Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Biomedical Informatics 37 (2004) 56–76
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Methodological Review

Cognitive and usability engineering methods for the evaluation of clinical information systems

Andre W. Kushniruk[a,*] and Vimla L. Patel[b]

[a] Information Technology Division, Department of Mathematics and Statistics, York University, Toronto, Canada
[b] Laboratory of Decision Making and Cognition, Department of Biomedical Informatics, Columbia University, New York, NY

Received 29 September 2003

Abstract

Increasingly healthcare policy and decision makers are demanding evidence to justify investments in health information systems. This demand requires an adequate evaluation of these systems. A wide variety of approaches and methodologies have been applied in assessing the impact of information systems in health care, ranging from controlled clinical trials to use of questionnaires and interviews with users. In this paper we describe methodological approaches which we have applied and refined for the past 10 years for the evaluation of health information systems. The approaches are strongly rooted in theories and methods from cognitive science and the emerging field of usability engineering. The focus is on assessing human computer interaction and in particular, the usability of computer systems in both laboratory and naturalistic settings. The methods described can be a part of the formative evaluation of systems during their iterative development, and can also complement more traditional assessment methods used in summative system evaluation of completed systems. The paper provides a review of the general area of systems evaluation with the motivation and rationale for methodological approaches underlying usability engineering and cognitive task analysis as applied to health information systems. This is followed by a detailed description of the methods we have applied in a variety of settings in conducting usability testing and usability inspection of systems such as computer-based patient records. Emerging trends in the evaluation of complex information systems are discussed.

1. Introduction

Effective evaluations of health care information systems are necessary in order to ensure that systems adequately meet the requirements and information processing needs of users and health care organizations. A range of approaches have been used in the assessment of information systems with the initial focus on summative evaluation with the objective of assessing how well-completed systems meet a set of pre-defined goals regarding issues of functionality, safety, and impact on outcome measures such as cost of health care and work efficiency [1]. However, in recent years an additional focus has emerged: the development of approaches to evaluation that can be used in the iterative evaluation of systems *during* their development (i.e., formative evaluation), with the objective of improving the design and deployment of information systems as well as ensuring that the process of design of health care systems leads to effective systems [2]. In the general software industry it is increasingly recognized that continued evaluation is needed throughout the system development lifecycle, from early design to summative testing, in order to ensure final products meet expectations of designers, users, and organizations [2,3]. A variety of cognitive approaches to the assessment of health information systems have been developed based on ideas from cognitive and usability engineering. The methods typically borrow from an interdisciplinary perspective and draw from a number of areas including cognitive psychology, computer science, systems engineering, and the field of usability engineering. Usability can be broadly defined as the capacity of a system to allow users to carry out their tasks safely, effectively, efficiently, and enjoyably [4,5]. In recent years the field of usability engineering has emerged to address the need for the application of scientific methods for improving system development

* Corresponding author. Fax: +1-416-736-5392.
E-mail address: andrek@mathstat.yorku.ca (A.W. Kushniruk).

1532-0464/$ - see front matter
doi:10.1016/j.jbi.2004.01.003

and in particular human–computer interaction [6,7]. The profound influence of information systems on cognitive processes of the users is documented in the literature [8]. However, conventional summative and outcome-based evaluations are lacking in their ability to describe these potentially important effects of computer systems on human cognitive processes. In this paper we will focus on methods of evaluation emerging from cognitive and usability engineering that can be applied during a system's development to provide feedback and guidance for its ultimate design [9]. The generic methodological approach has also been used to assess the impact of implemented systems on human reasoning and decision making processes. In addition, variations of the methods described can also be used for assessing the information and processing needs of users of health information systems [8].

Cognitive and usability engineering approaches to the assessment of health information systems involve: (a) characterizing how easily a user can carry out a task using the system, (b) assessing how users attain mastery in using the system, (c) assessing the effects of systems on work practices, and (d) identifying problems users have in interacting with systems. Evaluation in this context involves gathering information about the actual *process* of using a system by representative users performing representative tasks. The results of such evaluation can be used to improve features of the system prior to completion of the design of the system, or alternatively to assess the impact of fully implemented systems. Thus some of the methods described in this paper blur the boundary between formative and summative forms of evaluation. Along these lines, it has been argued that input from the behavioral, cognitive, and social sciences is essential for not just critiquing completed systems, but also to provide essential input into the design process itself [5]. From this perspective, the processes of design and evaluation can be considered to be highly inter-related. In particular, during the iterative development of systems, evaluation during design is essential in order to ensure that a new information technology takes into account the needs and limitations of its end users rather than the preconceptions of the designers with regard to user requirements. The user-centered approach to evaluation focuses on characterization of cognitive skills involved in using a system to carry out representative tasks and description of problems of users with varying levels of expertise and experience, as they learn how to use and master the system [10]. A cognitive approach to evaluation emphasizes the fact that users must gain sufficient knowledge, skill, and familiarity with systems to use them effectively and safely. Much of this approach borrows from cognitive systems engineering which attempts to situate the development of systems in the context of how systems can be designed to facilitate and enhance human decision making and reasoning processes, in particular as they are applied in real work tasks [11,12]. Along these lines, Rasmussen, Pejtersen, and Goodstein [11] argue that:

> "The fast change of pace in available technology makes it difficult to develop appropriate information systems through an incremental updating of existing concepts. Instead design (and redesign) has to be based on a conceptual framework capable of supporting the analysis of work systems, and the prediction of the impact of new information systems, as well as the evaluation of the ultimate user–system interaction. By necessity, such a framework has to integrate modeling concepts from many different disciplines, such as engineering, psychology, and the cognitive, management, information and computer sciences."

In general, the methods represent a shift from a focus on the design of software and system components to gaining a better understanding of the interaction between health care information systems and end users in conducting day-to-day tasks. We begin the next section of the paper with a discussion of some limitations of conventional approaches to evaluation in order to situate our work in the application of complementary assessment methods emerging from cognitive science and usability engineering.

1.1. Need for new evaluation methodologies for health systems

Conventional outcome-based evaluations include quantitative assessments of the economic impact, accuracy, safety, and reliability of completed information systems. In such studies comparisons are usually made between experimental group of subjects using a technology (e.g., a clinical information system) and a control group. Typically, such studies have pre-defined outcome measures which are measured after the system has been deployed in some setting [1,13]. This may provide us with information about aspects of system we are already aware of and interested in (e.g., effects of use of a system on hospital costs). If the outcome is positive or as expected, then our assumptions about the trial-based studies are correct. This provides valuable information on the evaluation. However, if the outcome is negative, then there is often no way of knowing the reason for this outcome, using these methods of data collection. In addition, many effects of health information technology can be considered to be "emergent"—i.e., they are identified or discovered only through the monitoring of the process of system use [8,14]. Thus, ultimately technology not only affects outcomes of a process but may clearly alter the process itself, which must also be assessed. For example, the findings from one of our studies of use of a computer-based patient record system (CPR) in a diabetes clinic has indicated that slightly changing the order of information presented to a physician-user of a CPR can have a significant impact on the nature and order of questions posed to patients by physicians, and ultimately affects the process of diagnostic decision making [8]—an effect of technology which was only discovered from the in-depth observation and analysis of the process of computer use while subjects performed complex tasks.

A variety of approaches to evaluation of health care information systems have appeared in the domain of biomedical informatics [15]. Friedman et al. [13] have provided a summary of their assessment of the evaluation field and give a categorization of evaluation methodologies in biomedical informatics. This work and work of others who have described evaluation of health systems have focused on what Friedman and colleagues term the "objectivist" approach, which can be characterized by a focus on numerical measurement and an attempt to obtain statistical analysis of performance or outcomes that could considered be considered precise, replicable and in this sense "objective." Studies that Friedman et al. cite in this classification include "comparison-based" approaches, which employ experiments and quasi-experiments, where the information resource under study is compared to a control group, a placebo, or a contrasting resource. Controlled clinical trials would fall under this category. A second category which Friedman et al. focus on is referred to as "objectives-based," where the concern of a study is to determine if a system or resource meets its designer's objectives, which is again a type of outcome-based evaluation. The focus of their text is on "objectivist" approaches of the first category, namely the "comparison-based" approach. Friedman et al., in contrast, describe the "subjectivist" method as consisting of approaches such as the following: (1) "quasi-legal," an approach involving a mock trial or other formal adversary proceeding to judge a resource; (2) "art criticism," which is based on formal methods of criticism; (3) "professional review," exemplified by the well-known "site visit" approach to evaluation; and (4) "responsive/illuminative," an approach that seeks to represent the viewpoints of users by beginning with orientating questions and then probing deeper.

The classifications of evaluation of health systems described above include discussion of approaches that go beyond conventional methods in biomedical informatics, although the focus of the majority of reviews to date has been on outcome-based approaches that might fall under the Friedman and Wyatt's "objectivist" label. However, a variety of new approaches to evaluation have emerged that blur the distinction between objectivistic and subjectivist approaches and these approaches are not typically discussed in health systems evaluation taxonomies. One such group of methods that falls under the "emergent" category are methods based on advances in software engineering, and specifically from usability testing and inspection methods [2]. Prior to discussing these approaches, we will describe limitations to conventional approaches to evaluation of health information systems.

One of the most widely used methods for evaluating health information systems continues to be the use of questionnaires, either as the primary method of data collection in system evaluations, or alternatively as one of several types of data collected in multi-method evaluations of information systems. Questionnaire-based survey methods have a number of distinct advantages, including ease of distributing questionnaires to a large number of users (e.g., via the Web) and automated analysis of results. They can also provide quick feedback on how the system is being perceived. In asking subjects about certain categories of information (e.g., subject demographics, age, and how often they use computers) questionnaires are often very useful. However, from the perspective of obtaining information needed for improving system design there are a number of limitations of exclusive use of questionnaires. The usefulness of questionnaires as the primary method for assessing the impact of a system on work tasks may be limited. There are a number of reasons for this. Questionnaires used for assessing the results of a system may not reveal how a technology under study fits into the context of actual system use. They are also limited in providing detailed information about the process of use of a system in performing complex tasks. Questionnaires contain items that are pre-determined by the investigators and consequently are of limited value in identifying new or emergent issues in the use of a system that the investigators have not previously thought of. Furthermore, by asking subjects to rate a system, using a questionnaire typically presented sometime after system's use, the results are subject to problems of the subjects' recall of their experience in using the system (i.e., subjects must reflect back on their use of the system under study). It is often the case that when the actual process of using a computer system is video-recorded in its entirety and compared to questionnaire results (from the same subject), the questionnaire results often do not reflect what the user actually did in practice in using a system, as it was captured on video [16,17]. In many cases in using technology, both experienced and inexperienced users may not be aware of what they actually do, limiting the usefulness of evaluation measures that rely exclusively on retrospective self-reports.

