources, such as the labor of highly skilled clinicians. These analyses provide valuable, formative evidence about the HIT impact during the design stage of a project to guide decision making about HIT functionality.

The MATH method

MATH documents how care is currently performed with existing information resources, identifies problems and how they can be mitigated or eliminated, with a focus on better designed HIT (Figure 1). The first step of MATH is similar to popular process improvement methods [189], but the rest of MATH is a variant of "concurrent engineering" in which

multidisciplinary teams collaborate on a common design objective [190]. MATH iteratively combines "patient-centered design" with conventional "technology-centered design." The objective is a pair of matched designs that work smoothly together as: 1) a measurably better workflow of care, and 2) a cost-effective, highly usable HIT application whose information flow maps to the needs of the better workflow.

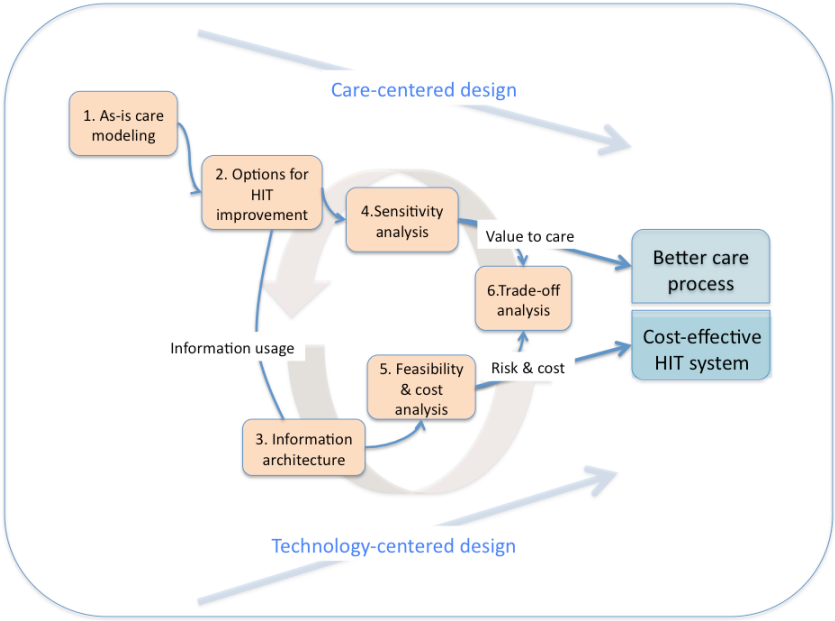


Figure 1. The MATH method.

Step 1 is an observational study that applies ethnographic methods to discover, model and understand how clinical care is currently performed using existing information resources. This produces the dictionary of the information required to support it and identifies awkward workflows and problems.

Step 2 is an analysis of HIT options to address problem areas, such as information organization or sequence that does not match workflow, presentation format does not match tasks, information degraded because diverse resources must be integrated manually, or excessive user attention to control existing HIT functionality.

Steps 3 through 6 progress through several iterations. In Step 3 we analyze the information architecture for each option. The information

architecture defines the content and organization of a body of information needed to support the workflow. MATH analyzes the information architecture for each option in Step 3. It can be readily translated into key software specifications using UML [180, 181], establishing a connection between the workflow benefits of HIT and its technical feasibility and cost.

Step 4 uses formative evaluation to compare at least two options: "as-is" and a proposed "to-be." A key principle of evidence-based HIT is that computing functions should be prioritized on the basis of their impact on better care. The evidence of improvement should be the benefits to workflow, such as gains in the quality or efficiency of care. In our experience, user participation in this step can quickly generate ideas for several to-be options. We organize these into several coherent options from Steps 2-3 and estimate the dimensions and magnitude of improvement for each. The results provide valuable formative evidence about beneficial impact on workflows of clinical care.

Step 5 considers each HIT option's technical difficulty, time to availability and cost. These are the typical decision making factors in conventional design, but they should be weighed against their impact on the efficiency and quality of care to select the best option.

Step 6 transitions responsibility for prioritizing the options to the clinic's leadership, stakeholders, and information technology staff. We facilitate their analysis to rank order the options based on their values for patient care. The analysis weighs the trade-offs of three factors for each option: 1) the value of care improvement, 2) the technical risk, and 3) a cost estimate for the project to acquire or implement each option.

MATH enables researchers working with care stakeholders to capture workflow improvements and connect them to HIT design. This connection enables design trade-offs between the added value of an HIT system in terms of impact on the quality or efficiency of clinical workflow, and HIT technical implementation factors, such as feasibility, risk, cost and schedule. MATH addresses three of the Institute of Medicine's dimensions of quality: patient-centered, efficiency, and timeliness [3]. MATH closes the gap in conventional methods by integrating the capture of how clinical care is actually performed, the options for how it could be improved in measurable ways if supported by

better HIT applications, and the algorithms, data, and user interface concepts of those applications. A benefit of bridging the gap between workflow and the design and HIT is reduced cost and time for software development.

MATH provides a methodical and understandable means for stakeholders to direct strategic, cost-effective workflow improvements [27, 78]. A feasibility study illustrated how increased insight into the workflow and information flow of a large primary care center resulted in a predictable and measurably better workflow.

Feasibility demonstration

We applied the MATH method, tools and techniques in a study at a primary care clinic in the Puget Sound region. In collaboration with VHA Medical Informatics and clinicians, our team of analysts followed the MATH method to:

- Develop an as-is workflow model of how care was practiced using existing information resources,
- Analyze how workflow could be improved with better HIT,
- Perform formative evaluations to predict the HIT impact on a to-be workflow,
- Build and test an alpha version of HIT software and compared results with MATH predictions.

Modeling and analysis took about sixteen hours of semi-structured interviews and observations with providers and nurses. Thirty hours were spent for model-building, analysis, and design. The design was then implemented in an alpha version of software, tested in a summative evaluation that took another 22 hours of subject interviews, and 20 hours of software testing by subjects.

Evidence-based HIT requires understanding the way clinicians currently perform care so that improvements can be identified, prioritized, and incorporated into design of a better system. Figure 2 shows the top-level MATHflow model of current practice for patient visits to a primary care clinic in the Puget Sound region. Patients arrive at the clinic in the upper left corner of the diagram. Patients follow one of

the optional paths defined by decision gates (diamonds) based on probabilities or logic rules.

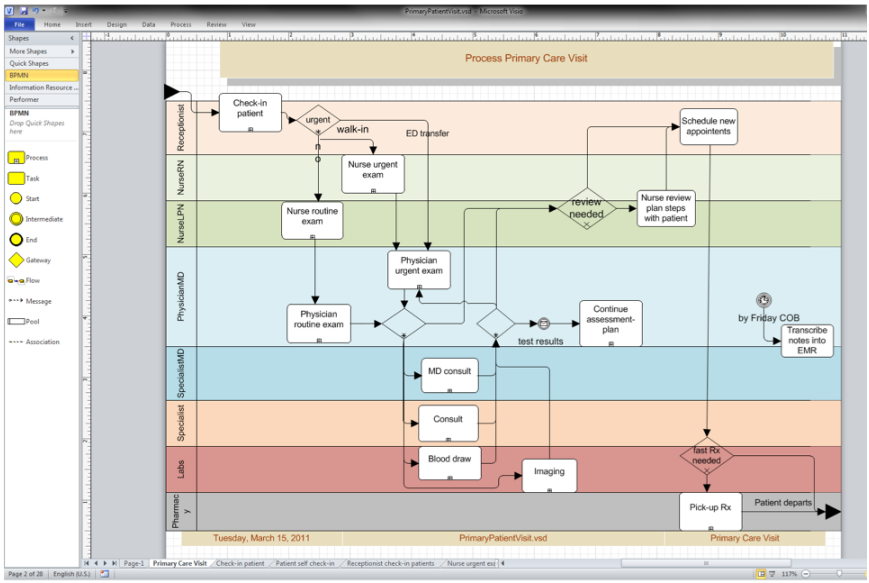


Figure 2. High-level primary care workflow.

MATHflow uses the colored rows as "swim-lanes" to organize the visual layout of various job types in the clinic. Figure 2 shows one task activity in the Receptionist lane *Schedule new appointments* (top right) and one sub-process for *Check-in patient* (top left). Sub-processes have a small cross at the bottom-middle, indicating they contain a lower level flow made up of tasks or sub-processes. For example, in the swim-lane for PhysicianMD there is a sub-process for *Continue assessment-plan*, which waits at the message symbol until test results arrive.

The *Continue assessment-plan* sub-process has been opened in Figure 3 to display more detail of the workflow it contains. It has several tasks, and also several more sub-processes, which in turn can be opened for more detail. Workflows reflect many factors, including the type of patients and the care they need, the nature of the work entity that moves through a workflow, the personnel and organization of the clinic, facilities and equipment, regulations, and clinic policies. Managing detail with hierarchies of sub-processes is one feature that allows MATHflow to represent large, complex workflows without displaying an overwhelming amount of detail.

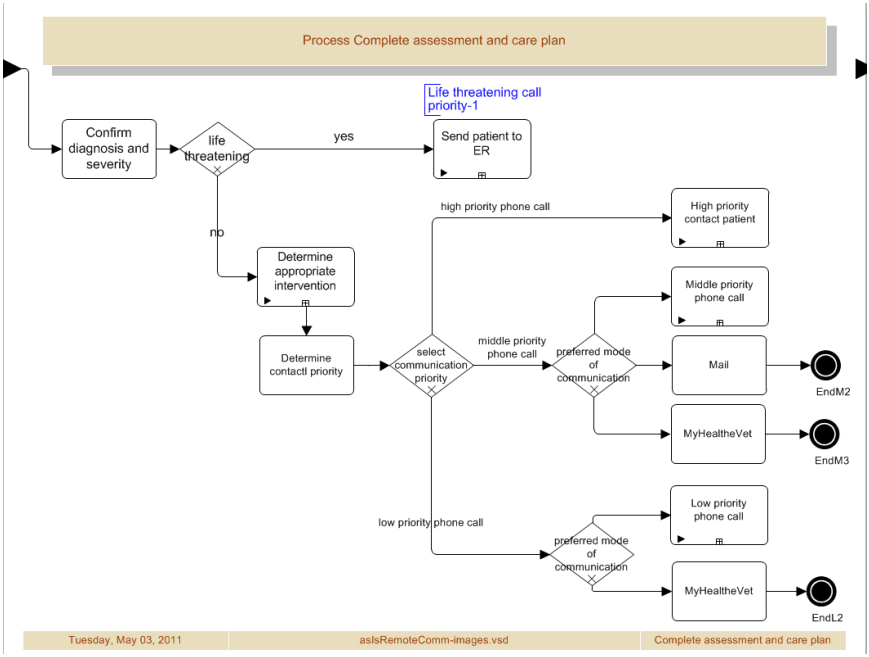


Figure 3: Detail for the sub-process Continue assessment plan.

Figure 3 shows a sub-process of a provider receiving the test results and beginning the tasks of the sub-process by confirming the diagnosis and severity. This workflow reflects clinic policy for contacting patients depending on the severity. The *life threatening* decision gate represents a rare, but important possibility that the diagnosis requires the provider to contact patients immediately to get them to nearest emergency room. Otherwise, the flow continues to *Determine appropriate intervention*, where treatment is planned or further tests ordered. VHA policy requires that patients be informed of all test results. The workflow in the right half of Figure 3 reflects three non-critical priorities and the permissible means within each level for contacting patients. The workflow was carried out manually at the cost of about 40 hours a month per provider. According to interviews only about 33% of phone calls actually reach the patient directly. Churn and phone-tag identified this workflow as a candidate for HIT improvement.

Scoping sub-processes

A new sub-process is needed when the entity of clinical work changes. For example, in Figure 2 at patient arrival the work entity is a

patient's visit registration. Other sub-processes in Figure 2 have entities for patient exams, treatment plans, and lab tests. These are distinctly different and the scope of each sub-process must account for the transformation of its work entity to its goal state. For example, the sub-process for a lab must account for how the entity of a blood draw is transformed to the goal state, which is a lab test report. Importantly, MATH can also model and analyze HIT support for work entities that are conceptual, such a diagnosis, or a treatment plan [29].

A workflow is also constrained by resources, including information resources. The initial as-is MATHflow model of the clinic captured how care was actually performed and how that was constrained by the context and availability of resources, including physical resources and information resources, as shown in Figure 4. Our focus is to understand current workflow to determine how HIT should improve care. But we are primarily focused on modeling workflow at the level of detail that reveals how information is accessed, used, changed and recorded.

Capturing information flow

The flow of information is not identical to the workflow of clinical care. Patients, tests, diagnoses, treatment plans, etc. are some of the entities that flow through clinical care. In contrast, information about them flows in and out of care activities, which can change it.

Most diagramming tools for software are aimed at creating elegant design solutions, as opposed to the complex, and often informal way that healthcare is actually performed. MATHflow represents information that is distinct from, but related to clinical workflow, providing the flexibility needed to capture the way care is actually performed with information resources. All the information resources that are used in a clinic must be documented, whether they are physical or electronic. In a clinical environment the information resources may include media that are paper, digital, mechanical equipment or analog instruments. Information resources also include people, adding complexity that may overwhelm conventional modeling languages. Doctors and staff may play the multiple roles of labor resource or information resource. Further adding to the complexity, the patient may have multiple roles: as an entity of care, as an actor for self care; or as an information resource.

The complexity of clinical care requires a representation for information that is distinct from, but related to clinical workflow. The information modeling capability in MATHflow allows it to represent large, complex workflows without displaying an overwhelming amount of detail. By treating information as a resource (instead of a task) the models are visually simpler, while capturing complexities of the variety of important information resources and the interaction between computing functions and manually performed functions.

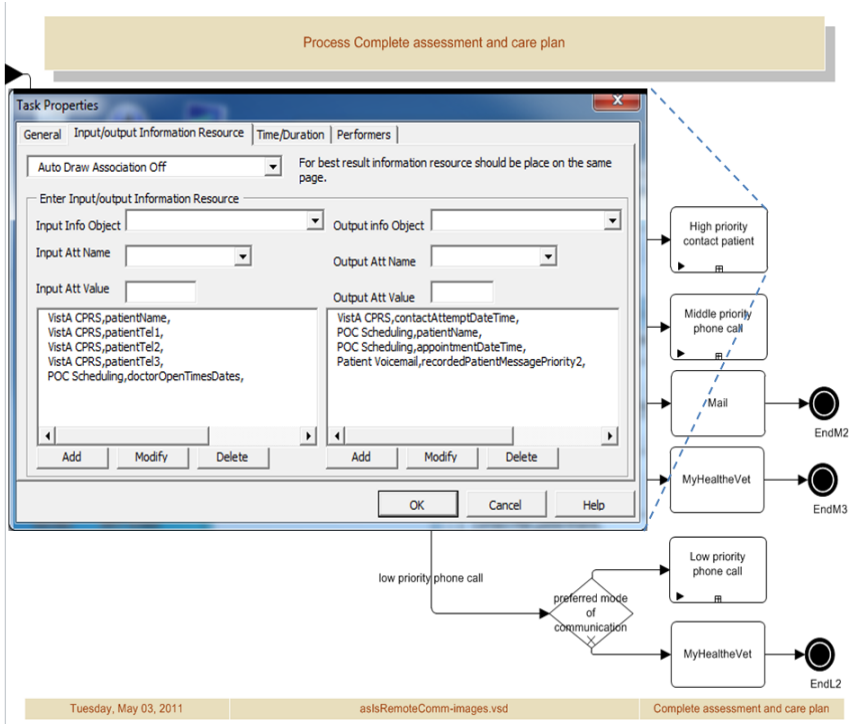


Figure 4. Information Use Editor.

Figure 4 shows how information used for *High priority contact patient* is modeled. The *Task Properties* editor tab has been opened for Input/Output Information Resource. On the left is information needed to contact patients, entered in terms of the attribute name and the resource object where it was accessed. The right panel shows outgoing information attributes for logging the call into the computerized patient record system (CPRS), leaving a priority-2 message on the patient's voicemail, or scheduling an appointment. The editor captures information use from all media.

There are several important reasons to capture all the types of information resources that are used in a clinical workflow, whether digital or not: requirements for an HIT application must reflect how it fits into this complex information environment; some physical information resources may be good candidates for an improved HIT system; there may be overlapping records in different media that need better configuration management, and; integration of physical and digital resources can be a source of inefficiencies that should be corrected in a new system.

Until recently workflow models captured information requirements in an ad-hoc manner. For very small incremental effort, however, MATHflow captures the use and flow of information in a manner that integrates it with activities and their flow. Through the use of the information property editor, as the analyst builds the workflow model the editor also captures a dictionary of all the information attributes that the workflow tasks need (and only the information that they need) and their flow as well. MATHflow automatically builds the dictionary from the entries in the properties editor. Table 1 shows the concept of the information dictionary.

	Information				
Workflow	1	2	3	. . .	n
A	0	1	1	0	1
B	1	0	0	1	0
C	1	0	0	1	0
. . .	0	1	1	0	1
Z	1	1	1	0	0

Table 1: MATHflow information dictionary.

The information dictionary lists each task in the workflow down the left column and the information attributes across the top. The resulting matrix captures information usage patterns needed by the tasks of a workflow. For example, task A does not need attribute 1, but it does need attributes 2 and 3, and so on.

The information dictionary offers a unique way to manage the complexity of models. MATH's implementation of complex gateways

differs from the BPMN standard, since the standard does not share MATH's concept of information resources. MATHflow has complex decision gateways that can check the value of an information variable in the dictionary, and then select which outgoing flow to take based on its value. This allows activity in one sub-process to determine the behavior in another without drawing long flows across pages. The dictionary, thus, becomes the connection between HIT improvements and care workflow.

MATH allows us to analyze how an as-is model of workflow can be improved by changing the resources that provide information. This innovation has the added benefit of representing both manual tasks and HIT tasks in the same notation, which is key for integrating HIT into workflow improvements. Rather than focusing on the features of HIT it is treated as an information resource that supports or performs tasks.

Analyzing options for improvement

Our study of the clinic identified a significant problem area and lead to the design of measurable improvement. Using MATH led to important understanding of the demographics and the context in which care takes place. For example, the VA offers web-based MyHealthEvet for access to veteran health benefits and services, but clinic personnel estimated that only about 10% of their patients use it. In addition, the patients of this clinic typically have multiple comorbidities and complex treatment plans. Consequently, providers wanted to talk with patients to check how well they could carry out new orders. Recent surveys show a strong preference for real-time phone conversations with providers when new orders are issued [191, 192].

The new workflow enabled by the new HIT application is shown in Figure 5. The unproductive activity (right half of Figure 3) has been replaced with an software product named *Priority Contact*, which won an award in the national challenge competition SMART Apps for Health [193]. *Priority Contact* was designed to interact with EHRs by reading and writing data via the new SMART Connect interoperability standard, while running on its own, separate web server to allow maximum functional flexibility.

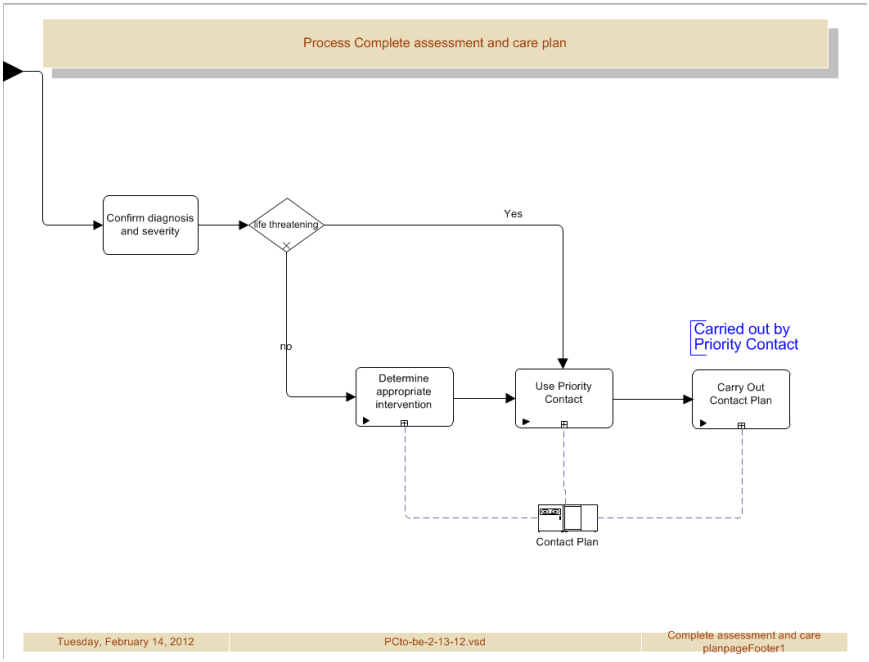


Figure 5. More efficient workflow with a new HIT resource.

MATH trades better information resources for less use of physical resources, such as the work time of highly skilled clinicians. In exchange for the large reduction in effort clinicians set up and monitor a contact plan for each patient. *Priority Contact* integrates this information with patient activity and the remaining manual clinician activity.

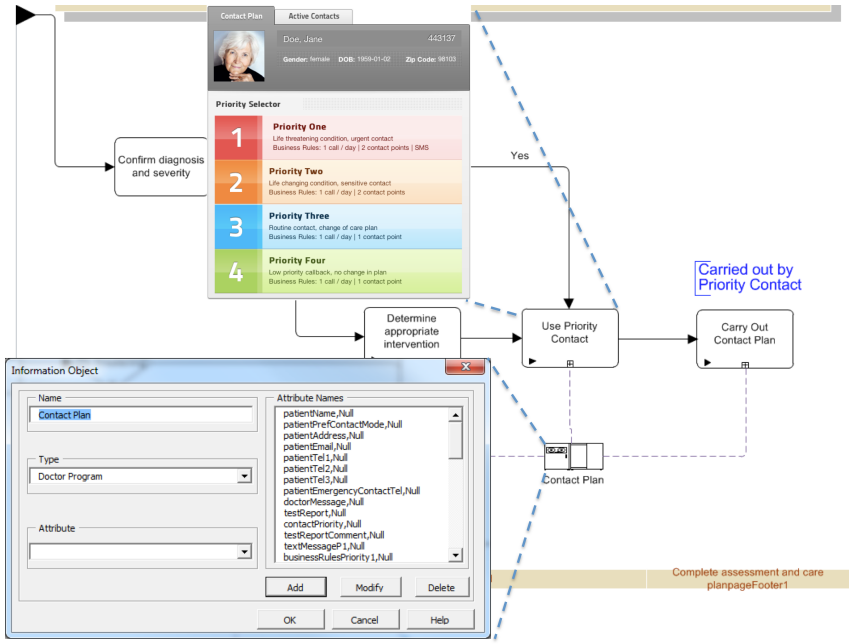


Figure 6. User interface and contents of the contact plan.

The contact plan waits in the background until the doctor enters a patient's identity and contact priority via the user interface shown in Figure 6. Another benefit of design integration is that the user interface is simplified when it is based on the desired workflow and information flow. To start the new patient contact workflow a clinician reviews, edits, and launches the plan, which reads the EHR for the remaining information it needs and then provides the data needed to carry out the algorithm shown in Figure 7.

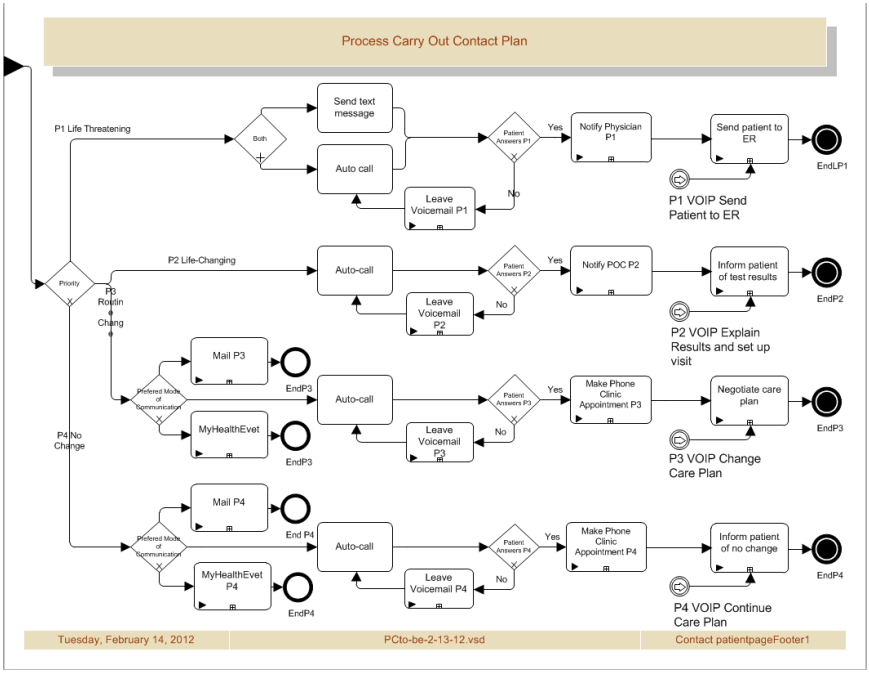


Figure 7: Algorithms derived from workflows.

Figure 7 shows how the as-is workflow of manual activities was translated into an algorithm that is now carried out by the patient, clinicians, and the software agent. The algorithm was derived from the as-is workflow. They accomplish the same purpose but far more efficiently by integrating HIT functions and automation with manual tasks. *Priority Contact* does not replace important real-time conversations between clinicians and patients (represented by the four sub-processes on the far right). It enables them to happen more quickly, with less frustration and wasted effort. The system can handle urgent, Priority 1 contacts by calling multiple phones simultaneously and repeatedly until the physician is notified that one of the phones answers or calls back.

Principles of formative evaluation

Developers need guidance during the design stage of HIT for their project to reliably benefit clinical care. The importance of evaluating HIT applications for their impact is well established [194, 195], but conventional methods require some version of the software to evaluate impact. This has traditionally meant conducting evaluations after the major design decisions of an application were already decided, so the

results had to wait until the next version of the software. Little could be done to improve any fundamental design problems during in the intervening months or years while clinicians dealt with unplanned overhead and risk of errors. Evaluation needs to provide feedback before a project has reached the point where it is too expensive or difficult to make major changes.

The growing use of usability evaluation like TURF (Chapter 6) is an example of how formative evaluation can provide timely feedback to improve user interfaces [7]. MATH's focus on achieving synchrony of workflow and information flow complements TURF's deep focus on individual user interface designs. By moving the evaluation of workflow impact much earlier in the HIT life-cycle MATH can "build-in" the benefit of HIT through model-based design iterations.

Formative evaluation for quality

Clinicians were able to understand and critique MATHflow diagrams with minimal training. They performed exhaustive manual model checking to evaluate the quality of to-be workflows were conductive using the cognitive walk-through technique [36]. Clinicians and developers worked in joint sessions to review every path and drill-down into sub-processes for correct conformance with appropriate care, efficiency, and feasibility. As shown earlier in Figure 6, user interface images could be linked to tasks in MATHflow to make the evaluation more tangible.

An important patient safety criterion was a workflow that allowed clinicians to maintain positive situation awareness until the need to contact the patient was resolved, e.g., "What should happen if the patient does not have voicemail, or nobody checks it?" When either a provider or a developer recognized a problem they negotiated by suggesting other options, then checking if it worked from both perspectives.

Another key part of the walk-through was the flow of information, as depicted in Figure 8. The integration of human tasks and those performed by *Priority Contact* was mediated by the common use of information objects: i.e., both HIT and the human users operate on the same information objects, but with a more appropriate allocation of

responsibility. Figure 8 illustrates how MATHflow makes the information requirements explicit for review by both clinicians and developers.

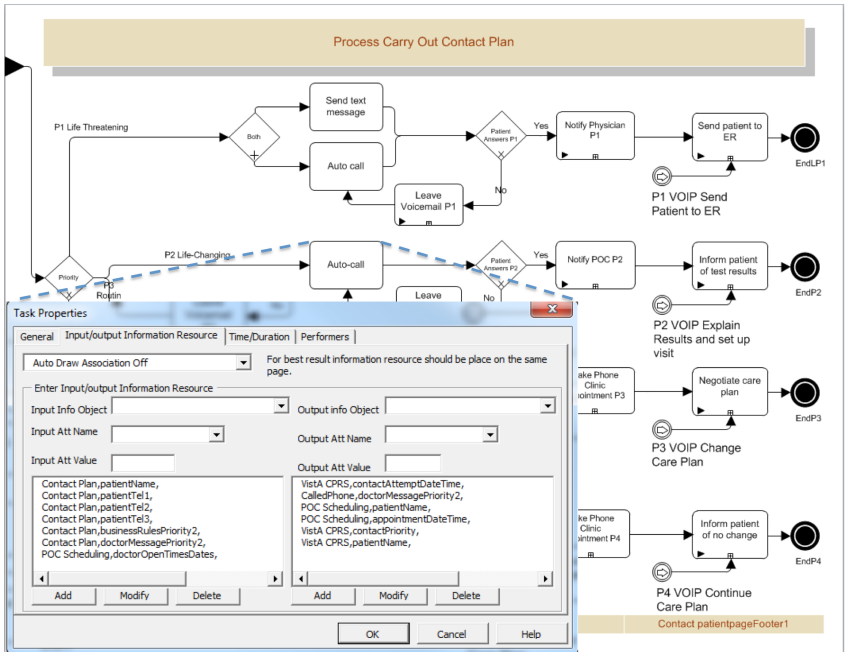


Figure 8. Information used by tasks in the to-be workflow with Priority Contact.

Formative evaluation for efficiency

The property sheets of tasks include tabs for Time/Duration and for the Performers of a task. They allow MATHsim to analyze an important trade-off: less use of labor resources for access to better information resources.

MATHsim is a discrete-event simulation engine [37] that is integrated with MATHflow. It reads models from a MATHflow database, and performs Monte Carlo simulation to measure the performance of the models under user-supplied workloads, allowing multiple models to be compared against one another. The results are quantitative distributions of task times and resource usage. MATHsim first provided a baseline for the as-is model, then evaluated the impact of different options for *Priority Contact*.

MATHSim runs several independent trials, each of which contains process instances that may interfere with another's operation. Monte

Carlo uses a pseudo-random number generator. Process state is maintained by a discrete event queue.

MATHSim runs several independent trials, each of which contains several process instances that may interfere with one another's operation. MATHsim uses a strong pseudo-random number generator, and the process state is maintained by a discrete event queue [196].

MATHsim executes a computer-based simulation by generating all possible workflow and information resource combinations to estimate important performance statistics, such as the distribution for how long a workflow will take or how much time it will require from a given type of resource, such as a doctor. One of the key innovations of MATHsim is that makes appropriate distinctions between the constraints for using digital information resources vs. physical information resources.

Figure 9 shows MATHsim's formative evaluation that compares the number of clinician hours required for the workflow to contact 100 patients about test results with thirty replications to approximate a normal distribution around the modes.

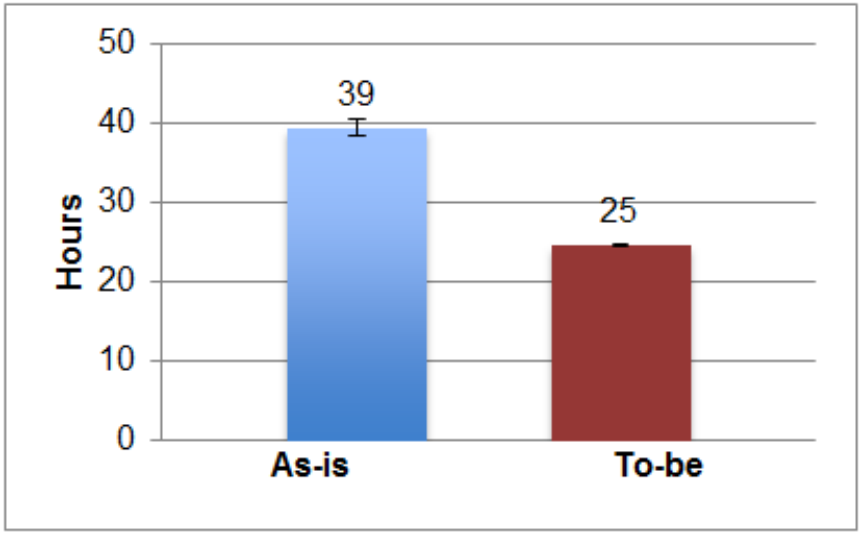


Figure 9. Comparison of monthly hours spent contacting patients.

The comparison of as-is and to-be in Figure 9 was evidence of a promising advantage of about 26% time savings for a workflow using

Priority Contact. The results were also consistent with available clinic historical data and also with test data from user interface prototypes.

RESULTS

Based on the formative evaluation we developed the software for an alpha test of *Priority Contact*. The to-be model and its information dictionary provided detailed specifications of how the application should work in the context of the workflow. The alpha version was web-based and used web-voice services to call and send text messages to mobile phones belonging to members of the research team (playing the role of patients); a set of 20 test lab results designed to represent four levels of patient contact priority (Table 3) from urgent (Priority 1) to routine follow-up (Priority 4). The fictitious test patients, (created by SME; validated by the principal investigator) contained (1) minimal clinical history (name, age, gender, presence of diabetes), whether the test was for a new or existing problem and (2) lab values that represented an increase, decrease, or new change from previous value. The percentage of cases in each priority level were weighted to represent an average week.

Test data were recorded on a convenience sample of ten clinicians from a variety of roles. Participants were not compensated but were incentivized by the opportunity to influence the design of their future work systems. The study setting was an outpatient primary care clinic at a VA healthcare facility in the Puget Sound area. Participants tested the software on their desktop PCs at their normal places of work.

Each evaluation trial included a pre-intervention interview, summative user testing by clinicians in their own work settings, followed by a post-test interview, and an optional observation session.

The test users performed patient contact tasks with the prototype at their normal workstations while test administrators recorded the same variables as in the formative evaluation. They tested the application using the internet browsers on their workstation computers using simulated patient records. No patients were contacted and no identifiable personal health information were used. Because the alpha version of the prototype is not linked to the EHR, study subjects were delivered realistic, mock alerts regarding test results via email (1-5 alert emails per day at various times of day throughout the 7-14 day study period. Total = 20 cases).

Test administrators recorded the data that included the same variables as in the formative evaluation for efficiency. The revised models based on the summative evaluation results for contacting 100 patients are shown in Figure 10.

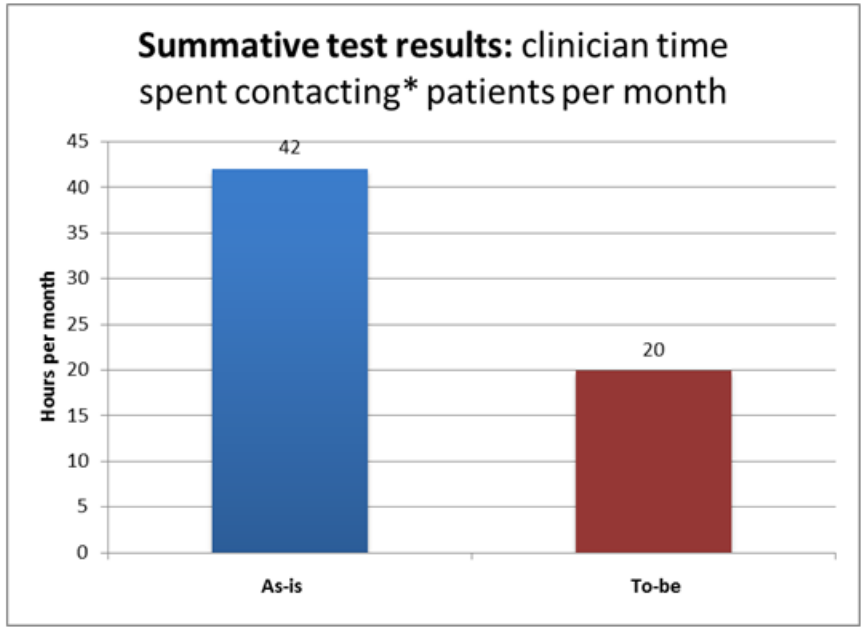


Figure 10. Alpha test results.

Figure 10 shows the revised models based on the results for the summative test results on the work time for the sample of medical professional users. It produced a highly similar pattern of results to those predicted by the model-based, formative evaluation. Both were highly significant, but the formative prediction was more conservative then the benefit measured empirically in the summative test. The results of the summative evaluation showed a larger 53% savings for the quantitative impact on workflow, but also revealed policy changes that would be required to realize the benefits of the new system.

Discussion

The concordance of Figures 9 and 10 demonstrates the feasibility of MATH's modeling techniques to make the impact of HIT on clinical workflow predictable and measurably beneficial. Although the results should be replicated across a range of clinic situations, patient contact

represents a considerable scope integrating HIT functions with automated functions, communications and manual activities of clinicians and patients into a coherent, efficient workflow, with error handing for patient safety.

The patient contact project is currently at the step for Feasibility and Cost Analysis (Step 5 in Figure 1). An earlier prototype won an award using the common SMART Apps API to access test patient records [38]. For ethical reasons the alpha study used graduate students acting as patients. The promising results of the feasibility demonstration should justify additional investigations leading towards widespread adoption. The alpha version of *Priority Contact* is made up of web services that allow for flexibility and customization. Data interoperability, however, is not only a technical issue. Policies about security and permission have to be negotiated. So, the value of the benefit will be weighed against the difficulty of changing policy or getting exception to it in the Trade-off Analysis (Step 6).

Although we cannot make claims about causality on the basis of one study, the accuracy of the predictions was certainly based in large part on the level of detail in the workflow models. The to-be workflow was planned as part of the design and task time estimates for clinicians included data from testing user interface prototypes, three-point estimates from subjects and estimates based on similar tasks. Team member skills and the enthusiastic participation by clinicians factored in the study's success.

The main methodological conclusion is that a systems modeling approach using MATH can work well to discover how HIT should be applied to improve the workflow of clinical care. MATH adds value in several distinct ways:

Information modeling

Workflow models can serve as a heuristic to increase the accuracy for identifying the information that is actually used. A workflow can add important context that aids recollection of information use, as compared to conventional methods, such as focus groups. Workflow modeling also reveals inconsistencies or gaps in our understanding of information use, which can be addressed in follow-up interviews or observations. Also, by

treating information as a resource, instead of a task, the workflow models are more tractable in size and complexity.

Cost effectiveness

Our methodical technique of stepping through the workflow to identify needed information provides a technique to cross-check for greater thoroughness of both tasks and information needs. This iterative part of our method increases its focus for cost effectiveness. Our method for model-based systems engineering requires moderately high levels of skill in several areas: planning and carrying out semi-structured interviews, analyzing existing information resources and standards, and systems modeling in a diagramming language based on the BPMN standard. This combination of skills pays off by needing less time from the clinic personnel. This is an important practical factor since cooperation of clinicians is essential to the success of any modeling project and their time is scarce. MATH models can be reused for other projects. We expect libraries of reusable model components to reduce the modeling effort over time.

Automated generation of an information dictionary is another cost-effective feature of MATH. The dictionary indexes information attributes to the each of the tasks where they are used, giving some indication of the value of that specific information. It also indexes the information to the immediate information resource, which reflects the value of the resource. Redundant information resources often add overhead cost to manage them and keep them synchronized. The information dictionary has important implications for standardizing information types/usage.

Information resources, like other types of resource, constrain the way people can use them to do their work. A workflow reflects many other factors as well, but our method shows how, given those conditions, a given set information resources impact a work system. We have identified three possible strategic options to take advantage of this principle:

1. Understanding and problem discovery: MATH identifies what information is actually used, where it is used in workflows, and how the resources that provide it constrain care. This was illustrated by the as-is model in the feasibility demonstration of this report.

2. Problem investigation: A given problem area in operations could be specified, and MATH could be applied to diagnose the role of current information resources and analyze options to mitigate or eliminate the problem.

3. Evaluating the impact of new information resource: A model of current operations could be analyzed for the impact of a new information resource during its design stage. This formative evaluation would guide design decision making to prioritize functionality by positive impact and identify negative impacts and assist to design ways to avoid or mitigate them.

MATH stops short of deciding how the beneficial impact of HIT should be used. Instead, MATH makes the benefits explicit to allow healthcare leaders to decide their value in an informed manner. For example, time-savings could be used to see more patients, to spend more time with current patients, or to alleviate overworked personnel. MATH brings innovations in workflow modeling for HIT design:

1. Integrating the representation of healthcare workflow and information flow,

2. Identifying information problems and solution options,

3. Synchronizing the design of HIT functions with manual tasks to form a coherent overall workflow,

4. Prioritizing options for HIT functions on the basis of evidence of benefit to workflow, including analyzing trade-offs of better information resources for less need of physical resources,

5. Providing a clear connection between improvements of clinical workflow and the design of the HIT software.

These capabilities enable a new, evidence-based approach to HIT design that can be rapidly translated into software specifications. By filling the strategic gap of conventional approaches we can move towards a vision where HIT serves as a methodical means to reduce health care cost while improving its quality. Physicians, executives, and health care leaders must select and direct HIT projects, but in conventional approaches they do not have sufficient information to answer such fundamental questions as:

A. How will a new EHR change current clinical care activities and decision making?	D. How large are the reductions in cost of care?
B. What benefits to care will the new arrangement of activities and decisions bring?	E. Is there a range of options available for HIT functionality ?
C. Are there undesirable impacts on care or cost?	F. How favorable is the return on investment for each option?

Table 2. Fundamental questions about EHR.

Failure to answer such fundamental questions during software engineering for HIT creates risk of unpredictable, negative impacts on care. Until HIT can be understood in terms of added-value to care and applied reliably to realize those benefits, its potential to improve the quality of care while reducing cost could remain elusive. Conversely, supporting health care leaders with the answers will enable them to them to plan and compare HIT projects, and provide the visibility needed to direct the execution of projects in a manner that reliably achieves planned improvement to care and its cost. Most importantly, it will allow healthcare leaders to participate in concept design by deciding the appropriate role of computing for their professional responsibilities.

Workflow reflects a variety of factors that combine in unique ways. Some of the factors unique to a clinic may be too important to its success to sacrifice for a one-size-fits-all HIT application. HIT must play a better supporting role to realize its great potential to improve healthcare. MATH can quickly design custom solutions to maximize benefit and avoid unwanted impacts.

Future Direction

Patient contact is intended as just one example of how MATH can capture current care with existing information resources, identify options for measurable improvement with HIT, and provide evidence to select an HIT solution. Current MATH projects include a Patient-Centered Case Management System for multiple sclerosis, a referral clinic for chronic pain treatment, and a hospital admissions workflow planned to start in September, 2014.

These projects will take advantage of more powerful features that have recently been added to MATH to take advantage of information modeling. For example, the decision logic for gateways can now be governed by information values that are output by tasks. This feature, in turn, allows us to model more complex interactions between information systems and workflow when the information from one task can govern the behavior of another part of the workflow. MATH also now has the capability to calculate the information architecture needed by the information flow, and export them as Java classes. A new web-based version of the MATH toolsuite is also in development.

ACKNOWLEDGMENTS

This project was supported by Grant No. 10510592 for Patient-Centered Cognitive Support under the Strategic Health IT Advanced Research Projects (SHARP) from the Office of the National Coordinator for Health Information Technology. This study was approved by the University of Washington (UW) Human Subjects Division (IRB40065) as well as the Veteran's Administration (VA) Institutional Review Board and Research & Development Committee (MIRB00553).

SUGGESTED READING

Dekker, M., Butler, K. A., Esposito, C., & Hebron, R. (1999). Connecting the design of software to the design of work. Communications of the ACM, 42(1), 38-46.

Chen, Y. (2010). Documenting transitional information in EMR. Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta, Georgia, USA.

Kushniruk, A. W., Triola, M. M., Borycki, E. M., Stein, B., & Kannry, J. L. (2005). Technology induced error and usability: the relationship between usability problems and prescription errors when using a handheld application. Int J Med Inform, 74(7-8), 519-526. doi: 10.1016/j.ijmedinf.2005.01.003.

Zhang, J., & Walji, M. F. (2011). TURF: toward a unified framework of EHR usability. J Biomed Inform, 44(6), 1056-1067. doi: 10.1016/j.jbi.2011.08.005.

12: Developing EHR Design Guidelines

Yang Gong, MD, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Lei Hua
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Xinshuo Wu
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Hsingyi Song
The University of Texas Health Science Center at Houston School of Biomedical Informatics

ABSTRACT

We describe the development of usability and safety-enhanced design (SED) guidelines for creating electronic health record (EHR) systems. A systematic search and expert review process identified 303 design principles from four major guideline documents. Principles were grouped into 14 categories (consistency, visibility, match, minimalism, memory, feedback, flexibility, messages, errors, closure, undo, language, control and help) and illustrated examples added. The EHR Design Guidelines are freely available at https://sbmi.uth.edu/nccd/ehrusability/design/guidelines/.

INTRODUCTION

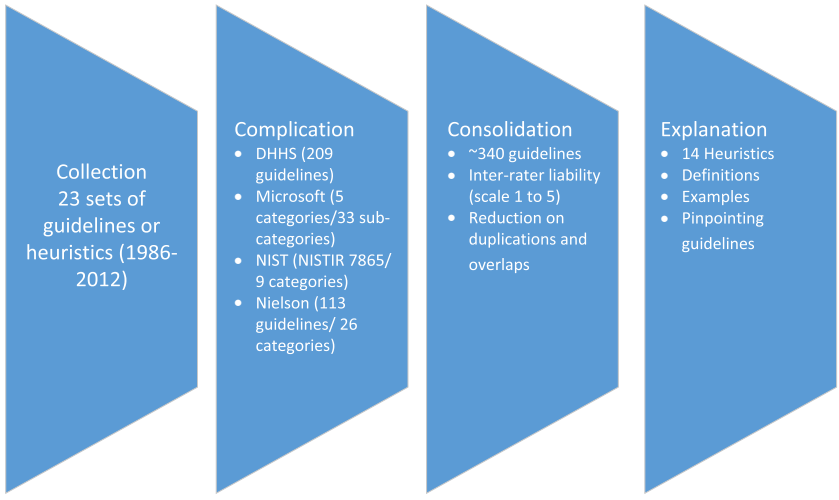
We developed user-centered electronic health record (EHR) system design guidelines to minimize issues hindering EHR usability. Compared to the term "standards," which implies mandatory quality levels and minimum requirements, "guidelines" are less stringent general suggestions and advice. EHR Design Guidelines are best practices that target novice and expert designers of EHR systems. Purchasers of EHRs may also find the Guidelines useful, especially when evaluating new systems or customizing existing products.

EHR Design Guidelines differ from general information technology interface design recommendations because they are specific to healthcare. The Guidelines are intended to be practical, but flexible—neither overly

broad or finely detailed. The Guidelines encourage safety-enhanced design (SED). Many of the recommendations are based on academic theory and validated by empirical studies. Most are presented with justification and concrete examples.

APPROACH

The EHR Design Guidelines were developed in four steps (Figure 1).



DHHS: Department of Health and Human Services
NIST: National Institute of Standards and Technology

Figure 1. Summary of general guideline development.

Data collection

We identified 23 documents related to usability through electronic searches of Medline (1946-2012), Google, and Google Scholar with the search terms and keywords:

- *Guidelines* (MeSH & "explode"), *principle*, *heuristic*,
- *Software design* (MeSH & "explode"), *information systems* (MeSH & "explode"), and *usability*,

Search terms and keywords were conjuncted by "or" in each category, and "and" between categories. MeSH terms included narrowed terms.

Inclusion criteria

We reviewed articles:

- That were related to a computer-based system,
- That were related to usability, human computer interaction or human factors,
- Whose guidelines or principles included system design "do's and don'ts,"
- Originated from a trustworthy source or were supported by empirical studies and/or validation procedure.

Four documents were selected for consideration. Two examiners rated each document as "potentially relevant" or "potentially not relevant." From each "potentially relevant" document we collected: year of publication, author, level of descriptive granularity (high/low), reliability (high/low), healthcare relevance, number of guidelines, and URL. In case of missing references, we circulated findings to a group of usability experts for justification and help uncovering missing items.

Guideline compilation

The four documents included instructional principles for system design. All principles were extracted into a Microsoft Excel spreadsheet for group review to ensure each guideline was applicable to the design and development of EHRs and SED. Duplicates and overlaps were consolidated using an inter-rater method.

To ensure guidelines were EHR specific, SED's eight meaningful use (MU) objectives were mapped: medication list, drug-drug and drug allergy interaction checks, medication allergy list, e-prescribing, computerized provider order entry (CPOE), clinical decision support, electronic medication administration record and clinical information reconciliation. We found some guidelines did not fit into any MU objectives whiles others fit into many MU objectives. Reviewers eliminated guidelines that did not fit an MU objective and labeled multiple MU objective guidelines as general principles.

Consolidation and explanation

A total of 303 guidelines were categorized into 14 heuristic principles [57] with illustrative examples and cross-category references. Consolidation occurred via group discussion using inter-rater methods to resolve duplicates and overlapping guidelines from multiple sources of documentation and to classify guidelines into an established coding scheme.

We used the heuristics developed by Zhang, et al [57] as our coding scheme and generated 14 heuristic-specific reports listing applicable principles with examples.

RESULTS

Our Medline search returned 25 articles. After reading each title and abstract, none was considered "potentially relevant" to EHR system design. Google and Google Scholar identified 23 documents. Through group discussion, experts suggested two additional documents as "potentially relevant" that were not identified by our search strategy: white papers and industrial brochures that lacked explanations and validation processes.

We excluded articles that were either rules of thumb or reorganizations of established guidelines. Four documents from reliable sources were selected for general guideline compilation (Table 1).

Year	Title	Domain	Volume	Method
2001	113 Design Guidelines for Homepage Usability	General	26 categories 113 guidelines	Empirical study based
2006	The Research-Based Web Design & Usability Guidelines	General	209 guidelines	Expert review
2012	Microsoft Health Common User Interface Guidelines	Health	33 categories with thousands guidelines	Expert review
2012	A Human Factors Guide to Enhance EHR Usability of Critical User Interactions when Supporting Pediatric Patient Care	Health	9 categories of recommendations	Expert review

Table 1. General features of qualified guideline documents [197-200].

Compilation of guidelines based on MU objectives

Table 1 principles were tabulated as shown in Figure 2. Principles were categorized by applicability to MU process, sequence ID, principle summary, source document and access hyperlinks. Numbers represent an internal index to locate related descriptors. For records without an index, such as principles from Microsoft Corporation's common user interface (CUI), hyperlinks provide access.

Reviewers justified the relevance of each principle to EHR design using a color code (Figure 2). Green indicates the principle was applicable, red inapplicability, and yellow uncertainty. Reviewers then matched each principle to a MU task and labeled with the top two-most applicable MU processes. Each record was duplicated in additional spreadsheets by MU task. A principle was considered "general" if it applied to more than four MU tasks. Inconsistent labels were resolved through group discussion.

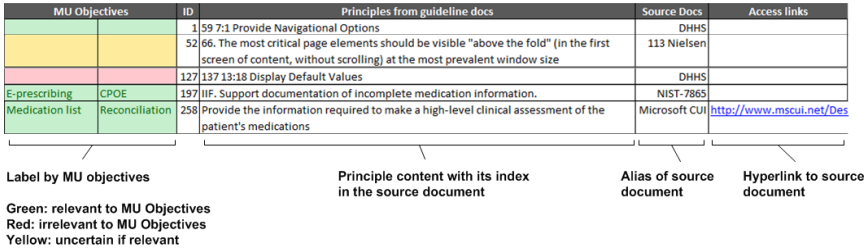


Figure 2. Table structure of compiled guidance principles.

A total of 303 principles were identified as applicable to the design of EHR systems. Approximately one-third (133) were classified as general. The remaining 170 principles were categorized by the eight MU processes (Table 2). While we tried to assign principles to exclusive categories, approximately one half were cross labeled, meaning they applied to more than one process. Table 2 shows the distribution of principles by MU process. Table 3 shows e-prescribing as an example of how assigned principles were organized under each category of MU objectives.

MU objectives	Number of principles that fit into this category (may overlap)
Medication list	8
Drug-drug and drug allergy interaction checks	25
Medication allergy list	25
E-prescribing	55
CPOE	64
Clinical decision support,	12
Electronic medication administration record	13
Clinical information reconciliation	45

Table 2. Distribution of principles by MU objectives.

