Journal of Biomedical Informatics 44 (2011) 1056–1067

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

TURF: Toward a unified framework of EHR usability

Jiajie Zhang [a,b,*], Muhammad F. Walji [a,c]

[a] National Center for Cognitive Informatics and Decision Making in Healthcare, United States
[b] School of Biomedical Informatics, University of Texas Health Science Center at Houston, United States
[c] School of Dentistry, University of Texas Health Science Center at Houston, United States

ARTICLE INFO

Article history:
Received 20 May 2011
Accepted 5 August 2011
Available online 16 August 2011

Keywords:
Usability
Useful
Usable
Satisfying
User
Task
Representation
Function
Usability metrics
EHR
TURF

ABSTRACT

This paper presents a unified framework of EHR usability, called TURF, which 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) once fully developed, a potential principle for developing EHR usability guidelines and 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. TURF provides a set of measures for each of the useful, usable, and satisfying dimensions of usability. TURF stands for task, user, representation, and function, which are the four components that determine the usability of an EHR system. These four components are described with theoretical descriptions along with examples of how usability is measured in several case studies. How TURF can be used to improve usability through redesign is also demonstrated in a case study. In summary, this paper states that usability can not only be defined scientifically under a coherent, unified framework, it can also be measured objectively and systematically.

1. Introduction

Electronic Health Records (EHR) systems have great potential to increase care quality, efficiency, and safety through its wide adoption and meaningful use [1–7]. This is the major rationale behind the national HIT Initiative, started by President Bush in 2004 and strengthened by President Obama in 2009 with the $19 billion HITECH Act under ARRA, to have every American's medical records 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 [8–11]. The cognitive challenge is mainly concerned with usability issues, which did not receive significant attention in the EHR community until recently [11–21]. 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 the lack of sufficient attention to usability and the lack of EHR-specific usability frameworks and methods. Designing and implementing EHR is not so much an IT project as a human project about usability, workflow, patient safety, and organizational change [8,11,18,22–27]. To facilitate the adoption and meaningful use of EHR, we need an EHR-specific usability framework that can be used to increase efficiency and productivity, increase ease of use and ease of learning, increase user retention and satisfaction, and decrease human errors, decrease development time and cost, and decrease support and training cost. In this paper we present the initial form of a unified framework of EHR usability, called TURF, that has the following properties: (1) for describing, explaining, and predicting usability differences; (2) for defining, evaluating, and measuring usability objectively; and (3) for designing built-in good usability. Once fully developed, TURF could also be used as a principle for developing EHR usability guidelines and standards.

2. Definition of usability

Under TURF, usability refers to 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. Useful, usable, and satisfying are the three major dimensions of usability under TURF (see Table 1).

TURF definition of usability is based on the ISO definition (ISO 9241-11) but differs from it in significant ways. ISO defines usability as "the extent to which a product can be used by specified users

* Corresponding author at: School of Biomedical Informatics, University of Texas Health Science Center at Houston, United States. Fax: +1 713 486 0117.

E-mail addresses: Jiajie.Zhang@uth.tmc.edu (J. Zhang), Muhammad.F.Walji@uth.tmc.edu (M.F. Walji).

1532-0464/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2011.08.005

Table 1
Dimensions and measures of usability under TURF.

	Dimensions	Descriptions	Representative measures
	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
Usability	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

to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.'' Under ISO definition of usability, effectiveness refers to the accuracy and completeness with which users achieve specified goals; efficiency refers the resources expended in relation to the accuracy and completeness with which users achieve goals; and satisfaction refers to the comfort and acceptability of use.

TURF and ISO definitions of usability differ in the difference between ''effective'' in ISO and ''useful'' in TURF and between ''efficient'' in ISO and ''usable'' in TURF. Under TURF, ''useful'' refers to how well the system supports the work domain where the users accomplish the goals for their work, independent of how the system is implemented. A system is fully useful if it includes the domain and only the domain functions that are 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 the development of new knowledge, with the availability of innovations in technology. Usefulness can be measured by the percentage of domain functions that are in the EHR over all domain functions (those in the system and those not in the system), and the ratio of domain functions vs. non-domain functions in the system. More details about domain functions are described in Section 3.2.

Under TURF, a system is usable if it is easy to learn, efficient to use, and error-tolerant. How usable a system is can be measured by learnability, efficiency, and error tolerance. Learnability refers to the ease of learning and re-learning. It 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, the number of items that need to be memorized, and the number of sequences of task steps that need 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 steps 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 times a task can be successfully completed. Task success rate is referred to as the completion rate of tasks and as a measure of effectiveness under ISO definition of usability. Under TURF, however, effectiveness, including task success rate, is considered as one of the measures of efficiency because it is a measure of user performance, just like time on task. And mental effort, under TURF, is the amount of mental effort required for the task, such as the percentage of mental steps over all steps (physical and mental). Error prevention and recovery refers to the ability of the system to help users prevent and recover from errors. Error can be measured by frequency of errors, recovery rate of errors, and other measures. Under ISO definition of usability, error is one of the measures of effectiveness. Under TURF error is one of the measures of efficiency, for the same reason that task success rate is considered as an efficiency measure under TURF.

Satisfaction under TURF is similar to satisfaction under ISO definition of usability. Under TURF, satisfaction refers to the subjective impression of how useful, usable, and likable the 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's satisfaction is an important component of usability. However, this aspect of usability 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 of usability under TURF are objective, evidence-based, and systematic. Only when both of them are considered is usability evidence-based. Satisfaction alone should never be used as the complete measure of EHR usability.

TURF considers usefulness as a major dimension of usability because TURF takes a work-centered approach to usability [28–32]. Usefulness is also often referred to as utility or functionality. And its importance in successful applications has been long acknowledged. For example, Landauer argued that successful applications should be not only usable, their functionality should also be useful [33]. Goransson and colleagues [34] compiled a list of applications that have 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 to users. 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. [35] who developed a work-centered framework on how to allocate functionality across machines and users. If the system does not have a desired function, the users may have to find a workaround workflow that could complicate the usableness of the system. Thus the choice of functionality will not only determine how useful it is but also how usable it is [36]. For this reason, usefulness (functionality or utility) is considered as an integral component of usability under TURF.

3. TURF

The essence of usability is the ***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 [37–41]. The form of a 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 [41].

