Supporting Collaborative Clinical Trial Protocol Writing through an Annotation Design

Chunhua Weng

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2005

Program Authorized to Offer Degree:

Department of Medical Education and Biomedical Informatics

University of Washington

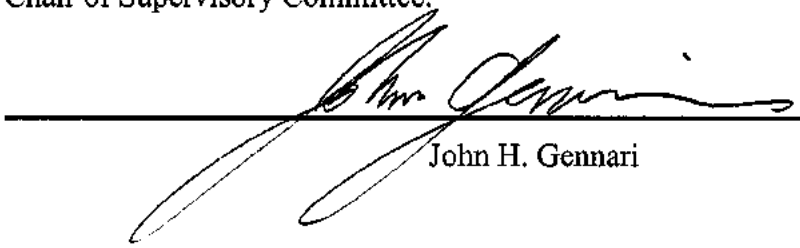
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Chunhua Weng

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

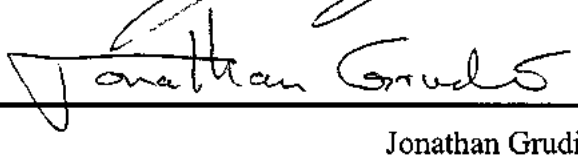
Chair of Supervisory Committee:



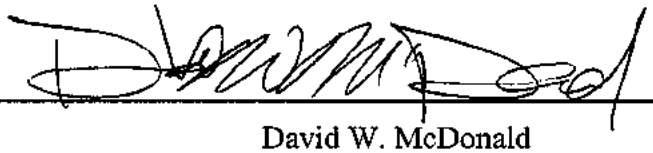
John H. Gennari

Reading Committee:

John H. Gennari



Jonathan Grudin



David W. McDonald

Date: July 21, 2005

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature 

Date 8/12/2005

University of Washington

Abstract

Supporting Collaborative Clinical Trial Protocol Writing
through an Annotation Design

Chunhua Weng

Chair of the Supervisory Committee:
Assistant Professor John H. Gennari
Department of Medical Education and Biomedical Informatics

Clinical trial protocols are important documents that guide clinical research. Modern protocol development requires collective expertise from a group of Loosely-Coupled protocol writers, who work across distances and time zones. Email has been the primary communication tool for these protocol writers. Unfortunately, it inadequately supports collaborative writing tasks. Without appropriate groupware technology, these protocol writers often compromise work efficiency and the degree of collaboration to complete their tasks. This situation is exhibited at the Southwest Oncology Group (SWOG), one of the Cooperative Group Programs under the direction of NCI. While it is clear that its current work practices do not support optimal collaboration, it is unclear how to improve the collaboration and communication in such group work because the complexities of collaborative protocol development has rarely been studied. This research utilizes and extends Computer-supported Cooperative Work (CSCW) theories to identify the problems in protocol development and to design groupware technology for

supporting this group work. This dissertation consists of four parts: (1) qualitative fieldwork of the collaborative protocol writing process at SWOG; (2) a design of an annotation model that facilitates in-context communication around evolving documents during the iterative reviewing and revising process; (3) a design and an implementation of a protocol collaborative authoring tool (PCAT) that embodies the annotation model from #2 to address group work problems identified in #1; and (4) a validation of the usability of the annotation model and the PCAT prototype. In addition, this dissertation implements a grounded design process and contributes a socio-technical design of groupware technology in a healthcare setting to the literature of socio-technical approaches for system design.

Table of Contents

Pages

List of Figures

Pages

List of Tables

Pages

Acknowledgements

This dissertation is by far the most significant accomplishment in my life and would have been impossible without the people who constantly believed in and supported me. They have made my study at the University of Washington an enjoyable and rewarding experience and I give them my heartfelt thanks.

I first thank my advisor, John Gennari, who has devoted so much time, energy, and caring to my education. He took a risk in recruiting me from overseas and has been patient when I was slow and understanding and forgiving when I was wrong. He motivated me to embark on my research journey, gave me thorough and timely advice at each step of my research and academic progress, and has fully supported my career. My professional growth in the past five years is directly due to John's help and guidance.

Many thanks to Ira Kalet for being my role model of scholarship and for promoting my career. He taught me how to find graceful solutions to problems both in life and in research. I enjoyed our discussions on topics ranging from the modern physics of black holes to William Shakespeare's quote "Brevity is the soul of wit." His example has been an important inspiration for me.

Cornelius Rosse gave me fatherly care and mentored me without reservation. He gave me the strength and taught me the strategies to battle against all kinds of difficulties. His insightful wisdom saved me a lot of troubles. Throughout my life I will benefit from his advice.

Jonathan Grudin showed great interest in my work, introduced me to relevant literature, reviewed my research, gave me valuable advice, and provided his full support. I appreciate all this very much.

I cannot remember for how many times after me trying to describe some vague research ideas, David McDonald focused me by not only teaching me the appropriate CSCW terminology but also pointing me towards relevant articles. I still have the stacks of papers that David printed for me without asking because he thought they might be helpful in my research. Thank you, David, for guiding me through the broad field of CSCW and for all of your helpful advice.

I also thank David Farkas for motivating my research at its early stage, for introducing relevant literature from the technical communication field to me, and for reviewing my dissertation and research with enthusiasm.

This work could not have been possible without the cooperation of the Southwest Oncology Group (SWOG), which opened its doors to my fieldwork and system implementation. The president and CEO of SWOG statistical center, John Crowley, generously provided research funding for my work, offering me valuable research facilities in his organization, and introducing me to his people. I owe much to John for giving me this great opportunity.

Many other people at SWOG contributed their time, introspection as well as their friendship. To name just a few, Dana Sparks, Jacqueline Benedetti, Diana Lowry, Jason McCoy, James Moon, James Faulkner, Larissa Rois, Jack West, Ron Bredehoeft, Catherine Rankin, Stephanie Green helped me understand the complex process of protocol development. I appreciate their efforts from my heart!

Thank you to Michael Kahn from FastTrack Ldt Co. for funding my research during my first year in the United States. It would have been impossible for me to come to this country to study without his funding support.

I thank my close friends in the past three years in Seattle who have made my life so enjoyable. They have given their friendship through thick and thin. They are Xiao Li, Yeping Su, Tao Xie, Ping Lin, Yu Qi, Chuanli Jiang, Shuai Lu, Meliha Yetisgen-Yildiz, Feng Gao, Tudor Georgescu, Haixia Dai, and Xiaodan Song.

Additional thanks go to Tudor Georgescu, Andrea Civan, Phil Spory, Alan Au, Ping Lin, Richard Philips, and Raven Travillion, who generously reviewed my dissertation and research papers and corrected my English errors.

I thank the Biomedical and Health Informatics program and its wonderful people. I am honored to have had the chance to be part of this department and contribute to its future success.

I thank my fellow students, especially Jim Tufano, Alan Au, Ping Lin, Raven Travillion, and Richard Philips, for their kind support during my study. They helped me with lots of public presentation practices.

I also thank Brian Brown for his prompt help and advice all the time.

Finally I would like to thank my family in China. My mother Jianlan Huang always has faith in me and encourages me to reach high. My father Yuantou Weng constantly supports me especially when I fail. They understand my life goals and accepted the fact that they would not see me over the five years of my PhD study. During the final stage of my dissertation, they even came to the US to take care of my daily life so that I could focus on my writing. They never complained when I could not spare time to show them around the beautiful city of Seattle. They sacrifice much for me and give me the best love they can. Whatever I accomplish will always be inseparable from their love and support.

My older brother, Qiuhua Weng, takes care of my parents in their retirement and has fully supported me pursuing my PhD degree in the United States. It is his efforts that enable me to concentrate on my studies and to explore my ideas secure in the knowledge that my family is well taken care of.

I dedicate my dissertation to my parents and my brother.

Dedication

To my parents

Yuantou Weng
Jianlan Huang

To my brother

Qiuhua Weng

Chapter 1: Annotation for Collaborative Writing

Collaborative writing permeates many aspects of our working life. It plays a role in a number of different tasks, such as collaborative design, collaborative information integration, and group learning. Writing improves with iterative reviews and revisions. However, the iterative reviewing and revising process is often fraught with significant challenges for conflict resolving, change representation, group coordination, and so forth (Farkas 1987; Posner and Baecker 1992; Beck 1993; Sharples 1993). This dissertation addresses such challenges for iterative reviews and revisions during the collaborative writing process. Over the past decade, various annotation systems have been designed to support collaborative reviewing of digital documents (Churchill, Trevor et al. 2000; Kahan, Koivunen et al. 2001; Brush 2002; Handschuh and Staab 2002). In these systems, annotations are created as in-line comments embedded in digital documents. Most of them, however, only work with static documents and barely support annotation management. There has been little technology that supports annotation on evolving documents and effectively manages shared digital annotations from multiple reviewers throughout iterative reviews and revisions.

My primary research goal is to understand and to support annotation during the collaborative writing process in the hope of improving in-context group communication around evolving documents. Here, "document" is a broad concept that refers to any written artifact, such as a text document, a software program, or

a flowchart diagram. In addition, "communication" has multiple meanings: (1) group discussion; (2) group awareness through shared feedback; and (3) persistent group conversations. "In-context communication" means communication situated in the context where the communication is inspired from or focused on; the context could be a piece of text in a document or a topic on which multiple messages are focused.

To achieve my primary research goal, I have selected a particular organization that is dedicated to cancer clinical trial protocol development through collaborative writing efforts, the Southwest Oncology Group (SWOG). I will describe in section 1.2 the characteristics of the group and the documents that they develop in the daily work.

Suchman initiates the concept of "situated action" (Suchman 1987) and emphasizes the importance of "making work visible" for system designers (Suchman 1995). Guided by her advice, I try to situate my system designs in a thorough understanding of the work. Therefore, my second research goal is to understand the group work of creating cancer clinical trial protocols.

Following these two research goals, my dissertation research consists of three parts: qualitative fieldwork, system prototyping, and evaluations for both. In the field work, I investigated the annotation uses in SWOG, identified the dynamic properties of annotations, and created a new annotation model to support collaborative writing of cancer clinical trial protocols. In the system prototyping

process, I designed a web-based collaborative writing system that improves group awareness and in-context communication for collaborative clinical trial protocol writers. In addition, the first two parts are closely intertwined, as I will describe in Section 1.4 and Chapter Three. Next I describe the broad context of my research and the major messages in each chapter.

1.1. Broad Context: Annotations and Collaborative Writing

Collaborative writing has become a well-known group activity because of the increasing need for collaboration in work and life. Many organizational leaders or school teachers have used it as an effective pedagogical tool to facilitate collaborative learning in classrooms or within organizations (Wolfe 2002). The writing product can vary from a software program to a document. All of these writing processes need iterative review and revisions because writing improves over iterations and is not usually an all-at-once effort. Below I demonstrate how iterative review and revisions are both essential and similar in two example scenarios of collaborative writing.

Scenario One: Software Program Design

Modern software design often involves team work, where software designers play varied roles or work on different parts of a software project. Team members, such as testers or requirements engineers, often make annotations on a shared piece of code or a shared requirements specification document. These annotations can indicate bugs in the code, suggest improvement for coding techniques, or record

revision notes associated with the code or the requirements specification document. Software designers later revise the code or the requirements specification document following the suggestions indicated by the annotations. The review and revision process continues to iterate throughout the entire software design and development cycle.

Scenario Two: Collaborative Writing of Research Documents

Scientific collaboration requires collaborative researchers to write research documents or scientific papers together, where they play different roles such as reviewers or editors. Currently there is no widely adopted annotation tool for collaborative writers to carry out the collaborative reviewing task. Following the most popular group writing strategy, writers write independently first, and then review one another's writings by making and sharing the comments within the group. Later the chief editor selectively incorporates the comments, revise the writings, and distribute the newer versions of documents to the group for subsequent reviews.

In both of these scenarios, software designers and researchers use annotations or comments in their collaborative writing processes. According to Webopedia (WebOpedia 2005), an annotation is "a comment attached to a particular section of a document. It is a way to use computers in a workgroup environment to edit and review work. The creator of a document sends it to reviewers who then mark it up electronically with annotations and return it. The document's creator then

reads the annotations and adjusts the document appropriately." These shared annotations or comments serve as communication channels and knowledge sources for collaborative writers during iterative reviews and revisions. Two major challenges confront collaborative writers here: (1) to ensure that communication carried out through annotations always occurs in the appropriate context while development documents are being changed and (2) to consolidate inconsistent annotations from collaborative writers. In reality, collaborative writers often have to communicate and share review comments separately from documents, as I will illustrate in Chapter Four. There is little prior knowledge about how comments have been used by collaborative writers and how technology could improve annotation uses in collaborative writing settings.

Over the past decade, digital annotation technology has become a burgeoning research area to support the development of digital libraries and the semantic web. However, approaches to digital annotations are still limited by the view of annotation as static labels that serve as metadata or commentary messages for static documents. Most annotation designs ignore the fact that documents may change. Therefore, once the document that an annotation is attached to is changed, the annotation may become orphaned, obsolete, or ineffective (Cadiz, Gupta et al. 2000; Brush 2002).

Limited by the assumption of a static document, prior research rarely addresses the challenges involved in collaborative annotation on evolving written artifacts and there is no efficient mechanism to manage annotations throughout the

lifecycle of a developing document. In addition, little research has been done to investigate the functions of digital annotations in collaborative design activities, such as collaborative writing.

Motivated by the above research problems, I focus my research on understanding and supporting annotation for iterative reviews and revisions of the collaborative writing process. In this dissertation, I have carried out my research in an example setting of collaborative writing in a medical domain, an organization that develops cancer clinical trial protocols through collaborative writing efforts.

1.2. Collaborative Clinical Trial Protocol Writing

Evidence-based medical practice relies on carefully controlled clinical research. As described by the National Cancer Institute (NCI):

> *Over the last several decades the medical community has come to recognize that the reliable assessment of new therapies in people must be a formal, structured process. Determining whether a new therapy actually makes sick people better, or whether a new preventive agent really prevents disease in people at risk, requires that studies be designed, performed, analyzed, and reported with great care. Otherwise, they will not yield reliable conclusions.(Armitage 2003)*

Clinical trial protocols are the important documents that provide details for rigorous clinical research and assure the quality of clinical care. Protocol

development is a very important step early in the life cycle of clinical research because any error generated during this step may propagate and magnify itself downstream in the process. In reality, the current protocol development process is far from ideal. It is a tedious process for most protocol writers. Moreover, generated protocols may contain errors, such as inconsistency and incompleteness, which are primarily introduced during writing.

Protocol development is complex in part because it requires collective expertise from biostatistics, oncology, radiology, patient care, immunology, and possibly other domains. Therefore, group work among multidisciplinary clinical trial researchers is indispensable. The NCI has sponsored the Cooperative Group Program (CGP), which promotes and supports multi-center trials for cancer treatment, prevention, and early detection. The Southwest Oncology Group (SWOG) is a CGP member. Protocol writers affiliated with SWOG are widely distributed across the country. Specifically, the statistical center is located in Seattle, WA; the operational office is in San Antonio, TX; numerous principal investigators for clinical trial designs are all over the country. Currently, SWOG protocol writers use email and Microsoft Word as their collaborative writing tools. They frequently face challenges from version control, communication, and group coordination. Word provides a "Track Changes" feature, but unless every writer uses it, the entire collaborative process can break down. Additionally, although email is the primary communication tool for protocol writers, it is not designed to support group discussion, process knowledge management, or group progress

tracking. In addition, email does not adequately support information management within a group. Very often a response to an important question posted by NCI is buried somewhere in someone's personal emails; hence it is difficult for the writing team to retrieve this email six months later when they want to justify their answers to NCI.

Unaided by groupware technology support, SWOG adjusts its work practices to minimize mistakes in protocol development. While SWOG protocol writers are aware that their work practices do not support optimal collaboration, they do not know how to improve the collaboration and communication in such group work practices because the complexities of collaborative clinical trial protocol development have rarely been studied. Although many efforts have been made to support clinical trial protocol development, none has investigated the process for protocol development (van der Lei 2000); therefore, little is known about what technology might be useful and available for improving the quality and efficiency of the group work for protocol development. Over the past several years, SWOG has made efforts to improve its protocol development productivity and work efficiency by trying out different information systems from outside software vendors. Unfortunately, so far none of them has worked successfully.

1.3. Support for Collaborative Clinical Trial Protocol Writing

The current situation within SWOG motivates this research to gain an in-depth knowledge about collaborative protocol writing and to explore the design space of

an appropriate tool for supporting the group work and in-context communication among collaborative clinical trial protocol writers.

1.3.1. Research Questions

This dissertation work aims to answer the following research questions:

1. What is the SWOG collaborative clinical trial protocol writing process like?
2. How can we create an annotation design to support in-context communication for collaborative writers during iterative review and revisions?
3. How can the annotation design support collaborative protocol writing at SWOG?
4. What lessons about healthcare groupware design can we learn from this research?

Research question #1 motivates me to investigate collaborative writing at SWOG and to understand communication complexities and annotation uses in the collaborative clinical trial protocol writing process. Research question #2 motivates me to design a new annotation model that supports annotation on evolving documents generated during the collaborative writing process. To address research question #3, I validate the usefulness of the annotation design by implementing it as a system prototype at SWOG. Through the system design and prototyping process, I try to understand the possible challenges for healthcare

groupware design. By addressing research question #4, I reflect on my work and summarize useful techniques for healthcare groupware designs, with a hope that this knowledge may be generalizable to group work support in other healthcare settings.

1.3.2. Research Hypotheses

I propose that a system supporting annotations on evolving documents during iterative reviewing and revisions can enhance in-context communication among collaborative writers. An understanding of the work details and user needs of collaborative clinical trial protocol writers at SWOG is integral to the design of such a system and can be strengthened through the evolutionary system prototyping process.

In support of this general hypothesis, I make two usefulness claims in my research: (1) an understanding of the current protocol writing process within SWOG is useful for the design of a system that supports collaborative clinical trial protocol writing over distances and (2) the system proposed in this research is useful to SWOG protocol writers. In Section 1.4, I describe the methodology that I used to carry out my research and to validate these claims.

1.3.3. Related Prior Work

Group work for clinical trial protocol development has barely been examined from the perspective of Computer-supported Cooperative Work (CSCW).

Previous efforts to support clinical protocol development are mainly from the knowledge representation perspective and are represented by Design a Trial (Modgil and Hammond 2003), EON (Tu and Musen 1996), and PROforma (Fox J 1998). Most of these existing knowledge-based protocol design systems are designed for a single clinical trial designer or are limited in research stages. They have not been through field trials by clinical trial researchers. They focus on encoding developed protocols in computerized formats and supporting execution or sharing of protocols on computer, but fail to augment the natural group work for protocol development among collaborative and multidisciplinary clinical researchers. In contrast, my research is focused on the support of group work, especially group communication, for collaborative clinical trial protocol writers.

To support in-context communication around evolving documents among collaborative clinical trial protocol writers, my work integrates and extends prior work from the following research areas: (1) communication support from collaborative writing research; (2) in-context asynchronous communication support from collaborative design research; and (3) socio-technical approaches for system designs.

There are few longitudinal studies of collaborative writing (Beck 1993). Most of these studies are limited in that they were carried out in experimental settings for a short period of time (less than three months). Protocol writing is special for these characteristics: (1) it has a longer duration (usually 4-12 months in average); (2) it happens in a real-world work setting; (3) it involves people with more varied

specialty expertise; and (4) there are rigorous review processes and high quality expectations for the resulting document. It is unknown how the collaborative clinical trial protocol writing process may be different from the previously studied processes, what challenges may be specific to the domain of clinical research, and what challenges may generalize.

A system design is the ultimate goal of a work process study. The communication issue has been one of four core issues in collaborative writing system designs (Sharples 1993). However, many synchronous collaborative writing systems do not have direct support for communication: they either assume collaborative writers can coordinate communication outside editors by other means or that the writing is used for the communication purpose. Prior systems do not address issues such as design rationale capture, in-situ discussion, and feedback support between reviewers and editors; therefore, the potential for group communication support that annotations hold has not been fully brought out.

Web-based collaborative writing systems quickly emerged with the development of the web; representative systems include wiki (Burrow 2004), BSCW (Horstmann and Bentley 1997), REDUCE (C. Sun 1997), and Web-Blogging (Treese 2004). These systems view annotations as merely static metadata or annotations, rather than as a leading force that drives the evolution of shared documents. Clinical protocol development has rigorous reviewing and revision procedures, which are poorly supported by these collaborative writing systems.

Writing is an iterative process and reviewing is an important component for quality control. Discussions among collaborative writers are crucial in framing ideas and generating content. If those discussions are not well linked to the evolving document, collaborative writers can easily lose the context for their communication. Sharples concurred on this point by saying that successful communication in collaborative writing should be in context (Sharples 1993). Existing collaborative writing system designs rarely address support for in-context communication. However, this topic has been active in other two closely related research areas, design rationale capture and annotation support, which have made great advancement supporting in-context communication among asynchronous collaborators. My research to support annotation during collaborative writing, with a particular focus on improving in-context communication during iterative reviews and revisions, connects multiple research areas, including annotation design, collaborative writing support, and design rationale capture (Lee and Lai 1991).

An appropriate design methodology is essential to a successful system design. In addition to creating an annotation model and a collaborative writing system, I refer to the literature for socio-technical approaches to system design, including prior work on participatory design, ethnographic fieldwork, and other groupware design methodologies. In Chapter Two I elaborate on this literature in more detail.

1.4. Methodology: A Socio-Technical Approach

Aiming at increasing the mutual feedback between qualitative fieldwork and participatory design and achieving the synergy between both, I implemented a socio-technical approach as illustrated in Figure 1.1. Expanding on Cockton's use of the term (Cockton 1999), I call this approach grounded design.

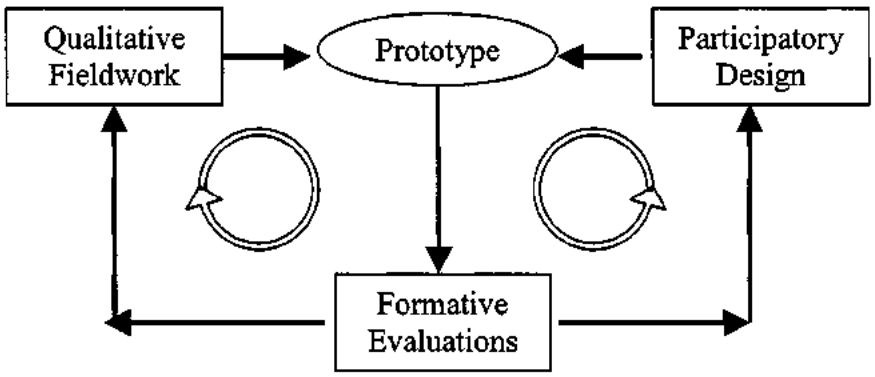


Figure 1.1 A Grounded Design Model

According to Figure 1.1, this grounded design process consists of three intertwined components: qualitative fieldwork and participatory design, both aided by formative evaluations. Qualitative fieldwork investigates instances of working roles and their work activities, and then sensitizes designers with the working context. Participatory design engages role-based user advocates, who understand users and can articulate their user needs to designers, to participate in the evolutionary prototyping process. I conducted formative evaluations to understand the interaction between user advocates and the prototypes, and to elicit user feedback for the prototype design. Formative evaluations also complement

the fieldwork with user input from the prototyping process. With this design model, I carried out qualitative data analyses throughout the iterative design process for deriving subsequent design hypotheses, identifying the focus for the subsequent field work, and adding to the knowledge of the work context. I will provide details for my research methodology in Chapter Three. Using this model, my work leads to three categories of research results: field knowledge about collaborative writing at SWOG (Chapter Four), a design to support collaborative writing through annotations (Chapter Five and Six), and evaluation results and lessons learned about groupware design (Chapter Seven).

1.5. Results: Field Knowledge for Protocol Development

Through interviews, participatory observations, and participatory designs, this work provides an understanding of the collaborative process of protocol development in the distributed organization SWOG. The major findings focus on the following aspects:

- What are the characteristics of collaborative clinical trial protocol writing processes?
- How do protocol writers review and revise clinical trial protocols under development?
- How do distributed protocol writers communicate and coordinate within a group?

- What are the major challenges for collaboration among distributed clinical trial protocol writers with various specialty expertises?