Some of the same problems underlie other commonly used evaluation methods, in particular, retrospective interviews and focus groups, where subjects are asked to reflect on their prior experience with a computer system. Despite these potential limitations, these approaches are still widely used forms of data collection for gathering system requirements upon which systems are developed and also for evaluating the effects of newly introduced health information systems.

The use of interviews or questionnaires alone may be insufficient for revealing how health care workers actually use a system to perform a complex task and may need to be complemented by using other methods. It is important to match the methods to the objectives of the evaluation, keeping in mind the limitations of each evaluation technology. Furthermore, since it is difficult to anticipate all possible errors that can occur in using a complex information system, a careful analysis of how human subjects react and deal with system errors is needed in order to be able to develop rational strategies for dealing with them, thus move closer towards what has been termed a "safety culture" for biomedical informatics [18].

1.2. Cognitive task analysis in biomedical informatics

Cognitive task analysis (CTA) is an emerging approach to the evaluation of medical systems that represents an integration of work from the field of systems engineering and cognitive research in medicine. It is concerned with characterizing the decision making and reasoning skills, and information processing needs of subjects as they perform activities and perform tasks involving the processing of complex information [4,12]. There are a variety of approaches to cognitive task analysis [19]. The approach to cognitive task analysis as described in this paper is closely related to that described by Means and Gott [20], which has been used as the basis for development of intelligent tutoring systems. The first step in such a cognitive task analysis is development of a task hierarchy describing and cataloging the individual work activities or tasks that take place in an organization (with or without the aid of information technology). In health care, these tasks might consist of activities such as a physician entering data into an information system or a nurse accessing on-line guidelines to help in management of a patient. Once tasks have been identified, the method typically involves observation of subjects with varying levels of expertise (e.g., medical students, residents, and physicians) as they perform selected tasks of interest. In our studies this has often involved the subjects carrying out the task while using a computer system. Our approach, described in detail below, typically involves video recording of subjects as they work through selected tasks. An important focus of this approach is to characterize how user variation (e.g., differences in users' educational or technical level) affects the task and the occurrence of potential problems characteristic of different types of subjects studied. CTA has also been applied in the design of systems in order to create a better understanding of human information needs in development of systems [12,21–25].

1.3. Usability engineering in biomedical informatics

Usability engineering has become an important field with applications across a range of software domains. The field can be considered to have emerged from the integration of evaluation methods used in the study of human–computer interaction (HCI) aimed at providing practical feedback into design of computer systems and user interfaces. Usability engineering can be distinguished from traditional systems engineering approaches by its emphasis on obtaining continual input or feedback from end users, or potential end users, throughout the development cycle of information systems [7]. In health care settings, a number of researchers have begun to apply methods adapted from usability engineering towards the design and evaluation of clinical information systems. This has included work in developing portable and low cost methods for analyzing use of health care information systems, along with a focus on developing principled qualitative and quantitative methods for analyzing usability data resulting from such study [17]. Since the mid-1990s a number of groups and laboratories involved in clinical informatics have emerged for testing and designing software applications. For example, Elkin and colleagues describe the use of a usability laboratory for testing a medical vocabulary embedded within the Unified Medical Language System [26]. Kushniruk, Patel, Cimino, and Barrows [9] also describe the use of usability engineering methods for evaluating the design and refinement of both a user interface to a CPR system and the analysis of the system's underlying medical vocabulary. Coble et al. [27] have described the use of usability engineering in the iterative development of clinical workstations. Others have focused on these methods to deal with the 'inspection' of user interfaces [14,28]. Recent work in biomedical informatics has attempted to extend the emerging trend towards usability engineering to include consideration of cognitive issues surrounding design and implementation of clinical information systems, namely cognitive engineering [28,30].

There are a number of specific methods associated with usability engineering and foremost among these is *usability testing*. Usability testing refers to the evaluation of information systems that involves testing of participants (i.e., subjects) who are *representative* of the target user population, as they perform *representative* tasks using an information technology (e.g., physicians using a CPR system to record patient data) in a particular clinical context. During the evaluation, all user–computer interactions are typically recorded (e.g., video recordings made of all computer screens or user activities and actions). Types of evaluations using this approach can vary from formal, controlled laboratory studies of users, to less formal approaches. Principled methods for the analysis of data from such tests, which may consist of video recordings of end users as they interact with systems, can now be used as tools to aid in the analysis. These techniques generally include the collection of "think aloud" reports, involving the recording of users as they verbalize their thoughts while using a computer. Over the past decade, in the technology industry a range of commercial usability laboratories have appeared for conducting usability testing, ranging from elaborate laboratories with simulated work environments and one-way observation mirrors [31,32], to less elaborate facilities and even portable approaches to usability testing, where the recording equipment is actually taken out to field sites [29]. Many of these techniques borrow from work in the application of cognitive science to the study of human–computer interaction [17,33,34]. The practical role of usability engineering in the development lifecycle of clinical information systems has also come under consideration, particularly in the context of use of rapid prototyping methodologies for the design of health care information systems [2,27]. Such methods differ from traditional life cycle models where a system is developed over time using an approach involving fixed stages with limited input from users into redesign. In contrast, rapid prototyping methods typically involve the development of *prototypes* (defined as partially functioning versions of a system) which may be shown to users early in development process in order to assess their usability and functionality. If such assessment indicate that changes are needed, a further cycle of design and testing is initiated. This process continues until the system is deemed to be acceptable to users and shows the desired functionality. This is illustrated in Fig. 1, which shows initial system analysis leading to basic architectural design and implementation of a prototype. The prototype is evaluated (involving usability testing with end users) and if the results of the testing indicate that changes should be made to the design, the changes are made and the system is again tested. Once the testing indicates that the results from the evaluation are adequate and the system has achieved the desired functionality and usability, the final implementation of the system proceeds. In this light, usability testing, involving application of methods to be described in the remainder of this paper, can be considered a central aspect of such design methodologies, where emerging system prototypes must be evaluated and a decision is made to either modify the design or to move to the final engineering of the software product. The use of usability engineering methods in providing a principled approach to assessing the usability of software prototypes during system development has begun to be accepted as an important aspect of the design of emerging health care systems. Such approaches have even been extended to the analysis of clinical systems in the process of selecting from existing software products [35]. In addition, the approach has also been successfully used to assess new innovations in the provision of medical information technologies for home use, such as the IDEATel project, which provides a home-based telemedicine system for diabetic patients [36].

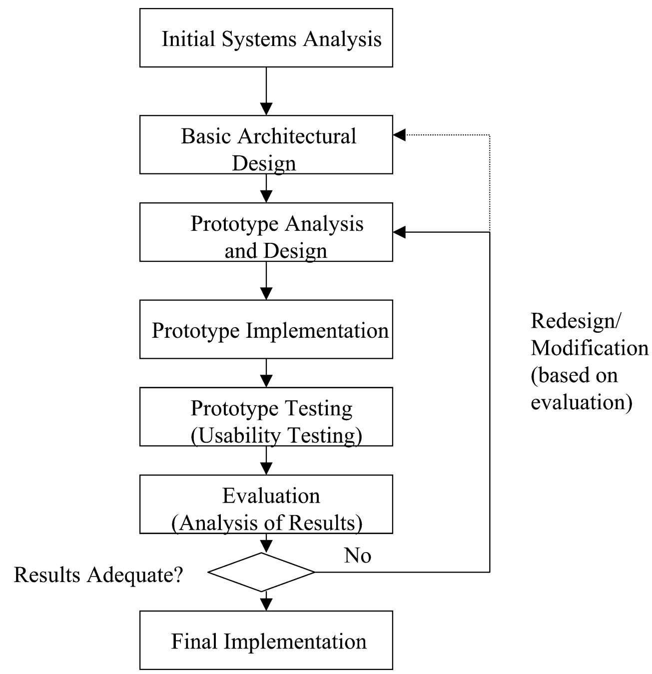

Fig. 1. Systems development based on prototyping and iterative usability testing.

1.4. Usability engineering applied throughout the development cycle of health care information systems

The understanding of how complex information technologies can be successfully integrated into the process of human decision making and practical day-to-day use is critically important in increasing the likelihood of acceptability. Information from usability testing regarding user problems, preferences, suggestions and work practices is applied not only towards the end of system development (to ensure that systems are effective, efficient and sufficiently enjoyable to achieve acceptance), but throughout the development cycle to ensure that the development process leads to effective end products. As shown in Fig. 2, there are a number of points in the system development lifecycle (SDLC) at which usability testing may be useful in the development of new technologies. The typical system development life cycle is characterized by the following phases, which define major activities involved in developing software: (1) project planning, (2) analysis (involving gathering of system requirements), (3) design of the system, (4) implementation (i.e.,programming), and (5) system support/maintenance [3]. There are a number of types of usability tests, based on when in the development life cycle they are applied: (1) Exploratory Tests—conducted early in the system development cycle to test preliminary design concepts using prototypes or storyboards. (2) Testing of prototypes used during requirements gathering. (3) Assessment Tests—conducted early or midway through the development cycle to provide iterative

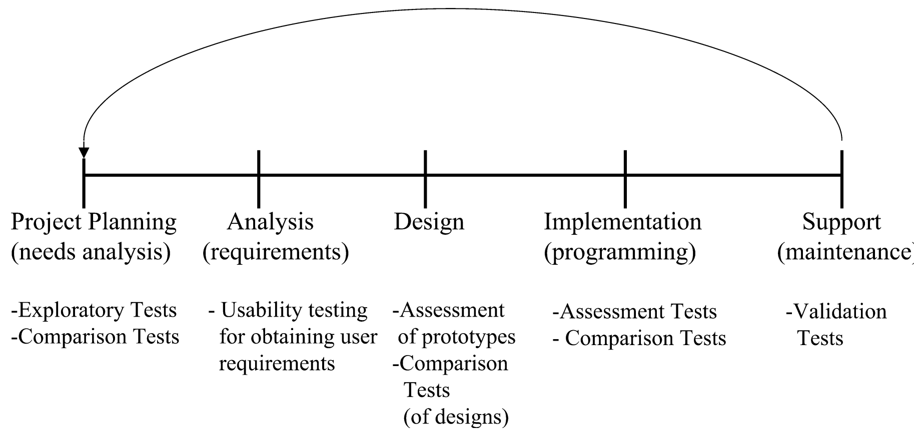

Fig. 2. Usability testing in relation to the phases of the systems development life cycle (SDLC).

feedback into evolving design of prototypes or systems. (4) Validation Tests—conducted to ensure that completed software products are acceptable regarding predefined acceptance measures. (5) Comparison Tests—conducted at any stage to compare design alternatives or possible solutions (e.g., initial screen layouts or design metaphors). From this perspective evaluation in biomedical informatics is seen as being essential throughout the entire life cycle of systems, not just for summative final evaluation.