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 EHR for a display, a module, or the entire system is also a representation effect. In Fig. 1, the 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 the usefulness of the system. 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 user uses a specific representation or user interface to perform a specific task and it is an indication of the usableness of the system. Extrinsic difficulty is mainly determined by the formats of representations and the workflows of tasks. Intrinsic complexity and extrinsic difficulty together reflect the usability of the system. The next few sub-sections describe the details of intrinsic complexity and extrinsic difficulty in terms of the four components of TURF: Task, User, Representation, and Function, along with the results of several case studies.

EHR systems, just like many other products, are used in real world settings which are typically interruption-laden, unpredictable, and stressful, and involve many other factors such as organizational, social, physical, spatial, temporal, financial, and historical factors. All of these factors can contribute to the representation effect in various ways and they should always be considered in the

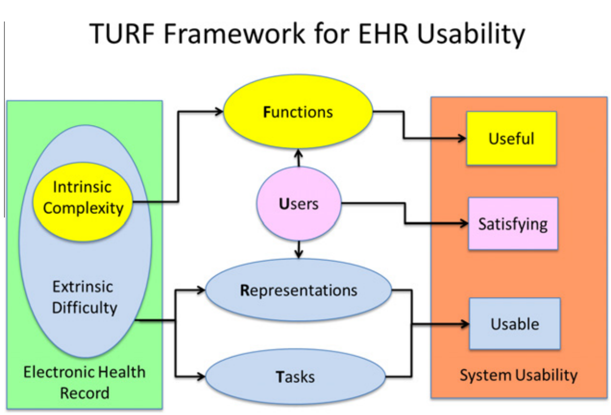

Fig. 1. The TURF framework of EHR usability. See text for details.

design and evaluation of EHR usability. The focus of this paper, however, is only on the uninterrupted tasks performed by individual users.

TURF is an expansion of the UFuRT framework developed earlier in our research [31,42,43]. TURF is based on the work-centered research [31,35,39,42]. TURF stands for the four key components of usability: Task, User, Representation, and Function. TURF is proposed as a framework for (1) describing, explaining, and predicting usability differences in terms of the representation effect; (2) for defining, evaluating, and measuring usability objectively; (3) for designing built-in good usability; and (4) for developing EHR usability guidelines and standards. This paper focuses on the first three aspects. We are also in the process of developing a software application that implements a subset of the TURF features to partially automate some usability evaluation processes, measure usability along several metrics, and analyze usability and patient safety patterns. In the near future, we will use TURF as a principle in the development of EHR usability guidelines and standards.

3.1. User analysis

User analysis is the first step of applying TURF for the design and evaluation of usability. It provides user information necessary to conduct function, representation, and task analyses. User analysis is the process of identifying the types of users and the characteristics of each type of users. For the EHR domain, 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 at various specializations, medical technicians, medical staff, patients and family members, and so on. User characteristics for each type of users 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 us design systems that have the right knowledge and information structure that match those of the users. There are many established methods for doing user analysis in textbooks (e.g., [44]) which we will not duplicate in this paper.

3.2. Function analysis

3.2.1. Work domain ontology

Function analysis is the process of identifying a work domain's abstract structure: the ontology of the work domain [32,35]. The ontology of work domain is the basic structure of the work that the system together with its human users will perform. It is an explicit, abstract, implementation-independent description of that work. It describes the essential requirements of that work independently of any technology systems, strategies, or work procedures; it tells us the inherent complexity of the work, it separates work context (physical, social, organizational, etc.) from the inherent nature of the work; and it supports identification of overhead activities that are non-essential for the work but introduced solely due to the way the system is implemented. In other words, work domain ontology is invariant with respect to work context, application technology, or cognitive mechanisms. If the system does not support the ontology of the work, the system will fail, regardless of its large collection of functions, fancy and cutting-age features, and purely technical merits.

Work domain ontology has four components: goals, objects, operations, and constraints. Operations are performed on the objects under the constraints to achieve the goals. Let us consider the following example: Dr. Townshend prescribes 90 day supply of Metformin 500 mg tablets by mouth twice daily to patient John Joe who is a pre-diabetic patient 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 patient name, doctor's name, diagnosis, medication name, dosage, frequency, duration, route, etc.; the *constraints* include the dependency relations between operation and objects (e.g., operation "write a medication prescription" and the objects "Metformin" and "500 mg"), between objects (e.g., the object "glucose level" and the object "Metformin"), and between operations (e.g., the operation "write a prescription" and the operation "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 the operations in the work domain ontology specify the functionality (or utility) of the system. The identification of the operations and their relations in the function hierarchy is the most important task for establishing the work domain ontology of the system. For our current 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 the set of functions that are implemented in an EHR the *Designer Model*. Identifying the functions in the Designer Model is relatively unambiguous as the functions in an EHR system are defined as all the 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 the functions in the User Model involves interviews and surveys and ambiguities of functions in the User Model can be minimized through systematical application of ontology engineering methods and qualitative methodologies. And the set of functions that are actually used in real activities by users is called the *Activity Model*. The functions in the Activity Model are typically identified through ethnography and extensive qualitative data analyses. For an ideal design with perfect functionality, these three models should be identical. However, discrepancies of functions across the three models are almost always present and they are the subject of the function analysis and they offer the opportunities for design improvement. One recent doctoral graduate in our lab developed a methodology for reducing the function discrepancies across the three models as part of her doctoral dissertation [45]. She described the function discrepancies as seven areas in the Venn diagram in Fig. 2.

The left side of Fig. 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 it was obtained through a complete system walk-through. The User Model has 80 functions and it was developed by conducting interviews and surveys with end users. The Activity Model has 97 functions and it was developed by doing a field study that involved 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 a medication) that were not directly relevant for the EDR. The functions in the three models were matched and merged into 190 functions in the Integrated Model. The functions in the Integrated Model (167, excluding the 23 clinical functions) were given in a survey to end users who rated each function on a 1-to-5 Likert scale for both usefulness and criticality of each function. Eighty functions received an average rating of 3 or above for both usefulness and criticality (see the right side of Fig. 3) and they were operationally defined as domain functions – the functions in the work domain ontology of the EDR. The functions with ratings below 3 are called overhead functions.