- What are the design recommendations for supporting collaborative protocol writers?

I elaborate on the answers to these questions in Chapter Four. Briefly, collaborative clinical trial protocol writing process is a loosely-coupled collaboration (Haake and Wilson 1992), where collaborators work asynchronously and autonomously most of time. Protocol writers have disparate training backgrounds and expertise, and work in different roles. Many protocol writers also do not share a consistent view of the protocol writing process. SWOG uses several strategies to avoid errors in collaborative writing, including the adoption of the single-scriber strategy and several writers' strategy (Posner and Baecker 1992) at different stages of the protocol development and the policy to distribute about only three drafts for reviews. The biggest challenge for SWOG protocol development comes from iterative reviewing and revising of protocol drafts, where in-context communication support is greatly needed. Part of this field knowledge inspired my design of an annotation model (Chapter Five).

1.6. Results: An Annotation Model for Collaborative Writing

I propose that an annotation model with dynamic properties such as contextual and activity information is useful for collaborative work around evolving documents. This annotation model defines both the communication context and

the document context for an annotation. A design of life cycle management for annotations helps users distinguish resolved annotations from unprocessed ones. I propose that it is unnecessary to robustly anchor every annotation to a new draft in collaborative writing if this annotation has already been incorporated or resolved. My annotation model defines an activity status for every annotation and this status information could help alleviate the "annotation overflow" problem in an evolving information space. The transitions of the annotation status enable collaborative writers to understand what happens to a particular annotation by indicating whether there are responses to an annotation, whether annotations are incorporated into a newer version of the document, and so on. Milestones in an annotation's life cycle could vary from system to system depending on the particular work. For collaborative writing support purposes, I define five statuses for an annotation: "unread", "read", "responded", "resolved", and "incorporated". With the status information, reviewers can stay aware when their opinions are received by others; writers can indicate their work progress by changing the status of annotations to "incorporated". Compared to existing annotation models, which do not support annotation incorporation and do not provide feedback to reviewers about how annotations are received by editors, this annotation design provides improved feedback between reviewers and editors.

Compared to existing annotation models such as RDF (Brickley and Guha 2004) and Dublin Core (Campbell 2002), this model has the following major additions: (1) it supports life cycle management for annotations; status information

determines the stage of an annotation in its life cycle and supports progress tracking and improves feedback between reviewers and editors; (2) it provides richer context information for annotations, including version information of the document, a physical text anchor in the document, as well as conversation context through pointers to threaded discussions; and (3) it defines structured communication messages and facilitates flexible annotation sorting based on annotation's different fields. Figure 1.2 shows an example of the status transition diagram for annotations and how annotations are carried across versions during collaborative writing processes. In Chapter Five, I will elaborate on this annotation model, its design principles and properties.

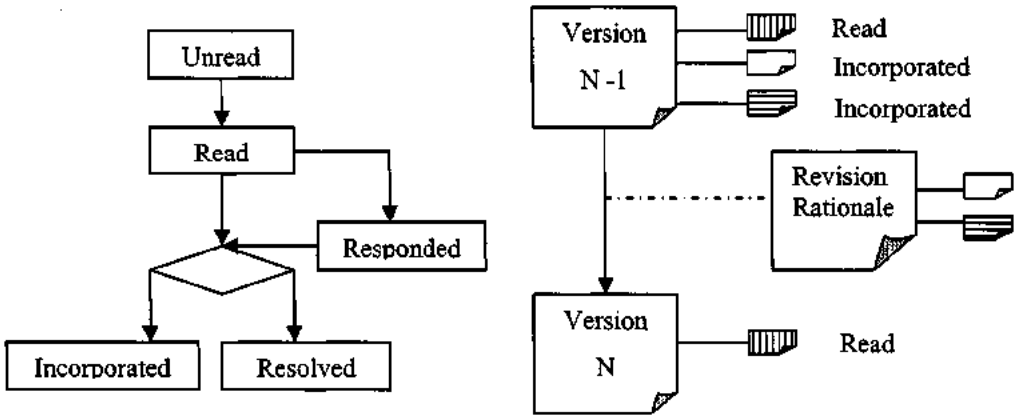


Figure 1.2 Status Transition and Design Rationale Capture

1.7. Results: The Protocol Collaborative Authoring Tool (PCAT)

To embody the above annotation design and to further understand the collaborative work among protocol writers by using a tool for user engagement, I

designed and prototyped a web-based collaborative writing system prototype called the Protocol Collaborative Authoring Tool (PCAT). PCAT supports these features: (1) awareness of the group work and individual contributions to the evolving document, (2) a shared repository of documents and annotations with version control for both, and (3) a web-based integrated reviewing and revising environment.

PCAT enables multiple protocol writers distributed over distances to review a shared web document or to edit different sections of the document synchronously. When protocol writers work asynchronously, they can track review and revision progress by subscribing to email notifications or by browsing activity histories for all annotations. Annotations can be shared within the group immediately after they are created. An annotation starts its status with "Unread;" later the status may be changed to "read", "responded", "resolved", or "incorporated." (See Figure 1.2) Annotations can also be threaded based on their topics to support in-context communication among collaborative writers. Once the shared document is changed or revised, unprocessed annotations will receive updated version information to stay consistent with the evolving document, while incorporated or resolved annotations will be automatically archived and hidden from the view of the updated document. In this way, collaborative writers can focus their work on unprocessed annotations only.

As an integrated collaborative writing environment, PCAT also supports user management, web email, progress report, and other web services.

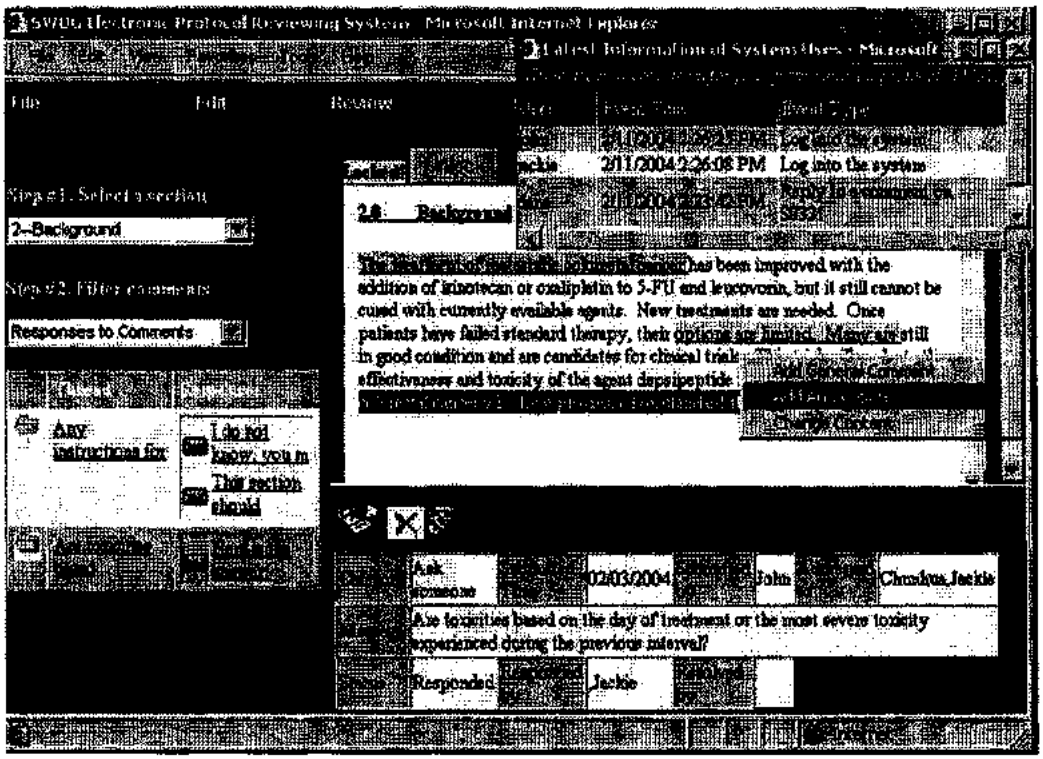


Figure 1.3 In-Context Discussion within PCAT

Figure 1.3 shows a screen shot of PCAT. I present the full details and functions of this system in Chapter Six.

1.8. Evaluation Results

I conducted formative evaluations throughout the PCAT system prototyping process and carried out two case studies of the PCAT prototype to validate the design of the annotation model and PCAT. I used participatory observation, semi-structured interviews, and survey techniques to collect user feedback during their uses of the prototype system. I will present the full details of these results in Chapter Seven. To summarize, my results show that SWOG users acknowledge

the usefulness of the annotation design. Among all the features, the top three highest-rated features are (1) pinpointing text in a document for an annotation (2) highlighting annotations from different reviewers in different colors and (3) addressing annotations to specified users for communication purposes. User feedback indicated that annotation sharing among collaborative writers is a double-edged sword. On one hand, there is an increased accessibility of annotations; on the other hand, the work load is shifting among users and there is an increase in the efforts for users to read shared annotations, which were not accessible before using the system.

I also analyzed the accumulated formative evaluation results throughout the participatory design process and identified a number of socio-technical challenges for system adoption in this healthcare setting. The challenges include changing user requirements, expanding user needs, lacking user incentives, disparate user backgrounds, and subtle organizational nuances. Such challenges may generalize to other group work settings.

Overall these evaluation results help me better understand: (1) the uses of digital annotations during collaborative writing; (2) the organizational challenges for designs of healthcare groupware; and (3) the user needs for collaborative protocol writing support. I present the details of the evaluation results in Chapter Seven.

1.9. Significance

This work has contributions for three fields: Medical Informatics, Computer Supported Cooperative Work (CSCW), and Social-technical Designs. This work provides an initial knowledge about the collaborative clinical trial protocol development process at SWOG and extends research on asynchronous collaboration around evolving documents. I designed an annotation model with improved support for in-context communication for collaborative writers. I also contributed a prototype collaborative clinical trial protocol writing system to cancer clinical trial researchers at SWOG. Moreover, this work supports collaborative writing by integrating and extending research from multiple connected areas such as annotation, collaborative writing, and design rationale capture. To validate the generalizability of my field knowledge and my system design is beyond the scope of my dissertation, but my work provides a basis to further this research. I will elaborate on these contributions in Chapter Eight.

1.10. Guide for the Reader

The remainder of this dissertation elaborates on the ideas and work presented in this chapter. In Chapter Two, I review and discuss related work that this research has built on. In Chapter Three, I describe the methodological approach that is being taken by the work. In Chapter Four, I describe a field study of the group work for clinical trial protocol development within SWOG. Chapter Five provides the details for an annotation model that supports asynchronous collaboration

around evolving documents. Chapter Six describes the PCAT prototype system for collaborative clinical trial protocol reviewing and writing over the Internet. Chapter Seven describes the evaluations designed to validate the usefulness of the designs of the annotation model and the PCAT prototype system. Chapter Eight provides a reflection of the work and lessons learned about healthcare groupware design and the major contributions made by this work. I conclude in Chapter Eight with challenges for healthcare groupware design and remaining research issues for my future work.

Chapter 2: Related Work

To support communication around evolving documents in the collaborative writing process, it is important to understand the embedded communication complexities and challenges. I describe previous efforts to support protocol development in section 2.1. My work integrates and extends on prior work from multiple areas in a fruitful way. In section 2.2 I describe previous studies and understandings about collaborative writing processes including their communication characteristics and tool support needs. On this basis, in section 2.3, I describe and compare communication support in multiple different domains. In section 2.4 I describe existing socio-technical approaches to groupware designs and analyze their strengths and limitations.

2.1. Efforts Supporting Protocol Development

A few earlier medical informatics researchers have recognized the importance of supporting clinical trial designs. For example, Hammond and Modgil et al. developed a knowledge-based system called "Design a Trial" to critique trial design ideas provided by doctors and to generate natural language protocol text (Modgil and Hammond 2003). Musen and Tu used a knowledge-based approach to support protocol knowledge representation in EON (Musen, Tu et al. 1996). Some other efforts for clinical trial design support vary from workflow streamlining, such as WITH (Fazi 2002), to providing sharable and reusable text, such as WebCTES (Franciosi, MacLeod et al. 2001).

These tools facilitate the automation, standardization, and dissemination of clinical protocol knowledge; but they have some common limitations. First, most of them take a knowledge-based approach to design rigorously computable models for protocols. Such knowledge-based systems are not intuitive to domain experts and necessitate assistance from knowledge engineers, who might complicate the task. Second, group-writing task is often characterized by a high degree of ambiguity and the lack of a fixed goal. Such work requires support for interactive and expressive group communication, which is often missing in knowledge-based systems for protocol development. Most of them are designed for a single author. A few recent studies have shown that protocol development is a collaborative scientific process that relies on achieving consensus within a interdisciplinary group of clinical trial experts (van der Lei 2000; Fazi, Luzi et al. 2001; McDonald, Weng et al. 2004). However, so far no prior research has investigated the work process, either single or group, for clinical trial protocol development.

2.2. Understandings of Collaborative Writing Processes

Previous studies about collaborative writing offer relevant knowledge to my work from the following aspects: (1) the nature and complexities in the collaborative writing groups; (2) the challenges involved in iterative reviews and revisions; and (3) the communication support needs among collaborative writers. I conclude this

section with an analysis of the need to investigate clinical trial protocol writing processes.

2.2.1. Nature of Collaborative Writing

Quite a number of researchers have gained understandings of collaborative writing processes through surveys or interviews (Lunsford 1992; Posner and Baecker 1992; Beck 1993). Beck surveyed the experiences of twenty three academic collaborative authors. He found that discussions took place during writing more than before or after writing, and adherence to an initial plan was not seen as an important determinant of success. Collaborative writing is reported to be a dynamic process with continuous negotiation and re-negotiation of questions related to content or the group. Posner and Baecker interviewed individuals working on twenty two collaborative writing projects from computer science, medicine, psychology, and other areas. They demonstrated the rich strategies used by co-writers, such as single writer, single scriber, separate writers, and joint writers. They also provided thirteen design requirements for collaborative writing systems, such as "*preserve collaborator identities*", "*support communication such as annotation*," and "*support separate document segments.*" Such requirements partly inform the design of the collaborative writing system in this work as Chapter 6 describes. Ede and Lunsford also studied collaborative writing and concluded that collaborative writing is not a stable or coherent construct, and it occurs in complex and multiple modalities. They found that (1) authority drives the hierarchical structure of a collaborative writing group; (2) productivity and

efficiency is valued through assembling information to solve problems; (3) the dialog is loosely structured; and (4) the roles of writers are fluid. These results imply that it is important to understand the organizational structure and characteristics of collaboration among protocol writers when we try to support communication in a collaborative writing process.

2.2.2. Support for Iterative Review and Revisions

Communication among editors and authors during iterative review and revisions was first thoroughly studied in the Technical Communication domain (Farkas 1987; Farkas 1991; Farkas 1995). Farkas pointed out that editors need to indicate, explain, negotiate, and distribute changes to authors. Therefore, a good annotation model should support mutual feedback between editors and authors. Farkas also found that it is important to archive not only developed documents, but also review comments, all drafts, and personal notes throughout document development. This finding implies that document development rationale capture is very important. However, currently no review tool captures all this information. Later, Neuwirth et. al (Neuwirth, Kaufer et al. 1990) noted that writers could not understand review comments well, and were frustrated about inconsistent comments. Green summarized requirements for document review support (Green 1997), which include inserting comments in context, identifying reviewers, viewing comments and documents together, sharing comments immediately, and so on. Kim et. al. used interviews to understand review practice of collaborative writers (Kim and Eklundh 2001) and identified some common strategies within

groups, such as centralized document control and small group size. They also observed that change representation disturbs the flow of reading and collaborative writers tend to give general comments about the changes that they make. In summary, little is known about how writers incorporate review comments and share review comments within the writing team.

2.2.3. Communication Support Needs in Collaborative Writing

Some field observations about communication in collaborative writing include studies of grade school children writing in a classroom (Mitchell, Posner et al. 1995), MBA students writing in an experimental laboratory (Olson, Olson et al. 1992), and university researchers collaborative writing over a web-based tool (Collaboratus) (Lowry 2002). These studies are limited in that they were carried out in experimental settings for a short period of time, less than three months. As Beck and Bellotti noted that there are few longitudinal studies of collaborative writing (Beck 1993). Beck & Bellotti conducted in vivo case studies of three collaborative authoring projects that lasted between three weeks and one year. The three groups relied heavily on phone, diskette, fax, and electronic file exchange to communicate and exchange the shared document.

2.2.4. Summary

Compared to previously collaborative writing processes, protocol writing has some unique characteristics: (1) it has a longer duration (usually 4-12 months); (2) it happens in a real-world work setting; (3) it involves more varied people with

diverse backgrounds; and (4) it has rigorous quality control and involves both peer and hierarchical reviews. It is unknown how collaborative clinical trial protocol writing processes may be different from the above studied processes and what challenges may be specific to the domain of clinical trial research and what challenges may generalize. Although many efforts have been devoted to clinical trial protocol development support, none has addressed what the process is exactly like for protocol development (van der Lei 2000). An understanding of protocol development processes is missing but valuable to the collaborative writing literature. The current deficiency of this understanding motivates me to carry out a fieldwork at SWOG and the results are described in Chapter Four.

2.3. Previous Approaches for Communication Support

Writing is a creative design process with a document as the end product. Therefore, communication support for collaborative writing can borrow ideas from multiple areas, including (1) communication support for collaborative writing; (2) document-centric communication and collaboration support through annotations; and (3) in-context communication support for collaborative designs.

2.3.1. Communication Support in Collaborative Writing Systems

A system design is the ultimate goal of a work process study and many collaborative writing systems have been created. Communication issue is one of four core issues in collaborative writing system design (Sharples 1993). However, many synchronous collaborative writing systems, such as ShrEdit (Dourish and

Bellotti 1992) and GROVE (Ellis, Gibbs et al. 1991), do not have direct support for communication by assuming collaborative writers can coordinate communication outside editors by all means or the writing itself is communication. There is only some implicit communication support for reviewers and editors in asynchronous collaborative writing systems through annotation designs.

The most relevant systems include Quilt (Fish, Kraut et al. 1988), PREP (Neuwirth, Kaufer et al. 1990), and InterNote (Catlin, Bush et al. 1989). Quilt supports annotations on work-in-progress drafts or annotations, and organizes these artifacts in a tree structure. It also supports information sharing through an activity log and information exchanging through semi-structured messages. PREP is another work-in-preparation editor that supports "writing through commenting." This system explicitly highlights the importance of supporting communication between writers and reviewers via comments. PREP is advantageous for its column-structured interface for displaying a writing plan, content, and associated comments. This interface design is the initial one that enables reviewers and writers to see content and comments side by side. Both Quilt and PREP are limited in that they do not provide support for annotation incorporation; InterNote filled this gap. InterNote is designed to enhance annotative collaboration through a hypermedia framework. InterNote supports two types of annotations: suggested changes and textual commentary for notes, comments, and non-specific suggestions. Researchers of InterNote also specified a set of widely useful requirements for annotation support in collaborative writing

systems, including "annotations from multiple reviewers should be sharable, sortable, and merge-able" and "authors should be able to incorporate annotations easily." To satisfy the first requirement, an annotation in InterNote is a structured object with multiple properties. To satisfy the second requirement, InterNote uses a mechanism called "Warm Linking" to update content by incorporating annotations through dynamic links, through which users can "push" the content of an annotation into a document or "pull" content of an annotation from a document for revising. The annotation model design in my research borrows ideas from the above designs to a great degree, as I will describe it in Chapter Five.

These three systems are all prior-to-web generations without the wide accessibility needed by a mobile group, especially those protocol developers who are often temporarily involved in protocol writing and distributed all over the country. It is hard to ask these protocol developers to install a system if they just write only one protocol in their whole career.

Web-based collaborative writing systems quickly emerged with the development of web; representative systems include wiki (Burrow 2004), BSCW (Horstmann and Bentley 1997), REDUCE (C. Sun 1997), and Web-Blogging (Treese 2004). These systems utilize the popular collaborative platform, the web, to support collaborative writing. BSCW provides a shared information space to facilitate collaboration among authorized users; wiki enables everyone to change the content of a web page from anywhere; and web blogs chronically organize content from multiple sources. Unfortunately, these systems do not inherit the

annotation designs in the previous three systems and share these limitations: (1) most annotations in these web-based systems are appended at the end of a document segmentation or a document, rather than precisely located around the text where the annotations are attached to; (2) there is no further advancement on annotation design to facilitate automatic annotation activity tracking or notification. None of these systems support "annotation incorporation" as InterNote supported a decade ago. These systems view annotations as static metadata or comments, rather than a leading force that drives the evolution of shared documents. Protocol development has rigorous reviewing procedures and iterative reviewing and revising processes; the above systems do not satisfy such needs as well.

As Reeves pointed out, writing is an iterative process; reviewing is an important component for quality control for a shared document (Reeves and Shipman 1992). Reviewers and writers often need to discuss or clarify some writing problems; it is very important to support such discussions or communication in the context of an evolving document for several reasons (Trigg, Suchman et al. 1986; Reeves and Shipman 1992). First, research has shown that problems and solutions often co evolve (Fischer and Reeves. 1992); discussions among collaborative writers are crucial for framing the idea and generating the content. A related concept, situated cognition, points out that problem solving, understanding, and learning often occurs in the context of the environment. If the activities of collaborative writers could not be appropriately linked to their context, collaborative writers can have

difficulty in making sense of their communication. They may also run into the same discussions again and waste time. Sharples also concurred on this point by saying that "*successful communication in collaborative writing should be in context.*" (Sharples 1993) Unfortunately, the above collaborative writing systems do not adequately support this feature, which however relates to research efforts on in-context communication support from other areas as follows.

2.3.2. Communication Support through Annotations

With the further development of Internet, research supporting collaborative reviewing makes substantial progress in interface design and annotation modeling. Popular annotation models include RDF (Brickley and Guha 2004) and Dublin Core (Campbell 2002), which define static properties for annotators or the annotation content. Based on such data models, most collaborative reviewing systems are called "annotation systems" because reviewing messages are designed as embedded and sharable annotations attached to documents. These annotation systems, such as comMentor (Roscheisen, Mogensen et al. 1997), Annotator (Ovsiannikov, Arbib et al. 1999), Annotea (Kahan, Koivunen et al. 2001), critLink, and CREAM, primary different from one another in interface design. Among them, most relevant systems to this research include D3E (Sumner and Buckingham Shum 1998), CoNote (Davis and Huttenlocher 1995), WebAnn (Brush 2002), and StickyChats (Churchill, Trevor et al. 2000). The first three are web-based and support asynchronous working mode, and the last one is based on instant messenger technology and requires synchronous collaboration. D3E is a

digital document discourse environment designed to facilitate scholarly publication and peer reviewing. It displays comments as threaded discussions side by side to a reviewed document. Reviewers can view ongoing topics of discussion and join interested ones easily. In D3E, documents are also divided into segmentations and comments are anchored to the end of them. Granularity for annotation anchoring is not high in D3E. Similar problem exists in CoNote, which is designed as a collaborative learning environment based on HyperNews for student uses in classrooms. A reviewer has to go to pre-defined locations in a document to add annotations. Therefore, this system lacks the flexibility for "annotating anywhere in the document." CoNote embeds a lot of needs from classroom uses, which sacrifices its generalizability. In contrast, WebAnn is designed to be a generalizable annotation system based on the common annotation framework developed by Bargeron et. al (Bargeron, Gupta et al. 1999). Compared to previous annotations system, WebAnn supports enhanced annotation notification service and improved annotation anchoring granularity. A user can highlight any text span and insert an annotation. It also supports in-situ discussions through annotation replying, editing, and deletion.

Green described a review tool Revufile used within IBM (Green 1997). Revufile also defines a type and a disposition for each comment. A type indicates the importance of a comment. A disposition could be either "unassigned", "open", or "accepted". It is helpful for work coordination and group communication. The definition of these two properties supports group communication without

interruption or cost. It is one of the earliest effot to define dynamic properties in annotation designs. However, these properties only support "review" processes and provide little editing support. It is unknown how comments are incorporated.