2. Usability testing approaches to the evaluation of clinical information systems

Given the motivation for applying usability engineering in clinical settings described above, in this section of the paper we describe the phases employed in performing usability evaluations of health care systems and user interfaces extending ideas from usability testing [8,14,17]. Although there may be some variations in the phases, usability evaluation often involves consideration of each of the phases. In order to illustrate the approaches, the description of the phases will be followed by discussion of a case study involving the evaluation of new information technology in health care.

Phase 1. Identification of evaluation objectives. Possible objectives for conducting evaluations can range considerably, including but not limited to the following examples: (1) assessment of system functionality and usability, (2) input into refinement of emerging prototypes, (3) identifying problems in human–computer interaction, (4) evaluating the effects of a system on physician decision making processes, and (5) assessing the impact of a new information technology on clinical practice and workflow. The approach described below can be used to provide practical input into system redesign, e.g., identifying problems with human–computer interaction that need to be rectified.

Phase 2. Sample selection and study design. The second phase involves the identification and selection of a sample of target subjects for the evaluation, resulting in a clearly defined *user profile* which describes the range of skills of target end users of a systems. Subjects should be representative of end users of the system under study. For example if a system is being designed for implementation for use in a particular clinical setting, subjects could consist of personnel who are representative of those who would be expected to actually use the system (e.g., if the system is designed to be used by residents and junior attending staff, it is important to select test subjects that are representative of these groups). Criteria need to be applied for classifying subjects in terms of their prior computer experience. Although there are a number of ways of categorizing users, in our work on usability we have found that considering users along the following dimensions is often useful: (1) expertise of subjects in using computers; (2) the roles of the subjects in the workplace (e.g., physicians, nurses, etc.); and (3) subjects' expertise in the domain of work the information system is targeted for. As evaluation involving cognitive analysis provides a rich source of data, a considerable amount of information may be obtained from a small number of subjects (e.g., 8–10 subjects in a group being studied) particularly if subjects selected are representative of target users of the system being assessed.

In addition to describing the tasks that different types of users will be expected to perform using a system, it is also important to describe as much as possible the most critical skills, knowledge, demographic information and other relevant information about each class of users. Much of our work is an extension of the "expertise approach" [37], which involves comparison of problem solving of subjects of different levels of expertise, to the testing and evaluation of health information systems.

Number of subjects. Prior studies have shown that carefully conducted usability studies involving as few as

8–10 subjects can lead to identification of up to 80% of the surface level usability problems with an information system [7,31]. However, more subjects are required in order to conduct inferential statistics (e.g., 15–20 per study group).

Study design. The study design of our evaluations borrow from approaches in experimental psychology, with a number of options for conducting practical assessments. Study designs may consist of within-group designs where individual subjects may be asked to try out different versions of a prototype system, or one or more subjects may be followed over time as they learn how to use a system. Alternatively, studies may involve between-group designs. Between-group testing might involve for example, comparison of two different systems, with two groups of different health care workers using each system for conducting the same task, such as physicians or nurses looking up patient information in a CPR system. Furthermore, testing may involve use of a CPR system by two groups of subjects of the same medical designation (e.g., attending physicians): one group of subjects who have been identified as being highly computer literate (based on a background questionnaire), and another group of subjects who have had little experience with computer systems. Within-group studies may focus on longitudinal study of how health care workers learn to use and master clinical information systems over time, with testing occurring at specific intervals following initial training in use of a system [8]. Simpler study designs might consist of having a single group (e.g., 10–15 physicians subjects) interacting with a CPR system (with each subject carrying out the same task or set of tasks) in order to assess problems with the design of the user interface.

Phase 3. Selection of representative experimental tasks and contexts. Studies of use of systems can be situated on a continuum ranging from controlled laboratory studies (e.g., studies involving artificial conditions or tasks) to naturalistic studies of doctor–patient–computer interaction involving use of computer systems in real contexts (e.g., tasks involving subjects being asked to interview a patient while entering data into a computerized patient record system). For laboratory-based evaluations involving controlled experimental conditions, we have sometimes used written medical case descriptions, or vignettes, to be used as stimulus material (e.g., subjects may be asked to develop a diagnosis in response to presentation of a hypothetical or real medical case, while using a CPR). The development of medical cases for use in such studies (often consisting of short written descriptions) may require careful design so that the cases are realistic and representative of real-life clinical situations and elicit high quality data about user interactions. For example, cases or scenarios can be drawn or modified from the type of cases commonly used for evaluation in medical education, or presented in medical textbooks or journals such as the *New England Journal of Medicine*. They can also be generated from real health data with the assistance of an appropriate medical expert working with the investigators.

Naturalistic studies of actual doctor–patient interactions sacrifice ability to experimentally control the study for an increase in ecological validity (e.g., collection of data on use of a system in a real clinical setting). In naturalistic studies we generally do not present subjects with artificial written cases, but rather monitor the use of systems (using recording methods to be described below) in real clinical contexts (e.g., a physician using a CPR while interviewing a patient). Regardless of the desired level of experimental control, tasks chosen for study should be representative of real uses of the information technology being evaluated.

Phase 4. Selection of background questionnaires. A background questionnaire may be given either before or after actual testing of a subject's interaction with a system being evaluated. This questionnaire can be used to obtain historical information about the participants that will help the evaluators to understand their behavior and performance during a test. These can include items to assess level of subjects' typical health practice, or prior experience with computer systems [38]. Some usability tests may include examination of educational systems, where the focus is on assessing how much learning takes place during the process of use of a system (e.g., a Web-based educational resource). This may involve the presentation of questionnaires or multiple choice test items before and after testing using a system. For example, in conducting an evaluation of physicians using an educational software system on a specific topic (e.g., advances in breast cancer treatment), subjects were given a set of multiple choice questions to assess their knowledge of that topic both before and after actually recording them interacting with the system, in order to assess the impact of their interactions with systems on their knowledge and learning.

The actual task scenarios to be used during testing also need to be developed during this phase. These may range from simple written descriptions of medical cases, to more elaborate scripts for conducting simulated doctor–patient interviews, where an experimenter "plays" the part of a patient while the subject interviews or interacts with the "patient" while using a technology such as a CPR system [14].

Phase 5. Selection of the evaluation environment. The next step is the selection of the evaluation environment, i.e., where the evaluation will take place. The physical location of the evaluation can vary considerably depending on the degree to which the study is conducted under controlled experimental conditions or in a naturalistic setting. As described in the "Introduction" a number of fixed laboratories have arisen where commercial organizations conduct testing of developing software products in domains ranging from the aerospace industry to brokerage [32]. During the 1990s there was a trend towards the development of large and expensive fixed commercial usability laboratories, which included simulated environments for testing use of systems (e.g., simulated classrooms or work environments). Such laboratories may consist of testing rooms (containing computer systems subjects interact with) and adjoining observation rooms with one-way mirrors, for experimenters to watch subjects. However, it has been shown that many of the methods of usability engineering can be applied in a more cost-effective manner, using inexpensive and portable equipment that can be taken to actual work settings. For example, Cimino et al. [39] have described the development of a portable usability laboratory for use in clinical settings. For the majority of our studies we have adopted such a portable discount usability engineering approach which involves video recording of subjects in the most convenient setting possible, in some cases right in the hospital or clinic under study [9].

Phase 6. Data collectionvideo recording and recording of thought processes. Instructions given to subjects may include asking subjects to perform particular tasks using the computer system (e.g., "Please enter data into the computerized patient record system we are testing while 'thinking aloud' or verbalizing your thoughts"). In addition, instructions might involve asking a physician to conduct a doctor–patient interview while using a system, with full video recording of computer screens and concurrent audio taping of the doctor–patient dialogue (see [14]). In some studies subjects may also be prompted by experimenters at key points in their interaction with a system to comment on aspects of a system or its design. For example, a study might involve comparison of two screen layouts and for each layout the experimenter might ask the user to comment on the screen's layout. In most of our studies the complete interaction of the subject, starting with the initial instructions to completion of all tasks asked of the user is video and audio recorded (using equipment such as that detailed below).

Think-aloud reports. The collection of "think-aloud" reports is one of the most useful techniques emerging from cognitive science. Using this approach, subjects are instructed to "think aloud" (i.e., verbalize their thoughts) as they interact with computer systems (while the computer screens are recorded). There is a principled formal method for analyzing such qualitative data. In our studies of human–computer interaction (HCI) we typically capture the computer screens using video recording (with the computer screen output to a PC-Video converter and then input into a VCR) or screen capture software (e.g., Lotus ScreenCam) for detailed analysis of actions, such as mouse clicks and menu selections. The data collected of users' interactions typically include the video recording of all computer screens along with the corresponding audio recording of subjects' verbalizations as they use the system under study [9].

Equipment typically consists of a PC-Video converter, for converting the output of computer screens to video (to go into the video-in of a VCR). This allows for recording of all computer screens to video as a user interacts with an information system. In addition, we record all subject verbalizations by using a microphone that inputs into the audio-in of the same VCR. Thus on a single videotape we can record all computer screens and user verbalizations made while a subject performs a task using the computer system under study [40].

A schematic diagram illustrating one approach to collecting video and audio recordings of user interactions with a computer system under study is given in Fig. 3. In order to obtain video recordings of computer screens, a commercially available PC-Video Converter is used to convert the VGA computer display output to the video input (i.e., the video-in jack) of a standard VCR. In order to obtain concurrent audio input to the recording of the user–computer interaction we have employed a standard microphone connected to a standard audio mixer (available at most audio stores) or preamplifier, which then outputs into the audio-in jack of the same VCR being used to record computer screens (using a standard RGA cable). This approach allows for recording of user interactions in both the usability laboratory setting, as well as in actual clinical settings since the equipment required is both standard and portable. In a recent paper by Kaufman et al. [36] the use of an inexpensive PC-Video Converter is described for collecting video data portably. In that study, portable recording equipment was taken to the homes of patient subjects, where complete recordings of subjects' interaction with a diabetes management system were made. The result of this phase include a complete video recording of user interaction with a computer system along with the audio track containing the verbalizations of subjects interacting with the system.