Fig. 3 shows a few interesting points. First, 73% of the functions that are in the system, wanted by the users, and used in activities are included in the ontology (with a rating of 3 or above for both usefulness and criticality). It means that a function is very likely to be part of the ontology it is in all three models. Second, about half (52%) of the functions that are 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 that the users are not aware of and do not use in their activities, and that they represent innovations by the vendor. On the other hands, the other half of the functions in the same category are not considered useful and excluded from the ontology. These excluded functions are overhead functions that are not essential for the work domain and can add to the intrinsic complexity of the system (see Fig. 1). Third, 80% of the functions

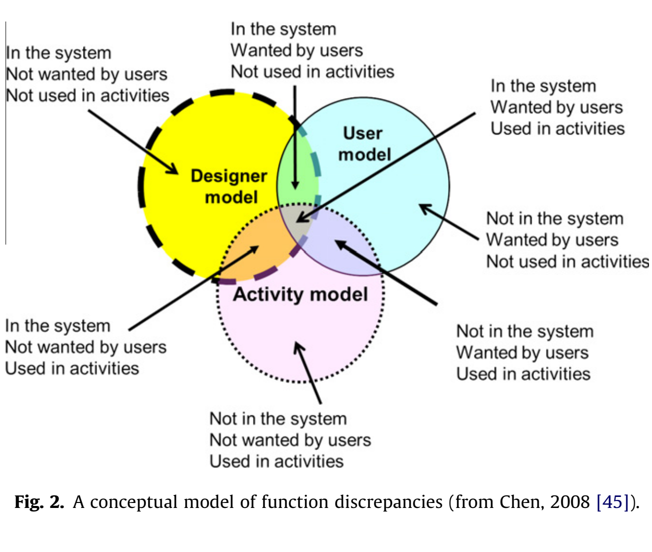

Fig. 2. A conceptual model of function discrepancies (from Chen, 2008 [45]).

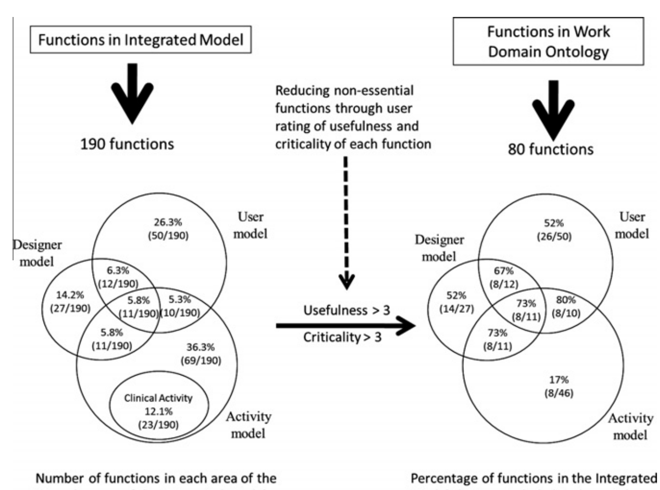

Fig. 3. The left Venn diagram shows the number of functions in each of the areas defined in Fig. 2 for an Electronic Dental Records (EDR) system. The right Venn diagram shows the percentage of functions in each of the areas on the left that are included in the work domain ontology, which is defined by the set of functions that are rated 3 or above for both usefulness and criticality on a 1–5 Likert scale by users (from Chen, 2008 [45]).

that are wanted by the users and used in activities are included in the ontology. In addition, 52% of the functions wanted by the users but not in the system and not used in activities are also included in the ontology. This means that there are many important domain functions that are wanted by the users but not available in the system and they should be added in future updates of the system. Fourth, only 17% of the functions that are used in activities but not in the system and not wanted by the users are included in the ontology. This means that most of the functions in this category are considered by the users as inappropriate for inclusion in an electronic system, at least for the time being. Most of the functions (about three quarters) that are included in two or more models are included in the ontology. This means that the functions that have cross model agreement are likely to be the functions that are useful.

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

1. *Within-Model Domain Function Saturation: Percentage of domain functions in the Designer Model over all functions in the Designer Model*. This is the ratio of the number of functions in the Designer Model that are included in the ontology over the total number of functions in the Designer Model. For the EDR system in Fig. 3, the ratio is 38/60 = 63%.This ratio means that 63% of the functions in the EDR are considered useful by the users, i.e., 37% are overhead functions that are not useful.
2. *Across-Model Domain Function Saturation: Percentage of domain functions in the Designer Model over domain functions in all three models*. This is the ratio of the number of functions in the Designer Model that are included in the ontology over the total number of functions in all three models (Designer, User, and Activity Models) that are included in the ontology. For the system in Fig. 3, the ratio is 38/80 = 48%.This ratio means that the EDR system has implemented about 48% of all domain functions that are considered useful by the users.
3. *Across-Model Function Saturation: Percentage of all functions in the Designer Model over all functions in all three models*. This is the ratio of the number of all functions in the Designer Model over the total number of all functions in all three models (Designer, User, and Activity Models). For the system in Fig. 3, the ratio is 60/190 = 32%. This ratio means that the EDR system has implemented about 32% of all functions that are proposed by the designers, wanted by the users, and used in activities. This ratio does not exclude the non-domain (overhead) functions in the three models that are considered not useful by the users. This third ratio is similar to the second one, although it is not so direct a measure of usefulness as the second one. The advantage of the 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 the three usefulness metrics that need further discussion. First, the functions in the User Model and the Activity Model are empirical data collected from interviewing, surveying, and observing the 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 the purpose of the evaluation.

3.2.3. Domain vs. overhead functions through expert review

In the last section we discussed the relationship of the functions in the three models: functions available in an EHR system, the functions wanted by the users, and the functions actually used in real activities. The method used to conduct the analysis described in the previous section is based on empirical data and usually requires a lot of effort and resources. In this section, we focus on the functions in the Designer Model only and describe a relatively more efficient expert review method we developed in the usability evaluation of the AHLTA (Armed Forces Health Longitudinal Technology Application) EHR system [46].

This method starts with the identification of the system hierarchy of an EHR system. The system hierarchy was created by visually inspecting the user interface items from top to bottom and left to right. Each interface item (label, field, drop-down menu etc.) is 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. The AHLTA EHR has six levels with more than nearly two thousand items. The first three levels of the system hierarchy of AHLTA are shown in Fig. 4.