The majority of results retain their original section or sequential number under the Principles column. Original numbers are locators for retrieving principle details in corresponding source documents. Because guideline principles extracted from Microsoft's CUI do not carry numbers [200], we added hyperlinks to source files. The numbers attached to Principles from other source documents include:

- DHHS – the format of numbers is (A B: C) – A is page number, B is chapter number and C section number [198].
- 113 Nielsen – the format of numbers is (A) – A is the sequence number [197].
- NIST-7865 – the format of numbers is (A) – A is the sequence number in an aggregated table of guidelines [199].

ID	Principles	Source Documents
113	121 13:1 Distinguish required and optional data entry fields	DHHS
114	123 13:3 Label data entry fields consistently	DHHS
…	…	…
135	71. Use dropdown menus sparingly, especially if the items in them are not self-explanatory	113 Nielsen
136	24. Only use imperative language such as "Enter a City or Zip Code" for mandatory tasks, or qualify the statement appropriately	113 Nielsen
…	…	…
192	IIA. Protect against mode errors for mg/kg dosing and ml dosing.	NIST-7865
193	IIB. Flag that an intended dose is unusual.	NIST-7865
…	…	…
247	Provide a visually-rich chart of information relevant to, and prioritized for, the administration of drugs	Microsoft CUI
248	Support the presentation of drugs with different characteristics (such as Significant Duration, Once Only or As Required drugs) within one view	Microsoft CUI
…	…	…

Table 3. Compilation of guidelines for e-prescribing.

14 heuristic-based guideline reports

Each guideline was assigned the most appropriate heuristic category and referenced in other relevant categories. Reports begin with a definition, applied situation and heuristic example (Figure 3). Guideline principles are organized into sub-categories.

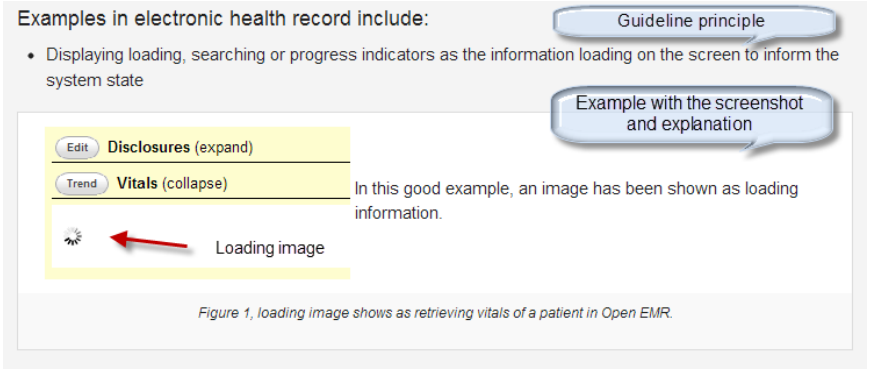


Figure 3. An example in visibility guidelines.

Each guideline includes a locator linking the document and page where the principle originates. Detailed descriptions and examples are also given (Table 4).

Visibility regarding navigation
Locate the primary navigation area in a highly noticeable place, preferably directly adjacent to the main body of the page (Zhang et al., 2003) Nielsen's 40
Group items in the navigation area so that similar items are next to each other (Zhang et al., 2003) Nielsen's 41
Use site maps (Nielsen, 2001) Usability.gov 7:10
Breadcrumb navigation (Nielsen, 2001) Usability.gov 7:12
Place primary navigation menus in the left panel (Nielsen, 2001) Usability.gov 7:5
Provide navigational options (Nielsen, 2001) Usability.gov 7:1
Provide feedback on user's location (Nielsen, 2001) Usability.gov 7:4
Present tabs effectively (Nielsen, 2001) Usability.gov 7:7
Use descriptive tab labels (Nielsen, 2001) Usability.gov 7:6
Visibility regarding page and content layout
Show dates and times for time-sensitive information only (Zhang et al., 2003) Nielsen's 105
Use appropriate menu types (Nielsen, 2001) usability.gov 7:9
Order elements to maximize user performance (Nielsen, 2001) usability.gov 12:1
Format lists to ease scanning (Nielsen, 2001) usability.gov 12:3

Table 4. Layout of guideline principles in a typical report.

DISCUSSION

Time constraints and the innate disadvantages of search engines and databases for systematic reviews made it impossible to include all relevant resources. Our results are, therefore, based on a subset of existing guideline documents. To compensate, we detailed our methods to make our search strategy reproducible to others. This may also help expand the

guideline collection to further drive improved EHR usability and SED outcomes.

13: Safety-enhanced Design Briefs

Todd R. Johnson, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Krisanne Graves, PhD, RN
The University of Texas Health Science Center at Houston School of Biomedical Informatics

J. Franck Diaz-Garelli
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Catherine Plaisant, PhD
University of Maryland

C. Adam Probst, PhD
Baylor Scott & White Health

ABSTRACT

We created single-page, evidence-based, safety-enhanced (SED) design briefs to help electronic health record (EHR) developers and implementers improve system usability. Some briefs address SED meaningful use, such as clinical information reconciliation. Others cover human factors issues, such as effective use of color. SED design briefs are available https://sbmi.uth.edu/nccd/SED/Briefs/ and are accompanied by supplemental material. The briefs are also cross-referenced to other EHR design, meaningful use, and SED materials.

INTRODUCTION

Safety-enhanced design (SED) certification for Meaningful Use Stage 2 requires summative testing of electronic health record (EHR) usability functions such as clinical information reconciliation and clinical decision support. SHARPC supports SED with improvements to EHR system usability and learnability. SHARPC teams developed tools to evaluate EHR design and usability, and EHR inspirational prototypes interfaces. The National Center for Human Factors in Healthcare was engaged to verify SHARPC products' suitability and identify additional vendor needs. Interviews indicated vendors lacking human factors design experts desired short, actionable advice towards SED certification. Feedback also

revealed inconsistencies in how vendors understood usability and user-centered design. Some equated user-centered design to vendor responses to user feedback. These results suggested that, in addition to brief design suggestions, vendors could benefit from in-depth information. We developed 12 single-page, SED briefs and a dissemination website with references to supplemental information.

APPROACH

SED Briefs are part of a suite of SHARPC usability products that include an EHR usability website and an online EHR design book written from a clinical perspective. A team of approximately 25 SHARPC researchers teleconferenced weekly for several months to coordinate and develop the three products. Several members were active in more than one guideline product. For coordination and collaboration, we created a guidelines website on *Basecamp*, a web-based project management and communication tool that provides a team calendar, electronic to-do lists, file repository, messaging and mailing lists.

SED Brief selection criteria included:

- Target audience: Developers at small to medium EHR companies lacking experts in user-centered design.
- Length: Single page.
- Select guidelines that:
    1. Are the most critical, actionable, and relevant to EHR usability and safety-enhanced design,
    2. Can be implemented in existing EHRs,
    3. Distill current theory into practical advice.
- Media: PDF with hyperlinks.
- Cross link, harmonize, and coordinate with other SHARPC guidelines products.

We planned one design brief for each SED certification requirement and additional briefs for SHARPC inspirational prototypes:

Safety-enhanced design briefs

- Computer physician order entry (CPOE)
- Clinical decision support
- CPOE medicine orders and e-prescribing
- Clinical information reconciliation:
    1. Medication reconciliation
    2. Allergy reconciliation
    3. Problem reconciliation
- Effective alert design
- Medication allergy list
- Medical list
- Electronic administration record

Additional design briefs

- Results management
- Reducing wrong patient selection errors
- Effective use of color
- Table design

We refined our list as work progressed. We developed style guidelines outlining required elements for each brief and a writing style to ensure consistency, including:

Title

- Meaningful use SED title and subtitle if needed, for example, *Clinical Information Reconciliation: Medication Reconciliation.*
- SED goal, for example, *Reducing medication errors at transitions of care.*

Background

- The SED importance/purpose of the guidelines.

Guidelines

- Include concepts most relevant, critical, and vendor actionable.
- Include what should be done instead of what should not be done.
- Optionally organize by category, such as System Design, Screen and Interaction Design, Workflow, etc.
- Guideline formats:
    1. Goal + an imperative statement, for example, *To ease comparison of medications, highlight differences between similar drugs.*
    2. Goal + a list of imperative guidelines, for example:

        To help verify as order is placed:

        - *Display patient's information in the submit button*
        - *Or consider placing the submit button near the patient information.*

    3. Imperative statement, for example, *Allow users to group drugs by therapeutic intent.*

    - Mockups or brief explanations near or after each guideline or group of guidelines.
    - Mockups and examples that show positive applications of the guidelines instead of examples not following the guidelines.

To Learn More

- Pointers to additional resources.

Clinical Information Reconciliation: Medications
Preventing Medication Errors at Transitions of Care

DRAFT – June 17 2013

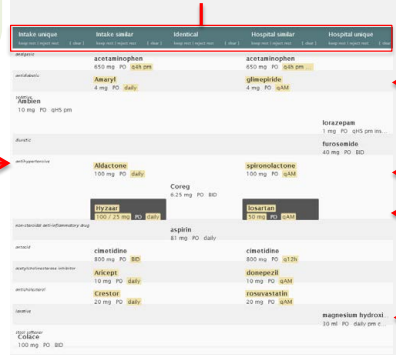
Background: Accurate medication reconciliation at transitions of care can reduce medication related adverse events, thereby decreasing the chance of costly complications. Reconciliation involves comparing two or more lists of medications (such as a home list and in patient list) to determine the appropriate meds for a patient.

GUIDELINES

Screen and interaction design:

- **To help compare medications across lists**
  1. Visually indicate Identical, similar, functionally similar, and unique drugs from each list by using spatial proximity and/or color
  2. Display medications that need to be compared close together
  3. Highlight differences between similar medications
  4. Use visual properties (such as font size) to make the most important elements (such as the name) of medications the most salient.
  5. Show brand and generic names, such as Atorvastatin (Lipitor)

1. Columns show unique, similar and identical drugs



2. Similar medications are shown on the same line

3. Differences of similar meds are highlighted in yellow. Hovering over a med highlights it and any similar meds

4. Med names in bold and on separate line from other details

6. Meds may be grouped by therapeutic intent

5. Generic and brand names are shown for all meds [WE NEED TO MODIFY PROTOTYPE TO SHOW BOTH]

6. **Allow users to group medications by therapeutic intent**
7. **Provide a way for users to see the ingredients in compound drugs**
8. **To minimize alert fatigue and improve efficiency and quality, display information such as medications that are on the patient's insurance, medication interactions, side effects, allergies, etc.**
9. **Display medication information as: Generic (Brand Name), Dose and dosing unit, route, frequency, for indication xxx.** (Atorvastatin (Lipitor) 80 mg by mouth at bedtime for high cholesterol.)
10. **NEVER use error-prone medication abbreviations (e.g., AD, OD, QD), symbols, or dose designations** (See http://www.ismp.org/tools/abbreviations/)
11. **Follow table design guidelines**
12. **Follow color use guidelines**

To learn more:
For more details and background see: a Short Paper , a Video demonstration, or a Usable Prototype.

Figure 1. Draft SED Brief distributed to team members as a template.

We created a to-do list in *Basecamp* for each SED brief and assigned responsibilities. Because some briefs were obtained from other ongoing SHARPC projects, briefs were prioritized by project status. Briefs dependent on results of SHARPC work not yet begun were placed on hold. *Basecamp* allows users to upload files and create discussions organized by to-do. This feature allowed us to track progress and

feedback. We annotated to-do lists to indicate the status of each brief (Figure 2).

We sought input from stakeholders outside of SHARPC, including vendors and members of the EHRA Clinician Experience Group. Their feedback led to a number of changes in the final briefs.

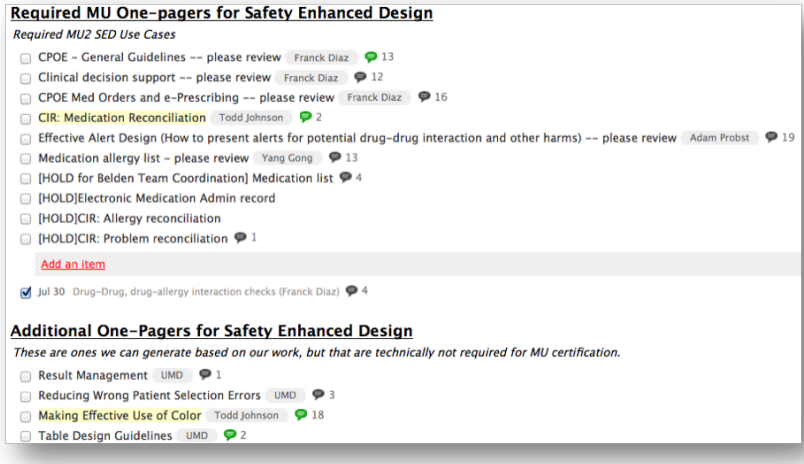


Figure 2. Example of a Basecamp *to-do list. Each brief had a to-do list, responsible lead, indicators for number of comments, and status. "HOLD" indicates a brief is delayed pending results of other SHARPC efforts.*

We organized SED Briefs into those deliverable by the November, 2013 Pre-AMIA 2013 SHARPC workshop and those for delivery at the end of the SHARPC project in 2014. Six briefs in the initial set were given to a graphic designer to produce a template. Version 1 of each SED Brief and the supporting website launched in November 2013. Six additional SED Briefs were later produced.

RESULTS/PRODUCTS

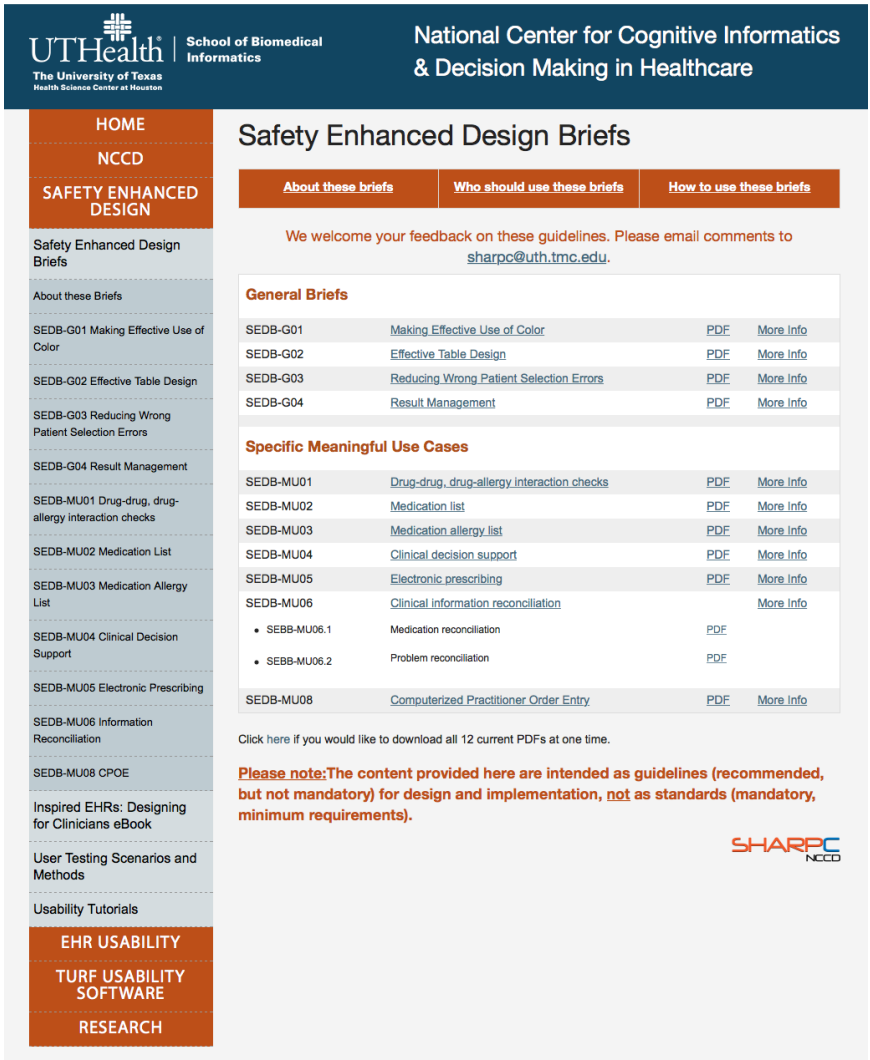
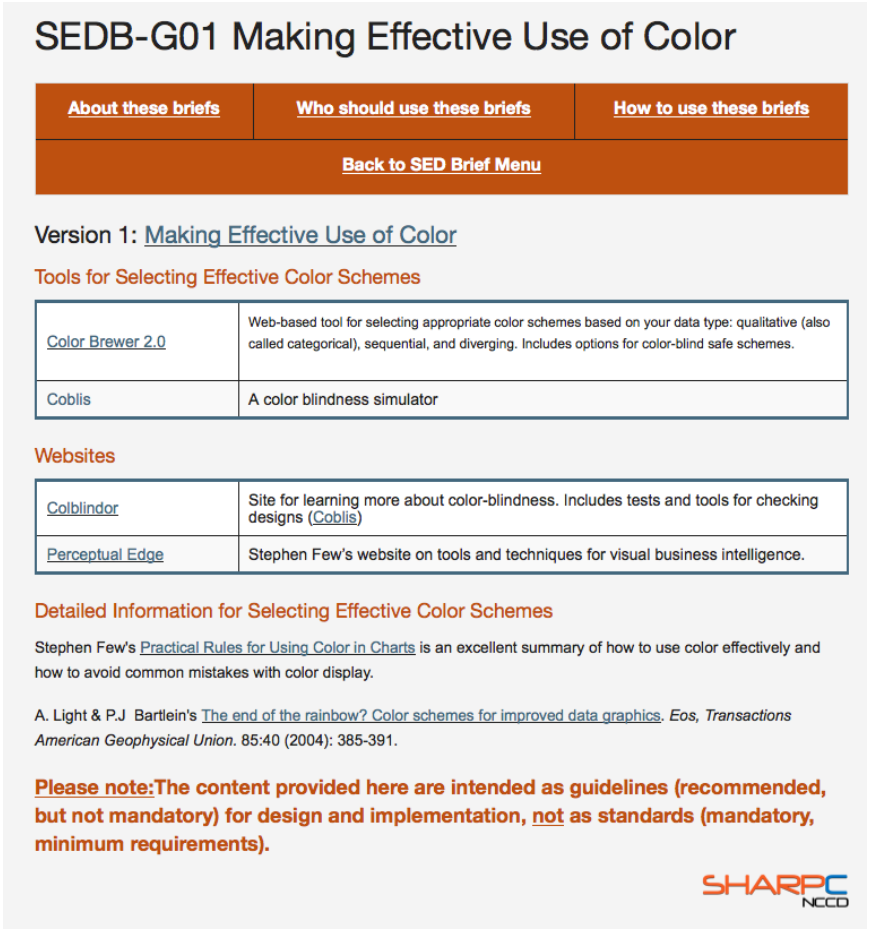


Figure 3: SED Briefs website.



SEDB-G01 Making Effective Use of Color

About these briefs | Who should use these briefs | How to use these briefs

Back to SED Brief Menu

Version 1: Making Effective Use of Color

Tools for Selecting Effective Color Schemes

Color Brewer 2.0	Web-based tool for selecting appropriate color schemes based on your data type: qualitative (also called categorical), sequential, and diverging. Includes options for color-blind safe schemes.
Coblis	A color blindness simulator

Websites

Colblindor	Site for learning more about color-blindness. Includes tests and tools for checking designs (Coblis)
Perceptual Edge	Stephen Few's website on tools and techniques for visual business intelligence.

Detailed Information for Selecting Effective Color Schemes

Stephen Few's Practical Rules for Using Color in Charts is an excellent summary of how to use color effectively and how to avoid common mistakes with color display.

A. Light & P.J Bartlein's The end of the rainbow? Color schemes for improved data graphics. *Eos, Transactions American Geophysical Union*. 85:40 (2004): 385-391.

Please note:The content provided here are intended as guidelines (recommended, but not mandatory) for design and implementation, not as standards (mandatory, minimum requirements).

SHARPC NCCD

Figure 4. Webpage for a single SED Brief containing supporting material for developers interested in a deeper understanding of the principles and evidence behind the brief.

Each SED Brief has its own webpage with supporting information (Figure 4) and references to tools, papers, and websites offering additional information, evidence, and a deeper understanding than possible in a single-page document (Figure 5).

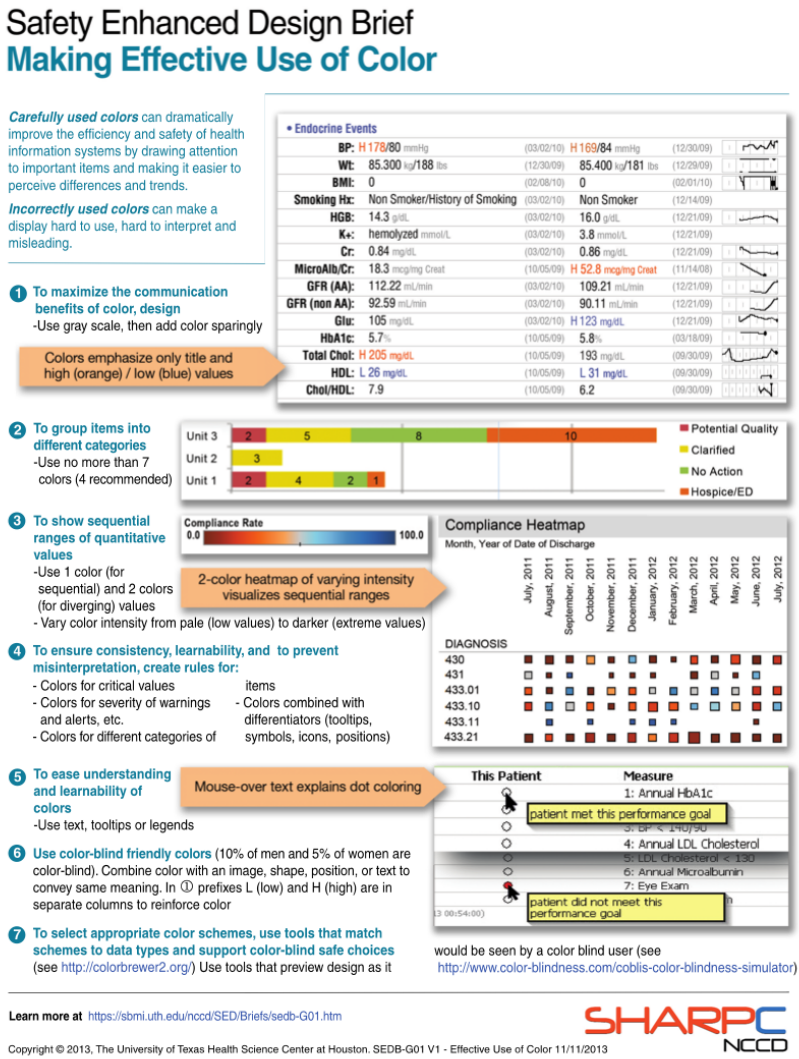


Figure 5. SED Brief showing final graphic design and content for Version 1.0.

Figures 1 and 5 show changes in graphic design and style from draft briefs to final based on feedback from other SHARPC researchers, vendors, and clinicians. The Harvard SHARP team provided an initial round of graphic redesign, including highlighting each guideline to make it stand apart from other content and using the floating arrows to tie guidelines to visual examples. The bottom of each brief includes *Learn more at* that links to the brief's webpage. Each brief has a unique

pneumonic code, for example SEDB-G01, incorporated in the URL, bottom of the brief, and webpage. Version number and date provide users with means to determine if they have the most recent version. Vendor representatives expressed concern that SEB Briefs and examples might be seen as prescriptive requirements. We subsequently added a qualification to each webpage. The main website includes a link for users to provide feedback.

Discussion

To our knowledge, SED Briefs are the first attempt at concise, actionable guidelines to help EHR vendors improve EHR usability, efficiency, and safety. EHR systems are often highly configurable in ways that can directly affect user experience, safety, and efficiency. SED Briefs may, therefore, help implementers choose more effective configuration options. Sites evaluating EHRs or other clinical information systems may also benefit by noting if systems follow these guidelines. We consider the briefs a success if stakeholders value and use them, but we recognize that evaluating their use may be difficult.

Future Direction

We are encouraging use of SED Briefs by contacting interested groups and dissemination at related meetings, such as HIMSS. We continue seeking feedback from vendor and user communities. We plan to revise SED Briefs as work on EHR usability continues, and as EHRs and their underlying technologies evolve.

Suggested Reading

Safety Enhanced Design Briefs. Retrieved July 9, 2014, from https://sbmi.uth.edu/nccd/SED/Briefs/

What is Safety Enhanced Design? Retrieved July 8, 2014, from https://sbmi.uth.edu/nccd/SED/

EHR Usability. Retrieved July 8, 2014, from https://sbmi.uth.edu/nccd/ehrusability/

14: EHR Design eBook

Inspired EHRs: Designing for clinicians

Jeffery L. Belden, MD
University of Missouri-Columbia

Nathan Lowrance, MS
University of Missouri-Columbia

Jennifer Patel
Involution Studios

Richelle Koopman, MD, MS
University of Missouri-Columbia

Joi Moore, PhD
University of Missouri-Columbia

Catherine Plaisant, PhD
University of Maryland

Todd Johnson, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Juhan Sonin
Involution Studios

ABSTRACT

We produced *Inspired EHRs: Designing for Clinicians*, a clinically-inspired electronic health records (EHR) usability design guide eBook. Richly illustrated and interactive, *Inspired EHRs* is based on feedback from an expert advisory panel and the EHR vendor community. eBook chapters cover medication lists, medication allergy lists, medication reconciliation, ePrescribing, computerized provider order entry, drug alerts, clinical decision support, human factors, and usability design principles. *Inspired EHRs* was sponsored by the California HealthCare Foundation and SHARPC and released June 30, 2014 at InspiredEHRs.org.

INTRODUCTION

Electronic health record (EHR) systems' potential to improve the coordination and quality of healthcare is widely recognized [4, 10, 201].

Impaired EHR usability however, frustrates many users [17, 142, 202]. Learnability, efficiency, and cognitive load are frequent barriers. Current EHR designs may not incorporate visual design psychology, cognitive science, and usability best practices. Healthcare providers are acutely aware that their information needs are not being met, but do not know why. American healthcare is a complex socio-technological system. Information structures, needs and workflow often vary by institution. Displays optimized for healthcare providers may fail for patients and caregivers, and vice versa. Instead of improving healthcare, many EHR systems reduce physician productivity and efficiency, and endanger patient safety by making information opaque, hard to find, or misleading.

Human factors specialists and visual designers apply cognitive and visual perception science to produce better products. Numerous articles about user interface design [56, 203], data visualization [204, 205] and human perception [206, 207] have been published, but outside of health information technology and, therefore, lacking clinical perspective. We developed *Inspired EHRs: Designing for Clinicians*, [4] an interactive guide based on expert design principles and recommendations by the American Medical Informatics Association (AMIA) EHR Usability Task Force Report [147]. Many EHR usability studies are research-centric [208-210] or policy-driven [161, 211-213]. *Inspired EHRs* models EHR system design and health information displays that foster efficient, safe, patient-centered care. Our objectives were to:

1. Create an interactive eBook of common design patterns that incorporate EHR patient-safety functions identified by the Office of the National Coordinator, including medication lists, medication allergy lists, medication reconciliation, ePrescribing, computerized provider order entry (CPOE), drug alerts, clinical decision support, human factors, and EHR usability design principles.
2. Incorporate EHR vendor input and feedback.
3. Promote the guide through Health Information Management Systems Society (HIMSS) Electronic Health Records Association (EHRA) publicity and educational channels, the EHRA Clinician Experience Workgroup, HealthIT.gov, sponsor web sites, physician specialty societies, press campaigns, and social media.

APPROACH

A team of health IT design, evaluation, and instructional technology experts collaborated with EHR vendors to address interface issues for seven patient-safety related EHR functions. The 12-month schedule included three major milestones and two final deliverables: an interactive eBook with visual examples and models, and an outcome dissemination report due three months after eBook publication.

Activities

1. An initial meeting was held in Boston to define eBook structure and select a platform (iBook vs. PDF with web supplement vs. fully web-based). Items discussed included:
    a. Medication lists and other key issues, such as cognitive and perception science principles regarding how humans see, read, think, pay attention, remember, and decide.
    b. Clinical scenarios and tasks involving medication lists.
    c. Image galleries with annotated illustrations of medication list designs comparing good and poor examples.
    d. "Deep dive" explanations of technical details for readers seeking in-depth information.
    e. Interactive prototypes illustrating key design components. Example: TwinList for medication reconciliation (http://www.cs.umd.edu/hcil/sharp/twinlist/index.shtml).
2. A kickoff meeting with EHR vendor representatives was held at Involution Studios in Boston. Items discussed included medication allergy lists, medication reconciliation, e-prescribing and CPOE, drug alerts, clinical decision support and other eBook features, such as interactive display modules, image galleries, videos, model prototypes, and evaluation modules.
3. Subsequent vendor meetings were held in Columbia, MO and Boston regarding development of visual designs and interactive prototypes.
4. Weekly project team teleconferences were held between Involution Studios in Boston, the University of Missouri, the University of

Maryland, and The University of Texas Health Science Center at Houston.

5. eBook illustrations were iteratively developed, starting with sketches and EHR screenshots, and then refined to include key teaching points (Figures 1 through 5).

Figure 1. Interactive medication list annotated screenshot from the Medication List chapter.

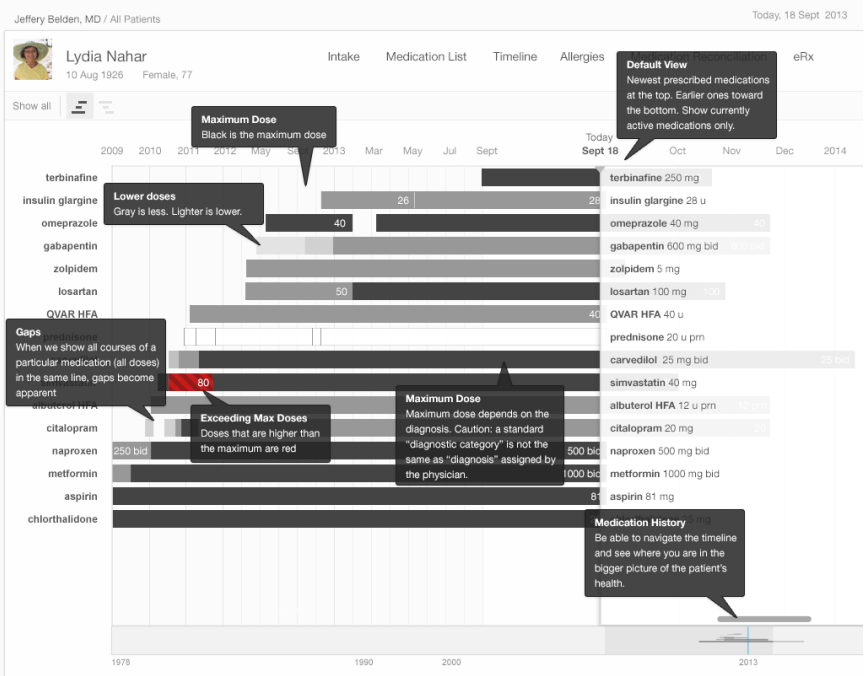


Figure 2. Medication timeline annotated screenshot from Medication List chapter.

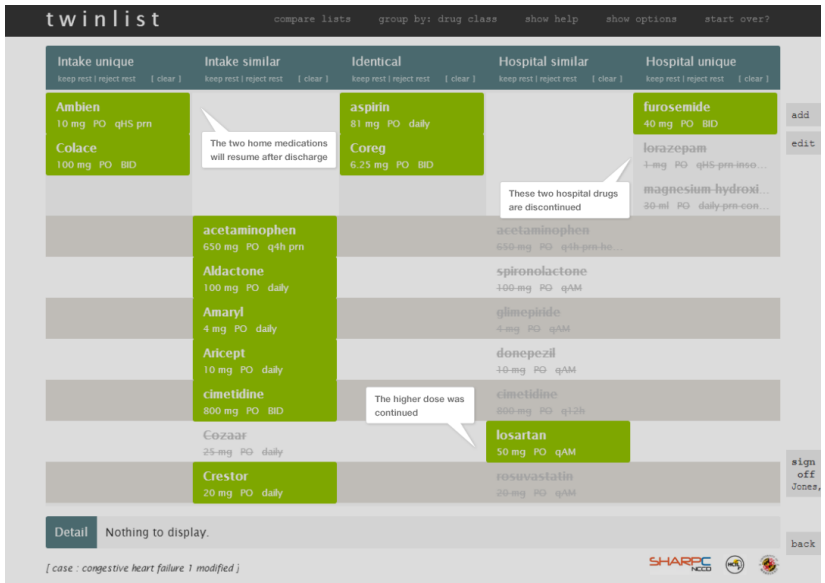


Figure 3. TwinList annotated screenshot from the Medication Reconciliation chapter.

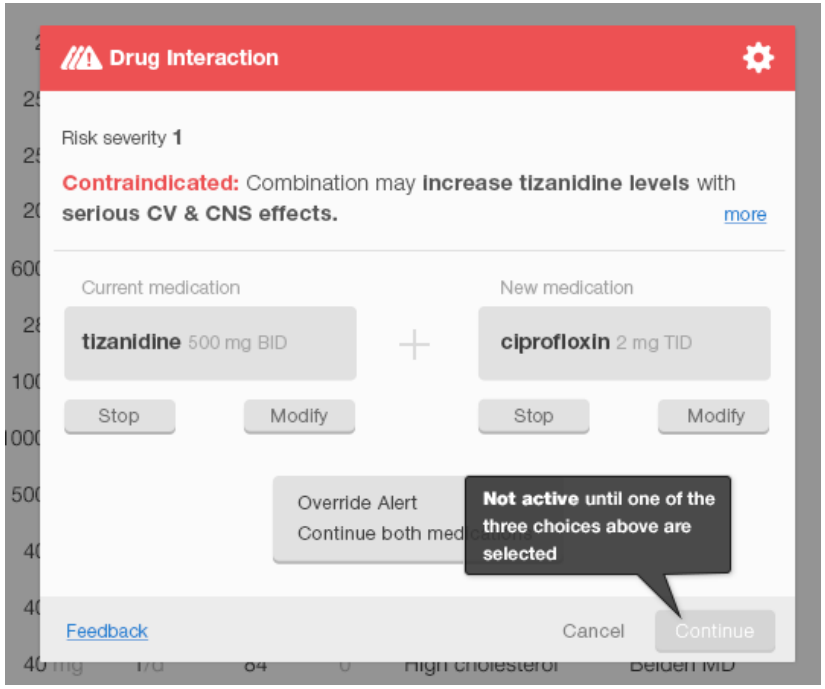


Figure 4. Drug alert annotated screenshot from Drug Alerts chapter.

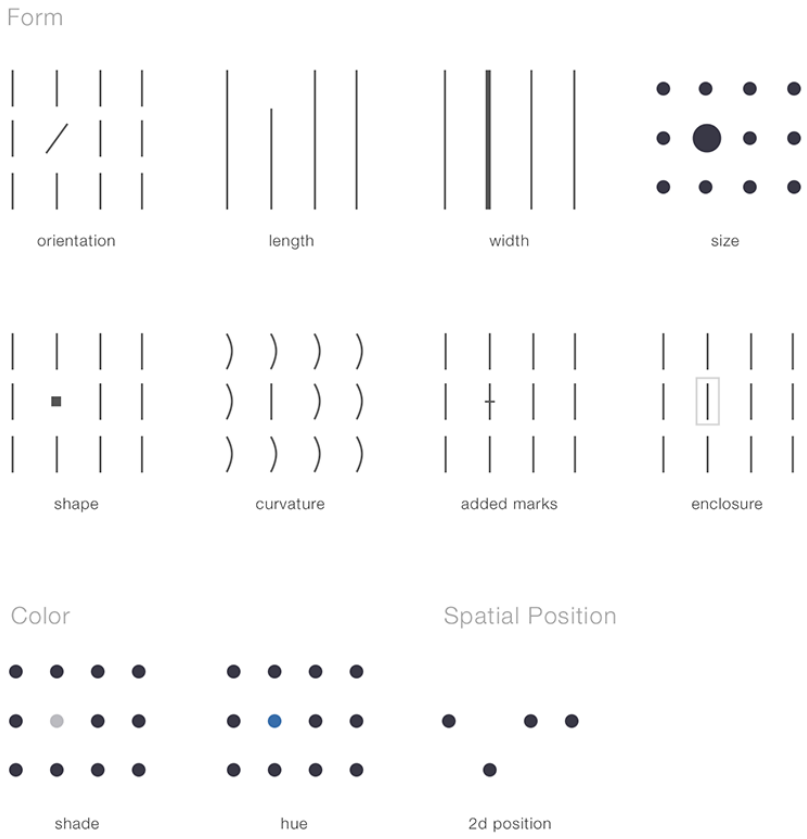


Figure 5. Pre-attentive attributes screenshot from the Human Factors chapter.

6. eBook sections were designed in sprints ranging from four to five weeks.

7. A graduate student assisted with research efforts, instructional design, coordinating and collecting user feedback, and writing.

Expert review

1. Three drafts were distributed to an advisory panel for feedback. Panel members were chosen representing the academic community in human factors research; the electronic/personal health records and health IT vendor communities; other health IT application developers; and others with unique health IT expertise.

2. Two eBook drafts were distributed to a volunteer target audience for review. Readers were solicited through our website, the HIMSS

Annual Conference and the EHRA Clinician Experience Workgroup. Feedback questions included: Was the eBook clear and effective? What elements should be expanded? Which elements should be eliminated or re-conceived? Did the eBook offer the potential to drive improved vendor design processes which could foster improved EHR and PHR products? (Because of platform and resource constraints, reviewers and readers were unable to submit alternative designs via screenshots or mockups).

Deliverables

1. *Inspired EHRs* was completed in March 2014 with seven patient-safety sensitive interface topics inside five sections with illustrations, interactive elements and recommendations.
2. An enhanced version of *Inspired EHRs* was delivered to the California HealthCare Foundation at the end of June 2014 with revised text and clinical scenarios, additional interactive elements and additional illustrative interface design examples. The enhanced eBook is available at InspiredEHRs.org. A downloadable PDF version is also available for those preferring hardcopy.

Dissemination

1. The March 2014 version of *Inspired EHRs* is available at SHARPC and the ONC web sites.
2. *Inspired EHRs* is also available through HIMSS organizational communication and education channels, and EHRA Clinician Experience Workgroup members' individual and group efforts.
3. It was our intent to produce *Inspired EHRs* as an Apple iBook available at no cost on the Apple iBookstore. However, technical issues (user unfamiliarity and difficulty creating "navigation links") caused us to instead release *Inspired EHRs* as an HTML5 eBook website.

Coordinating with other SHARPC projects

Our team coordinated weekly with SHARPC teams producing Safety Enhanced Design Briefs (EHR Safety Enhanced Design Briefs) and the Designing for Usability website (Designing for Usability).

RESULTS

Inspired EHRs is a clinically-inspired, richly illustrated, interactive EHR usability design guide targeted at the EHR vendor community. The eBook is available at InspiredEHRs.org with a downloadable PDF of the latest version. Interactive prototypes include:

1. Medication timeline
2. Interactive medication table
3. TwinList medication reconciliation prototype.

We enjoyed enthusiastic EHR vendor participation from across the industry and close collaboration with the EHRA Clinician Experience Workgroup.

Representative quotes from our EHR vendor target audience:

> I have had a long-standing argument with some engineers about a specific design issue – and thankfully you have suggested exactly what I've been recommending all along. So I can now go back to your documentation and arm myself with more support for my argument. I don't know yet if this will win the fight, but it sure helps to be able to point to a specific central 'voice' for our specific users and use contexts.
>
> *Senior User Experience Researcher and Designer, EHR vendor*

> Congratulations! - By the way, I sent this around to our User Experience team here, and there was a lot of discussion and appreciation for the work you've done.
>
> *Director of User Experience, EHR vendor*

> We built a medication timeline prototype as proof of concept, taking inspiration from your work.
>
> *Director of User Experience, EHR vendor*

> We are building new database structures with the intent of building a medication timeline, thanks to this work.
>
> *Senior Strategist, EHR vendor*

We received numerous inquiries from stakeholders (clinicians, EHR vendors, and the human factors community) about future volumes on additional topics, such as displaying lab results, Problem Lists, and clinical notes.

Discussion

Microsoft produced a Microsoft Health Common User Interface guide in 2010. We similarly considered an *Inspired EHRs* common user interface. The EHRA Clinician Experience Workgroup, while supportive, repeatedly expressed concern that our work could be applied prescriptively. They cautioned that if potential readers were alienated by the title, the content would not have a chance. We decided being illustrative and inspirational, not prescriptive, offered vendors greater freedom in tailoring guidelines to their products. For the same reason, we changed our title from *EHR Usability Style Guide* to *Inspired EHRs: Designing for Clinicians,* believing the term "style guide" was too prescriptive. Our iterative process uncovered other weaknesses in original assumptions, for example:

- It was unclear when describing clinician tasks versus development team tasks. Therefore, we included subsections for "Clinician Challenges" and "Developer Challenges."
- Some clinical scenarios, although realistic, were too complex for non-clinicians to follow. These were simplified or in some cases eliminated.
- After we found the default display for our timeline prototype confusing for may reviewers we produced a short narrated video to give a simple walkthrough.

Based on our expert advisory panel's comments we found the human factors and clinician communities affirming our content and approach.

Clinicians were particularly supportive, with more than one of them using hyperbole to express their enthusiasm about the medication timeline. However, not all designs were met with uniform acceptance. EHR developers using *Inspired EHRs* for inspiration will need to validate their designs through user testing.

Future Direction

Several *Inspired EHRs* chapters could benefit from expanded treatment as eBooks of their own: ePrescribing, Drug Alerts, CPOE, and Clinical Decision Support. Strong interest has been expressed for additional volumes, such as graphing laboratory results, Problem Lists, and progress notes. We are considering a number of single-chapter eBook projects, including:

- Dashboards
- Graphing lab results and vital signs
- Problem Lists
- Data reconciliation

As meaningful use Stage 3 requirements are written for health outcomes and patient engagement, new usability design issues will likely be discovered.

Suggested Reading

Beasley, J. W., Wetterneck, T. B., Temte, J., Lapin, J. A., Smith, P., Rivera-Rodriguez, A. J., & Karsh, B. T. (2011). Information chaos in primary care: implications for physician performance and patient safety. J Am Board Fam Med, 24(6), 745-751. doi: 10.3122/jabfm.2011.06.100255.

DeLeeuw, K. E., & Mayer, R. E. (2008). A comparison of three measures of cognitive load: Evidence for separable measures of intrinsic, extraneous, and germane load. Journal of Educational Psychology, 100(1), 223-234. doi: 10.1037/0022-0663.100.1.223.

Few, S. (2009). Now You See It: Simple Visualization Techniques for Quantitative Analysis (1st ed.). Oakland, CA: Analytics Press.

Johnson, J. (2010). Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. San Francisco: Morgan Kaufman.

Norman, D. A. (1999). Affordance, Conventions, and Design. Interactions, 6(3), 38-43. doi: 10.1145/301153.301168.

Weinschenk, S. (2011). 100 things every designer needs to know about people. Berkeley, CA: New Riders.

15: TwinList

Novel user interface designs for medication reconciliation

Catherine Plaisant, PhD
University of Maryland

Tiffany Chao
University of Maryland

Johnny Wu
University of Maryland

A. Zach Hettinger, MD
Medstar Institute for Innovation

Jorge R. Herskovic MD, PhD
The University of Texas MD Anderson Cancer Center

Todd R. Johnson, PhD
University of Kentucky

Elmer V. Bernstam, MD, MSE
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Eliz Markowitz
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Seth Powsner, MD
Yale University

Ben Shneiderman, PhD
University of Maryland

ABSTRACT

Medication reconciliation is an important and complex task. Careful user interface design has the potential to reduce errors and improve quality of care. We describe a novel user interface called "TwinList" that uses multistep animation to assist clinicians to first differentiate between lists, and then rapidly choose medication to include in a reconciled list. A series of design alternatives with comparative advantages and disadvantages are discussed. We also report pilot study results suggesting animation can help users learn new interface layouts.

INTRODUCTION

Medication reconciliation is a complex iterative task [214-217], a collaborative process where many things can go wrong. For example, patients may not recall what medications they are taking or may be unable to communicate; information may not be recorded properly, leading to uncertainty (e.g., dosage, name or indication); records of past medication may be incomplete or inaccessible; and not all sources of medication orders may be known (e.g., the patient may have consulted a specialist on their own). Eventually the clinician is presented with medication lists from different sources for reconciling into a single, complete and accurate list to be signed and saved in the medical record. User interface design has the potential to reduce errors and improve quality of care. We studied the last step of the reconciliation process—reviewing and sorting medications into lists containing those to be continued and those to be stopped. The result is "TwinList," a prototype user interface providing cognitive support for improved speed and accuracy.

The following example describes a single clinical scenario—discharging a patient from the hospital (Figure 1). The reconciliation process involves comparing two lists; determining what drugs are unique, identical or similar between lists; and making decisions about what to keep, what to discontinue, and what to add or modify.

Intake					Hospital			
Acetaminophen	PO	q6h	325 mg		Acetaminophen	PO	q64h	325 mg
Darbepoetin	SC	qFriday	60 mg		Darbepoetin	SC	qFriday	60 mg
Calcitrol	PO	daily	0.25 mg		Folic Acid	PO	daily	1 mg
Ramipril	PO	daily	5 mg		Omeprazole	PO	daily	400 mg
Meloxicam	PO	daily	7.7 mg		Ciproflocacin	PO	daily	500 mg
Folvite	PO	daily	1 mg		Ramipril	PO	daily	5 mg
					Calcitrol	PO	daily	0.25 mg
					Ferrous Glocanate	PO	TID	300 mg

Figure 1. Discharging a patient from the hospital requires providers compare the "Intake list" (left) and the "Hospital list" (right) and determine what drugs are identical, unique, or similar.

TwinList uses a spatial layout with multistep animation to first help clinicians better understand the similarity of drugs included in lists, and then rapidly choose those to include in a reconciled list (Figure 2).

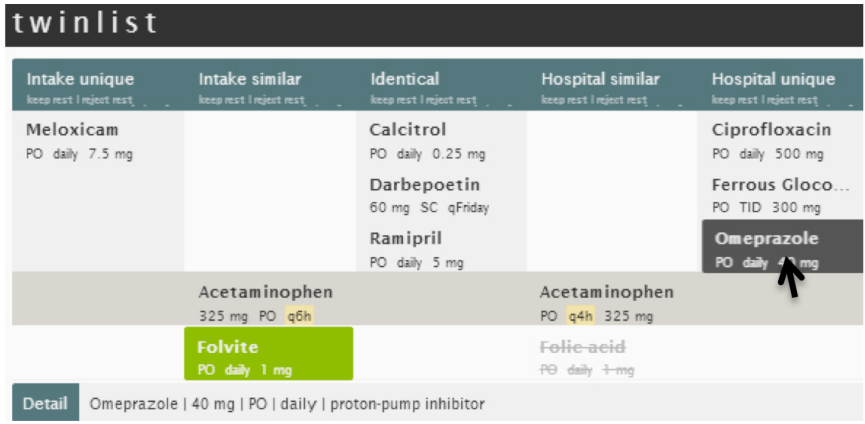


Figure 2. TwinList moves identical drugs to the middle column. Drugs unique to the Intake list move to the left, drugs unique to the Hospital list move to the right. Similar drugs are aligned and differences highlighted in yellow (e.g., q6h versus q4h for acetaminophen). A click on Folvite (a brand name for Folic Acid) selects it–highlighted in green–and deselects Folic acid (grayed out with a strikethrough). Hovering the mouse Omeprazole (dark gray) reveals details at the bottom of the screen.

Background and Related Work

Research shows the need for improved medication reconciliation [215, 218-223]. Duplicate or similar medications may result in overdose and adverse interactions, as well as non-continuation of important medication. Problems can compound by patient misunderstanding or mistrust of new medication, lack of outpatient followup, and changes in medication due to formularies and drug shortages. Trial implementation of medication reconciliation policies show significant improvements. In one study, 94% of patients had medication errors that were eliminated by a medication reconciliation process [214].

There are three kinds of medication error outcomes: harmful (preventable adverse drug events, or PADEs), potentially harmful (near-misses, either intercepted or avoided by luck), and harmless (the most common) [221]. At least 1.5 million harmful errors occur every year

[220]. Patients are particularly vulnerable to errors at care transitions [222, 224] where medication regimens frequently change. Properly reconciling medication at transition points is crucial, but complete and accurate reconciliation is often difficult. Reconciliation is often overlooked or simply not performed (although this is rapidly changing to satisfy new regulations).

While many papers report the severity of medication reconciliation problems, few describe user interfaces used in clinical settings. It is difficult to completely assess current commercial system interfaces due to industry concerns of intellectual property, however many work similar to Pre-Admission Medication List (PAML) Builder [215]. PAML's interface presents medications from all sources in one combined "super list" grouped alphabetically by generic name. The interface exhibits a visual homogeneity that does little to help clinicians distinguish similar from unique medications. We found systems where a clinician might see an intake list in one window, a hospital medication list in a separate window, and the final discharge list in a third. Other systems present a single merged list of all drugs [215] and group drugs with the same name, providing some level of comparison. Algorithms have been proposed to automatically detect similarities between medications [218, 222]. A review described different levels of drug equivalence and showed revealing equivalent drugs can simplify reconciliation based on a detailed keystroke analysis. Other research tried to augment medication lists by linking prescribed medications with clinical problems or indications (either automatically [222] or crowdsourcing [218]) with limited, although promising success.

Overview of TwinList User Interface

TwinList's single window user interface consists of three parts (Figure 2): a header at the top, a list viewer at center, and an item detail panel along the bottom. The list viewer is where users interactively accept/keep or reject/discontinue medication. An early prototype [222] led to a complete rewrite using JavaScript and HTML5. See www.cs.umd.edu/hcil/sharpc for video demonstrations.

Preprocessing: A preprocessing phase identifies similar drugs found in both lists. We use an algorithm [225] (https://github.com/jherskovic/

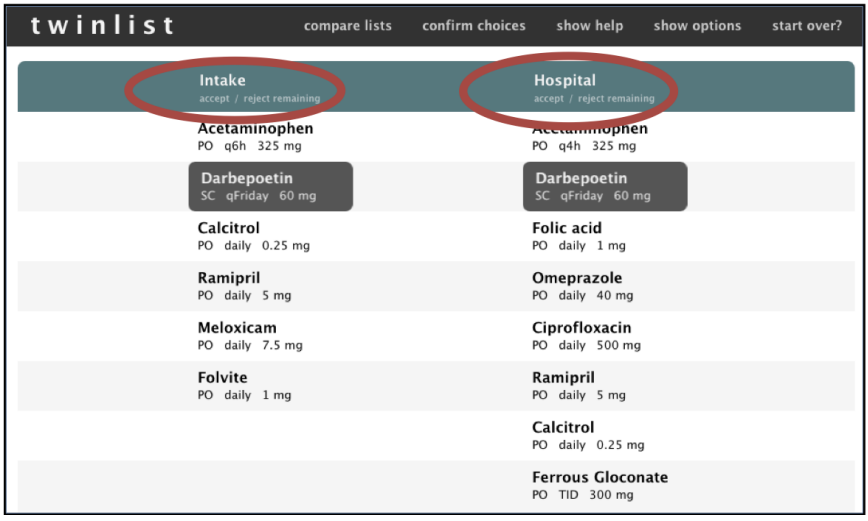
MedRec) to find form equivalence (e.g., Tylenol is a brand name for the generic medication acetaminophen or paracetamol) or functional equivalence (Atenolol and propranolol are both beta blockers). The interface categorizes drugs as "identical" when the same drug appears on both lists (matching name, dosage, route and form), "unique" when appearing on only one list, and "similar" when drugs are equivalent in form, but vary in dosage or other attributes (e.g., acetaminophen 650 mg vs. Tylenol 325 mg). Displayed class information helps group drugs.