Annotation systems provide a better support for collaborative reviewing than those collaborative writing systems presented in the previous section; however, most of the annotation systems work well for static documents only and have the "annotation orphaning" problem (Brush 2002) for dynamic documents generated in collaborative writing processes. Sharples et. al viewed communication and version management as two separate issues for collaborative writing support (Sharples 1993), which is not quite right in this context. Communication support unaided by version management could easily go out of context. The annotation orphaning problem leads to a need for an improve annotation system design that could integrate appropriate version control, annotation incorporation, and collaboration support around annotations. Unfortunately, by far these issues have been tackled separately by researchers in different areas.

2.3.3. Communication Support for Collaborative Design

It is unclear what the communication support requirements are for asynchronous collaborative writing systems. Barely addressed questions include (1) what communication means to asynchronous collaborative writers, (2) what should be communicated among them, and (3) how we can support in-context communication. To help address these issues, I borrowed some ideas from a

closely relevant research area that supports in-context communication: design rationale capture.

Design rationale is an explanation of why an artifact is designed as such (Lee and Lai 1991). The usefulness of design rational capture has been well recognized by software designers. Zaychik and Regli state the importance and challenges to capture communication and context in software project lifecycle (Vera Zaychik 2002). They point out that email as the widely used communication tool poorly supports context-based communication among co-workers; therefore, valuable design knowledge is often hard to be extracted from email communications. This is also true in document development organizations.

Design rationale capture systems provide a framework for recording discussions about design decisions. Grudin identified three key reasons for design rationale capture, which are for *subsequent consultation, design improvement through focused attention to capture rationale, and communicating progress to other group members* (Grudin 1996). Applying his ideas to asynchronous collaborative writing, I infer that such things should be communicated timely: document evolution rationale, work progress, and feedback for the ongoing writing. Collaborative writers need design rational capture to understand how a work-in-progress document evolves and what resources justify or lead to its evolution. There is also a need to communicate progress between reviewers and editors, because their progress inter-depends on each other. Document design rationale capture helps reviewers understand how comments are incorporated by editors.

A famous design rationale capture system is the Issue-based information system (IBIS) for helping members of a project team discuss issues related to a problem and come to a consensus on a solution (Kunz and Rittel 1970). In IBIS, participants in the online discussion argue about design issues by taking positions and making arguments for and against those positions. IBIS provides a nice hypertext interface for organizing positions and arguments around issues. Users can display overview diagrams to glance at captured design rationales and look at the underlying text if required. For unknown reasons, the above nice ideas from IBIS have received little attention or adoption.

2.3.4. Summary of Communication Support Approaches

In sum, during asynchronous collaboration writing, collaborative writers need to communicate writing rationale, work progress, and review comments in the context of their work. The need some mechanism that support co-evolving discussions and documents. A new system design can extend previous annotations system designs to achieve these goals by integrating research ideas from the research areas in version control, design rationale, annotation notification services, and collaborative writing support. Inspired by the related work summarized above, I provide such an annotation design in Chapter Five.

2.4. Groupware Design Methodology

Supporting collaborative protocol writing requests an effective groupware design methodology. Only in recent years, group work has received more attention and

studies in the medical informatics community (Pratt, Reddy et al. 2004), and groupware design methodology is even rarer. However, with more and more healthcare activities carried out by group work, groupware design will be urgently needed by this community sooner or later. My work as an exploratory study toward an understanding of challenges involved in groupware design for multidisciplinary healthcare researchers has been inspired by several groupware design theories or methods.

2.4.1. Social-technical Design

The theoretical foundation of this research design is the social-technical approach to system design (Emery and Trist 1960). Ehn points out that "*requirements specification and system description based on information from interviews were not very successful* (Ehn and Kyng 1991)." Ehn also acknowledges that there is more to software systems development than building the right system or building the system right; there is a social element of joint discovery and learning. O'Day provides a social-technical design cycle (O'Day, Bobrow et al. 1996), and points out that field studies and system designs can inform each other.

There are two limitations of this design method. First, it is unknown how mature the system design should be before it could be used to inform field studies. Second, it is a "system-oriented" research method, and the field study is used to serve the system design. There is a one-way communication from field work to

system design, but the potential of the system design for deepening the field understanding is omitted.

2.4.2. Ethnographic Fieldwork

Some researchers intertwined social and technical aspects by basing requirements on an empirical understanding of the social organization where the system will be used. Suchman is the first researcher who emphasizes the importance of "*making work practice visible*" for system designers in her work study research (Suchman 1995). Since then, ethnography has been widely adopted to understand work practice in work-oriented design or ethnographically-informed design. However, the attempt to make work practice visible and intelligible for system design necessarily relates to two bodies of knowledge: the actual work activities and knowledge of practitioners, and what is considered relevant information for requirements analysis in system design. Researchers have recognized that it is not easy to precisely identify these two parts. Many sociologists think that pure ethnography should not be interpretive: "*Enacted work practice details are not texts which symbolize meaning or events. They are in detail identifying them, not anything else. (Garfinkel 1996)*" Therefore, ethnography is often criticized for two major limitations: (1) it is time consuming and impractical for fast and efficient system design; and (2) it is hard to evaluate what ethnographical results are relevant to system design, and what designs are feasible or applicable (Shapiro 1994; Crabtree 1998). To address problem #1, John Hughes defines four practical ethnography strategies to make ethnographic studies match the fast pace of

software design. These are (1) concurrent ethnography (2) quick and dirty ethnography (3) evaluative ethnography and (4) re-visit previous studies (Hughes, King et al. 1994). Similarly, Millen proposes "*rapid ethnography for system development* (Millen 2000)," which is a collection of field methods intended to provide a reasonable understanding of users and their activities given significant time pressures in the filed, but these field methods are centered on values of commercial products.

2.4.3. Participatory Design

Meanwhile, another method to approach "socio-technical" designs is to carry out a participatory design (Muller 2002), which was first used to maintain the democracy, and later used to engage users in software design with a central idea to actively engage users throughout the design process. This method relates to schools of techniques to addressing its fundamental problems such as "how design choices can be warranted" and "how to select participants that represent users." Pankoke-Babatz et. al. use "user advocacy" in system adoption studies and prove its effectiveness for representing users (Pankoke-Babatz, Mark et al. 1997) because user advocates can communicate between designers and real workers based on their experiences of system uses and their knowledge of the work. And there are all sorts of techniques to engage users throughout a participatory design, including mockups, evolutionary prototypes, and simulations.

2.4.4. Combining Ethnography and Participatory Design

Later, pure participatory design is critiqued for possible overreactions to user opinions and potential dangers of designing systems that work best for wrong work scenarios (Crabtree 2003). On this basis, instead of using general system design methods, researchers started to intertwine ethnography and participatory design in particular and address problem #2 identified above. Blomberg and Suchman intertwined ethnography and participatory design in software designs (Blomberg, Suchman et al. 1996). Helena Karasti extends Crabtree's idea and proposes "*work practice sensitive participatory design* (Karasti 1997)." In her dissertation research, she developed a video analysis method to use ethnography results in participatory design; however, it only involved workers in design workshops and engaged participants in discussions only. It is hard to tell how that method would be effective because they did not evaluate their design ideas by embodying the design ideas into interactive artifacts. By furthering this research, Crabtree pointed out that "*ethnography should run in parallel to exploratory and experimental activities of user participation, thus enabling the design to maintain an adequate grasp on the current practice in working up the future through cooperative design* (Crabtree 2003)." Even so, Crabtree still struggled with challenges from conventional ethnography methods, including intervention, interpretation, and user proxy. Crabtree and Karasti recognized the importance of integrating ethnography and participatory design, but did not give a practical solution to the involved challenges.

Hartswood et. al. called for a principled synthesis of ethnomethodology and participatory design and coined a concept called Co-realization. Co-realization means that the full implications of a new system design for work practices cannot be grasped by studying the work as it is now, but will only be revealed in and through the systems subsequent uses (Hartswood, Proctor et al. 2002).

2.4.5. Summary of Design Methodologies

In this section I review the relevant groupware design methodologies including social-technical design cycle, ethnography, participatory design, and combined ethnography and participatory design. Overall, this research has separated qualitative fieldwork from system designs, or has largely focused on design-oriented ethnography without considering how designs might provide feedback for ethnography. Those approaches created a one-way communication between users and designers, "*information is floating from the work practices to the designer, but no information about the future technology or uses is floating back to the future users in the workplace* (Bardram 1996)."

2.5. Summary of Related Work

In sum, my work in this dissertation is focused on designing annotations for collaborative writing by integrating and extending work from areas including annotations, in-context communication, and design rationale capture. I also borrowed ideas from the social-technical design cycles, and adopted a variety of

socio-technical design methods and come up with a "Grounded Design" method as I will describe in Chapter Three.

Chapter 3: Methodology

Many system design methods make qualitative field work an integral part of system designs, such as contextual design (Beyer and Holtzblatt 1999) and participatory design (Kensing and Blomberg 1998; Muller 2002). However, these embedded field studies are often design-oriented and are prescriptive. Alternatively, ethnography has been widely used to provide descriptive field knowledge. The typical problem of conventional ethnography is that it is hard to communicate relevant field knowledge to system designers and to derive useful system design ideas from this knowledge.

The objectives of this dissertation research are to gain knowledge about the work for protocol development and a deeper understanding of the field through a system prototyping process. Therefore, in this chapter, I describe my design method called "grounded design" that achieves both a descriptive field understanding and a specific system design by intertwining concurrent fieldwork and system prototyping.

3.1. A Grounded Design Model

In early 2004, Cockton proposed a new design model called "Grounded Design" (Cockton 1999), which is inspired by the Grounded Theory (Glaser and Strauss 1967). As Cockton described, grounded theory grounds theories in data and cycles through data collection and coding until coding stabilizes, whereas

grounded design grounds a design in contextual and evaluation data, and cycles through contextual research, designing, and evaluating until problems require no further contextual research and/or design changes. Cockton also emphasized that verification of design should be achieved through evaluations. Cockton coined a beautiful concept but did not provide concrete methods to achieve it. I have borrowed this concept and implemented it in my way, as illustrated in Figure 3.1. This method is based on the following premises:

- Qualitative fieldwork identifies representative working roles for participatory design.

- A system prototype is not only a means to validate design ideas but also an interactive tool to help achieve an in-depth understanding of the current work practices and validate the accuracy of this understanding.

- Evaluation should be an integral component of an iterative system design process.

- Parallel qualitative fieldwork and participatory design simultaneously inform system prototyping and effectively provide feedback to each other.

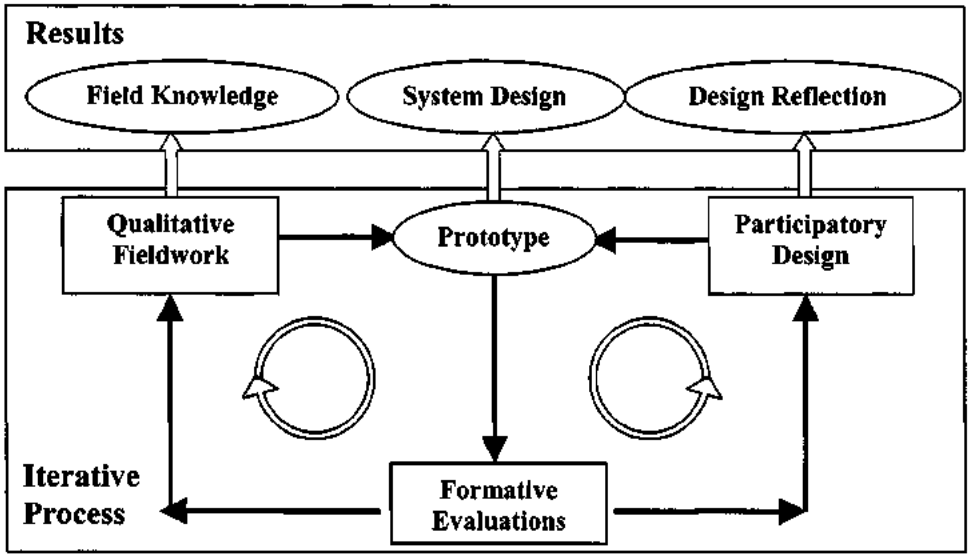


Figure 3.1 The Grounded Design Model

There are three concurrent processes represented by three squares in Figure 3.1: qualitative fieldwork, participatory design, and formative evaluation. This design model leads to three research results, as represented by three ovals in Figure 3.1: field knowledge about protocol development (Chapter Four), a system design (Chapter Five and Six), and evaluation results as well as a design reflection (Chapter Seven).

Qualitative fieldwork describes the work process and assembles instances of working roles and their work activities to sensitize system designers with knowledge of the work context. The participatory design process engages role-based user advocates into evolutionary prototyping of the system. Both fieldwork and the participatory design inform the designs of the prototype. The prototype plays two important roles in the whole research process: (1) as an embodiment of

the design ideas and (2) as a measurement tool to better understand the users through their interactions with the tool. Both of the qualitative fieldwork and the participatory design process receive feedback from formative evaluations, which are carried out to understand the interaction between the user advocates and the prototypes. The three processes are concurrent without any strict ordering among them. Qualitative data analyses are conducted throughout the iterative designs for deriving design hypotheses, for adjusting subsequent fieldwork, and for improving understandings of the work context.

The primary objective of this grounded design model is to derive system designs from field knowledge. This method is different from some modern participatory design processes that include qualitative fieldwork and formative evaluations. First, qualitative fieldwork in conventional participatory design methods is prescriptive and focused on a particular system design. However, a field contains much knowledge that may lead to more than one system prototype. In this method, qualitative fieldwork is not driven by a particular design. During this research, I was open to all possible ideas that may improve the current work practice of protocol writers at SWOG and was trying to make the field research and the participatory design meet in the middle from both ends.

Second, formative evaluations in conventional participatory design processes are focused on the system prototypes' usefulness, effectiveness, or other system measurement. In my grounded design method, formative evaluations have dual purposes: one is to assess the system prototype, and the other is to assess the

accuracy of the field knowledge. As Figure 3.1 shows, there are two feedback loops, rather just one. Qualitative fieldwork largely relies on interviews, observations, or other qualitative methods. The availability of a system prototype makes it possible to deepen the field understandings through the dynamic interactions between users and the prototype. The prototyping process complements interviews by eliciting some tacit knowledge from users. Users can also better articulate their ideas with a concrete prototype system.

I expect this grounded design methodology to increase the synergy of both participatory design and qualitative fieldwork in three ways: (1) qualitative fieldwork provides work context information to designers; (2) participatory design engages users and brings out tacit work knowledge; and (3) ongoing formative evaluations validate the accuracy of the fieldwork results and verify the evolving system designs.

3.2. Data Collection

Before my data collection process, I had received an approval for human subject research from the University of Washington. Next I will describe how I collected data for this research during the qualitative fieldwork, the participatory design, and the formative evaluations.

3.2.1. Qualitative Fieldwork

My qualitative fieldwork at SWOG focused on group communication, information integration, and group workflow. In further detail, this work used participant observations of Protocol Review Committee (PRC) meetings, semi-structured interviews, and collection and analysis of protocol development artifacts to leverage information gained from these sources.

Participant Observation

At the outset of this research, protocols were reviewed twice in PRC meetings. I was on site for three months to observe PRC meetings, which were attended primarily by local statisticians and one protocol coordinator manager from San Antonio via conference calls. I made notes on the behaviors of reviewers and the content of discussions. I observed where in protocols major questions or discussions occurred, how reviewers made and shared comments, discussed questions, and resolved disparate opinions.

Artifact Collection

During the fieldwork and the participatory design process, I collected the following protocol development artifacts including (1) email communication among protocol developers, which were forwarded to me by protocol writers; (2) protocol review comments: some of which were from PRC meetings or from other roles such as data coordinators or study coordinators; and (3) organizational

protocol development guidelines or standards. These data complement one another by contributing different knowledge. I analyzed emails to derive communication patterns and identify problematic scenarios. I analyzed comments to identify typical problems in developing protocols and their distributions across sections and to understand the uses and functions of review comments. I also referred to SWOG guidelines or standards to understand the organizational expectations or views of protocol development.

Semi-Structured Interviews

During the fieldwork study, I identified five key roles for protocol development: *(1) protocol coordinator, (2) study coordinator (or principle investigator), (3) biostatistician, (4) data coordinator, and (5) Protocol Review Committee (PRC) member*. A writing team usually consists of the first three roles, aided by reviewers working in the other two roles. I interviewed fifteen SWOG protocol writers covering the above five roles all of whom had been recently involved in protocol writing. I accessed them through the snowball sampling method, a special non-probability method used when the desired sample characteristic is rare. This method is appropriate for my research because it was difficult to locate widely distributed protocol writers. Snowball sampling relies on referrals from initial subjects to generate additional subjects.

Each semi-structured interview lasted about an hour. During the interviews, I used two techniques. One was the "process query" technique by asking each

interviewee to draw a diagram of the writing process based on their perceptions. The other method was the "Critical Incident Technique" by asking the interviewees to recall some extreme scenarios that they experienced. Critical Incident Technique helped me gain knowledge about real personal experiences while minimizing interference from stereotypical reactions. I taped all the interviews, transcribed half, and made notes for the remaining half.

I divided my qualitative fieldwork into two parts. One was prior to the first attempt at a prototype for system designs. The first part helped me gain design-sensitive work practice details and informed my participatory design process. New knowledge gained during the participatory design was used to guide the second part of field study. Therefore, the second part of my fieldwork was carried out concurrently with my participatory design. The whole design process has been illustrated in Figure 3.1.

3.2.2. Participatory Design

The participatory design spanned more than one year. Next I describe three techniques of participatory design which I used in my research. They are role-based user advocacy, iterative prototyping, and change management.

Role-based User Advocacy

Protocol development at SWOG is a group work divided among a number of strictly predefined roles. I came up with the idea of "role-based user advocacy" to

engage the important roles in the group work into a system design process with the hope that the design would not just support certain roles while ignore the needs of the rest.

With the help of the SWOG leader, I formed a participatory design team including three users who were advocates of new technology, had ample protocol writing knowledge, understood the work complexities, and represented a normal protocol development team. They were a protocol coordinator and two statisticians, who had been closely involved throughout the system prototyping process. These participants evaluated system prototypes in their real work practices and provided feedback for the design.

In addition, I kept a couple of executive officers informed about the ongoing participatory design. Periodically I reported the status of my prototyping and demonstrated my prototypes to them. They understood the organization's culture, provided the leadership, and encouraged the participatory users to try out the new technology from this research.

Iterative prototyping

A prototype is a modifiable and extendable representation of a planned software system. Prototypes vary in fidelity, excitability, and the scope of representation. In this dissertation research, I started with an exploratory prototype, then used an experimental prototype, and finally worked on an operational prototype.

At the beginning of my participatory design, I developed low-fidelity paper-based mockups to assess the logical feasibility of my designs. I call it an exploratory prototype because it is a throw-away prototype used to clarify design goals, to identify requirements, and to examine alternative designs. I designed two alternatives: one is a web-based collaborative writing system, the other is a Microsoft Word-based additional plug-in that supports collaborative writing features. My participatory designers at SWOG preferred the web-based design to the Word plug-in. These mockups were designed using Microsoft Visio, a drawing package that helps create sample screenshots that look as though they are from real Windows systems. I developed new workflow scenarios with these system prototypes. I performed cognitive walkthrough on the scenarios for my participants and elicited their feedback to refine the design. This process took a few months to complete and was cost-effective compared to prototyping using computer programming languages.

After I reached a consensus with participants on the interface and workflow designs using my exploratory prototype, I developed the exploratory prototype into an experimental web prototype with increased fidelity and operability for my participants. My major goal in this prototyping stage is to assess the technical feasibility of the system design by considering design issues such as data backend and concurrency control strategies. The prototypes were designed as web-based to facilitate distributed protocol writers to access and test it. I went through two major versions of the prototypes at this stage. One was to store protocols as text

files based on my field study results that "each protocol writing team only has one editor." This is the Single Scriber strategy (Posner and Baecker 1992), by which only one editor can revise the shared documents. However, my participants were encouraged by the collaboration support in the system designs and claimed that they were willing to give up the Single Scriber Strategy. Considering this user need and the challenges for concurrency control associated with a file-based backend, I upgraded the earlier design to one with a database as the backend.

Later, I extended the experimental prototype into an operable prototype by adding a number of features that were necessary to make the experimental prototype truly usable. These features also allowed me to carry out some usability evaluations. In Chapter Seven, I will provide the usability evaluation results.

Change Management through Negotiations

A big barrier in innovation deployment is user resistance to changes. Markus identified three categories of user resistance: user-centered, system-centered, and interactional (Markus 1983). Since participatory design emphasizes the user, I particularly watched for user-centered resistance, a resistance to innovation caused by a lack of knowledge or reluctance to change by the user. Within the SWOG setting, I tried to minimize this resistance through frequent negotiation of changes with users. I prepared a "*What would be changed*" document (see Appendix II: An Sample Change Management Document) prior to some participatory design sessions to demonstrate the potential value of the design and

to explain to users about the changes to their work practices. I also I elicited their feedback and discussed the tradeoffs between design ideas with them.

In summary, I held participatory design meetings once every two weeks and made design meeting notes to document the decision making process and the problems that arose. In this way, at the end of this research, captured design rationale for this research may help me analyze my design methodology.

3.2.3. Formative Evaluations

There was no prior collaborative clinical trial protocol writing system for comparison when I carried out this design; therefore, I adopted a self-reflective evaluation in this research and collected feedback from participants as the designs evolved. As Kaplan pointed out, well recognized-areas of system evaluation in the medical informatics literature include: (1) technical and system features that affect system use, (2) cost-benefit analysis, (3) user acceptance, and (4) patient outcomes (Kaplan 1997). Since (2) cost-benefit analysis and (4) patient outcomes are long-term research goals beyond my dissertation, I focused my evaluations on (1) usability of the designs and (3) user acceptance.

For evaluation research in medical informatics, Kaplan's suggestions are to (1) set up evaluation objectives from multiple perspectives and from the beginning of system design; and (2) use multiple evaluation methods (Kaplan 1997). Therefore, I used both qualitative and quantitative methods in my evaluations, including follow-up interviews, observations of system uses, focus-groups, and surveys. I

also used a four-staged evaluation plan, which included (1) system testing from a software engineering perspective (2) formative evaluation through simulated task role-playing or demonstrations (3) task-oriented, role-centered, and scenario-based user testing and (4) case studies in the field.

I scheduled formative evaluations with participants periodically, once every couple of months on average. At every meeting, I asked the participants to carry out the tasks of their roles for less than half an hour or demonstrate the relevant design features, and I elicited their feedback to refine the prototype design.

On this basis, I carried out one scenario-based study for the system. I asked the SWOG executive officers to help identify an on-going development protocol and ask the involved protocol writers to use the prototype system by carrying out real protocol writing tasks, such as collaborative protocol reviewing or comment incorporation, in an imagined scenario. This scenario-based study lasted two days involving one protocol coordinator, one study coordinator, and two statisticians. Participants performed tasks for their individual roles as well as group interactions. I observed how users completed key protocol writing tasks based on pre-designed scenarios and used focus-groups to elicit feedback from all the participants.

Later, I also carried out a case study for another development protocol and this study lasted nearly a week. A real protocol writing team and its PRC committee as well as its data coordinator, about eight volunteers was selected to participate

in the study. The focus of this study was to assess how the system may support real collaborative protocol reviewing tasks and the protocol coordinator was asked to incorporate comments online. I observed their PRC meetings and interviewed all the participants afterwards. I taped every evaluation interview or focus group study and made notes when I observed participants' interactions with the system. I also used a survey to ask participants to rate the major properties of the annotation model and the major system features.

3.3. Driving Research Questions

The dual-purpose goal of this work is to understand the collaborative clinical trial protocol writing process at SWOG and to support this process. Therefore, my work was centered on a set of research questions as described below. The answers to these questions can be found in the later chapters of this dissertation: Chapter Four for the fieldwork questions; Chapter Seven for the system evaluation questions and the participatory design questions.

3.3.1. Questions for the Fieldwork

RQ.1. What are the major steps in the protocol writing process?

RQ.2. Who is involved in each step and what are their job responsibilities?

RQ.3. How has email been used to facilitate group communication and how well does it work?

RQ.4. What is the most challenging part of the writing?

RQ.5. How do protocol writers coordinate the group work?

RQ.6. What is the protocol review and revision process like?

RQ.7. What are the views of the SWOG protocol development process held by protocol writers working in different roles?

RQ.8. What are the uses and problems revealed in review comments?