Each interface item in the system hierarchy 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. Domain function was an operation inherent in and necessary for the work domain rather than dependent on the artifacts or interfaces, whereas overhead function was an operation introduced to deal with specific implementations of the user interface rather than the work domain. Fig. 5 shows that among the 1996 interface items identified in the AHLTA hierarchy, 61% were classified as Operations and 39% as Objects (kappa > 0.6 for inter-rater reliability between the two evaluators). Of the 1218 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 one of the metrics of usefulness in a more efficient way: **Percentage of domain functions in the Designer Model over all functions in the Designer Model** through expert review. In Section 3.2.2 the percentage was obtained through a much more effortful empirical data collection process. From the AHLTA study, this percentage was obtained through the assessment by two expert evaluators. Although the process still requires significant effort, it is much more efficient than the method using empirical data collection. From this expert review process, the usefulness metric for the AHLTA EHR as defined

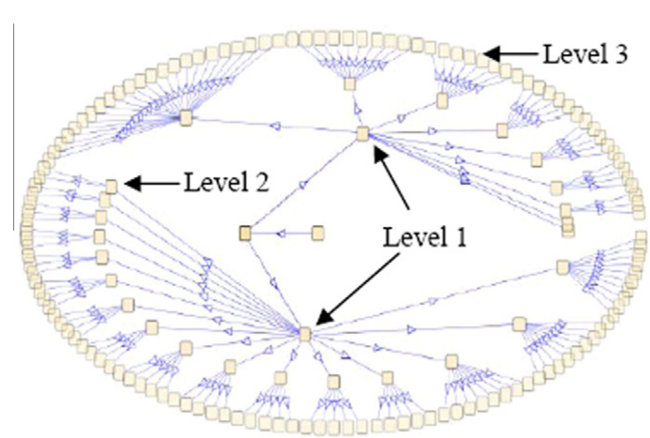

Fig. 4. Visualization of the top three levels of the six levels of the system hierarchy of AHLTA user interface (from [46]).

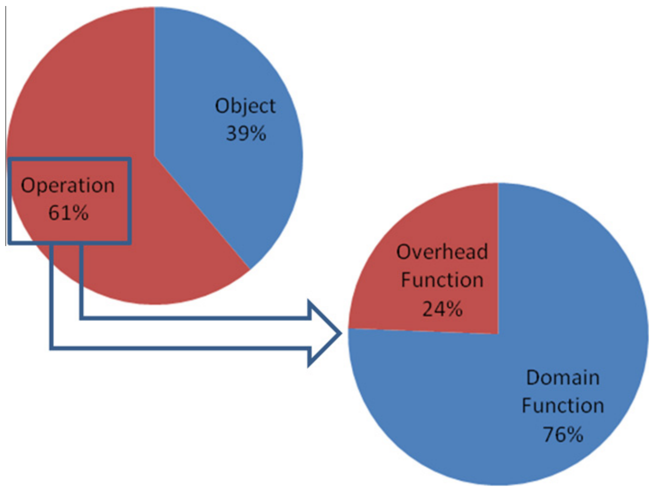

Fig. 5. Among the 1996 interface items in ALHTA EHR, 39% are objects and 61% are operations. Out of the 1218 operations, 76% are domain functions and 24% are overhead functions (from [46]).

by the percentage of domain functions in the Designer Model over all functions in the Designer Model is 76%. The detailed results for the breakdown components of the AHLTA EHR are shown in Fig. 6, which shows that most of the functions in the "summary" subsection are overhead functions and are not useful, whereas most the functions in the "readiness" subsection are domain functions and are useful.

3.3. Representation analysis

Representation analysis is the process of evaluating the appropriateness of the representations for a given task performed by a specific type of users such that the interaction between users and systems is in a direct interaction mode [47]. Representation analysis is based on the representation effect described in Section 3 [37–41]: different representations of a common abstract structure can generate dramatically different representational efficiencies, task difficulties, and behavioral outcomes. One major type of representation analysis is to compare a representation with isomorphic representations of the same structure and determine 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 and this is described in Section 3.3.2. Expert review of usability violations against well-established principles also includes various types of representation analysis and this is described in Section 3.3.3. There are many other types of representation analysis, 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.

3.3.1. Isomorphic representations

Identifying and generating isomorphic (functionally equivalent but computationally different) representations is a major type of representation analysis. The work domain ontology is the common abstract structure that can be implemented in many different 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. These different representations have 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 [40] to demonstrate how to use isomorphic representations as a representation analysis. Relational information displays are a major category of displays in EHR systems.

Fig. 7 shows the representation taxonomy of relational information displays – displays that represent relations such as tabular and graphic displays [40]. This 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, the 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 representations, each scale type can be implemented by different physical dimensions. In Fig. 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),

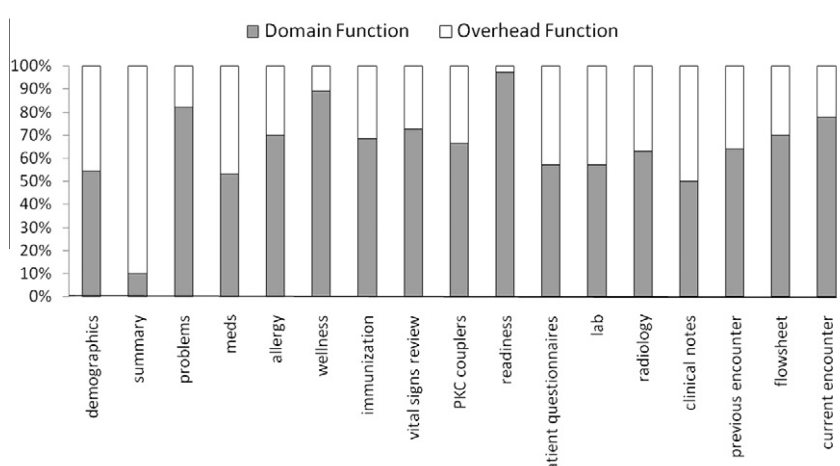

Fig. 6. Percentage of domain vs. overhead functions in each of the subsections of the AHLTA patient record section (from [46]).

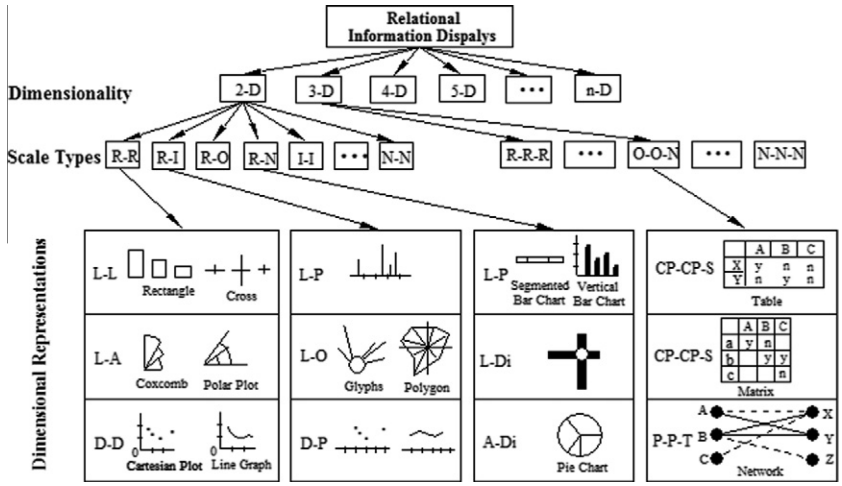

Fig. 7. A representation taxonomy of relational information displays (from [40]).

