Participatory Design and an Eligibility Screening Tool

John H. Gennari, Ph.D. and Madhu Reddy, M.S.

Informatics in Biology and Medicine,
Information and Computer Science Department,
University of California, Irvine

Abstract

For most medical informatics software products, insufficient effort is spent on the design phase of production. However, poor design often leads to systems that are either not well accepted, or far less effective than they could be. In this paper, we describe the ideas of participatory design and discuss why these ideas are especially applicable to medical informatics systems. In particular, we present a case study in the area of clinical trial protocol management. We designed and developed a tool aimed at increasing accrual to clinical trial protocols at an oncology center. However, the design evolved over time, and features of this design were only discovered through iterative development and interaction with the users within the context of the workplace.

Software for Eligibility Screening

Eligibility determination and enrollment into clinical trial protocols is a chronic problem for many clinical research centers. Enrollment rates may be low for a number of reasons. First, physicians may not know about relevant clinical trial protocols—the National Cancer Institute's CancerNet system lists more than 1,800 open clinical trial protocols in oncology alone (see http://cancernet.nci.nih.gov/). Second, there may be significant organizational barriers, such as the administrative and paperwork cost of treating patients on protocols. Finally, the eligibility criteria of protocols may be overly restrictive and difficult to assess. Ultimately, low enrollment rates result in longer delays in the transfer of research results into improved standards of health care.

Especially because clinical trial protocols and their accompanying eligibility requirements are typically stored and disseminated on paper, the task of eligibility determination is a clear target for medical informatics and the development of a decision support system. If the eligibility requirements could be captured in a computational manner, a tool could be developed to automatically match patient characteristics and history information against available clinical trial protocols. Compared to manually searching through the paper documents associated with each clinical trial protocol, such a tool could decrease the amount of work needed to enroll patients, and potentially increase the number of patients enrolled in protocols.

At the Chao Family Comprehensive Cancer Center of the University of California, Irvine (UCI) Medical Center, we set about to try and test this hypothesis. This cancer center has about 100 open oncology clinical trials, and is actively trying to both increase enrollment into these trials and to increase the number of protocols available at the center. Currently, enrollment depends on the ability of staff and physicians to be familiar with protocol requirements. Patient characteristics are matched by hand, comparing the patient's chart against the protocol eligibility requirements—both of which are paper documents. The cancer center does not have plans to install an electronic medical record system in the near future.

Our goals are to increase patient enrollment and to decrease the work associated with enrollment. To reach these goals, our first step is to design and install a software system: a decision support tool for eligibility screening. In this paper, we report on this early process of software design, development and installation, and describe a methodology known as *participatory design*. We claim that good design is particularly important for medical informatics applications. In addition, we claim that the use of the participatory design methodology increases the likelihood that a system will be accepted and used in the medical work setting.

What is Participatory Design?

The design of a software application is, at best, a complicated and costly activity involving diverse stakeholders. In general, software engineers are beginning to look more carefully at the design stage, using a variety of different design methods.[1, 2] Before describing participatory design, we discuss design in general, and then address why design issues are particularly crucial to medical informatics. We believe that if medical informatics systems developers attend to these issues, they will be able to build medical informatics applications that are more likely to meet user needs and therefore be successfully adopted into users' work activities.

Design

Notions of design can be drawn from fields ranging from architecture to software engineering. Unfortu-

nately, this range of use makes the word notoriously difficult to define. Herbert Simon, a Nobel Laureate whose work crossed several domains including computer science, defined design as an activity concerned "with how things ought to be, with devising artifacts to attain goals."[3] Thus, from a software development perspective, the essence of software design is to build applications (or devise artifacts) that meet the requirements (or goals) of its users.

Even though defining design is difficult, it does have certain characteristics that can guide us in thinking about what constitutes design. Winograd describes some of these characteristics:[2]

- Design deals with human concerns.
- Design has social consequences.
- Design is a social activity.
- Design contains creativity.
- Design is communication.

A common thread through these different characteristics is the view that design is a distinctly *human* activity, involving communication and creative thought among a group of participants. Designers must take imperfect information from users to build an approximate model of the application. Users' information is usually imperfect because it is difficult to communicate and articulate all aspects of the problem the application should solve. This communication difficulty can lead to designs that are faulty, i.e., designs that do not actually meet the users' needs. Systems built from such faulty designs inevitably fail. Thus, good software design is a difficult, yet essential step in the process of building successful, high-quality applications.