3.3.2. Questions for Participatory Design

I particularly paid attention to these two aspects during the participatory design:

RQ.9. What are the sources for the design ideas?

RQ.10. How do qualitative data from different sources complement one another in this research?

3.3.3. Questions for System Evaluation

Here is a list of usability research questions for the evaluations in this research:

RQ.11. How does the system design impact the users and how do they react to these impacts?

RQ.12. How do users rate the usefulness of the properties in the annotation model?

RQ.13. What are user behaviors concerning shared digital annotations during collaborative writing?

RQ.14. Do protocol writers working in different roles share the same tool support needs? Why or why not?

3.4. Data Analysis

Throughout the data analysis process, data accumulate over time. I continuously compared newly gathered data with previously collected data; in each comparison, I looked for similarities and differences, coherence and incoherence, and relative importance between old and new data. I have data from heterogeneous sources including email messages, comments, interview transcriptions, evaluation notes, and participatory design notes. Each type of data had its strength and/or weakness in revealing certain work process knowledge, and each needed slightly different analysis methods. In Table 3.1 I use a "data analysis matrix" to describe the concrete methods for each type of data in order to answer a set of research questions most suitable for that type of data.

Table 3.1 A data analysis matrix

Data Type	Analysis Methods	Research Questions
73 email messages from an observed protocol and about 50 emails from many other protocols	Content Analysis	RQ3
1140 protocol review comments	Content Analysis	RQ8
15 protocol development process interview transcripts/notes	Content Analysis Data Triangulation	RQ1 through RQ8
Participatory design meeting notes over the past > 2 years	Content Analysis Data Triangulation	RQ9 through RQ10
6 evaluation interview transcripts and 8 short surveys	Content Analysis Data Triangulation	RQ11 through RQ14

Content Analysis

The content analysis method is for protocol review comments and emails. Specifically, for comments, I analyzed these data at the semantic concept level based on ten categories of protocol problems from literature (Musen, Rohn et al. 1987). I coded the data for existence, not frequency. Email analysis was similar to comment analysis except for the differences in the coding books. I extracted extreme scenarios or critical incidences from interview transcripts, emails, or comments to explore complex problems in protocol development about group communication, group work coordination, or version control.

Data Triangulation

I applied triangulation, a process that combines multiple methods to study the same phenomenon, to study the qualitative data from this study. I triangulated interview transcripts to compare views of the SWOG protocol development process held by different working roles. Similarly, I triangulated evaluative results to compare expectations and feedback for the system design from the different working roles. Moreover, I triangulated all the qualitative data that I have collected including protocol comments, emails, interview transcripts, observation notes, and evaluation results to address RQ 10.

3.5. Summary

I applied this grounded design method to the Southwest Oncology Group (SWOG) to carry out both a field study and a system design. As illustrated in Figure 3.1, the qualitative fieldwork has led to an understanding about the SWOG protocol writing process and its communication complexities (Chapter Four). The prototype designs have evolved into the designs of an annotation model (Chapter Five) and a collaborative writing system (Chapter Six). Evaluations have led to an understanding of the usability of the design and the social-technical gap for healthcare groupware designs (Chapter Seven). A reflection of this research process has resulted in some empirical knowledge about how we could achieve a synergy between fieldwork and participatory design (Chapter Seven).

As explicated above, the philosophical stance that people and work practice must take preeminence in systems design requires that this work begin with detailed qualitative fieldwork, which leads to an understanding of the collaborative clinical trial protocol writing at SWOG. I will describe this understanding in the following Chapter.

Chapter 4: Understanding Protocol Development

There is little prior knowledge about the work process for clinical trial protocol development. In this chapter I present an understanding of the protocol development process at the Southwest Oncology Group (SWOG) with a focus on these three aspects: (1) characteristics of collaborative writing in a distributed clinical research organization; (2) collaborative writers' work practices during iterative reviews and revisions; and (3) loosely coupled collaboration among protocol writers with different expertise.

In the rest of this chapter, I first provide some background knowledge for clinical research and clinical trials in section 4.1. Then I briefly summarize the data collection method in section 4.2. Before describing the nature of collaborative clinical trial protocol writing, I describe the organizational facts and strategies at SWOG in section 4.3. Then I describe the major roles in a SWOG protocol development team and their work responsibilities in section 4.4. On this basis, I analyze the complexities in group coordination and communication in 4.5, and describe the iterative reviewing and revising process for protocol development in section 4.6. Later I summarize the group work support needs among clinical trial protocol writers exemplified at SWOG in 4.7.

4.1. About Clinical Trial Protocols

Well-designed clinical trials are the fastest and safest way to find treatments that work on humans. Clinical trial protocols are important documents used in medicine to support evidence-based medicine, which is "the conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients with specific sets of goals and treatment guidelines" (Sackett, Haynes et al. 1991). A clinical trial protocol describes a carefully planned study that evaluates the benefits and risks of treatments on the study participants. A protocol must be developed before a trial begins so that clinical researchers can follow the plan throughout the clinical trial study. Any error introduced during the protocol development process may propagate and magnify itself downstream in the life cycle of the clinical trial study.

Figure 4.1 shows the TOC of an example SWOG clinical trial protocol and lists all the standard sections in this protocol. As a rigorously structured document, a protocol includes information for participant eligibility, test schedules, treatment procedures, end point definitions, dosage adjustment rules, and a timeline for data submission. Therefore, a clinical trial protocol is a complex and long document. It is supposed to provide accurate details for every aspect of a clinical trial study. It is no easy task to produce this complex document. Clinical researchers with various expertise, including nurses, biostatisticians, radiologists, oncologists,

phamaceutical company representatives, and many others, often work as a group to create a clinical trial protocol.

SOUTHWEST ONCOLOGY GROUP
A PHASE II STUDY OF CHIMERISM-MEDIATED IMMUNOTHERAPY (CMI) USING NONMYELOABLATIVE ALLOGENEIC PERIPHERAL BLOOD STEM CELL TRANSPLANTATION IN OLDER PATIENTS WITH ACUTE MYELOID LEUKEMIA (AML) IN FIRST COMPLETE REMISSION
(A BMT STUDY)

Figure 4.1 TOC for an example SWOG Protocol

So far there has been no prior informatics system for supporting the group work aspect of protocol development. In fact, group work for clinical research has been an unspoken truth for many years. The National Cancer Institute (NCI) established the Clinical Trials Cooperative Group Program for clinical trial protocol development in 1955. Every year thousands of clinical trial protocols are created through the Cooperative Group Program. The Southwest Oncology Group

(SWOG) is one member of the Cooperative Group Program and is dedicated to cancer clinical trial protocol development by involving collaborative partners distributed all over the country. In addition, a lot of pharmaceutical clinical trial protocols are also authored through group work efforts. Therefore, a lot of research issues involved in the protocol development in Cooperative Groups also apply to the protocol development in pharmaceutical settings.

Besides the research value to the clinical domain, protocol development as a lengthy real-world collaborative writing task also has an important value for collaborative writing research. Descriptive knowledge about this group work may inform future medical informatics researchers, who are interested in designing technology to improve the protocol development process. This factor is a great motivation for my research described as follows.

4.2. Data Collection

In my work, I used a hybrid approach including participant observation, semi-structured interviews, and artifact collection and analysis to carry out the field work and to generate ethnographically informed research results in this work. I have described these methods in Chapter Three. Following the questions for fieldwork listed in Section 3.3.1, I designed a semi-structured interview question list, as shown in Appendix I: Questions in Semi-Structured Interviews. I first interviewed four protocol coordinators, four statisticians, three principle investigators, and one data coordinator, each with a one-hour semi-structured

interview. Later I informally interviewed some of the participants multiple times, depending on the clarity of their input and their accessibility, to obtain further input for the protocol development process at SWOG. In addition, I involved four of them in my participatory design process of the PCAT prototypes. I also participated in the Protocol Review Committee (PRC) meetings for more than three months. During this period, I collected and analyzed 1140 protocol review comments. I also monitored email communication within a protocol development team and collected about 150 emails from protocol writers. Some emails were from the protocol development team, and some were forwarded to me by the protocol developers in my study, to help convey the problems and challenges in the collaborative protocol writing process.

4.3. The Research Setting SWOG: Organizational Facts

Development of cancer clinical trial protocols involves extensive collaboration among experts from different disciplines. As one of the largest adult cancer clinical trial organizations in the world, SWOG has approximately 4,000 leading clinical experts, statisticians, protocol editors, and other cancer researchers distributed across the country. Figure 4.2 illustrated the organizational structure of SWOG.

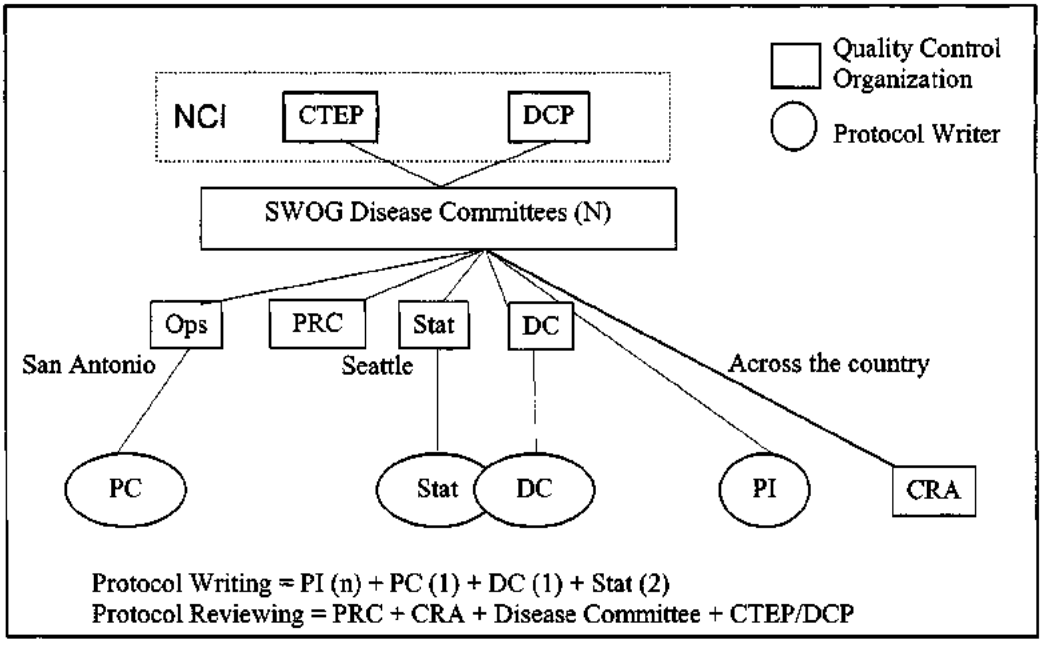


Figure 4.2 Organizational Structure of SWOG

SWOG reports to National Cancer Institute (NCI) directly, especially to its Clinical Trial Evaluation Program (CTEP) and Disease Control Program (DCP). SWOG has multiple distributed branches, including an operational office (Ops), a protocol review committee (PRC), a statistical center (Stat), and a number of disease committees (DC). The descriptions of these branches follow below:

1. **The operational office (Ops).** It is located in San Antonio, Texas and serves as a resource center for protocol writers. It also carries out protocol administrative tasks. There are six to seven full-time protocol coordinators in San Antonio. They are responsible for about 120 SWOG protocols at various stages every year.

2. **The statistical center (Stat).** It resides in Seattle, Washington and houses biostatisticians. They are responsible for designing and reviewing statistical considerations of clinical trial designs. Its second mission is to carry out data analyses for opened clinical trials and to publish trial results to related research communities.

3. **The data coordination center (DC).** It is also in Seattle. The mission of this center is to be responsible for supporting clinical trial development and collecting data from distributed sites after clinical trials are activated. Data coordinators provide input for patient eligibility criteria during trial designs and answer inquiries about patient eligibility criteria after trials are activated.

4. **Protocol Review Committee (PRC).** It belongs to the statistical center and carries out an official review for each protocol. Depending on the quality and the complexity of the protocol, this review happens once of twice during the protocol development process. The PRC committee consists of approximately ten biostatisticians, who often participate in protocol designs themselves.

5. **Disease Committees**. The Disease Committee is the liaison between SWOG and NCI. Every protocol goes through the review of its disease committee and then is submitted to CTEP or DCP for formal reviews at the national level. SWOG has about twelve disease committees that are in

charge of clinical trials for different types of cancers. The chairs of disease committees provide input or review feedback for the developing protocols of each committee. Under each Disease Committee there are numerous PIs and Clinical Research Associations (CRA).

SWOG's distributed organizational structure affects the protocol writing process in several ways, including email-based group communication, loosely-coupled group work, mixed peer reviews and hierarchical reviews. Next, I describe the characteristics of the protocol development group.

4.4. Characteristics of the Group

The composition of protocol development groups may affect the group writing process. In this section, I describe the major roles (Section 4.4.1), the formation of the group (Section 4.4.2), the diverse incentives of different roles (Section 4.4.3), and the loosely coupled collaboration within the group (Section 4.4.4).

4.4.1. The Major Roles

The comprehensive treatment plan described in a clinical trial protocol requires a wide range of knowledge. Therefore, protocol development requires collective expertise from multidisciplinary experts. To reduce the complexity entailed by a large group, SWOG adopts the small group strategy. In SWOG, a protocol writing team only includes four to nine members and it is assisted by a bigger group of quality control reviewers.

The writing team generates the substantial content of a protocol. The team is additionally supported by the PRC, drug company representatives, and CRA representatives; all these people provide consultation for the writing team and review the protocols. Each member in the writing group performs one of the following key roles: protocol coordinator (PC), study coordinator (SC), and statistician. Multiple people often serve the same role. Each role has unique expertise and is in charge of certain protocol development or review tasks as follows.

(1) Protocol Coordinator (PC)

SWOG maintains a number of permanent staff to assist with the writing of clinical trial protocols. One key staff member is the Protocol Coordinator (PC). PCs are considered as the gate keeper of protocols and are responsible for shepherding protocols through the protocol development process. A PC contributes to the development by writing sections, providing boilerplates, developing supporting forms (e.g., informed consent forms), and checking consistency among sections. The PC also sets the pace of collaboration and functions to coordinate much of the collaborative activity. PCs also schedule peer reviews with the PRC, as well as CTEP or DCP. PCs are responsible for collating feedback from the reviews, for getting relevant experts to respond to the feedback, and for incorporating the feedback to revise the protocol.

The PC is one of the key professional members dedicated to clinical trial protocol development. PCs usually do not have medical domain expertise; their expertise lies in boilerplate management, group communication and coordinator, word processing, and compilation of a large document from heterogeneous sources.

Here is an example from the "SWOG Clinical Trial Protocol Development Guideline" (SWOG 2004) with a list of responsibilities of a PC:

> The protocol development responsibilities of a Protocol Coordinator for the Southwest Oncology Group are:
>
> - To coordinate and assist in the development and activation of studies.
> - To assist the Study Coordinators by sending them information, answer questions and provide clarification regarding the protocol development process and group procedures.
> - Specifically, to coordinate each proposed study to activation, including:
>   - Putting proposed studies into a proper and acceptable format, and
>   - Making sure that all pertinent physician coordinators, statisticians and committee chair perform a detailed review of the study and incorporate comments and changes into the study.
> - To serve as a liaison between NCI and the Committee.
> - To ensure that the study is consistent in content and contains all the information that is required by the NCI and the Group.
> - To submit the study to the NCI for review and distribute information about the study (such as approval or disapproval) to the appropriate individuals.

As the pivotal person in the protocol development process, a PC is proficient with protocol development standards and formatting requirements provided by NCI. These frequently updated standards from NCI are to improve data comparability across trials carried out at multiple sites. Here is one example standard:

> 8.1 This study will utilize the CTCAE (NCI Common Terminology Criteria for Adverse Events) Version 3.0 for toxicity and Serious Adverse Event report. A copy of the CTCAE Version 3.0 can be downloaded from the CTEP home page (*http://ctep.cancer.gov*). All appropriate treatment areas should have access to a copy of the CTCAE Version 3.0.

A SWOG protocol has twenty sections; most sections have a set of standard wordings. These standards are updated at least twice a year to follow the changing policies of NCI. Therefore, in protocol development teams, PCs master these standards and are responsible for enforcing these standards in every SWOG development protocol.

(2) **Study Coordinator (SC) or Principle Investigator (PI)**

According to the protocol development guideline (SWOG 2004), the Study Coordinator (SC), who is also referred to as PI, is the primary advocate for a research idea and the primary decision maker in protocol development. SCs work for various organizations across the country and get affiliated with SWOG through various disease committees. SCs are responsible for answering questions regarding medical and scientific issues that arise during the conduct of the clinical trial experiment. SCs respond to requests from biostatisticians and PCs during protocol development, and later analyze the data in conjunction with the biostatistician to write the manuscript summarizing the results of the clinical trial study.

SCs often create the initial draft of a complete clinical trial protocol, although sometimes PCs do this for them when there have been a very similar protocol so that PCs can copy lots of content from existing protocols. SCs are in charge of writing the important sections in a protocol, including background or literature review, treatment plan, objective, and study calendar. They are also the people who have the right to approve all revisions made by Stats or PCs.

SCs all have a medical doctoral degree (MD), but have specialties in different medical subfields. Sometimes a clinical trial research idea requires multiple SCs with expertise in different areas to collaborate.

(3) **Biostatistician (Stat)**

Biostatisticians (Stats) are key personnel in the protocol development process because they review protocols and write substantial sections, such as statistical considerations and data collection forms. They carry out quality control for protocols and they are the people who eventually analyze data collected through clinical trials and write data analysis summaries for clinical trials. Well-designed protocols can help them to collect the right clinical data for ultimate research purposes. In this sense, statisticians are the ones who care protocol quality better than anyone else.

Most stats obtain master degrees or doctoral degrees in philosophy. Therefore, in protocol writing teams, they are one of the few people who are capable of directly communicating the deficits in clinical trial protocols to PIs. In contrast, PCs

usually do not have this power. I will describe the loosely couple group in Section 4.4.4.

(4) Data Coordinator (DC)

Data coordinator (DC) has been included in the protocol development process because of their important role in the later stage of the clinical research life cycle. DCs take patient registrations and answer phone calls when the protocol is unclear; they review data for quality control after it comes in. They conduct quality control for clinical data and make requests for missing data when biostatisticians conduct analyses. They participate in protocol writing by designing some variables such as response (based on tumor measurements), eligibility, deviations, etc. Their role in protocol review has been reduced due to the enormous amount of other work they have to do after trials are activated.

To summarize, PC, PI, Stat, and DC are the four roles that form the essential force for protocol development. Since they come from different backgrounds and have different tasks in protocol development, they have diverse incentives in this group work.

4.4.2. Group Formation

In SWOG, protocol developers are organized around disease committees. For a development protocol, sometimes PIs are assigned by the head of a committee; other times it is voluntary. The PI proposes a research idea for a trial and brings it

to the committee for approval to move forward. Stats are usually assigned to a committee or two and are usually the default statistician on trials of that committee. There are seven PCs in SWOG. Therefore, every protocol coordinator is in charge of trials for one or two disease committees. Protocol coordinators always work on multiple trials at the same time because the six to seven PCs help SWOG develop 120 developing protocols at various stages every year. Similar situations apply to DCs, so every DC is responsible for trials from one or two disease committees.

Among all the roles, there is more collaboration among PCs, DCs, and Stats than between these three roles and PIs. PCs, DCs, and Stats are all assigned to certain disease committees and continuously work on protocols of those disease committees. PIs are a dynamic population. A lot of PIs just write one protocol for SWOG in their whole career. They have looser commitments and connections to the disease committees compared to other roles for protocol development. Some PIs may never meet or never have prior familiarity with other protocol developers. In the eyes of PIs, the group work for SWOG protocol development is not based on a long-term, but a short-term collaboration.

In addition, very often the protocol development group is a mixture of both novices and experts and there is a disparity of expertise within the group. For example, many PIs are new to protocol development; they need a lot of training before and throughout their collaboration with others to design a new clinical trial protocol. SWOG suggests that every new PI should participate in a one-day

protocol development workshop to learn the process. But junior PIs still have to learn a lot along the way throughout the protocol development process. Sometimes, DCs, Stats, or PCs can also be novices too. There is a great need for group learning support during protocol development. However, existing technology available to SWOG protocol developers rarely addresses this need. A lot of process knowledge, such as discussion and decision making, has never been captured.

4.4.3. Vested Interests for Group Work

Protocol development is different from other collaborative writing processes in that not all participants get authorship during protocol development. The authorship of a clinical trial protocol and its subsequent research result reports consist of PIs and Stats. PCs and DCs are usually not counted as co-authors. Despite this fact, all people have vested interests in the quality of protocols to different degrees.

For PIs and Stats, an error-free protocol ensures that they get accurate and complete clinical data to test the research ideas. For the statistical center in general, working out logistics ahead of time prevents future headaches while monitoring the study. For the operational office, if there are multiple revisions to a protocol after it activates, it greatly increases the workload of PCs.

For all of the major roles for protocol development, if they do not get things right the first time, they will have to do more work down the road to fix the problems.

Unfortunately, the PI usually is out of reach after the protocol is completed and the clinical trial study is activated, and then the responsibility falls on the primary DC, the Stat and the PC, so there is more incentive for those individuals than the PI.

For PCs, PRC and CTEP, it is their job to review protocols to enforce some standards for protocol development and to provide quality control for protocols developed at SWOG. They do not have the incentive of more work when things go wrong.

4.4.4. Loosely Coupled Group

One type of collaborative work is called Loosely-Coupled collaboration (Haake and Wilson 1992), which is characterized as being asynchronous and autonomous with few dependencies among the work carried out by different people. Protocol development at SWOG falls into this category very well. The SWOG protocol development group consists of multidisciplinary experts, who collaborate over distances and across disciplinary cultures. These protocol writers are affiliated with various institutes, they often work autonomously and asynchronously. For most of them, protocol development is not their full-time job. Very often, most clinical trial designers such as PIs and biostatisticians have to fit protocol development into their busy work. Below are a few quotes from users at SWOG that indicates the dynamics of protocol writers' lives. All the names have been changed from the actually emails in the rest of this chapter.

> "I've got some raging fires that will take me until late next week
> to put out; after they've been doused, I'll put S0417 at the top of my list."
>
> "I'm hoping to start my review by the end of next week.
> I may not finish it until the middle of the following week. Busy times."

Therefore, protocol writers are loosely coupled with one another. This looseness leads to indirect communication or misunderstanding across roles in the protocol writing team.

Protocol writers usually do not have familiarity with other group members; therefore, they tend to rely on PCs for centralized group communication. For example, the statistician may have a question for the PI but asks the PC to relay this question instead of asking it directly. Below is an example email communication. In this example, Alice, Brian, and Cathy each represents a protocol developer. The email writer wanted to distribute a document to Brian and Cathy, but she did not contact them directly. Instead, she asked Alice, a protocol coordinator to do it for her.

> Hi Alice - attached is the NCI concept submission form with the information you requested. Please see my comments added to your email below. Will you forward this to Brian and Cathy, as there are parts which they each need to do, or would you like me to send it?

Indirect communication also happens because the different familiarity or different social statuses among protocol writers. Here is another example:

> Hi Angela...any input? Can you nudge Bartt about this? Cathy told me not to go any further on the protocol until the NCI concept submission was completed and reviewed. Did you ever talk to Doug about the Barriers meeting at SWOG?

In this example email, Ethon (who wrote this email) is an assistant for a PI Angela. Bart and Doug are medical doctors, and Cathy is a PC. Ethon asks Angela to nudge Bart and Doug. However, Ethon is situated at a different level from Angela, Bart, and Doug in the hierarchical healthcare system. In addition, the roles in the protocol development team have different authorities: PI has the highest authority, and then is the stat, and the PC has the least power over others. It would be inappropriate for Ethon to nudge other PIs. Angela is at the same level as Bart and Doug so she can make requests to these people.

Second, the role-based group work for protocol development and clear division of labor among multiple roles within the protocol development team make everyone works solely in their roles; they hardly have the opportunity to understand the work details of other roles. In addition, the wide geographical distribution of group members adds more challenges to this understanding. The visibility of the group work is low within the group during the protocol development process. I and my colleagues, David McDonald and John Gennari, once asked SWOG protocol developers working in different roles to draw a process diagram for the whole protocol development process (McDonald, Weng et al. 2004) and we found that no two diagrams contain the same set of steps. Protocol developers that we

have studied can describe their jobs but cannot provide details for other group members' work. I will elaborate on this observation in Section 4.5.2.