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.

This taxonomy of relational information displays can be used for two types of representation analysis for EHR. The first analysis is to analyze the dimensions of component displays (e.g., the flow sheet table in an EHR) and evaluate whether each dimension in the display is appropriately represented according to the taxonomy. The second analysis is to use the taxonomy to generate new designs. Once the 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 the 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. There are many other EHR user interfaces that are more granular or more abstract than relational information displays. Developing a comprehensive library of EHR user interface representations along with the mappings to tasks and users is a major ongoing effort in our EHR Usability Lab at the National Center for Cognitive Informatics and Decision Making in Healthcare.

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 the set of allowable actions specified by the display coupled with the knowledge of the user [50,51]. It 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 the action clicking, it does not afford it 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 of AHLTA. Two evaluators independently analyzed each operation and determined the degree of affordance. Any discrepancies in ratings were resolved by consensus after further discussion. Each operation was 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

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

In further analysis we plan to extend representation analysis to classify the degree of correct or incorrect mappings between AHLTA displays and specific tasks. Ideally the information perceivable from a display should exactly match the information required for the task, no more and no less. In other words, the tasks assigned to a display should be the tasks afforded by the external representations of the display and the displays assigned to a task should be the displays whose external representations support the task [50].

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], consists of a large portion of representation analysis. Heuristic evaluation is an easy to use, easy to learn, discount usability evaluation technique used to identify major usability problems of a product in a timely manner with reasonable cost. This technique requires a few evaluators to independently apply a set of usability principles to a product, identify violations of the principles, and assess the 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 fourteen principles customized for the health domain [57]. We have since applied the fourteen principles to a variety of healthcare domains [57–60]. The fourteen principles are as follows:

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 principles (Consistency, Visibility, Match, Minimalist, Memory, and Feedback) are all about representation properties of user interfaces and they are considered as one type of representation analysis. Fig. 8 shows the evaluation of the AHLTA EHR using the fourteen principles. The evaluation was performed by three independent evaluators whose results were integrated into a master list of all violations. Then each evaluator independently rated each violation for its severity on a scale of 1 to 4 (1 = cosmetic; 2 = minor; 3 = major; 4 = catastrophic) and their ratings were then averaged, as shown in Fig. 9. Fig. 10 shows details results of where the violations occur for the Health History module of the AHTLA. The violations were documented in details and recommendations for addressing each violation were generated.

Representation analysis through expert review of usability principles is an efficient method that is inclusive of a large range of usability violations and 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 our EHR Usability Lab is to develop and validate a reliable, systematic, and operationalized process for a subset of the usability principles that are relevant to representations.

3.4. Task analysis

Task analysis is loosely defined in many different ways in the literature [61,62]. For our current discussion of EHR usability, we define task analysis as the process of identifying the steps of carrying out an operation by using a specific representation, the relations among the steps, and the nature of each step (mental or physical). Our definition of task analysis is based on the GOMS approach to task analysis [63,64]. One important point about cognitive task analysis is that the steps include not only physical steps but also mental steps. By considering mental steps, we can identify the cognitive factors that make a task easy or difficult [39,65]. In addition, the 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). One important objective of task analysis is to find out which representation is better for which task, why it is better, and how to generate a better representation. By performing task analysis 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, which are all metrics of efficiency for usability (see Table 1).

We have conducted a series of task analysis for many EHR systems. In the following, we describe one task analysis study we did for the AHLTA EHR [66]. In this study, we used the Keystroke Level Modeling (KLM) to estimate time on task, task steps, and mental effort for fourteen prototypical use cases for the AHLTA EHR. KLM is a well-established and validated method that estimates performance levels by experts [67,68]. Over one hundred research publications have shown that the performance levels generated by KLM are within 20% of expert performance through empirical studies [63,69]. The 14 use cases, which were provided to us by expert AHLTA clinician users, are:

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 the 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.

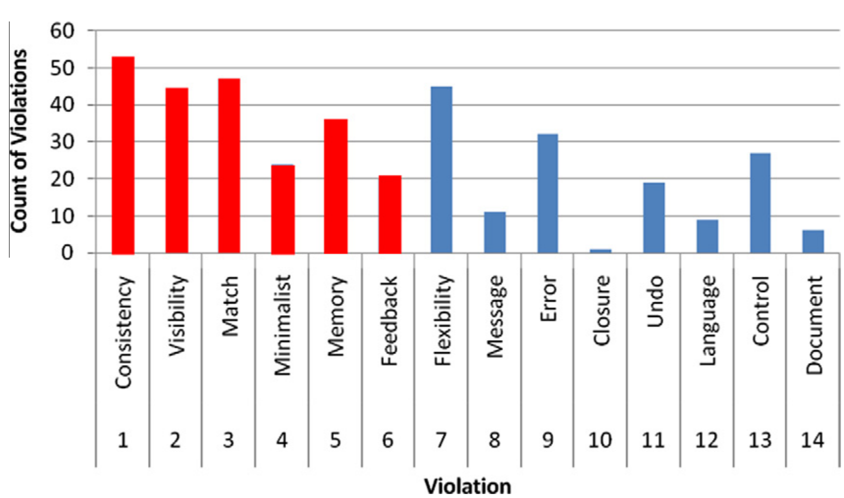

Fig. 8. Violations of usability principles for the AHLTA EHR. The first six principles (consistency, visibility, match, minimalist, memory, and feedback) are all about the representation properties of the user interfaces, and they are considered as one type of representation analysis.