Spatial groupings: TwinList places drugs on screen in a multicolumn spatial layout (see Figure 2, and a more complex example in Figures 4 and 5). We believe spatial grouping helps TwinList provide an intuitive way for users to quickly differentiate items (and highlight those differences) between the two lists. The left half of the screen contains drugs from the Intake list. The right half is for drugs taken at the hospital. In the center column are identical drugs (i.e., those present in both lists: Darbepoetin, Calcitriol and Ramipril). Below are three lists of drugs that are similar and aligned to facilitate comparison. For example, acetaminophen is present in both lists, but frequency of use is different (q6h instead of q4h), so both medications and their details are aligned in the same row with differences highlighted in yellow. Folvite is a brand name for folic acid, so both drugs are also aligned on a common row to help the clinician select which similar drug is most appropriate. We found from more than 20 hours of interviews with clinicians and pharmacists that making the source of each list (Intake vs. Hospital) clearly visible gave clinicians the ability to make reconciliation decisions from the perspective of the patient.

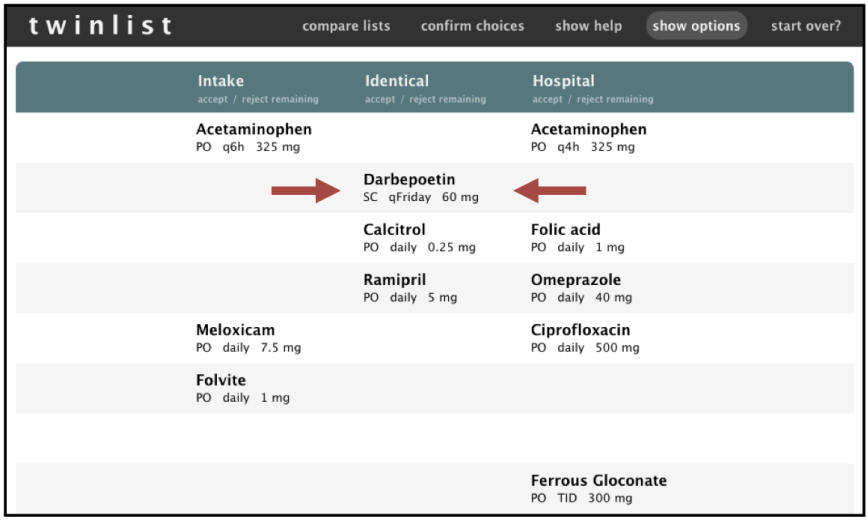
Multistep animation: We used multistep animation to help users understand drug groupings (Figure 3). When lists are loaded into TwinList, they are first presented side-by-side. Options are available to change the speed of the animation or turn it off, which is helpful once a user becomes familiar with the interface. The animation sequence is as follows (Figure 3):

1. *Identical* drugs move to the center column in between the original lists, then merge, one pair at a time.

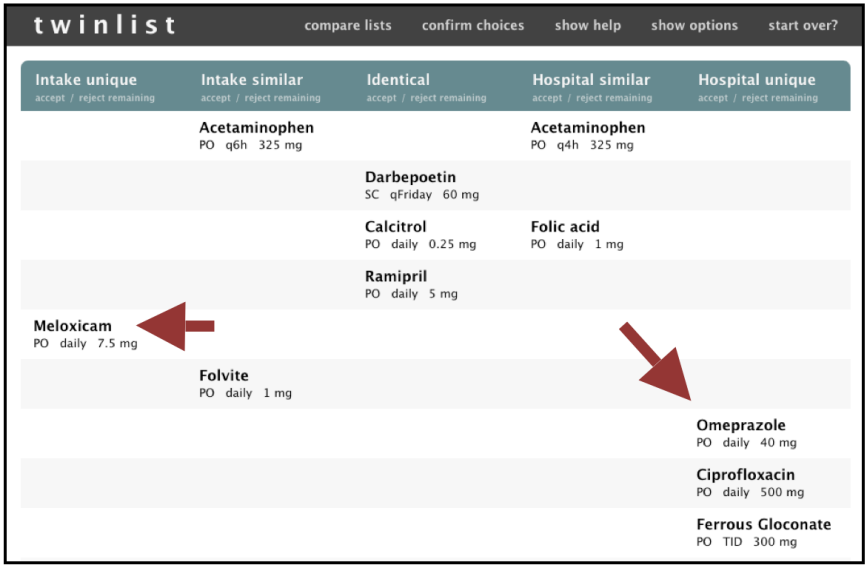
2. *Unique* drugs move away from the center to their respective side, first to the left for drugs unique to the intake list, then to the right for drugs unique to the hospital.
3. *Similar* drugs are aligned and highlighted in gold-yellow to indicate differences between similar drugs.
4. *Compaction* of the display saves vertical space by stacking identical and unique drugs at the top of their respective columns and sliding rows of identical drugs together below.



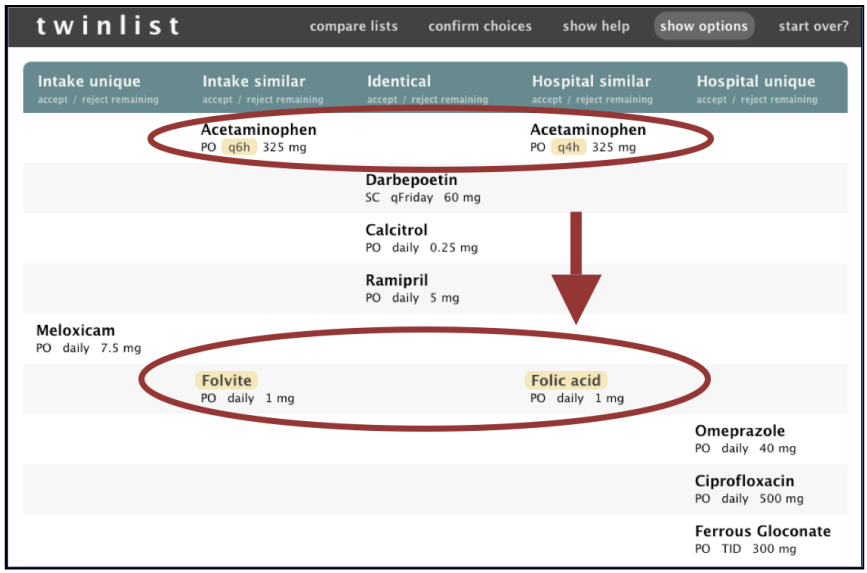
Reconciliation begins as two separate lists: Intake and Hospital.



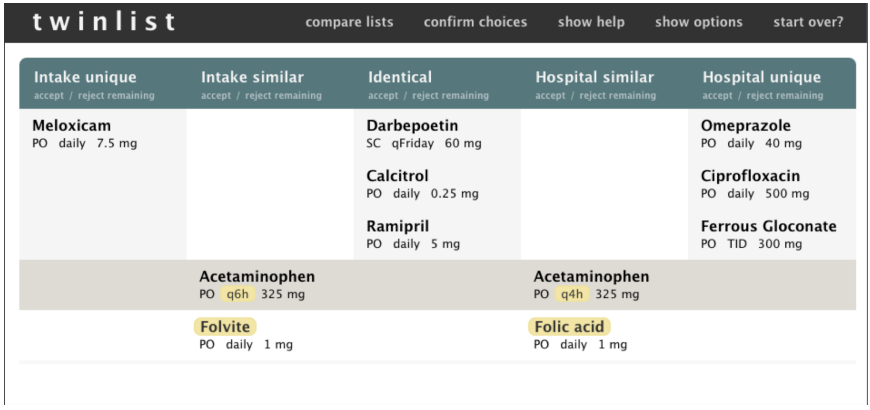
Step 1: ***Identical*** *drugs move to the middle one at a time.*



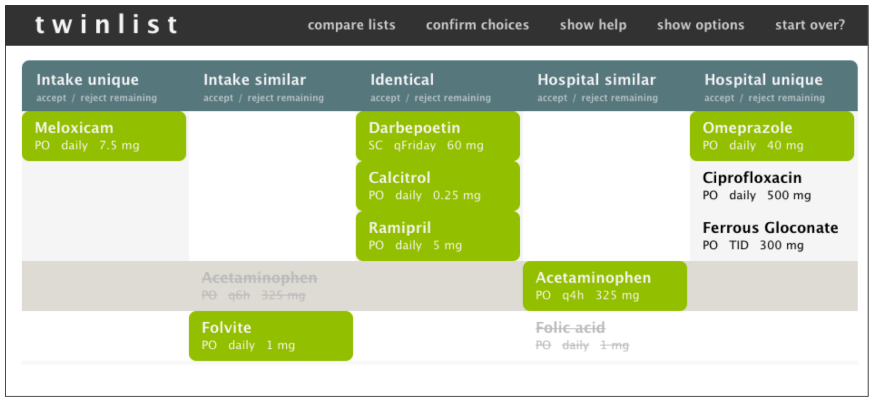
Step 2: ***Unique*** *drugs move to the left, then the right.*



Step 3: ***Similar*** *drugs are aligned and differences highlighted.*



Step 4: ***Compaction*** *of display.*



Step 5. ***Selected*** *drugs to be kept (green) or not (grayed out with a strikethrough).*

Figure 3. Animation sequence used to explain spatial groupings. See video demonstrations at www.cs.umd.edu/hcil/sharp.

Drug Selection: Spatial groupings and highlighting drug differences can assist clinicians make decisions about keeping or discontinuing drugs, one at a time or for entire columns. A left-click accepts the drug, a right-click rejects it. When a drug is selected it appears green, e.g., Folvite was selected (Figure 1 or 3). Rejected drugs appear grayed out and with a strikethrough, e.g., ~~Folic acid~~. Further clicking on a medication toggles through three states: Accepted, Rejected and Undecided. States are selected with a single click (left or right click), with a two-click maximum for users not aware of the right-click-to-reject shortcut. When two drugs are similar (e.g., Folvite and folic acid) the initial selection of one automatically deselects the other, speeding up the selection process. Subsequent clicks make it possible to reject both drugs if needed. The bottom detail panel provides information about the drug if needed and is available by rolling over the drug or by drug selection. All similar drugs (i.e., brand name vs. generic, different dose/route/frequency) in the other list are also highlighted dark gray to attract users' attention. Since users must hover over a drug before selecting it, information about similarities is always displayed (Figure 4). Explicit Keep and Reject buttons beneath column headers provide a convenient way to accept or reject entire columns when appropriate. We chose to err on the side of caution and only apply column commands to medications remaining "Undecided" to avoid overwriting previous decisions.

Signing off: Providers click a Sign off button at the bottom right of the screen when their reconciliation is complete. To reduce chances of medication errors, the Sign off button remains grayed out until every medication has been reviewed and acted upon (Figure 4 and 5). The grayed out button also indicates how many drugs are "Undecided" and includes the name of the patient, which may reduce wrong patient errors.

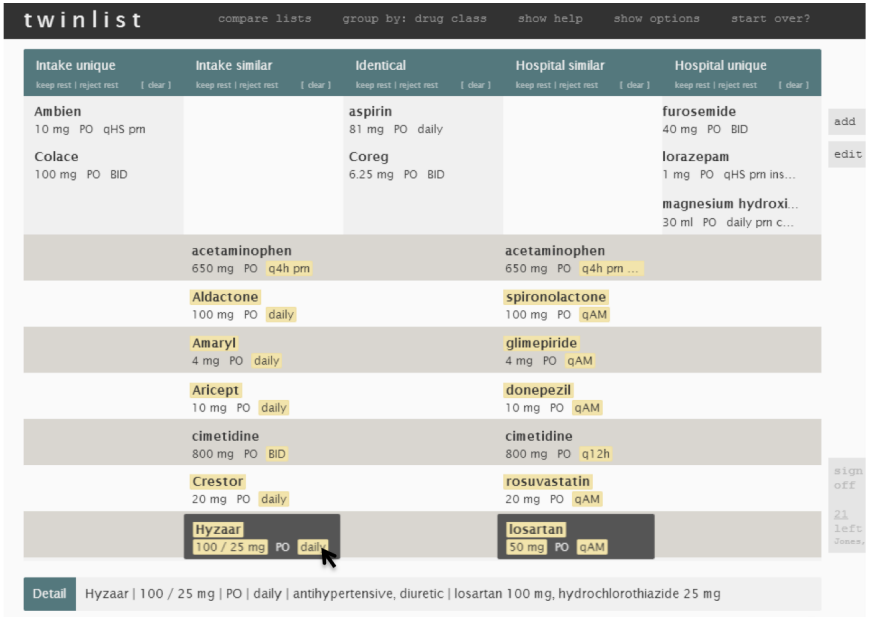


Figure 4. A complex example of congestive heart failure with 11 drugs in the Intake list and 12 in the Hospital list. Here the cursor is over Hyzaar, so details for that drug appear in the detail panel at the bottom (including drug class information). The similar medication Losartan is aligned and also highlighted. Dosage and frequency differences are highlighted in yellow.

Figure 5. All drugs have been acted upon (bright green for "kept" and grayed out with strikethrough for "rejected"). The Sign off button at the bottom right is now active.

Visual design

Line, color, texture, form and space design decisions can make user interfaces simple and understandable, or overly complex. In TwinList particular attention was paid to visual design. Solid colors used sparingly define the interface. Dark gray anchors the header to the top of the page (see Figure 4 for a full view of the interface). Bright white creates a feeling of spaciousness. Color provide accents: gold-yellow highlights important differences between related items; yellow-green lets users know which drugs have been selected at a glance and allows quick review. Clickable objects provide animated feedback on mouse-over. For example, the list viewer nudges to the right to group related medications, exploiting the immediacy of motion and the Gestalt principle of common fate to guide visual exploration. The use of unifying background colors and color coding complements and supports the animation. Reconciliation user interface color schemes and interaction cues should be consistent with those of its main application (e.g., EHR).

Dealing with complex cases with further grouping

Interviews with clinicians, pharmacists and quality assurance officers indicate medication reconciliation errors and less-than-optimal choices are more likely to occur when clinicians deal with difficult cases and long

medication lists (see Figure 4 for a case of congestive heart failure). TwinList's approach reveals numerous cases of similarities and differences in drug name, dosage or frequency. The final reconciled set of drugs is clearly indicated in green (Figure 5).

Interviews suggested different types of grouping (e.g., associated problem, clinical condition, diagnosis, drug class, etc.) provide additional cognitive support for the medication reconciliation process. The current prototype allows medication to be tagged with such attributes, which can then be used to group drugs on the screen. In an ideal setting, individual medications would be linked to the patient's problem list (demonstrating therapeutic intent), however many EHRs do not provide the ability to link diagnosis to medication (or the function is not reliably used), limiting its utility in the reconciliation process. Several ongoing efforts are underway to automatically link information between drug and the therapeutic intent [226, 227]. When available, this information could be shown in TwinList's detail panel with other medication details and used to further organize drugs. TwinList employs high level drug classifications to help users identify potential problems created by patient transition from one healthcare environment to another. Clinical condition may be even more useful. Figure 6 shows an example of grouping by primary drug class. The grouping reveals this complex case includes a large number of anti-hypertensive medications, some of them less commonly used than others and, therefore, at higher risk of misidentification.

Unfortunately, primary class alone may not be always appropriate or sufficient. Medications may be prescribed for other indications or even off-label reasons (acceptable but not FDA-recognized indications). This was a highly debated topic in our interviews, so we explored how interface could show multiple (N) class affiliations. One method was to duplicate the drug N times on the screen; one for every class to which a drug belongs. To indicate additional drug labels were merely ghost copies (not duplicate prescriptions) they are displayed in Figure 7 in pale gray instead of black. While the grouped-by-N-class display became complex (more items on the screen resulting in longer lists), some physicians noted the visual complexity represented the complex reality of the case. Grouped-by-N-class display may not be best as a default user interface, but may be useful: 1) during training; 2) to review decisions before sign-

off in complex cases, or 3) for users preferring drugs displayed metaphorically with case complexity. Although an imperfect solution, Grouped-by-N-class may sometimes be more useful than common alphabetic grouping and could be offered as a display option or preference [228].

Alternative design: Using only 2 columns and showing similarity by dynamic highlighting only

We felt grouping by class potentially useful, but realized five columns creates layouts of drugs spread thinly over the entire screen, losing much of TwinList's original compactness (compare Figures 6 and 7 with Figure 4). Sparsity results from two spatial methods: grouping based on comparison between the lists (i.e., identical, unique and similar, resulting in five columns) and slicing by class, resulting in many small sets of drugs spread over the screen. This led us to reconsider the original grouping of five columns.

Another alternative interface used only two columns. We preserved the strong horizontal separation between Intake (left) and Hospital (right), but reserved the main vertical grouping for drug classes. A disadvantage is that similarity and differences between lists are no longer shown spatially, instead revealed temporarily via highlighting when the cursor hovers over a drug (Figure 8). An advantage is a more compact layout than five columns of class grouping, using about the same screen space as the five column layout, but with a taller, narrower design. Another possible advantage is that the layout can be extended to three or more lists side-by-side, perhaps useful when merging data from multiple sources (e.g., inpatient, outpatient and a pharmacy generated list).

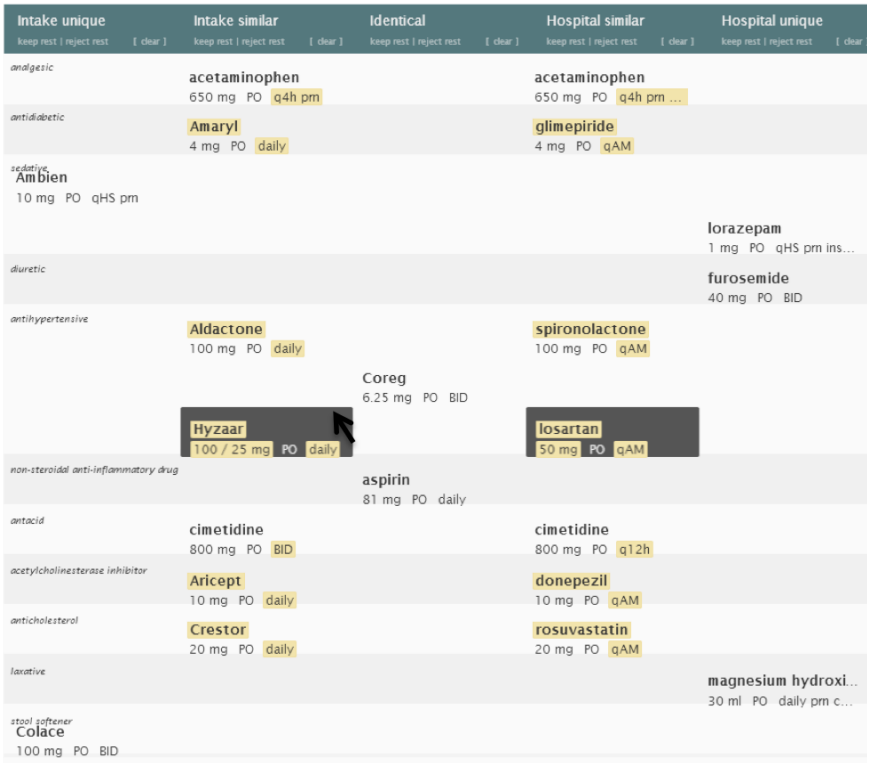


Figure 6. The same case as Figure 5, but now the drugs have been grouped by primary drug class, revealing this complex case includes a total of five different antihypertensive medications. Ambien and Lorazepam are also now grouped in the sedative section, even though they were originally separated. While we use drug class here, the same interface could be used to group drugs by patient diagnosis if linking information were available.

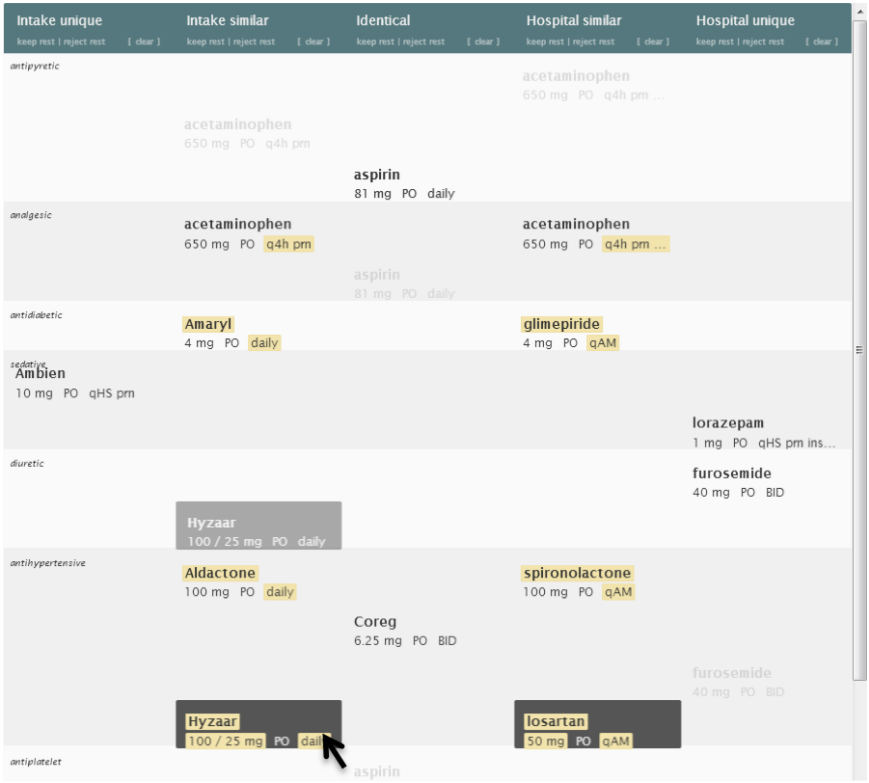


Figure 7. Grouped by all drug classes. Each drug appears in the class to which it belongs. Primary is shown bold, secondary copies appear gray. We see six antihypertensive drugs (Furosemide appears as a ghost copy of its main listing in diuretics). Moving the cursor over Hyzaar reveals it is also a diuretic. Note that the list becomes longer and may require scrolling to see all the classes.

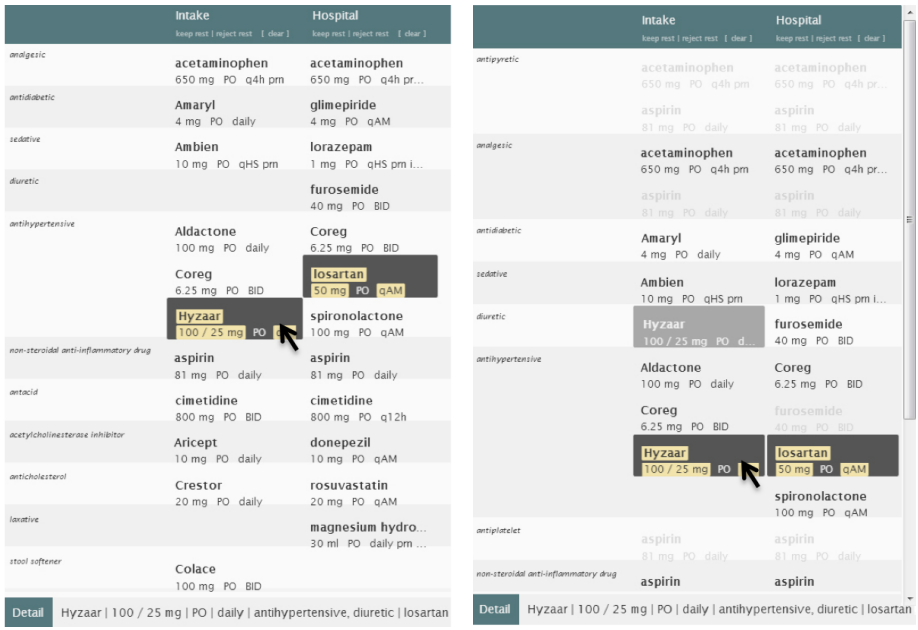


Figure 8. Two columns (Intake and Hospital). Initially (left) drugs are grouped by primary drug class, which naturally brings similar drugs close together, here showing a large group of antihypertensives. Highlighting reveals further similarities (e.g., when the user mouse overs Hyzaar they can see the similarity with Losartan). Optionally, we can show all classes, with additional ghost copies when drugs belong to more than one class.

Alternative design: Single column merged list

For reference, we contrast the current TwinList interface with an earlier design [229]. Figure 9 shows drugs in two merged lists: the unreconciled original list at the top and the reconciled list below. Identical drugs (white background) are automatically moved to the reconciled list. Remaining unreconciled drugs are displayed and color coded by similarity. Unique drugs are dark orange. Similar drugs are grouped pale orange, and drugs of form equivalence (brand vs. generic) are grouped with a white background. Whenever two drugs have the same dosage or other attribute, their table cells are merged (e.g., 25 mg dosage for Coreg and similar Carvedilol). Drugs that are unique are displayed as a bright orange color.

All information about a drug is visible in a wide row, however, this scheme makes it harder to tell which list a drug belongs. Instead of origin spatially separating drugs, a dedicated column provides the information.

To reconcile, users drag rows from the top list to the bottom. They can also change their mind and slide reconciled rows back. We found the method particularly effective on touch screens (e.g., tablets) where users are accustomed to tap and drag gestures. Drag rows with a mouse was slower and more error prone on PCs, becoming more difficult reconciling long lists. Merging more than two lists was possible, but multiple levels of similarity were problematic. Simply grouping all drugs lost details of drug connections. Grouping by class or indication was difficult because groupings were repeated in both lists.

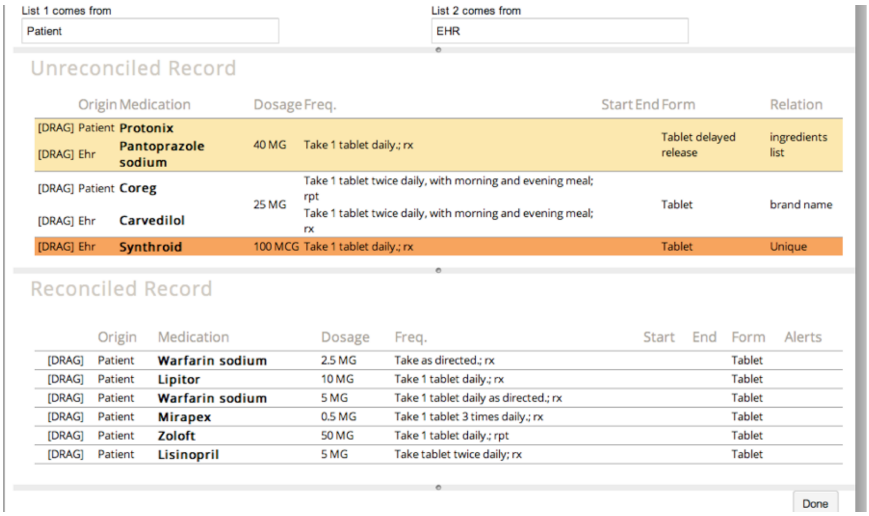


Figure 9: An alternate design of two stacked lists, unreconciled at top and reconciled below. Drugs are grouped by similarity. Color indicates type of similarity. Dragging rows from one list to the other indicates which drugs are to be kept.

Additional design considerations

When to use animation? While animation has been shown compelling and helpful revealing transformations in complex graphical representations (such as trees or graphs), other studies cast doubt on animation's usefulness in learning [230]. To gauge if animation was beneficial in TwinList, a pilot study was conducted with 20 participants comparing TwinList with multistep animation versus a direct jump to final layout [231]. The study found no significant difference in training time, but differences were observed in user comments and clarification

questions. Only 3 of the 10 participants who learned with the multistep animation reported being initially confused about the five-column layout, compared to 9 of 10 for those learning without animation. Fourteen out of 20 stated they favored learning with animation, citing its ability to "show you where everything goes" and how everything "connects." A paired t-test for a related survey question indicated full animation was considered more helpful for learning ($p = 0.02$). Seventy percent of participants ($n = 20$) preferred the full animation for initial learning. Ninety percent stated they would prefer to go directly to the final layout for regular use (i.e., after learning).

The danger of scrolling: In any user interface design, long lists may spill over a single screen. In reconciliation, scrolling may cause users to forget to take action on some of drugs. This led to our decision to keep the Sign off button inactive until a decision had been made regarding all drugs. Scrolling also may cause some drugs to be off screen when highlighting multiple drugs. For this, we added a popup notification at the edge of the screen. In Figure 10 a "…more (1)" prompts users to scroll. An alternative would be to temporarily animate/move the related information closer to the cursor.

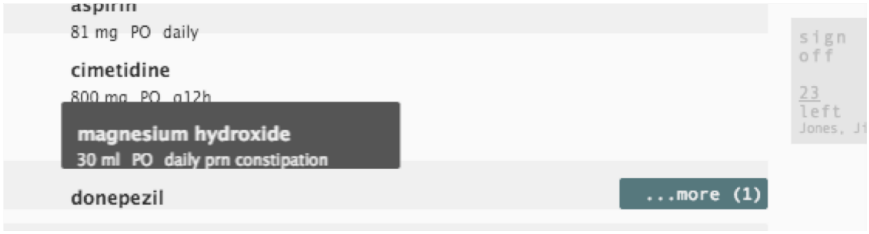


Figure 10. A box labeled "…more (1)" pops up from the bottom right when more information is available by scrolling (here signaling the drug magnesium hydroxide also appears in a different class below). This subtle use of animation draws users' attention.

Options and user control: The decision to group drugs by class or diagnosis can be left to users by providing easily accessible controls. In TwinList a single click on the top menu toggles grouping by class on and off, allowing rapid switching between two views. Keyboard shortcuts are also available (C for grouping by Class, M for multi-class, N for None).

Similarly, animation can be turned on and off. Drug name display can be changed from "as prescribed" to all brand name or all generic. User testing should decide which options should be on by default, or even what options are offered. Our prototype allows users to remove/hide medication from a list once a decision has been made. This makes progress visible as users see the list shrink, decreasing the need for scrolling.

Revealing similarities within the lists: While the role of preprocessing and spatial layout is to clearly indicate similarities between lists, TwinList can also show similarity within lists. When a drug has been prescribed twice, similar drugs within the list are highlighted in dark grey during mouseover events.

Automatic reconciliation—or not? For efficiency, it's theoretically possible to automatically reconcile some drugs. For example, interface designers could choose to automatically reconcile: 1) all identical drugs, 2) all unique intake drugs, or 3) all intake drugs. Each scenario provides modest gain, but still requires review by the prescribing clinician. Automatic reconciliation increases the chance of a patient accidentally placed on a medication that should have been held due to changes in clinical condition.

DISCUSSION

Many reconciliation interface designs are possible. Dr. Belden in his HIMSS 2013 talk suggested a separate column for grouping by diagnosis and highlighting to reveal linkages between drugs and diagnoses. Yet another option is reconciling drugs one group at a time, for example, by starting with large drug classes (e.g., all the antihypertensive medications in our earlier example). Faced with many options, EHR developers should design interfaces that match their product's overall design philosophy.

We trust further research will quantify the benefits of individual interface elements (animation, groupings, etc.) and guide the development of new designs. Continued research will help interface designers make better decisions, enabling healthcare workers to accomplish tasks more safely and efficiently.

Our work demonstrates the complexity and importance of designing HIT user interfaces that provide cognitive support to improved clinician speed and accuracy. To gain full benefit, effective interface design should be applied to all clinical tasks supported by EHRs.

We received positive feedback from two dozen clinicians, but acknowledge the need for further evaluation. Comments indicate animation was helpful and our groupings meaningful. This led to a pilot implementation of TwinList in Microsoft Amalga, an adaptation for problem list reconciliation at Massachusetts General Hospital, and several ongoing projects that added TwinList to existing EHR systems. A user study of speed and accuracy between the TwinList interface and baseline systems is underway.

ACKNOWLEDGMENTS

This work is supported in part by the Center for Clinical and Translational Sciences at the UT Health Science Center at Houston (NCATS UL1 TR000371) and Grant No. 10510592 for Patient-Centered Cognitive Support under the Strategic Health IT Advanced Research Projects Program (SHARP) from the Office of the National Coordinator for Health Information Technology.

A version of this chapter originally appeared in the Proceedings of AMIA 2013, 1150-1159, Plaisant, C., Chao, T., Wu, J., Hettinger, A., Herskovic, J., Johnson, T., Bernstam, E., Markowitz, E., Powsner, S., Shneiderman, B. (May 2013) TwinList: Novel User Interface Designs for Medication Reconciliation.

16: Guidelines for Ensuring Timely Management of Medical Orders

Sureyya Tarkan, PhD
University of Maryland

Lyndsey Franklin, MS
University of Maryland

Catherine Plaisant, PhD
University of Maryland

Ben Shneiderman, PhD
University of Maryland

ABSTRACT

Medical professionals issue and review many orders for patients, such as lab tests, imaging studies and referrals. Medical order management is a complex process. When it fails, treatment effectiveness, patient safety and satisfaction suffer. We developed design guidelines for rich tabular displays that: 1) show pending results, 2) prioritize by late and lost status, 3) clarify responsibility, and 4) embed actions. We found benefits applying these guidelines in a user study and application in a variety of other domains.

INTRODUCTION

> Sue injured her leg in a bad fall and visited her doctor. Her primary care physician ordered an X-ray, one of dozens of orders the physician wrote that day. Sue scheduled an appointment at an independent radiology center for the following day. A technician took images. The radiologist was supposed to review the images and write a report to be faxed to her primary care physician, but something went wrong. The physician never saw the results. Sue's primary care physician had Electronic Health Records, but did not notice

> Sue's test results missing. Her fracture was never treated. Sue lost her leg.

This is a real story. Many things can go wrong in a complex process with many steps, multiple actors and various responsible parties. Patients can be physically harmed if a lab test or imaging study is lost or mishandled, and patients can become sicker or die if referrals to specialists are lost [232]. Failure to follow up on abnormal test results is one of the most frequent causes of medical malpractice litigation in outpatient medicine [233]. Timely management of medical orders improves efficiency and effectiveness of treatment, patient safety, and overall satisfaction [234].

There are no standards regarding how best to manage medical results [235]. A study of contemporary test result management systems discovered interface and logic errors in routing, physician records, system settings, and system maintenance tools [236]. During our interviews, we saw environments where needed results were received in a timely and reliable fashion (e.g., emergency rooms in hospitals with all test facilities in house), while others reported high rates (20%) of late or lost results. We observed physicians keeping paper "cheat sheets," and staff and patients spending hours on the phone tracking missing results. Even when physicians have reliable systems, medical staff do not routinely check the status of all pending orders [236]. Better management of medical orders and test results could yield significant benefits.

Because lists are the most common way of managing daily work [237], we focused on interactive rich tables to indicate medical order progress. We developed a prototype of rich tables generated using the Multi-Step Task Analyzing, Reporting, and Tracking (MSTART) system from workflow models of *processes* (multiple steps associated with possible user actions) [238]. Rich tables were refined by conducting iterative design reviews with medical professionals. The result are guidelines to improve timely management of medical orders by using tables that: 1) show pending results, 2) prioritize orders and results by late and lost status, 3) clarify responsibility, and 4) embed actions. Items 1 and 2 encourage users' attention. Items 2 and 4 allow users to rapidly take action. Our guidelines extend Stephen Few's table design recommendations [239] and Microsoft's Common User Interface design

guidance [240] to medical systems. A user study was conducted to formally compare rich tables based on our guidelines to a common interface used for reviewing medical test results. We learned that our guidelines can help reduce the problem of missed results.

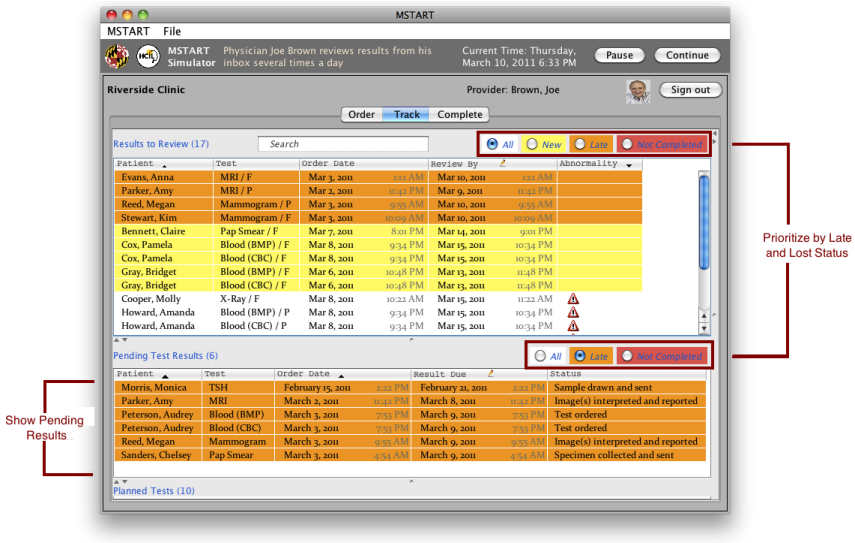


Figure 1. Rich tabular displays as seen by a physician. Rich tables adhere to our design guidelines. Hovering over rows reveals informational tooltips.

Related work

To the best of our knowledge, state-of-the-art medical order tracking is represented by Partners Healthcare *Results Manager* [241]. *Results Manager* is limited by not showing late results or impediments to taking rapid actions on them. Other clinical event notification systems [242] remind physicians to follow up on results, but research shows such systems can generate undesired alerts and cause clinician alert fatigue, potentially resulting in providers bypassing or missing important alerts [243]. Other researchers applied workflow management techniques to clinical situations [244] with models that assist optimizing or testing workflows in the context of an ideal environment. In contrast, we focused on what can go wrong in real world situations.

Many principles can be learned from well designed alert systems. Researchers have built tools to support organizing users' daily tasks and roles [237]. These systems generally display tasks as chronologically-ordered lists [245] and remind users about pending tasks. Users may

switch between screens to complete tasks, but such interruptions have been found to reduce awareness [246, 247]. In the medical domain, interruptions contribute to error risk. Information visualization research has applied situation awareness theory [248], but this requires users focus on the visualization to discover anomalous behavior. We found medical workflow to be more dynamic and time critical.

DESIGN GUIDELINES

We offer design guidelines based on shortcomings of existing EHR interfaces. For example, in many systems, physicians scan a list (either of all patients, or per patient), which serves as a reminder to review results. Pending orders are usually not visible unless a physician reads details of a patient's record or uses a reporting tool. In many EHR systems, physicians are forced to remember orders they have placed. Systems generally have no notion of latency between orders and results. Result lists may be sorted by arrival date, with newer results inserted at the bottom. A physician may have no way of knowing if an expected result is missing, but even if they do, their only option may be tracking it down via phone. Once results have been reviewed, there is often no mechanism to ensure followup.

Show pending results

Tables should provide access to results, pending orders, and planned orders, whether all patients or only one. Figure 1 shows an example of Dr. Brown's orders. Returned results are listed on top under "Results to Review," while orders that have not yet returned are shown under "Pending Test Results." Orders that have been issued, but were intended for the future (e.g., a mammogram on a patient's 50th birthday) are accessed under "Planned Tests" (shown here collapsed). Orders automatically move from planned to pending at the appropriate time. As results arrive, their entry moves to the topmost table. Results are removed once the physician has reviewed them and confirmed followup.

Prioritize by late and lost status

Our prototype employs an underlying result management workflow model that assigns normal and maximum durations to each step. For example, a patient may be given between three days and a week to

schedule and get an X-ray. We calculate normal and maximum expected durations for each step, although physicians can overwrite the normal duration if a rapid return is desired. An order exceeding normal expected duration (reflected in the "Result Due" date) is determined late and shown in orange (e.g., an X-Ray still being processed on the tenth day). After the maximum time has passed, the order is considered lost or not completed (shown in red) and may have to be repeated or cancelled. Time to complete each step can be logged for later retrospective analysis to discover bottlenecks, determine best and worst performers, or adjust normal and maximum expected durations.

Steps completed by physicians also have normal and maximum durations. The result list at the top has a "Review By" date. A physicians' work might be late (orange) or incomplete (red). Orders not yet reviewed are coded yellow, while those in white indicate they have been reviewed, but followup not completed. Color acts as a filter. The Pending table shows only severe (late and lost) cases with lateness information available to all users of the system. For example, clinic managers can track if physicians follow up their orders on time and can forward results to alternative physicians if needed (e.g., in case of physician illness). Due dates can be modified directly in the table if needed.

Results are sorted first by lateness, second by review status, third by abnormality (shown with a warning sign), and finally by patient name, which groups results. Pending orders are sorted by lateness, then by patient name.

Clarify responsibility

Pending orders have a column for order status (see Figure 1) indicating the last completed step. A click on the row pops up a menu (Figure 2) showing who is responsible for progress of the order and its expected completion. For late orders, there is information on who to call to speed up the process. When an order is considered lost, a reorder button appears.

The popup reveals completed steps in chronologically descending order. The first step of each pending order is the patient scheduling an exam, making the patient the first responsible person (see the last item under "Completed steps" in Figure 2). The next step involves an outside

facility processing the order (see the first two bullets under "Completed steps" in Figure 2). The letter 'P' or 'F' in the results table distinguishes preliminary from final results (Figure 1). These results appear in both Pending and Results tables because the order has not finished processing (the outside facility responsible is still finalizing the report).

Embed actions

While some test results require careful review in separate screens (e.g., patient history), there are many situations where action can be taken directly, for example, when test reports come back normal. We allow users to take immediate action within the results list. Possible actions depend on the role of the logged-in user. When physicians or residents click a result, the report and simple follow-up actions appear side-by-side and below the row (Figure 3). If more information is needed to deal with a complex case, a double-click opens the patient record. In other cases, a panel of common actions is accessible and remains on screen until users indicate they either need to return for further review (the result remains in the list and is colored white), or that the followup is complete (moving the result from the list into the "Complete" panel in Figure 1).

USER STUDY

We undertook a within-subjects study to quantify benefits of our guidelines and approach. Eighteen study participants took on the role of physicians and answered questions about the timeliness of orders using three interface variations. The baseline interface consisted of a single list of chronologically ordered results. A second interface added a separate list for pending orders. A third interface prioritized pending orders by lateness. Interface presentation order was counterbalanced and randomly assigned. Participants were given five minutes to read and remember a list of twenty patients orders with normal and maximum durations to complete different order types. Participants were then asked to perform a distraction task for five minutes (so that they did not remember details). After a short introduction to the interface, participants were asked to identify which orders were late (i.e., longer than normal) and which were lost (exceeding procedure time limits). We recorded the time to arrive at the correct answer and the number of corrections participants had to

make. In addition to $10 compensation, a bonus $10 was offered to best performers in each interface.

Results (Figure 4) suggest showing pending results can decrease the time needed to answer questions by more than a half and that prioritization of results helps even more. We ran a repeated measures one-way ANOVA (three treatment levels) with pairwise comparisons using the Holm adjustment method. Differences were statistically significant ($p < 0.01$) and post-hoc paired t-tests established differences between interfaces: baseline to second ($p < 0.01$), baseline to third ($p < 0.01$), second to third ($p < 0.01$).

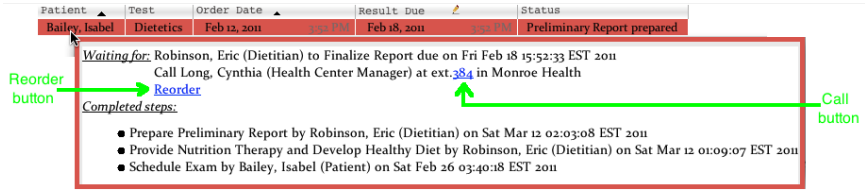


Figure 2. Popup menu for pending orders.

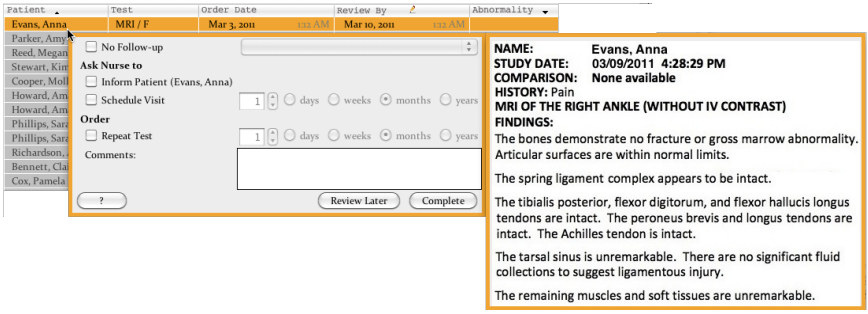


Figure 3. Interactive action panels facilitate rapid completion. Results appear on the right.

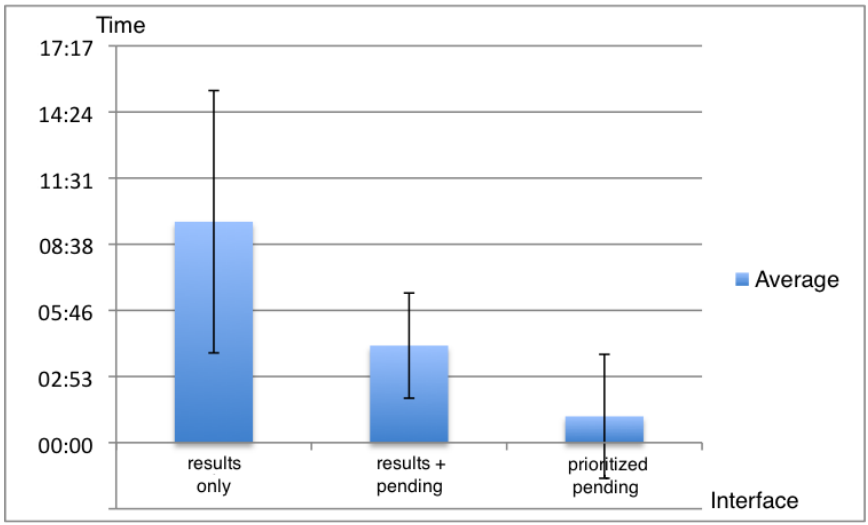


Figure 4. Results of our user study (eighteen participants).

DISCUSSION

Physicians provide better care for patients when they manage test results promptly. EHR systems should report delayed orders and effectively guide clinicians to act. Our results suggest applying these design guidelines can improve timely management of medical orders.

These guidelines are generalizable to tracking interfaces built for other processes where parties separated by time and distance collaboratively handle different steps. Examples include, but are not limited to, software development cycles, paper-review process in academic journals, and business processes such as return merchandise authorizations.

There are limitations to our study. It was difficult to simulate a real environment in an hour-long study. Our distraction task separated ordering from the reviewing step, but did not include distractions that may occur during order or review time. We believe our study's results could have been more prominent if there were more distractions, although this may have overwhelmed participants. Although we offered a prize to increase motivation, participants did not develop strategies to remember orders. As the study progressed, participants seemed to realize attempting to memorize orders was useless, and ultimately gave up—

which confirmed our finding of physicians using cheat sheets to remember orders.

Conclusion

Missed medical test results and lost referrals are serious problems. We propose design guidelines can ensure timely management. The results of our user study confirm better designs can have a dramatic effect on performance. Clarifying responsibility and embedding actions in rich tables can further reduce the problem of missed results. We believe our guidelines can offer improvements to similar workflows. We are currently working on interfaces that facilitate retrospective analyses of performance data to identify bottlenecks, best and worst performers.

Acknowledgements

This work was supported in part by Grant No. 10510592 for Patient-Centered Cognitive Support under the Strategic Health IT Advanced Research Projects Program (SHARP) from the Office of the National Coordinator for Health Information Technology. For more information: www.cs.umd.edu/hcil/sharp.

We thank our collaborators, Todd Johnson, Jorge Herskovic, Elmer Bernstam, Zach Hettinger, and Seth Powsner, for their feedback and suggestions on the project. We also thank Nicholas Chen for his help with the paper.

A version of this chapter originally appeared in Tarkan S, Franklin L, Plaisant C, Shneiderman B. Design Guidelines for Ensuring Timely Management of Medical Orders. University of Maryland Human-Computer Interaction Lab, 2012 HCIL-2012-25. Retrieved from http://hcil2.cs.umd.edu/trs/2012-25/2012-25.pdf.

Clinical Design Support

The *C* in SHARPC stands for *cognitive*. SHARPC considers an EHR has cognitive support if the system was designed for problem solving and decision making to achieve the highest quality of care measured by the Institute of Medicine's six dimensions of quality: safe, effective, timely, efficient, equitable, and patient centered. SHARPC researchers studied problems challenging cognitive decision support and developed knowledge bases (Chapter 18), models and ways to summarize complex, chronically-ill patients' electronic health records (Chapter 17).

Improving problem list accuracy, critical for patient care and decision support, was also studied (Chapter 19). SHARPC created ways that decision support rules can be formalized, shared, and customized for local use. Research led to the development of an authoring tool for knowledge engineers and subject matter experts (Chapter 20). Chapter 21 describes an application for diagnosing systemic inflammatory response syndrome that demonstrates how a deep understanding of expert clinicians' cognitive processes can be transformed into a practical EHR tool.

17: Clinical Summarization

A model-based technique

Dean F. Sittig, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Allison B. McCoy, PhD
Tulane University School of Public Health & Tropical Medicine

Adam Wright, PhD
Brigham & Women's Hospital

ABSTRACT

We developed model-based techniques to automatically generate clinical summaries from complex electronic health record (EHR) data. Research included studying how clinicians collect, distill, interpret, and synthesize patient information. To organize our work, we developed an eight-dimension socio-technical model of safe and effective EHR implementation and use. We also developed AORTIS (Aggregation, Organization, Reduction and Transformation, Interpretation and Synthesis), a six-stage model of data summarization. We used the socio-technical model to explore clinician utilization of EHRs in various ambulatory settings and the AORTIS model to evaluate the clinical summarization capabilities of several EHR products. We then developed prototype clinical summarization displays using the Harvard SMART platform. This research provides a theoretical and practical foundation for future work in computer-generated clinical summarization.

INTRODUCTION

The ability of clinicians to collect, distill and accurately interpret patient information is critical. Clinicians often face volumes of data from a variety of sources and strain to separate important information from background noise. They must also condense and refine information to communicate with colleagues in the course of providing continuous and coordinated care [249, 250]. The way information is structured and presented to clinicians can profoundly influence their decision making [251]. Medical information is often fragmented, existing in a wide range

of locations and formats, which puts patients at an increased risk of errors, adverse events and inefficient care [252]. An accurate, well-designed, context-specific summary could save time, improve clinical efficiency and mitigate errors.

Clinical summarization can be defined as collecting, distilling, and synthesizing patient information to facilitate a range of clinical tasks. Discharge summaries, daily progress notes, patient handoffs at shift change and oral case presentations are common. We narrowed the definition to healthcare provider efforts that result in patient-specific clinical data to assist communication and decision making. This differs significantly from the concept of *text summarization,* which broadly refers to creating a text summary from one or more source documents (e.g., scientific articles, literature abstracts, and multimedia).

While some aspects of clinical summarization have become easier through electronic health records (EHRs), other aspects are now more complex. Clinicians collect and process enormous amounts of clinical data, creating the potential for information overload and error [243, 253]. This overload can cause frustration, inefficiency and communication failure [254], leading to important data being overlooked [174]. Problems will likely increase with health information exchanges (HIEs), which allow broad sharing of patient data. Poor presentation of clinical information can also impair medical decision making, contributing to medical errors and reduced quality of care [255].

Readily accessible and perfectly legible clinical notes, laboratory results, images, and provider correspondence inform clinical care [256], but pose a challenge for time-pressured clinicians working in busy settings [243]. Early EHR adopter organization patients may have accumulated more than ten years of electronic health data. As the number of clinicians using EHRs increase and HIEs capable of exchanging patient-level data expand, the quantity of data that clinicians need to review for safe and effective care will exponentially grow. Clinicians must integrate these data with their medical knowledge and the scientific literature, then integrate this knowledge with their institution's policies and government regulations. Solutions require explicit, unified, accurate, and comprehensive patient-centered models that reflect the true work domain ontology [257].

Automated methods of using patient-centered knowledge to summarize and display patient clinical information are needed. The ultimate goal of our research are knowledge models and knowledge bases that produce clinical summaries as clear and concise as Figure 1.