As indicated in Fig. 3, video recordings of the actual users themselves (e.g., the faces and gestures of the users as they interact with systems under study) may also be obtained on a separate video recording, although for many of the types of analyses described below, the recordings of computer screens and concurrent audio may be sufficient. If recordings of the actual user are required (e.g., in a study of use of a CPR system where we may want to record how often a physician uses the system as well physically interacts with other objects such as notes or papers on the desk) in addition to the computer screen recording, this can also be conducted in a cost-effective manner (without requiring the use of an expensive usability laboratory) by using a separate video camera and tripod directed at the users, or users, of the system (see Fig. 3). In studies requiring unobtrusive

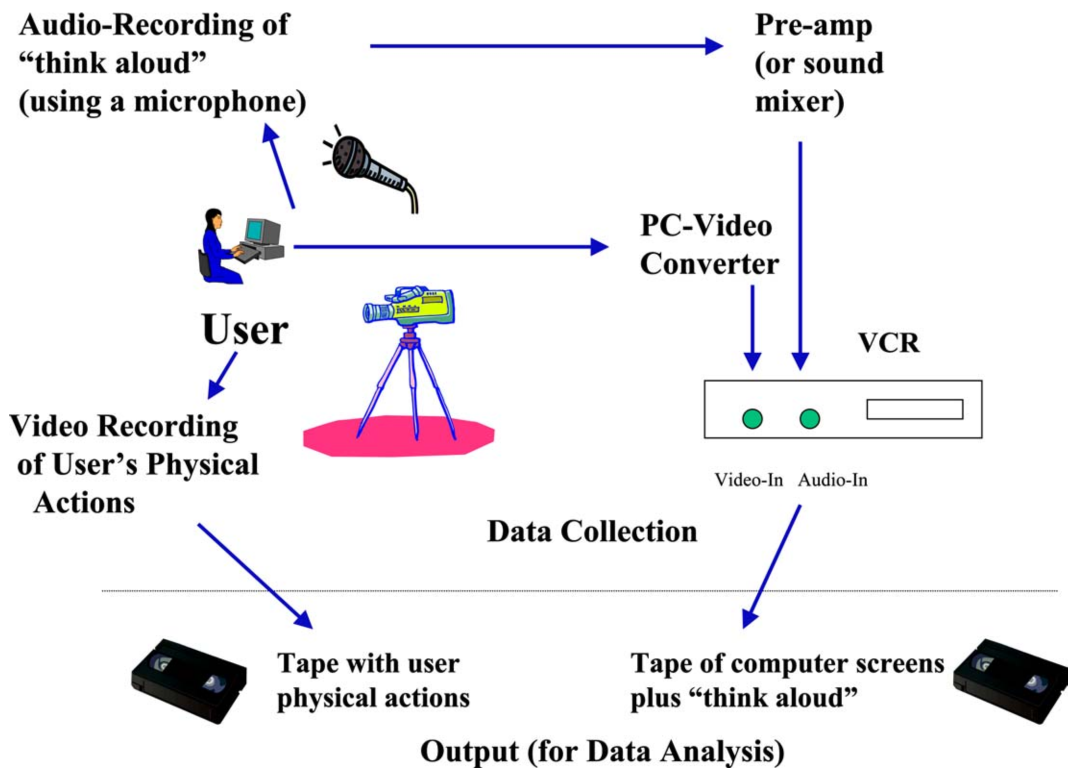

Fig. 3. Video-based usability testing.

observation of user physical interactions with system, rooms having video cameras placed in unobtrusive locations (e.g., ceiling mounted cameras) are ideal. In our work in hospital settings, we have on occasion conducted such recordings in rooms that are typically used for other purposes, e.g., rooms outfitted with ceiling mounted cameras used by medical educators in evaluation of resident and student interviewing skills.

In addition to using standard video recording equipment for recording user interaction with a system, in some studies we have employed a range of software that allows for the recording of screens and audio as movie files directly on the computer being used for testing, removing the need for video cameras and VCRs for recording of the computer screens. For example, the commercially available product ScreenCam allows for directly recording of the computer screens, along with audio input to the same computer via a computer microphone. However, due to storage requirements of such approaches (the resulting recordings are stored as large files that may quickly exceed storage allocation on a standard PC) in many studies we continue to employ standard video recording techniques described above, particularly when collecting data in real clinical settings, where the computer equipment and capabilities may be more limited than in the laboratory.

Phase 7. Analysis of the process data. The output of phase six may consist of video tapes of computer screens (with an audio overlay of the subject "thinking aloud") and/or a tape of the actual user's interactions with the computer system (e.g., their facial expression, movements, gestures, etc.). In many studies, the objective of the evaluation may be to analyze such data to identify problems subjects experience in using a system (e.g., a computerized patient record system or a decision support system). The transformation of data into recommendations involves qualitative and quantitative analyses of the video-based usability data. The advantages of video recordings as a source data include the fact that video tapes of user–computer interactions provide a record of the "whole event." Furthermore, the same video recordings of user interactions can be examined from a number of theoretical perspectives and analyzed using a range of methodological approaches.

There are a variety of approaches to analyzing data on human–computer interaction from video data, ranging from informal review of the resulting taped data, to formalized and precise methods for analyzing the number and type of errors or user problems. The richness of video data requires principled methods for conducting full analysis and coding. The use of computer tools to aid the analysis of video data has greatly facilitated usability testing [17]. Computer programs are now available that interface between VCR and computer in order to facilitate video coding. A software tool we used extensively in our earlier analyses was called CVideo (Envisionology Inc.)—a program which allowed the verbal transcriptions (e.g., of subjects' "thinking aloud" to be annotated on a MacIntosh computer and linked (time-stamped to) the corresponding video sequence (using a cable that connects the Mac to the VCR while reviewing the tape of a usability testing session). In recent years a number of commercially available tools have become available for assisting in the qualitative analysis of audio and video-based data (including MacShapa and other related software tools for conducting qualitative analyses that allow for interfacing and indexing of video tapes). Computer-supported analysis of video data allows researchers to document video frames with textual annotations, notes, and codes on a computer saving time in analysis and allows for automatic indexing and retrieval of video frames and sequences. Such analyses also facilitate inter-rater reliability in coding and allows for coding of user actions and verbalizations.

The procedure for data analysis we employ first involves having the audio portion of the test session ("think aloud" reports) transcribed separately in a word processing file. That file then serves as a computer-based log file for entering annotations and codes, that are linked or "time-stamped" to the corresponding video scenes [9,40]. However, it should be noted that for the types of analyses described below (involving application of coding schemes) computer-supported coding tools are not a requirement for conducting principled analysis of video data. The coding tool will aid in the annotation of the transcripts by linking the computer word processing file containing the transcripts to the actual video tape sequences. However, this can also be accomplished manually—i.e., by watching the video tape and entering into the word processing file containing the audio transcripts, the actual corresponding video counter numbers (as will be illustrated below).

Application of a coding scheme in analyzing video data. Prior to analyzing video data, a coding scheme should be refined for use in identifying specific occurrences of user problems and aspects of cognitive processes from transcripts of the subjects' thinking aloud and interactions with a computer. Coding categories we have applied in a number of studies include the following: *information content* (e.g., whether the information system provides too much information, too little, etc.), *comprehensiveness of graphics and text* (e.g., whether a computer display is understandable to the subject or not), *problems in navigation* (e.g., does the subject have difficulty in finding desired information or computer screen?), and *overall system understandability* (e.g., understandability of icons, required computer operations and system messages). In addition to these categories, which focus on classical aspects of HCI, one can also extend the analyses to allow for the identification of higher level cognitive processes. For example, in some studies we code each occurrence of the generation of a diagnostic hypothesis by a subject, or request for information from a patient, in the case of studies of doctor–patient interaction involving use of a CPR system.

As an illustration, to assess ease of use of computer systems, a coding system can be used as shown in Fig. 4. The scheme shows definitions of coding categories, along with examples of coded statements made by test subjects while interacting with a system that fall under each category (an example of a coded transcript will be provided below in our discussion of a case study). The coding scheme essentially forms a manual for researchers as they watch and annotate the video tapes obtained from experimental sessions. The categories used for coding were developed from examination of categories of interactions from the HCI and cognitive literatures [10,17,41].

In Fig. 5, we show the application of coding categories (from Fig. 4) in analyzing a video log of a user's interaction with a CPR. The procedure for analysis of the subjects' thinking aloud is based on the well-known method of protocol analysis, as described in detail by Ericsson and Simon [33]. Note that the transcript of the subject's thinking aloud report is marked up with annotations from the coding scheme and that the numbers in the log file (containing the transcript) refer to the corresponding section of the video tape (i.e., the video counter number) where they occurred. Also note that codes that indicate user problems are coded as such (with the additional coding tag "PROBLEM").

We have found that up to 80% of user-interface problems with a particular clinical system can be detected with as few as 8–12 transcripts of subjects' interaction with the system under study, which is consistent with the literature emerging from the application of cognitive engineering methods in HCI [7].

Important advances have been made in the development of computer-based tools that aid in the detection and analysis of patterns contained in usability data. In our studies, we have developed a variety of schemes for analyzing video data in a principled manner. These allow coders to identify events of interest, such as user problems, and use of system features (preliminary schemes are typically refined and then verified). Coding schemes can include categories for user–system aspects and problems including categories for human factors issues and cognitive issues [29]. We have developed categories that characterize at a top level the following aspects of human–computer interaction: (1) the *usefulness* of the system being tested in terms of its contents, and (2) the *ease of use* of the system or interface. The former top level category deals with issues such as whether the system being tested provides useful, up-to-date or valuable information to a user, while the latter category characterizes potential problems or issues related to the actual user interface or system design. The coding schemes we have developed are based on and extend categories which have

NAVIGATION
Coded when subject comments on basic navigation, or indicates can't move through program/interface etc. to find or go somewhere (e.g. "How do I get back to the last screen?")

GRAPHICS – Coded for if subject explicitly mentions graphics (e.g. "the use of graphs and tables is nice")

LAYOUT/SCREEN ORGANIZATION
Coded for if the subject comments on the layout or screen organization (e.g. "I find this page very cluttered")

COLOR
Coded for if the subject mentions the use of colors (e.g. "The colors used on this page clash")

RESOLUTION
Coded for if the subject mentions the resolution of information presented (e.g. "The resolution of the information presented is not very good")

MEANING OF LABELS - Coded for if the subject comments on the meaning of labels in the interface itself (e.g. "I don't know what this button means here, the one that says "free download" on it")

UNDERSTANDING OF SYSTEM INSTRUCTIONS/ERROR MESSAGES
Coded for if the subject comments on understanding of instructions or errors
(e.g. "It says "fatal error 404" and I don't know what to do now")

CONSISTENCY OF OPERATIONS
Coded for if the subject comments on the consistency of operatons
(e.g. "how come there are two different ways to exit on the last two screens?")

OVERALL EASE OF USE
Coded for if the subject comments on the overall ease of use (e.g. "I find this system very hard to use")

RESPONSE TIME
Coded for if the subject mentions response time (e.g. "I seem to be waiting a very long time, and still there is no response from the computer")

VISIBLITY OF SYSTEM STATUS
Coded for if the subject comments on visibility of system status (e.g. "I'm not sure what the system is doing now – it seems to be hanging")

Fig. 4. Excerpt from a coding scheme for analyzing human–computer interaction.

been applied in protocol analysis in the study of medical cognition (see [10] for details).

Phase 8. Interpretation of findings. The data collected from usability testing can be compiled and summarized in numerous ways, depending on the goals of the evaluation. The results may summarize any number of aspects of system use including task accuracy, user preference data, time to completion of task, frequency and classes of problems encountered. In addition, qualitative analyses of the effects of the technology on health care professional reasoning and decision making can be conducted. Results of process evaluations may include a summary of types and frequency of problems that occur when subjects interact with a computer system under evaluation. If the system under study is under development, the information provided from the analysis phase should be communicated to system designers. For further investigations, the findings should be interpreted for what they mean, within the context of the theoretical frame work.