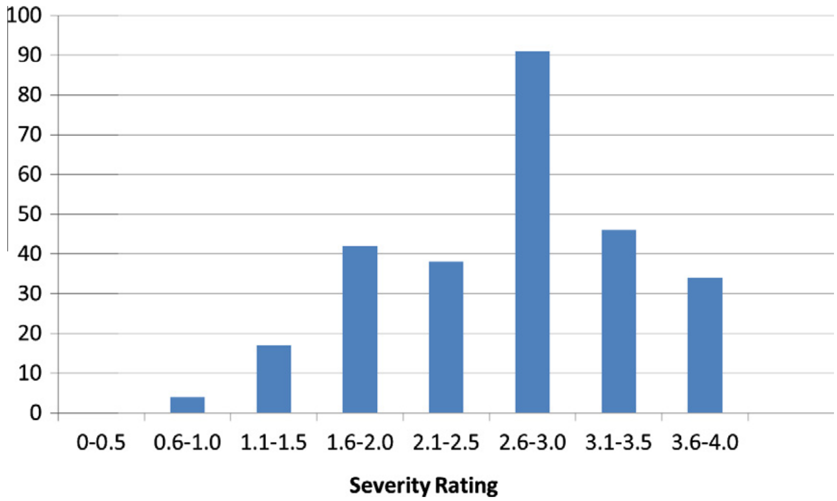

Fig. 9. Severity ratings of the violations for the AHLTA EHR.

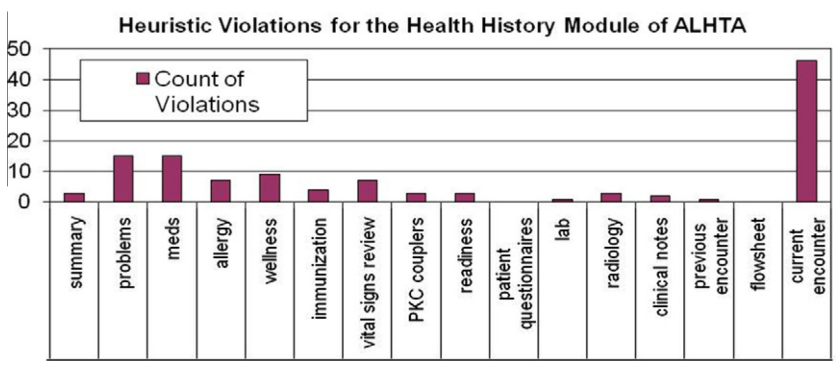

Fig. 10. Violations of usability principles for the Health History module of the AHLTA EHR. It shows that most of the violations are in the current encounter section.

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

In this AHLTA study, three metrics for the efficiency measure of usability (see Table 1), time on task, task steps, and mental effort, were estimated using KLM modeling. These are the expert performance levels following optimal paths of the tasks, and they provide a set of benchmarks for EHR usability. Performance levels by actual users in real clinical environments will be different from these estimated expert performance levels, and they have to be collected through more effortful user testing.

KLM is an excellent method for estimating expert performance levels. However, it does not support a straightforward process for controlling inter-rater reliabilities in an efficient way. To address this issue, we recently adopted the Cogtool [69] method to do our usability evaluation of EHR systems. Cogtool is based on KLM but incorporated the Act-R model of human cognition [70,71].

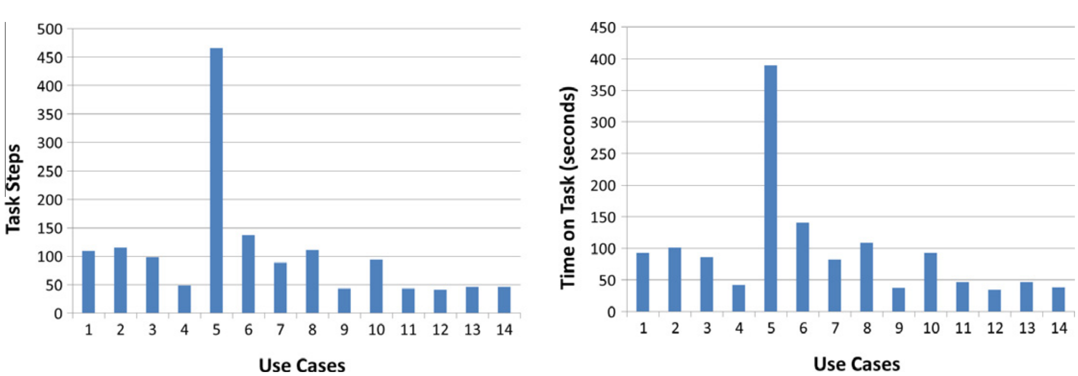

Fig. 11. The left panel shows the number of task steps needed for each of the 14 use case, and the right panel shows the time on task (from [66]).

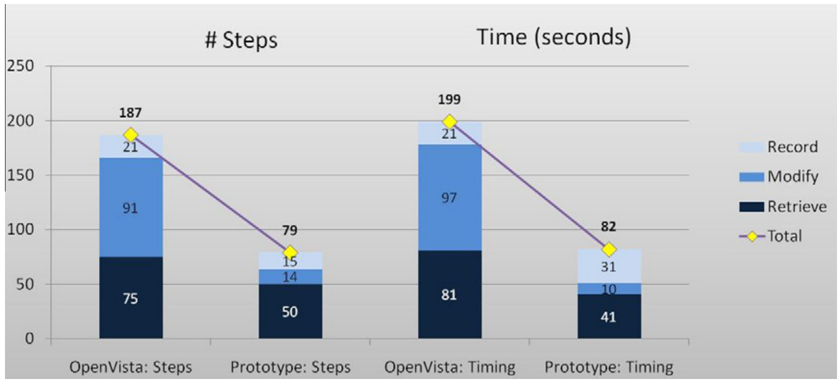

Fig. 12. This figure shows the improvements for both time on task and task steps after the design of an OpenVista module: 187 total steps for the original product to 79 total steps for the new prototype design, and 199 s for the original product to 82 s for the new prototype design (from [73]).

CogTool has increased the accuracy of the KLM and has been reported to be within about 10% of empirical data [72]. In addition to better accuracy, Cogtool does not require two evaluators to achieve significant inter-rater reliability because the 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 [73], 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 of this OpenVista module, identified usability problems, developed new design mockups for this module, and then compared the original product and the new design using KLM and function analysis. Fig. 12 shows the results of KLM task analysis. It shows dramatic improvements on both time on task and task steps: 187 total steps for the original product to 79 total steps for the new design, and 199 s for the original product to 82 s for the new design. The biggest improvement was for the Modify Allergy subtask: the improvement was from 91 to 14 steps and 97 to 10 s. The function analysis shows similar patterns. Overhead functions were reduced from 99 in the original design to 19 in the new design, and domain functions were increased from 28 in the original design to 53 in the new design.