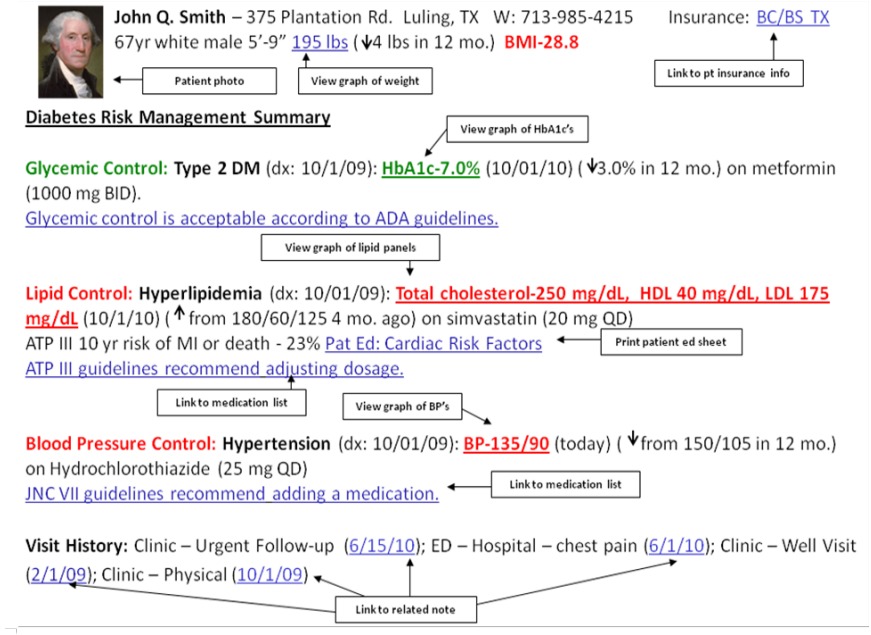


Figure 1. Example of automatic clinical summarization (problem-oriented view). Reprinted with permission from J Biomed Inform. 2011 Aug;44(4):688-99.

Clinical summaries can be divided into three interrelated categories: *source-oriented*, *time-oriented* and *concept-oriented* views [258]. Source-oriented views derive from traditional paper charts where information is filed in separate categories to facilitate document retrieval [259]. Source-oriented views are part of most EHRs. Information is organized according to source, allowing data to be grouped into categories, such as laboratory results, imaging studies and medication. Time-oriented views organize information based on when it was collected and order data chronologically in normal time or reverse time (most recent first). Information may delineate a sequence of events or details of a care plan and are common in both paper and electronic records. In a concept-oriented view, data is organized around specific clinical concepts, such as medical problems or organ systems. Using a concept-oriented view requires significant clinical knowledge (physician expertise or

computerized knowledge database), but can speed information retrieval and improve medical decision making [260, 261]. Each view, alone or in combination, is a valuable way to analyze patient data across a range of clinical tasks.

Our first challenge was identifying data and methods to model a clinicians' desired interactions with patient medical history, then summarizing patient histories and identifying appropriate actions to improve clinician decision making. Our second challenge was designing automated methods to create accurate, succinct, condition-dependent and independent computer-generated summaries of patients to improve patient safety, clinician efficiency and satisfaction, and reduce cost of care.

APPROACH

Our work was divided into three parts. Part 1 identified the data and methods clinicians use to make sense of vast clinical information contained in state-of-the-art EHRs. For this, we developed a form of ethnography called the Rapid Assessment Process (RAP). Part 2 developed a theoretical framework to study the data summarization process. Part 3 used clinical knowledge bases (described in Chapter 18) to create prototype clinical summarization displays.

Part 1: The rapid assessment process

RAP methods were used to understand user needs for summarization. RAP is a modification of traditional ethnography's ability to address "how" and "why" [262]. Often used in international and public health settings, we modified the approach to study different clinicians (e.g., primary care, sub-specialists, hospitalists) using EHRs in different settings (e.g., clinics, emergency rooms, hospitals). RAP consists of: 1) selecting sites and participants to maximize what is learned in the shortest time, 2) using local collaborators to identify clinicians to interview and observe, 3) collecting data using multiple observers and methods (e.g., observation, think-aloud, focus groups, surveys, task analyses, document analysis), and 4) collaborative, structured data analysis.

Part 2: A Theoretical framework for clinical summarization

We developed a conceptual framework based on existing summarization theories and clinical summary real-world use. We characterized tasks inherent in clinical summarization and the structure and function of clinical summaries to:

1. Provide a common framework applicable to clinical summaries of different types (narrative vs. structured) and uses (e.g., discharge summary, patient handoff);
2. Describe a method of analyzing human and computer-generated summaries;
3. Facilitate standardization or automation of clinical summaries;
4. Encourage future research on clinical summarization.

The creation of clinical summaries can be modeled in five steps: Aggregation, Organization, Reduction and Transformation, Interpretation and Synthesis (AORTIS). Any or all of these steps could be performed by a clinician or automated system to produce concise and accurate summaries.

The AORTIS model is sequential (Figure 2). Output from one step is input to the next and varies based on the clinical task the summary supports. Not all steps are of equal importance or apply to every summarization scenario. For example, if only one data type is aggregated (e.g., weight), there may be little need to organize the data (e.g., sort by time/date) before reducing (e.g., finding the most recent or current value or the maximum, minimum or mean) or transforming (e.g., graphing). A step may be bypassed, with data from aggregation flowing directly to reduction and transformation. The model can also terminate early. For example, aggregating and organizing lab results may be useful even without reducing, transforming, interpreting or synthesizing.

Aggregation

Aggregation is the collection of data from various sources. Clinical data may exist in both paper and electronic formats, or in multiple databases and multiple care sites in organizations with EHRs. Types of data include numerical (e.g., laboratory results), structured and/or coded text, (e.g., problem lists), and unstructured free text (e.g., progress notes).

Data aggregation may be accomplished by a clinician and facilitated by electronic tools (e.g., a lab results review module).

An aggregation example is collecting a patient's LDL cholesterol results over ten years. Accomplishing this basic task of aggregation is relatively straightforward if the patient received care primarily at one location with an integrated EHR (e.g., a VA healthcare facility). The task becomes much more difficult if the patient moved or changed providers, with data in multiple places and/or under multiple naming conventions. After aggregation, clinical data is often available in excess and difficult to interpret. Difficulty increases as the amount of stored information grows.

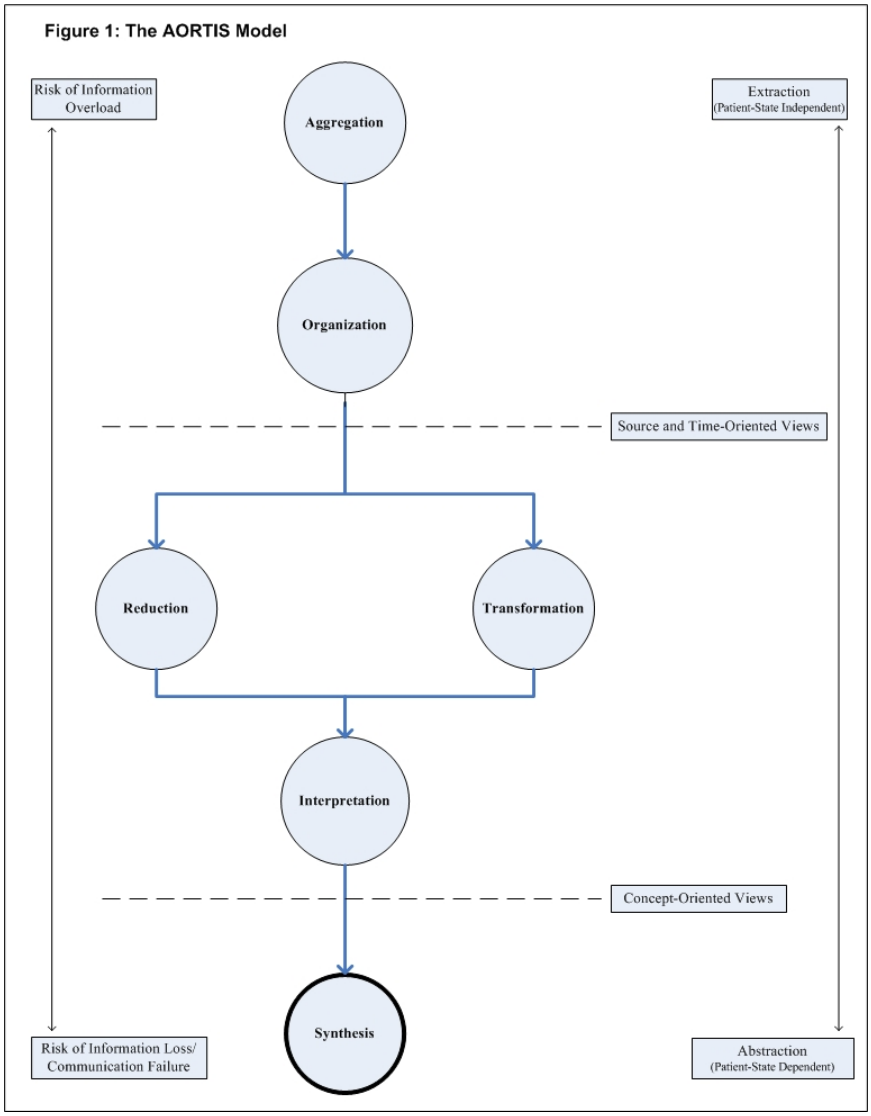


Figure 2. Illustration of the AORTIS Model of clinical summarization.

Organization

Aggregation and organization are distinct stages, each with unique challenges. Organization is the structuring of data according to a principle without condensing, altering, or interpretation. Two common organization operations are grouping (e.g., putting all HbA1c values together) and sorting (ordering lab results by date or value). When using paper charts, organization typically occurs following aggregation, but in

an EHR can occur near-simultaneously to aggregation. Realistically, most patient information must progress to the organization stage to be of clinical value.

LDL cholesterol test results could be sorted chronologically, by value, or grouped by laboratory of origin (hospital, PCP, previous PCP, VA, etc.). Both source and time-based LDL value organization aids clinicians' understanding aggregated data. In paper records, views must be manually created by clinicians or administrative staff. Physical properties of the record may be designed to accomplish time-oriented organization (e.g., an hourly flow sheet, space for entering daily progress notes), but manual organization can be a lengthy process. Electronic systems, in contrast, can organize the same data almost instantaneously.

Reduction and Transformation

Clinicians are vulnerable to information overload without further data processing after organization. Data condensation can occur by reduction or transformation. Reduction culls salient information from the database without altering it to decrease the amount of data presented. For numerical information, this might include a selection of the most recent values, maximum values (i.e., medication peak levels), minimum values (i.e., medication trough levels), or statistical reductions such as medians. For text-based information, this might include selecting results or notes over a certain timeframe or category (e.g., endocrinology consult notes, radiology reports, all notes that mention the term "back pain" or the "assessment" section of all progress notes).

Transformation is the process of altering a data view or data density to facilitate understanding. A simple form of transformation is trending: the qualitative description of a basic pattern in data (e.g., transforming an array of HbA1c values to the statement, "the patient's HbA1c level decreased 29% (from 8.6 to 6.2) over one year"). Another example is the graphical display of laboratory results (e.g., HbA1c levels) over time. In this transformation, values are translated from numeric representation to spatially-oriented displays. Transformation can also be accomplished using other visual tools, such as metaphorical graphics overlaid on a schematic diagram of the human body or timeline [208, 263].

Reduction and transformation require less contextual clinical or general scientific knowledge than interpretation or synthesis. For example, HbA1c is a numeric value. It's fairly simple to find the mean HbA1c by summing all values and dividing by the number of values summed. Another example: qualitative urine human chorionic gonadotropin is a discrete text value. Finding a true arithmetic mean value is not possible. Theoretically, a median or modal value could be calculated, although the clinical significance of an average value would depend on the nature of the test and its clinical context. In the absence of high-level summarization, reduction and transformation are tools for producing extracts because these steps do not depend on patient-specific information.

Continuing the LDL example, reduction might be used to create an extract of a patient's LDL by reporting the most recent or maximum and minimum results for a given time period. Transformation might yield a line graph of available values over time or a description of the data's trend (which is also considered an extract).

Interpretation

Interpretation is context-based analysis of a single type of clinical data using general (versus patient-specific) medical knowledge. For example, selecting abnormal lab results to include in a patient handoff summary requires interpretation by a clinician or computer program to identify which results are abnormal. Many lab result reports include an indication of abnormally high or low results made by a computer using a knowledge base of abnormal and critical ranges. This is an example of simple interpretation because it applies general medical knowledge to a single data type for a specific patient.

Interpretation requires a clinical knowledge base. Despite progress in machine learning, artificial intelligence, expert systems, natural language processing and clinical decision support, interpretation beyond abnormal flags and reference ranges remains largely in the hands of clinicians. However, for the purpose of accomplishing a highly specific clinical task, automated high-level systems can be created, such as computer-assisted acid-base interpretation [264] and EKG interpretation [265].

Interpretative elements can be added to transformed data. For example, one could add a text alert indicating recent changes, such as "LDL level has increased over the past year and now exceeds goal level." Horizontal lines showing the limits of the normal range could be added to a graphical display, thus facilitating visualization that a patient's results are outside normal limits. Both require general medical knowledge to define "goal levels."

Synthesis

The final phase of AORTIS is combining two or more data elements with knowledge-based interpretation of patient state to create meaning or to suggest action. Synthesis is the most sophisticated and valuable form of clinical summarization because concept-oriented views are possible. Following knowledge-based interpretation, clinical information can be understood in relation to other parts of the medical record and can be viewed with respect to the patient's unique clinical status. Synthesis depends heavily on previous steps to create a reliable and complete clinical information summary.

When one interpreted piece of information (such as an abnormal lab result) is synthesized with other types of patient information (such as medications used to treat the condition), more sophisticated meaning is generated. For example, a simple synthesis of LDL results might yield the statement, "In response to elevated LDL levels on 12/01/09, a statin was initiated and LDL levels decreased to normal on 2/1/10." This synthetic statement brings together the identification of an abnormal value with pertinent medical history and succinctly provides a rich array of patient information. The statement, thus, captures: 1) a previous abnormal LDL on December 1st, 2) the now normal LDL on February 1st, 3) the initiation of a lipid-controlling medication, 4) the downward trend in LDL, 5) the implication that medication helped lower LDL, and 6) the impression that the patient's hyperlipidemia is well-controlled with medication.

Part 3: Development of clinical summarization prototypes

We developed a patient summarization application using the Harvard University SHARP project's Substitutable Medical Apps Reusable Technologies (SMART) platform [266]. The app is based on

our proof-of-concept problem-medication linkage SMART app based on a National Drug File – Reference Terminology knowledge base [267]. The summarization app can be run in any EHR environment that either supports SMART or runs the SMART-enabled i2b2 clinical bridge [268]. The bridge demonstrates a pathway for reusable app development that does not require EHR vendors to immediately adopt the SMART API. Apps can be developed in SMART and run by clinicians in the i2b2 repository, reusing clinical data extracted from EHRs.

We modeled our summarization app's user interface on a previously designed OpenVista prototype interface of a problem-oriented view. The interface was evaluated using the TURF framework (see Chapter 2) for EHR usability [7, 269]. We developed the app using HTML and JavaScript, the Bootstrap front-end framework (http://getbootstrap.com/) and Google Visualization API [270]. Our original proof-of-concept SMART app showed all problems and medications on one screen, which was unwieldy for complex patients, and displayed output in a rigid HTML table. The new app featured a cascading style sheet and fluid grid design to ensure proper proportions for key screen resolutions.

Results

Results from the clinical summarization project are divided into the same three parts as our Approach.

Part 1: The rapid assessment process

We used RAP in clinical settings at The University of Texas professional practice plan, Baylor College of Medicine faculty practice plan, Houston VA ambulatory clinics, and Brigham & Women's Hospital Ambulatory clinics. We found:

1. The majority of patient problems were not recorded using structured clinical vocabulary terms in the EHR section commonly known as "problem list." This early finding caused us to devote significant effort developing new ways to infer patient clinical problems from structured and unstructured EHR data (Chapter 19).
2. Understanding the roles health information technology plays in clinical care and patient summarization is challenging. We developed

and refined an eight-dimension socio-technical model to help us study the design, development, implementation, use, and evaluation of HIT within complex healthcare systems (Figure 3) [271].

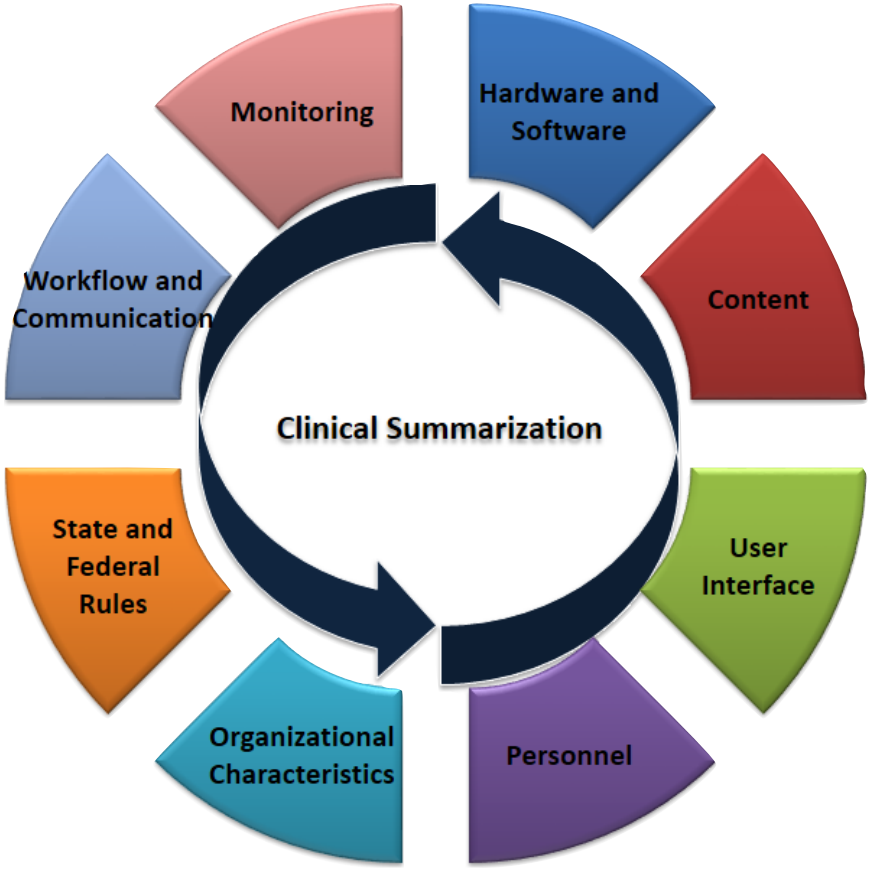


Figure 3. The eight-dimension socio-technical model of safe and effective EHR implementation and use applied to Clinical Summarization.

3. Clinicians face numerous social, legal, ethical, and financial issues on a daily basis that have the potential to affect their usage of HIT and EHRs. One example: "…key legal dilemmas that must be addressed in the near-term pertain to the extent of clinicians' responsibilities for reviewing the entire computer-accessible clinical synopsis from multiple clinicians and institutions, the liabilities posed by overriding clinical decision support warnings and alerts, and mechanisms for clinicians to publicly report potential EHR safety issues. Ethical dilemmas that need additional discussion relate to opt-out provisions

that exclude patients from electronic record storage, sale of de-identified patient data by EHR vendors, adolescent control of access to their data, and use of electronic data repositories to redesign the nation's health care delivery and payment mechanisms on the basis of statistical analyses. Finally, one overwhelming financial question is who should pay for EHR implementation because most users and current owners of these systems will not receive the majority of benefits." [272].

4. Lack of an evidence-based definition of EHR-related errors is holding back progress toward a safe and effective EHR-enabled healthcare system. We created our own definition: EHR-related error occurs anytime HIT is unavailable for use, malfunctions during use, is used incorrectly by someone, or when HIT interacts with another system component incorrectly, resulting in data lost or incorrectly entered, displayed, or transmitted [273].

5. There is emerging evidence of EHR-related safety events [274]. Unfortunately, no national program to facilitate the collection, analysis, or investigation of these events exists. We proposed the creation of a national EHR oversight program for dedicated surveillance of EHR-related safety hazards and to promote learning from errors, close calls, and adverse events [275].

6. Despite considerable progress in the adoption and use of EHRs following the US government's 2009 Health Information Technology for Economic and Clinical Health Act [276], EHR adoption has resulted in larger than expected challenges in day-to-day clinical processes. For example, many clinicians perceive the *total* cost of EHR (money spent on hardware, software, and consulting services *plus* the additional time required to complete orders, notes, and billing, *plus* required changes to their workflow) outweighs direct benefits. Still, they acknowledge patients and payers are likely to benefit from EHR use. We hypothesized that by providing clinicians with assurances EHRs will deliver the features and functions they need and that the regulatory environment will support them, would improve EHR adoption rates. We, therefore, developed a set of rights and responsibilities for EHR users [277]. Following publication of a

first manuscript, we were asked to develop an additional set of rights for clinicians caring for children [278].

7. At a time when health care organizations (HCOs) are focused on "meaningful use," we believe clearer guidance should be provided (to both clinicians and HCOs) to better align patient safety activities with those required to support a safe EHR-enabled health care system. We developed EHR-specific safety goals modeled after the Joint Commission's National Patient Safety Goals [279] to provide HCOs with focus areas for sustained improvements in organizational infrastructure, processes, and culture as they adapt to new, state-of-the-art health information technology.

Part 2: A theoretical framework for clinical summarization

We compared different aspects of 12 ONC certified EHR systems' general clinical summary screens using our AORTIS model [280] (Table 1).

EHR Product	Version	Implementation Site	Type of system
Partner's LMR	Fall 2010	Partners Healthcare System, MA	Locally developed
Allscripts Enterprise	v11.2.0	UTHealth Practice Plan, TX	Commercially available
CPRS	v1.0.27.90	VA Houston, TX	Freely available
GE Centricity	2008 version	University of Medicine & Dentistry, NJ	Commercially available
OCW	v1.9.802	Oschner Clinic, LA	Locally developed
StarPanel	N/A	Vanderbilt Practice Plan, TN	Locally developed
Springcharts	v1.6.0_20	Web demo	Commercially available
OpenMRS	v1.7.1	Web demo	Open Source; Freely available; Disease-specific (HIV/AIDS)
Cerner	v2010.01	Stonybrook, NY	Commercially available
ClinicStation	v3.7.1	MD Anderson, TX	Locally developed; Disease-specific (Cancer)
NextGen	Early 2008 version	Mid-Valley Independent Physician's Association, OR	Commercially available
Epic	v 2009 IU7	Harris County Hospital District	Commercially available

Table 1. A complete listing of the EHR systems chosen is included.

We found a wide variation in EHR product clinical summarization capabilities. For example, all EHRs were capable of simple aggregation and organization of clinical data, but only one demonstrated an ability to synthesize information from the data.

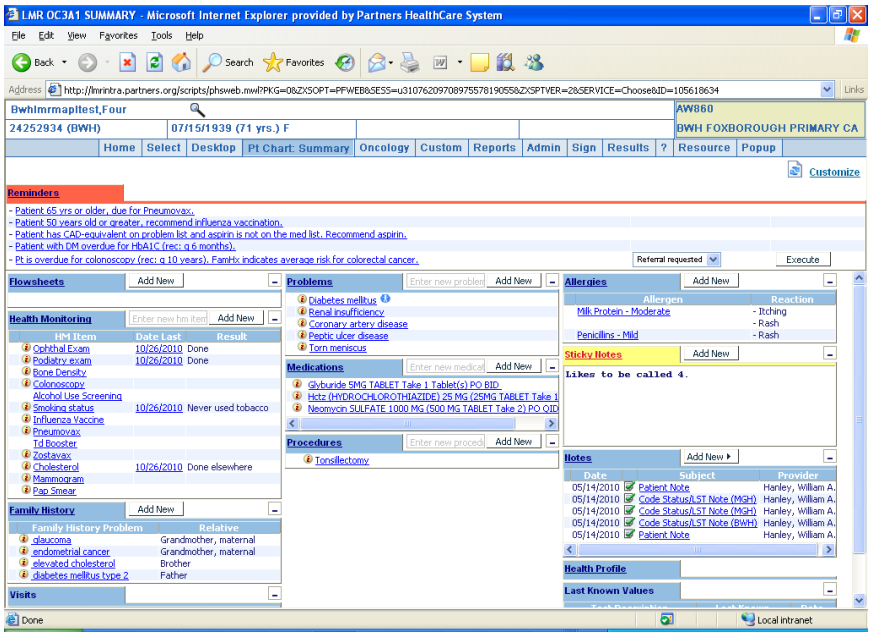


Figure 4. Screen print of Partners HealthCare System's Longitudinal Medical Record (LMR) clinical summary screen. Used with permission.

Part 3: Clinical summarization prototype

We partnered with the Harvard SHARP team [281] to test prototype clinical summarization displays using SMART. Our clinical summarization SMART app user interface displays a list of active problems on the left. Users may select a problem from the list and display associated medications on the right. Users can also click the "All Medications" text to toggle a list of all prescribed medications for a patient. We have not yet integrated a knowledge base with lab results, so the app displays all historical lab results and vital signs below problems and medications. Users may click a lab result or vital sign to toggle the values display. Any lab result with multiple values is shown as a graph, generated using the Google Visualization API. (See video at: http://www.i-jmr.org/article/downloadSuppFile/2454/6210. The app is open-source and available as a free download [282].

Figure 5. The patient summarization app running inside the SMART-i2b2 container. Shown here: urinary incontinence is highlighted and a relevant medication (oxybutynin) is displayed to the right; lab results are shown as line graphs below. Used with permission from: Interact J Med Res. 2013 May 30;2(1):e11. doi: 10.2196/ijmr.2454.

DISCUSSION

Developing automated methods to accurately and succinctly summarize a patient's clinical history from the vast amount of structured and unstructured data in an EHR system remains one of the "grand challenges" in clinical decision support [283]. It's also one of the most important problems to be solved if we are to provide safe and effective EHRs to all clinicians [262]. We believe we have made considerable progress understanding and solving this problem. For example, using our RAP methodology, we were able to identify socio-technical barriers that clinicians face. Our eight-dimension socio-technical model of safe and effective EHR implementation has proven useful in several HIT-related venues:

- Analysis of EHR-related safety concerns [170].
- Comparison of comparative effectiveness research platforms [284].
- Analysis of Medicare's bundled payments demonstration project [285].
- Evaluation of a new tablet-based, clinical data collection system for use in rural India [286].
- Evaluating the effectiveness of EHR-based clinical referrals [287].

- Understanding the management of electronic test result notification in the out-patient setting [288].

Our AORTIS clinical summarization model has been used by outside research groups to evaluate a prototype clinical documents visualization tool [289] and in-patient clinical documentation system for physicians [290]. The model illustrates steps to enable robust clinical data synthesis, bringing multiple data elements together to allow clinicians to rapidly process clinical information. AORTIS provides a roadmap to guide clinicians to information contained in patient records. Although clinicians are well suited to completing final high-level steps in summarization, it's possible that with further study of clinical cognition and workflow automated tools could support high-level summarization steps across a broader range of clinical tasks.

We demonstrated how large informatics projects could collaborate at a distance using freely available, open-source tools to develop a working prototype. This required agreement on syntax, software architecture and clinical content.

The burden for developing innovative clinical data entry and displays has fallen largely on EHR vendors. A significant amount of EHR vendor time, effort and money over the last four years has gone toward meeting Health and Human Services' meaningful use requirements [114]. As a result, we found collaborating with commercial EHR vendors to design and develop the next generation of EHRs difficult. Instead, we used the Harvard SHARP team's SMART tool for testing clinical knowledge bases against real-world patient data. Our clinical summarization prototype user interfaces demonstrated the utility of these knowledge bases. SMART has several important advantages:

1. SMART shortens the learning curve of app development by leveraging existing Web standards (e.g., JavaScript Object Notation data structures and Web service interfaces).
2. The SMART API is a straightforward data model designed to meet the needs of app development without trying to solve all use-cases for external clinical data views. This avoids more complicated formats such as the Clinical Document Architecture, a health care data standard for representing all types of clinical data.

3. SMART's current read-only approach will be extended with methods to write data back to the record. SMART enables clinical app innovation by giving developers access to clinical data elements on individual patients, complemented by data analytical platforms such as i2b2 (for aggregate, research-oriented data repositories and reporting).

Future Direction

We plan to use large clinical databases from multiple organizations to improve knowledge base accuracy, leading to higher quality automatically-generated patient clinical summaries.

Suggested Reading

Sittig, D. F., & Singh, H. (2010). A new sociotechnical model for studying health information technology in complex adaptive healthcare systems. Qual Saf Health Care, 19 Suppl 3, i68-74. doi: 10.1136/qshc.2010.042085.

Sittig, D. F., & Singh, H. (2011). Legal, ethical, and financial dilemmas in electronic health record adoption and use. Pediatrics, 127(4), e1042-1047. doi: 10.1542/peds.2010-2184.

Sittig DF, Singh H. (2012). Rights and responsibilities of users of electronic health records. CMAJ, Sep 18;184(13):1479-83. doi: 10.1503/cmaj.111599.

Sittig, D. F., & Singh, H. (2012)., Longhurst C. A. (2013). Rights and responsibilities of users of electronic health records. Cmaj, 184(13), 1479-1483. (EHR) users caring for children. Arch Argent Pediatr, Dec; 111(6):468-71. doi: 10.1503/cmaj.1115991590/S0325-00752013000600003.

Laxmisan, A., McCoy, A. B., Wright, A., & Sittig, D. F. (2012). Clinical Summarization Capabilities of Commercially-available and Internally-developed Electronic Health Records. Appl Clin Inform, 3(1), 80-93. doi: 10.4338/aci-2011-11-ra-0066.

Klann JG, McCoy AB, Wright A, Wattanasin N, Sittig DF, Murphy SN. (2013). Health care transformation through collaboration on open-source informatics projects: integrating a medical applications platform, research data repository, and patient summarization. Interact J Med Res, May 30;2(1):e11. doi: 10.2196/ijmr.2454.

Portions of this chapter are adapted from Feblowitz, J. C., Wright, A., Singh, H., Samal, L., & Sittig, D. F. (2011). Summarization of clinical information: a conceptual model. J Biomed Inform, 44(4), 688-699. doi: 10.1016/j.jbi.2011.03.008. http://www.j-biomed-inform.com/article/S1532-0464(11)00059-1/abstract

18: Developing Knowledge Bases for Automated Clinical Summarization

Making electronic health records smarter

Allison B. McCoy, PhD
Tulane University School of Public Health & Tropical Medicine

Adam Wright, PhD
Brigham & Women's Hospital, Harvard Medical School

Dean F. Sittig, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

ABSTRACT

Finding relevant patient information in electronic health records' (EHRs) large datasets is difficult, especially when organized only by data type and time. Automated clinical summarization creates condition-specific displays, promising improved clinician efficiency. However, automated summarization requires new kinds of clinical knowledge (e.g., problem-medication relationships). We studied eight problem-medication pair knowledge bases using six different approaches. A standards-based ontology knowledge base created the largest number of pairs (33,894,415). A reputation metric knowledge base was the most limited (983 pairs). Further research is needed to better understand knowledge bases for automated EHR data summarization.

INTRODUCTION

Electronic health records (EHRs) contain voluminous data of many types: visits, problems, allergies, notes, laboratory test results, diagnoses, medications, health maintenance items, etc. The amount of information can overwhelm clinicians, leading to frustration, inefficiency and errors [16, 17, 95, 174, 255]. Automated clinical summarization has the

potential to alleviate the problem of too much information [277, 291], but few EHRs have implemented required components, in part due to difficulties developing knowledge-based tools [280].

Automatic clinical summaries require knowledge of data relationships. There are a number of approaches to developing knowledge bases for EHR summarization, each with advantages and disadvantages. Some approaches are manual and require expert clinician review. Others are automated, relying solely on computational methods. Manually created knowledge bases can be highly accurate, although time consuming to create. They may also be incomplete. Automation requires little clinician time, but may be inaccurate.

APPROACH

We compared six clinical knowledge base development approaches for clinical summarization: manual creation, a standards-based ontology, association rule mining, crowdsourcing, reputation metric and an ensemble method.

Data sources

We used data from three sources. Partners Healthcare (Partners) is a large integrated academic clinical care network. We randomly selected a cohort of 100,000 patients from the Brigham and Women's Hospital seen at least once during 2007 and 2008 who had two or more outpatient notes in their record. The EHR data set included 272,749 problems and 442,658 medications. There were 1,756 unique problems and 2,128 unique medications [292].

University of Texas Physicians (UTHealth) is a large, multi-specialty, ambulatory, academic practice. Clinicians are required to manually link medications to an indication within their patient's clinical problem list for all medications ordered through e-prescribing. Between June 1, 2010 and May 31, 2011, clinicians entered 418,221 medications and 1,222,308 problems for 53,108 patients [293].

Blue Cross Blue Shield of Texas (BCBS-TX) is the largest commercial insurance provider in Texas. We extracted billed diagnoses and prescribed medications for 6,486,226 patients with claims between 2008 and 2011 [294, 295].

Manual creation

Knowledge bases can be manually created through expert clinician review. We used a six-step rule development process, including automated identification of problem associations with other structured data, selection of problems of interest, development of preliminary rules, characterization of preliminary rules and alternatives, selection of a final rules, and validation of the final rule (Chapter 19) [296, 297]. We developed rules using data from Partners Healthcare that asserted "treats" relationships between medications and problems.

Standards-based ontology

The Veterans Health Administration National Drug File-Reference Terminology (NDF-RT) [298] system provides a formal content model to describe medications and definitional relationships (e.g., simvastatin "may_treat" hypercholesterolemia) [299]. We created a standards-based ontology knowledge base using the "may_treat" linkage from NDF-RT, with medications and problems mapped to the National Library of Medicine's RxNorm naming system [300] and Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) [301] using the Unified Medical Language System (UMLS) [267]. We expanded the linkage to all levels of problem and medication hierarchies within UMLS using the "is" relationship within SNOMED CT and the "ingredient_of" relationship within RxNorm (Figure 1). To evaluate the resulting knowledge base, we randomly selected 25 patients who had at least three problems and at least five medications and reviewed all potential pairs for appropriateness as the gold standard.

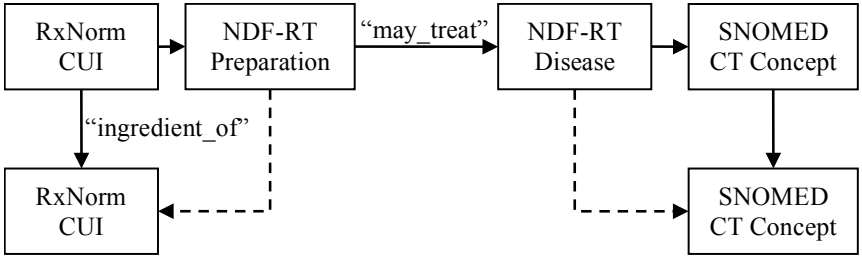


Figure 1. Generation of knowledge base for linking patient medications with clinical problems using RxNorm, NDF-RT, and SNOMED CT. Dashed line indicates inferred relationship.

Association rule mining

Association rule mining identifies related concepts using measures of interestingness and has been successful identifying relationships between clinical data elements [302]. We developed three knowledge bases from association rule mining Partners, UTHealth and BCBS-TX [292, 303] data. For each knowledge base, we used a minimum support threshold of five and a minimum confidence threshold of ten. The chi-squared statistic performed best when compared to a gold standard from our previous analysis.

Crowdsourcing

Crowdsourcing outsources tasks to a group or community. For example, Wikipedia depends on contributions from the public [304, 305]. Biomedical researchers are evaluating crowdsourcing to develop new resources, including drug knowledge [306, 307]. We created a crowdsourcing knowledge base from UTHealth data [227], using links between medications and problems manually asserted by clinicians during e-prescribing (Figure 2).

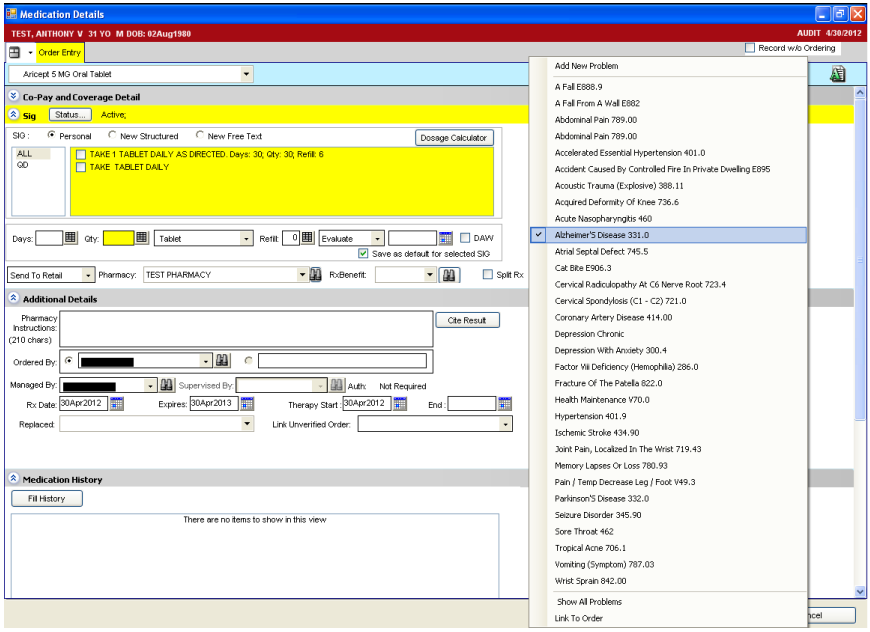


Figure 2. Example screen showing problem manually linked to medication during e-prescribing.

To ensure accuracy of retrieved links, we determined patient link frequency (the number of distinct patients for whom a link had been asserted) and link ratio (proportion of patients receiving a particular drug with a particular problem for which a link between the drug and problem had been manually asserted). We stratified problem-medication pairs into threshold groups using patient link frequency and link ratio. One hundred problem-medication pairs were randomly selected from each group to determine a threshold cutoff for which links had an estimated accuracy of 95% or greater. Pairs meeting the threshold were included the resulting knowledge base. We compared the knowledge base to a gold standard review of all potential problem-medication pairs for 100 patients.

Reputation metric

Reputation metrics are often used for evaluating user-generated content, such as e-commerce transactions [308], product reviews [309], and e-news or forum comments [310]. We developed a reputation metric knowledge base using logistic regression that included three contributing variables: clinician link sharedness (the proportion of links asserted by a given clinician also asserted by another clinician), clinician total distinct links (number of unique problem-medication pairs linked by a given clinician), and clinician link ratio (for each distinct problem-medication pair linked by a clinician, the average of the proportion of links asserted for all scenarios in which the clinician had the opportunity to link problem and medication) [311]. We included pairs by clinicians predicted to have 95% or greater accuracy for linked pairs by the regression model (Figure 3). We evaluated pairs with the same gold standard used with the crowdsourcing knowledge base.

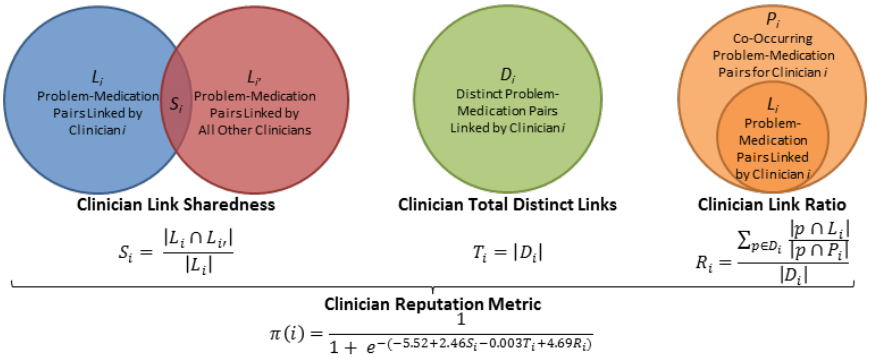


Figure 3. Definition of the clinician reputation metric approach.

Ensemble

We developed an ensemble method to integrate the five knowledge bases (association rule mining at Partners, UTHealth, and BCBS-TX; crowdsourcing; and reputation metric) into a unified computable problem-medication knowledge base. We mapped pairs from their source to standardized terminologies using natural language processing, then mapped medications to RxNorm at the ingredient level and problems to root ICD-9 codes (International Classification of Diseases [312]). Figure 4 shows the overlap of knowledge bases. Measures of confidence from each knowledge base approach were integrated into a matrix to facilitate problem-medication knowledge extraction from user-defined criteria.

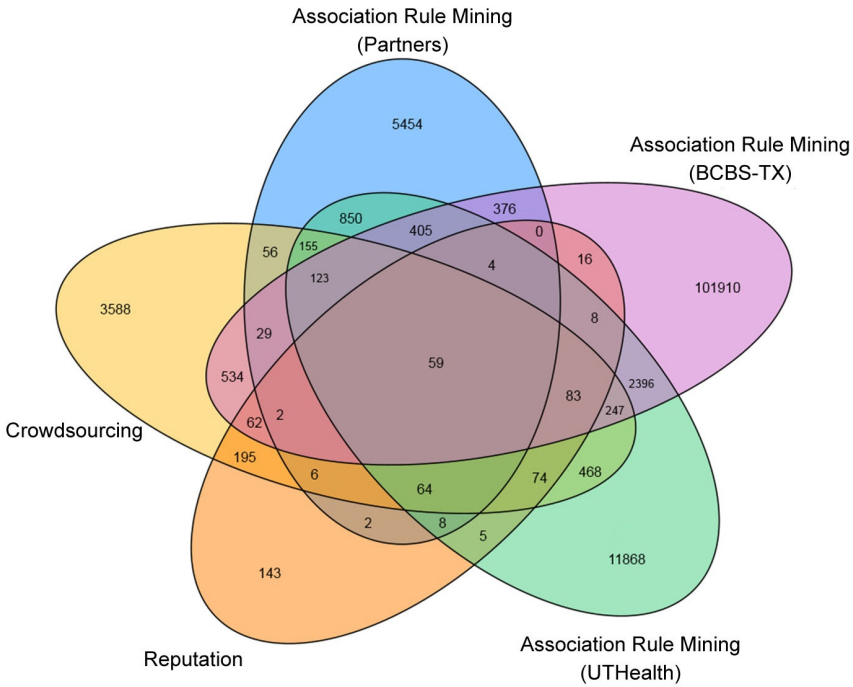


Figure 4. Overlap between previously developed knowledge bases after mapping to be included in the ensemble knowledge base.

RESULTS

Knowledge bases varied in number of pairs and estimated accuracy.

Knowledge Base Approach (Data Source)	Number of Pairs
Manually Created (Partners)	3,973
Standards-Based Ontology	33,894,415
Association Rule Mining (Partners)	6,427
Association Rule Mining (UTHealth)	104,424
Association Rule Mining (BCBS-TX)	271,853
Crowdsourcing (UTHealth)	41,203
Reputation (UTHealth)	982
Ensemble	128,928

Table 1. Summarization Knowledge Bases

The manually-created knowledge base included 3,973 problem-medication pairs, composed of eight distinct problems (asthma, breast cancer, coronary artery disease, depression, diabetes, dyslipidemia, hypertension, and hypothyroidism) with 53 distinct sub-problems and 909 distinct medications [296]. Although we did not formally assess accuracy of the manually created knowledge base, specificity is likely high due to the expert review, although coverage of the knowledge base compared to all possible clinical conditions and medications is low.

The standards-based ontology knowledge base was the largest with 33,894,415 problem-medication pairs. Pairs included 60,632 problems and 24,079 medications. Compared the gold standard review, the standards-based ontology knowledge base achieved 43.41% sensitivity and 98.28% specificity [267].

We performed association rule mining on the three source datasets. We identified 6,427 problem-medication pairs Partners, including 168 problems and 1,147 medications. Of the top 500 pairs, according to the chi square statistic, 89.2% were found in the gold standard, suggesting a high level of accuracy [292]. Using UTHealth, we identified 104,424 problem-medication pairs, including 563 problems and 9,088 medications. As with Partners, we found a high level of accuracy for the top pairs [303]. Using BCBS-TX, we identified 271,853 problem-medication pairs, including 1,693 problems and 2,459 medications.

The crowdsourcing approach identified 41,203 problem-medication pairs, including 4,676 problems and 4,903 medications. Compared to expert review, crowdsourcing achieved a sensitivity of 56.2% and specificity of 98.0%. Evaluation of the knowledge base combined with links asserted manually by clinicians found a sensitivity of 65.8% and specificity of 97.9% [227].

With the reputation metric approach, we identified 125 clinicians with an estimated link appropriateness greater than or equal to 95%. Problem-medication pairs linked by clinicians totaled 2,464 (982 pairs, including 368 problems and 572 medications), making it the smallest knowledge base. Our evaluation found the reputation metric knowledge base alone had a sensitivity of 16.1% and specificity of 99.5%. When combined with manual links by providers and the crowdsourcing knowledge base, sensitivity was 66.3% and specificity 97.8% [311].

The ensemble knowledge base had 128,928 problem-medication pairs among 2,118 normalized medications and 2,186 normalized problems. At present, we have not evaluated the accuracy of the ensemble knowledge base, although we anticipate good results as this approach is designed to combine the best aspects of each approach.

DISCUSSION

Each of the six knowledge bases identified a large number of potentially relevant problem-medication pairs with varying magnitudes and accuracy. The standards-based ontology knowledge base was largest, with over 33 million pairs. However, limiting source medications and problems to only those commonly prescribed (instead of all medications and problems, which include entries as specific as medication ingredients) would likely decrease the size to a number more closely aligned with the other knowledge bases. The reputation metric knowledge base was smallest, however, these pairs were highly accurate.

There are tradeoffs sacrificing specificity for sensitivity using a large knowledge base, or sensitivity for specificity with a smaller but more accurate knowledge base. Sensitivity is important for clinical summarization. Displaying a comprehensive overview of a patient's history for a given condition is also important and could be harmful if information is omitted. However, if specificity is not high enough, too much information may be displayed, which would render a clinical summary screen unhelpful compared.

All knowledge base development approaches have computational and accuracy limitations. Because of differing underlying terminologies, it is difficult to compare each approach. The ensemble knowledge base approach attempts to overcome this limitation, but we have not been able to develop an automated method to combine the various concepts encoded using differing clinical vocabularies from all knowledge bases, or evaluate the accuracy of the resulting knowledge base. Methods to accurately map each knowledge base to a consistent terminology are required, and as with the approaches for developing the knowledge bases, the varying mapping methods have advantages and disadvantages. Prior research has described methods for developing similar clinical knowledge bases, including use of standards-based ontologies, association rule

mining, and text mining. To our knowledge, our work is the first to combine the presentation of all approaches. A comprehensive evaluation of approaches is important to allow researchers and informatics personnel adopting these methods to understand benefits and drawbacks.

There are limitations to our work. Although each method is included and presented for comparison, we were unable to directly compare approaches without having each mapped to a single, standardized terminology assessing overlap. Analyses were only performed independently. Some approaches were only evaluated using a single source of data. It is unknown if our approaches generalize across all healthcare settings.

Future Direction

We plan to better map knowledge bases to standardized terminology. We will also develop improved ensemble knowledge and apply these approaches to additional clinical data elements, including laboratory values and procedures. Our preliminary findings suggest additional approaches might be necessary [292, 293]. We hope to use the resulting knowledge bases to generate actual clinical summary screens (Chapter 19) and conduct a trial to determine if clinical summarization improves patient safety.

Suggested Reading

Feblowitz, J. C., Wright, A., Singh, H., Samal, L., & Sittig, D. F. (2011). Summarization of clinical information: a conceptual model. J Biomed Inform, 44(4), 688-699. doi: 10.1016/j.jbi.2011.03.008.

Klann J,. Wright A., McCoy A. B., Murphy S. (2012). The Medical App Store, Research Data Repositories, and Physician Cognitive Overload: Uniting Three Large, Multisite Grants for Health Care Transformation. Panel Presentation presented at the AMIA Annu Symp Proc, Chicago, IL.

Laxmisan, A., McCoy, A. B., Wright, A., & Sittig, D. F. (2012). Clinical Summarization Capabilities of Commercially-available and Internally-developed Electronic Health Records. Appl Clin Inform, 3(1), 80-93. doi: 10.4338/aci-2011-11-ra-0066.

McCoy, A. B., Wright, A., Laxmisan, A., Ottosen, M. J., McCoy, J. A., Butten, D., & Sittig, D. F. (2012). Development and evaluation of a crowdsourcing methodology for knowledge base construction: identifying relationships between clinical problems and medications. J Am Med Inform Assoc, 19(5), 713-718. doi: 10.1136/amiajnl-2012-000852.

McCoy, A. B., Wright, A., Laxmisan, A., Singh, H., & Sittig, D. F. (2011). A prototype knowledge base and SMART app to facilitate organization of patient medications by clinical problems. AMIA Annu Symp Proc, 2011, 888-894.

McCoy, A. B., Wright, A., Rogith, D., Fathiamini, S., Ottenbacher, A. J., & Sittig, D. F. (2014). Development of a clinician reputation metric to identify appropriate problem-medication pairs in a crowdsourced knowledge base. J Biomed Inform, 48, 66-72. doi: 10.1016/j.jbi.2013.11.010.

Wright, A., Chen, E. S., & Maloney, F. L. (2010). An automated technique for identifying associations between medications, laboratory results and problems. J Biomed Inform, 43(6), 891-901. doi: 10.1016/j.jbi.2010.09.009.

Wright, A., McCoy, A., Henkin, S., Flaherty, M., & Sittig, D. (2013). Validation of an association rule mining-based method to infer associations between medications and problems. Appl Clin Inform, 4(1), 100-109. doi: 10.4338/aci-2012-12-ra-0051.

Wright, A., Pang, J., Feblowitz, J. C., Maloney, F. L., Wilcox, A. R., Ramelson, H. Z., . . . Bates, D. W. (2011). A method and knowledge base for automated inference of patient problems from structured data in an electronic medical record. J Am Med Inform Assoc, 18(6), 859-867. doi: 10.1136/amiajnl-2011-000121.

19: Clinical Knowledge Bases to Improve Problem List Completeness

Challenges

Adam Wright, PhD
Brigham & Women's Hospital

Allison B. McCoy, PhD
Tulane University School of Public Health & Tropical Medicine

Dean F. Sittig, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

ABSTRACT

Problem lists summarize patient medical histories. Accurate clinical problem lists are critical for patient care, clinical decision support, population health management, quality improvement and biomedical research. Unfortunately, clinical problem lists are frequently incomplete or out-of-date. We developed innovative methods of constructing inference rules for 17 clinical conditions and evaluated an electronic health record (EHR)-based intervention to improve problem list documentation in a cluster randomized trial. A total of 17,043 problem list-related alerts were presented and 41.1% accepted. Providers documented significantly more problems in the intervention arm (adjusted odds ratio=3.4, $p<0.0001$), with an absolute difference of 6,277 additional problems in the intervention group compared to the control group. Significant increases in documentation were observed for 14/17 problems. Problem inference alerts in EHRs increase documentation of important patient problems in primary care, which can facilitate quality improvement and enable knowledge-based problem-oriented summarization.

INTRODUCTION

Clinicians with a clear understanding of their patients' problems and diagnoses make better clinical decisions [313]. Overall quality of care increases, as does health organization quality improvement and measurement. Biomedical research efforts are also impacted.

Problem lists were first proposed by Dr. Lawrence Weed in 1968 [260]. Now a central component of problem-oriented medical records, problem lists describe active diseases, document risk factors, facilitate workups and treatment, promote continuity of care, help generate care plans and manage preventative care [314-316]. Computerized problem lists offer advantages over paper, such as linking laboratory results, imaging studies, medications and allergies to central problems [260, 317]. Electronic patient problem lists can be enhanced by structured problem vocabularies [318-320]. ICD-9 (International Classification of Diseases [312]) and SNOMED (Systematized Nomenclature of Medicine Clinical Terms) [301] are often used, but there are limitations [321-324]. Other methods to improve the structure, accuracy, and utility of problem lists have also been proposed [320, 325-328].

An accurate electronic problem list is a cornerstone of modern electronic health records (EHR). Clinicians use problem lists to familiarize themselves with the needs of a patient treated for the first time, to inventory conditions that may require management during a visit, or as a marker of contraindications for a therapy. Accurate problem lists make it easier to communicate with other care providers. Despite their importance however, problem lists are often incomplete and poorly maintained [329-331]. Inaccurate problem lists are associated with low quality of care [332, 333].