Phase 9. Iterative input into design. After implementation of changes to a system, based on the recommendations to the programming team (for studies involving formative evaluations) evaluation may be repeated to determine how the changes now affect the system's usability. In this way, evaluation can be integrated in the process of design and development of information systems, iteratively feeding information back into their continual improvement.

2.1. Usability testing approach: a case study

In this section of the paper we provide a case study based on some of our prior experiences in evaluating health care information systems, applying a usability testing approach. We will step through the example in the sequence of the phases of evaluation described in detail above. In order to provide a continuing example we will consider each phase in the context of evaluating a specific system designed to serve as a computerized patient record

00:00:00 start of session – user starts up the system

"I'm just starting up the system, well what a nice office this is, here we go. I often come here to see Dr. X. Well I have just tried turning the thing on. OK, here is the first screen, but it looks very poorly laid out"
LAYOUT/SCREEN ORGANIZATION – PROBLEM

00:01:15 user goes to help screen

"How do I move to the previous screen?"
NAVIGATION - PROBLEM

"I have diabetes you know and I'm hoping this can help me. So here I go, I'm clicking on this screen about diabetes information"

00:01:23 subject goes to diabetes help screen and clicks on help button

"Now what? It looks like everything has stopped"
LACK OF INDICATION OF SYSTEM STATUS – PROBLEM

Fig. 5. Excerpt of a coded section of a transcript of a user (a patient) interacting with a Web-based information resource.

system. The system described in the example was designed at Columbia Presbyterian Hospital and was known as the Doctor's Outpatient Practice System (DOP) (note—see [9] for a complete description of this case study) that allows clinicians to record patient problems, allergies, and medications—see Fig. 6 for a screen dump of the system's main screen. The larger underlying window is divided into a number of panes, one for entering each of the following: (a) adverse reactions, (b) current medications, (c) results, and (d) active problems. There is also a window (right-hand side of the screen) that is a term-look up function (called the MED-Viewer) that allows the user to enter a term for a new adverse reaction, medication or active problem. In the figure, the user has selected "Problem List" from the main screen and then the "MED-Viewer" appeared (the window in the right hand side of Fig. 6), the user then entered "travel" and three terms were retrieved. The user might then select a term (which means the term

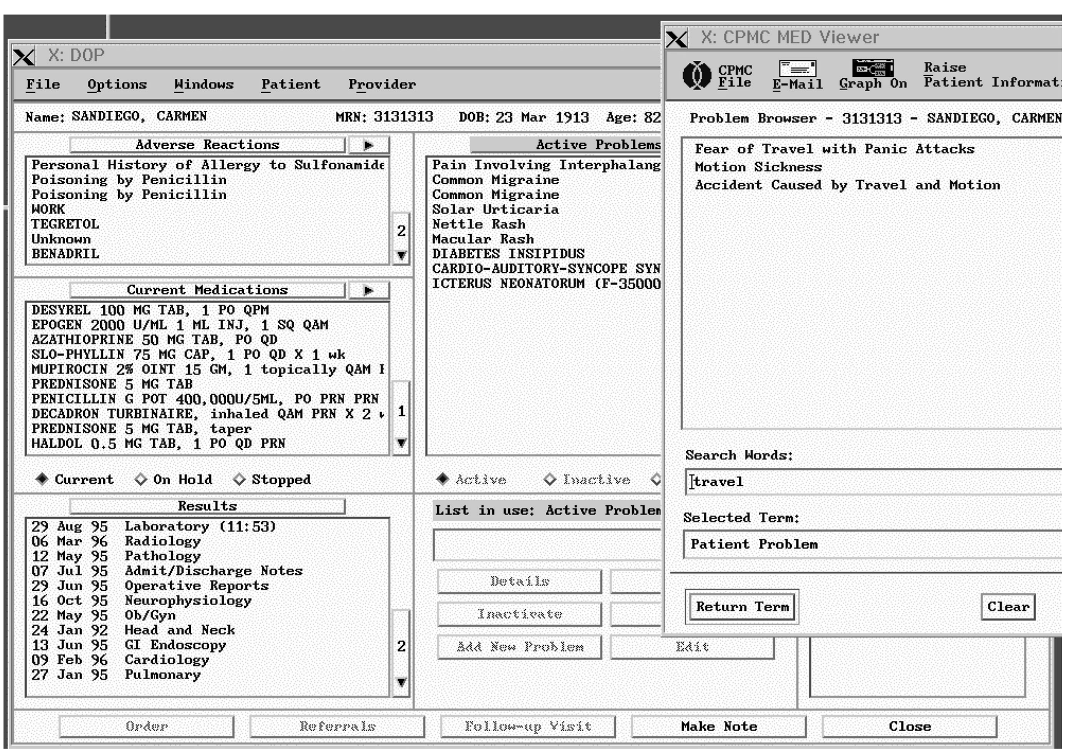

Fig. 6. DOP Main Screen with MED-Viewer.

will be entered into the current list of problems), modify the selected term, or attempt another search.

Phase 1. Identification of evaluation objectives. In delineating the objectives of our evaluation we first described what aspects of the system or user–system interaction we wished to assess. In this particular example, the evaluation objectives included the assessment of the usability of the DOP's overall user interface. In addition, we were also interested in evaluating the adequacy of the search term function (i.e., the MED-VIEWER). The evaluation plan that was created outlined the timeline, resources, and staffing of the evaluation effort. The system to be tested was in midway in its development and a prototype system was available. In this example, the total estimated time for conducting and completing the evaluation was three weeks, and regarding resources it was determined that the available personnel would consist of one full-time staff to run the study and analyze the data.

Phase 2. Sample selection and study design. In phase 2 it was decided that the subjects to be studied should consist of representative target users of the system, i.e., attending and resident physicians. Sixteen subjects were recruited from a local clinic. These subjects were physicians who had never used the system under study but who were planning on moving (i.e., transferring) their paper-based patient records into the DOP system.

Phase 3. Selection of representative experimental tasks and contexts. We decided to employ a portable approach, involving portable video recording equipment. It was also decided that the instructions should consist of asking the subjects to enter information (e.g., patient problems, adverse reactions, medications, etc.) contained in their patient records (which were handwritten) into the DOP system being evaluated. Subjects were then asked to "think aloud" as they interacted with the system to carry out this task.

Phase 4. Selection of background questionnaires. We presented subjects with two questionnaires after they interacted with the system. One questionnaire was designed to assess the subject's level of experience with computer systems, while the second questionnaire contained scales from standard usability questionnaires. The latter questionnaire was designed to determine how often and to what extent they generally used computer systems. (Specific questions included: (1) How often do you use a computer? (2) What type of systems do you use? (3) For what purposes? etc.)

Phase 5. Selection of the evaluation environment. The evaluation involved setting up recording equipment that would allow for continuous recording of computer screens as subjects interacted with the decision support system. To do this, we used a PC-video converter (commercially available for allowing PC displays to be presented on large screen televisions) that served to convert the PC screens to video input into a recording VCR. For the audio recording of users' "thinking aloud" we used a standard microphone which was input into the "audio in" jack of the recording VCR (using a standard audio mixer) as described earlier. The result was a video tape containing the video recording of users' screens with the audio track containing the users' think aloud (see [40] for details). Subjects were asked to enter their patient data into the system while "thinking aloud" as will be described in the next phase.

Phase 6. Data collection—video recording and recording of thought processes. During the actual data collection session, subjects were asked to interact with the system while entering their patient data. Subjects were asked to "think aloud" or verbalize their thoughts as they interacted with the system in carrying out the task. Full video and audio recording was made of their interactions with the system. Sessions took approximately 35 min for each subject to complete.

Phase 7. Analysis of the process data. To illustrate the analysis of the video data, the following is part of a coded log file of a subject's think-aloud protocols and actions while using the system to enter an allergy. As described above, the physician was video recorded as he interacted with the system. The numbers refer to "time stamps" (i.e., the video counter numbers) that link the transcript in the log file (in word processor format) to the actual video footage, while the caps are annotations and codes added by the investigator. The subject (whose thinking aloud is in quotation marks) is attempting to use the MED-Viewer to enter an allergy (see [9] for a complete description):

> "Adverse reactions, does she have any allergies? See if I can get her previous note. She's allergic to shrimp"
>
> 00:56:56 to 00:56:57 SUBJECT ENTERS SEARCH WORD "SHRIMP" INTO MED VIEWER; LIST OF TERMS DISPLAYED
>
> "I don't want any of these. I want to write down that she's allergic to shrimp. Food allergy, that's it, makes me specify in my comment and my entry here will be shrimp"
>
> 00:57:16 to 00:57:17 SUBJECT SELECTS TERM "FOOD ALLERGY" AND TRIES TO ENTER COMMENT
> "Oh, can't enter, try to enter again"
> DATA ENTRY BLOCKED
>
> "Alright, no big deal, it doesn't say which food allergy it is, I would like to see food allergy to shrimp, right up there"
> PARTIAL MATCH

In the excerpt below the same user is having further problems in interacting with the system:

> "It says that patient information gathering is incomplete, please wait a minute and select again. So I wonder if that means I can't select him. OK say that I read the message, now so his information gathering is incomplete, what does that mean?"
> UNDERSTANDING SYSTEM MESSAGES—PROBLEM
>
> 00:58:00 SUBJECT SELECTS ACTION "MAKE A NOTE"
>
> "So lets make a note, and that's what I want to make the note for. OK, blood pressure, weight 199 pounds, everyone in my

practice is heavy. 95 and it won't let me type irregular, no it won't let me type irregular"
DATA ENTRY BLOCKED

"Now this time, you see I have it highlighted, I can create a note, and I hit enter and now it doesn't do anything, so I actually have to click on the button"
00:59:02 LACK OF CONSISTENCY—PROBLEM

By coding the categories of user and system problems (for all subjects), a list of suggestions for improvement were made to system developers, as described under the next phase.

Phase 8. Interpretation of findings. The results from the video analysis of the users' interaction with the system were summarized by developing a list of user problems encountered during the testing sessions, along with tabulation of their frequencies. In this particular study, the most salient problems users encountered were the following, ordered by their frequency: "DATA ENTRY BLOCKED" (39 occurences), "LACK OF CONSISTENCY" (13 occurences), "RESPONSE TIME" (10 occurences), "PROBLEM UNDERSTANDING SYSTEM MESSAGES" (9 occurences). The evaluation of the searches by users for controlled medical terms indicated that a complete match (i.e., the term the user wanted could be obtained using the MED-Viewer) was found in 62% of the lookups (see [9] for further details), partial matches (i.e., the user selected a term but it was not exactly what he/she wanted) occurred 14% of the time. Video analysis also revealed the reasons why users did not always get the term they wanted (e.g., only part of a term was returned, the dosage did not match the medication, too many terms were returned etc.).