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, just like many other products, are used in real world settings which are typically interruption-laden, unpredictable, and stressful, and involve many other factors such as organizational, social, physical, spatial, temporal, financial, and historical factors. All of these factors can contribute to the representation effect in various ways and they should always be considered in the design and evaluation of EHR usability.

For example, interruption and multitasking are routine in real clinical settings [74–76] and they can cause medical errors [77]. A measure for an EHR's ability to handle interruptions and multitasking should also 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. Our colleagues at the National Center of Cognitive Informatics and Decision Making in Healthcare have been developing a framework and software modeling tool for capturing, analyzing, and predicting workflow across team members in healthcare settings [78]. The match between information flow and workflow is a key principle of usability for user tasks [79]. 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 [80]. In the future, we plan to expand our TURF framework to cover interruptions, workflow, team dynamics, and other socio-technical factors of usability.

4. Discussion and conclusion

In this paper we presented TURF, a unified framework of EHR usability, that has the following properties: (1) for describing, explaining, and predicting usability differences; (2) for defining, evaluating, and measuring usability objectively; (3) for designing built-in good usability; and (4), once fully developed, for 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 of these three dimensions. Most of these measures are evidence-based, repeatable, and objective measures that are established on over fifty years of research in cognitive psychology and human factors. Unlike most approaches to usability, we consider usefulness as an important component of usability, along with the 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 also be measured objectively and systematically. We presented a set of studies we did in the past to demonstrate how EHR usability could be evaluated and measured in a scientific and systematical way.

We also demonstrated how TURF can be used as a method to redesign products to improve their usability. Although we did not discuss how TURF can be used to develop usability guidelines and standards, TURF's theory-based approach, systematical method, and operationalized process are all essential tools for developing EHR usability guidelines, and we are actively moving in this direction in our EHR Usability Lab.

Acknowledgment

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. 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.