Design and Medical Informatics

Medical informatics system developers must confront the same set of design challenges that face all software engineers. However, the medical setting provides a set of additional constraints that highlight the importance of well-designed medical informatics applications.

First and foremost, many medical informatics applications deal with life-critical information. These applications use and manipulate information that can dramatically affect the health of a patient. If a system is poorly designed, it could provide misleading or even incorrect life-critical information to its users. It is crucial that medical informatics systems be well-designed so that the chances of misuse are minimized.

Second, users such as physicians and nurses are typically in time-pressured work settings. Such users do not have the time to learn to use applications that are not carefully designed. A well-designed application matches the users' needs closely, and is therefore easy and less time-consuming to use. Thus, good design is important for medical informatics systems, so that these systems are easy to use.

Finally, medical informatics applications are implemented in complex organizational environments, such as hospitals. Successful medical informatics systems are rarely "stand-alone"—instead, they must integrate with a range of other applications, systems, and organizational bodies that are part of the medical environment. A well-designed system insures that the software functions appropriately under a wide variety of organizational conditions.

Participatory Design

In a traditional software development process, software designers solicit requirements from users and build a requirements specification document. This document is then handed to the developers who actually build the system. Finally, the system is given to the users for either further testing or actual use. Unfortunately, it is well-established that this traditional approach has significant problems with: (1) requirements analysis and (2) design iteration. Although software engineers recognize that the traditional approach can lead to these problems, there are no standard, well-accepted alternative design methodologies.

In the U.S., participatory design (PD) is an engineering means for building better, more user-acceptable systems.[4] (In Scandinavia, where PD originated, the focus was primarily on bringing workers into the design process of software applications that affected their worklife.[5]) PD provides a set of techniques to deal with the problems of requirement analysis and design iteration. First, a major difficulty for design is gathering useful requirements from users and incorporating them into the system design. PD attempts to remove the barriers between system developers and users by more actively bringing the users into the design stages.[6] Second, traditional design methods have problems with the high cost of design iteration. In traditional design processes, iterating and making changes can become prohibitively expensive because systems are usually near completion before they are tested by users. On the other hand, PD techniques such as low-tech prototyping, storyboard prototyping, and mockups allow for user feedback on the system design without the high costs associated with fully implemented applications.[7] Muller et al. recommend different PD techniques depending on the number of the participants involved in the process and the type of interaction between the developers and users.[7] In short, participatory design addresses concerns about the lack of user involvement in traditional system design.

In medical informatics, PD has been used to investigate better ways for developing health care systems. Researchers in Sweden have examined using PD

methods in design meetings[8] and gathering requirements for building an organizational learning system in health care.[9] For our work designing and building a protocol screening tool, we used a technique known as *contextual inquiry*.[10, 11] Contextual inquiry uses workplace interviews as well as direct observations to inform software design. The strength of contextual inquiry is that researchers collect concrete descriptions of the work practices by observing interaction with, and directly inquiring about, work artifacts and work flow. Contextual inquiry "focuses people on articulating their work experience as they work—helping people to be more concrete".[10] Contextual inquiry seemed appropriate for our study for two reasons. First, the user group was small enough (1-5 users) for us to observe and understand their work tasks. Second, given the time constraints of our users, the best way to involve those users in the design process was to have developers visit and learn about the work tasks at the work site.

Participatory design is not a cure-all for the challenges of medical software design. However, by bringing attention to users' needs and the workflow setting for the technology, and by involving users in the development process, we claim that participatory design can increase the chances that the resulting system will be successfully adopted. In the next section, we describe how participatory design and contextual inquiry affected our design of a protocol eligibility screening tool.

Designing a Protocol Screening Tool

The general task of eligibility determination for protocol-based care has been addressed by a number of medical informatics efforts.[12-14] The primary way in which our work differs from these is that we are focused on protocol *screening* rather than definitive eligibility determination. That is, our tool provides advice about which protocols the patient *might* be eligible for enrollment, and does not attempt to assess all of the eligibility requirements of a protocol. Complete eligibility determination involves some uncertainty, and is therefore a decision that should made by a physician.[12] In contrast, our protocol screening tool is more like a sophisticated filtering system, and can be used by other health practitioners (e.g., oncology nurses) as a decision support tool for triaging patients.