The lack of knowledge about collaborators affects the collaborative relationships among protocol writers. Below is a quote from a statistician that indicates the misunderstanding of some PIs about the collaboration within the group:

> Most PIs consider the biostatistician as a co-author, but not all. Those who view the stat as a service provider are often the ones whose request starts getting ignored after a while. I personally have learned to assert myself (politely of course) with those PIs so that they know this is a collaboration and that I do not follow orders, but am happy to consider requests and help out where I can.

Similar misunderstandings are more common between PIs and PCs. Some PIs are hands-off people who expect PCs to create a complete protocol based on their sketchy research ideas. They do not know how they should be involved in the writing process. But PCs often do not master the necessary scientific knowledge for protocols. Such misunderstandings often lead to delays in the group work.

4.5. Group Work for Protocol Development

In previous sections, I described the organizational context and the group characteristics. Next I describe the group work for protocol development and its workflow, strategies, typical problems, and the common collaboration patterns.

4.5.1. An Abandoned Workflow Model

To help protocol developers understand the complex protocol development process and to increase the visibility of the distributed group work, SWOG officers tried to use workflow modeling to communicate the major steps of this task to protocol developers. The workflow in Figure 4.3 shows the temporal relationships of thirty-five major time milestones during a normal protocol development process with branches and conditions. This diagram is also the most complete workflow model that SWOG has created. However, my interviewees at SWOG all told me that this model has never been in use.

In reality, protocol writers cannot strictly follow this workflow step by step when they write protocols. In addition, group work for protocol development is largely taken care of by protocol coordinators in a more flexible and reactive mode. As one interviewee pointed out, "*every protocol is a new case.*" Protocol development often has unpredictability. Protocol development process varies from group to group and from protocol to protocol. This workflow model is an ideal and rational design of the organization, but is reported to be impractical.

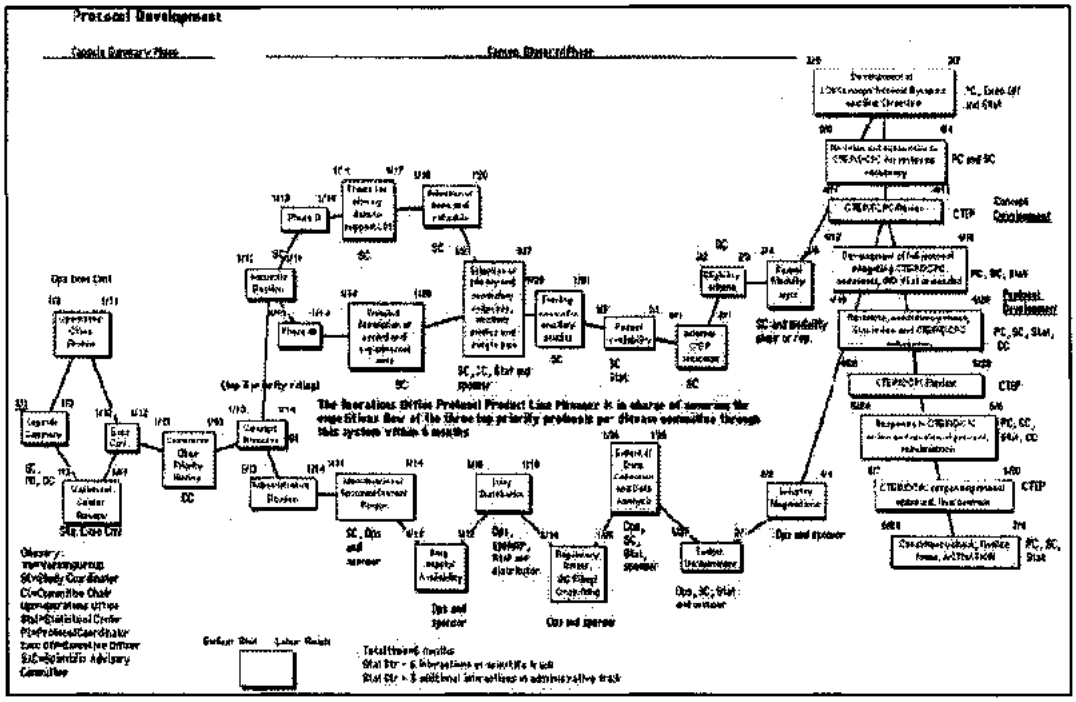


Figure 4.3 The SWOG Protocol Development Workflow

As an alternative solution, SWOG has provided a variety of guidelines to describe the protocol development to novice protocol developers, including (1) "How an idea becomes a protocol, a statistician's perspective" by the statistical center (SWOG 2004); (2) "SWOG Protocol Development Guideline" by the operational office (SWOG 2004); and (3) "SWOG protocol review guideline" by the SWOG Protocol Review Committee (PRC) (Green and Lowry 2004). These documents are more task-oriented, not workflow driven. They complement one another in putting together views of protocol development tasks from different perspectives and have been used as the "protocol development bible" in SWOG.

This study result indicates that workflow support might not work well for protocol development tasks. A good technology should consider the various needs of different roles in protocol development.

4.5.2. Multiple "Views" of the Group Work

During interviews, I asked the participants what they see the biggest challenge is for SWOG protocol development. Some said that there is no big challenge and the work is often smoothly done; while others said that the process is tedious and problematic. Below are some samples from my interviews, which are quite different.

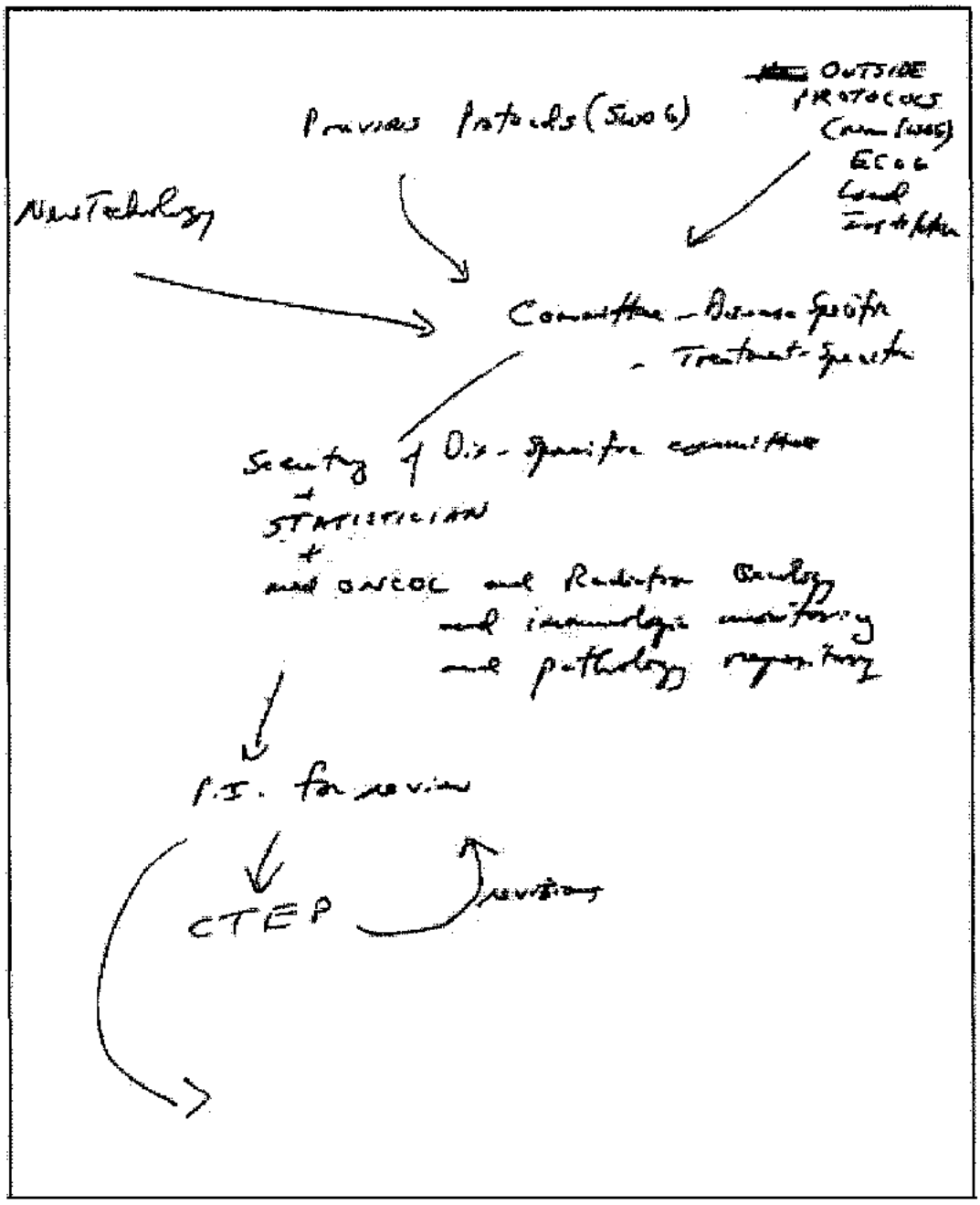


Figure 4.4 A View of an Experienced PI

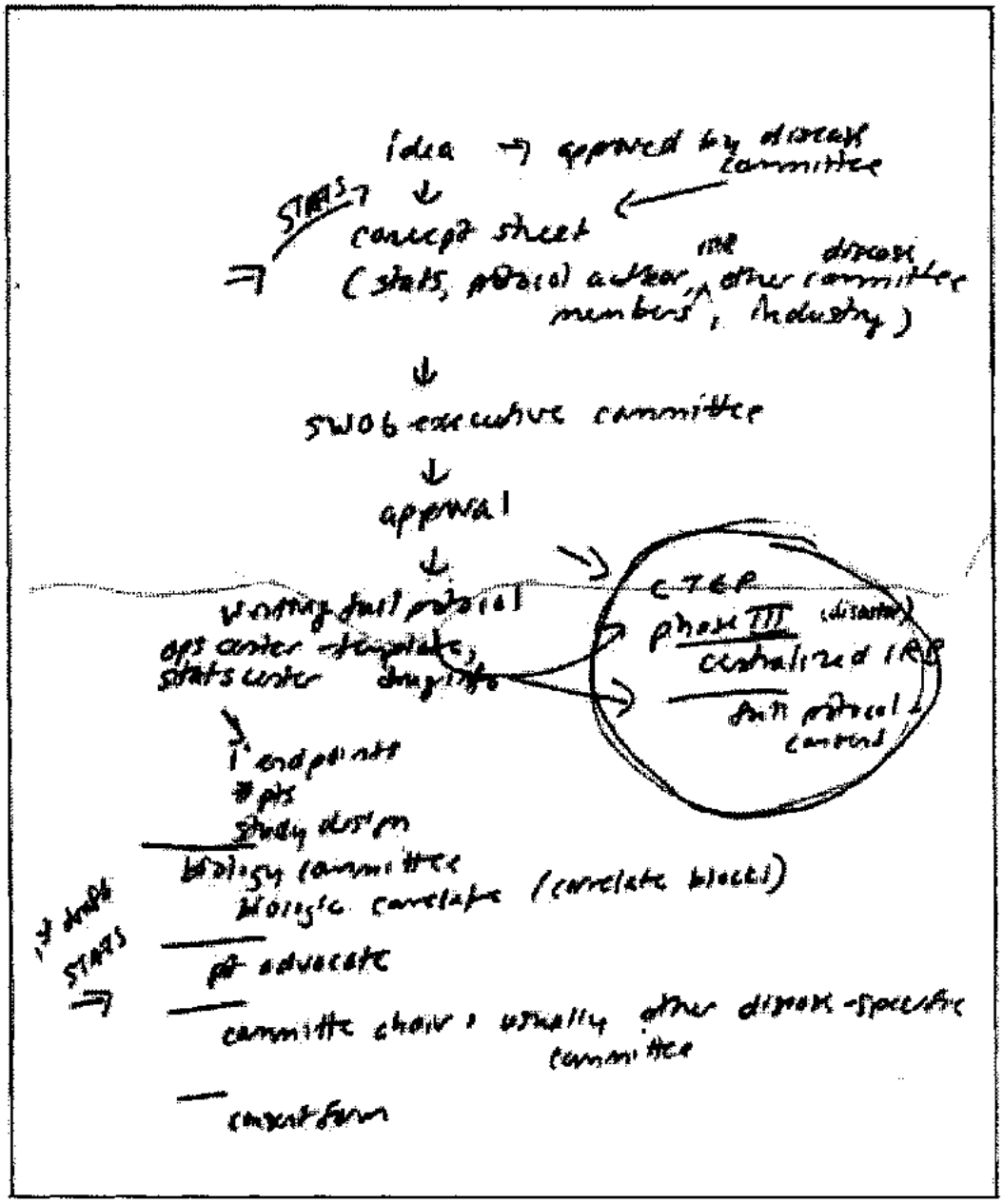


Figure 4.5 A View by an Experienced PI

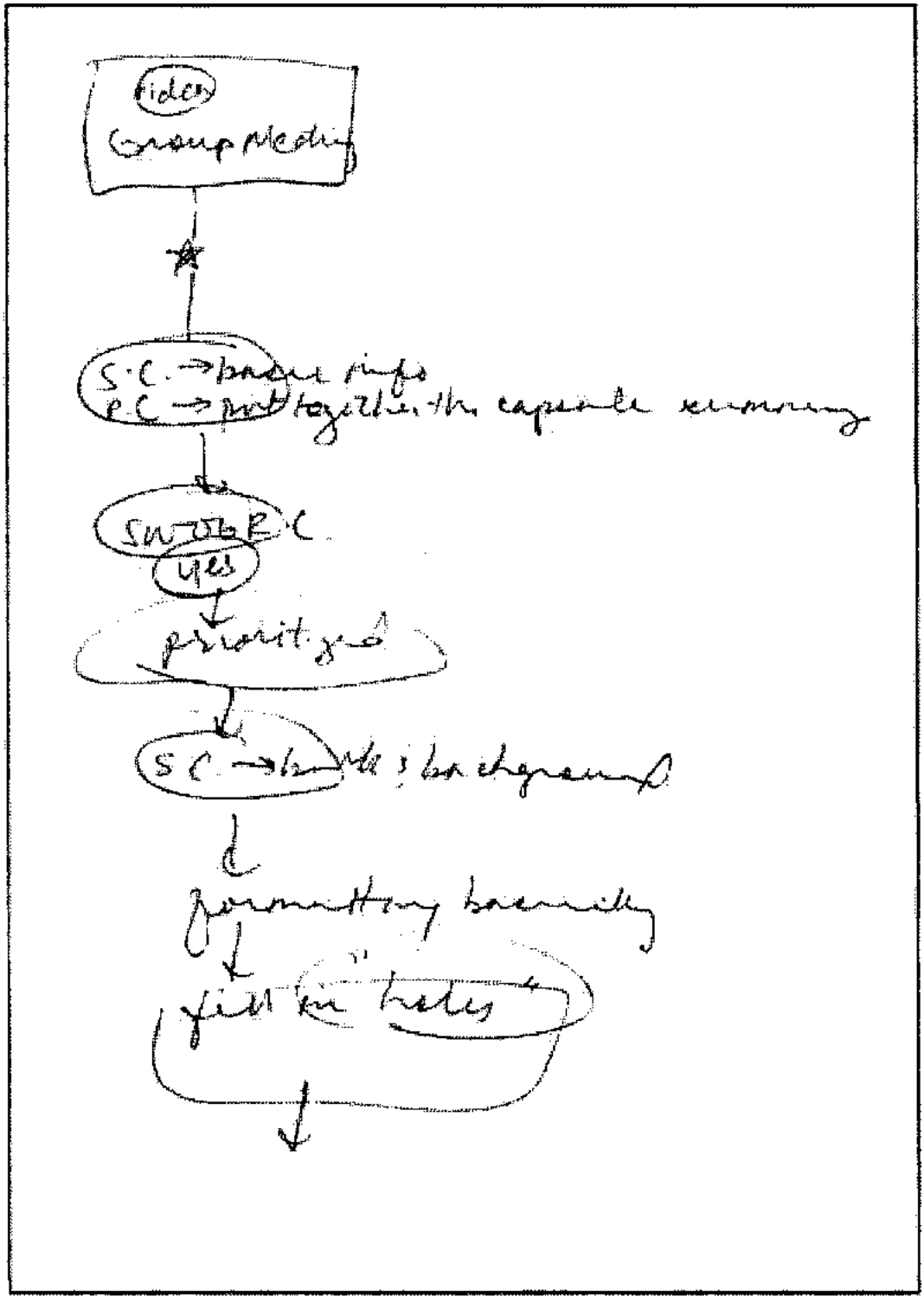


Figure 4.6 A View by a Novice PC

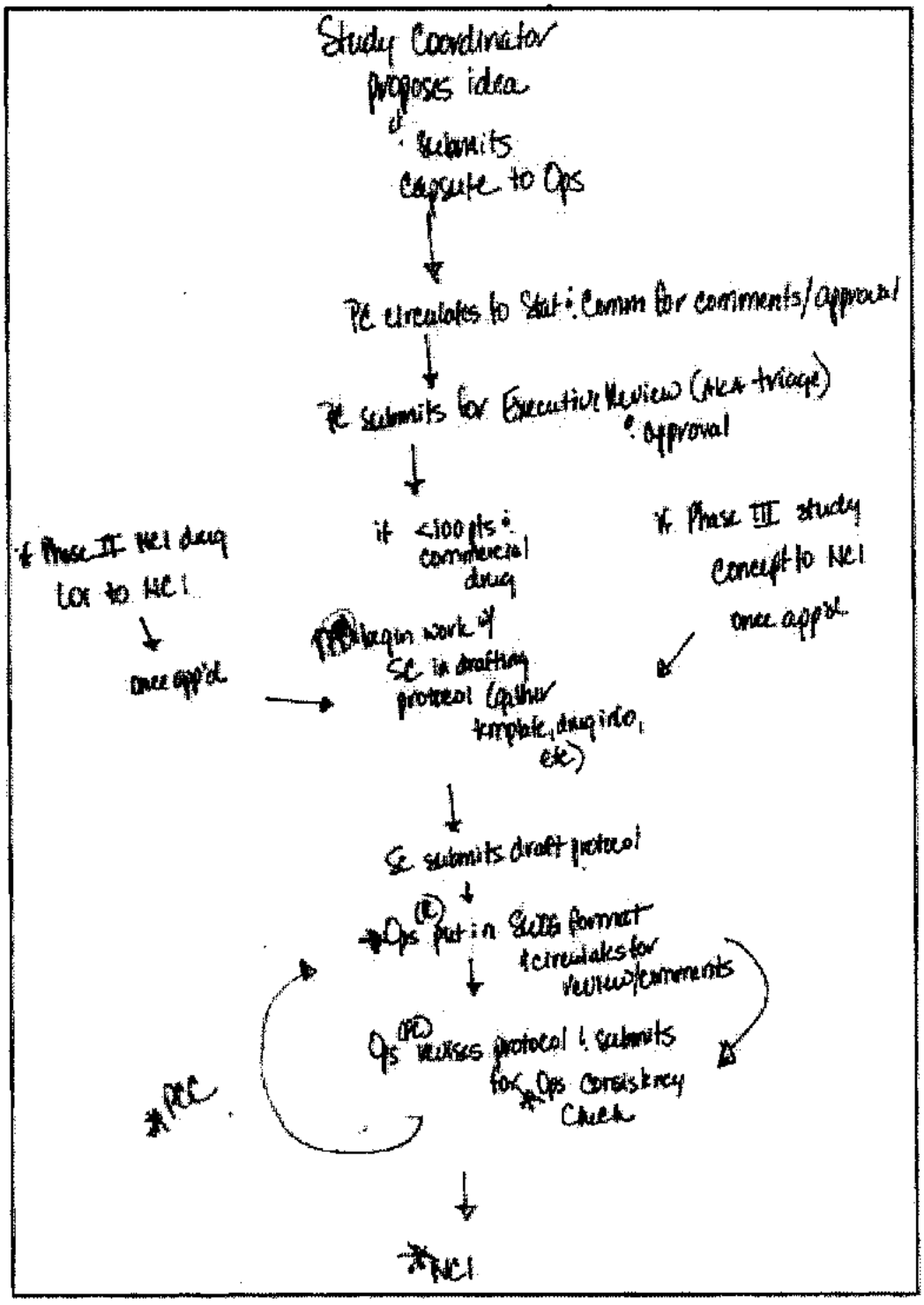


Figure 4.7 A View by an Experienced PC

In fact, not everyone in my fieldwork drew a diagram. Some data coordinators or biostatisticians working in PRC did not draw it. They asked me if "*this is a test.*" They seemed to be concerned about revealing their ability to capture a workflow precisely in a diagram and wondered what would happen if their view is different from some standard workflow model. Those who did not provide such a diagram ended up describing the sequential steps in protocol development.

The above sample diagrams look different and provide different details for the group work. There could be multiple reasons for the differences among these diagrams. First, the low visibility of the work among group members may contribute to the lack of knowledge among protocol developers. Second, the diversity in the personalities of interviewees in this study may affect the shapes and the details in the diagrams. Some interviewees are detail oriented; some interviewees are focused on high level issues and tend to draw sketchy diagrams. Third, it is possible that protocol developers working in different roles hold disparate views of the protocol development process. From a situated action perspective (Suchman 1987), individuals come to some shared understanding, but that understanding is bound in one specific instance of interaction. Since this difference does not interfere with the group work, a possible explanation is that maybe there is no need for the process views to converge.

4.5.3. Group Work Strategies

Currently SWOG does not have appropriate group writing technology support. Therefore, to write complex clinical trial protocols, SWOG adopts some well-known writing strategies. Below I describe their email-based communication, centralized resource and document management, as well as strategies for version control and group coordination.

(1) Email-based Communication

Email is the major group work and communication tool among widely distributed protocol writers. Protocol writers use email to share evolving protocol drafts, review comments, and changing protocol development standards. They also use email to coordinate group work, such as tracking progress, compiling review comments, etc. When I asked the study participants to compare different communication methods, such as phone calls, emails, and face-to-face meetings, they told me that email is the most widely used and they like it. Here are two quotes from the participants.

> A: Most of the communication has been by email, some through discussion at SWOG meetings. It is a combination of direct communication with the study coordinator, communication with the protocol coordinator, and broader communication to several kev members of the committee.
>
> B: We like email communication better than telephone calls or face-to-face meetings. It is very easy to track who said what in emails. It also allows more flexibility for you to respond to.

Communicating via email entails much meta-communication (communication about communication) since protocol writers have to send out emails to ask questions like "*Why is the protocol written in this way*" and "*Have you received my comments?*" when they are not aided by shared background group awareness or design rationale capture. Below is an example of an email message that shows both how a protocol coordinator explains a new version and has difficulties in getting prompt responses from participants:

> "Hello Alice, attached below is the latest version of the protocol. I am also forwarding the suggestions made by the statistical center that I have already incorporated. I never heard from Brian. Please feel free to call/email with further comments and suggestions, Chris."

(2) Centralized Group Coordination

A PC is the hub of information and resources during protocol development. A PC distributes available protocol drafts to a protocol development group, and then collects and integrates either protocol sections or comments from everyone. The PC is in charge of all the versions created for an evolving protocol and also keeps all old drafts for that protocol to provide subsequent consultation. Therefore, in SWOG, all group communication and coordination is centered on protocol coordinators.

(3) Combined Single Scriber Strategy and Separate Writers Strategy

The protocol coordinator is the key person who compiles a protocol. Posner and Baecker described a number of different writing strategies, including the Single writer scribe strategy, the Separate writers strategy, the Joint writing strategy, and the Consulted strategy (Posner and Baecker 1992). The protocol development actually uses two of them at different stages of the process. At the earlier stage of the protocol development, biostatisticians, PIs, and the protocol coordinator each write their sections separately. This is called the Separate Writers strategy by Posner. In this strategy, individuals break up the document into parts with each one responsible for one part. After the initial draft is created, the Single Scriber strategy is adopted. From then on, only the protocol coordinator can revise the protocol; all others provide suggestions or comments. Therefore, protocol writers do not carry out collaborative writing 100% of the time. Instead, they divide the writing amongst the group; they use parallel and separate writing first and collaborative reviews later.