Phase 9. Iterative input into design. In our example case study, the results of the evaluation, as described for the previous phase, were presented to the system developers. Based on the results of type and frequency of user errors identified, suggested changes to the system's interface and content were proposed to the developers and the appropriate changes made to the system. For example, examination of the video indicated that the "DATA ENTRY BLOCKED" problem that all subjects encountered occurred when a cursor was blinking (indicating to the subject that data could be entered) but data entry was not actually enabled (i.e., the user was actually blocked from entering data). This problem was easily corrected and in later testing of the DOP interface was largely responsible for a drop in average number of user problems (from 19 per session to 1.9 per session in subsequent usability tests).

3. Usability inspection approaches to the evaluation of clinical information systems

Unlike the usability testing methods described above, usability inspection methods are based on evaluation of information systems by an analyst, hence these types of assessment are referred to as usability inspection methods. Like the usability testing approach, inspection methods are based on the concept of task analysis, where the evaluation is conducted in the context of particular information processing tasks which are initially defined. However, inspections are not based upon empirical testing of end users of a system, but rather a trained analyst (or team of analysts) steps through and simulates the task under study, applying principles of usability inspection. The approach involves the methodical analysis of an interface, where an analyst notes problems or cognitive issues as she steps through or "walks through" the system in order to carry out the task under study. As the methods do not involve testing with end users in real situations, they can be considered to be relatively cost-effective, however, they do presuppose the availability of an analyst trained in the methodology.

Several types of inspection methods have appeared in the literature. *Heuristic evaluation* involves having usability specialists judge the user interface and system functionality as to whether they conform to established principles of usability and good design (i.e., heuristics) [42]. Heuristic evaluation basically involves the estimation of the usability of a system by a user interface expert who systematically examines a system or interface using a set of heuristics. *Guideline reviews* can be considered to be a hybrid between heuristic evaluation and standard software inspection, where the interface or system being evaluated is checked for conformance with a comprehensive set of usability guidelines. *Pluralistic Walkthroughs* involve conducting review meetings where users, developers and analysts step through specific scenarios together and discuss usability issues that they feel might arise [43]. *Consistency Inspections* refer to an evaluation of a system in terms of how consistent it is with other related designs (or other systems belonging to a similar family of products). *Standard Inspections* involve an expert on system standards inspecting the interface with regard to compliance with some specified usability or system standards. The *Cognitive Walkthrough* is a method which applies principles from the study of cognitive psychology to simulate the cognitive processes and user actions needed to carry out specific tasks using a computer system [44,45].

The methods which we have applied and found most useful for adaptation to evaluation of health care information systems are the *heuristic evaluation* and the *cognitive walkthrough*. These two approaches are described below.

Walkthrough methodologies are task or scenario-based, in that they focus on aspects of usability and potential problems with respect to performing a task such as clinical diagnosis. The emphasis is on how easily action sequences can be executed. Heuristic evaluation and consistency inspection focus more on the dialogue elements of the interface and examine the extent to which they conform to or violate established usability principles [7]. These methods allow for a more comprehensive assessment of different facets of the system not specific to any task. Another dimension of difference is the degree of formality involved in applying the method. The cognitive walkthrough is a relatively formal approach in which the process of application is relatively standardized. On the other hand, heuristic evaluation is often considered to be an informal approach because the actual process is specified in less detail [42].

3.1. An example of evaluation of a clinical information system using a cognitive walkthrough approach

In our research, we have employed a variation of the cognitive walkthrough [44]. The cognitive walkthrough methodology was developed from methods and theories of cognitive science, including problem-solving, skill acquisition, Norman's theory of action, and the influential GOMS model for HCI [46]. It involves identifying sequences of actions and subgoals for successfully completing a task and assigning causes to usability problems. The approach has a focus on evaluating how well a task can be completed while using a system, and thus can be considered a form of task analysis. The approach also has a focus on assessing how easy it is to learn how to use a system, especially learning by exploration of a system. In the case of a cognitive walkthrough this involves evaluation of the system in the context of specific user tasks (e.g., a walkthrough of a patient record system for the task of entering patient data). As a preliminary to the walkthrough, assumptions about the user population, context of use and level of end users' computer and domain competency must be considered in the context of the task to be analyzed. As the walkthrough proceeds, the analyst steps through the interface or system sequentially (screen by screen) in order to carry out the task. At each step (e.g., selection of a particular patient record from a list of records on the screen, in the example of a walkthrough involving a CPR), the analyst (or analysts) considers and notes what user actions are needed to carry out the steps, what goals the users would have and what potential problems might be encountered. The objective is to uncover possible errors in design that would interfere with users' ability to learn how to use the system and carry out tasks. Walkthroughs may be conducted at several stages in the development life cycle of a system—during early phases, when an early mockup has been developed, to later stages, when a partially completed system is ready to be tested. It should be noted that the walkthrough may be used to identify potential problems which then might be evaluated using testing methods, described in a previous section of this paper, which involve testing with real end users in real situational contexts. In addition, a cognitive walkthrough may be extended to include consideration of design heuristics (from the heuristic evaluation approach) at each step, leading to a hybrid evaluation approach as described below.

3.2. Phases in conducting a walkthrough

Phase 1. Defining the users of the system. The first step in conducting a walkthrough is the systematic identification of the end user population, i.e., what type of users is the system designed for and what type of background experience might they have that could influence their interaction with the system under study.

Phase 2. Defining the task(s) for the walkthrough. This involves identification of tasks around which the walkthrough will be conducted. For example, an evaluator of a system may want to conduct a walkthrough of all the tasks associated with a new component order entry function of a CPR.

Phase 3. Walking through the actions and critiquing critical information. This phase of the walkthrough process involves the detailed examination of each action undertaken in order to complete the tasks for which the system is being evaluated. The cognitive walkthrough, as described by Polson et al. [44] involves explicit identification by the analysis of the following for each step taken in carrying out a task using the computer system: (1) the goal, or subgoal, that is involved (e.g., the goal of selecting from menu items a desired function), (2) the users' action that needs to be taken in order to achieve the goal, or move closer to it, (3) the behavior of the system (or interface) in response to the users' action, (4) identification of potential problems, given assumptions about the users' background and knowledge (described in Phase 1).

To illustrate this process, the following is an excerpt (from a log) of a walkthrough conducted by an analyst in evaluating a component of the DOP patient record system (described above) designed for entering a problem to the list of a patient's problem (see Fig. 6 for the screen dump of the system and the window used to enter terms, known as the MED-Viewer):

GOAL: Enter a patient's problem into the system
Subgoal 1: Select System Operation—"Add New Problem"
Action 1 —Note that the "Add New Problem" button allows one to switch to this function
Action 2 —Note that the button is available from the main screen
Potential Problem —Other windows may obscure the users view of the main screen and the available options

Subgoal 2: Enter the Problem
Action 1 —Click on the button labeled "Add New Problem"

System Response —A keyword search window (the MED-Viewer) appears for the user to enter the problem (see right-hand side of Fig. 6)
Potential Problem —User may not realize that they must now enter a term in the search terms window

Subgoal 3: Use the Search Term Window (the MED-Viewer) to Select an Appropriate Term
Action 1 —Note that a search term window (the MED-Viewer) appears, for entering the users' term describing the problem
Action 2 —Enter the term (for the problem) in the search words text box—e.g., the user enter the term "Travel"
System Response —The system returns a list of controlled terms that most closely match the users' input (e.g., a list containing the terms: "Fear of Travel with Panic Attacks," "Motion Sickness," "Accident Caused by Travel and Motion")
Action 3 —The user must select from the list returned by the system the term most closely matching their needs
System Response —The system accepts the selected term, the search term window disappears and the list of problems (in a separate window) becomes updated with the new problem
Potential Problem —The user may not realize that he/she must enter a term in the search term window
Potential Problem —The terms returned by the system may not exactly match what the user wanted to enter into the system regarding the nature of the allergy
Potential Problem —None of the terms returned by the system may match what the user wanted to enter into the system regarding the nature of the allergy
Potential Problem —The user may misspell the term they wish to enter in the system
Potential Problem —The user may not realize that the window containing a list of problems is automatically updated

It should be noted that the walkthrough in the above example indicated that the task required three subgoals, and at least six actions. In total, seven potential problems were identified in performing the task. This suggests that relative to the number of steps needed to carry out the task, there is considerable potential for user problems (see next phase for discussion of analysis of results). In order to facilitate the process of analyzing the data that results from the walkthrough, the entire process may be video recorded (with video recording of the screens that the analyst walks through along with the video recording of the analysts comments at each step, using the method for recording described in a previous section of this paper).

Phase 4. Summarization of the walkthrough results. The data obtained from the type of walkthrough described above vary but typically provide a measure of the number of actions taken to carry out a particular task. This can be useful information, especially when comparing two different systems (or possible alternative designs) regarding the complexity of carrying out the same task (e.g., entering patient data). A principled walkthrough is not a replacement for end user testing of a system (under real conditions). However, it does provide considerable insight into problems that might be encountered and can be conducted prior to release of a system or prior to end user testing.

Phase 5. Recommendations to designers. The output of phase 4 can consist of a list of recommended changes for presentation to the design team in the case of a system being developed (i.e., during iterative development). The same type of walkthrough could also be potentially applied to compare two different systems (e.g., two different patient records) for carrying out the same task (for example in deciding between possible systems to purchase). In addition, a well conducted walkthrough can form a prelude to later usability testing involving end users (i.e., the walkthrough may indicate potential problem areas with a system or interface that should be further examined applying usability testing methods).

3.3. Heuristic evaluation

Heuristic evaluation is a usability inspection method in which the system is evaluated on the basis of well-tested design principles such as visibility of system status, user control and freedom, consistency and standards, flexibility, and efficiency of use. This methodology was developed by Jakob Nielsen [7]. There are several stages to carrying out a heuristic evaluation. First, a list of heuristics is given to the analysts who use them in evaluating the system or the interface. The analyst(s) then "steps through" or inspects the user interface or system, and in doing so notes any violations of the heuristics described in the next section of this paper (this could be done in the context of carrying out a specific task in using the system). It is often advisable to have two to four usability experts (analysts) independently assess a system or its interface. Each analyst independently evaluates the user interface and generates a list of heuristic violations which can be compiled into a single list. The results of the evaluation can then be summarized (e.g., number and type of violations of usability heuristics) and presented to the design team along with recommendations for improvement.

Usability heuristics. The following heuristics have been applied in the heuristic evaluation of a wide range of information systems. Initially proposed by Nielsen [47] and inspired by work such as Norman's [48] this widely cited list of heuristics has been applied and modified for use in clinical settings by a number of researchers (e.g., Zhang et al. [28]). Any violations of these heuristics are noted during the evaluation.