This initial design pre-dated any interaction with the UCI Cancer Center. However, we began exploring how our prototype tool could be adapted to the specific needs of the cancer center during a series of group meetings between the users from the Clinical Research Office (CRO) and the main developer in the Spring and Summer of 1999. The staff of the CRO are responsible for enrollment and all administrative paperwork associated with the clinical trials open at the cancer center. One concern of the CRO is to quickly respond to telephone requests for information about available clinical trial protocols. Obviously, complete eligibility determination cannot occur over the phone, and thus, it seemed appropriate to use a protocol screening decision support tool for this task. In addition, the research office was looking for ways to reduce reliance on imperfect "cheat sheets" and other paper-based methods for understanding the eligibility criteria for these clinical trials. Therefore, a computer-based decision support tool seemed attractive in the face of increasing numbers of clinical trials available at the cancer center.

An Initial Screening Tool

For participatory design to succeed, there must be the capability to rapidly evolve some type of mockup system from which designers can solicit user feedback. In our case, we used the capability to rapidly build prototype systems. Our protocol screening tools were built as plug-ins within the Protégé-2000 architecture.[15] In general, Protégé is an environment for building knowledge-based systems—for our system, we stored all protocol information in a knowledge base, and then built a series of different views and user interfaces as plug-ins into that knowledge base.

The first prototype we presented to the clinical research office was not tailored to any hospital or clinic—it simply used a nation-wide list of open clinical trials as provided by the National Cancer Institute. Thus, the first participatory design task was to adapt this prototype to the UCI Cancer Center and its set of available clinical trials. We found that it was useful to carry out part of this design work while physically at the cancer center site. As part of the design, we queried specialists about the clinical trials available at the cancer center, for particular oncology domains. For example, during one visit, we interviewed a gastro-intestinal (GI) oncologist to ask what the appropriate set of questions would be to discriminate among the available GI clinical trial protocols. We could then rapidly translate these discriminating questions into a set of pull-down menu items for incorporation into our tool. This sort of user-driven iteration with a prototype is exactly what is needed for successful participatory design.

Figure 1 shows a version of our protocol screening tool that resulted from our initial set of interviews. The first way to screen out inappropriate protocols is to ask for the primary cancer site—in Figure 1, breast cancer has been selected. Once a cancer site has been chosen, a set of specific questions about the patient appears on the left side. Our assumption is that a small amount of patient data can distinguish among the candidate trials available for a particular cancer