(4) Using Both Hard Copies and Digital Copies

As SWOG protocol coordinators reported, at every single moment, they only keep the latest version of a protocol electronically, and all previous versions are saved as hard copies. They do not use version control tools. In addition, there is no widely adopted document management system available on market. Protocol developers use a preliminary version control method: writing the versioning date on the cover page of protocol documents, in both electronic and hard copies. Since protocol coordinators have to work with multiple protocols concurrently in

their daily work, current version control support is insufficient for them. As I mentioned above, only protocol coordinators can revise and generate new versions for developing protocols. They also have to manage all group communication by coordinating questions and answers as well as asking for clarification or further explanations. They are so busy that they can only create two or three versions of a protocol during the whole lengthy protocol development process.

(5) Multiple Document Formats

One challenge in collaborative writing is to authorize the editing right to different group members. Without group editing technology support, SWOG currently uses different file formats at different stages of protocol development to satisfy their need.

At the beginning of the protocol development process, all are allowed to write in Microsoft Word and send their text to the PC in Word documents. The PC compiles such input and creates the initial draft in Word and distributes it to everyone. After iterative reviews and revisions, the PC start using PDF files to distribute protocol drafts to everyone else. From this point on, everyone who wants to comment the protocol has to write a list of comments on a separate document from the PDF protocol file.

This work practice is not a perfect solution, and it requires reviewers to specify clearly which part of the protocol their comments refer to. This can be a very tedious process. As an example, to suggest a change of text, a reviewer wrote:

> "...section two, second paragraph, first sentence, suggest 'over the use of single agents' instead of 'over the sequential use of them' since I decided against a crossover."

Before resorting to multiple formats at different stages during protocol development, SWOG tried to suggest protocol writers using the "track changes" feature of Microsoft Word among protocol developers and it did not work because protocol writers did not follow this request consistently. Some reviewers liked to comment directly on the Word document. When they provide revision suggestions, they even write directly on the document in Word, and send back the changed Word file to the PC. But others liked to write comments in a separate document. Therefore, when the protocol coordinator gets feedback from various reviewers, it is a challenging job to separate comments from content and to merge heterogeneous information sources.

In sum, SWOG protocol developers facilitate group work by using a limited number of drafts, read-only formats, centralized document control, and multiple document formats at different time. Despite all these efforts, the iterative reviewing and revising process remains the most problematic during protocol development, but it is also a very important process for protocol quality control

and improvement. Next I will focus on this process and describe my understandings.

4.6. Comments and Iterative Reviews and Revisions

All clinical trial protocols have to be peer reviewed and meet all requirements specified by the Food and Drug Administration (FDA). To ensure the quality of the document, every development protocol goes through the following formal reviews within SWOG: (1) reviews by PRC; (2) reviews by the chair of the disease committee; (3) reviews by the Disease Control Program (DCP) or Clinical Trial Evaluation Program (CTEP). In other words, every protocol goes through both peer reviews within the protocol development team and hierarchical reviews within SWOG. The iterative reviews and revisions of developing protocols remain a big challenge for collaborative protocol writers.

During the lengthy collaborative protocol reviewing process, data collection is challenging because data are scattered in a large space and over long time periods. Protocol writers email the group with input for, or comments on, specific sections of the protocol draft, and the protocol coordinator in return redistributes a new version of the document. Because new drafts can be sent out before receiving feedback from all team members, it is often unclear to which version a comment is referring. Protocol reviewers also have to specify very precisely which part of the document they are commenting on.

Comments have been reported by the participants in this study to be central to all group writing activities. Comments also drive the evolution of the lengthy protocol document. Below I describe the major categories of comments that identify the most frequent problems in protocols, the major functions of comments for group writing, and the typical challenges for using comments among distributed group editors and reviewers.

4.6.1. Categories of Comments

For a long time, clinical and informatics researchers have been interested in identifying the major problems in clinical trial protocols. Knowledge about these problems can help develop better protocol development tools. Musen studied some clinical trial protocols manually and identified some recurring problems in clinical trial protocols such as ambiguity, inconsistency, and redundancy (Musen, Rohn et al. 1987). However, I did not find later extensive work. In my research, I took a different perspective: rather than examining developed protocols, I focused on review comments for developing protocols and identified the problems in developing protocols identified by reviewers.

To develop this categorizing system, I collaborated with two other faculty members and ran through statistical analyses of 1140 comments collected through PRC meetings. I first classified 1140 comments and noticed some patterns in the comments. Then I proposed a list of categories and used it to code the 1140 comments again. After I got a reliable coding system, the other two faculty

members and I refined the categories and tested the validity and accuracy our categorization schema. We randomly selected 170 comments out of the comments pool and categorized them among three of us. Two people have prior experiences with SWOG protocol review committee meetings and prior knowledge of the comments for SWOG clinical trial protocols (A and B); one person (C) has none of such experiences. Out of 170 test comments, we achieved a 3-way consensus on 134 comments, which is 78.8%; we also achieved a 2-way consensus between A and B of 89.8%, or 152 comments.

We identified six major categories of comments. Here is the coding schema that we have developed and used for further statistical analysis; each is with an example comment.

1. **Incompleteness**: lacking a part of parts or missing necessary information. For example, "*In Eligibility Criteria section, missing data input lines or conditions in criterion. For example, "please add accepted ___ declined - ___.*"

2. **Ambiguity**: descriptions being able to be understood in more than one way for people with different contextual knowledge. For example, "*This seems confusing. Should patients continue on treatment regardless of disease status, or should they be removed for progression? One or the other, please!*"

3. **Inconsistency**: no agreement or harmony in parts of different things. For example, "*This table reverts to using %'s instead of the 0, -1, -2 dose level reductions specified everywhere else.*"

4. **Inaccurate reference**: this category includes two types of problems. One is referring to outdated standards. For example, "*Third paragraph states that it is okay to have dose escalate back to the original dose, but section 8.2 states no dose re-escalations. I think this is just a matter of the standard GCSF template needs to be adjusted for this protocol.*" The other is making a reference to a non-existent, inappropriate, or a mismatched section in the protocol, for example, "*Unmatched reference: Sec 12.2, change section 14.0 to 14.7.*"

5. **Uncertainty or Questions**: posting questions and requesting the author to provide more details for analysis needs. For example, "*Is it well understood that GFR is calculated creatinine clearance, or could this be made clearer? What is the timing for the TURBT if the patient receives fewer than three cycles of chemo?*"

6. **Revision suggestions**: suggesting how to revise a part of a clinical trial protocol. For example, "*Suggest adding the word "Any..." to the beginning of this sentence. That helps make it clear that this includes delays not related to toxicities.*" Typical revision suggestions include insertion, deletion, and rewording.

Using the above categorization scheme, I analyzed the characteristics of comments on selected research questions; each question is followed by a figure with results.

(1) How many comments does a typical protocol receive from PRC reviews?

Table 4.1 shows an analysis of 23 protocols. The average number of PRC comments for a typical protocol is 41.

Table 4.1 Number of comments of each protocol

Protocol ID	Total Comments from PRC
S0314	105
S0226	86
S0221	82
S0221	74
S0217	72
S0125	67
S0219	59
S0011	55
S9927	49
S0222	47
S0218	44
S0327	41
S0216	37
S0129	37
S0211	36
S0207	35
S0219	34
S0224	31
S0312	27
S0121	26
S0025	16
S0232	12
S0227	11
Standard Deviation	24.62
Mean	47.08695652

In Section 7.2, I will compare this with the number of annotations generated using the PCAT prototype system and show how the online review system PCAT elicited more annotations.

(2) How frequently do comments occur in different sections of a typical clinical trial protocol?

According to Figure 4.8 the most problematic sections are ordered as Adverse Events, Eligibility Criteria, Study Calendar, Treatment Plan, Data Submission Schedule, and so on. Therefore, these sections are worthy of more attention. I showed this analysis to the participants in this work, and they confirmed that this result is consistent with their experiences. For example, the section of "Adverse Events" specifies dosage modification rules when adverse events occur during clinical trials; this section includes lots of intricate variables and rules. In addition, the sections at the top of the list in Figure 4.8 are also considered the most challenging sections during protocol development; this statistical analysis result matched the interview results very well.

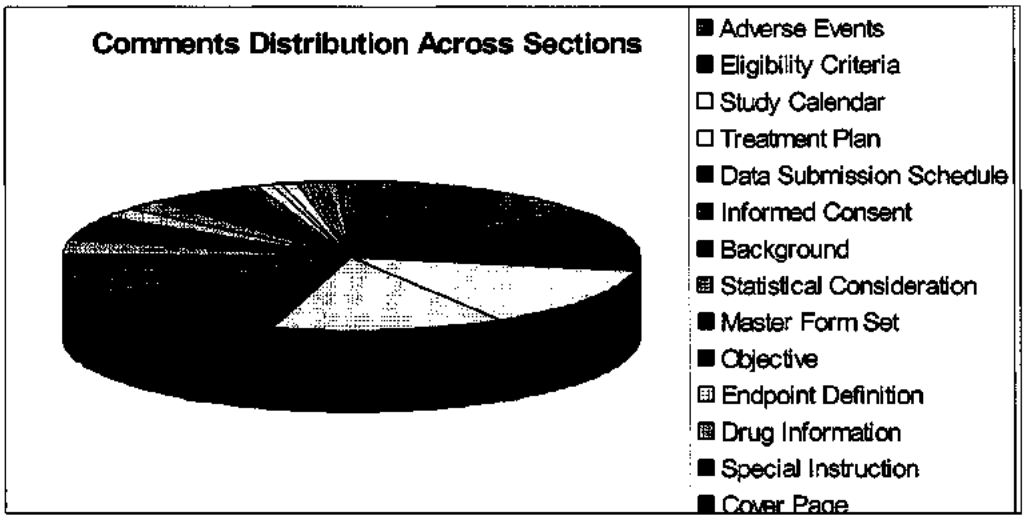


Figure 4.8 Comments Distribution across Sections

(3) How frequent do comments fall into the major categories?

Figure 4.9 shows that a great portion of review comments are revision suggestions. These comments may suggest inserting, deleting, or modifying existing text in a protocol document. The second biggest category is uncertainty, by which comments are written as questions that protocol reviewers post for other reviewers or the editor. Inconsistency, incompleteness, and ambiguity are also comment problems to protocols. This result confirms Musen's analysis of common problems in clinical trial protocols (Musen, Rohn et al. 1987). That analysis manually examined the errors in more than a dozen clinical trial protocols.

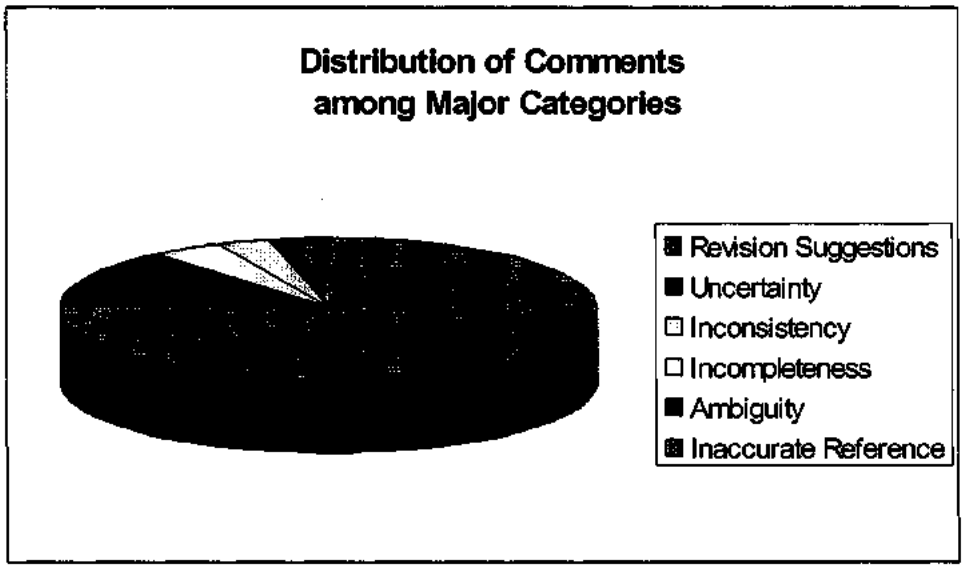


Figure 4.9 Distribution of Comments among Major Categories

Why are there so many questions in review comments? By carefully reading these questions and talking to protocol writers, I identified three possible reasons. One possible reason could be that reviewers feel uncertain of their comments, especially when they do not have expertise on that topic. They also feel something wrong with the protocol text and therefore raise questions to draw the attention of some people. The second possible reason is that protocols themselves have lots of ambiguities; reviewers use a question mark to suggest group discussions on this topic. The third possible reason is that reviewers are aware that some "experts" (prestigious doctors, for example) are wrong, but they try to be gentle and use polite language to point out the problem. Therefore, they use a question, which often sounds negotiable, to draw attention from those experts.

In addition, protocol writers told me that not all review comments would be eventually incorporated; some of them would be resolved after discussion as irrelevant or bad opinions. Comments are middle products of the review process; they entail a lot of group activities such as discussion, negotiation, and selective incorporation. Based on this analysis, I used the following "review and revision" model to highlight the major tasks in protocol development in Figure 4.10. In contrast to the complex workflow that SWOG provides in Figure 4.3, this diagram shows that iterative reviews and revisions are essential to the protocol development work.

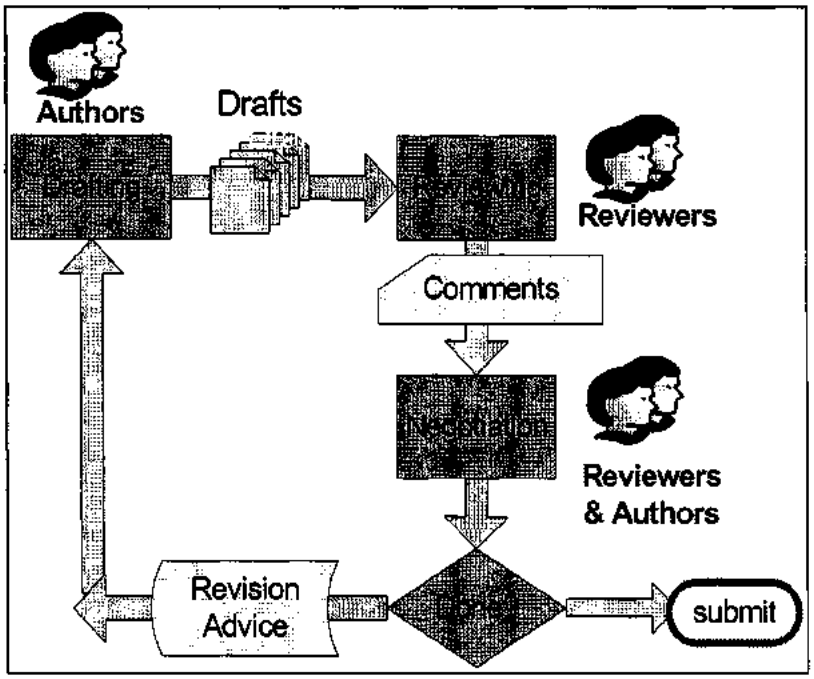


Figure 4.10 A Model of Iterative Review and Revisions

4.6.2. Roles of Comments in Iterative Reviews and Revisions

Commenting practices have received lots of attentions (Farkas 1987; Marshall 1997; Kim and Eklundh 2001; Wolfe and Neuwirth 2001; Kim 2002). Marshall examined the marking practices of university students on text books and identified rich functions of physical annotations (Marshall 1997), including procedural signaling for future attention, place marking and memory aiding, in-situ locations for problem-working, interpretation, tracing progress through difficult narrative, and incidental reflection on the reading material. Kim explored the dialogues formed by comments exchanged among co-authors in a web-based collaboration tool and called for support of persistent dialogue and comment

categorization in collaborative writing activities (Kim 2002). Comments are considered as important communication artifacts among collaborative writers. I extended previous research and studied how comments have been used by collaborative clinical trial protocol writers. The major roles of comments are as follows:

(1) Group Communication Artifacts

During the protocol development process, a lot of protocol writers never meet one another except by exchanging comments. Comments are essential communication artifacts in the group for multiple reasons. Protocol development starts from an initial draft created by a PI or a PC, and then evolves with iterative reviews and revisions. Except for the person who can edit the protocol, most participants are reviewers. These reviewers participate by giving comments in emails to provide their input or feedback. Therefore, commenting is the only job for most participants. Even when some people want to revise the evolving protocol, they tend to give revision suggestions in comments rather than making changes themselves to avoid possible version control complexities. As described in section 4.6.2, comments convey questions or problems in protocols and are communicated within the group during collaborative writing. Very often comments develop into dialogue among protocol writers, especially when there are conflicting opinions among multiple reviewers. Here is an example from my study.

A: Hi A! We are struggling to put together a standard definition of current, former and never smoker for use across the lung protocols in SWOG. It occurred to us that there might be an agreed-upon NCI definition or even a common data element. The best definition we have is below. Would you know about this?

B: There are no CDEs (Common Data Elements) for smoking status. I would have you query Ellen Grits at MDACC to see if there is common terminology within the behavioral research arena. However, the standard definitions, as we use them in DCP are:

Former - no smoking for 1 year or more

Never - less than 100 cigarettes lifetime

Current - everyone else

C: I brought this up to Dr. Gandara when we added prior to submission. He said he wanted to leave it up to insts. Perhaps you should run this by him...

D: A definition was requested by CTEP in comment #12 and it is important to clearly (and simply) define this variable. There was a good deal of back and forth regarding this a couple weeks ago and I believe Dr. Gandara and Dr. Albain were a part of it initially with respect to the fact that we need a definition. This is the standard definition that we have come up with to use on all new SWOG lung studies. We'll also be referencing it in the CRA manual. Maybe Kari or John kept the string of e-mails and can concur the DRG was in the loop. Maybe it was only Dr. ALbain. I can't remember.

In the above example, the protocol developers hold different opinions about the definition for "current, former, and never" smokers. They comment on others' comments and carried out a discussion in the group. This case is very common. There are often lots of negotiation and discussion around different opinions; in this process, some comments are decided as "irrelevant comments" and resolved, and other comments are some ideas that make sense. Later the protocol editor will incorporate ideas from those relevant comments and revise the protocol accordingly. Comments as group communication artifacts are circulated among

the group writers. They are great resources for subsequent group activities such as group learning.

(2) Group Learning Channel

As described in section 4.4.4, a protocol writing group often consists of both novices and experts for protocol development. These protocol writers learn from one another through the protocol development process. Comments generated in the review process often convey expert advice about what the standards are and what the norms are for similar protocols. Well written comments can also help novice protocol developers learn how to phrase a question. Discourses carried out through comments are lively educational resources for new protocol developers. Very often junior protocol developers ask other senior protocol developers specific questions; comments attached to certain parts of a protocol enable learning to happen precisely within the context.

(3) Activity Justification Artifacts

During collaborative writing, comments are middle products that provide grounding facts for a lot of group activities such as group discussion, draft revisions, and subsequent consultation. Since a protocol involves a large group of participants, it is difficult to keep everyone on the same page at the same pace. When group writers or reviewers hold different opinions, the editor has to selectively incorporate comments when making changes to developing protocols. Editors rely on some comments to justify their appropriate changes to developing

protocols. It is important to manage well those comments that are discarded or left out in subsequent revisions, because the reviewers who give these comments care how their opinions are received. In addition, for those comments which will be incorporated, it is also important for the editor to keep the comments because the reviewers may change their minds afterwards. Recurring comments from multiple reviewers is not usual during protocol development; in addition, sometimes editors have to overwrite a change to rollback to previous versions. Without keeping those comments that lead to changes, editors would have great difficulty in consolidating diverse opinions from multiple reviewers.

During my study of comments, I found that the biggest category of comments is "revision suggestion." A clinical trial protocol improves while review comments are incorporated. Therefore, these incorporated comments carry the change rationale across different versions of a clinical trial protocol. Conventional version control software has a feature called "Change Representation". In Microsoft Word, this feature is designed to show old text using "Strikethrough formatting". According to some earlier user studies, this interface is actually distracting. The users of this study at SWOG also have a similar complain. In addition, they said they care more about how review comments are received by editors, but currently no tools can help them track how changes in the evolving documents are related to review feedback.

Moreover, one protocol coordinator pointed out, that they once developed a protocol and made a change based on a review comment from NCI. This

comment was sent to an individual protocol developer as an email message. However, six month later, someone from NCI came back to the operational office and asked why the protocol was changed in that way. It was difficult for the operational office to find the email. Comments communicated between NCI and the operational office should be archived as organizational resources for subsequent consultation; however, currently they are all kept by individual protocol developers.

(4) Progress Tracking Artifacts

Comments effectively link both reviewers and editors because they point out problems in old text, and give suggestions about how to revise the old text. Protocol coordinators have to manage heterogeneous information during collaborative writing. In this process, some protocol coordinators develop a progress tracking document around comments they received. In this document, they record how each comment has been incorporated or resolved, or note that some comments are still pending further information. This document usually is used privately; but during the final review stage with CTEP or DCP, some protocol coordinators submit this document as a report to CTEP and let them understand how all review comments are finally received. To illustrate such progress tracking documents, below I provide an example portion:

> Sec 5.8: Ineligible for hypermagnesium too? This was problem in S9451 when the intent was only to make patients with hypomagnesium ineligible. – **DR. YOO HAS STATED THAT A PATIENT WILL BE INELIGBLE FOR HYPERMAGNESIUM ONLY. ELIGIBILITY 5.8 HAS BEEN CHANGED TO READ "...SERUM MAGNESIUM ≤ INSTITUTIONAL LIMITS OF NORMAL..."**
>
> 5.12: add pregnancy test? (don't forget 9.1 too) – **WE DON'T TYPICALLY REQUIRE INSTITUTIONS TO PERFORM A PREGNANCY TEST. WE ALLOW INSTITUTIONS TO UTILIZE THEIR INSTITUTIONAL GUIDELINES FOR MEETING THIS ELIGIBILITY CRITERIA.**

In this example, the protocol coordinator manually wrote down what he or she did for each comment. Special formatting was adopted to distinguish an old comment and the progress comment on an old comment. According to my interview results, such documents are considered effective for tracking progress in iterative reviews and revisions. However, this is no better tool that supports this activity.

(5) Organizational Memory

Organizational memory refers to stored information from an organization that can be brought to bear on present decisions (WALSH and UNGSON 1991). Organizational memory is often missing when a large number of people engage in the design and construction of large, complex systems over long periods of time (Conklin 1996). The design rationale of large, complex systems is thoroughly and systematically lost. The consequence is that organizations have difficulty in transferring previous learning to current problems. Such phrases as "reinventing

the wheel", "going in circles", "having the same discussion over and over," are often heard in large engineering organizations.

In my study at SWOG, I found that organizational memory is also an outstanding problem. These complaints are not unusual in SWOG:

"Every protocol is a new case; training protocol developers is a big challenge."

"Sometimes we have to run into the same discussions over and over!"

Where does knowledge about protocol development reside? How can this knowledge be shared among different protocol developing individuals and among different protocol development teams? I kept these questions in mind when I studied the iterative reviews and revisions of protocols and the collected comments. I found that comments are valuable organizational memory because they convey reusable expert opinions and contain information for subsequent consultation when the operational office interacts with NCI or other organizations. Some comments are actually applicable to multiple protocols. The users in interviews pointed out that they have to repeat some comments when they review different protocols. In another word, they hope some comments could be shared among protocol writers.

4.7. Summary

My study of protocol development at SWOG revealed four outstanding problems in the current group work among collaborative clinical trial protocol writers.

First, group communication among protocol writers is insufficiently supported. Currently email is the primary communication tool for protocol developers, but it does not support in-context communication and information management for a group. Email also does not support group awareness and persistent conversations well. It is hard to retrieve past communication records since they are often buried in personal email boxes.

Second, collaborative reviewing and revising of protocols is complicated by the lack of support for version control and in-context communication. Although tools for version control, change tracking, or online discussions are available, during reviews and revisions, collaborative writers need an integrated environment that can support all of these three functions. Unfortunately, such technology barely exists.