References

[1] Blumenthal D, Glaser J. Information technology comes to medicine. N Engl J Med 2007;356.
[2] Davenport K. Navigating American health care: how information technology can foster health care improvement. Center for American Progress; May 2007.
[3] Chaudhry B et al. Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. Ann Intern Med 2006;144:742–52.
[4] Hagland M. Performance improvement special report. Healthcare Informatics; 2007.
[5] Hillestad R et al. Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. Health Aff 2005;24:1103–17.
[6] RAND Corporation. Health information technology: can HIT lower costs and improve quality? 2005.
[7] The leapfrog group. Computer physician order entry: fact sheet; 2007.
[8] Southon G et al. Lessons from a failed information systems initiative: issues for complex organisations. Int J Med Inform 1999;55:33–46.
[9] Koppel R et al. Role of computerized physician order entry systems in facilitating medication errors. J Am Med Assoc 2005;293:1197–293.
[10] Han YY et al. Unexpected increased mortality after implementation of a commercially sold computerized physician order entry system. Pediatrics 2005;116:1506–12.
[11] Zhang J. Human-centered computing in health information systems: Part I – Analysis and design (editorial). J Biomed Inform 2005;38:1–3.
[12] Johnson CM et al. A user-centered framework for redesigning health care interfaces. J Biomed Inform 2005;38:75–87.
[13] Kushniruk AW, Patel VL. Cognitive and usability engineering methods for the evaluation of clinical information systems. J Biomed Inform 2004;37:56–76.
[14] Patel VL, Zhang J. Cognition and patient safety. In: Durso F et al., editors. Handbook of applied cognition. New York: Wiley; 2007. p. 307–31.
[15] Armijo D et al. editors. Electronic health record usability: evaluation and use case framework. Agency for Healthcare Research and Quality; 2009.
[16] Armijo D et al. editors. Electronic health record usability: interface design considerations. Agency for Healthcare Research and Quality; 2009.
[17] Stead W, Lin H, editors. Computational technology for effective health care: immediate steps and strategic directions. Committee on engaging the computer science research community in health care informatics. Washington, DC: National Research Council; 2009.
[18] Zhang J. Human-centered computing in health information systems: Part 2 – evaluation (editorial). J Biomed Inform 2005;40:1–2.
[19] Patel VL et al. Translational cognition for decision support in critical care environments: a review. J Biomed Inform 2008;41:413–31.
[20] Schumacher RM, Lowry SZ. NIST guide to the processes approach for improving the usability of electronic health records. National Institute of Standards and Technology NISTIR 7741.
[21] Schumacher RM, Lowry SZ. Customized common industry format template for electronic health record usability testing. National Institute of Standards and Technology NISTIR 7742.
[22] Aarts J, Peel V. Using a descriptive model of change when implementing large scale clinical information systems to identify priorities for further research. Int J Med Inform 1999;56:43–50.
[23] Berg M. Implementing information systems in health care organizations: myths and challenges. Int J Med Inform 2001;64:143–56.
[24] Goddard BL. Termination of a contract to implement an enterprise electronic medical record system. J Am Med Inform Assoc 2000;7:564–8.
[25] Kaplan B, Shaw NT. People, organizational, and social issues: evaluation as an exemplar. In: Haux RKC, editor. Yearbook of medical informatics. Stuttgart: Shattauer; 2002. p. 71–88.
[26] Lenhard J et al. An analysis of trends, perceptions and use patterns of electronic medical records among US family practice residency programs. Family Med 2000;32:109–14.
[27] Wager KA et al. Life after a disastrous electronic medical record implementation: one clinic's experience. Idea Group Publishing; 2002.
[28] Butler K, Zhang J. Design models for interactive problem-solving: context & ontology, representation & routines. In: Presented at the proceedings of the SIGCHI conference on human factors in computing systems, Atlanta, GA; 2009.
[29] Butler K et al. Ontology models for interaction design. In: Presented at the proceedings of the SIGCHI conference on human factors in computing systems, Atlanta, GA; 2010.
[30] Butler KA et al. Work-centered design: a case study of a mixed-initiative scheduler. In: Presented at the proceedings of the SIGCHI conference on human factors in computing systems, San Jose, California, USA; 2007.
[31] Zhang J, Butler K. UFuRT: a work-centered framework for the design and evaluation of information systems. In: Proceedings of HCI international; 2007.
[32] Zhu M. Formalizing a conceptual framework of work domain knowledge. PhD School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston; 2010.
[33] Landauer T. The trouble with computers: usefulness, usability, and productivity. Cambridge, MA: MIT Press; 1995.
[34] Goransson B et al. The interface is often not the problem. In: Proceedings of CHI+GI 1987; 1987.
[35] Butler K et al. Work-centered design: A case study of a mixed initiative scheduler. In: CHI 2007 Proceedings; 2007. p. 747–56.
[36] Kieras DE. Task analysis and the design of functionality. In: Tucker A, editor. The computer science and engineering handbook. Boca Raton: CRC Inc.; 2004.
[37] Kotovsky K, Simon HA. What makes some problems really hard: explorations in the problem space of difficulty. Cognitive Psychol 1990;22:143–83.
[38] Larkin JH, Simon HA. Why a diagram is (sometimes) worth ten thousand words. Cognitive Sci 1987;11:65–99.
[39] Zhang J, Norman DA. Representations in distributed cognitive tasks. Cognitive Sci 1994;18:87–122.
[40] Zhang J. A representational analysis of relational information displays. Int J Hum–Comput Stud 1996;45:59–74.
[41] Zhang J. The nature of external representations in problem solving. Cognitive Sci 1997;21:179–217.
[42] Zhang J et al. Designing human-centered distributed information systems. IEEE Intell Syst 2002;17:42–7.
[43] Nahm M, Zhang J. Operationalization of the UFuRT methodology for usability analysis in the clinical research data management domain. J Biomed Inform 2009;42(April):327–33.
[44] Kuniavsky M. Observing the user experience: a practitioner's guide to user research. San Francisco: Morgan Kaufmann; 2003.
[45] Chen JW. Developing a process for reducing functional discrepancies. Doctoral dissertation. School of Health Information Sciences, University of Texas Health Sciene Center at Houston; 2008.
[46] Zhang Z et al. Functional analysis of interfaces in US military electronic health record system using UFuRT framework. In: Presented at the AMIA proceedings; 2009.
[47] Hutchins E et al. Direct manipulation interfaces. In: Norman DA, Draper S, editors. User centered system design: new perspectives in human–computer interaction. Hillsdale, NJ: Lawrence Erlbaum; 1986.
[48] Gibson JJ. The theory of affordances. In: Shaw RE, Bransford J, editors. Perceiving, acting, and knowing. Hillsdale, NJ: Lawrence Erlbaum Associates; 1977.
[49] Gibson JJ. The ecological approach to visual perception. Boston: Houghton Mifflin; 1979.
[50] Zhang J, Patel VL. Distributed cognition, representation, and affordance. Cognition Pragmatics 2006;14:333–41.
[51] Gaver WW. Technology affordances. In: Presented at the proceedings of the SIGCHI conference on human factors in computing systems; 1991.
[52] Nielsen J, Molich R. Heuristic evaluation of user interfaces. In: Proceedings of ACM CHI'90; 1990. p. 249–56.
[53] Nielsen J, Mack R, editors. Usability inspection methods. New York: Jon Wiley and Sons; 1994.
[54] Nielsen J. Usability engineering. Boston: AP Professional; 1994.
[55] Nielsen J. Finding usability problems through heuristic evaluation. In: Proceedings of ACM CHI'92; 1992. p. 372–80.
[56] Shneiderman B. Designing the user interface. 3rd ed. Reading, Massachusetts: Addison-Wesley; 1998.
[57] Zhang J et al. Using usability heuristics to evaluate patient safety of medical devices. J Biomed Inform 2003;36:23–30.
[58] Tang Z et al. Applying heuristic evaluation to improving the usability of a telemedicine system. J Telemed Telecare 2006;12:24–34.
[59] Zhang J et al. Evaluating and predicting patient safety in medical device. In: Henriksen K et al, editors. Advances in patient safety: from research to implementation; 2005. p. 323–36.
[60] Graham MJ et al. Heuristic evaluation of infusion pumps: implications for patient safety in intensive care units. Int J Med Inform 2004;73:771–9.
[61] Diaper D, Stanton N, editors. The handbook of task analysis of human–computer interaction. CRC Press; 2003.
[62] Crandall B et al. Working minds: a practitioner's guide to cognitive task analysis. Cambridge, MA: MIT Press; 2006.
[63] Card SK et al. The psychology of human–computer interaction. Hillsdale, NJ: Erlbaum; 1983.
[64] John B, Kieras DE. The GOMS family of user interface analysis techniques: comparison and contrast. ACM Trans Comput–Hum Interact 1996;3:320–51.
[65] Wright PC et al. Analyzing human–computer interaction as distributed cognition: the resources model. Hum–Comput Interact 2000;15.
[66] Saitwal H et al. Assessing performance of an electronic health record (EHR) using cognitive task analysis. Int J Med Inform 2010;79:501–6.
[67] Card SK et al. The psychology of human–computer interaction. Hillsdale, NJ: Lawrence Erlbaum; 1983.

[68] Kieras D. Using the keystroke-level model to estimate execution times; 2001. <eecs.umich.edu/people/kieras/GOMS/KLM.pdf> [15.05.03].

[69] John B. Cogtool; 2009. <cogtool.hcii.cs.cmu.edu/> [16.05.11].

[70] Anderson JR. Rules of the mind. Hillsdale, NJ: Lawrence Erlbaum; 1993.

[71] Act-R Research Group. Act-R; 2011 <act-r.psy.cmu.edu/> [16.05.11].

[72] Luo L, John B. Predicting task execution time on handheld devices using the keystroke-level model. In: Human factors in computing systems: CHI 2005 conference proceedings; 2005. p. 1605–8.

[73] Harrington C et al. Using a unified usability framework to dramatically improve the usability of an EMR module. In: Proceedings of AMIA 2011; 2011.

[74] Franklin A et al. Opportunistic decision making and complexity in emergency care. J Biomed Inform 2011;44:469–76.

[75] Brixey JJ et al. Towards a hybrid method to categorize interruptions and activities in healthcare. Int J Med Inform 2007;76:812–20.

[76] Brixey JJ et al. Interruptions in a level one trauma center: a case study. Int J Med Inform 2008;77:235–41.

[77] Westbrook JI et al. Association of interruptions with an increased risk and severity of medication administration errors. Arch Intern Med 2010;170:683–90.

[78] Butler KA et al. MATH: method and toolsuite for integrating HIT with clinical workflow. University of Washington; 2011.

[79] Butler KA et al. Connecting the design of software to the design of work. Commun ACM 1999;42:38–46.

[80] Chen Y. Documenting transitional information in EMR. In: CHI proceedings; 2010.