Partners Healthcare, a large integrated academic clinical care network, uses problem lists to trigger reminders to help clinicians manage chronic diseases, which account for a large proportion of healthcare costs. Twenty-two percent of Partners' clinical decision support rules depend on coded patient problem lists [313]. A clinician with a diabetic patient, for example, will receive appropriate alerts and reminders to guide care. Quality care is measured and tracked, and the patient possibly flagged as eligible for special care management programs.

To receive federal incentives for "meaningful use" under the HITECH Act (up to $44,000 through Medicare and $63,750 through Medicaid), providers must "maintain an up-to-date problem list of current and active diagnoses." Eighty percent of patients having at least one problem must be recorded or an indication of "no known problems" entered [114, 334]. Since problem lists are often incomplete, alternative information sources have been sought. Several systems have been reported using natural language processing to infer clinical problems (25-27). Researchers have also used data mining to identify clinical data as proxies for problems [226, 292, 335]. Carpenter and Gorman (2002) used medication to identify possible problem mismatches [332] and Poissant et al. employed a combination of billing codes, single-indication drugs and prescription indications to infer problems in an electronic prescribing system [336, 337]. Inferring patient problems through data mining is promising, but reported systems are limited. Most use only a single type of data (medication, billing code, or narrative text) to make inferences. Many rely on time-consuming manual techniques for generating knowledge bases. To our knowledge, none have provided a full knowledge base for use or validation by others.

We describe a knowledge base for improving problem list completeness [296], the results of a randomized trial conducted using this knowledge base with an alert intervention [297] and the results of a user survey [338].

APPROACH

We developed a problem list knowledge base using a six-step process designed to yield high quality rules with known performance characteristics [296]:

1. Automated identification of problem associations with other structured data;
2. Problems of interest selection;
3. Preliminary rules development;
4. Characterization of preliminary rules and alternatives;
5. Final rule selection; and

6. Final rule validation.

Additional knowledge bases were subsequently developed (Chapter 18). We then created an electronic alert intervention for Brigham and Women's ONC-ATCB (Hospital Office of the National Coordinator - Authorized Testing and Certification Body) certified Longitudinal Medical Record (LMR) system. The alert notified providers when a patient had an undocumented clinical problem. When a provider saved a note or reviewed a dictation, the system analyzed the patient's medications, laboratory results, billing codes, and vital signs and used the knowledge base to determine if the patient likely had any of 17 study problems.

When the system detected one or more potential problems, it reviewed the problem list to determine if the problem was documented. If not, an actionable alert was shown onscreen. If more than one undocumented problem was detected, alerts for all undocumented problems were displayed in a single window (Figure 1). The reason for the alert was shown next to a checkbox to add the problem to the list. Problems were "pre-checked" for ease-of-use. Providers could accept the alert to add the problem to the problem list, and add details or select a related term (e.g., "gestational diabetes" or "diabetes mellitus type 2" instead of simply "diabetes mellitus"). Or, providers could ignore the alert, causing the alert to re-appear when a new note was completed. The provider could also override the alert, which suppressed the problem for the duration of the study.

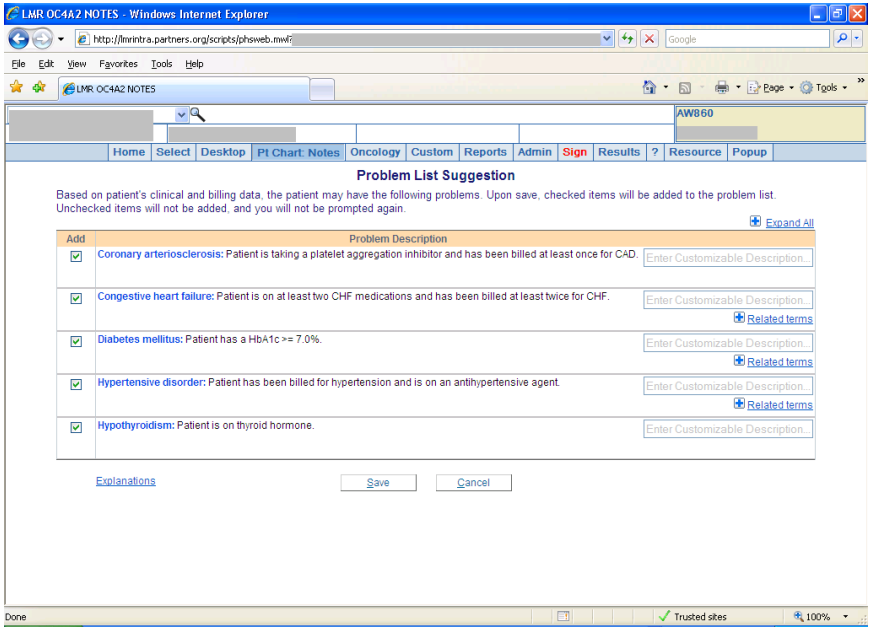


Figure 1. Partners Healthcare LMR showing the problem inference alerts.

We conducted a six-month randomized controlled trial of the alert intervention and collected baseline data prior to the intervention. The study was approved by the Partners HealthCare Human Research Committee and registered with ClinicalTrials.gov (NCT01105923). We used a cluster randomization method to reduce risk of contamination,

Clusters (n = 28) were designated based on pre-existing administrative divisions within clinics. For example, one primary care clinic was divided into adult medicine, family medicine, and pediatric medicine, and another into suites A, B, and C. In both cases, sub-units were treated as separate clusters. Clusters were then grouped into three bands: hospital based, community and federally-qualified health center. Once grouped, clusters within each band were randomly allocated to the control or intervention arm, with 14 clinics randomized to the control arm and 14 to the intervention arm.

Providers were not aware which arm their sub-clinic group was assigned until the intervention was implemented. Patients were not made aware of the intervention. No pre-intervention orientation or training took place in the intervention arm. Blinding was not possible given the nature of the intervention. Data were collected over a six-month pre-

intervention period and a subsequent six-month intervention period. The system went live May 16, 2010 in intervention group clinics. Post-period data were collected prospectively for six months (183 days) in both arms, concluding November 14, 2010. Six months (183 days) of pre-period data from both arms were retrospectively collected as a baseline.

Primary outcome was alert acceptance rate, defined as the number of accepted alerts divided by the number of unique alerts presented. In certain instances, providers might see the same alert serially, so we aggregated presentations and acceptances of the same alert for the same patient. We calculated acceptance rates for each of the 17 conditions, as well as an overall acceptance rate.

We measured the number of study problems documented in the groups during both time periods as a secondary outcome. The unadjusted relative rate of problem notation in the intervention group was calculated by comparing the number of problems recorded in the intervention arm during the intervention period for all other groups and tested for equality with 1 using a normal approximation. We modeled our data as Poisson-distributed counts.

We used Poisson regression with an interrupted time series to control for potential exogenous temporal effects. Five coefficients and a scale parameter modeled starting rate, four slopes (pre and post-period for the control and intervention arms) and a parameter for effect of the intervention. The effect parameter was an odds ratio for the immediate effect of the intervention. We removed related terms from the model when differences between control and intervention groups were non-significant. This resulted in a new intervention parameter measuring overall effect. The parameter has a similar interpretation to our unadjusted relative rate, and was compared for equality with 1 using a chi-square test.

Results

Twenty-eight clinics completed the study with no loss to followup. Overall, 41,039 patients were seen in the control clinics during the study period, and 38,025 patients in the intervention clinics. A small number of patients (n = 3,894, 5.2%) were seen in both intervention and control clinics and appear in both study arms.

Intervention and control groups were clinically similar across a range of demographic and clinical variables. During the six-month pre-intervention period, greater problem list use was observed in the control group, with 3,230 study problems (17.8 problems/day) added in the intervention group and 3,597 study problems (19.8 problems/day) added in the control group ($p < 0.001$).

Problem inference rules fired a total of 17,043 times during the intervention period for a total of 11,508 patients in the intervention arm. The overall problem inference alerts acceptance rate was 41.1%. Glaucoma alerts had the highest acceptance rate of the 17 conditions (55.7%). Alerts for myasthenia gravis and sickle cell disease were infrequently presented and infrequently accepted.

During the intervention period, 10,016 study problems were added in the intervention group compared to 3,739 to the control group, representing an absolute difference of 6,277 problems (compared to 367 fewer problems added in the intervention group during the pre-intervention period, $p<0.0001$). The unadjusted relative rate of study problem addition was 2.98 times more problem notation in the intervention group ($p < 0.0001$), and the adjusted odds ratio was 3.43 ($p < 0.0001$). The cumulative number of study problems added over the course of the entire study is shown in Figure 2.

The rate of study problem notation during the pre-intervention period was slightly lower in the intervention group compared to the control group. The inflection point in the intervention group line was coincident with the initiation of the study intervention. By completion of the study, the intervention group had added significantly more problems than the control group.

We used Poisson regression and interrupted time series analyses to control for temporal trends. The overall odds ratio for intervention effect on problem list notation was 3.43 ($p < 0.0001$).

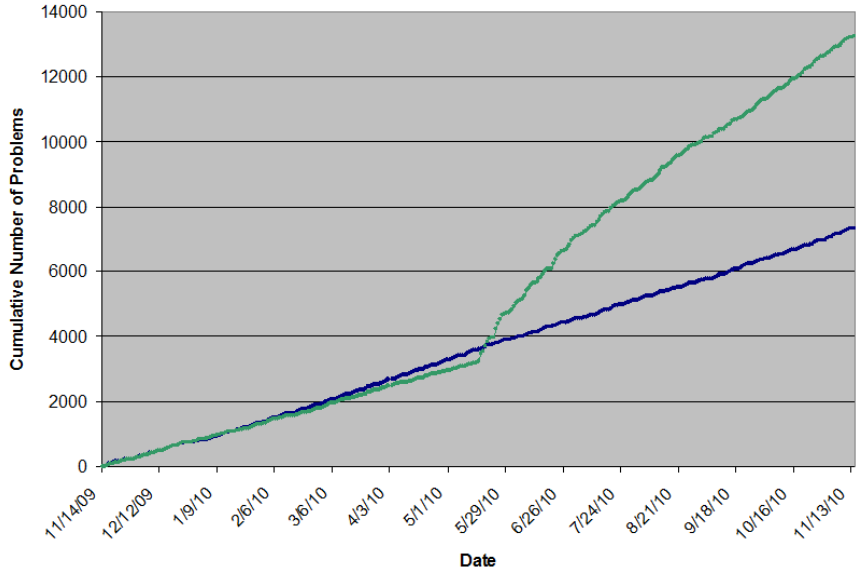


Figure 2. Cumulative number of problems added in the intervention and control groups.

We then conducted a survey of provider attitudes toward the problem list and alert intervention. In total, 103 of 140 providers completed an online survey (response rate: 73.6%). Twenty-eight providers (20.0%) declined to participate and nine providers (6.4%) could not be reached via email. Seven of the 103 responding providers indicated they had not received the alerts, despite electronic logs indicating all had received them. Non-responders were significantly more likely male, significantly younger, had significantly fewer total notes (a proxy for visit volume) and had significantly less unique alerts than responders. Providers' attitude towards the intervention varied widely. Of 103 survey respondents, users reported a median alert frequency of 5.0 (a few times per week, [IQR 5.0-7.0) across the study period. For the 96 providers reported receiving alerts, median alert accuracy and self-reported acceptance rate were both 5.0 ("sometimes accurate," [IQR 3.0-6.0]; and "accepted alerts sometimes," [IQR 3.0-7.0]). Users reported rarely accepting alerts when covering patients for another provider (median = 2, IQR 1.0-3.0).

To assess potential predictors of alert acceptance, we performed linear regression on both the providers participating in the trial (n = 140) and providers responding to the survey (n = 103). For the trial, we

assessed if degree (MD/NP/PA), gender, age, medical school (top 25 or non-top 25), or graduation year predicted acceptance. For the survey, we assessed whether degree, gender, age, medical school (top 25), graduation year, years of experience, years of experience using an EHR or patient volume (patients/week) predicted acceptance rate. We found no factors predicted provider acceptance in the trial. Graduating from a Top 25 medical school was significantly positively associated with increased acceptance rate of alerts ($r = 0.198$, $p = 0.009$) in the survey.

DISCUSSION

We found electronic problem list alerts frequently accepted by users, resulting in a substantial increase in study problem notation. Study problems were approximately three times more likely to be documented when alerts were shown. This increase is important because many of these problems are used for quality improvement and clinical decision support.

Results suggest problem inference rules are valuable for improving problem list completeness and, ultimately, patient care. Better problem lists are easier for providers to assess patient issues, which is important when seeing an unfamiliar patient such as in emergency rooms or inpatient wards. Because health problems are not only used for clinical decision support, but research study recruitment and quality measurement, inference rules could have a wide-ranging impact.

An important question is how increased problem notation could benefit patients. Assuming a given alert was correct, there were two potential scenarios: 1) the alert called attention to an undocumented problem and the provider was unaware of it, or 2) the alert recommended a problem the provider was aware of, but had not documented in the problem list. While the first scenario may have an immediate clinical impact (making the provider aware of an unknown diagnosis), it is likely to be less common. Both scenarios, however, provide significant positive clinical benefit, including enabling clinical decision support (such as relevant preventive care reminders), facilitating quality measurement and research, and promoting awareness of a patient's active problems among the entire care team, including providers that may not know the patient well.

An implication of this study is helping providers achieve EHR "meaningful use." Stage 1 and 2 meaningful use goals must demonstrate using problem lists for 80% of patients. By meeting meaningful use criteria, clinicians receive incentive funds to offset the expense of implementing and maintaining longitudinal medical records. An alert intervention tool may help providers just implementing EHRs and struggling to populate problem lists.

Future Direction

We plan to expand our work by:

1. Increasing the size, scope and accuracy of the clinical knowledge base (Chapter 18).
2. Supplementing the structured data currently used in our system with free-text data through natural language processing.
3. Expanding the intervention to other hospitals and healthcare systems using a variety of EHR products.

Implications

A knowledge-driven approach to clinical problem documentation can be effective. Providing clinicians with tailored tools to support their cognitive processes can improve problem documentation quality, resulting in a more complete problem list.

Suggested Reading

Wright, A., Maloney, F. L., & Feblowitz, J. C. (2011). Clinician attitudes toward and use of electronic problem lists: a thematic analysis. BMC Med Inform Decis Mak, 11, 36. doi: 10.1186/1472-6947-11-36.

Wright, A., Pang, J., Feblowitz, J. C., Maloney, F. L., Wilcox, A. R., McLoughlin, K. S., . . . Bates, D. W. (2012). Improving completeness of electronic problem lists through clinical decision support: a randomized, controlled trial. J Am Med Inform Assoc, 19(4), 555-561. doi: 10.1136/amiajnl-2011-000521.

Wright, A., Feblowitz, J., Maloney, F. L., Henkin, S., & Bates, D. W. (2012). Use of an electronic problem list by primary care providers and

specialists. J Gen Intern Med, 27(8), 968-973. doi: 10.1007/s11606-012-2033-5.

Holmes, C., Brown, M., Hilaire, D. S., & Wright, A. (2012). Healthcare provider attitudes towards the problem list in an electronic health record: a mixed-methods qualitative study. BMC Med Inform Decis Mak, 12, 127. doi: 10.1186/1472-6947-12-127.

Wright, A., McCoy, A., Henkin, S., Flaherty, M., & Sittig, D. (2013). Validation of an association rule mining-based method to infer associations between medications and problems. Appl Clin Inform, 4(1), 100-109. doi: 10.4338/aci-2012-12-ra-0051.

20: Authoring & Editing of Decision Support Knowledge

Facilitating broader use & interoperability of best-practice CDS knowledge

Robert A. Greenes, MD, PhD
Arizona State University

Davide Sottara, PhD
Arizona State University

Peter J. Haug, MD, PhD
Intermountain Healthcare

Abstract

We worked with a national initiative to refine a model for representing clinical decision support (CDS) knowledge in unambiguous, sharable, standardized form to help electronic health record system vendors better integrate best-practice CDS into products. We also created a tool for knowledge engineers and subject matter experts to author and edit CDS knowledge in this sharable form.

Introduction

This project began by addressing the complexity faced by health care organizations and practices in developing computable rules from narrative decision support recommendations and customizing the rules to unique setting-specific factors (SSFs). The customization of knowledge to smoothly integrate into the workflow in a particular setting and adaptation to other local considerations have long been recognized as requiring a major effort. Failure to do this well will often impede successful implementation. Based on prior work by a consortium in which we were engaged (The Morningside Initiative [339]), we developed a four-stage knowledge refinement paradigm:

- Initial markup and categorization of a recommendation based on purpose, user, domain, and other components using narrative text entries.
- Formalization of the above using information modeling, coding systems, and value sets.
- An iterative process of modeling of adaptations of the rule based on SSFs, such as how a rule would be triggered in a particular setting, in what clinical context, refinement of inclusion and exclusion criteria, incorporation of timing considerations (such as pre-event firing when an action is due vs. post-event firing only if the action was not done when expected), and to whom and how the recommendations or actions are to be delivered.
- Conversion to an executable form for use in a particular environment, typically involving translation to a proprietary electronic health record (EHR) system's internal knowledge representation format and mapping of the information model (the patient data and the rule's clinical knowledge) to the EHR's internal representation.

The initial goal of the project was to develop the details of the Stage 2 and Stage 3 knowledge refinement process and to create an Implementer's Workbench (an editing tool) that would facilitate this process. Stage 4 was beyond the scope of the project, but there had been a prior demonstration showing this was feasible by automatically converting rules developed in an XML-based version [259] of the Arden Syntax [340, 341], a standards-based rules language for healthcare, to Drools [342], a popular generic rules management system representation.

With the onset of the Health eDecisions (HeD) Initiative in 2012 [343] sponsored by the Office of the National Coordinator for Health Information Technology (ONC), midway through the project the team was asked to turn its efforts to working with a national team on the development of a standard model-based representation and XML exchange format for best-practice knowledge that could be included as a requirement for EHR systems as part of Meaningful Use Stage 3. Knowledge to be exchanged was intended to include decision rules, order sets, and documentation templates. All of the above types of artifact in the HeD Initiative were considered typically to consist of sections of the

model representing: (a) metadata about authorship, focus, provenance, version, etc.; (b) possible triggering events; (c) a standards-based description of the data necessary for the artifact; (d) a logical condition expression; and (e) a set of actions to be performed if the logic evaluates to true.

Rules, order sets and documentation templates each can be considered to be special cases where these sections are constrained in different ways. CDS rules generally require the specification of events, conditions and actions, whereas order sets and documentation templates may involve conditions as indications, but most of the knowledge is specified in the action part. Regarding rules, the focus of the HeD effort was to model these in a context-agnostic mode, since the customization for particular SSFs (such as trigger conditions, workflow, mode of delivery of actions, etc.) were considered beyond the scope of national-level, best-practice knowledge distribution. Thus, the focus corresponded to stage 2 of our original model.

The team was also tasked with building an authoring and editing tool for HeD knowledge artifacts (KAs). A key feature of this tool is that it would enable authoring not only by knowledge engineers (KEs) but also by subject matter experts (SMEs) who could work at a level that does not require deep technical knowledge. Addressing this requirement was greatly facilitated by having a formal underlying model such as was being developed as part of the HeD initiative. Using a formal model provides a number of benefits in terms of the ability to predict and represent knowledge that should be co-associated, to apply constraints on types of values that could be used in a particular part of a KA, and to potentially render the output in a variety of other languages and representation formats, provided that adaptors for doing so are built.

Here we describe the results of this effort and its initial evaluation by application to some specific knowledge authoring tasks. The authoring tool that has resulted from our work provides one of the first available tools for model-based authoring of knowledge artifacts designed for non-technical use. While many extensions and refinements are required, this work has the potential to be a foundation for a variety of other activities in the future.

The knowledge authoring problem for health care-oriented KAs typically suffers from a disconnect between the ability of a human expert to comprehend and grasp it and the detail required for mapping the knowledge to formal patient data/information model elements, coding schemes, value sets, and proprietary record formats and modes of execution. Thus, most knowledge authoring today is done by using custom or system-specific authoring/editing tools provided by the EHR vendor and is typically at a level that must be carried out by a KE or software engineer. There is little ability to organize the corpus of knowledge to review what it contains, search it by specific attributes (such as domain, setting, usage, or mode of intended execution), manage the corpus of knowledge or update it, or identify gaps in knowledge requiring attention. Thus, we believe that this work, while limited in initial scope to the goals of the HeD initiative, has the potential to be extended to enable it to be useful also for the original goals of the project to accommodate specialization with SSFs, and for a variety of other purposes that are enumerated in the final section.

APPROACH

We worked in conjunction with other national participants in the HeD Initiative to develop a formalized model of KAs that could be standardized (working with the Health Level Seven (HL7) [344] standards development organization) and that could be used as a basis for creating a distribution/exchange rendition of the KA in XML format [345]. The model-based approach is in contrast to focusing only on the XML representation and has enabled a variety of powerful capabilities to:

- View, refine, and extend the model;
- Provide model-based associations and constraints as a guide to authoring/editing and run-time usage of KAs;
- Translate the KA to a variety of formats (not just XML) but English or other languages, Drools, Arden Syntax, or various rule engine syntaxes, provided the target format has a well-defined syntactical structure and that adaptors are explicitly built for this purpose;

- Be incorporated into a knowledge repository allowing appropriate indexing and tagging based on the model component types, coding systems, and value sets.

The internal model is based on Description Logic, a widely adopted formalism with descriptive and inferential capabilities [346]. More specifically, we have chosen the Web Ontology Language v2 (OWL2-DL) [347], a W3C standard designed for interoperability over the web. This choice facilitated the development of the models and the software, since there exist several software tools and libraries, such as Protégé [348] or the OWL APIs [349] that support the OWL2-DL language natively. Moreover, OWL2-DL has been specifically designed for open, collaborative environments such as the Web. Many general-purpose (or "upper") ontologies have been released by research groups and/or standard-defining organizations using OWL, providing initial foundations for domain-specific models. Some of these ontologies inspired the creation of the HeD XML schema in the first place. However, the XML specification was mostly focused on the ability to deliver the content as output; it was not intended to capture or represent the complete semantics present in the original ontologies. The editor, however, tries to leverage both the content and the context. In particular, the foundations of our work are:

- SKOS [350], which was used to conceptualize clinical and medical terms and vocabularies;
- The Dublin Core (DC) [351], which was the basis for the HeD metadata;
- The Production Rule Representation [352]OMG standard (PRR), which provided the general structure of a KA;
- and a combination of Object Constraint Languages [353], inspiring the HeD expression language.

We also included the LMM [354] to capture concepts and the ability to reference and mention them, as well as the DULCE/IO-Lite [355] ontologies, which allowed us to contextualize our required concepts and provide support for a future integration of SSFs (Figure 1).

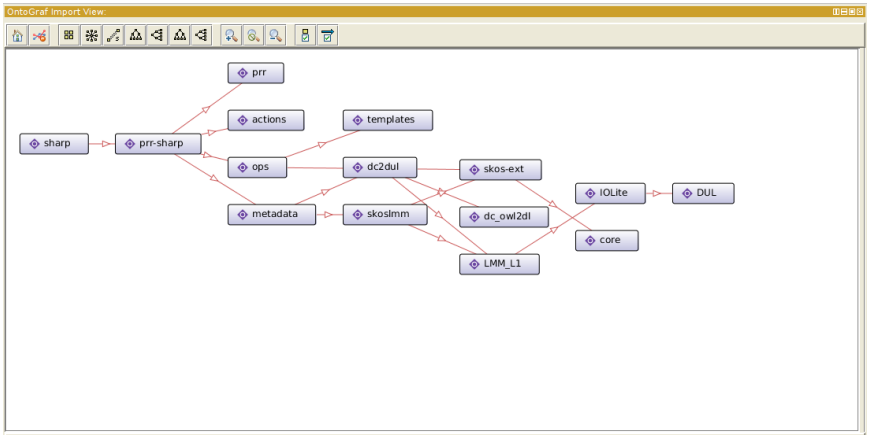


Figure 1. The modular HeD ontology and its dependencies.

We extended and harmonized these ontologies to include the specific concepts needed to model HeD artifacts and their content. We also adopted a modular approach to preserve the original components and facilitate future extensions. Notice that, although some ontologies have been created manually, others have been generated dynamically. For example, the rule authoring process requires a description of the domain-specific information model used to deliver the data at runtime (HL7 vMR [356], in our case). This model is also described using an ontology, which has been derived from the vMR schema. Similarly, the ontology module that covers the expression language is the result of a partially manual and partially automated generation process. At the time, in fact, the HeD schema did not completely define and constrain the use of the expression language, delegating that responsibility to an implementation guide. Being expressed in natural language, the implementation guide could not be integrated directly in our semantic framework. We first had to formalize the additional content of the implementation guide before we could integrate the expression language with the rest of the editor's framework.

The ontologies are the models driving the editor which, in turn, are based on a simple 3-tier architecture (Figure 2). The persistence layer allows storage and retrieval of a KA from a repository (currently a simple repository based on a file system implementation is provided, but APIs will allow replacement of it with a more robust implementation). The KAs are stored in RDF [357]format rather than HeD/XML to preserve

the additional information in the semantic description. The editor core is responsible for loading the artifact being authored and the ontologies required to model it. The core will also analyze the artifact, generate the internal data structures required during the authoring process and apply the additions and transformations requested by the user through the user interface. The presentation layer is a pure web-based application written in JavaScript which interacts with the core through a set of RESTful [358] application programming interfaces (APIs). The core is packaged as a Play™ [359] application, which allows the editor to be deployed in the cloud, as well as a web application container such as Tomcat [360].

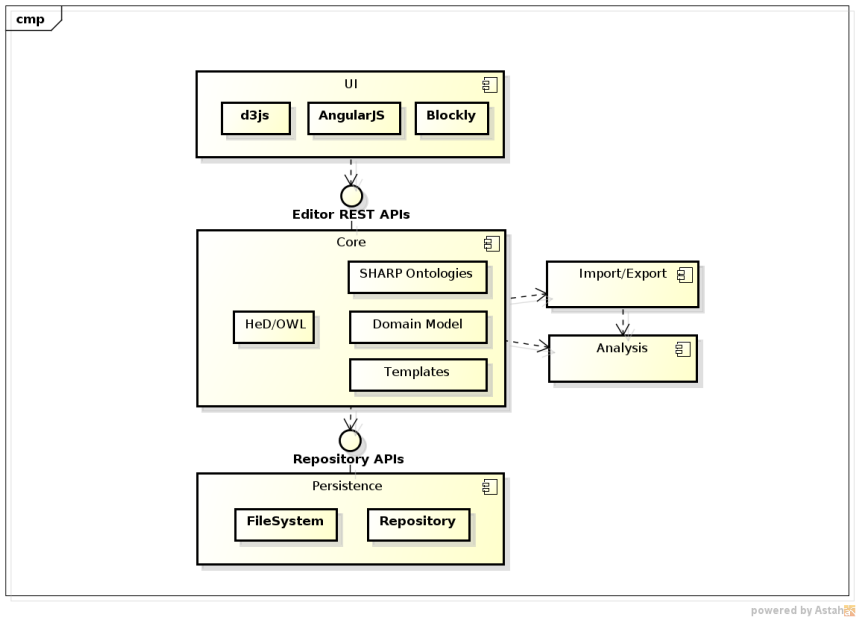


Figure 2. Implementer's Workbench *conceptual architecture.*

Relying on the underlying formal model and the editor's modular architecture, our basic approach to developing the editor was to create constructs that SMEs could use to define a KA at a somewhat high, conceptual level and to associate with each construct the specific attributes needed to be specified in order to create a placeholder for it. For example, if a rule is to refer to the existence of a specific laboratory test result being available within a timeframe and above a threshold value, then the name and the timeframe and value need to be specified. A default coding scheme can be associated with it based on the model

provided, and a presumptive coded value or value set can be provided, but these can be refined later by a KE.

We used five approaches to design the SME-oriented interface:

1. *Extensible library of templates for higher-level constructs.* If one is entering a clause for a rule about laboratory test results, the model defines certain attributes that are associated with the concept of laboratory test results, that might be required to be specified in a rule, such as the date the test was done and its value. But we can go further by identifying likely clause types that a laboratory test might participate in, such as whether the test was performed within a specified date range or whether its result exceeded a threshold. By creating prototypes or templates for such constructs, we make it easier for an author to create a rule with such a clause, since it pre-associates the appropriate operator and operand(s), and suggests needed attributes as well as possible optional ones, as well as possible default coding systems and values. A prior study by Greenes and Sordo [356] in reviewing rules at Partners Healthcare in 2004 showed that the many thousands of rules in use tended to reuse some 40+ clause types. This suggests that creating templates for commonly used clause types would be both feasible and useful. The templates determine which data element properties are relevant, so selecting a template type means the author need only focus on specifying those properties needed for a specific clause type. Moreover, templates support default values and/or constraints on operations and values, which further simplifies the authoring and allows validation routines to be run. The templates are defined using a dedicated ontology, but can be pre-loaded from a spreadsheet compliant with a simple schema, derived from the official HeD template specification for "Use Case II".

2. *Extensions of the terminology server to recognize concept classes.* Ideally, authors can begin typing a concept or a clause and, as they do so, have the system suggest the full concept corresponding to what they are typing. "HbA…", as it is being typed, could be anticipated to be "HbA1c," which is a synonym for "Hemoglobin A1c," which, in turn, is recognized as a lab test name. This would enable a user to select a lab test clause for a rule without having to first declare that he/she is intending to write a clause about a lab test. Moreover, based on the

library of templates in which clause types have a primary concept, the kinds of relationships available for that concept type can be listed (filtering the full list of templates based on the concept type being entered) which can then enable the user to choose the particular template desired and guided for completion of the necessary associated attributes (Figures 3, 4).

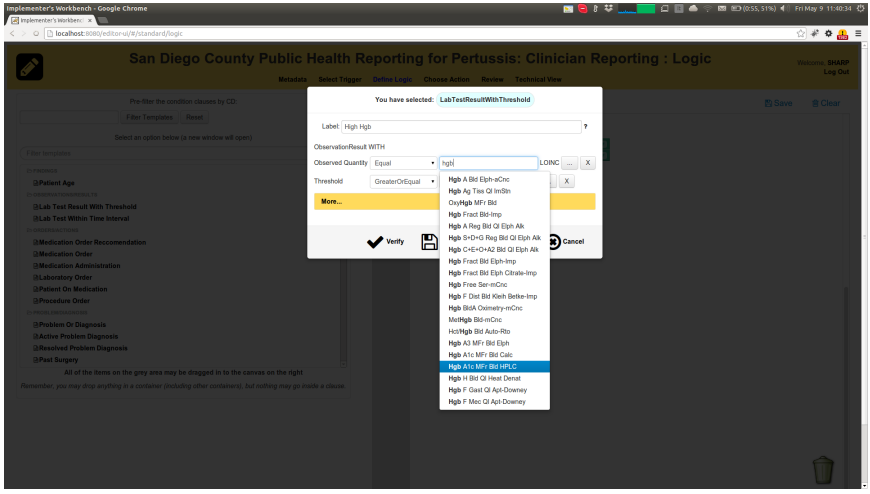
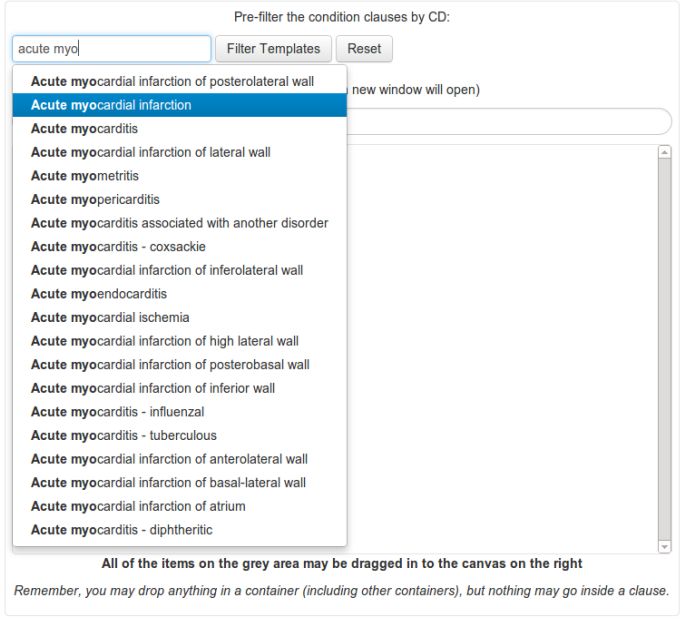


Figure 3. Template-based clause authoring.



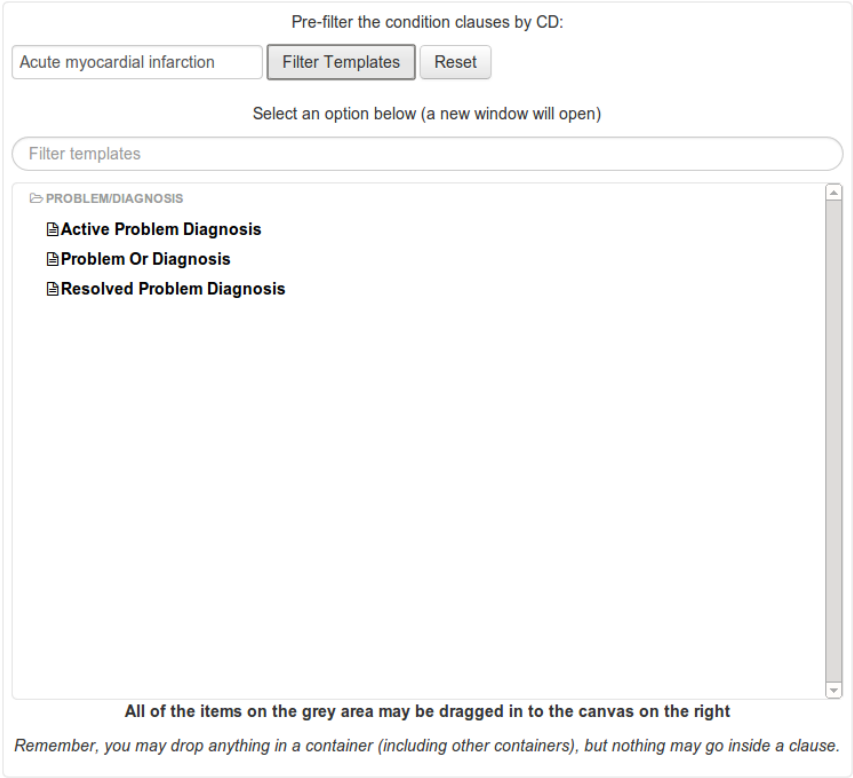


Figure 4a/b. Template filtering based on coded concept compatibility.

3. *Definitions capability for complex concepts.* A limited ability to define concepts that include several possible alternatives is provided by value sets. There are a number of organizational entities that create such value sets, e.g., for concepts for which quality measures are to be constructed. An example is the Value Set Authority Center (VSAC) of the National Library of Medicine (NLM), which compiles sets of quality measure value sets.

   By indicating the component of a KA does not refer to a specific coded concept but is in a defined value set, an author can use that as a shorthand for a more complex rule. A more extensive definition capability that uses conditional expressions to define a concept, such as "presence of diabetes" as inferred from the existence of not only diagnosis codes but evidence of insulin medication or repeated

elevated HbA1c values is also desired, but beyond the scope of the current HeD model and is identified in the discussion of possible extensions at the end of this chapter.

Once defined, value sets can be uploaded to the terminology service. The editor will then be able to use the value sets, either as an aggregated concept or explode them to pick individual codes. For initial testing, we have been able to load value sets into our version of the CTS2 terminology server through a CSV file/spreadsheet, invoking a custom utility.

4. *A visual building-block approach to constructing KAs.* Especially for rules, logic expressions can be complex, involving multiple AND, OR, and NOT clauses, sometimes nested. Triggers and actions can also have several components, and actions, in particular, can be complex, particularly for representing multi-part actions in rules or for order set or documentation template types of KAs. To address this, we adopted the Google Blockly [361] visual expression authoring tool to support the creation of complex constructs using Lego-like building blocks. Individual components as well as aggregations can be given short names that are highly readable.

   This capability is used in several authoring system components. The expression authoring section, designed principally for KEs, allows HeD named expressions to be composed using blocks generated dynamically from an expression ontology. This expression ontology is derived from the HeD model. A similar visual approach is used to allow navigation of the domain model (the vMR in our current approach to HeD KA authoring) to include references to domain classes and properties. (See Fig. 5.)

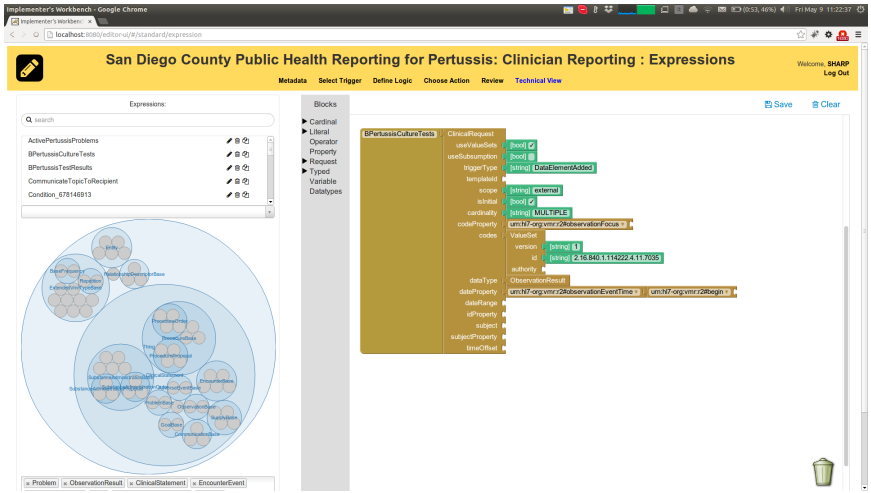


Figure 5. Authoring HeD named expressions with Google Blockly.

5. *Knowledge review.* Based on the compact naming of KA components described above and the hierarchy of nesting represented in the building block construction, we can produce a highly readable indented outline of the KA. This provides a compact view of the actual artifact, which can be exported in the HeD/XML format. A style sheet allows rendering of KAs into narrative English report form, but the outline list is a very compact way for the author to review what he/she has created. (See Fig. 6).

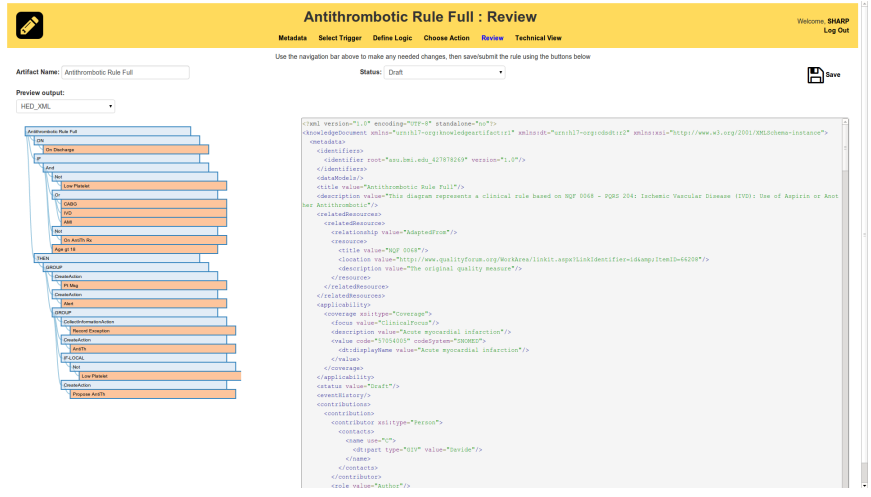


Figure 6. KA review and preview.

PRODUCTS

The products of this work included the team's contribution to the HeD modeling process itself and, specific to this project, the development and initial testing of the HeD knowledge artifact editor.

A beta version of the HeD editor has been released under the Apache Software License v2 Open Source license and is available at https://github.com/sharpc2b, where it is periodically updated. Documentation is also available at this location. The community is encouraged to extend this tool and contribute to the repository.

As of June 2014:

- The SHARPc2B/HeD core, derived ontologies, and an initial set of templates are available, possibly as independent models.
- The editor core allows import, authoring, and export of HeD-XML artifacts in a controlled way. The major components of HeD, such as metadata, expressions, triggers, conditions and actions (atomic, conditional and composite), are supported and can be authored simply.
- The authoring process is assisted and constrained both at the SME level using templates and the KE level, where the Blockly-based expression model allows for type validation and additional analysis.
- The editor core is based on the Play Framework so that it can be deployed in a cloud environment. It can also be packaged as a WAR application (a means of packaging and distributing a collection of JavaServer pages [362] and deployed in a web container.
- The editor UI is a pure web application distributed as a WAR file.
- To enable support for medical terminologies and value sets, a CTS2 service must be available and reachable by the editor. Recommended open source implementations are listed in the editor's documentation.

Known limitations:

- While the editor is model driven, both the domain models (such as vMR) and the template lists are processed at compile time and loaded when the editor is launched. The architecture would also allow loading of the resources at runtime, but this functionality is not yet exposed to users.

- The application backend is cross-platform. Likewise, the front end is a pure web application. However, the latter has only been tested with Google Chrome. Compatibility is not yet guaranteed with other browsers. Corollary services such as authentication, security, or a proper repository implementation have not yet been implemented in a robust way.
- The editor's UI should adjust to and constrain the user when authoring different types of artifacts, but this functionality has not yet been implemented.
- The editor has not yet been subjected to a QA process. While HeD artifacts are generally supported, fine-grained tests to ensure coverage for every possible element have not yet been performed to this date.

An example of the use of the HeD editor for a rule is the following, adapted from NQF 0068 for antithrombotic therapy on discharge of patients who have ischemic vascular disease, such as an acute myocardial infarction, or who have had a coronary artery bypass graft, or a percutaneous coronary intervention. This was originally developed as a quality measure by the National Quality Forum [363], but a proactive decision rule was created by NewMentor and provided as part of the pilot evaluation of the HeD model.

The quality measure rule conditions are: patients 18 years and older with ischemic vascular disease (IVD) who were discharged alive for acute myocardial infarction (AMI), coronary artery bypass graft (CABG) or percutaneous coronary interventions (PCI) from January 1 to November 1 of the year prior to the measurement year, or who had a diagnosis of ischemic vascular disease (IVD) during the measurement year and the year prior to the measurement year and who had aspirin or another antithrombotic during the measurement year.

Clinical concepts of AMI, CABG, IVD, and antithrombotic medications are defined by specified value set groupings published by NCQA, and maintained by the NLM VSAC. The flow chart for the proactive decision rule adapted by Motive Medical Intelligence (formerly NewMentor) as part of one the HeD Use Case 1 pilot projects is depicted in Figure 7.

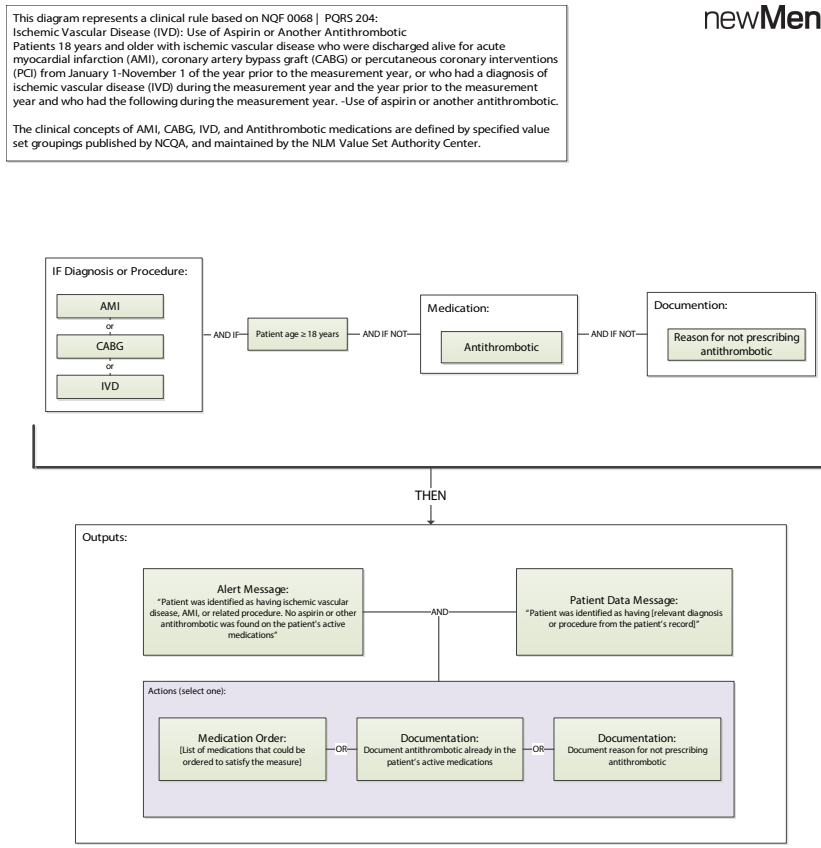


Figure 7. Informal specification of the NQF0068-derived CDS Rule. (Retrieved from http://wiki.siframework.org/file/history/open+house+_final_v2.pptx (New Mentor). Reproduced by permission.)

Using the editor, we modeled the trigger event as being in pre-discharge status. This could either be a status field in a patient's record or identified as a scheduled event to discharge within, say, 12 hours. For simplicity we use the trigger of patient status change to "pre-discharge" (Figure 8).

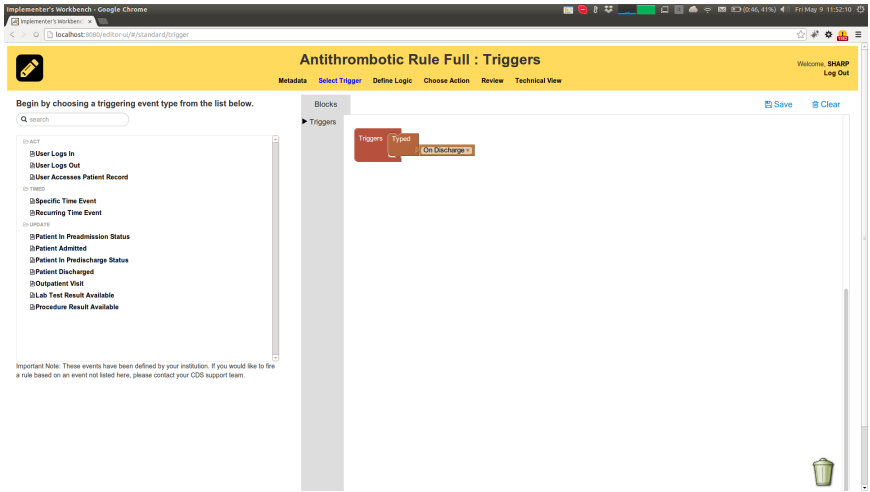


Figure 8. NQF0068 – Trigger, based on change of status to "pre-discharge."

We modeled the IF conditions as shown in Fig. 10. This shows a complex conditional expression being built up by a set of simple clauses, combined into AND ("all") or OR ("any"), and NOT clauses (Figures 9, 10).

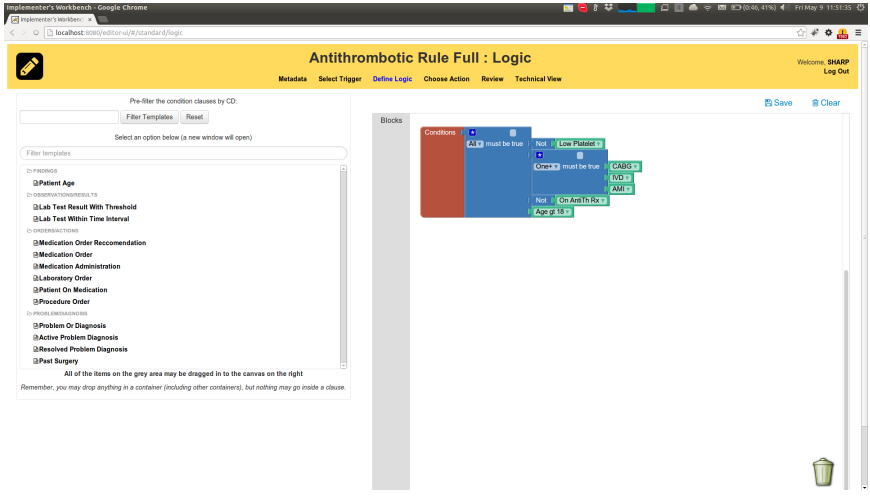


Figure 9. Conditional logic expression for the proactive rule addressing NQF0068.

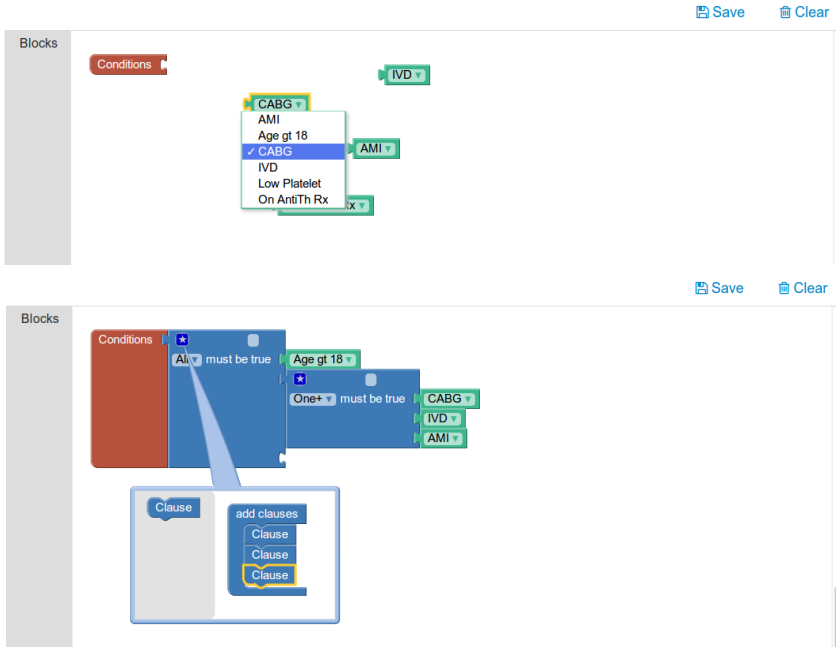


Figure 10a/b. Building up the conditional expression in Figure 9.

We modeled the ACTIONS by recognizing that there are two actions that should be done, plus "exactly one" of three other actions (a variant of an OR clause) (Figures 11). Note that actions steps can have conditions for them individually, which we have illustrated.

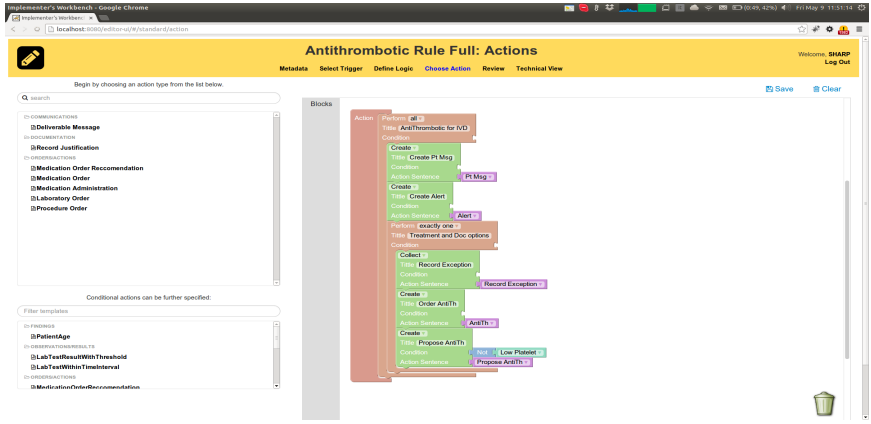


Figure 11. Actions corresponding to the rule addressing NQF0068.

The summary and its XML rendition are shown in Figure 6. At the time of this writing, the HeD editor still needs to go through additional usability testing and updated based on feedback. Once stabilized, it needs to go through production engineering and quality assurance testing. These latter tasks will require the help of a community of users and/or further external support.