Third, there exists a large amount of unnecessary meta-communication (communication about communication) during protocol development. This occurs when protocol writers have to manually communicate with the group about some communication artifacts created earlier, such as a comment or a draft. This indicates the problems of insufficient progress tracking support as well as insufficient group awareness.

Last, comments created in the collaborative reviewing process need better management. For protocol writers working in the same group, they need to share comments and carry out in-context discussions through related comments; they

also need support for persistent conversation support through archived comments for justifying their revision activities or for providing subsequent consultation. For protocol writers who do not work in the same group, they need to learn process knowledge from comments created during previous protocol development processes.

In summary, the clinical trial protocol writing process at SWOG is currently confronted with lengthy protocol review processes, poor version control support, inaccessible reusable knowledge, ineffective group coordination, and challenging heterogeneous information source integration process. All of the above problems could contribute to clinical trial protocol errors and delays. Currently, email systems, instant messengers, and other modern communication tools are readily available. However, these generic tools cannot by themselves provide the communication support for the practice of clinical trial protocol writing that I observed. Improvement of collaborative reviewing and revising requires a good integration of version control, design rationale capture, in-context communication support because all these of these parts are closely connected in an evolving information space.

SWOG has been a very successful cancer research organization. In spite of the problems in the current work practice, SWOG is still a productive protocol development organization. One important factor that leads to success if the efforts made by protocol writers at SWOG. Therefore, I have been motivated to focus my

research on augmenting the natural work process among distributed protocol writers.

Through this field study, I found that protocol review comments have been essential to this collaborative writing process. Currently there is no appropriate group work technology that supports effective comments sharing among reviewers and editors during iterative reviewing and revising of evolving documents. This factor motivated me to design a new annotation model for collaborative writing support, as I will describe in Chapter Five.

Chapter 5: An Annotation Model for Collaboration

In Chapter Four I described that comments are exchanged among collaborative writers as important artifacts with rich functionality. To support commenting behaviors, many annotation models and systems have been designed in recent years. However, existing annotation technology only works well for static documents and barely supports the above functionality in development documents. There is also no good management of digital annotations during the collaborative writing process. An important piece of my research is an annotation model for supporting iterative reviews and revisions during collaborative writing. Next, I start the description of this model with its design desiderata.

5.1. Design Desiderata

In the past decade, a number of annotation models have been designed. Conceptually, annotations have been well accepted as metadata, data about data. The representative annotation models are Dublin Core (Campbell 2002) and Resource Description Framework (RDF) (Brickley and Guha 2004). Dublin Core is designed to facilitate interoperability among diverse metadata sets; RDF is designed to support information integration among heterogeneous resources. Such annotation models are designed for developed documents. In contrast to prior work on annotations, I focus on annotation for dynamic, developing documents, especially during the collaborative writing process.

Digital annotation researchers have encountered many real world challenges for collaborative writing activities. Cadiz et. al. conducted a large case study of digital annotation uses among a group of software specification document writers (Cadiz, Gupta et al. 2000). He identified the following major problems that impede the uses of digital annotations during a collaborative writing process: (1) technical orphaning of annotations in evolving documents: the system fails to decide which text the annotation should be associated with when the document is changed; (2) users being unable to stay aware of changes; (3) users lacking responsiveness from one another; (4) lack of public nature of annotations; and (5) insufficient richness of annotations. Existing annotation frameworks or systems only support static annotations; they assume the objects that annotations are attached to never change. This assumption does not hold during collaborative writing because development documents evolve over time. Some annotations need to be re-anchored to the changing document while others become irrelevant after being incorporated or resolved.

In section 4.6.2, I described the important roles that comments play during collaborative writing, including (1) group communication artifacts (2) group learning channels (3) activity justification artifacts (4) progress tracking artifacts and (5) organizational memory. My main design objective is to empower digital annotations with the above functions that comments have and to overcome the challenges involved in serving these functions. To achieve this goal, below I describe the desiderata for an annotation model that supports collaborative writing

from two aspects: three desiderata for group activity support and four desiderata for user interface design needs.

5.1.1. Desiderata for Group Activity Support

1. **An annotation model should support in-context communication and decision-making among collaborative writers**

Although researchers have recognized the importance for annotations as communication artifacts (Brush 2002; Kim 2002), a big challenge is to support "in-context discussion" when documents are subject to changes and to avoid "orphaned annotations", which refer to the common problem that systems fail to find the appropriate text that an annotation should be attached to when the document is changed (Cadiz, Gupta et al. 2000; Brush, Bargeron et al. 2001). This desired feature requires an integrated version control for both documents and annotations.

In addition, annotations carry opinions of collaborative writers, which may be redundant or contradictory. Iterative reviews and revisions of development documents evolve through the debates or discussions among collaborative writers. It is helpful to support the representation and analysis of arguments, to support consolidating of annotations, and to capture the decisions and reasoning in this process. A solution to consolidate shared annotations should consider both social and technical issues involved.

2. An annotation model should strengthen feedback across roles

Annotations are central group activity artifacts, especially during collaborative reviewing and revising of the development document. Collaborative writers can read, reply, incorporate, or resolve annotations made by others. During the collaborative writing process, there are often multiple roles, including reviewer, author, and editor. People working in different roles have different division of labor; what they share are mostly annotations and the evolving document. Therefore, annotations play an important role in linking these different roles. Efficient group work relies on smooth transitions or interactions between activities of people working in various roles. There is a need for supporting timely cross-role feedback within the collaborative group. For example, reviewers may want to stay aware how their opinions are received and editors may want to inform reviewers how they incorporate review feedback to revise the shared document.

I also observed this need during my fieldwork at SWOG. Without cross-role feedback support, protocol coordinators at SWOG have to manually provide feedback to their group members when they incorporate a comment or create a new draft, either by writing an email or sending back meta-comments on existing review comments. A lot of efforts are spent to carry out unnecessary meta-communications within collaborative writing groups for development documents.

3. **An annotation model should support persistent conversations among collaborative writers**

In collaborative writing, annotations convey opinions of writers and include information about revision rationales. Moreover, over time, annotations form dialogues among collaborative writers (Sumner and Buckingham Shum 1998; Brush 2002; Kim 2002). These dialogues embed expert knowledge and record the idea formulation process for a development document. Erickson et. al. have pointed out the importance of supporting persistent conversations in work processes (Erickson, Smith et al. 1999). Successful management of persistent conversations during collaborative writing can enable subsequent consultation of writings, can capture reusable expert knowledge, and can provide change explanations for the shared and evolving document. Such persistent conversations also support organizational memory and design rationale capture.

Storing an annotation discussion can help users better understand the design rationale for the evolving document. This is the classic concept, design rational capture. It was proposed for software process management. Design rationale expresses elements of the reasoning which has been invested behind the design of an artifact (Lee and Lai 1991). There are two major types of design rationale (Garcia, Howard et al. 1993). The first type is argumentation based: the design rationale is primarily used to represent the arguments that define a design. These arguments consist of issues raised, alternative responses to these issues, and arguments for and against each alternative. The second type is history-based: the

rationale consists of the design history, the sequence of events that occurred while performing the design. This information can be stored in many forms. It could be in the form of entries in a design notebook, an archive of e-mail messages, or other types of documents that capture actions taken over time.

During collaborative writing, annotations serve as communication and discourse artifacts, and are also central to group activities such as replying and incorporation. Therefore, Annotations could be used to capture both types of design rationale for collaborative writers. Currently annotations are underused valuable resources that represent arguments and store revision rationales. Advancement in research on both design rationale capture and in-context discourse enhancement should be blended into the design of a new annotation model to support collaborative writing activities.

5.1.2. Desiderata for an Annotation Interface Design

Successful technology often requires a good human-computer interaction design. A usable annotation design should enable collaborative writers to access and manage annotations flexibly and easily, without interfering with the writing process. Therefore, I identified these desired features for an annotation interface.

1. **Annotations should be robustly anchored to documents**

Annotations should be precisely linked to specific areas in the document when the document goes through changes so users can make sense of the context of the

annotations. For this purpose, each annotation should provide context information about the document and the anchor in the document.

2. Annotations and documents should be mergeable and detachable

Annotations and documents should be stored separately so that annotations could be able to be linked to documents or removed from documents in a flexible way. This feature could also address the information overload problem by enabling users to selectively view a subset of annotations in documents. For this purpose, annotations should have detachable anchors to the document. Hypertext technology makes it possible to dynamically embed annotations as URL links into documents and to remove such links for generating an integrate document.

3. Annotations should be annotatable

Annotations are also information objects subject to debates or discussions; therefore, they should be annotatable. Annotated annotations can further be threaded according to their topics to support in-context communication. Therefore, the annotation model should store information about "the thread."

4. Annotations should be traceable

Annotations are not static objects. Group activities often take place after annotations are created. If annotation systems can capture group activities around annotations and provide shared-feedback awareness to users (Dourish and Bellotti 1992), users can easily trace work progress around annotations. One solution to

support this functionality is for an annotation model to store information about activities, such as an activity status.

Moreover, annotations during collaborative writing all expire at certain time. Reviewers or editors will not be interested in reading these annotations again after they incorporate or resolve them. In other words, annotations should have a life cycle, which consists of multiple activity statuses. Annotation systems should support life cycle management for annotations by tracing status transitions for them.

5.1.3. Summary

In sum, we need an annotation model that defines expressive and dynamic properties. An annotation object should include dynamic context and status properties. The anchor of an annotation is subject to changes. If the annotation is part of a discussion, the communication context may change while the discussion is evolving.

Moreover, an annotation in a design or writing process could have a life cycle including multiple statuses. Status information could help users distinguish completed annotations from unprocessed annotations. Status information could also indicate activities that happen to an annotation, such as a revision to the document or a response to the annotation. Moreover, collaborative writers can use status information to trace the annotation incorporation progress. Finally, status could facilitate annotation management. It is unnecessary to anchor every

annotation to a new draft during collaborative writing if the annotation has been incorporated. Therefore, status could help minimize the annotation overflow problem in an evolving design process or information space.

5.2. Design Specification

I created a new annotation model to satisfy the above desiderata. In Table 5.1 I describe the major properties and their details of this annotation model. Among them, the property of "context" and "responses" to support group communication; the property of "status" is to support life cycle management of annotations and shared feedback group awareness; the property of "response deadline" is to support annotation notification service.

Table 5.1 Properties and fields in my annotation model

Property	Details
Context	Document version id, Anchor text, Discussion thread pointer,
Commentary Message	Textual field
Annotation Creator	User id for the creator
Annotation Recipient	User ids for the recipients
Annotation Time	A time label
Response Deadline	A time label
Responses	Pointers to related annotations on the same topic
Status	Activity status, e.g., "Unread"
Category	Problem classification, e.g., "Inconsistency"

1. Context: A description of the object that an annotation is attached to.

Context information includes three properties. The first property is the document context, which specifies the version information for the annotation's document. The second property is the anchor within the document, which pinpoints the surrounding text for an annotation. The third property is the discussion context, which refers to previous annotations on the same topic. This information helps position an annotation in a threaded discussion and helps collaborative writers understand the conversational context.

4. Annotation Recipient

This feature records the recipient of the annotation. With this piece of information, automatic notification services can send reminders to the recipient.

6. Response Deadline

This feature specifies the time by which the annotation should be addressed. It could be used by notification services to facilitate the prioritization of annotations.

7. Responses

This is a multi-value feature, as an annotation can receive multiple responses. This information can be used to form threaded discussions.

8. Status

The transitions of annotation statuses enable collaborative writers to know what happens to a particular annotation. The status can indicate whether there are

responses to an annotation, whether annotations are incorporated in new versions, etc. Milestones in an annotation life cycle could vary from system to system depending on concrete tasks. For collaborative writing support purposes, I defined five statuses for an annotation: "unread", "read", "responded", "resolved", and "incorporated". With status information, reviewers can stay aware how their opinions are received by other reviewers and editors; editors can indicate their work progress by changing the status of annotations to "incorporated". Group writers can understand the group activities on annotations by looking at the statuses of annotations.

The status information indicates the stage that an annotation is at during its life cycle. Every newly created annotation starts with "unread." Later if users browse this annotation, the status changes to "read"; if users respond to or annotate on this annotation, the status changes to "replied"; if users incorporate this annotation and make changes to the document, the status changes to "incorporated"; if users resolve this annotation by deciding that it is irrelevant, the annotation's status changes to "resolved." Below I show an example status transition path for an annotation:

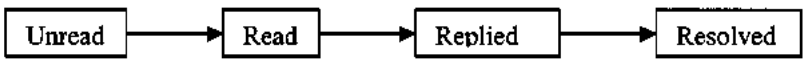


Figure 5.1 An Example Status Transition Path

9. Category

This feature captures the characteristics of the problems brought up in an annotation. Example categories are "incompleteness", "inconsistency", "question", etc. I came up with a list of categories based on my field study results at SWOG as section 4.6.1 shows.

Most prior annotation schemes or models use annotations to support information retrieval or integration. Compared to existing annotation data models such as RDF and Dublin Core Metadata Element Standard, this model has the following major additions: (1) annotations are associated with a life cycle with status information, which supports progress tracking and strengthens cross-role feedback between reviewers and authors; (2) annotations have extended activity-oriented properties such as rating, category of problems, response deadline, etc; and (3) annotations have richer context information, especially with version control. The design of this annotation model better manages annotations for development documents by supporting annotation-based change explanation and annotation version control.

5.3. Annotation-based Change Explanation

This annotation model further enables a novel service for collaborative writers, change explanation. Before I describe that concept, below I will introduce a related concept, change representation. Change representation has been an important research issue in collaborative writing research (Sharples 1993; Kim and Eklundh 2002). The better writers understand the changes, the better they can understand group progress and communicate about the writing task. Change

representation varies in different tools. Microsoft Word supports change highlighting, document comparing, and change accepting or rejecting. Group editor PREP supports similar change highlighting features to Word (Neuwirth, Kaufer et al. 1990), where old text is scratched out and new text is highlighted in a different color. When I ask the SWOG participants how they liked this feature, they told me that this feature has not been well adopted by protocol developers. Sometimes the scratched-out text from older versions interferes with reading.

The concept of change explanation is inspired by both research ideas for design rationale capture and the observation of underused annotation resources at SWOG. In SWOG, after the initial draft is complete, most revisions are driven by the suggestions in annotations. The annotation system designs I will describe in Chapter Six utilize the activity status information in annotations to summarize change rationale for evolving documents: all incorporated annotations are captured by the system to provide semi-automatic revision explanations. It is semi-automatic because some changes are not based on annotations; to capture the reasons for these changes require user input. I illustrate design in Figure 5.2.

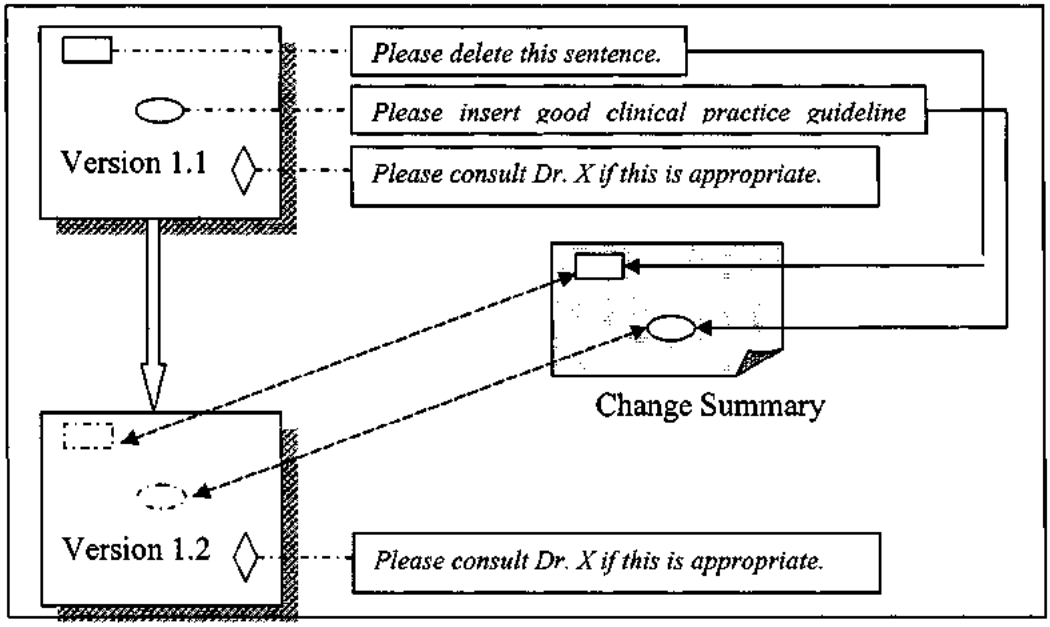


Figure 5.2 Annotation-based Change Explanation Service

In Figure 5.2, a shared document is first at version 1.1. Reviewers create three annotations on version 1.1. The editor incorporated the first two annotations, "Please delete this sentence" and "Please insert good clinical practice guideline 2.0" and made changes accordingly. Since the editor could not access Dr. X for two weeks, he decided to leave this annotation as it is. Then the editor made some other changes. When he thought some changes are important and should inform others, he made notes for these changes manually. After completing all necessary changes, he created a new version 1.2 for that document. In the new version, only the third annotation, which is un-processed, is visible. Later, a group writer noticed that the document had been changed and used the "change explanation" feature to find out what is new. This group writer received a "change summary" document. Each record in this "change summary" document consists of three

parts: (1) old text (2) new text (3) related annotations or manually input notes. Some power users can browse this "change summary" document, and accept or reject changes through the warm-linking technology originally innovated by the InterNote designers (Catlin, Bush et al. 1989). In Figure 5.2, the two-way arrows illustrate the two directions of warm-linking. If a change is rejected, then the older anchor text from version 1.1, which is captured in the change summary document, will be pushed forward to replace the newer but rejected anchor text in version 1.2.

5.4. How Annotations Are Carried across Versions

Brush designed an algorithm for robustly anchoring annotations back to changed documents (Brush 2002). This algorithm emphasizes linking all annotations to where they belong in the original document by searching for unchanged surrounding text. However, during collaborative writing, writers may not care about always anchoring annotations back to the changing document, but may only care how to anchor those unprocessed annotations to the changing document. In other words, during collaborative writing, annotations expire after they have been incorporated. Therefore, in this work, I propose to selectively anchor unprocessed annotations to changing documents.

Here is an interactive algorithm for annotation anchoring in changing documents.

First, unprocessed annotations refer to those annotations whose status is neither "resolved" nor "incorporated."

Second, when there is an unprocessed annotation attached to a certain version (N) of a document, when this document is revised, there are the following possibilities:

(a) If the revised part has 0% overlapping with the annotation's anchor text, then go to step (c) immediately; otherwise, continue to step (b).

(b) Pop up a warning message to the user and elicit user option. If the user agrees to update the anchor text in the annotation, then replace the old anchor text with the new anchor text. If the user does not agree, then prevent the user from editing that part of the document. Continue to step (c).

(c) When the document gets a new version ID, update the version ID for associated annotations accordingly.

In short, to robustly anchor annotations to changing documents, the two important steps are to link annotations to the appropriate version and to the precise location in the document. Since this model emphasizes making annotations and documents mergeable and detachable, annotations are inserted into documents through hyper links.

5.5. Summary

Current annotation design research falls into two major categories: one is the development of annotation data schema, and the other is the design of annotation systems that focus on annotation for static information. So far little consideration of activities on annotations in an evolving information space has been put into the

design of annotation data models and systems. My work fills in this gap by designing an activity-oriented annotation model. I have also grounded the design on an understanding of asynchronous collaborative writing processes at SWOG. Therefore, this work integrates disconnected research on asynchronous collaborative writing and research on annotation.

Rather than building application-level capabilities, I provide a data-level structure: the annotation model. This model includes task-specific features such as category, life cycle status, and changing context. I define life cycles status and dynamic context information for annotations. I also support in-context communication and problem solving, change explanation, and progress tracking with this annotation model. I hope this model bridges the gap between the functions of digital and physical annotations.

Collaborative writing needs effective group coordination and progress tracking support. One design goal of this annotation model is that manual emails for purposes such as version notification and change tracking or explanation can be replaced with status indicators in sharable digital annotations and automatically generated notifications.

Finally, information quickly amasses in an evolving information space. I hope to alleviate human cognitive overhead when they search for information. A physical annotation can be thrown away easily once I do not need the message on it. In this design, I make annotations detachable, versioned, and life-cycled. Therefore,

when an annotation reaches the end of its life cycle, it will be deactivated but archived in the shared space.

To test this annotation model, I implemented all the ideas described in this chapter in a prototype collaborative writing system, Protocol Collaborative Authoring Tool (PCAT). I describe the design details for PCAT in Chapter Six.

Chapter 6: Protocol Collaborative Authoring Tool (PCAT)

One important result of my dissertation is the Protocol Collaborative Authoring Tool (PCAT) that I designed in collaboration of the Southwest Oncology Group (SWOG). In this chapter, I will describe the design of this system and how it works. I start with the design objectives (Section 6.1); then I describe the system architecture and major system features (Section 6.2 and 6.3). I summarize this chapter with its generalizability and advantages over peer collaborative writing systems.

6.1. Design Goals

In Chapter Four, I highlighted the group work support needs of collaborative clinical trial protocol writers and identified the most problematic part of this group work, which is the iterative reviewing and revising process. Based on this understanding, I proposed a new annotation model to address these needs in Chapter Five. Following my design method described in Chapter Three, I view the PCAT prototype system as not only an embodiment of design ideas, but also a probing tool that helps engage users articulate their needs through their interactions with the tool.

Therefore, the design goals of a prototype system to support collaborative clinical trial protocol authors at SWOG are multi-fold as follows:

(1) To enhance iterative reviewing and revising of clinical trial protocols

(2) To assess the usefulness of the annotation model described in Chapter Five

(3) To use the prototype as a probing tool to engage users into the participatory design process and to help them articulate the tacit knowledge of their collaborative writing process that is hard to capture through qualitative fieldwork.

In this chapter, I describe how my design meets goals #1 and #2; in Chapter Seven I will describe how my evaluation studies of the design meet goals #3.

Protocol development is a complex group work with challenges from multiple aspects: (1) heterogeneous information input; (2) involvement of clinical researchers with a variety of expertise; and (3) various evolving clinical trial standards. Support for collaborative protocol development can be approached from different perspectives including knowledge reuse, information management, and computer supported cooperative work (CSCW). My work focused on designing a system to provide CSCW support to protocol developers at SWOG and to address the above challenges through the annotation design described in Chapter Five. My approach to protocol writing support is to augment the natural collaborative protocol writing process and to facilitate better interactions and more expressive communications among protocol writers. Aiming at improving the quality of the resulting clinical trial protocols, I support the natural work process to iteratively review and revise clinical trial protocols.

6.2. System Architecture

PCAT is a web-based and database-driven asynchronous collaborative writing system. The web has several advantages for a system design. First, it has wide accessibility: the world-wide-web has permeated into most people's daily life and work, and has become a popular collaborative platform. Second, it eases centralized information management and sharing. Third, "thin client" web application development technologies make it easy for clients to access the system installed on a server without installing lots of software on the client side. PCAT is database-driven because database management systems such as SQL Server provide reliable data concurrency and transaction control. The asynchronous working mode is the most suitable option for clinical trial experts, who often work at varied schedules. The complete system infrastructure is illustrated in Figure 6.1.