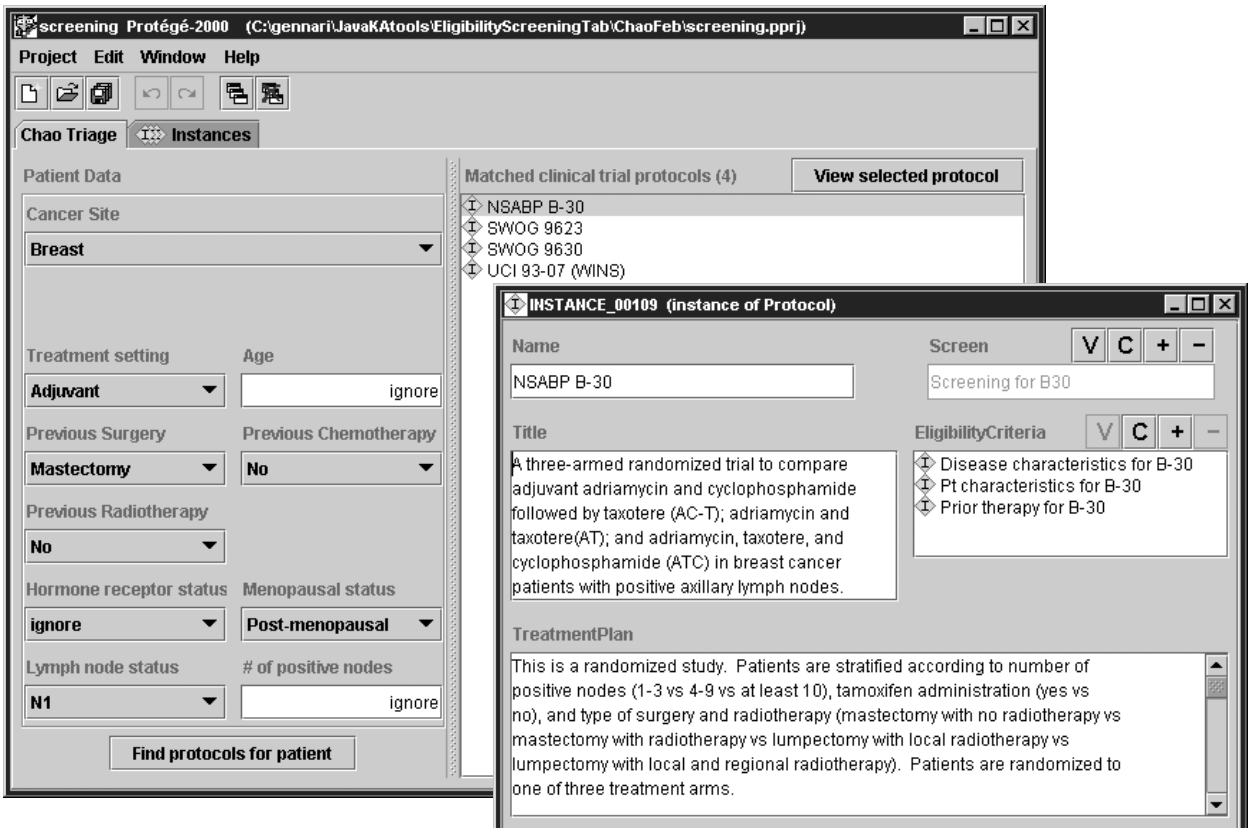

Figure 1. Our tool for clinical trial protocol screening. In the main window, the user enters minimal patient information on the left, and the tool returns a list of matching protocols on the right. The smaller window shows some information about protocol NSABP #B-30; this protocol may be further explored to view details in additional windows, such as complete eligibility requirements.

site. Figure 1 shows the questions for breast cancer, and their answers for a particular patient. After providing this patient information, the user may ask the system to "Find Protocols". In the case shown in Figure 1, four possible protocols are returned, and one (B-30) has been selected to view additional details.

PD and Requirements Shift

A classic problem with software development is that users' requirements may shift over time. These shifts may occur because (1) there are changes in the workplace organization (e.g., new roles or new work tasks), or because (2) users have changed views and ideas about the use of the software, or because (3) the hardware and equipment capabilities in the workplace have changed. Due to its emphasis on user interaction and rapid prototyping, participatory design is able to gracefully adapt to these requirements shifts. More specifically, for our project, we were able to track shifts by observing the use of the tool in the actual work setting.

In our project, the requirements shifted both because the users changed ideas about how the tool would be used, and because we introduced new sets of users to the tool. After our primary user, a clinical research nurse in the CRO, began to be familiar with the tool, she felt that it would be of greater use in conjunction with the paper patient charts, rather than only with telephone inquiries. Thus, she began to use the tool on a regular, weekly basis for breast cancer patients, rather than in response to phone calls. She also suggested installing the system on the clinic floor, thereby expanding our user base to other oncology nurses and attending physicians.

These are significant changes to the system requirements, and for a traditional software development process, they might be difficult to incorporate into the design process. In contrast, participatory design expects this type of iteration and design change in response to user input. The change from a telephone setting to a paper chart setting added a crucial requirement for the software tool—printing. The set of possible protocols must be available as a printed report that can be added to the patient's chart for review by a physician. Furthermore, expanding the set of users to oncology nurses and physicians meant that

the printing capability must be quick and easy—using cut-n-paste from screenshots (such as Figure 1) to produce printed reports was certainly not acceptable.