DISCUSSION

The HeD editor as it exists is somewhat unique, in that it provides a non-proprietary editor that is both model- and standards-based. For reasons cited earlier, this provides a number of potential advantages for further development and extension, the production of adaptors to render the knowledge in various other output formats and representations, and the ability to use the technology for some of the other purposes described in the next section.

The main challenge currently is that the natural user constituency for this tool does not yet exist in strength. EHR vendors and knowledge vendors typically have editors for their own knowledge resources and have little incentive to use an external editor without the requirements of Meaningful Use Stage 3 (assuming that it ultimately requires the vendors to be able to import knowledge in HeD format). There are also no significant repositories of knowledge being created or maintained yet in HeD format, and the tasks of HeD did not address the creation of repositories. Indeed the primary other focus of the HeD initiative is to promote decision support as a service (HeD "Use Case II") by defining the characteristics of APIs for decision support based on the HeD model.

Healthcare organizations, especially those with more than one EHR vendor, may have some incentive to use a HeD editor. However, they would need to be assured that the knowledge artifacts could be translated into EHR-compatible formats and that appropriate data mappings could be imported and incorporated into an existing EHR system's knowledge repositories.

All of this could change dramatically, if the requirement to import HeD knowledge is indeed part of Meaningful Use Stage 3 and if considerable knowledge is distributed in that way.

FUTURE DIRECTION

The natural constituency for HeD KA exchange, and therefore authoring and editing of the KAs, does not yet exist because of the lack of regulatory or other stimulus to raise its importance to EHR, knowledge vendors and healthcare organizations. Anticipating that such stimulus will indeed be present in the near future, there are a number of enhancements that should be made to the tool.

First, if import and export of HeD KAs does become a requirement, the following activities would add to the tools value:

1. More templates for commonly used constructs (i.e., trigger types, conditional expression clause types, and action types) should be added to the template library. This could be targeted based on experience with authoring rules and analysis of them to determine the most frequent constructs. Templates could then be collected in libraries and enhanced with domain-specific extensions, e.g., for pharmacogenomics CDS.

2. Automatic selection or filtering of potential templates as a user begins to type a phrase based on concept type of the term being typed could be enhanced. Although the current version of the CTS2 terminology server doesn't include this, an approach could be to look up the term being typed in the UMLS [364] SPECIALIST lexicon, which contains most abbreviations, synonyms, and variants of clinical concepts. These point to a Concept Unique Identifier in UMLS. The various coding schemes used for that concept and the appropriate code(s) in them are also provided (e.g., for a medication, RXNORM is identified as a coding scheme and the RXNORM code for the medication is provided). In addition, the concepts have an associated semantic type in the UMLS [364] Semantic Network, for example, a medication would have a type "Pharmacologic Substance" or "Biologically Active Substance." A laboratory test result would have a type "Laboratory or Test Result." If the main concepts of templates are pre-associated with such semantic types, then entering a specific term, synonym, or abbreviation can be recognized as calling for use of templates with main concepts of the corresponding semantic types. The set of possible templates, e.g., those relating to conditions about laboratory tests, could thus be

offered to the user and guide the remainder of the clause construction, suggesting other relevant attributes that need to or could be specified.

3. The formal incorporation of useful libraries of value sets, such as those for the quality measures currently being assembled by the NLM VSAC, can be incorporated into the CTS2 server.

4. Additional definition capability might be considered as an extension of the HeD model. This would involve the ability to create a logical conditional expression that has as its action, if true, the assertion of a definition being true. For example, the presence of diabetes could be asserted for a patient by either the existence of an appropriate diagnosis code in the EHR, or by two or more abnormal HbA1c values, or by the existence of insulin or other hypoglycemic agent medications in the record.

5. Although HeD does not define a particular patient information model, most of the work to date has been done assuming that the model to be used will be based on the vMR. However, the surge of interest recently in the Fast Healthcare Interoperability Resources (FHIR) specification as an alternative model for both data representation and transport suggests an authoring system should be able to generate KAs capable of including FHIR-based data references. In addition, efforts by the Clinical Information Modeling Initiative (CIMI) may establish definitions of compound clinical concepts in which concepts have associated relevant attributes, based on the Clinical Element Model (CEM) approach pioneered by Huff et al. [365] at Intermountain Healthcare. There are now some efforts underway to expand FHIR profiles to include CEMs. If these efforts gain broad acceptance, it would be useful for the knowledge editor to support them.

6. To further the usefulness of a standards-based tool such as the HeD editor, it would be useful to create a set of adaptors to render created artifacts in other useful formats, such as Drools, Arden Syntax, and other rules engine languages.

In addition, the capabilities developed have a number of other potential uses if opportunities for funding and carrying out the necessary

work can be identified. We outline these here because of the potential value for those purposes, but also because they represent another route to stimulate development and use of the editor beyond waiting for the natural constituency for HeD artifacts to develop. These include:

1. *Quality measure authoring.* There is much overlap between the CDS proactive rules and retrospective quality measures. Measures look for whether something was done in a previous time period, whereas a CDS rule seeks to recommend an appropriate action that the quality measure will subsequently confirm. Thus, the logic itself may vary in the two circumstances, but they are companion artifacts. Quality measures, however, do not run on a per-patient basis, but rather compute numerators and denominators over a population. The trigger part of a rule needs to use time or other event triggers to initiate the computation, which then must range over a set of patients selected by the logical condition expression. The action part can be to determine the numerator or denominator, or the ratio of the two. Appropriate extensions to the HeD model and to the editor need to be created to allow for this. The HeD model can be extended to include calculation of measures over a population of patients, for example for computation of quality measure numerators and denominators. This is currently a very promising area of investigation.

2. *Knowledge management.* Another major opportunity is to adapt the editor to manage a repository of knowledge in HeD format. This would be particularly useful for national repositories, those of professional specialty or disease-oriented organizations, or health care organizations. In the latter case, it would be especially valuable if the healthcare organization has multiple EHR vendors in its enterprise and must manage the knowledge resources across these EHRs, and if adaptors can be developed to import the knowledge back into the EHR systems or if CDS can be provided as a service from an HeD-compatible CDS service. There are three aspects of the HeD model and editor that make this potentially useful:

    2.1. Because the HeD KAs are model-based, each KA concept (metadata, triggers, conditions, actions) has formal ontology-based semantics, associated attributes,

code systems, and code values, all of which can be used to index the KA. This makes it possible to readily search a repository for KAs pertaining to particular diseases, settings, uses, providers, etc. This facilitates maintenance, identifying duplicates and conflicting KAs, as well as gaps.

2.2. Because the modeling can be extended to include a variety of triggers, conditions, and action types, KAs can be characterized by a full set of SSFs, as was our original goal in this project. Many variations of the same basic rule exist in some large enterprises, owing to differences in workflow, local preferences, and other factors. Proper management of these variations allows them to be tracked, performance metrics to be applied to determine which configurations are most effective, and referenced to a catalog of existing variations for a given KA when an update is required, among other benefits.

2.3. Most knowledge editing in a health care organization is within EHR vendor proprietary knowledge environments, which must largely be done by KEs. The human-readable form of such knowledge is typically disconnected from the implementation, usually as separate human-readable documents, and may get out of synchrony if companion notes are not rigorously maintained every time the knowledge resource is updated. If the HeD editor is used, not only is the KE version synchronized with the human-readable version, but the SME can do much of the initial editing and specification directly.

3. *Knowledge access for context-aware use.* As we develop increasingly context-aware, situation-aware applications, that for example, know what the user is doing, what role/specialty the user has, what patient is being addressed and the patient's problems, and in what setting this is occurring, it may be possible to use the knowledge repository's indexing scheme to immediately retrieve the set of KAs most relevant to a current context and setting.

Implications

This work has provided an opportunity to develop a tool that we believe, if properly positioned, can be a foundational for future CDS knowledge representation, distribution, management, and incorporation into applications. Its current natural constituency is limited so far by a lack of appropriate stimulus or requirement for use, but assuming that limitation will be overcome, there is a broad collection of potential applications for this technology waiting.

Acknowledgments

We thank other members of the SHARPC project 2B team who contributed at various earlier stages of this work and provided technical assistance during the latter stages. Earlier contributors included Mary Goldstein, MD, VA Medical Center, Palo Alto, CA; Samson Tu, MS, Stanford University; Emory Fry, MD, Cognitive Medical Systems; David Yauch, MS, Banner Health, and software developers Matthew Ebert, Intermountain Healthcare; Edinardo Potrich, Intermountain Healthcare, and Randy Kerber, MS, independent consultant.

Suggested Reading

Musen MA, Middleton B, Greenes RA. (2014). Clinical Decision Support Systems, Chapter 22, in Shortliffe EH, Cimino JJ (eds). Biomedical Informatics: Computer Applications in Health Care and Biomedicine (Health Informatics), Springer; 4th ed. 2014 edition.

Greenes RA (ed). (2014). Clinical Decision Support, 2nd Edition: The Road to Broad Adoption. Elsevier, 2014.

21: Cognitive Support for Clinical Comprehension

SHARPC project 2A

Trevor Cohen, MBChB, PhD
The University of Texas Health Science Center at Houston School of Biomedical Informatics

Thomas Kannampallil, MS
The New York Academy of Medicine

Vimla L. Patel, PhD, DSc
The New York Academy of Medicine

ABSTRACT

We developed and evaluated a Cognitive Support System (CSS) that organizes clinical information to support decision making. Previous research showed organizing information is a distinguishing characteristic of expertise and a prerequisite to effective problem solving [366]. We elicited knowledge from experts and characterized their decision making processes. Decision models were then rendered computer-interpretable to organize patient data. CSS user interfaces were developed to present real and synthetic patient data. Studies comparing participant interpretation with and without CSS interfaces suggest CSS mediates high-level clinical data organization and is conducive to problem solving.

INTRODUCTION

Clinicians' attention to complex patient conditions is a precious resource too often consumed by the extra cognitive demands of information overload, time pressure, aggregating and synthesizing data from disparate sources. Research has shown that comprehension of a situation is a prerequisite to accurate mathematical [367] and medical problem solving [368]. The ability to organize information to facilitate rapidly generating solutions is a distinguishing characteristic of experts, including expert physicians [369]. The cognitive burden of organizing clinical information is compounded by the need to elicit, select, gather and integrate relevant data [370]. We propose a cognitive support system

(CSS) that uses supporting knowledge structures to help clinicians make effective decisions by organizing and aggregating information in a problem-specific manner.

The EHR as a cognitive artifact

The Electronic Health Record (EHR) is under scrutiny. Unprecedented in scale and scope, the Office of the National Coordinator (ONC) for Health Information Technology (HIT) launched initiatives to spur national HIT adoption. Potential benefits include significant cost savings and improved patient safety [371]. EHR implementation, however, carries the risk of unintended consequences, including errors [17, 174, 202]. While some errors have been attributed to socio-technical factors, others relate to technology's impact on clinicians' cognitive processes [372]. Unintended effects were demonstrated in a study comparing records produced by physicians with and without an EHR [251]. The EHR changed the nature of information recorded by physicians and the impact continued even after physicians stopped using the EHR. HIT's unintended effects can be positive or negative. Consequently, EHRs can be considered a cognitive artifact—"artificial devices that maintain, display or operate upon information in order to serve a representational function and that affect human cognitive performance" [373]. As cognitive artifacts, EHRs have an effect on human cognitive performance, planned or not. The design, implementation and deployment of EHR systems must, therefore, be closely aligned with the information processing skills and limitations of its users.

The perils of sub-optimal design

Consider the case of a potassium chloride overdose [255]. Intravenous (IV) medication errors are well recognized as a recurring cause of potentially harmful events [374]. Computerized Provider Order Entry (CPOE) systems have been proposed as a way to reduce errors [375]. In this case, however, the opposite occurred. CPOE screens for drip and IV bolus orders were superficially similar, yet required different mental calculations to estimate dosage. Orders for IV bolus doses were specified by *dose*. Orders for other IV administrations were specified by *duration*, but labeled *Total Volume* on the entry screen. Total Volume was meant to indicate size of IV bag, not total amount of fluid to be delivered

to the patient. IV fluid orders were not displayed on the medication review screen, further complicating bolus calculations. Calculating correct infusion dose is an essential task. Unfortunately, the CPOE system did not provide tools to assist with this process, and its interface design was a poor fit for the conceptual operations clinicians use calculating dosage (i.e., volume vs. duration). Research literature is littered with examples demonstrating how poor interplay between technology and clinical decision making can negatively affect health care [17, 95].

Characterizing cognition

The overdosing case exemplifies a larger problem: current EHR systems are not always designed to support clinical decision making. Many serve primarily as media for information storage and retrieval, and are not aligned with mental processes underlying clinical decisions. While the idea of engineering systems to accommodate the constraints of a given task and those of the human cognitive system is not new [376, 377], the approach taken in our research differs in focus and methods. Our work is rooted in decades of medical decision making and reasoning cognitive research [369, 378-380]. Theories and methods allowed us to characterize knowledge structures, conceptual operators and decision strategies underlying clinical tasks.

We based reference models on how experts organize information, since experts have shown superior abilities to generate representations conducive to problem solving [381], including medical problems [382]. Organizing clinical information in an expert-like manner enhances the clinical comprehension of less experienced practitioners [383]. However, the extent to which expert representations must be adapted to facilitate non-expert comprehension is controversial and should be empirically evaluated. A good example of expert cognition as a basis for expert-like performance is provided by Staszewski's research on landmine detection [384]. A training program based on expert strategies dramatically improved performance in detecting low metal mines, from approximately 10% to greater than 90%. Spatial representations of mines was found a distinguishing characteristic of expert mine detectors. Staszewski's group also developed an augmented display that represented buried mines from auditory signals emitted by a mine detector [385, 386].

We researched how an automated system with patient data organized according to expert knowledge structures impacts clinical decision making and reasoning.

APPROACH

Overview

Our approach has four stages (Figure 1): 1) Expert decision models are created to characterize how domain experts organize information to solve a clinical problem of interest, 2) Decision models are rendered computer-interpretable to organize information drawn from real-world or simulated clinical data, 3) A functional CSS prototype backend is created, 4). User interfaces that draw on the CSS backend to arrange clinical data are developed and evaluated.

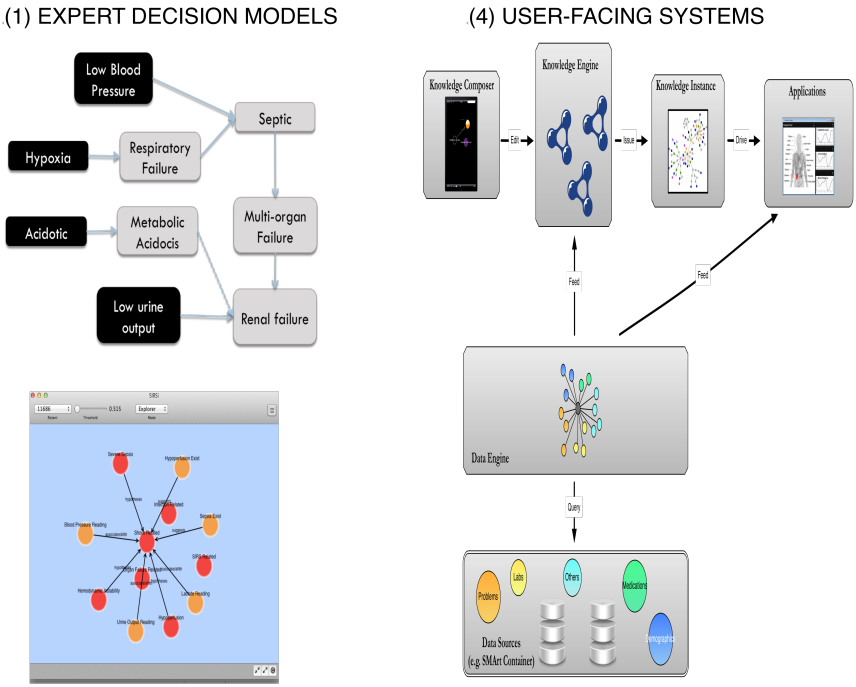


Figure 1. Overview

Stage 1: Characterizing cognition

Systemic Inflammatory Response Syndrome (SIRS) is an inflammatory state with ominous prognostic implications. Clinically deciding if a patient exhibits SIRS is a sub-component of a larger assessment process, including the probability of underlying infection and likely infection causes. For the purposes of illustration, we restrict our focus to the question, "is this SIRS?" Despite a well-defined set of criteria for SIRS [387], this decision point is considerably more complex than it appears.

Our approach characterized knowledge structures and cognitive processes underlying a decision. Of particular interest is how experts facilitate problem solutions. For example, in the reasoning of a master chess player, the cluster of pawns, rook and king making up the powerful defensive "castled-king position" is perceived as a unit [388]. In the literature on medical cognition, this structure is referred to as *intermediate constructs* [389]. Expert physicians are distinguished by their ability to recognize these constructs [390] to facilitate efficient and accurate diagnostic reasoning. In medicine, intermediate constructs are meaningful clusters of clinical findings (e.g., "right-sided cardiac failure" or "respiratory problem"), which are not in themselves diagnostic, but partition the search space of possible diagnoses. Cognitive processes of interest include comparison or generation of new hypotheses during diagnostic and therapeutic reasoning.

The excerpts in Table 1 are drawn from an in-depth knowledge elicitation session involving two intensivists. Participants were encouraged to discuss the condition and generate a visual representation of their knowledge on a whiteboard, which was captured for further analysis. The intensivists were told to steer the discussion toward intermediate constructs. SIRS constitutes one possible intermediate construct.

Excerpt 1	Physician 1: "…and white count is greater than 12 but less than 4…" Physician 2: "Right." Physician 1: "…and fever is greater than 38 or less than 36."
Excerpt 2	Physician 2: "It would, so there's variations where obviously if you have heart condition, you may not get to a heart rate of 90, um, if you are on sedative drugs."
Excerpt 3	Physician 2: "…over the course of 2-3 days if it goes up you start worrying about it."

Table 1. Excerpts from a knowledge elicitation session.

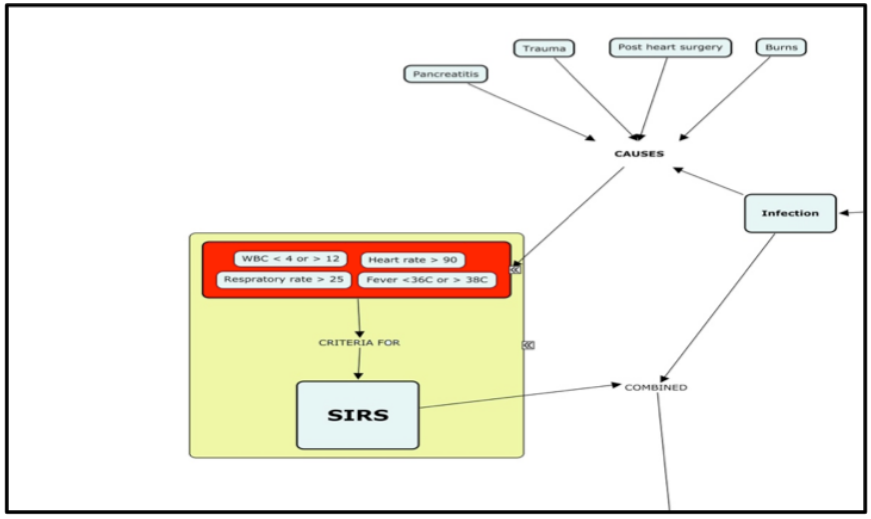


Figure 2. An excerpt from the SIRS Concept Map. The four relevant criteria that trigger a possible SIRS diagnosis are shown within the shaded box.

Sessions were recorded, transcribed and key sets of concepts captured as Concept Maps [391] using the CMapTools software package [392] (Figure 2). Included are SIRS diagnostic criteria, such as reference ranges for white blood count, heart rate, respiratory rate and temperature. Note that this is a component of a larger model that includes prognostically important factors such as presence of organ failure.

The decision making process was characterized at a granular level by applying a coding system from previous research in pediatric cardiology [380]. We isolated and coded conceptual operators used to reach the categorical decision whether this constitutes a case of SIRS or not. For

example, the first excerpt in Table 1 illustrates the compare-to-norm conceptual operator where an observed clinical parameter is compared to reference ranges included in official SIRS criteria. The second excerpt illustrates a compare-to-expected operator in which mitigating circumstances affect the extent to which usual reference ranges apply. The third excerpt illustrates a compare-over-time operator, which may provide an early indicator that SIRS is approaching.

The characterization of a decision informs the presentation of information supporting it. Required data elements are identified displayed without extraneous details distracting from the decision. Conceptual operators suggest visual representation of the data. For example, compare-to-norm requires rapid assessment of the relationship between a data point and an established norm, and could be facilitated by an annotated bar graph, while compare-over-time suggests a trend line over time. In contrast, compare-to-norm suggests integrating additional information, in this case the dose of a specific class of medications.

Stage 2: Computer-interpretable models

Information generating a decision model-based interface must first be rendered computer-interpretable. Decision models developed in Stage 1 were granular and included a range of concept relations that could be rendered computer-interpretable for cognitive support. A core set of conceptual relations were identified and reduced to a set of Resource Description Framework (RDF) triples, rendering them amenable to automated reasoning with semantic web technology.

Core relations were identified by studying verbal think-aloud protocol transcriptions collected in a Medical Intensive Care Unit [370]. This ensured the identified set of conceptual relations were sufficiently general to apply to a range of clinical decision tasks. Data consisted of think-aloud protocols generating a description of a patient case (including collection of prerequisite data). Participants in this study involved eight ICU physicians, each of whom accounted for one case. Verbal protocol data of this nature has been widely employed to characterize cognitive processes underlying decision making [393]. We analyzed the protocols to define a set of relations sufficient for cognitive support.

Transcribed data were analyzed in three stages using a grounded theory approach [394]. The first pass (open coding) deconstructed transcripts and phrases for identification and categorization (e.g., urine analysis, "make us think", source of infection, CT chest, "make sure", etc.). Then, coded key phrases were grouped into categories (axial coding) according to form an initial coding framework (e.g., vital signs, diagnoses, medication orders, suggestions, causes, etc.). Categories were integrated into higher-level themes (selective coding) to form a structured conceptual framework, which described physicians' cognitive processes during medical problem solving (e.g.,observations, interpretations, suggests, explains, etc.). A constant comparison approach was subsequently adopted in which concepts deconstructed from each transcript were analyzed and either integrated into the existing coding scheme or resulted in the emergence of new codes which were then added to the overall scheme.

Coders with medical and informatics backgrounds recorded details of how coding decisions coding were made and how codes were linked to the original transcripts. This provided justification for the creation of codes. Debriefing sessions involving another member of the research team, also with a medical background. Coders discussed the rationale for code development. If disagreements arose, each would elaborate reasons for their decision until a consensus was reached (Table 2).

Deterministic	Probabilistic	Consequential	Temporal	General
Causes	Increases likelihood	Accentuates	Follows	Associates with
Confirms	Reduces likelihood	Has effect		
Contraindicates		Has risk		
Explains		Has sign		
Is consistent with		Prevents		
Rules out				
Suggests				
Treats				

Table 2. Summary of 16 relationships identified and included in the framework categorized into 5 classes.

A relations set validation study was conducted using case reports from the Clinical Problem Solving feature of the New England Journal of Medicine (NEJM-CPS) [395]. NEJM-CPS articles provide interpretations with detailed explicit description of a contributing expert's reasoning, instead of only reporting clinical details. The capacity to represent these explanations would support the cognitive utility of our core set of relations. To assess this capacity, graphical representations of five NEJM-CPS case explanations (Figure 3) were evaluated for completeness by two domain experts. Ratings were generally positive (mean=9/10), supporting that our relation set was sufficient for a broad array of clinical reasoning tasks.

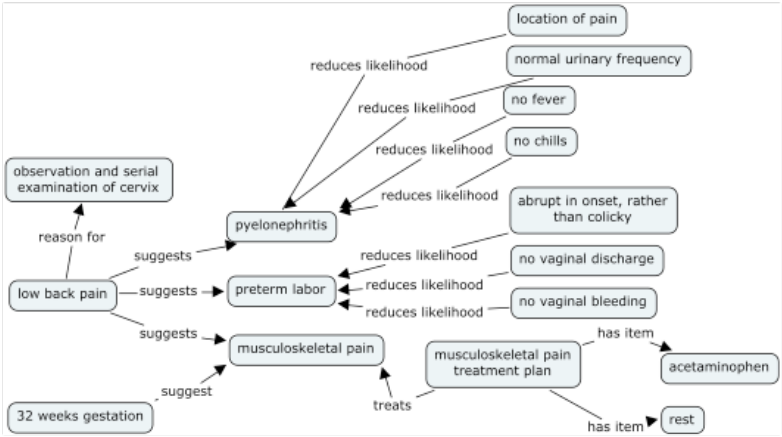


Figure 3. From the NEJM-CPS article, "A Problem in Gestation" [396]. The graph was created using CmapTools [392] from: "Labor is rarely described as abrupt in onset, is usually colicky in nature, and is often associated with other symptoms or signs, such as blood tinged vaginal discharge. When there is doubt regarding the cause of the pain, observation and serial examinations of the cervix for evidence of change are helpful. Musculoskeletal pain is common, and its likelihood increases as pregnancy advances, owing to weight gain, the loosening of connective tissues with the hormonal changes of pregnancy, and the shift forward in the woman's center of gravity. Pyelonephritis is a concern with an abrupt onset of back pain, but it is unlikely in this patient, given the reported location of the pain and the absence of fever, chills, urinary frequency, and dysuria."

This core set of relations was used to encode the SIRS-related decision model in a computer-interpretable form. Our "MedSense Knowledge Engine" uses semantic web technologies to represent model concepts and relations as a set of RDF triples and makes these services available to other programs. Figure 4 illustrates concepts and relations

from a RDF graph based on our SIRS decision model and relations in Table 2.

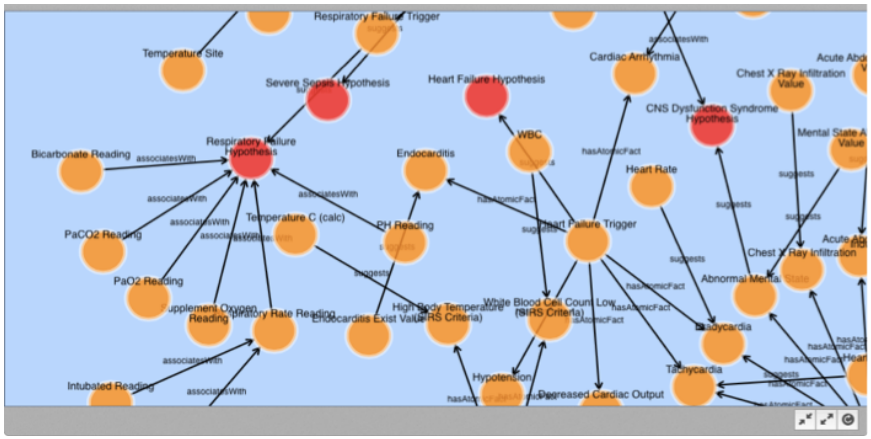


Figure 4. Visualization of SIRS relational knowledge elicited and stored using our MedSense Knowledge Engine. Darker color vertices represent unifying hypotheses. Lighter color vertices represent lower-level information. Edge labels represent type of relationship. The "hasAtomicFact" link is outside the framework, but is included for grouping purposes.

Stage 3: Cognitive support system

The MedSense Knowledge Engine was developed as a modular component of our CSS to allow knowledge models to be interchanged for different problems. Though our focus is on the biomedical domain, conceptual relations are sufficiently general in nature that, given an appropriate data source, the same system could be applied to problems outside of medicine (Figure 5).

Key functions are: 1) aggregation of medical data from various sources (e.g., database or SMArt compliant sources), 2) translation of data into a consistent representational format, 3) organization of data in a problem-appropriate flexible knowledge representation based on knowledge modules of expert clinical decisions, and 4) a visual module for representing data and results.

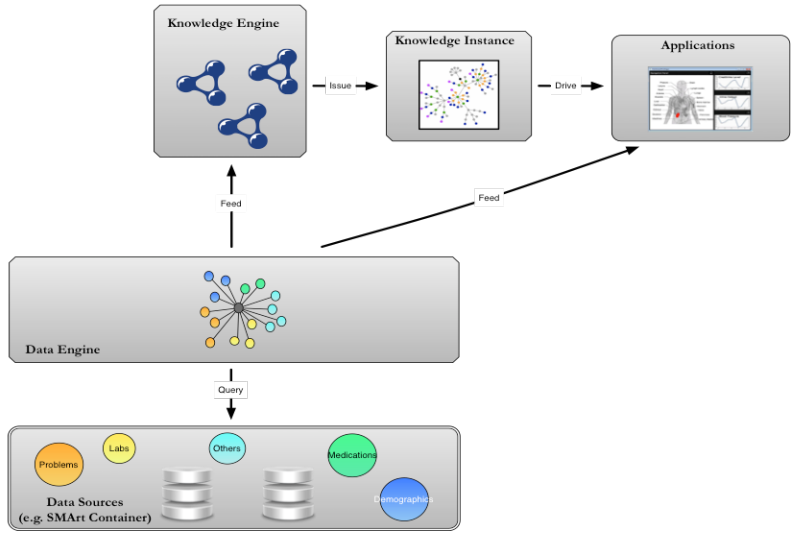


Figure 5. CSS architecture for clinical decision support showing three main components: data engine, knowledge engine, and user interface.

Key components of the CSS architecture are:

1. Data sources and data engine

Although we use a generic data source in the current version of our system, we previously used SMArt platform, a SHARP-funded effort to promote development of applications that could function on multiple EHR platforms [397]. We extended the model to incorporate the ability to retrieve data from multiple sources, including SMArt or similar platforms, to allow integration of MIMIC-II [398] data. MIMIC-II is a large-scale database of de-identified intensive care unit (ICU) data available under license that provides an invaluable source of physiological data for organization in accordance with our knowledge models. Without MIMIC-II, it would be difficult to evaluate the effect of knowledge organization because simulating ICU cases with high fidelity is time-consuming given the large number of synthetically estimated data points needed.

The core of our data engine is an abstract ontology model that describes raw clinical information with flexible metadata, such as class, value, value unit, normal value range, and possible interpretation of the value. Ultimately, patient data will be mapped onto the information model and exposed to the rest of the system through a set of application

programming interfaces. One of our design goals is a source-agnostic engine to interface with clinical data provided by various sources regardless of information exchange protocol.

2. Knowledge engine

Our MedSense Knowledge Engine contains computational representations of the knowledge elements and conceptual operators required to support key decisions. The engine encodes information, such as which data elements (e.g., respiratory rate, heart rate, fever and white cell count) occur together as meaningful clinical patterns (e.g., intermediate construct, SIRS) to support a particular decision (e.g., "is this SIRS?").

3. Applications

This component contains rules governing aggregation and display of data elements according to conceptual operators required at decision points. The view controller takes MedSense input data, decisions and conceptual operators and produces recommendations for on-screen display that can interpreted on different platforms (e.g., PC, iPad or Microsoft Surface).

Stage 4: User-facing systems

Completing our CSS backend allowed for development of front-end applications, which were based on prototypical designs developed by an in-house interaction design specialist working in collaboration with our research team. Two front-end applications emerged, both named SIRSi. The first was a prototype iPad app (SIRSi-iPad). The second was a web browser-based interface (SIRSi-Web) supporting switching between conventional and knowledge-driven representations of the same MIMIC-II patient data to allow comparing the effects each representations for research purposes. With both applications, case-related information is presented graphically due to the the prominent role of temporal trends (e.g., "white cell count is increasing").

1. SIRSi-iPad

SIRSi-iPad (Figure 6) allows users to explore de-identified patient data from the MIMIC-II database. Data points relating to SIRS-related decision making points are presented together. Navigation is guided by

hypotheses likely triggered by data elements. Navigation uses a screen-swipe control, which leverages the iPad's gesture-based interface. Data point details are revealed using pinch-to-zoom.

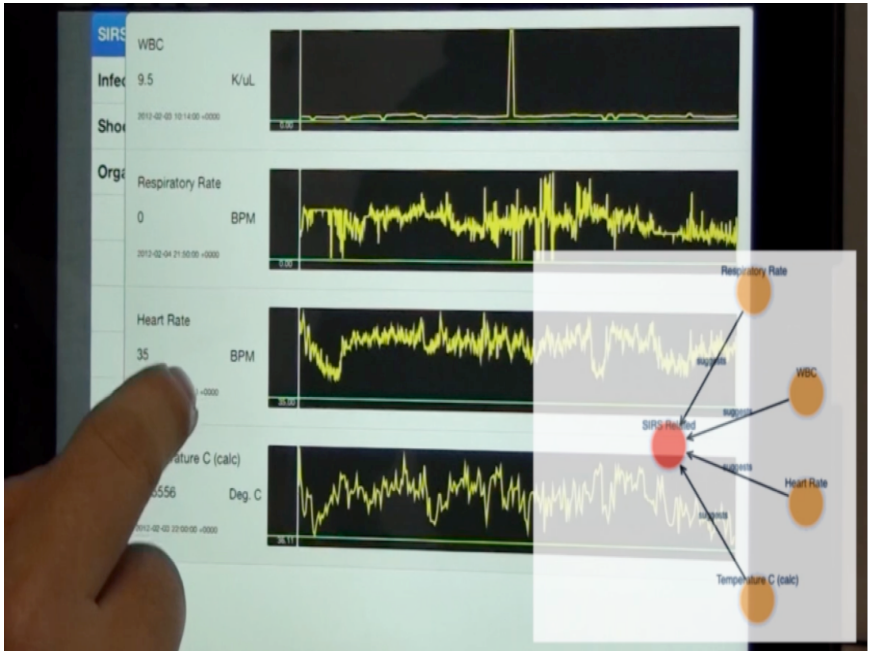


Figure 6. SIRSi-iPad showing de-identified patient data from the MIMIC-II database. The knowledge model shown inset is for illustration purposes and is not part of the interface itself).

2. SIRSi-Web

While the iPad's advanced user capabilities allowed development of novel knowledge-driven interface, we also developed and evaluated a browser-based version that better approximates contemporary clinical record systems. This interface used Google's Web Toolkit (http://www.gwtproject.org/) for graphical representation of data points. The current version of SIRSi-Web can switch from a knowledge-driven organization of data to a conventional arrangement based on source and lab panel.

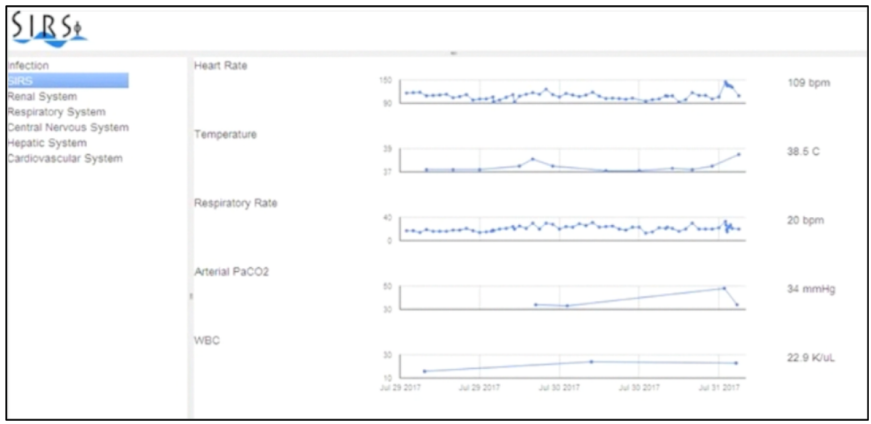


Figure 7. SIRSi-Web showing de-identified patient data from the MIMIC-II database organized in accordance with SIRS knowledge model.

We developed a psychiatry CSS during previous research [399], but did not evaluate it for clinical comprehension at the time. The psychiatry CSS categorizes text [400] to draw associations between short segments of a clinical narrative and four diagnostically and/or prognostically relevant intermediate constructs: "psychosis," "mood," "substance abuse" and "dangerousness" (Figure 8).

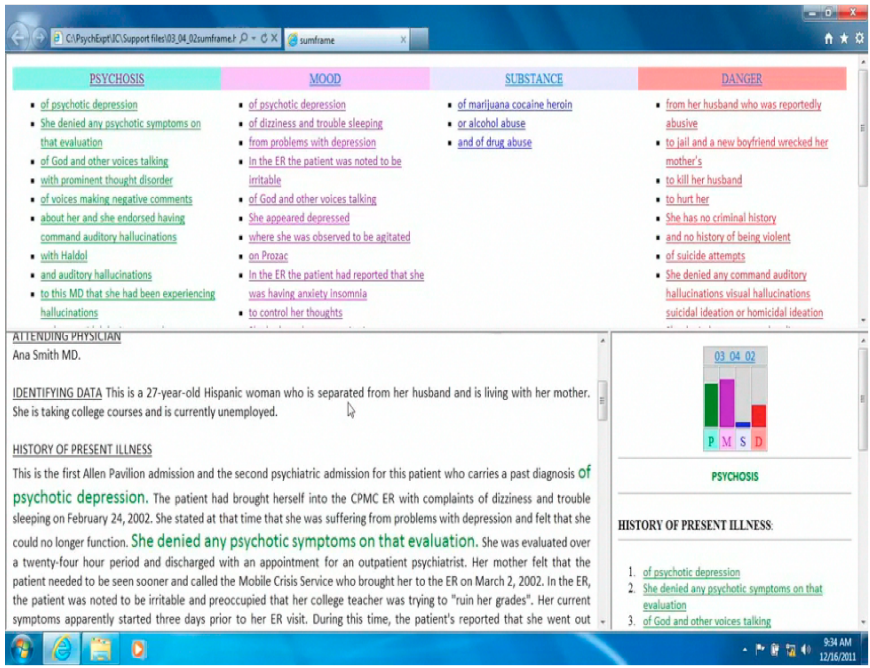


Figure 8. The psychiatry CSS interface. Aspects of the narrative deemed relevant are listed at top and highlighted if the intermediate construct or any of its elements is clicked. Clicking a text element scrolls the narrative to its location.

RESULTS/PRODUCTS

These interfaces demonstrate ways expert knowledge can be leveraged to support clinical decisions. In the following sections, we present results of some experimental work conducted during the course of the project. We will describe results of observational studies from information gathered in an ICU practice, studies comparing the cognitive process of psychiatry CSS system users to those without cognitive support, and results from a similar analysis of SIRSi-Web.

1. Information Gathering in the ICU

We found physicians extensively relying on paper records in an observational information gathering study of eight Medical Intensive Care Units (MICU) [370]. Physicians spent equal time on paper and electronic records during decision making. Yet, an analysis of audio-recordings revealed electronic records afforded more structured information and more non-redundant information. We characterized level of abstraction differences between information types using an epistemological framework by Evans and Gadd [401] previously applied to characterize clinical dialogue [402] (Figure 9). Electronic records were used to retrieve low-level information, such as clinical observations (e.g., patient had a chest pain) and findings (e.g., aggregated results from multiple tests). High-level representations such as intermediate constructs (or facets) and diagnoses made up a larger proportion of information retrieved from paper records.

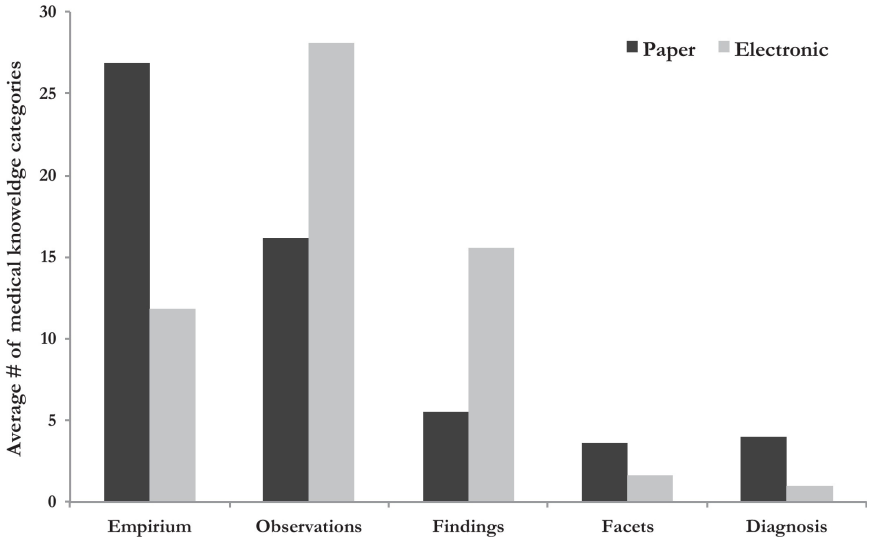


Figure 9. Information retrieved from paper and electronic records as portrayed in Kannampallil et al., 2012 [370].

Data suggest cognitive support for decision making in the ICU is currently sub-optimal. Clinical data are aggregated according to data sources (e.g., EHR, paper notes) rather than meaningful decision model. Electronic data sources played a minimal role supporting higher-level cognition.

2. Psychiatry CSS

We studied the performance of 16 psychiatry residents interpreting two clinical cases developed in previous research [383, 403]. Half of the residents used the psychiatry CSS interface and half did not. Verbal think-aloud protocols were collected during evaluations, allowing a fine-grained analysis of cognitive processes. A qualitative analysis revealed five ways in which the system was used to mediate clinical decision making [403]:

- *Hypothesis evaluation:* The interface was used to evaluate hypotheses generated while reading narrative text summaries. Residents reviewed related information organized at intermediate construct levels or reviewed highlighted facet-relevant components.
- *Leveraging text juxtaposition:* Sequential organization of information associated with interface elements at facet level led to the generation of new diagnostic hypotheses. A similar strategy occurred when text

highlighted by the interface and juxtaposed text contained narrative rich in diagnostically useful information, which lead to the generation of facet-level diagnostic hypotheses.

- *Review to exclude:* For the sake of completeness, the interface was used to rule out alternative diagnostic hypotheses by reviewing elements organized at facet level.
- *Review to confirm:* The interface was used to confirm thoughts and recollections by reviewing the findings organized at facet level (both in the interface and highlighted text).
- *Facet-level preview:* Facet-level elements were reviewed before narrative text was read.

With the exception of two outliers (IC-6 and IC-8) with sparse think-aloud protocols, residents in the IC group attended better to nine clinically important points selectively attended to by experts in previous research [383], particularly those highlighted by the system (Table 3). Nine key points were either highlighted by the interface (IC) or gathered during interpretation of cases without the CSS interface (NO-IC).

	IC								NO-IC							
SUBJ	2	4	6	8	10	12	14	16	1	3	5	7	9	11	13	15
WC1	613	761	65	88	1296	1373	1058	600	274	296	215	264	995	1336	1115	356
1A	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1B	✓	✓			✓	✓	✓	✓			✓		✓	✓	✓	
1C		✓			✓	✓	✓		✓	✓			✓	✓	✓	
1D		✓			✓	✓		✓	✓	✓			✓	✓	✓	
1E	✓	✓			✓	✓	✓	✓					✓	✓		
1F	✓	✓			✓	✓	✓	✓					✓	✓	✓	
WC2	371	925	0	174	1434	827	1002	820	193	0	232	591	614	1799	1104	128
2G		✓			✓	✓	✓	✓						✓	✓	
2H	✓				✓	✓	✓				✓			✓	✓	
2I	✓	✓			✓	✓	✓						✓	✓	✓	

	Proposition(s)	Facet	Clinical significance
1A	Dizziness, trouble sleeping	Mood	Differential – somatic disease
1B	Patient irritable	Mood	Irritable mood (mania/depression)
1C	Preoccupied that college teacher was trying to ruin her grades	None	Paranoid delusion: symptom of schizophrenia and related d/o
1D	Shopping frequently	None	Textbook manic symptom
1E	Prominent thought disorder	Psychosis	'Class A' symptom of schizophrenia and related d/o
1F	Command hallucinations to kill herself and her husband	Danger	Indicates potential dangerousness
2G	Content of command auditory hallucinations: command "to kill herself by cutting her wrists"	Danger	Indicates potential dangerousness
2H	Flashback to a past sexual assault	Danger	Symptom of post-traumatic stress disorder
2I	Racing thoughts	Mood	Textbook manic symptom

Table 3. From Dalai et al., [403]. Comparison across points of "divergent recall" characterized by Sharda et al. 2005. The top half of the table is derived from the think-aloud protocol captured during exploration of the case. SUBJ=subject number. WC1= protocol word count for case 1 and so forth. ✓indicates recall of the proposition(s) concerned. Grey cells indicate

no think aloud data was produced by this participant during interpretation of the case. 1A=proposition(s) A for case 1 and so forth. White text on a black background indicates this information was highlighted by the system. The bottom half of the table describes propositions highlighted by Sharda et al., as well as their clinical significance and which facet, if any, they were grouped by the system.

Points 1F and 2G suggest the concern was potentially dangerous. Failing to attend to these points could have dire consequences. More users of the system suggested the correct diagnosis in more complex cases (three IC vs. one NO-IC). However, more of the group not using the system suggested the correct diagnosis in simpler cases (four IC vs. six NO-IC). Users of the CSS system were less prone to discuss psychosocial aspects of cases when summarizing. This is perhaps not surprising, as psychosocial aspects were not highlighted by the system, which was originally developed for emergency psychiatry acute care assessment, but nonetheless illustrates the double-edged nature of cognitive support—directing attention to one aspect of a case may lead to the neglect of another.

3. SIRSi-Web

We are currently evaluating the results of a SIRSi-Web interface study on participant reasoning. This differs from the psychiatry study in several respects. Because of performance variation across individual clinicians, we used a within-subjects design to evaluate performance of the same individuals with and without cognitive support. For this, we developed two case scenarios based on de-identified patient data from MIMIC-II. Cases are of similar complexity and involve SIRS/sepsis patients. Order of case presentation and case selection varied across participants to mitigate for learning and case-specific effects. Twenty participants, including residents, fellows and attending physicians were recruited. Each interpreted two cases, one with and one without cognitive support. Participants were encouraged to think aloud while interpreting and navigating available data. Audio recordings and video screen capture were made using TURF (Chapter 6). SIRSi-Web includes a granular logging system that records and timestamps each mouse-click or mouse-over event. The latter is particularly informative as mouse-over actions can be used to reveal the exact value at a particular point in any of the graphical displays of laboratory and other results. This makes it possible to synchronize high-level thinking revealed in think-aloud data with low-level observations. Figure 10 shows synchronization of data from

different sources. Intermediate constructs (facets) identified during clinical reasoning are labeled "5" and highlighted. Figure 11 shows a higher frequency of constructs during the course of reasoning using the interface for two participants.

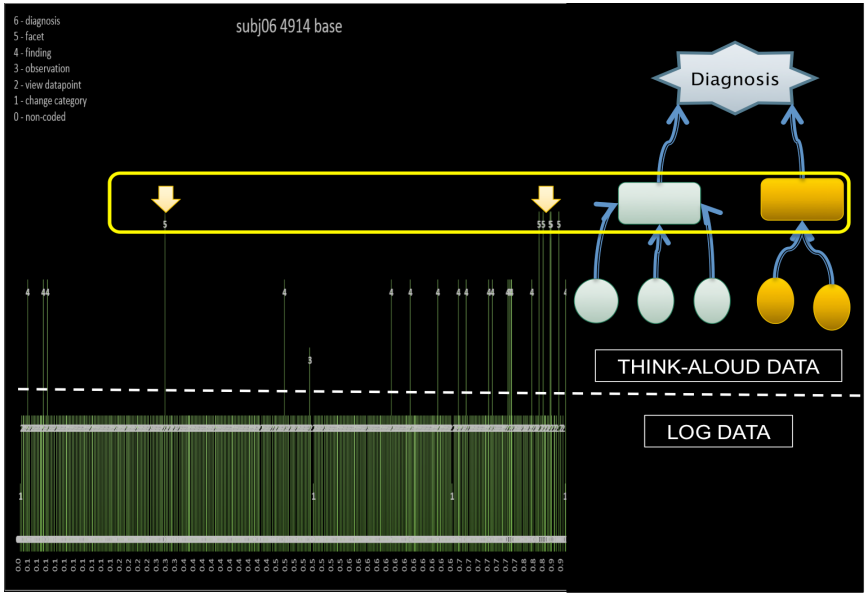


Figure 10. Synchronization of log data and think-aloud data.

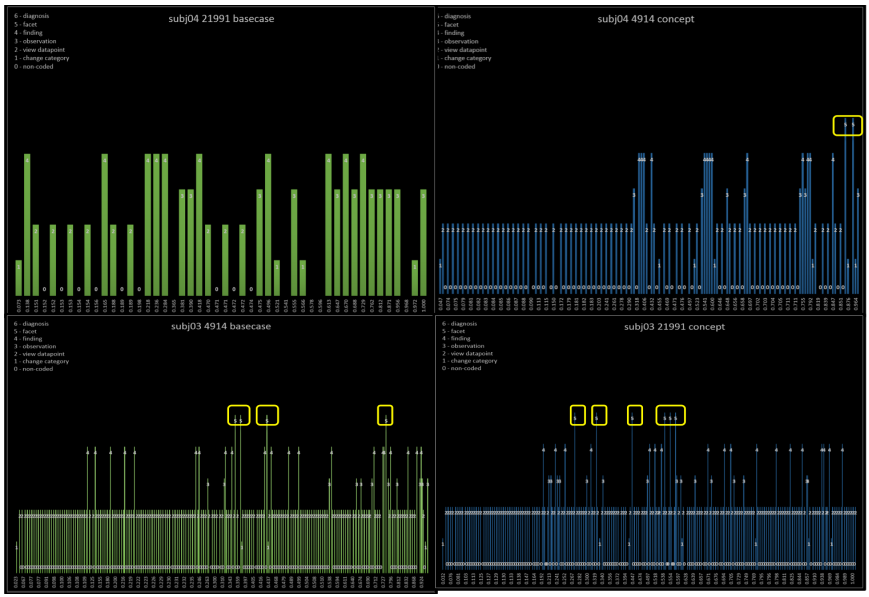


Figure 11. Higher level reasoning with SIRSi-Web interface (right) than without interface (left) for two participants.

Figures 12 (without CSS) and 13 (with CSS) show stratified counts of aggregated think-aloud and log data. Each count represents the number of times a participant moused over, clicked on or spoke about data for each of the data types available in the system. For many data points, distribution across participants is even with cognitive support. This is most striking for "heart rate" data, where without cognitive support the two expert participants (2,6) attended to the data more frequently than their non-expert counterparts (3-5). With cognitive support, non-expert participants expended more attention on heart rate. The number of mentions or accesses approximated the expert participants. Expert participants were more consistent across interfaces, suggesting they depended more on their own knowledge structures. These, however, are preliminary results. It remains to be seen if these patterns hold across all twenty participants.

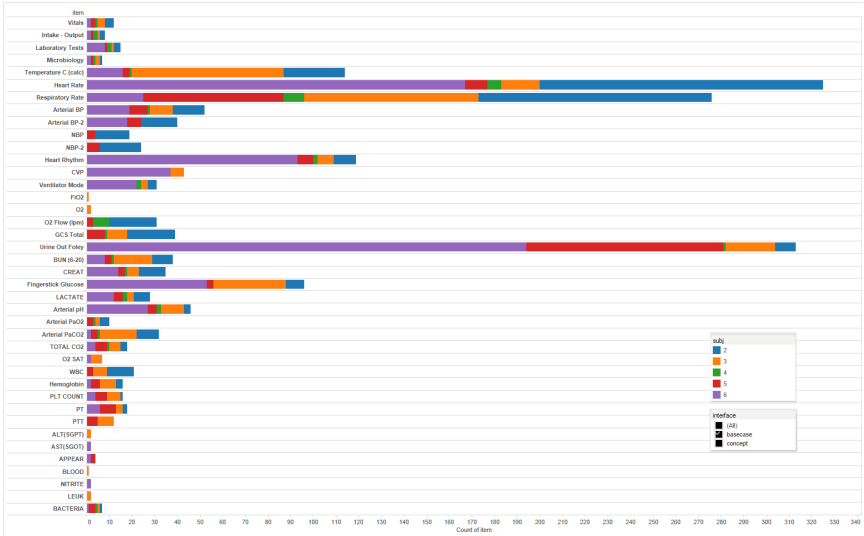


Figure 12. Counts of access to or mention of participant data points for one of two cases, this one without cognitive support. Participants #'s 3-5 are residents. Participant #2 is a fellow. Participant #6 is an attending physician.