Heuristic 1: Visibility of system status. This principle states that the user should be informed as to the state of the system at any given moment. The user should know where he or she is in terms of carrying out a procedure using a system (e.g., if patient data are being uploaded, or if the system is currently processing in response to a user request for data retrieval). Users should know if an operation was successfully completed (e.g., a medication was discontinued in a computerized patient record system) or what additional steps are needed in order to complete a task successfully.

Heuristic 2: Match the system to the real World. This heuristic embodies two concepts. First use of the "natural" language of the user (not jargon or computer system terms, particularly if the users are not computer literate) is recommended, e.g., a button with the label "Obtain free sample program" is preferable to a button labeled with the word "Downloads"). Second, it is best to use real-world conventions or natural mappings. A natural mapping refers to the approximation of the real world in the computer system, for example, having a "rewind" button on a computer to indicate backwards navigation maps to the physical rewind button on the common VCR or cassette recorder.

Heuristic 3: User control and freedom. The main concepts embodied in this heuristic, which essentially states that the user should feel he/she is in control of the system (and not the reverse) are the following: (1) provide clearly marked exits, (2) support undo and redo transactions, and (3) make it difficult to perform irreversible actions. Essentially, there should always be a way for users to back out of current actions (e.g., aborting entry of a medication in a computerized patient record system) and that they should not perceive that they are controlled or irreversibly locked into actions or procedures by the system.

Heuristic 4: Consistency and standards. The user interface and basic system operations should be consistent. Ideally, one module of a system should have the same or similar conventions for exiting or entering, or carrying out basic operations. For example, in one part of a CPR system, a user might have to exit by clicking on the corner of a window, while for another module there may be an exit pop-up box. Due to lack of consistency, users may end up being confused and find learning the basic operations of the interface more difficult than if consistent standards for carrying out such a basic operation were employed. Consistency also applies to the general layout and position on the screen of things like menus, exit buttons and other controls, use of standard terminology (e.g., for files, operations like "new," "open," and "close" used throughout the system).

Heuristic 5: Error prevention. This principle states that designers should design interfaces that prevent error from occurring. This includes simplifying screens, avoiding complex modes that may be confusing to users and testing interfaces to ensure that they are simple and easy to use. Interfaces should be designed specifically to decrease the potential for occurrence of *slips and mistakes*. *A slip is* defined as an unintentional error in using a system, such as making a typo or pressing the wrong key or selecting the wrong menu item by overshooting and a *mistake* is defined as an error occurring through conscious deliberation (e.g., resulting from misunderstanding of how to carry out a basic operation with the system such as the sequence of steps to document a medical problem in a CPR system) [49].

Heuristic 6: Minimize memory load—support recognition rather than recall. Based on the psychological principle that human beings rarely are required to remember all of the features of any object by rote memory, this heuristic states that user interfaces should support recognition rather than recall. Consistent with this principle is the psychological finding that people remember and process information in limited number of "chunks" (typically of five plus or minus two items) [4]. The implications of this for user interface design is that interfaces should support recognition of a meaningful and limited number of items or chunks (e.g., with a menu consisting of five plus or minus two items versus 50 items in order to support information and recognition processes).

Heuristic 7: Flexibility and efficiency of use. Although all users cannot be satisfied regarding their preferences in the design of a user interface, designers should try to make user interfaces as customizable and flexible as possible. For example, some systems allow experienced users to create shortcuts for common operations or allow users to set up their own preferences for display of screens or information that appears upon the starting up a system.

Heuristic 8: Aesthetic and minimalist design. This heuristic states that often the simplest and most minimal design options are often the best for ensuring usability. For example, adding many features and more items to a user interface will not necessarily make it easier to use and at some point detract from the system's overall usability (there are many examples of Web sites that violate this principle with too much information presented and too many flashy "add-ons" and features that may detract from actual use). Other approaches to ensuring minimalist design include the "layering" of information, so that rather than providing too much complex information on one screen, information is layered into a number of simple and easier to understand screens.

Heuristic 9: Help users recognize, diagnose and recover from errors. If the user makes a mistake, the system should provide clear and easy to understand information about how to recover from the error. Error messages should be phrased in clear and meaningful language and not by using cryptic statements such as "fatal error #5." Furthermore, error messages should be precise and constructive as well as being "polite." If users make mistakes there should be obvious ways to correct them.

Heuristic 10: Help and documentation. This heuristic states that help should be available to users when needed. This could consist of a list of topics (like a table of contents) giving help about specific topics, such as printing, formatting, etc. Other forms of help include FAQs (frequently asked questions) as well as context sensitive help facilities, designed to adjust their content and advice based on the type of interaction the user needed help with.

Severity rating scale. In addition to noting usability problems, the following scale [7] can be used for the assessment of the severity of each heuristic violation: **0** = Not a usability problem; **1** = Cosmetic problem only: need not be fixed unless extra time is available on project; **2** = Minor usability problem: fixing this should be given low priority; **3** = Major usability problem: important to fix, so should be given high priority; and **4** = Usability catastrophe: imperative to fix this before product can be released. The purpose of rating each heuristic violation in terms of severity is to help designers in prioritizing aspects of the interface that need fixing. For example, a violation of the heuristics that is rated to be a "usability catastrophe" should be given a high level of consideration by the design team for improvement.

In this section of the paper we provide a brief example of an excerpt from the heuristic evaluation of the DOP user interface we described earlier in this paper. Fig. 6 shows the main screen of the system, for which our excerpt for a heuristic evaluation here will be based around. Examination of the main screen (and associated operations that can be initiated from that screen) reveals several heuristic violations:

1. *Aesthetic and minimalist design*. This heuristic indicates that information should be displayed simply, for example in progressive layers of detail. Visual inspection of the DOP main screen (Fig. 6) indicates that the screen shows a similar level of detail for all categories of information displayed. Further layers of information can be obtained by clicking on categories on the screen, however, the level of layering of information to simplify the complexity of the interface is minimal.
2. *Minimize memory load.* This heuristic indicates that users should not be required to memorize lots of information to carry out tasks. The heuristic evaluation of DOP indicated that the user must know a sequence of non-intuitive steps in order to enter basic information (e.g., to first select a category of information to enter, and to then enter the MED-Viewer, attempt to enter a medical term, etc.)
3. *Consistency and standards.* The heuristic to support consistency is violated in a number of ways in this system. As one example, some of the text and entries displayed on the screen are selectable (by clicking on them), while others are not.
4. *Help and documentation.* These key facilities are not available on the DOP main screen.

Using this approach, the various screens and subcomponents of the DOP interface were systematically examined. These results (along with the results from carrying out usability testing of the same interface with real users, as described above) were used to dramatically improve the usability of the system [9].

4. Advances in usability evaluations in biomedical informatics

In recent years a number of trends have occurred in the refinement and application of the methodological approaches described in this paper. These include advances in the following areas: (a) application and extension of the approaches to the distance analysis of the use of systems over the World Wide Web, (b) automation of some of the key components in the analysis of data, (c) extension to evaluation of mobile applications, and (d) advances in conducting evaluations in naturalistic or simulated environments.

Evaluation methods described in this paper have been extended to the remote distance assessment of a range of health information applications deployed over the World Wide Web [50]. Along these lines there are two types of new developments, the first consisting of use of the Internet to conduct one-on-one evaluations remotely. For example, we have used Web-based video conferencing software (including Microsoft NetMeeting) to conduct usability sessions over the WWW, with the user of the system being studied interacting with the system at a different location (e.g., different city) from the test monitor and evaluators. Specifically, we studied physicians (at a site in Boston) who were interacting with experimental software tools for encoding clinical guidelines, with all screens and subjects' verbalizations recorded remotely at the evaluators' site using NetMeeting to access and record the subjects' screens and interactions [38]. A second line of work in the study of remote use of Web-based systems has focused on automated analysis of use of Web-based health care applications that go beyond mere logging of user interactions to include automated interviewing and automated triggering of questionnaires at points where users interact with components of a Web site of interest (initial work along these lines is described in [38]). Such approaches allow for collection of usability data from a large number of users at distant and varied geographical locations, what has been termed "televaluation." Other advances in this area include the development of comprehension-based simulation models of navigation on the World Wide Web. Results from this line of work are leading to the development of theoretically motivated design methodologies for improving the usability of sites [51].

In recent years, new approaches are emerging to support automated usability analysis. For example, the Web Static Analyzer Tool (WebSAT) is one such tool which has been designed to automatically check the underlying html of a Web page against a number of usability guidelines [52]. The software automatically identifies potential usability problems, which could then be further investigated through usability testing. The automation of the most time-consuming aspects of analyses described in this paper (including heuristic analyses and analysis of data resulting from video recordings) will likely lead to the wider use of the methodologies in the coming years.

The rapid development of mobile and wireless applications in biomedical informatics has led to the need for application of cognitive and usability based testing of these new innovations. In particular, the limited physical screen size and bandwidth limitations of mobile devices (such as PalmPilots and Pocket PCs) have made designing usable applications challenging [53,54]. Nowhere is this more the case than in the development of complex medical applications, including those designed for accessing clinical records via wireless devices [55]. Issues related to cognitive load involved in use of such devices while carrying out other activities and the use of these devices in complex work environments underlie the need for cognitive and usability testing to ensure that the devices are usable and do not introduce error [54].

Another recent trend in application of some of the approaches described in this paper include cognitive analysis of use of health care information systems by health consumers and laypeople. In one line of study we have examined the process of information search and retrieval of health related information. This work has employed full video recording and protocol analysis to determine what features of health information and search engines are most desired by lay population [56]. Subjects (consisting of patients and patients' families) were asked to "think aloud" as they obtained information about their own health care questions using an experimental search engine and three commercial search engines. Adapting the methods described in this paper for video analysis we were able to identify features of the search engines that all subjects liked, as well as undesirable feature that were associated with usability problems. Other related work includes that of Kaufman et al. [36] who also used a multifaceted approach to the analysis of home care information technologies, employing both usability testing and usability inspection methods. Their analyses were designed to assess barriers to optimal use of computer-based system designed for patient use in their homes. By applying methods of usability testing and inspection, similar to those described in this paper, they were able to reveal aspects of the interface that were sub-optimal as well as identifying a range of patient-related factors that constituted barriers to productive use. Other researchers, including Eysenbach and Kohler [57] are applying multi-method approaches to analyzing the information needs of consumers of health information on the World Wide Web.

The approach to evaluation described in this paper is consistent with another trend towards conducting evaluation studies in naturalistic settings. For example, by using low-cost portable usability testing methods, recording of user interactions with systems in settings such as clinics and even home use by patients is clearly feasible [36]. Although there will continue to be a need for further development of laboratory based approaches and facilities for conducting usability evaluations (particularly for testing involving simulation and/or experimental control), increasing use of the methods described in this paper is likely to occur in real-world settings as they become more widely disseminated [58]. The readers are referred to a recent article by Patel and Kaufman [59] which has a comprehensive discussion of usability engineering issues in the context of biomedical informatics and cognition.