We were able to track these shifts only by actual visits to the workplace, i.e., only by using contextual inquiry to discover how the tool was used. The shift from a telephone setting to a paper chart setting was not immediately articulated by the primary user—it was simply assumed that the tool would be appropriate for use with paper charts. This example illustrates how traditional requirements analysis might fail, since this approach would miss information that is left implicit by users. In contrast, contextual inquiry emphasizes that queries about requirements occur in the context of the work setting—thereby allowing developers to directly observe how their systems (or prototypes) are actually used in the workplace.

Conclusions

We have introduced the ideas of participatory design as a means by which medical informatics applications can be successfully built and deployed. As an example, we used participatory design during the development of a decision support tool for clinical trial protocol eligibility screening. During the design and development of this tool, we encountered problems that are typical of medical software development: requirements shift and the challenge of satisfying the complexities of the medical workplace setting. By using iterative prototypes, and by directly interacting with the users at their workplace, we believe that participatory design allowed us to manage these challenges gracefully. For our specific project, we hope that this design methodology will allow us to build a better tool to improve enrollment into clinical trials, and thereby improve the overall process of clinical trials research.

Acknowledgements

This work was partially funded by the Chao Family Comprehensive Cancer Center. We wish to acknowledge the cooperation and work of Donna Jackson, and the assistance of Christine McLaren.

References

1. Moran TP, Carroll JM, eds. *Design Rationale: Concepts, Techniques, and Use*. Mahwah, NJ: Lawrence Erlbaum Associates; 1996.
2. Winograd T, ed. *Bringing Design to Software*. New York: Addison-Wesley; 1996.
3. Simon HA. *The Sciences of the Artificial*. 2nd ed. Cambridge, MA: The MIT Press; 1961.
4. Kensing F, Blomberg J. Participatory Design: Issues and Concerns. *Computer Supported Cooperative Work* 1998;**7**:167-85.
5. Bjerknes G, Ehn P, Kyng M, eds. *Computers and Democracy: A Scandinavian Challenge*. Brookfield, USA: Avebury; 1987.
6. Chin G, Rosson MB, Carroll JM. Participatory Analysis: Shared Development of Requirements from Scenarios. In: *Conference on Human Factors in Computer Systems (CHI 97)*; 1997; Atlanta, Ga., pp. 162-69.
7. Muller MJ, Wildeman DM, White EA. Taxonomy of PD practices: A Brief Practitioner's Guide. *Communications of the ACM* 1993;**36**(4):26-28.
8. Sjoberg C, Timpka T. Participatory Design of Information Systems in Health Care. *Journal of the American Medical Informatics Association* 1998;**5**(2):177-83.
9. Timpka T, Sjoberg C, Hallberg N, Eriksson H, Lindblom P, Hedblom P, et al. Participatory Design of Computer-Supported Organizational Learning in health care: Methods and experiences. In: *19th Annual Symposium on Computer Applications in Medical Care*; 1995; New Orleans, LA, pp. 800-04.
10. Holtzblatt K, Jones S. Contextual Design: Principles and Practices. In: Schuler D, Namioka A, eds., *Participatory Design: Principles and Practices*. New York: Lawrence Earlbaum; 1993.
11. Coble JM, Maffitt JS, Orland MJ, Kahn MG. Contextual inquiry: Discovering physicians' true needs. In: *19th Annual Symposium on Computer Applications in Medical Care*; 1995; New Orleans, LA, pp. 469-73.
12. Tu SW, Kemper CA, Lane NM, Carlson RW, Musen MA. A methodology for determining patients' eligibility for clinical trials. *Methods of Information in Medicine* 1993;**32**(4):317-25.
13. Brietman P, Weisburd M, Overhage J, Sledge G, Tierney W. Pilot study of a point-of-use decision support tool for cancer clinical trials eligibility. *Journal of the American Medical Informatics Association* 1999;**6**(6):466-77.
14. Ohno-Machado L, Wang SJ, Mar P, Boxwala AA. Decision support for clinical trial eligibility determination in breast cancer. In: *Proceedings of the AMIA Annual Symposium*; 1999; Washington DC, pp. 340-44.
15. Musen M, Fergerson R, Grosso W, Noy N, Crubezy M, Gennari J. Component-Based Support for Building Knowledge-Acquisition Systems. In: *Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress (WCC 2000)*; 2000; Beijing, China.

(Also, http://www.smi.stanford.edu/projects/protege/)