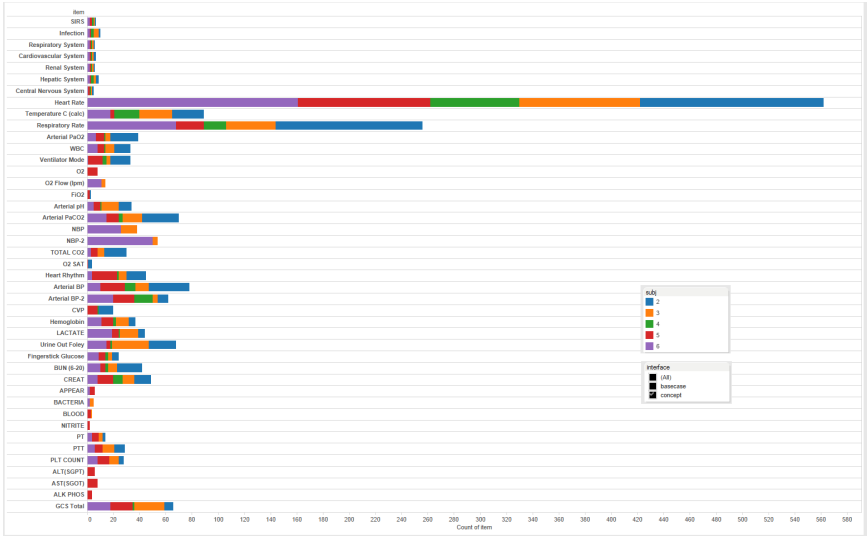


Figure 13. Counts of access to or mention of participant data points for one of two cases, this one with cognitive support. Participants #'s 3-5 are residents. Participant #2 is a fellow. Participant #6 is an attending physician.

FUTURE DIRECTION

We plan to complete analysis of SIRSi-Web, believing data will enhance our understanding of the impact of cognitive support in

practice. We also plan to continue exploring options for embedding our CSS interface, or aspects of it, within an active EHR. This will provide the means to evaluate the effects of cognitive support on sepsis-related downstream indicators of healthcare quality, such as time to treatment. Demonstrating positive effects will help realize the translational potential of our work.

IMPLICATIONS

Our CSS assumes some cognitive burden of clinical comprehension by organizing key clinical information in accordance with expert-derived reference models. Based on decades of research in medical and general expertise [381, 404], we drew inspiration from work on training personnel in landmine detectors [384]. Staszewski coined the term "Cognitive Engineering Based on Expert Skill" (CEBES) for this paradigm. Earlier demonstrations of CEBES effectiveness exist, perhaps the most famous being in the domain of mnemonic expertise [388]. Landmine detection also lead to a prototype system to assume cognitive work of expert decision makers [385].

We developed a CSS prototype and evaluated a psychiatry CSS [400] that encode aspects of knowledge structures underlying expert decisions. A system that could assume some of the cognitive burden of expert comprehension relates to the theory of distributed cognition [405], which views cognition as the product of a distributed system involving human actors and the external media supporting them in their cognitive tasks. Rather than being confined to the mind of a single clinician, clinical comprehension can be viewed as a distributed process involving, for example, a human reader and a textual display. Comprehension involves the construction of a mental representation of a clinical case influenced by the structured knowledge stored in the mind of the clinician [390, 406]. By organizing information presented in accordance with a simulation of expert knowledge structure, a CSS can redistribute part of the expert comprehension cognitive work from man to machine. Our research confirms this is possible. Our baseline studies of information from MICUs confirm some current electronic systems do not support the organization of clinical information in accordance with higher-level knowledge structures [370]. Our studies of cognitive support

effects in psychiatry [403] and SIRS/sepsis reveal numerous examples of reorganization effects on diagnostic reasoning.

In the case of the psychiatric CSS, participants were able to navigate through the summary to investigate diagnostic hypotheses and review information organized thematically. This appears to have improved their ability to attend to clinically-relevant elements, including evidence that suggested the patient might be harmful to themselves or others. Preliminary findings from our studies of the SIRSi-Web interface suggest the same participants may generate more facet-level (i.e., intermediate construct based) pre-diagnostic hypotheses than when exploring patient data using an approximation of a conventional interface. However, redirection of attention may come at the expense of neglecting aspects of the case not highlighted by the system. For example, users of the psychiatry CSS generally paid less heed to psychosocial aspects. Ideally, the design of CSS interfaces should be based on a robust canonical model of the decision task at hand. Any attempt to support decision making through intelligent organization of clinical data is arguably an advance over organizing this information arbitrarily.

This project resulted in a number of software and design products, including a sepsis user interface, a flexible CSS system architecture, and prototype front end interfaces. The SIRSi backend and web interface are available under an open source license with desirable research features, such as: 1) the ability to replace the sepsis knowledge model with another model of interest, 2) the ability to switch between a knowledge-driven and default presentation of the same clinical case, and 3) a granular logging system to capture user attention foci. Note that none of these features are necessary to incorporate cognitive support into existing systems. It is sufficient merely to replicate the organization obtained through application of the decision model. Many contemporary EHRs are sufficiently configurable to accommodate restructuring to some degree. Our clinical collaborators are currently engaged in reconfiguring vendor system to include aspects of our decision model for sepsis.

Suggested Reading

Patel, V. L., Arocha, J. F., & Kaufman, D. (1994). Diagnostic reasoning and medical expertise. In D. L. Medin (Ed.), The Psychology of Learning

and Motivation: Advances in Research and Theory (Vol. 31). San Diego: CA Academic Press.

Kannampallil, T. G., Franklin, A., Mishra, R., Almoosa, K. F., Cohen, T., & Patel, V. L. (2013). Understanding the nature of information seeking behavior in critical care: implications for the design of health information technology. Artif Intell Med, 57(1), 21-29. doi: 10.1016/j.artmed.2012.10.002.

Sharda, P., Das, A. K., Cohen, T. A., & Patel, V. (2006). Customizing clinical narratives for the electronic medical record interface using cognitive methods. Int J Med Inform, 75(5), 346-368. doi: 10.1016/j.ijmedinf.2005.07.027.

Cohen, T., Blatter, B., & Patel, V. (2008). Simulating expert clinical comprehension: adapting latent semantic analysis to accurately extract clinical concepts from psychiatric narrative. J Biomed Inform, 41(6), 1070-1087. doi: 10.1016/j.jbi.2008.03.008.

Staszewski, J. (2004). Models of expertise as blueprints cognitive engineering: Applications to landmine detection. Paper presented at the Human Factors and Ergonomics Society 48th Annual Meeting, New Orleans.

Afterword

We began describing how an EHR could go wrong if design and implementation does not consider usability, workflow and cognitive support. Now let's consider a brighter scenario. Imagine a patient arrives at his physician's office for a visit.

A camera in the waiting room…

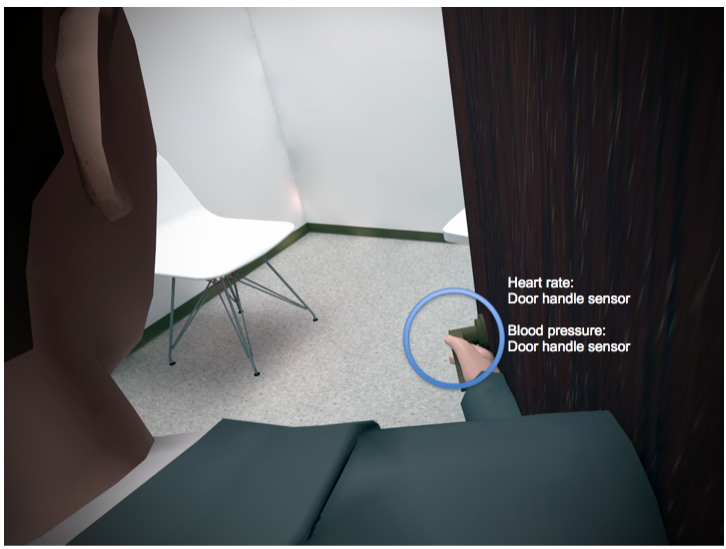
… uses facial recognition to identify the patient

… and instantly checks him in. His records are automatically retrieved for confirmation. Updated information can be directly entered by the patient using a tablet at the desk.

As the patient walks into an exam room, sensors embedded in the floor measure his weight. Cameras mounted on the wall measure the

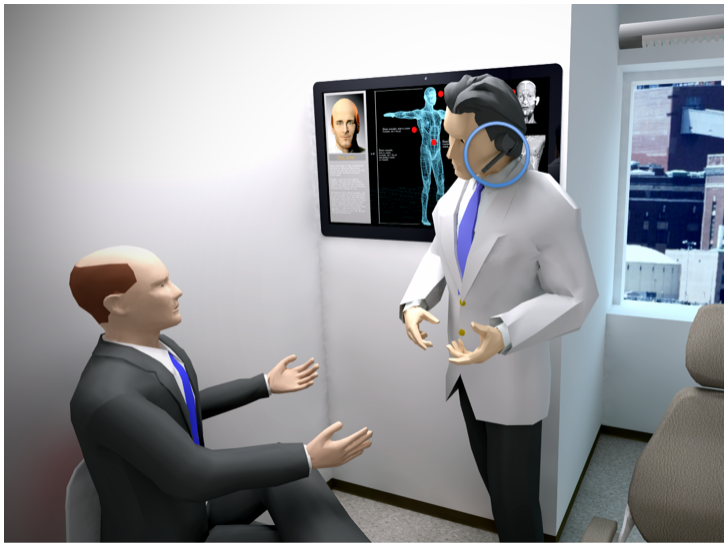
patient's height and body temperature. All of this information is automatically transmitted to his EHR.

Upon entering the exam room, sensors in the door handle take the patient's heart rate and blood pressure, and then update his medical record.



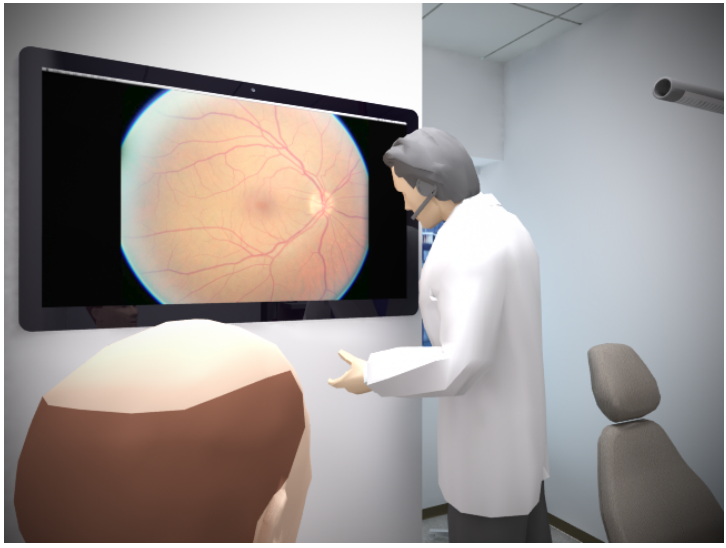
The physician enters the exam room where the patient's EHR has automatically been called up on a large touch screen. The physician uses

speech recognition to capture his exam notes and orders directly into the EHR system.

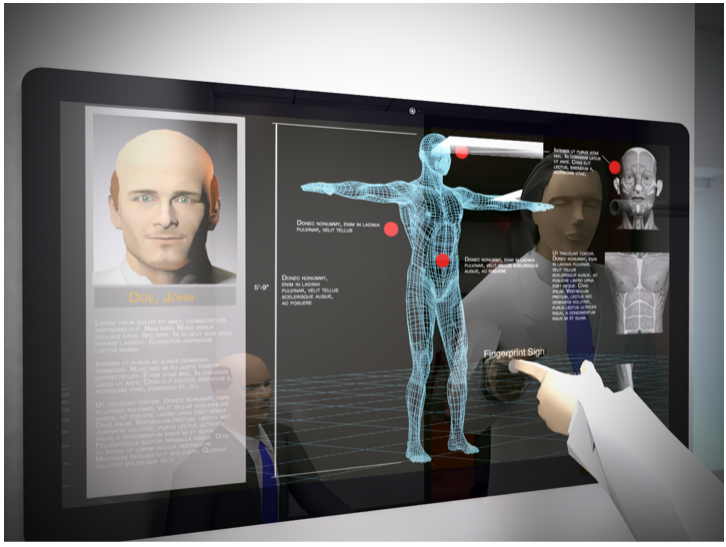


The physician discusses treatment options with the patient and reconciles his medications. The system automatically checks for drug interactions. The physician then e-prescribes an order to the patient's pharmacy.

Patient education information is automatically displayed. After a discussion between the physician and patient, the information is transmitted to the patient through encrypted email.



The encounter ends with the physician reviewing the patient's chart. The physician uses a fingerprint sensor to sign off.



The EHR system is nearly invisible to the patient in this scenario. The physician takes care of the patient with his full attention to personal details. Efficiency of the care is increased and patient safety is guarded. The patient is also satisfied with the encounter experience. As a result of the increased efficiency, safety, and satisfaction, the cost of care drops—exactly what electronic health record systems were envisioned to do in the first place.

We are not there yet, but we are absolutely on the way…

Appendices

SHARPC Publications

1. Butler K., Haselkorn M.P., Ali B., Schroder K. (2011). Introducing the MATH Method and Tool suite for Evidence-Based HIT. Paper presented at the AMA-IEEE, Boston MA.
2. Butler K. A., Haselkorn M.P., Braxton M., Lyon L., Nichol W. P., Berry A. B. L., Chung C. (2013). Evidence-based Health IT for Patient Provider Communication. Paper presented at the ACM CHI 2013, Paris, France.
3. Chao T. Visual techniques for medication reconciliation: spatial metaphor, animated explanation, and flexible decision-making (Undergrad Honor Project report - Dec 2011).
4. D'Amore J. D. (2011). The Promise of the Continuity of Care Document. (MS Master's), The University of Texas Health Science Center at Houston, Houston TX.
5. D'Amore J. D., Sittig D. F., Ness R. B. (2012). How the continuity of care document can advance medical research and public health. Am J Public Health, 102(5), e1-4. doi: 10.2105/AJPH.2011.300640
6. D'Amore J. D., Sittig D. F., Wright A., Iyengar M. S., Ness R. B. (2011). The promise of the CCD: challenges and opportunity for quality improvement and population health. AMIA Annu Symp Proc, 2011, 285-294.
7. Feblowitz J. C., Wright A., Singh H., Samal L, Sittig D. F. (2011). Summarization of clinical information: a conceptual model. J Biomed Inform, 44(4), 688-699. doi: 10.1016/j.jbi.2011.03.008
8. Harrington C., Wood R., Breuer J., Pinzon O., Howell R., Pednekar M, . . . Zhang J. (2011). Using a unified usability framework to dramatically improve the usability of an EMR Module. AMIA Annu Symp Proc, 2011, 549-558.
9. Joffe E., Byrne M.J., Reeder P., Herskovic J.R., Johnson C. W., McCoy A.B., Sittig D.F., Bernstam E. V. A benchmark comparison of deterministic and probabilistic methods for defining manual review

datasets in duplicate records reconciliation.. J Am Med Inform Assoc. 2014 Jan-Feb;21(1):97-104. doi: 10.1136/amiajnl-2013-001744.

10. Johnson T. R., Markowitz E., Bernstam E. V., Herskovic J. R., Thimbleby H. (2013). SYFSA: a framework for systematic yet flexible systems analysis. J Biomed Inform, 46(4), 665-675. doi: 10.1016/j.jbi.2013.05.003.

11. Kannampallil T. G., Franklin A., Mishra R., Almoosa K. F., Cohen T., Patel V. L. (2013). Understanding the nature of information seeking behavior in critical care: implications for the design of health information technology. Artif Intell Med, 57(1), 21-29. doi: 10.1016/j.artmed.2012.10.002.

12. Klann J. G., McCoy A. B., Wright A., Wattanasin N., Sittig D.F., Murphy SN. (2013). Health care transformation through collaboration on open-source informatics projects: integrating a medical applications platform, research data repository, and patient summarization. Interact J Med Res, 2(1). doi: 10.2196/ijmr.2454.

13. Laxmisan A., McCoy A. B., Wright A., Sittig D. F. (2012). Clinical Summarization Capabilities of Commercially-available and Internally-developed Electronic Health Records. Appl Clin Inform, 3(1), 80-93. doi: 10.4338/ACI-2011-11-RA-0066.

14. Lowry, S. Z., Quinn, M. T., Ramaiah, M., Brick, D., Patterson, E. S., Zhang, J., Abbott, P., & Gibbons, M. C. (2012). A Human Factors Guide to Enhance EHR Usability of Critical User Interactions when Supporting Pediatric Patient Care. (NISTIR 7865). National Institute of Standards and Technology.

15. Markowitz E, Bernstam E. V., Herskovic J., Zhang J., Shneiderman B., Plaisant C., Johnson T. R. (2011). Medication Reconciliation: Work Domain Ontology, prototype development, and a predictive model. AMIA Annu Symp Proc, 2011, 878-887.

16. McCoy A. B., Melton G. B., Wright A., Sittig D. F. (2013). Clinical Decision Support for Colon and Rectal Surgery: An Overview. Clin Colon Rectal Surg, 26(1), 23-30. doi: 10.1055/s-0033-1333644.

17. McCoy A. B., Sittig D. F., & Wright A. (2013). Comparison of clinical knowledge bases for summarization of electronic health records. Stud Health Technol Inform, 192, 1217.

18. McCoy A. B., Wright A., Eysenbach G., Malin B. A., Patterson E. S., Xu H., Sittig D. F. (2013). State of the art in clinical informatics: evidence and examples. Yearb Med Inform, 8(1), 13-19.

19. McCoy A. B., Wright A., Kahn M. G., Shapiro J. S., Bernstam E. V., Sittig D. F. (2013). Matching identifiers in electronic health records: implications for duplicate records and patient safety. BMJ Qual Saf, 22(3), 219-224. doi: 10.1136/bmjqs-2012-001419.

20. McCoy A. B., Wright A., Laxmisan A., Ottosen M. J., McCoy J. A., Butten D, Sittig D. F. (2012). Development and evaluation of a crowdsourcing methodology for knowledge base construction: identifying relationships between clinical problems and medications. J Am Med Inform Assoc, 19(5), 713-718. doi: 10.1136/amiajnl-2012-000852.

21. McCoy A. B., Wright A., Laxmisan A., Singh H., Sittig D. F. (2011). A prototype knowledge base and SMART app to facilitate organization of patient medications by clinical problems. AMIA Annu Symp Proc, 2011, 888-894.

22. McCoy A. B., Wright A., Rogith D., Fathiamini S., Ottenbacher A. J., Sittig D. F. (2013). Development of a clinician reputation metric to identify appropriate problem-medication pairs in a crowdsourced knowledge base. J Biomed Inform. doi: 10.1016/j.jbi.2013.11.010.

23. Murphy D. R., Reis B., Kadiyala H., Hirani K., Sittig D. F., Khan M. M., & Singh H. (2012). Electronic health record-based messages to primary care providers: valuable information or just noise? Arch Intern Med, 172(3), 283-285. doi: 10.1001/archinternmed.2011.740.

24. Murphy D. R., Reis B., Sittig D. F., & Singh H. (2012). Notifications received by primary care practitioners in electronic health records: a taxonomy and time analysis. Am J Med, 125(2), 209 e201-207. doi: 10.1016/j.amjmed.2011.07.029.

25. Osheroff J. A., Teich J. M., Levick D., Saldana L., Velasco F. T., Sittig D. F., . . . Jenders R. A. (2012). Improving Outcomes with Clinical

Decision Support: An Implementer's Guide (2nd ed.): Healthcare Information and Management Systems Society.

26. Plaisant C., Chao T., Wu J., Hettinger A. Z., Herskovic J. R., Johnson T. R., Bernstam E. V., Markowitz E., Powsner S., Shneiderman B. (2013). TwinList: novel user interface designs for medication reconciliation. AMIA Annu Symp Proc, 2013, 1150-1159.

27. Radecki R. P., Sittig D. F. (2011). Application of electronic health records to the Joint Commission's 2011 National Patient Safety Goals. JAMA, 306(1), 92-93. doi: 10.1001/jama.2011.937.

28. Rind A., Wang T. D., Aigner W., Miksch S., Wongsuphasawat K., Plaisant C., Shneiderman B. (2013). Interactive Information Visualization to Explore and Query Electronic Health Records. Foundations and Trends in Human–Computer Interaction, 5(3). doi: 10.1561/1100000039.

29. Saitwal H., Qing D., Jones S., Bernstam E. V., Chute C. G., Johnson T. R. (2012). Cross-terminology mapping challenges: a demonstration using medication terminological systems. J Biomed Inform, 45(4), 613-625. doi: 10.1016/j.jbi.2012.06.005.

30. Schumacher, R. M., Patterson, E., S., North, R., Zhang, J., Lowry, S. Z., Quinn, M. T., Ramaiah, M, & National Institute of Standards and Technology. (2012). Technical Evaluation Testing and Validation of the Usability of Electronic Health Records. (NISTIR 7804). National Institute of Standards and Technology.

31. Shneiderman B., Plaisant C., Hesse B. W. (2013). Improving Healthcare with interactive visualization. Computer, 46(5). doi: 10.1109/MC.2013.38.

32. Silva P. A., Bernstam E.V ., Markowitz E., Johnson T. R., Zhang J., Herskovic J. R. (2011). Automated medication reconciliation and complexity of care transitions. AMIA Annu Symp Proc, 2011, 1252-1260.

33. Singh H., Classen D. C., Sittig D. F. (2011). Creating an oversight infrastructure for electronic health record-related patient safety hazards. J Patient Saf, 7(4), 169-174. doi: 10.1097/PTS.0b013e31823d8df0.

34. Sittg D., Ash J. S. (2011). On the Importance of Using a Multidimensional Sociotechnical Model to Study Health Information Technology. Ann Fram Med, 9(5).

35. Sittig D. F., Singh H. (2011). Defining health information technology-related errors: new developments since to err is human. Arch Intern Med, 171(14), 1281-1284. doi: 10.1001/archinternmed.2011.327.

36. Sittig D. F., Singh H. (2011). Legal, ethical, and financial dilemmas in electronic health record adoption and use. Pediatrics, 127(4), e1042-1047. doi: 10.1542/peds.2010-2184.

37. Sittig D. F., Singh H. (2012). Electronic Health Records and National Patient Safety Goals. New England Journal of Medicine, 367(19). doi: 10.1056/NEJMsb1205420.

38. Sittig D. F., Singh H. (2012). Rights and responsibilities of users of electronic health records. CMAJ, 184(13), 1479-1483. doi: 10.1503/cmaj.111599.

39. Sittig D. F., Singh H., Longhurst C. A. (2013). Rights and responsibilities of electronic health records (EHR) users caring for children. Arch Argent Pediatr, 111(6), 468-471. doi: 10.1590/S0325-00752013000600003.

40. Sittig D. F., Wright A., Meltzer S., Simonaitis L., Evans R. S., Nichol W. P., Ash J. S., Middleton B. (2011). Comparison of clinical knowledge management capabilities of commercially-available and leading internally-developed electronic health records. BMC Med Inform Decis Mak, 11, 13. doi: 10.1186/1472-6947-11-13.

41. Sopan A., Plaisant C, Powsner S., Shneiderman B. User Interface Techniques to Reduce Wrong Patient Selection Error. Proc. AMIA 2014 Annual Symposium.

42. Staggers N., Xiao Y., Chapman L. (2013). Debunking health IT usability myths. Appl Clin Inform, 4(2), 241-250. doi: 10.4338/ACI-2013-03-IE-0016.

43. Wright A., Feblowitz J., McCoy A. B., Sittig D. F. (2011). Comparative analysis of the VA/Kaiser and NLM CORE problem subsets: an empirical study based on problem frequency. AMIA Annu Symp Proc, 2011, 1532-1540.

44. Wright A. Feblowitz J., Samal L., McCoy A. B., Sittig D. F. (2014). The Medicare Electronic Health Record Incentive Program: provider performance on core and menu measures. Health Serv Res, 49(1 Pt 2), 325-346. doi: 10.1111/1475-6773.12134.

45. Wright A., Henkin S., Feblowitz J., McCoy A. B., Bates D. W., Sittig D. F. (2013). Early results of the meaningful use program for electronic health records. N Engl J Med, 368(8), 779-780. doi: 10.1056/NEJMc1213481.

46. Wright A., McCoy A., Henkin S., Flaherty M., & Sittig D. (2013). Validation of an association rule mining-based method to infer associations between medications and problems. Appl Clin Inform, 4(1), 100-109. doi: 10.4338/ACI-2012-12-RA-0051.

47. Wright A., Pang J., Feblowitz J. C., Maloney F. L., Wilcox A. R., Ramelson H. Z,. Schneider L. I., Bates D. W. (2011). A method and knowledge base for automated inference of patient problems from structured data in an electronic medical record. J Am Med Inform Assoc, 18(6), 859-867. doi: 10.1136/amiajnl-2011-000121.

48. Xu H., Stenner S. P., Doan S., Johnson K. P., Waitman L. R., & Denny J. C. (2010). MedEx: A medication information extraction system for clinical narratives. J Am Med Inform Assoc, 17(1), 19. doi: 10.1197/jamia.M3378.

49. Zhang J., Walji M. F. (2011). TURF: toward a unified framework of EHR usability. J Biomed Inform, 44(6), 1056-1067. doi: 10.1016/j.jbi.2011.08.005.

For updates to this list, please visit:

https://sbmi.uth.edu/nccd/research/publications.htm

SHARPC Products

SHARPC research teams have developed several resources to help assess and improve EHR Usability. These include:

Clinical Summarization App and Knowledge Bases

Increased amounts of data contained in electronic health records (EHRs) have led to inefficiencies for clinicians trying to locate relevant patient information. Automated summarization tools that create condition-specific data displays rather than current displays organized by data type and time have the potential to greatly improve clinician efficiency. Methodologies for modeling and summarizing complex chronically-ill patients' electronic health records were designed. Knowledge bases include:

- MAPLE
- Problem-Medication Linkage
- MedEx

Health eDecisions XML Editor

This Editor can create computer-readable healthcare knowledge artifacts (in xml format) based on the formal HeD knowledge model. The goal is to enable a subject matter expert with little or no programming experience to create a knowledge artifact such as a decision rule without technical support.

Inspirational Prototypes

Demonstration prototypes were produced for several EHR related situations that can be problematic in visual display. These include:

- Medication Reconciliation
- Result Management

- Reducing Wrong Patient Selection
- Discussing Treatment Options with Patients
- Discharge Summary Patterns
- Clinical Summarization and Knowledge Bases

Increased amounts of data contained in electronic health records (EHRs) have led to inefficiencies for clinicians trying to locate relevant patient information. Automated summarization tools that create condition-specific data displays rather than current displays organized by data type and time have the potential to greatly improve clinician efficiency.

Medication Reconciliation Algorithm

The reconciliation Algorithm which computes similarity between drugs is freely available at https://github.com/jherskovic/MedRec.

Rapid Usability Assessment Protocol

To improve our understanding of the potential for EHR induced errors, seven commercial EHR products have been evaluated following the Rapid Usability Assessment (RUA) created at SHARPC. Our protocol, combining heuristic evaluation and keystroke level models of expert use, was used to identify usability problems that occurred during the completion of 12 meaningful use-related test procedures (e.g. computerized provider order entry for a medication order, updating of allergy lists). The identified usability violations were then scored for their potential for harm using a severity score and grouped according to test procedure. Confidential reports were provided to the participating vendors to help improve the usability of their EHRs.

Safety-enhanced Design: User testing scenarios

The 2014 EHR safety-enhanced design requirements for testing and reporting of system usability also exposed need for validated, contextually-rich, scenarios for testing. As part of SHARPC efforts we

developed assessment materials and detailed our methods for generating these materials as part of a call for open discussion regarding public exchange of protocols. Our hope is that scenarios used in testing capture the needs of clinical providers, robustly measure the usability of systems and provide a means of identifying safety risks in existing systems. These materials were written to support vendors engaged in usability studies and certification procedures. Included are participant instructions, moderator guides, standard usability scales and other materials included in user testing.

SIRSI Cognitive Support System

The SIRSI Cognitive Support System prototype facilitates the organization of clinical data in accordance with decision models derived from domain experts. It includes a back-end interface to clinical data sources, a knowledge module that can be interchanged depending on the decision model of interest, and iPAD and browser based interfaces through which clinical data can be viewed organized in a manner conducive to decision making related to the Systemic Inflammatory Response Syndrome (SIRS) in an ICU context.

SYFSA

Systematic Yet Flexible Systems Analysis (SYFSA) is a framework developed by SHARPC for designing and analyzing SYF systems. It is based on analyzing a task using three related problem spaces: the idealized space, the natural space, and the system space.

TURF Framework for EHR Usability

SHARPC developed a framework of EHR usability, called TURF, to unify the concepts and methods around EHR usability. TURF is (1) a theory for describing, explaining, and predicting usability differences; (2) a method for defining, evaluating, and measuring usability objectively; (3) a process for designing built-in good usability; and (4) a guideline for developing EHR usability standards. TURF defines usability as how useful, usable, and satisfying a system is for the intended users to

accomplish goals in the work domain by performing certain sequences of tasks. One of the contributions of the TURF framework is to show that usability can not only be defined scientifically under a coherent, unified framework, it can also be measured objectively and systematically. Part of the TURF framework has been implemented as a software tool that is described in the next section.

Turf Software Tool to assess usability of EHRs

Many of the techniques for usability assessments have included pen and paper along with stand-alone recordings. To aid in usability evaluation as well as testing and design of electronic health record systems, we have developed the *Turf* software, which is based on the TURF framework of EHR usability. *Turf* is a software tool designed to evaluate, document, and assist in improving EHR usability.

Usability Design Guidelines

- Safety-enhanced Design Briefs - To determine whether these products were helpful to EHR vendors and to identify any additional vendor needs, SHARPC engaged a liaison to interview vendors. Their feedback showed that vendors without human factors design experts desired short, actionable advice to improve their EHRs in preparation for the safety-enhanced design certification. It also revealed an inconsistency in how vendors viewed usability or user-centered design, with some thinking that it simply meant responding to user feedback. Findings suggested that, in addition to short design tips, vendors could benefit from pointers to more in-depth material on EHR user interface design. To address these needs, we developed a set of one-page safety-enhanced design briefs (SEDB) along with a corresponding website for dissemination of the briefs and references to supplementary information.
- General Design Principles and Guidelines - Employing a systematic search methodology, SHARPC identified and compiled approximately 300 design principles from four major guideline documents that are significant to safety-enhanced and efficient design of EHR. These

principles fall into 14 heuristic categories in terms of consistency, visibility, match, minimalism, memory, feedback, flexibility, messages, errors, closure, undo, language, control and help, illustrated with examples of DOs and/or DON'Ts. With the instruction of the categorized principles and examples, the guidelines should facilitate the design and development of EHR systems toward the objectives of meaningful use (MU).

- Inspired EHRs: Designing for Clinicians eBook - SHARPC co-funded, with the California Healthcare Foundation, a clinically inspired, generously illustrated, interactive EHR (electronic health record) usability design guide targeted at the EHR vendor community, incorporating that group into its design and evaluation. Iterative feedback came from an expert advisory panel and the target audience.

For updates to this list, please visit:

https://sbmi.uth.edu/nccd/research/products/

Abbreviations

AHLTA - Armed Forces Health Longitudinal Technology Application EHR system used by the US Department of Defense.

AMI - Acute myocardial infarction

ANOVA - Analysis of variance statistical method

API - Application Programming Interface

ARRA - American Recovery and Reinvestment Act

ATM - Automatic Teller Machine

BPMN - Business Process Modeling nNotation

CABG - Coronary Artery Bypass Graft

CDS - Clinical Decision Support systems

CEBES - Cognitive engineering based on expert skill

CEM - Clinical element model

CMS - Centers for Medicare and Medicaid

CIF - Common Industry Format

CIMI - Clinical Information Modeling Initiative

CPOE - Computerized Provider Order Entry

CSS - Cognitive Support Systems

CWA - Cognitive Work Analysis

DC - Dublin Core

EDR - Electronic Dental Record

EHR - Electronic Health Record

EHRA - Health Information Management Systems Society Electronic Health Records Association

ETTO - Efficiency-Thoroughness Tradeoff

FDA - Food and Drug Administration

FHIR - Fast Healthcare Interoperability Resources

FSM - Finite State Machine

HeD - Health eDecisions

HIMSS - Health Information and Management System Society

HIE - Health Information Exchange

HIT - Health Information Technology

HITECH - The Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009.

HPI - History of Present Illness

HL7 - Health Level 7

ICD-9 - World Health Organization international classification of diseases, ninth revision.

IDE - Integrated Development Environment

IOM - Institute of Medicine

IPT - Integrated Product Team

ISO - International Standards Organization

IT - Information Technology

IVD - Ischemic Vascular Disease

KA - Knowledge Artifacts

KE - Knowledge Engineers

KLM - Keystroke Level Modeling

LMR - Longitudinal Medical Record

MATH - Modeling & Analysis Tool suite for Healthcare

MICU - Medical Intensive Care Unit

MIMIC-II - Multiparameter Intelligent Monitoring in Intensive Care II database.

MOOC - Massive Open Online Course

MSTART - Multi-Step Task Analyzing, Reporting and Tracking

MVC - Model View Controller

OMG - Object Management Group

ONC - Office of National Coordinator for Health Information Technology

OWL2-DL - Web Ontology Language version 2

NCCD - National Center for Cognitive Informatics and Decision Making in Healthcare

NDF-RT - National Drug File Reference Terminology

NIST - National Institute of Standards and Technology

NLM - National Library of Medicine

NLP - Natural Language Processing

PCP - Primary Care Provider

PMI - Present Medical Illness

PRR - Production Rule Representation OMG standard

RAP - Rapid Assessment Process

RDF - Resource Description Framework

RUA - Rapid Usability Assessment

SED - Safety-enhanced Design

SEDB - Safety-Enhanced Design Briefs

SHARP - Office of National Coordinator for Health Information Technology Strategic Health IT Advanced Research Program

SHARPC - Office of National Coordinator for Health Information Technology's Strategic Health IT Advanced Research Program focused on patient-centered cognitive support.

SIRS - Systemic Inflammatory Response Syndrome

SMART - Harvard University SHARP project: substitutable medical apps reusable technologies platform

SME - Subject Matter Expert

SNOMED-CT - Systematized Nomenclature of Medicine Clinical Terms

SSF - Setting-Specific Factors

SUS - System Usability Scale

SYF - Systematic Yet Flexible

SYFSA - Systematic Yet Flexible Systems Analysis

TURF - Task, User, Representation, and Function: a unified framework of EHR usability.

UCD - User-Centered Design

UI - User Interface

UML - Unified Modeling Language

UX - User Experience

W3C - World Wide Web Consortium

WDO - Work Domain Ontology

WPF - Windows Presentation Foundation

VSAC - Value Set Authority Center

Bibliography

1. Pedulli, L. KLAS survey: Usability ranked as most important feature for EMRs. 2014 October 14, 2014]; Available from: http://www.clinical-innovation.com/topics/ehr-emr/klas-survey-usability-ranked-most-important-feature-emrs.

2. Strategic Health IT Advanced Research Projects (SHARP). 2014 [cited 2014 June 30, 2014]; Available from: http://www.healthit.gov/policy-researchers-implementers/strategic-health-it-advanced-research-projects-sharp.

3. Crossing the Quality Chasm: A New Health System for the 21st Century. 2001: The National Academies Press.

4. Corrigan, J.M., M.S. Donaldson, and L.T. Kohn, eds. Crossing the quality chasm: A new health system for the 21st century. 2001, National Academies Press: Washington, D. C.

5. National Research Council Committee on Engaging the Computer Science Research Community in Health Care, I., The National Academies Collection: Reports funded by National Institutes of Health, in Computational Technology for Effective Health Care: Immediate Steps and Strategic Directions, W.W. Stead and H.S. Lin, Editors. 2009, National Academies Press (US) and the National Academy of Sciences: Washington (DC).

6. Health information technology: standards, implementation specifications, and certification criteria for electronic health record technology, 2014 edition; revisions to the permanent certification program for health information technology. Final rule. Fed Regist, 2012. 77(171): p. 54163-292.

7. Zhang, J. and M.F. Walji, TURF: toward a unified framework of EHR usability. J Biomed Inform, 2011. 44(6): p. 1056-67.

8. Hillestad, R., et al., Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. Health Aff (Millwood), 2005. 24(5): p. 1103-17.

9. RAND Corporation, Health Information Technology: Can HIT Lower Costs and Improve Quality? 2005, RAND Corporation: Santa Monica, CA.

10. Chaudhry, B., et al., Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. Ann Intern Med, 2006. 144(10): p. 742-52.

11. Blumenthal, D. and J.P. Glaser, Information technology comes to medicine. N Engl J Med, 2007. 356(24): p. 2527-34.

12. Davenport, K., Navigating American health care: how information technology can foster health care improvement. 2007, Center for American Progress: Washington, D. C.

13. Hagland, M. Performance Improvement Special Report: Part I. The Landscape. Healthcare Informatics, 2007.

14. The Leapfrog Group, Computer Physician Order Entry: Fact Sheet. 2007.

15. Southon, G., C. Sauer, and K. Dampney, Lessons from a failed information systems initiative: issues for complex organisations. Int J Med Inform, 1999. 55(1): p. 33-46.

16. Han, Y.Y., et al., Unexpected increased mortality after implementation of a commercially sold computerized physician order entry system. Pediatrics, 2005. 116(6): p. 1506-12.

17. Koppel, R., et al., Role of computerized physician order entry systems in facilitating medication errors. Jama, 2005. 293(10): p. 1197-203.

18. Zhang, J., Human-centered computing in health information systems part 1: analysis and design. J Biomed Inform, 2005. 38(1): p. 1-3.

19. Kushniruk, A.W. and V.L. Patel, Cognitive and usability engineering methods for the evaluation of clinical information systems. J Biomed Inform, 2004. 37(1): p. 56-76.

20. Johnson, C.M., T.R. Johnson, and J. Zhang, A user-centered framework for redesigning health care interfaces. J Biomed Inform, 2005. 38(1): p. 75-87.

21. Zhang, J., Human-centered computing in health information systems part 2: evaluation. J Biomed Inform, 2005. 38(3): p. 173-5.

22. Patel, V.L. and J. Zhang, Cognition and patient safety, in Handbook of applied cognition, F.T. Durso, et al., Editors. 2007, Wiley: New York. p. 307-331.

23. Armijo, D., C. McDonnell, and K. Werner, Electronic health record usability: Interface design considerations. 2009, Agency for Healthcare Research and Quality: Rockville, Maryland.

24. Aarts, J. and V. Peel, Using a descriptive model of change when implementing large scale clinical information systems to identify priorities for further research. Int J Med Inform, 1999. 56(1-3): p. 43-50.

25. Goddard, B.L., Termination of a contract to implement an enterprise electronic medical record system. J Am Med Inform Assoc, 2000. 7(6): p. 564-8.

26. Berg, M., Implementing information systems in health care organizations: myths and challenges. Int J Med Inform, 2001. 64(2-3): p. 143-56.

27. Butler, K.A., et al., Work-centered design: a case study of a mixed-initiative scheduler, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2007, ACM: San Jose, California, USA. p. 747-756.

28. Zhang, J. and K.A. Butler UFuRT: a work-centered framework for the design and evaluation of information systems. in HCI International 2007. 2007. Beiging, China.

29. Butler, K.A. and J. Zhang, Design models for interactive problem-solving: context & ontology, representation & routines, in CHI '09 Extended Abstracts on Human Factors in Computing Systems. 2009, ACM: Boston, MA, USA. p. 4315-4320.

30. Butler, K.A., et al., Ontology models for interaction design: case study of online support, in CHI '10 Extended Abstracts on Human Factors in Computing Systems. 2010, ACM: Atlanta, Georgia, USA. p. 4525-4540.

31. Zhu, M., Formalizing a conceptual framework of work domain knowledge, in School of Biomedical Informatics. 2010, The University of Texas Health Science Center at Houston.

32. Landauer, T., The trouble with computers: usefulness, usability, and productivity. 1995, Cambridge, MA: MIT Press.

33. Goransson, B., et al., The interface is often not the problem, in Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface. 1987, ACM: Toronto, Ontario, Canada. p. 133-136.

34. Kieras, D.E., Task Analysis and the Design of Functionality, in The Computer Science and Engineering Handbook, A.B. Tucker, Editor. 1997, CRC Press. p. 1401-1423.

35. Larkin, J.H. and H.A. Simon, Why a diagram is (sometimes) worth 10,000 words. Cognitive Sci, 1987. 11: p. 65-99.

36. Kotovsky, K. and H.A. Simon, What makes some problems really hard: Explorations in the problem space of difficulty. Cognitive Psychology, 1990. 22(3): p. 143-183.

37. Zhang, J. and D.A. Norman, Representations in distributed cognitive tasks. Cognitive Science, 1994. 18(1): p. 87-122.

38. Zhang, J., A representational analysis of relational information displays. Int. J. Hum.-Comput. Stud., 1996. 45(1): p. 59-74.

39. Zhang, J., The nature of external representations in problem solving. Cognitive Science, 1997. 21(2): p. 179-217.

40. Zhang, J., et al., Designing human-centered distributed information systems. IEEE Intelligent Systems, 2002. 17(5): p. 42-47.

41. Zhang, J., et al., Designing human-centered distributed information systems. IEEE Intelligent Systems, 2002. 17(5): p. 42-47.

42. Nahm, M. and J. Zhang, Operationalization of the UFuRT methodology for usability analysis in the clinical research data management domain. J Biomed Inform, 2009. 42(2): p. 327-33.

43. Zhang, J. and D.A. Norman, Representations in distributed cognitive tasks. Cognitive Science, 1984. 18(1): p. 87-122.

44. Kuniavsky, M., Observing the user experience: a practitioner's guide to user research. 2003, San Francisco: Morgan Kaufman.

45. Chen, J.W., Developing a process for reducing functional discrepancies, in School of Health Information Sciences. 2008, University of Texas Health Science Center at Houston: Houston, TX.

46. Zhang, Z., et al., Functional analysis of interfaces in U.S. military electronic health record system using UFuRT framework. AMIA Annu Symp Proc, 2009. 2009: p. 730-4.

47. Hutchins, E., Direct manipulation interfaces, in User centered system design: new perspectives in human–computer interaction, D.A.

Norman and S. Draper, Editors. 1986, Lawrence Erlbaum: Hillsdale, NJ.

48. Gibson, J.J., The ecological approach to visual perception. 1st ed. 1979, Boston: Houghton Mifflin.

49. Gibson, J.J., The theory of affordances, in Perceiving, acting, and knowing, R.E. Shaw and J. Bransford, Editors. 1997, Lawrence Erlbaum Associates: Hillsdale, NK.

50. Gaver, W.W., Technology affordances, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1991, ACM: New Orleans, Louisiana, USA. p. 79-84.

51. Zhang, J. and V.L. Patel, Distributed cognition, representation, and affordance. Cognition Pragmatics, 2006. 14(2): p. 333-341.

52. Nielsen, J. and R. Molich, Heuristic evaluation of user interfaces, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1990, ACM: Seattle, Washington, USA. p. 249-256.

53. Nielsen, J., Finding usability problems through heuristic evaluation, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1992, ACM: Monterey, California, USA. p. 373-380.

54. Nielsen, J., Usability Engineering. 1993: Morgan Kaufmann Publishers Inc. 358.

55. Usability inspection methods. ed. N. Jakob and L.M. Robert. 1994, John Wiley & Sons, Inc. 413.

56. Shneiderman, B., Designing the user interface: strategies for effective human-computer-interaction. 3rd ed. 1998, Reading, MA: Addison Wesley Longman.

57. Zhang, J., et al., Using usability heuristics to evaluate patient safety of medical devices. J Biomed Inform, 2003. 36(1-2): p. 23-30.

58. Graham, M.J., et al., Heuristic evaluation of infusion pumps: implications for patient safety in Intensive Care Units. Int J Med Inform, 2004. 73(11-12): p. 771-9.

59. Zhang, J., et al., Evaluating and predicting patient safety for medical devices with integral information technology, in Advances in Patient Safety: From Research to Implementation, K. Henriksen, et al., Editors.

2005, Agency for Healthcare Research and Quality: Rockville, MD. p. 323-336.

60. Tang, Z., et al., Applying heuristic evaluation to improve the usability of a telemedicine system. Telemed J E Health, 2006. 12(1): p. 24-34.

61. Diaper, D. and N.A. Stanton, eds. The handbook of task analysis of human–computer interaction. 2004, Lawrence Erlbaum Associates, Inc.: Mahwah, NJ.

62. Crandall, B., G. Klein, and R.R. Hoffman, Working minds: a practitioner's guide to cognitive task analysis. 2006, Cambridge, MA: MIT Press.

63. Card, S.K., T.P. Moran, and A. Newell, The psychology of human-computer interaction. 1983, London: Lawrence Erbaum Associates.

64. John, B.E. and D.E. Kieras, The GOMS family of user interface analysis techniques: comparison and contrast. ACM Trans. Comput.-Hum. Interact., 1996. 3(4): p. 320-351.

65. Wright, P.C., R.E. Fields, and M.D. Harrison, Analyzing human-computer interaction as distributed cognition: the resources model. Hum.-Comput. Interact., 2000. 15(1): p. 1-41.

66. Saitwal, H., et al., Assessing performance of an Electronic Health Record (EHR) using Cognitive Task Analysis. Int J Med Inform, 2010. 79(7): p. 501-6.

67. Kieras, D. Using the keystroke-level model to estimate execution times. 2001.

68. John, B.E. CogTool. 2009 [cited 2011 May 16]; Available from: http://cogtool.hcii.cs.cmu.edu/.

69. ACT-R Research Group. Home: ACT-R. [cited 2011 May 16]; Available from: http://act-r.psy.cmu.edu.

70. Anderson, J.R., Rules of the mind. 1993, Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

71. Luo, L. and B.E. John, Predicting task execution time on handheld devices using the keystroke-level model, in CHI '05 Extended Abstracts on Human Factors in Computing Systems. 2005, ACM: Portland, OR, USA. p. 1605-1608.

72. Harrington, C., et al., Using a unified usability framework to dramatically improve the usability of an EMR Module. AMIA Annu Symp Proc, 2011. 2011: p. 549-58.

73. Brixey, J.J., et al., Towards a hybrid method to categorize interruptions and activities in healthcare. Int J Med Inform, 2007. 76(11-12): p. 812-20.

74. Brixey, J.J., et al., Interruptions in a level one trauma center: a case study. Int J Med Inform, 2008. 77(4): p. 235-41.

75. Franklin, A., et al., Opportunistic decision making and complexity in emergency care. J Biomed Inform, 2011. 44(3): p. 469-76.

76. Westbrook, J.I., et al., Association of interruptions with an increased risk and severity of medication administration errors. Arch Intern Med, 2010. 170(8): p. 683-90.

77. Butler, K.A., et al. MATH: Method and toolsuite for integrating HIT with clinical workflow. n. d. [cited 2014 June 30]; Available from: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CB0QFjAA&url=https%3A%2F%2Fcatalyst.uw.edu%2Fworkspace%2Ffile%2Fdownload%2F8849c201a4042ae800dbc6501b49d8ebc5cebe3995be89c7766f6cb4a8303aec&ei=XtWxU-GZK8WiqAa63oLwDw&usg=AFQjCNEhI195vGf3ZU924eeB6DEyYsgNRw&sig2=LTk818SpqJ70n3fZAH-x2A.

78. Butler , K.A., C. Esposito, and R. Hebron, Connecting the design of software to the design of work. Communications of the ACM, 1999. 42(1): p. 38-46.

79. Chen, Y., Documenting transitional information in EMR, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2010, ACM: Atlanta, Georgia, USA. p. 1787-1796.

80. Koppel, R., et al., Workarounds to barcode medication administration systems: their occurrences, causes, and threats to patient safety. J Am Med Inform Assoc, 2008. 15(4): p. 408-23.

81. Hollnagel, E., The ETTO principle: efficiency-thoroughness trade-off: why things that go right sometimes go wrong. 2009, Ashgate Publishing: Burlington, VT.

82. Perer, A. and B. Shneiderman. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. in Proceedings of

the 13th international conference on intelligent user interfaces. 2008. Canary Islands, Spain.

83. Thimbleby, H., Permissive user interfaces. International Journal of Human-Computer Studies, 2001. 54(3): p. 333-350.

84. Norman, D.A., The design of everyday things. 2002, Basic Books: USA.

85. Swaney, R.E. and I.E. Grossmann, An index for operational flexibility in chemical process design. Part I:. AIChE Journal, 1985. 31(4): p. 621-630.

86. Sethi, A.K. and S.P. Sethi, Flexibility in manufacturing: a survey. Flexible Services and Manufacturing Journal, 1990. 2(4): p. 289-328.

87. Vokurka, R.J. and S.W. O'Leary-Kelly, A review of empirical research on manufacturing flexibility. Journal of Operations Management, 2000. 18(4): p. 485-501.

88. Shi, D. and R.L. Daniels, A survey of manufacturing flexibility: implications for ebusiness flexibility. IBM Systems Journal, 2003. 42(3): p. 414-427.

89. Gebauer, J. and F. Schober Information system flexibility and the performance of business processes. 2005.

90. Soffer, P. On the notion of flexibility in business processes. in Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005. 2005. Porto, Portugal.

91. Bider, I., State-Oriented Business Process Modeling: Principles, Theory and Practice. 2009: VDM Verlag. 124.

92. Regev, G., I. Bider, and A. Wegmann, Defining business process flexibility with the help of invariants. Software Process: Improvement and Practice, 2007. 12(1): p. 67-79.

93. Gupta, Y.P. and S. Goyal, Flexibility of manufacturing systems: concepts and measurements. European Journal of Operational Research, 1989. 43(2): p. 119-135.

94. Bates, D.W., et al., Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. J Am Med Inform Assoc, 2003. 10(6): p. 523-30.

95. Ash, J.S., M. Berg, and E. Coiera, Some unintended consequences of information technology in health care: the nature of patient care

information system-related errors. J Am Med Inform Assoc, 2004. 11(2): p. 104-12.

96. Kushniruk, A.W., et al., Exploring the relationship between training and usability: a study of the impact of usability testing on improving training and system deployment. Stud Health Technol Inform, 2009. 143: p. 277-83.

97. Rosenbloom, S.T., et al., Data from clinical notes: a perspective on the tension between structure and flexible documentation. J Am Med Inform Assoc, 2011. 18(2): p. 181-6.

98. Cummings, K. and R. McGowan, "Smart" infusion pumps are selectively intelligent. Nursing, 2011. 41(3): p. 58-9.

99. Rasmussen, J., A.M. Pejtersen, and L.P. Goodstein, Cognitive systems engineering. 1994: John Wiley \& Sons, Inc. 378.

100. Vicente, K.J., Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work. 1999, Mahwah, NJ: Lawrence Earlbaum Associates.

101. Jenkins, D.P., et al., Using cognitive work analysis to explore system flexibility. Theoretical Issues in Ergonomics Science, 2010. 11(3): p. 136-150.

102. Berenholtz, S.M., et al., Eliminating catheter-related bloodstream infections in the intensive care unit. Crit Care Med, 2004. 32(10): p. 2014-20.

103. Newell, A. and H.A. Simon, Human problem solving. 1972, Englewood Cliffs, NJ: Prentice-Hall.

104. Wolfram Research, I., Mathematica. Version 8.0. [software]. 2008, Wolfram Research, Inc.: Champaign, IL.

105. Thimbleby, H., Press on: Principles of interaction programming. 2007: The MIT Press.

106. Thimbleby, H., Formulating usability. SIGCHI Bull., 1994. 26(2): p. 59-64.

107. Thimbleby, H., P. Cairns, and M. Jones, Usability analysis with Markov models. ACM Trans. Comput.-Hum. Interact., 2001. 8(2): p. 99-132.

108. Gow, J. and H. Thimbleby, MAUI: An Interface Design Tool Based on Matrix Algebra, in Computer-Aided Design of User Interfaces IV, R.K.

Jacob, Q. Limbourg, and J. Vanderdonckt, Editors. 2005, Springer Netherlands. p. 81-94.