5. Conclusion

In this paper we have presented a number of methods which have been developed and refined for evaluating clinical information systems, particularly from the perspective of the end users' interactions with a variety of such systems. Although we do not propose that these methods be used exclusively, we have argued that conventional methods to evaluating health information systems have limitations and that they could benefit by complementing them with newer types of evaluation emerging from cognitive science and usability engineering. Given that some of these methods are time-consuming, the good news is that there are some efforts in automating these analyses, as described above.

A challenge for future work on evaluation of health information systems lies in the integration of data collected from multiple evaluation methods. In particular, an important area that needs to be addressed in future work is the potential relationship and integration of methods focusing on examining aspects of the process of use of systems (e.g., usability testing) with methods involving measurement of outcome variables, and the use of summative evaluation of health information systems. Indeed, as the information technology we seek to evaluate becomes more complex, the methods we use to evaluate those systems will inevitably need to evolve.

Acknowledgments

The authors thank James Cimino who gave us permission to include some examples of usability engineering applications from our previous joint work in the study of the usability of health information systems. The authors thank Elizabeth Borycki for editing the final version of the manuscript.

References

[1] Friedman CP, Wyatt JC. Evaluation methods in medical informatics. New York: Springer; 1997.
[2] Kushniruk AW. Evaluation in the design of information systems: applications of approaches from usability engineering. Comput Biol Med 2002;32(3):141–9.
[3] McConnell S. Rapid development: taming wild software schedules. Redmond, Washington: Microsoft Press; 1996.
[4] Preece J, Rogers Y, Sharp H, Benyon D, Holland S, Carey T. Human–computer interaction. New York: Addison-Wesley Publishing Company; 1994.
[5] Preece J, Rogers Y, Sharp H. Interaction design: beyond human–computer interaction. New York: Wiley; 2002.
[6] Rossen MB, Carroll JM. Usability engineering. New York: Morgan Kaufman Publishers; 2002.
[7] Nielsen J. Usability engineering. New York: Academic Press; 1993.
[8] Patel VL, Kushniruk AW, Yang S, Yale JF. Impact of a computer-based patient record system on data collection, knowledge organization and reasoning. J Am Med Inform Assoc 2000;7(6):569–85.
[9] Kushniruk AW, Patel VL, Cimino JJ, Barrows RA. Cognitive evaluation of the user interface and vocabulary of an outpatient information system. Proc AMIA Symp 1996; 22–6.
[10] Kushniruk AW. Analysis of complex decision making processes in health care: cognitive approaches to health informatics. J Biomed Inform 2001;34(5):365–76.
[11] Rasmussen J, Pejtersen A, Goodstein L. Cognitive systems engineering. New York: Wiley; 1994.
[12] Vicente KJ. Cognitive work analysis: toward safe, productive and healthy computer-based work. New York: Lawrence Erlbaum; 1999.
[13] Friedman CP, Owens DK, Wyatt JC. Evaluation and technology assessment. In: Shortliffe EH, Perreault LE, Wiederhold LM, Fagan, editors. Medical informatics: computer applications in health care and biomedicine. New York: Springer; 2001.
[14] Kushniruk AW, Kaufman DR, Patel VL, Levesque Y, Lottin P. Assessment of a computerized patient record system: a cognitive approach to evaluating an emerging medical technology. MD Comput 1996;13(5):406–15.
[15] Anderson JG, Aydin CE, Jay SJ, editors. Evaluating health care information systems: methods and applications. Thousand Oaks, CA: Sage Publications; 1994.
[16] Mackenzie CF, Jeffries NJ, Hunter WA, Bernhard WN, Xiao Y, Hosrt R. Comparison of self-reporting of deficiencies in airway management with video analysis of actual performance. Hum Factors 1996;38(4):623–35.
[17] Kushniruk AW, Patel VL. Cognitive computer-based video analysis: its application in assessing the usability of medical systems. Medinfo 1995:1566–9.
[18] Fox J. Decision-support systems as safety-critical components: towards a safety culture for medical informatics. Methods Informat Med 1993;32:345–8.
[19] Gordon S, Gill R. Cognitive task analysis. In: Zsambok C, Klein G, editors. Naturalistic decision making. Mahwah, NJ: Erlbaum; 1997.
[20] Means B, Gott S. Cognitive task analysis as a basis for tutor development: articulating abstract knowledge representations. In: Psotka J, Massey L, Mutter S, editors. Intelligent tutoring systems: lessons learned. Hillsdale, NJ: Lawrence Erlbaum; 1988.
[21] Rasmussen J. On information processing and human–machine interaction: an approach to cognitive engineering. Amsterdam: Elsevier; 1986.
[22] Patel VL, Arocha JF, Kaufman DR. A primer on aspects of cognition for medical informatics. J Am Med Inform Assoc 2001;8:324–43.
[23] Patel VL, Kaufman DA, Arocha JF. Emerging paradigms of cognition and medical decision making. J Biomed Inform 2002;35:52–75.
[24] Beuscart-Zepir MC, Bender J, Beuscart R, Menager-Depriester I. Cognitive evaluation: how to assess the usability of information technology in healthcare. Comput Methods Programs Biomed 1997;54(1–1):19–28.
[25] Zhang J, Patel VL, Johnson KA, Smith JW, Malin J. Designing human centered distributed information systems. IEEE Intell Systems 2002:42–7.
[26] Elkin PL, Mohr DN, Tuttle MS, et al. Standardized problem list generation, utilizing the Mayo Cannonical Vocabulary embedded within the Unified Medical Language System. Proc AMIA Symp 1997; 500–4.
[27] Coble JM, Karat J, Orland M, Kahn M. Iterative usability testing: ensuring a usable clinical workstation. Proc AMIA Symp 1997; 744–8.
[28] Zhang J, Johnson TR, Patel VL, Paige DL, Kubose T. Using usability heuristics to evaluate patient safety of medical devices. J Biomed Inform 2003;36(1–2):23–4.
[29] Kushniruk AW, Patel VL. Cognitive evaluation of decision making processes and assessment of information technology in medicine. Int J Med Inf 1998;51:83–90.
[30] Horsky J, Kaufman DR, Oppenheim ML, Patel VL. A framework for analyzing the cognitive complexity of computer-assisted clinical ordering. J Biomed Inform 2003;36(1–2):4–22.
[31] Rubin J. Handbook of usability testing: how to plan, design and conduct effective tests. New York: Wiley; 1994.
[32] Wiklund ME. Usability in practice. New York: Academic Press; 1994.
[33] Ericsson K, Simon H. Protocol analysis: verbal reports as data. Cambridge, MA: Academic Press; 1993.
[34] Carroll JM. Human–computer interaction in the new millennium. New York: Addison-Wesley; 2002.
[35] Beuscart-Zephir MC, Watbled L, Carpentier AM, Degroisse M, Alao O. A rapid usability assessment methodology to support the choice of clinical information systems: a case study. Proc AMIA Symp 2002:46–50.
[36] Kaufman DR, Patel VL, Starren JL, et al. Usability in the real world: assessing medical information technologies in patients' homes. J Biomed Inform 2003;36(1–2):45–60.

[37] Ericsson K, Smith J. Toward a general theory of expertise: prospects and limits. New York: Cambridge University Press; 1991.
[38] Kushniruk AW, Patel C, Patel VL, Cimino JJ. "Televaluation" of information systems: an integrative approach to design and evaluation of Web-based systems. Int J Med Inf 2001;61(1):45–70.
[39] Cimino JJ, Patel VL, Kushniruk AW. Studying the human-computer-terminology interface. J Am Med Inform Assoc 2001;8(2):163–73.
[40] Kushniruk AW, Patel VL, Cimino JJ. Usability testing in medical informatics: cognitive approaches to evaluation of information systems and user interfaces. Proc AMIA Symp 1997:218–22.
[41] Shneiderman B. Designing the user interface. 4th ed. New York: Addison-Wesley; 2003.
[42] Mack RL, Nielsen J. Executive summary. In: Nielsen J, Mack RL, editors. Usability inspection methods. New York: Wiley; 1994. p. 1–24.
[43] Bias RG. The pluralistic usability walkthrough: coordinated empathies. In: Nielsen J, Mack RL, editors. Usability inspection methods. New York: Wiley; 1994. p. 63–76.
[44] Polson P, Lewis C, Rieman J, Wharton C. Cognitive walk-throughs: a method for theory-based evaluation of user interfaces. Int J Man–Machine Stud 1992;6:741–73.
[45] Wharton C, Rieman J, Lewis C, Polson P. The cognitive walk-through method: a practitioner's guide. In: Nielsen J, Mack RL, editors. Usability inspection methods. New York: Wiley; 1994. p. 105–39.
[46] Norman DA. Cognitive engineering. In: Norman DA, Draper SW, editors. User-centered systems design: new perspectives in human–computer interaction. Hillsdale, NJ: Lawrence Erlbaum; 1986. p. 31–61.
[47] Nielsen J, Mack L. Usability inspection methods. New York: Wiley; 1994.
[48] Norman DA. The design of everyday things. New York: Doubleday Books; 1990.
[49] Norman DA. Categorization of action slips. Psychol Rev 1981;88:1–15.
[50] Nielsen J., Designing Web usability. Indianapolis; New Ridersl: 2000.
[51] Kitajima M., Backmon M, Polson, P. A comprehension-based model of Web navigation and its application to Web usability analysis. In: Proc HCI 2000 Conf 2000.
[52] Pearrow M. Web site usability. Rockland, MA: Charles River Media; 2000.
[53] Pearrow M. The wireless web usability handbook. Hingham, Mass: Charles River Media; 2002.
[54] Weiss S. Handheld usability. New York: Wiley; 2002.
[55] Chen E, Mendonca E, McKnight L, Stetson P, Lei J, Cimino J. PalmCIS: A wireless handheld application for satisfying clinician information needs. J Am Med Inform Assoc 2004; 11(1):19–28.
[56] Kushniruk AW, Kan MY, McKeown K, et al. Usability evaluation of an experimental text summarization system and three search engines: implications for the reengineering of health care interfaces. In: Proc AMIA Symp 2002; 420–4.
[57] Eysenbach G, Kohler C. How do consumers search for and appraise health information on the World Wide Web? Qualitative study using focus groups, usability tests, and in-depth interviews. Br Med J 2002;324(7337):573–7.
[58] Wichansky A. Usability testing in 2000 and beyond. Ergonomics 2000;43(7):998–1006.
[59] Patel VL, Kaufman DR. Cognitive science and biomedical informatics. In: Shortliffe EH, Cimino JJ, editors. Biomedical informatics: computer applications in health care and biomedicine. 3rd ed. New York: Springer; 2004.