109. Byrne, M.D. and S. Bovair, A Working Memory Model of a Common Procedural Error. Cognitive Science, 1997. 21(1): p. 31-61.

110. Pronovost, P., et al., An intervention to decrease catheter-related bloodstream infections in the ICU. N Engl J Med, 2006. 355(26): p. 2725-32.

111. Shannon, C.E. and W. Weaver, The mathematical theory of communication. 1963, Champaign, IL: University of Illinois Press.

112. Hick, W.E., On the rate of gain of information. Quarterly Journal of Experimental Psychology, 1953. 4(1): p. 11-26.

113. Hyman, R., Stimulus information as a determinant of reaction time. J Exp Psychol, 1953. 45(3): p. 188-96.

114. Blumenthal, D. and M. Tavenner, The "meaningful use" regulation for electronic health records. N Engl J Med, 2010. 363(6): p. 501-4.

115. Ford, E.W., N. Menachemi, and M.T. Phillips, Predicting the adoption of electronic health records by physicians: when will health care be paperless? J Am Med Inform Assoc, 2006. 13(1): p. 106-12.

116. Schumacher, R.M., J.M. Webb, and K.R. Johnson, How to Select an Electronic Health Record System that Healthcare Professionals can Use. 2009, User Centric, Inc.: Oakbrook Terrace, IL.

117. Johnson, C.M., T. Johnson, and J. Zhang, Increasing productivity and reducing errors through usability analysis: a case study and recommendations. Proc AMIA Symp, 2000: p. 394-8.

118. Chung, P.H., et al., An extended hierarchical task analysis for error prediction in medical devices. AMIA Annu Symp Proc, 2003: p. 165-9.

119. Rinkus, S., et al., Human-centered design of a distributed knowledge management system. J Biomed Inform, 2005. 38(1): p. 4-17.

120. Chen, J.W. and J. Zhang, Comparing Text-based and Graphic User Interfaces for novice and expert users. AMIA Annu Symp Proc, 2007: p. 125-9.

121. Robb, G. and M. Seddon, Measuring the safety culture in a hospital setting: a concept whose time has come? N Z Med J, 2010. 123(1314): p. 68-78.

122. Poissant, L., et al., The impact of electronic health records on time efficiency of physicians and nurses: a systematic review. J Am Med Inform Assoc, 2005. 12(5): p. 505-16.

123. Bloom, M.V. and M.K. Huntington, Faculty, resident, and clinic staff's evaluation of the effects of EHR implementation. Fam Med, 2010. 42(8): p. 562-6.

124. Buntin, M.B., et al., The benefits of health information technology: a review of the recent literature shows predominantly positive results. Health Aff (Millwood), 2011. 30(3): p. 464-71.

125. Card, S.K., T.P. Moran, and A. Newell, The keystroke-level model for user performance time with interactive systems. Commun. ACM, 1980. 23(7): p. 396-410.

126. John, B.E., Information processing and skilled behavior, in Toward a multidisciplinary science of human computer interaction, J.M. Carroll, Editor. 2003, Morgan Kaufman: San Francisco. p. 55-102.

127. Gomez, S.R. and D.H. Laidlaw. Modeling Human Performance from Visualization Interaction Histories. in IEEE InfoVis. 2011. Providence, RI.

128. Amant, R.S., T.E. Horton, and F.E. Ritter, Model-based evaluation of expert cell phone menu interaction. ACM Trans. Comput.-Hum. Interact., 2007. 14(1): p. 1.

129. Koester, H.H. and S.P. Levine. Validation of a keystroke-level model for a text entry system used by people with disabilities. in ASSETS. 2004. Marina Del Rey, CA.

130. B, O., #228, and lter, Keystroke level analysis of email message organization, in Proceedings of the SIGCHI conference on Human Factors in Computing Systems. 2000, ACM: The Hague, The Netherlands. p. 105-112.

131. Pettitt, M., G. Burnett, and A. Stevens, An extended keystroke level model (KLM) for predicting the visual demand of in-vehicle information systems, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2007, ACM: San Jose, California, USA. p. 1515-1524.

132. John, B.E., et al., Predictive human performance modeling made easy, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2004, ACM: Vienna, Austria. p. 455-462.

133. Anderson, J.R., M. Matessa, and C. Lebiere, ACT-R: a theory of higher level cognition and its relation to visual attention. Hum.-Comput. Interact., 1997. 12(4): p. 439-462.

134. Nielsen, J., Guerilla HCI: Discount usability engineering to penetrate intimidation barrier, in Cost-justifying usability, R.G. Bias and D.J. Mayhew, Editors. 1994, Academic Press: San Diego.

135. Nielsen, J., Usability engineering. 1993, Cambridge, MA: Academic Press, Inc.

136. Thyvalikakath, T.P., T.K. Schleyer, and V. Monaco, Heuristic evaluation of clinical functions in four practice management systems: a pilot study. J Am Dent Assoc, 2007. 138(2): p. 209-10, 212-8.

137. Chan, J., et al., Usability evaluation of order sets in a computerised provider order entry system. BMJ Qual Saf, 2011. 20(11): p. 932-40.

138. Thyvalikakath, T.P., et al., Comparative study of heuristic evaluation and usability testing methods. Stud Health Technol Inform, 2009. 143: p. 322-7.

139. Johnson, C.M., et al., Can prospective usability evaluation predict data errors? AMIA Annu Symp Proc, 2010. 2010: p. 346-50.

140. Health IT and Patient Safety: Building Safer Systems for Better Care. 2011, National Academies Press (US). Copyright 2012 by the National Academy of Sciences. All rights reserved.: Washington (DC).

141. Lowry, S.Z., et al., Technical Evaluation, Testing, and Validation of the Usability of Electronic Health Records, U.S.D.o. Commerce, Editor. 2012, National Institute of Standards and Technology: Gaithersburg,MD.

142. Karsh, B.T., Beyond usability: designing effective technology implementation systems to promote patient safety. Qual Saf Health Care, 2004. 13(5): p. 388-94.

143. Schumacher, R.M. and S.Z. Lowry, NIST Guide to the Processes Approach for Improving the Usability of Electronic Health Records, U.S.D.o. Commerce, Editor. 2010, National Institute of Standards and Technology: Gaithersburg, MD.

144. Likourezos, A., et al., Physician and nurse satisfaction with an Electronic Medical Record system. J Emerg Med, 2004. 27(4): p. 419-24.

145. McGinn, C.A., et al., Comparison of user groups' perspectives of barriers and facilitators to implementing electronic health records: a systematic review. BMC Med, 2011. 9: p. 46.

146. Zhang, J., Human-centered computing in health information systems. Part 1: analysis and design. J Biomed Inform, 2005. 38(1): p. 1-3.

147. Middleton, B., et al., Enhancing patient safety and quality of care by improving the usability of electronic health record systems: recommendations from AMIA. J Am Med Inform Assoc, 2013. 20(e1): p. e2-8.

148. Test Procedure for §170.314(g)(3) Safety-enhanced design. Available from: http://www.healthit.gov/sites/default/files/170.314g3safetyenhanceddesign_2014_tp_approved_v1.3_0.pdf.

149. HeatlhIT.gov. Standards & Certifications Criteria Final Rule. April 9, 2014 August 5, 2014]; Available from: http://www.healthit.gov/policy-researchers-implementers/standards-certifications-criteria-final-rule.

150. Usability.gov. July 22, 2014]; Available from: http://www.usability.gov.

151. Standardization, I.O.f., Ergonomics of Human-System Interaction, in Part 210: Human-Centered Design for Interactive Systems. 2006, International Organization for Standardization: Geneva.

152. International Organization for Standardization, Ergonomic requirements for office work with visual display terminals (VDTs), in Part 11 : Guidance on usability. 1998, International Organization for Standardization: Geneva.

153. Standardization, I.O.f., Ergonomics of human-system interaction - Usability methods supporting human-centered design. 2002, International Organization for Standardization: Geneva.

154. Standardization, I.O.f., Software engineering -Software product Quality Requirements and Evaluation (SQuaRE) -Common Industry Format (CIF) for usability test reports,. 2006, International Organization for Standardization: Geneva.

155. Standardization, I.O.f., Medical devices – Application of usability engineering to medical devices. 2007, International Organization for Standardization: Geneva.

156. Certified Health IT Product List (CHPL). July 22, 2014]; Available from: http://www.healthit.gov/policy-researchers-implementers/certified-health-it-product-list-chpl.

157. Research-based web design & usability guidelines. 2006 [cited 2014 May 10]; Available from: http://guidelines.usability.gov/.

158. Kaner, C. What is a good test case? in Florida Institute of Technology Department of Computer Sciences STAR East. 2003.

159. HealthIT.gov. Test Procedure for §170.314(g)(3) Safety-enhanced design. 2013 March 29, 2013; Available from: http://www.healthit.gov/sites/default/files/170.314g3safetyenhanceddesign_2014_tp_approved_v1.3_0.pdf.

160. Update on the adoption of health information technology and related efforts to facilitate the electronic use and exchange of health information. 2013 June, 2013; Available from: http://www.healthit.gov/sites/default/files/rtc_adoption_of_healthit_and_relatedefforts.pdf.

161. Kellermann, A.L. and S.S. Jones, What it will take to achieve the as-yet-unfulfilled promises of health information technology. Health Aff (Millwood), 2013. 32(1): p. 63-8.

162. Bangor, A., P.T. Kortum, and J.T. Miller, An Empirical Evaluation of the System Usability Scale. International Journal of Human-Computer Interaction, 2008. 24(6): p. 574-594.

163. Xiao, Y., et al., Development of a Tool to Measure User Experience Following Electronic Health Record Implementation. J Nurs Admin, 2014. 44(7/8).

164. Edsall, R.L. and K.G. Adler, The 2012 EHR User Satisfaction Survey: responses from 3,088 family physicians. Fam Pract Manag, 2012. 19(6): p. 23-30.

165. Otieno, G.O., et al., Measuring effectiveness of electronic medical records systems: towards building a composite index for benchmarking hospitals. Int J Med Inform, 2008. 77(10): p. 657-69.

166. Baylor EHR User Experience Survey. Available from: http://www.himss.org/files/HIMSSorg/content/files/ehr_usability/Baylor_EHR_UserExperienceSurvey.pdf.

167. Eastaugh, S.R., The total cost of EHR ownership. Healthc Financ Manage, 2013. 67(2): p. 66-70.

168. Hendrich, A., et al., A 36-hospital time and motion study: how do medical-surgical nurses spend their time? Perm J, 2008. 12(3): p. 25-34.

169. Friedman, C.P., A "fundamental theorem" of biomedical informatics. J Am Med Inform Assoc, 2009. 16(2): p. 169-70.

170. Committee on Patient, S., T. Health Information, and M. Institute of, in Health IT and Patient Safety: Building Safer Systems for Better Care. 2011, National Academies Press (US) Copyright 2012 by the National Academy of Sciences. All rights reserved.: Washington (DC).

171. American Medical Association. Health IT Policy Committee's Workgroups on Certification/Adoption and Implementation. TESTIMONY OF THE AMERICAN MEDICAL ASSOCIATION. Implementation and Usability of Certified Electronic Health Records. 2013 July 23, 2013 [cited 2014 July 9]; Available from: http://www.healthit.gov/sites/default/files/archive/FACA%20Hearings/2013-07-23%20Standards%3A%20Implementation,%20Meaningful%20Use,%20and%20Certification%20%26%20Adoption%20WGs,%20%20Implementation%20%26%20Usability%20Hearing/ama_usabilitytestimony_0.pdf.

172. Force, H.E.U.T. EHR Usability Pain Points Survey Q4-2009. 2010 [cited 2013 June 6]; Available from: http://www.himss.org/files/HIMSSorg/content/files/Usability_Pain_PointsHIMSS10.pdf.

173. Kushniruk, A.W., et al., Technology induced error and usability: the relationship between usability problems and prescription errors when using a handheld application. Int J Med Inform, 2005. 74(7-8): p. 519-26.

174. Ash, J.S., et al., The extent and importance of unintended consequences related to computerized provider order entry. J Am Med Inform Assoc, 2007. 14(4): p. 415-23.

175. Butler , K.A., Usability engineering, in Encyclopedia of Computer Science & Technology, v.33, A. Kent and J.G. Williams, Editors. 1995, Taylor & Francis, Inc.: New York.

176. Woods, D. and E. Roth, Cognitive systems engineering, in Handbook of Human-Computer Interaction, J.P. Womack and D.T. Jones, Editors. 1989, North-Holland: New York.

177. Cain, C. and S. Haque, Organizational Workflow and Its Impact on Work Quality, in Patient Safety and Quality: An Evidence-Based Handbook for Nurses, R.G. Hughes, Editor. 2008: Rockville (MD).

178. White, S.A. and D. Miers, BPMN modeling and reference guide: Understanding and using the BPMN. 2008, Lighthouse Point, FL: Future Strategies Inc.

179. Fitts, P.M. and R.E. Jones, Analysis of Factors Contributing to 460 "Pilot-Error" Experiences in Operating Aircraft Controls. 1947, Wright-Patterson Air Force Base: Dayton, OH.

180. Bahrami, A., Object Oriented Systems Development. 1998, New York: McGraw-Hill.

181. Rumbaugh, J., I. Jacobson, and G. Booch, The Unified Modeling Language Reference Manual. 1st ed. 1998, Boston: Addison-Wesley.

182. Rosenfeld, L. and P. Morville, Information Architecture for the World Wide Web: Designing Large-Scale Web Sites. 2002, USA: O'Reilly Media.

183. MASSPRO, DOQ_IT: A systems approach to operational redesign -- Workbook. 2009, Medicare Quality Improvement Organization for Massachusetts.

184. Carayon, P. and B.T. Karsh, Incorporating health IT into workflow redesign: Request for information summary report. 2010, Agency for Healthcare Research and Quality: Rockville, Maryland.

185. Unertl, K.M., et al., Traversing the many paths of workflow research: developing a conceptual framework of workflow terminology through a systematic literature review. J Am Med Inform Assoc, 2010. 17(3): p. 265-73.

186. Zafar A, D.B., Burton M, Ekbia H, Lehto M. Multidisciplinary perspectives on best practices for understanding and evaluating clinical workflows. in AMIA 2010 Annual Symposium. 2010. Washington, D. C.

187. Chircu, A.M., et al. Handoffs and medication errors: A community hospital case study. in Proceedings of the 44th Hawaii Hawaii International Conference on System Sciences. 2011. Kauai, HI.

188. Lowry, S.Z., et al., Integrating Electronic Health Records into Clinical Workflow: An Application of Human Factors Modeling Methods to Ambulatory Care, U.S.D.o. Commerce, Editor. 2014, National Institute of Standards and Technology: Gaithersburg, MD.

189. Womack, J.P. and D.T. Jones, Lean thinking: banish waste and create wealth in your corporation. 1996, New York: Simon & Schuster.

190. Ma, Y., G. Chen, and G. Thimm, Paradigm shift: unified and associative feature-based concurrent and collaborative engineering. Journal of Intelligent Manufacturing, 2008. 19(6): p. 626-641.

191. Baldwin, D.M., et al., Patient preferences for notification of normal laboratory test results: a report from the ASIPS Collaborative. BMC Fam Pract, 2005. 6(1): p. 11.

192. Leekha, S., et al., Patient preferences for and satisfaction with methods of communicating test results in a primary care practice. Jt Comm J Qual Patient Saf, 2009. 35(10): p. 497-501.

193. SMART. SMART-Apps Competition. n. d.; Available from: http://smartapps.challengepost.com/.

194. Freidman, C.P. and J. Wyatt, Evaluation Methods in Biomedical Informatics. Health Informatics, ed. K.J. Hannah and M.J. Ball. 2005, USA: Springer Science+Business Media, Inc.

195. Eisenstein, E.L., et al., Defining a framework for health information technology evaluation. Stud Health Technol Inform, 2011. 164: p. 94-9.

196. Wharton, C., et al., The cognitive walkthrough method: A practitioner's guide, in Usability inspection methods, J. Nielsen and R. Mack, Editors. 1994, John Wiley & Sons, Inc.: New York.

197. Nielsen, J. 113 design guidelines for homepage usability. 2001; Available from: http://www.nngroup.com/articles/113-design-guidelines-homepage-usability.

198. Services, U.S.D.o.H.a.H. Research-based web design & usability guidelines. 2006 [cited 2014 May 10]; Available from: http://guidelines.usability.gov/.

199. Lowry, S.Z., et al., A Human Factors Guide to Enhance EHR Usability of Critical User Interactions when Supporting Pediatric Patient Care, U.S.D.o. Commerce, Editor. 2012, National Institute of Standards and Technology: Gaithersburg, MD.

200. Microsoft. Microsoft Health CUI design guidance. 2012; Available from: http://www.mscui.net/DesignGuide/DesignGuide.aspx.

201. Bates, D.W. and A.A. Gawande, Improving safety with information technology. N Engl J Med, 2003. 348(25): p. 2526-34.

202. Campbell, E.M., et al., Types of unintended consequences related to computerized provider order entry. J Am Med Inform Assoc, 2006. 13(5): p. 547-56.

203. Norman, D.A., The Design of Everyday Things: Revised and Expanded Edition. 2013, USA: Basic Books.

204. Tufte, E.R., The visual display of quantitative information. 2001, Cheshire, CT: Graphics Press.

205. Few, S., Now You See It: Simple Visualization Techniques for Quantitative Analysis. 1st ed. 2009, Oakland, CA: Analytics Press.

206. Weinschenk, S., 100 things every designer needs to know about people. 2011, Berkeley, CA: New Riders.

207. Ware, C., Information visualization: perception for design. 3rd ed. 2012, New York: Elsevier.

208. Plaisant, C., et al., LifeLines: using visualization to enhance navigation and analysis of patient records. Proc AMIA Symp, 1998: p. 76-80.

209. DesRoches, C.M., et al., Electronic health records in ambulatory care--a national survey of physicians. N Engl J Med, 2008. 359(1): p. 50-60.

210. Hersh, W.R., et al., Caveats for the use of operational electronic health record data in comparative effectiveness research. Med Care, 2013. 51(8 Suppl 3): p. S30-7.

211. Adler-Milstein, J., C.E. Green, and D.W. Bates, A survey analysis suggests that electronic health records will yield revenue gains for some practices and losses for many. Health Aff (Millwood), 2013. 32(3): p. 562-70.

212. DesRoches, C.M., et al., Adoption of electronic health records grows rapidly, but fewer than half of US hospitals had at least a basic system in 2012. Health Aff (Millwood), 2013. 32(8): p. 1478-85.

213. Goveia, J., et al., Educational interventions to improve the meaningful use of Electronic Health Records: a review of the literature: BEME Guide No. 29. Med Teach, 2013. 35(11): p. e1551-60.

214. Pronovost, P., et al., Medication reconciliation: a practical tool to reduce the risk of medication errors. J Crit Care, 2003. 18(4): p. 201-5.

215. Poon, E.G., et al., Design and implementation of an application and associated services to support interdisciplinary medication reconciliation efforts at an integrated healthcare delivery network. J Am Med Inform Assoc, 2006. 13(6): p. 581-92.

216. Poon, E., Medication reconciliation: whose job is it? Aorn j, 2009. 89(6): p. 1180, 1122.

217. Cadwallader, J., et al., Design of a medication reconciliation application: facilitating clinician-focused decision making with data from multiple sources. Appl Clin Inform, 2013. 4(1): p. 110-25.

218. de Koning, J.S., et al., Quality of care in stroke prevention: results of an audit study among general practitioners. Prev Med, 2004. 38(2): p. 129-36.

219. Commission, T.J. Using medication reconciliation to prevent errors. Sentinel Event Alert 2006 [cited 2014 June 30]; Available from: http://www.jointcommission.org/assets/1/18/SEA_35.PDF.

220. Aspden, P., et al., eds. Preventing medication errors. 2007, National Academies Press: Washington, D. C.

221. Keohane, C.A. and D.W. Bates, Medication safety. Obstet Gynecol Clin North Am, 2008. 35(1): p. 37-52, viii.

222. Roughead, E.E. and S.J. Semple, Medication safety in acute care in Australia: where are we now? Part 1: a review of the extent and causes of medication problems 2002-2008. Aust New Zealand Health Policy, 2009. 6: p. 18.

223. Ellitt, G.R., et al., Drug related problems after discharge from an Australian teaching hospital. Pharm World Sci, 2010. 32(5): p. 622-30.

224. Using medication reconciliation to prevent errors. Sentinel Event Alert, 2006(35): p. 1-4.

225. Silva, P.A., et al., Automated medication reconciliation and complexity of care transitions. AMIA Annu Symp Proc, 2011. 2011: p. 1252-60.

226. Burton, M.M., L. Simonaitis, and G. Schadow, Medication and indication linkage: A practical therapy for the problem list? AMIA Annu Symp Proc, 2008: p. 86-90.

227. McCoy, A.B., et al., Development and evaluation of a crowdsourcing methodology for knowledge base construction: identifying relationships between clinical problems and medications. J Am Med Inform Assoc, 2012. 19(5): p. 713-8.

228. Senathirajah, Y., D. Kaufman, and S. Bakken, Cognitive Analysis of a Highly Configurable Web 2.0 EHR Interface. AMIA Annu Symp Proc, 2010. 2010: p. 732-6.

229. Markowitz, E., et al., Medication Reconciliation: Work Domain Ontology, prototype development, and a predictive model. AMIA Annu Symp Proc, 2011. 2011: p. 878-87.

230. Tversky, B., Visualizing thought. Topics in Cognitive Science, 2011. 3(3): p. 499-535.

231. Plaisant, C., et al., Multi-Step Animation to Facilitate the Understanding of Spatial Groupings: the Case of List Comparisons, in Human-Computer Interaction Lab, Univeristy of Maryland. 2012, University of Maryland Tech Report: College Park, MD.

232. Hickner, J., et al., Testing process errors and their harms and consequences reported from family medicine practices: a study of the American Academy of Family Physicians National Research Network. Qual Saf Health Care, 2008. 17(3): p. 194-200.

233. Phillips, R.L., Jr., et al., Learning from malpractice claims about negligent, adverse events in primary care in the United States. Qual Saf Health Care, 2004. 13(2): p. 121-6.

234. Murff, H.J., et al., Primary care physician attitudes concerning follow-up of abnormal test results and ambulatory decision support systems. Int J Med Inform, 2003. 71(2-3): p. 137-49.

235. Wahls, T., T. Haugen, and P. Cram, The continuing problem of missed test results in an integrated health system with an advanced electronic medical record. Jt Comm J Qual Patient Saf, 2007. 33(8): p. 485-92.

236. Yackel, T.R. and P.J. Embi, Unintended errors with EHR-based result management: a case series. J Am Med Inform Assoc, 2010. 17(1): p. 104-7.

237. Bellotti, V., et al., What a to-do: studies of task management towards the design of a personal task list manager, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2004, ACM: Vienna, Austria. p. 735-742.

238. Tarkan, S., et al., Reducing missed laboratory results: defining temporal responsibility, generating user interfaces for test process tracking, and retrospective analyses to identify problems. AMIA Annu Symp Proc, 2011. 2011: p. 1382-91.

239. Few, S., Show Me the Numbers: Designing Tables and Graphs to Enlighten. 1st ed. 2004, Oakland, CA: Analytics Press.

240. Microsoft. Design Guidance – Displaying Graphs and Tables. 2008; Available from: http://www.mscui.net/DesignGuide/Pdfs/Design

%20Guidance%20--%20Displaying%20Graphs%20and%20Tables.pdf.

241. Poon, E.G., et al., Design and implementation of a comprehensive outpatient Results Manager. J Biomed Inform, 2003. 36(1-2): p. 80-91.

242. Payne, T.H. and J. Savarino, Development of a clinical event monitor for use with the Veterans Affairs Computerized Patient Record System and other data sources. Proc AMIA Symp, 1998: p. 145-9.

243. Singh, H., et al., Timely follow-up of abnormal diagnostic imaging test results in an outpatient setting: are electronic medical records achieving their potential? Arch Intern Med, 2009. 169(17): p. 1578-86.

244. Christov, S., et al., Formally defining medical processes. Methods Inf Med, 2008. 47(5): p. 392-8.

245. Endsley, M.R., B. Bolte, and D.G. Jones, Designing for Situation Awareness: An Approach to User-Centered Design. 1st ed. 2003, New York: Taylor & Francis Group.

246. Czerwinski, M., E. Horvitz, and S. Wilhite, A diary study of task switching and interruptions, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2004, ACM: Vienna, Austria. p. 175-182.

247. Iqbal, S.T. and B.P. Bailey, Effects of intelligent notification management on users and their tasks, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2008, ACM: Florence, Italy. p. 93-102.

248. Aragon, C.R., et al., Using visual analytics to maintain situation awareness in astrophysics, in Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on. 2008. p. 27-34.

249. Arora, V. and J. Johnson, A model for building a standardized hand-off protocol. Jt Comm J Qual Patient Saf, 2006. 32(11): p. 646-55.

250. Kripalani, S., et al., Deficits in communication and information transfer between hospital-based and primary care physicians: implications for patient safety and continuity of care. Jama, 2007. 297(8): p. 831-41.

251. Patel, V.L., et al., Impact of a computer-based patient record system on data collection, knowledge organization, and reasoning. J Am Med Inform Assoc, 2000. 7(6): p. 569-85.

252. Bourgeois, F.C., K.L. Olson, and K.D. Mandl, Patients treated at multiple acute health care facilities: quantifying information fragmentation. Arch Intern Med, 2010. 170(22): p. 1989-95.

253. Hall, A. and G. Walton, Information overload within the health care system: a literature review. Health Info Libr J, 2004. 21(2): p. 102-8.

254. Lissauer, T., et al., Evaluation of computer generated neonatal discharge summaries. Arch Dis Child, 1991. 66(4 Spec No): p. 433-6.

255. Horsky, J., G.J. Kuperman, and V.L. Patel, Comprehensive analysis of a medication dosing error related to CPOE. J Am Med Inform Assoc, 2005. 12(4): p. 377-82.

256. Singh, H., et al., Reducing diagnostic errors through effective communication: harnessing the power of information technology. J Gen Intern Med, 2008. 23(4): p. 489-94.

257. Sittig, D.F., et al., Improving clinical quality indicators through electronic health records: it takes more than just a reminder. Pediatrics, 2009. 124(1): p. 375-7.

258. Zeng, Q., J.J. Cimino, and K.H. Zou, Providing concept-oriented views for clinical data using a knowledge-based system: an evaluation. J Am Med Inform Assoc, 2002. 9(3): p. 294-305.

259. Kim, S., et al., Modeling the Arden Syntax for medical decisions in XML. Int J Med Inform, 2008. 77(10): p. 650-6.

260. Weed, L.L., Medical records that guide and teach. 1968. MD Comput, 1993. 10(2): p. 100-14.

261. Tange, H.J., et al., The granularity of medical narratives and its effect on the speed and completeness of information retrieval. J Am Med Inform Assoc, 1998. 5(6): p. 571-82.

262. Ash, J.S., et al., A rapid assessment process for clinical informatics interventions. AMIA Annu Symp Proc, 2008: p. 26-30.

263. Cole, W.G. and J.G. Stewart, Human performance evaluation of a metaphor graphic display for respiratory data. Methods Inf Med, 1994. 33(4): p. 390-6.

264. Bleich, H.L., Computer evaluation of acid-base disorders. J Clin Invest, 1969. 48(9): p. 1689-96.

265. Mickelson, J.K., et al., Is computer interpretation of the exercise electrocardiogram a reasonable surrogate for visual reading? Veterans Affairs ACME Investigators. Clin Cardiol, 1997. 20(4): p. 391-7.

266. Mandl, K.D., et al., The SMART Platform: early experience enabling substitutable applications for electronic health records. J Am Med Inform Assoc, 2012. 19(4): p. 597-603.

267. McCoy, A.B., et al., A prototype knowledge base and SMART app to facilitate organization of patient medications by clinical problems. AMIA Annu Symp Proc, 2011. 2011: p. 888-94.

268. Murphy, S.N., et al., Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc, 2010. 17(2): p. 124-30.

269. Mohammad, M., et al., Design and evaluation of a problem-oriented view for OpenVista EHR using UFuRT framework, in AMIA Annual Symposium. 2011: Washington, D. C.

270. Google. Google visualization API reference-Google chart tools. n. d.; Available from: https://developers.google.com/chart/interactive/docs/reference.

271. Sittig, D.F. and H. Singh, A new sociotechnical model for studying health information technology in complex adaptive healthcare systems. Qual Saf Health Care, 2010. 19 Suppl 3: p. i68-74.

272. Sittig, D.F. and H. Singh, Legal, ethical, and financial dilemmas in electronic health record adoption and use. Pediatrics, 2011. 127(4): p. e1042-7.

273. Sittig, D.F. and H. Singh, Defining health information technology-related errors: new developments since to err is human. Arch Intern Med, 2011. 171(14): p. 1281-4.

274. Myers, R.B., S.L. Jones, and D.F. Sittig, Review of Reported Clinical Information System Adverse Events in US Food and Drug Administration Databases. Appl Clin Inform, 2011. 2(1): p. 63-74.

275. Singh, H., D.C. Classen, and D.F. Sittig, Creating an oversight infrastructure for electronic health record-related patient safety hazards. J Patient Saf, 2011. 7(4): p. 169-74.

276. Wright, A., et al., Early results of the meaningful use program for electronic health records. N Engl J Med, 2013. 368(8): p. 779-80.

277. Sittig, D.F. and H. Singh, Rights and responsibilities of users of electronic health records. Cmaj, 2012. 184(13): p. 1479-83.

278. Sittig, D.F., H. Singh, and C.A. Longhurst, Rights and responsibilities of electronic health records (EHR) users caring for children. Arch Argent Pediatr, 2013. 111(6): p. 468-71.

279. Sittig, D.F. and H. Singh, Electronic health records and national patient-safety goals. N Engl J Med, 2012. 367(19): p. 1854-60.

280. Laxmisan, A., et al., Clinical Summarization Capabilities of Commercially-available and Internally-developed Electronic Health Records. Appl Clin Inform, 2012. 3(1): p. 80-93.

281. Mandl, K.D. and I.S. Kohane, No small change for the health information economy. N Engl J Med, 2009. 360(13): p. 1278-81.

282. McCoy, A.B., Smart Summarization App source code. n. d., GitHub.

283. Sittig, D.F., et al., Grand challenges in clinical decision support. J Biomed Inform, 2008. 41(2): p. 387-92.

284. Sittig, D.F., et al., A survey of informatics platforms that enable distributed comparative effectiveness research using multi-institutional heterogenous clinical data. Med Care, 2012. 50 Suppl: p. S49-59.

285. Draper, A., Managing bundled payments. Healthc Financ Manage, 2011. 65(4): p. 110-6, 118.

286. Sittig, D.F., K. Kahol, and H. Singh, Sociotechnical evaluation of the safety and effectiveness of point-of-care mobile computing devices: a case study conducted in India. Stud Health Technol Inform, 2013. 192: p. 515-9.

287. Esquivel, A., et al., Improving the effectiveness of electronic health record-based referral processes. BMC Med Inform Decis Mak, 2012. 12: p. 107.

288. Hysong, S.J., et al., Understanding the management of electronic test result notifications in the outpatient setting. BMC Med Inform Decis Mak, 2011. 11: p. 22.

289. Farri, O., et al., Impact of a prototype visualization tool for new information in EHR clinical documents. Appl Clin Inform, 2012. 3(4): p. 404-18.

290. Hahn, J.S., et al., Rapid implementation of inpatient electronic physician documentation at an academic hospital. Appl Clin Inform, 2012. 3(2): p. 175-85.

291. Feblowitz, J.C., et al., Summarization of clinical information: a conceptual model. J Biomed Inform, 2011. 44(4): p. 688-99.

292. Wright, A., E.S. Chen, and F.L. Maloney, An automated technique for identifying associations between medications, laboratory results and problems. J Biomed Inform, 2010. 43(6): p. 891-901.

293. McCoy, A.B., et al., Use of the Crowdsourcing Methodology to Generate a Problem-Laboratory Result Knowledge Base, in AMIA Annual Symposium. 2012.

294. Franzini, L., O.I. Mikhail, and J.S. Skinner, McAllen And El Paso revisited: Medicare variations not always reflected in the under-sixty-five population. Health Aff (Millwood), 2010. 29(12): p. 2302-9.

295. Parikh, R., et al., Cesarean section rate variation across hospital referral regions in Texas: A claims analysis of privately insured population from 2008-2011, in Think Global, Act Local, 141st APHA Annual Meeting. 2013: Boston.

296. Wright, A., et al., A method and knowledge base for automated inference of patient problems from structured data in an electronic medical record. J Am Med Inform Assoc, 2011. 18(6): p. 859-67.

297. Wright, A., et al., Improving completeness of electronic problem lists through clinical decision support: a randomized, controlled trial. J Am Med Inform Assoc, 2012. 19(4): p. 555-61.

298. Brown, S.H., et al., VA National Drug File Reference Terminology: a cross-institutional content coverage study. Stud Health Technol Inform, 2004. 107(Pt 1): p. 477-81.

299. Carter, J.S., et al., Initializing the VA medication reference terminology using UMLS metathesaurus co-occurrences. Proc AMIA Symp, 2002: p. 116-20.

300. Liu, S., et al., RxNorm: prescription for electronic drug information exchange. IT Professional, 2005. 7(5).

301. SNOMED Clinical Terms® (SNOMED CT®). August 8, 2014]; Available from: http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html.

302. Goethals, B. Survey on frequent pattern mining. 2003.

303. Wright, A., et al., Validation of an association rule mining-based method to infer associations between medications and problems. Appl Clin Inform, 2013. 4(1): p. 100-9.

304. Giles, J., Internet encyclopaedias go head to head. Nature, 2005. 438(7070): p. 900-1.

305. Howe, J., The rise of crowdsourcing. Wired Magazine, 2006. 14(6): p. 1-4.

306. Brownstein, C.A., et al., The power of social networking in medicine. Nat Biotechnol, 2009. 27(10): p. 888-90.

307. Hughes, S. and D. Cohen, Can online consumers contribute to drug knowledge? A mixed-methods comparison of consumer-generated and professionally controlled psychotropic medication information on the internet. J Med Internet Res, 2011. 13(3): p. e53.

308. eBay. Establishing a good eBay reputation: Buyer or Seller. n.d. [cited 2013 August 5, 2013]; Available from: http://www.ebay.com/gds/Establishing-a-good-eBay-reputation-Buyer-or-Seller/10000000001338209/g.html.

309. Amazon.com. Help: About Customer Ratings. [cited 2013 August 5, 2013]; Available from: http://www.amazon.com/gp/help/customer/display.html/ref=hp_left_sib?ie=UTF8&nodeId=200791020.

310. FAQ Slashdot. n. d. [cited 2013 August 5]; Available from: http://slashdot.org/faq.

311. McCoy, A.B., et al., Development of a clinician reputation metric to identify appropriate problem-medication pairs in a crowdsourced knowledge base. J Biomed Inform, 2014. 48: p. 66-72.

312. WHO International Classification of Diseases (ICD). 2010 August 8, 2014]; Available from: http://www.who.int/classifications/icd/en/.

313. Wright, A., et al., A description and functional taxonomy of rule-based decision support content at a large integrated delivery network. J Am Med Inform Assoc, 2007. 14(4): p. 489-96.

314. Hurst, J.W., Ten reasons why Lawrence Weed is right. N Engl J Med, 1971. 284(1): p. 51-2.

315. Weed, L.L., The problem-oriented record, in The Problem Oriented System, J.W. Hurst and H.K. Walker, Editors. 1972, Medcom Press: New York. p. 23-24.

316. Lincoln, M.J., Developing and implementing the problem list, in Computerizing Large Integrated Health Networks: The VA Success, R.M. Kolodner, Editor. 1987, Springer: New York. p. 349-381.

317. Safran, C., et al., A computer-based outpatient medical record for a teaching hospital. MD Comput, 1991. 8(5): p. 291-9.

318. Feinstein, A.R., The problems of the "problem-oriented medical record". Ann Intern Med, 1973. 78(5): p. 751-62.

319. Brown, S.H., et al., Empirical derivation of an electronic clinically useful problem statement system. Ann Intern Med, 1999. 131(2): p. 117-26.

320. Wang, S.J., et al., Automated coded ambulatory problem lists: evaluation of a vocabulary and a data entry tool. Int J Med Inform, 2003. 72(1-3): p. 17-28.

321. Campbell, J.R. and T.H. Payne, A comparison of four schemes for codification of problem lists. Proc Annu Symp Comput Appl Med Care, 1994: p. 201-5.

322. Mantena, S. and G. Schadow, Evaluation of the VA/KP problem list subset of SNOMED as a clinical terminology for electronic prescription clinical decision support. AMIA Annu Symp Proc, 2007: p. 498-502.

323. Fung, K.W., C. McDonald, and S. Srinivasan, The UMLS-CORE project: a study of the problem list terminologies used in large healthcare institutions. J Am Med Inform Assoc, 2010. 17(6): p. 675-80.

324. Nadkarni, P.M. and J.A. Darer, Migrating existing clinical content from ICD-9 to SNOMED. J Am Med Inform Assoc, 2010. 17(5): p. 602-7.

325. Weed, L.L., Knowledge coupling, medical education and patient care. Crit Rev Med Inform, 1986. 1(1): p. 55-79.

326. Weed, L.L. and N.J. Zimny, The problem-oriented system, problem-knowledge coupling, and clinical decision making. Phys Ther, 1989. 69(7): p. 565-8.

327. Van Vleck, T.T., et al., Content and structure of clinical problem lists: a corpus analysis. AMIA Annu Symp Proc, 2008: p. 753-7.

328. Bashyam, V., et al., Problem-centric organization and visualization of patient imaging and clinical data. Radiographics, 2009. 29(2): p. 331-43.

329. Tang, P.C., M.P. LaRosa, and S.M. Gorden, Use of computer-based records, completeness of documentation, and appropriateness of documented clinical decisions. J Am Med Inform Assoc, 1999. 6(3): p. 245-51.

330. Szeto, H.C., et al., Accuracy of computerized outpatient diagnoses in a Veterans Affairs general medicine clinic. Am J Manag Care, 2002. 8(1): p. 37-43.

331. Kaplan, D.M., Clear writing, clear thinking and the disappearing art of the problem list. J Hosp Med, 2007. 2(4): p. 199-202.

332. Carpenter, J.D. and P.N. Gorman, Using medication list--problem list mismatches as markers of potential error. Proc AMIA Symp, 2002: p. 106-10.

333. Hartung, D.M., et al., Clinical implications of an accurate problem list on heart failure treatment. J Gen Intern Med, 2005. 20(2): p. 143-7.

334. Services, C.f.M.M. Comparison of Meaningful Use Objectives Between the Proposed Rule to the Final Rule. 2010 [cited 2010.

335. Cao, H., et al., Mining a clinical data warehouse to discover disease-finding associations using co-occurrence statistics. AMIA Annu Symp Proc, 2005: p. 106-10.

336. Poissant, L., R. Tamblyn, and A. Huang, Preliminary validation of an automated health problem list. AMIA Annu Symp Proc, 2005: p. 1084.

337. Poissant, L., et al., Assessing the accuracy of an inter-institutional automated patient-specific health problem list. BMC Med Inform Decis Mak, 2010. 10: p. 10.

338. Feblowitz, J., et al., Provider use of and attitudes towards an active clinical alert: a case study in decision support. Appl Clin Inform, 2013. 4(1): p. 144-52.

339. Greenes, R., et al., The morningside initiative: collaborative development of a knowledge repository to accelerate adoption of clinical decision support. Open Med Inform J, 2010. 4: p. 278-90.

340. Pryor, T.A. and G. Hripcsak, The Arden syntax for medical logic modules. Int J Clin Monit Comput, 1993. 10(4): p. 215-24.

341. Jenders, R.A., et al., Standards in clinical decision support: activities in health level seven. AMIA Annu Symp Proc, 2008: p. 1244-5.

342. Jung, C.Y., K.A. Sward, and P.J. Haug, Executing medical logic modules expressed in ArdenML using Drools. J Am Med Inform Assoc, 2012. 19(4): p. 533-6.

343. eDecisions, H. Health eDecisions Homepage, S&I Framework. 2013 [cited 2013 May 10]; Available from: http://wiki.siframework.org/Health+eDecisions+Homepage.

344. Health Level Seven International. About HL7. September 30, 2014]; Available from: http://www.hl7.org/about/index.cfm?ref=nav.

345. W3C.org. Extensible Markup Language (XML) 1.1 (Second Edition). 2006 September 30, 2014].

346. Baader, F., et al., eds. The Description Logic Handbook: Theory, Implementation, and Applications. 2003, Cambridge University Press: Cambridge, UK.

347. W3C.org. OWL 2 Web Ontology Language Document Overview (Second Edition). 2012 September 30, 2014]; Available from: http://www.w3.org/TR/owl2-overview/.

348. Stanford University. Protégé. September 30, 2014]; Available from: http://protege.stanford.edu.

349. Sourceforge.net. The OWL API. September 30, 2014]; Available from: http://owlapi.sourceforge.net/.

350. W3C.org. SKOS Simple Knowledge Organization System - Home Page. September 30, 2014]; Available from: http://www.w3.org/2004/02/skos/.

351. The Dublin Core Metadata Initiative. Home page. September 30, 2014]; Available from: http://dublincore.org/.

352. Object Management Group. Documents Associated with Production Rule Representation (PRR) Version 1.0. 2009 September 30, 2014]; Available from: http://www.omg.org/spec/PRR/1.0/.

353. Object Management Group. Object Constraint Language (OCL). September 30, 2014]; Available from: http://www.omg.org/spec/OCL/.

354. Picca, D., A.M. Gliozzo, and A. Gangemi. LMM: an OWL-DL MetaModel to Represent Heterogeneous Lexical Knowledge. in LREC. 2008. European Language Resources Association.

355. Laboratory for Applied Ontology. DOLCE : a Descriptive Ontology for Linguistic and Cognitive Engineering. Available from: http://www.loa.istc.cnr.it/old/DOLCE.html.

356. Greenes, R.A., et al., Design of a standards-based external rules engine for decision support in a variety of application contexts: report of a feasibility study at Partners HealthCare System. Stud Health Technol Inform, 2004. 107(Pt 1): p. 611-5.

357. w3C.org. Resource Description Framework (RDF). 2014 September 30, 2014]; Available from: http://www.w3.org/RDF/.

358. ACM Transactions on Internet Technology (TOIT). ACM.

359. Play Framework. September 30, 2014]; Available from: https://www.playframework.com.

360. The Apache Software Foundation. Apache Tomcat. September 30, 2014]; Available from: http://tomcat.apache.org.

361. Google. Blocky. October 29, 2014]; Available from: https://developers.google.com/blockly/.

362. Hunter, J. What's new in Java Servlet API 2.2? Java World 1999 October 15, 1999; Available from: http://www.javaworld.com/article/2076518/java-web-development/what-s-new-in-java-servlet-api-2-2-.html.

363. National Quality Forum. Home page. September 30, 2014]; Available from: http://www.qualityforum.org/Home.aspx.

364. US National Library of Medicine. Unified Medical Language System® (UMLS®). 2009 July 18, 2014 September 30, 2014]; Available from: http://www.nlm.nih.gov/research/umls/.

365. Tao, C., et al., An OWL meta-ontology for representing the Clinical Element Model. AMIA Annu Symp Proc, 2011. 2011: p. 1372-81.

366. Patel, V.L. and G.J. Groen, Knowledge-based solution strategies in medical reasoning. Cognitive Science, 1986. 10(1): p. 91-116.

367. Kintsch, W. and J.G. Greeno, Understanding and solving word arithmetic problems. Psychol Rev, 1985. 92(1): p. 109-29.

368. Groen, G.J. and V.L. Patel, he relationship between comprehension and reasoning in medical expertise, in The Nature of Expertise, M.T.H. Chi, R. Glaser, and M.J. Farr, Editors. 1988, Lawrence Erlbaum: Hillsdale, NJ. p. 287-310.

369. Patel, V.L., J.F. Arocha, and D. Kaufman, Diagnostic reasoning and medical expertise, in The Psychology of Learning and Motivation: Advances in Research and Theory, D.L. Medin, Editor. 1994, CA: Academic Press: San Diego. p. 187–252.

370. Kannampallil, T.G., et al., Understanding the nature of information seeking behavior in critical care: implications for the design of health information technology. Artif Intell Med, 2013. 57(1): p. 21-9.

371. Zlabek, J.A., J.W. Wickus, and M.A. Mathiason, Early cost and safety benefits of an inpatient electronic health record. J Am Med Inform Assoc, 2011. 18(2): p. 169-72.

372. Bloomrosen, M., et al., Anticipating and addressing the unintended consequences of health IT and policy: a report from the AMIA 2009 Health Policy Meeting. J Am Med Inform Assoc, 2011. 18(1): p. 82-90.

373. Norman, D.A., Cognitive artifacts, in Designing interaction, M.C. John, Editor. 1991, Cambridge University Press. p. 17-38.

374. Taxis, K. and N. Barber, Ethnographic study of incidence and severity of intravenous drug errors. Bmj, 2003. 326(7391): p. 684.

375. Husch, M., et al., Insights from the sharp end of intravenous medication errors: implications for infusion pump technology. Qual Saf Health Care, 2005. 14(2): p. 80-6.

376. Norman, D.A., Cognitive engineering, in User centered system design, D.A. Norman and S.W. Draper, Editors. 1986, Lawrence Erlbaum Associates: Hillsdale, NJ.

377. Woods, D.D. and E.M. Roth, Cognitive engineering: human problem solving with tools. Hum. Factors, 1988. 30(4): p. 415-430.

378. Shortliffe, E.H., Computer-based medical consultations. 1976, New York: Elsevier.

379. Clancey, W.J. and R. Letsinger, NEOMYCIN: reconfiguring a rule-based expert system for application to teaching, in Readings in Medical Artificial Intelligence: The First Decade, W.J. Clancey and E.H. Shortliffe, Editors. 1984. p. 361-381.

380. Hassebrock, F. and M.J. Prietula, A protocol-based coding scheme for the analysis of medical reasoning. International Journal of Man-Machine Studies, 1992. 37(5): p. 613-652.

381. Chi, M.T.H., R. Glaser, and M.J. Farr, eds. The nature of expertise. 1988, Lawrence Erlbaum Associates, Inc.: New York.

382. Patel, V.L. and G.J. Groen, The general and specific nature of medical expertise: A critical look, in Toward a General Theory of Expertise: Prospects and Limits, A. Ericsson and J. Smith, Editors. 1991, Cambridge University Press: New York. p. 93-125.

383. Sharda, P., et al., Customizing clinical narratives for the electronic medical record interface using cognitive methods. Int J Med Inform, 2006. 75(5): p. 346-68.

384. Staszewski, J. Models of expertise as blueprints cognitive engineering: Applications to landmine detection. in Human Factors and Ergonomics Society 48th Annual Meeting. 2004. New Orleans.

385. Jayatilaka, L.G., et al., PETALS: a visual interface for landmine detection, in Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology. 2010, ACM: New York, New York, USA. p. 427-428.

386. Jayatilaka, L.G., et al., Evaluating a pattern-based visual support approach for humanitarian landmine clearance, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2011, ACM: Vancouver, BC, Canada. p. 453-462.

387. Bone, R.C., et al., Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. The ACCP/SCCM Consensus Conference Committee. American College of Chest Physicians/Society of Critical Care Medicine. 1992. Chest, 2009. 136(5 Suppl): p. e28.

388. Chase, W.G. and H.A. Simon, Perception in chess. Cognitive Psychology, 1973. 4: p. 55-81.

389. Patel, V.L., J.F. Arocha, and D.R. Kaufman, A primer on aspects of cognition for medical informatics. J Am Med Inform Assoc, 2001. 8(4): p. 324-43.

390. Arocha, J.F. and V.L. Patel, Construction-integration theory and clinical reasoning, in Discourse Comprehension: Essays in honor of Walter Kintsch, I. Weaver, C. A., S. Mannes, and C.R. Fletcher, Editors. 1995, Lawrence Erlbaum Associates, Inc.: Mahwah, NJ. p. 359–381.

391. Novak, J.D. and A.J. Cañas, The Theory Underlying Concept Maps and How to Construct Them. 2006, Florida Institute for Human and Machine Cognition: Pensacola, FL.

392. Cañas, A.J., et al. CMAPTOOLS: A knowledge modeling and sharing enviornment. in Concept Maps: Theory, Methodology, Technology. 2004. Pamplona, Spain: Proc. of the First Int. Conference on Concept Mapping.

393. Ericsson, K.A. and H.A. Simon, Protocol analysis: Verbal reports as data. Rev'd edition ed. 1993, Campbridge, MA: Bradford.

394. Strauss, A. and J. Corbin, Basics of qualitative research: Techniques and procedures for developing grounded theory. 2nd ed. 1998, Thousand Oaks, CA: Sage Publications Inc.

395. The New England Journal of Medicine. September 3, 2014]; Available from: http://www.nejm.org/medical-articles/clinical-problem-solving.

396. Chamarthi, B., M.F. Greene, and R.G. Dluhy, Clinical problem-solving. A problem in gestation. N Engl J Med, 2011. 365(9): p. 843-8.

397. SMART Platforms. [cited September 3, 2014; Available from: http://smartplatforms.org.

398. Saeed, M., et al., MIMIC II: a massive temporal ICU patient database to support research in intelligent patient monitoring. Comput Cardiol, 2002. 29: p. 641-4.

399. Cohen, T., Augmenting Expertise: Toward computer-enhanced clinical comprehension, in Department of Biomedical Informatics. 2007, Columbia University: New York.

400. Cohen, T., B. Blatter, and V. Patel, Simulating expert clinical comprehension: adapting latent semantic analysis to accurately extract clinical concepts from psychiatric narrative. J Biomed Inform, 2008. 41(6): p. 1070-87.

401. Evans, D.A. and C.S. Gadd, Managing coherence and context in medical problem-solving discourse, in Cognitive science in medicine: Biomedical modeling, D.A. Evans and V.L. Patel, Editors. 1989, MIT Press: Cambridge, MA.

402. Patel, V.L., D.A. Evans, and D.R. Kaufman, Cognitive framework for doctor-patient interaction, in Cognitive science in medicine: Biomedical modeling, D.A. Evans and V.L. Patel, Editors. 1989, MIT Press: Cambridge, MA. p. 253–308.

403. Dalai, V.V., et al., Evaluating the Effects of Cognitive Support on Psychiatric Clinical Comprehension. Aritificial Intelligence in Medicine, (under revision).

404. Groen, G.J. and V.L. Patel, The relationship between comprehension and reasoning in medical expertise, in The Nature of Expertise, M.T.H. Chi, R. Glaser, and M.J. Farr, Editors. 1988, Lawrence Erlbaum: Hillsdale, NJ. p. 287-310.

405. Hutchins, E., Cognition in the Wild. Bradford Books. 1985, Cambridge, MA: MIT Press.

406. Patel, V.L., G.J. Groen, and C.H. Frederiksen, Differences between medical students and doctors in memory for clinical cases. Med Educ, 1986. 20(1): p. 3-9.

407. Banks, J., et al., Discrete-event system simulation. 1984: Prentice-Hall.

408. eDecisions, H. Health eDecisions Homepage, S&I Framework. 2013 [cited 2014 May 10]; Available from: http://wiki.siframework.org/Health+eDecisions+Homepage.

409. Plaisant, C., et al., Twinlist: novel user interface designs for medication reconciliation. AMIA Annu Symp Proc, 2013. 2013: p. 1150-9.

The Office of the National Coordinator for Health IT (ONC) funded four **S**trategic **H**ealth information technology **A**dvanced **R**esearch **P**rojects in 2010. Better known as SHARP, the goal was improving the use of information technology in healthcare.

One of the projects was called SHARP**C**, which studied patient-centered **cognitive** support—the idea that information technology should support physicians' reasoning and decision-making while using electronic health records (EHR) systems.

SHARPC was a collaboration of eleven academic medical institutions and health researchers. The effort was led by The University of Texas School of Biomedical Informatics at Houston, which created the National Center for Cognitive Informatics and Decision Making in Healthcare.

This book documents the results of SHARPC's research: new theoretical frameworks, better ways of designing EHR systems, and new tools for implementing health information technology.

ISBN 978-0-692-26296-2
90000>
9 780692 262962

SHARPC
NCCD