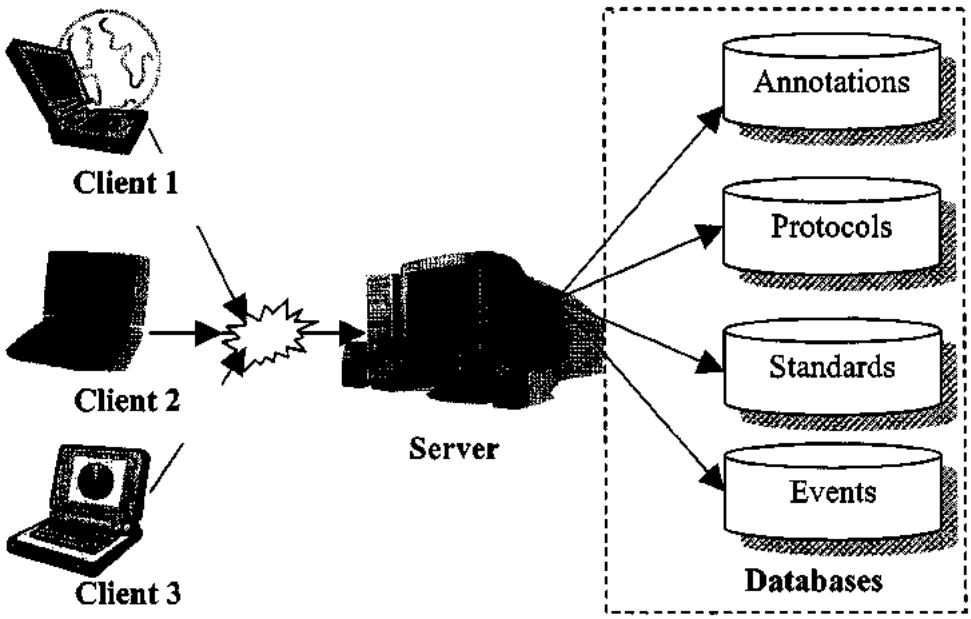


Figure 6.1 A Web-based and Database-driven Architecture

This system logically has four major databases: (1) annotations generated from the reviewing process; (2) evolving protocol documents; (3) protocol development standards such as boiler plate texts; and (4) group activities and events. According to Figure 6.1, all clients access the web services provided by the central server via the Internet.

In PCAT, annotations and evolving documents are stored separately. When collaborative clinical trial protocol writers edit or review the shared protocol document, links to annotations are inserted into documents. When protocol writers output protocol documents, links to annotations have to be removed. Therefore, this architecture enables linking annotations to documents and detaching annotations from documents dynamically and flexibly. The design is

implemented by combining DHTML and ADO.NET to represent annotations as hyperlinks. Each hyperlink points to an annotation stored in Microsoft SQL Server database. The editor can detect annotations by searching for hyperlinks that contain unique annotation identification numbers in documents.

The system also has two other databases for events and standards. The database for events records group activities and enables group awareness and progress tracking services. The database for standards stores reusable protocol templates and texts and enables knowledge sharing across protocols.

6.3. Major System Features

PCAT provides integrated protocol development support including collaborative editing, collaborative reviewing, progress tracking, and user management. Its collaborative reviewing features include in-context discussion and annotation life cycle management.

6.3.1 Collaborative Editing

As described in section 4.1, clinical trial protocols are formally structured documents. To meet this need, PCAT supports structured editing and adopts the current structure of clinical trial protocols developed at SWOG. Every protocol in PCAT is automatically divided into twenty predefined sections as Table 4.2 shows. In addition, at SWOG protocol developers are loosely-coupled and they work asynchronously and autonomously; therefore, PCAT provides concurrency

control at the section level. Multiple protocol developers can work on different sections of the same protocol at the same time, but each section is editable by only one user at any single moment.

Clinical trial protocols have strict formatting. To meet this need, PCAT provides a web-based rich-text editor by customizing an open source web-based editor called htmlarea (www.htmlarea.com 2005). PCAT's web-based editor has similar formatting functions to MS Word but has some advantages.

First, PCAT's web-based editor integrates DHMTL and Microsoft SQL server database programming. Therefore this editor provides robust data concurrency control coming together with SQL Server. In contrast, MS Word is a word processor with a file-based storage backend without concurrency control. MS Word also requires every user to have a copy of the shared document and does not provide a shared editing environment.

Second, PCAT's web-based editor allows users to read annotations and to incorporate annotations during revising. The feature enables users to provide feedback to reviewers who make the annotations.

Third, PCAT's web-based editor is connected to a database of boiler plate texts for protocol development. During editing, users can insert dynamic content on the fly.

Some of the important design features are embodied in the editing functionalities of PCAT's web-based editor. Figure 6.2 shows a set of editing buttons on top of the editor. Their brief functionality introductions are as follows:

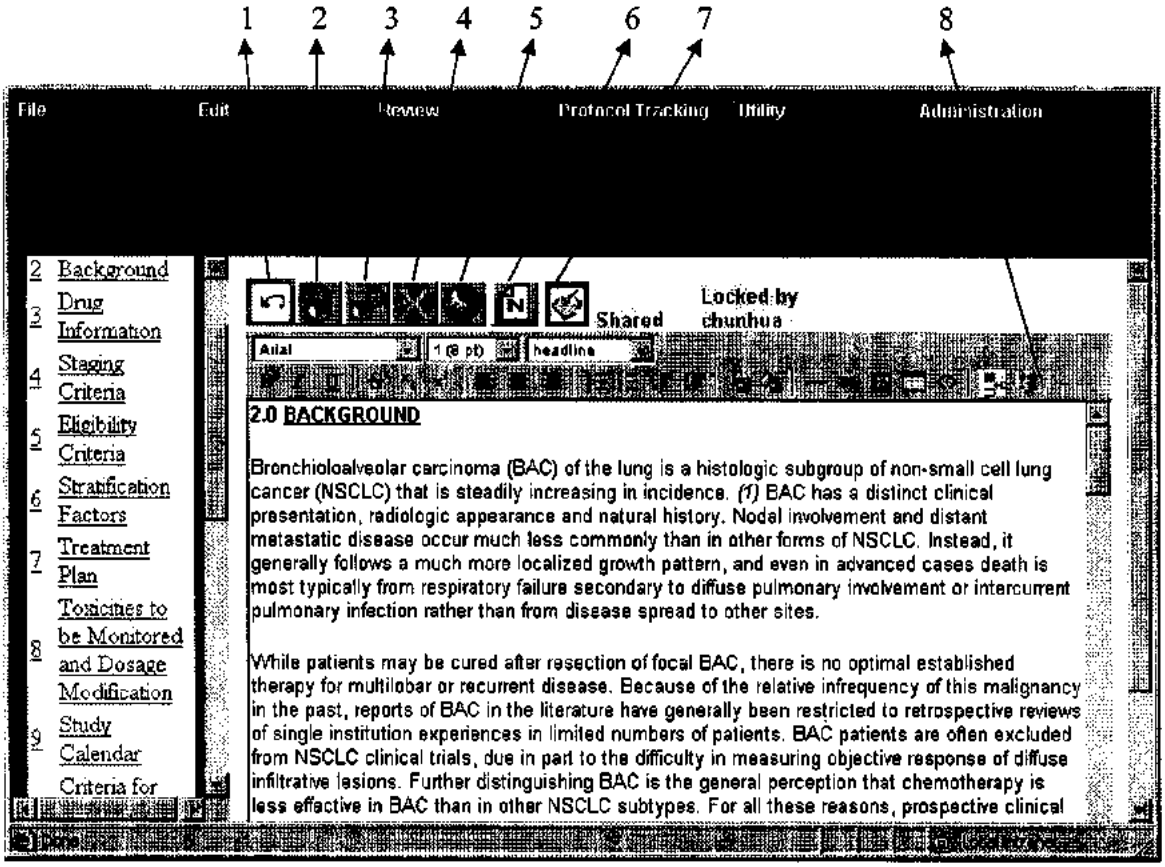


Figure 6.2 A Screenshot of PCAT's Web-based Editor

1. Recover—A user is able to revert the document to the last saved version.

2. Lock— Every time when a user gets the right to edit a section, that section locked for this user uses only by default. Sometimes when a user gives up the exclusive control of the editing right for a section, he or she can resume the

control by locking the section manually later if nobody else locks the section. This button is designed to encourage flexible locking control among group writers.

3. Unlock—When a user decides not to change the content by just reading it in the editor, he or she can press this button to unlock the section.

4. Save to the private copy—A user can select to save changes to the document to a private copy therefore those changes are visible to this user only.

5. Save to the shared copy—A user can select to save changes to the web-based shared copy; therefore, changes are immediately updated to everyone's view.

6. Insert standard wording—During the editing process, when the user presses this button, a window will pop up and show available standards for the user to choose from. Once a selection is made, the standard wording will be appended to the end of the content in the editor.

7. Making a note—To support change rationale capture, this function records some notes made manually by users. Such notes later will be combined with incorporated annotations to provide "change summary" to users.

8. Show annotations while editing—Annotations often contain rich information about how the current content should be revised. This function enables users to display annotation messages in a popup window during the editing process and to make changes accordingly.

Most of the above design ideas are informed by the field knowledge from Chapter Four. Collaborative clinical trial protocol writers have different priority because of their different status in the hierarchical health care system. The "locking" and "unlocking" functionalities enable flexible editing control transfer within the collaborative protocol development team. For example, a protocol coordinator can give the editing right on one section temporarily to a principle investigator, who happens to have a few minutes to edit the document that moment.

The "making a note" functionality is designed to help an editor provide revision feedback to reviewers and to help capture "revision rationale" that could be not connected to annotations. Such notes later could also be used to support group progress tracking.

Moreover, the "show annotations while editing" feature is informed by the fact that most changes in protocols during iterative reviews and revisions are based on change suggestions in annotations, as described in section 4.6.1. Therefore, if editors could view these annotations during the revising process, they do not have to memorize those annotations or to print a paper copy of them to follow suggestions about how to revise the document. In addition, editors could incorporate those annotations as they revise the document, as shown in Figure 6.6. Those incorporated annotations later will be used to provide group awareness and progress tracking support.

Figure 6.3 shows a screenshot for inserting boiler plate texts for protocol development. Users can use the functionality #8 ("inserting standard") to open this dialog.

Figure 6.4 shows the current locking status for different sections.

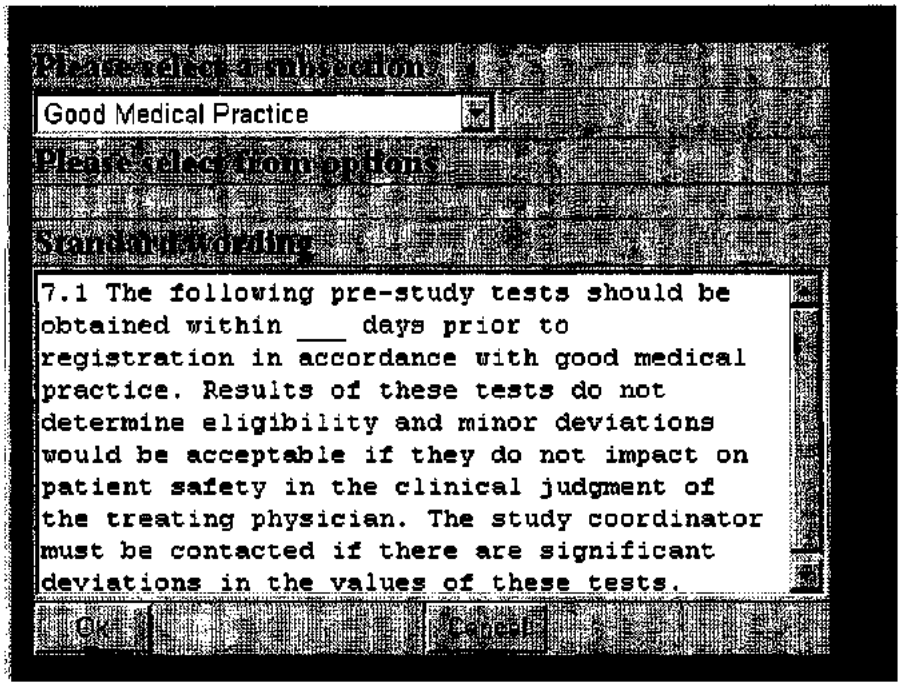


Figure 6.3 The Interface to Insert Boiler Plate Text

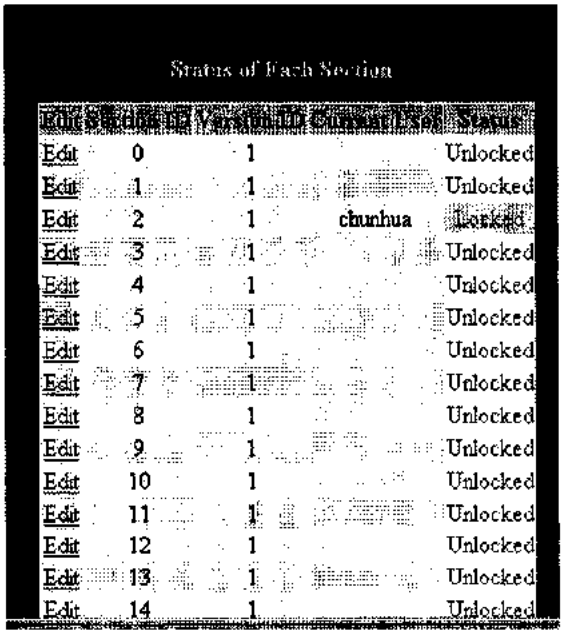


Figure 6.4 The Interface for Locking Status of Different Sections

6.3.2 Version Control in PCAT

PCAT has the following characteristics in its version control design:

1. Its version control is interactive through dialogues with users

2. Its version control is connected to the progress tracking function

3. It provides concurrent version control for both annotations and documents

Below I provide the details of these characteristics.

PCAT allows users to collaboratively edit the shared copy of the document, and archives all previous versions of the shared document. When a user chooses "Save to the Shared Copy," PCAT does not create a new version of the document automatically but overwrites the previous content of the shared copy. The reason for taking this strategy is that automatic version creation often leads to an overwhelmingly large number of versions during the lengthy protocol development process. The big pool of collaborative writers may also make this number even bigger. In addition, protocol developers at SWOG are not interested in version control over many versions. As we understand their work practice, they view a new version as a new milestone in the protocol development. In Chapter Four, I described how SWOG protocol writers use the single scriber strategy and the centralized version control strategy to simplify the document control problem. Based on such user needs, PCAT therefore provides a dialogue to support users

creating new versions of protocols when they feel the need. Figure 6.5 shows the interactive dialogue for users to create a new version of a document.

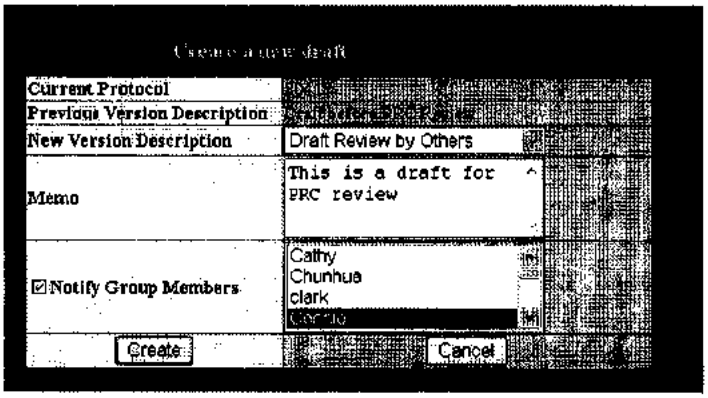


Figure 6.5 The PCAT Interface for Creating a New Version

In this dialogue, users can specify a version label and record a memo for the version control rationale. There is also an optional feature for "Notifying Group Members". Once this feature is checked by the user, he or she can select a list of users, who will get an email notification about the newly created version. This feature is designed to facilitate better group communication.

The content for the "New Version Description" field in the dialogue of Figure 6.5 is mapped onto the major milestones of the workflow for protocol development, which are described in Figure 4.3. Later I will describe in Section 6.3.4 how version information for completed versions of a development protocol is collected to support progress tracking.

Once a new version is created for the document, PCAT updates the version information for the annotations attached to that document. This updating process

is transparent to users but ensures unprocessed annotations, which may have a status as unread, read, or responded, to co-evolve with the changing documents.

6.3.3 Collaborative Reviewing

During protocol development, challenges in collaborative reviewing include ineffective group discussion, insufficient group awareness, inconsistent or redundant annotations, and poor information management. I have elaborated on these problems in section 4.7. To address these challenges, PCAT supports in-context discussion, life cycle management for annotations, and annotation incorporation during the collaborative reviewing process through my annotation design.

(1) In-Context Discussion

Collaborative reviewers often have to share annotations and to debate on some controversial problems. Available technology such as discussion forum supports discussions, but do not provide context for the discussions. In-context discussion means that discussions can be situated in the environment where the subject of discussions is accessible to reviewers. PCAT supports reviewers replying to annotations and threads related annotations. It also displays the threaded discussions side by side to the document content where the annotations are attached to. Figure 1.3 illustrates the interface of PCAT that supports in-context discussions.

The whole screen is divided to three frames: left, top-right, and bottom-right. The left frame shows threaded annotations; each annotation is a hyper link. When a user clicks the hyper link, the details for that annotation will be displayed in the top-right panel on the screen. In the bottom-right panel, there are three buttons: delete, edit, and reply. They enable users to delete or edit the annotations created by them and to reply to annotations by others. The right-top frame displays the document content where the currently selected annotation is attached to. The anchor text of the currently active annotation will be highlighted to draw a user's attention.

Since texts with annotations are highlighted in colors, collaborative reviewers can easily see which part of the document has been annotated by other reviewers. This helps draw reviewers' attention to the most problematic text. In addition, annotations threads enable collaborative reviewers to access related annotations easily and to find possible differences in opinions held by reviewers. This feature avoids unnecessary changes to the document where reviewers have inconsistent revision suggestions.

(2) Life-Cycle Management for Annotations

In section 6.3.2 I address how PCAT ensures that annotations are linked to the document during the version control process. However, a solution to annotation anchoring problem has to meet two needs: (1) link annotations to the appropriate versions and (2) link annotations to the appropriate anchor text. My above version

control design meets the first need; next I will describe how I meet the second need by supporting a life cycle management for annotations through designs of activity status in my annotation model.

This life cycle management for annotations is based on a support for annotation activity status transition. In Chapter Five, I describe five statuses for an annotation through the life cycle: "read", "unread", "responded", "resolved", and "incorporated." For a newly created annotation, if a user reads this annotation, PCAT records the activity and changes the annotation's status from "unread" to "read." Later if others reply to this annotation, PCAT will automatically change the status from "read" to "responded." The above transitions are automatically completed by PCAT. From the above three statuses, "read", "unread", and "responded" to the next two statuses, "resolved" or "incorporated", users need to get involved in interactions. If a user decides the annotation is irrelevant to the document, he can manually set the status for that annotation to "resolved" by pressing a button in the dialog that shows the annotation details.

A review process often generates many annotations. Management of these annotations can be very challenging. Specifically, challenges can be reflected in these aspects:

(a) Group Communication and Awareness: *What is a good way to notify group members of activities on annotations, such as reading, responses, or incorporation?*

(b) User Interface*: What is a good interface design for displaying relevant annotations on evolving documents?*

Although communication support for group work has gained wide attention from the CSCW research community, one of the open research questions related to my work is how to manage the balance between notifications and interruptions. In addition, an identified challenge for annotation user interface design is "orphaned annotations," where annotations cannot be robustly linked back to their context when the document is changed. To address the above research problems, I derive the following desiderata for my system design: (1) capable of detecting what annotations are relevant to the current document and reducing information overload; and (2) capable of providing shared feedback (Dourish and Bellotti 1992), which presents feedback on individual user's activities within the shared workspace. Based on these needs, I view a design that provides life cycle management for annotation is a natural need and can bring multiple advantages.

During collaborative writing processes, annotations are just middle-products. Annotations may be used for group discussions or revision suggestions; but they do not need to be linked back to the document after the discussions are completed or after the changes have taken place. Therefore, annotations should be selectively anchored to evolving documents. In other words, life cycle management for annotations can help filter unprocessed annotations and anchor them to evolving documents; resolved or incorporated annotations could be archived by the system but should be detached from the evolving documents.

During the editing process, when a user decides to incorporate an annotation, he or she can press a button labeled with "Incorporate this comment". The system then changes the status of the annotation to "Incorporated." Figure 6.6 shows the interface to incorporate an annotation. Incorporated annotations are archived in the system for later "change summary" uses. Figure 6.7 shows a change summary for a document, where the earlier version of a document is indicated to have three incorporated annotations.

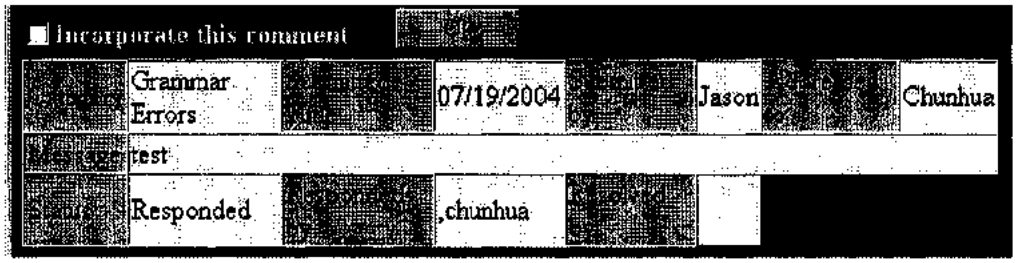


Figure 6.6 The PCAT Interface for Incorporating an Annotation

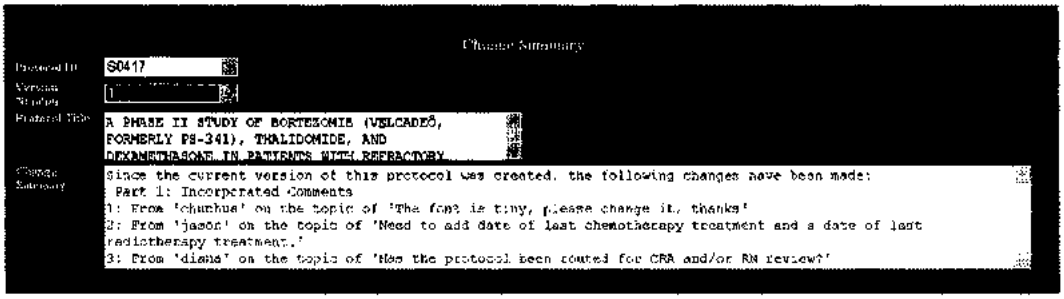


Figure 6.7 The PCAT Interface for a Change Summary

During editing, if a user changes the anchor text of an annotation, whose status is neither incorporated nor resolved, PCAT will generate a warning message after

detecting that being changed text contains unresolved or unincorporated annotations. PCAT does not accept such changes.

In sum, the statuses in the life cycle of annotations can be customized to specific needs of different users. For the collaborative writing process at SWOG, the five statuses defined above are sufficient for the iterative reviewing and revising process.

6.3.4 Group Progress Tracking

The group progress tracking functions in PCAT enable users to track progress on rich information, including (1) group activity awareness; (2) reviewing progress; and (3) overall protocol development progress.

(1) Group Awareness

PCAT records every user log in and every activity this user carries out, and updates every minute all user activities in a small always-on-top window. This function provides shared background group awareness (Dourish and Bellotti 1992) to all group members. My SWOG participating users find this feature very useful in facilitating synchronous communication within the group. Normally collaborative protocol writers work asynchronously because it is difficult for scheduling among a large group of users. With this group awareness support, they can see who else is online working on the same protocol, and then they can either

talk to each other in the system or chat on phone. This helps them coordinate the group work.

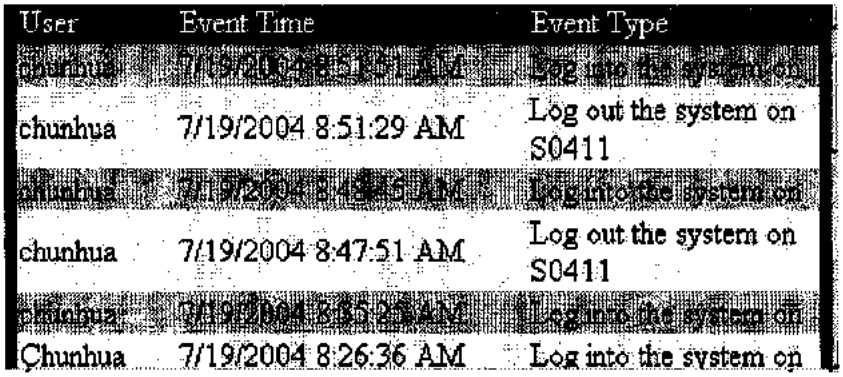


Figure 6.8 Group Awareness Support

(2) Annotation Status Browser

My field studies at SWOG show that protocol reviewers barely know how their suggestions are received by the protocol coordinator, why some comments are simply ignored, how different reviewers hold different opinions, and so on. To address this problem, besides supporting group activity awareness, PCAT also supports collaborative writers to track the review progress. Status information of annotations indicates how many annotations have been shared within the group and how many have been incorporated or resolved. Prior to the use of PCAT, this sort of information has rarely been available to reviewers.

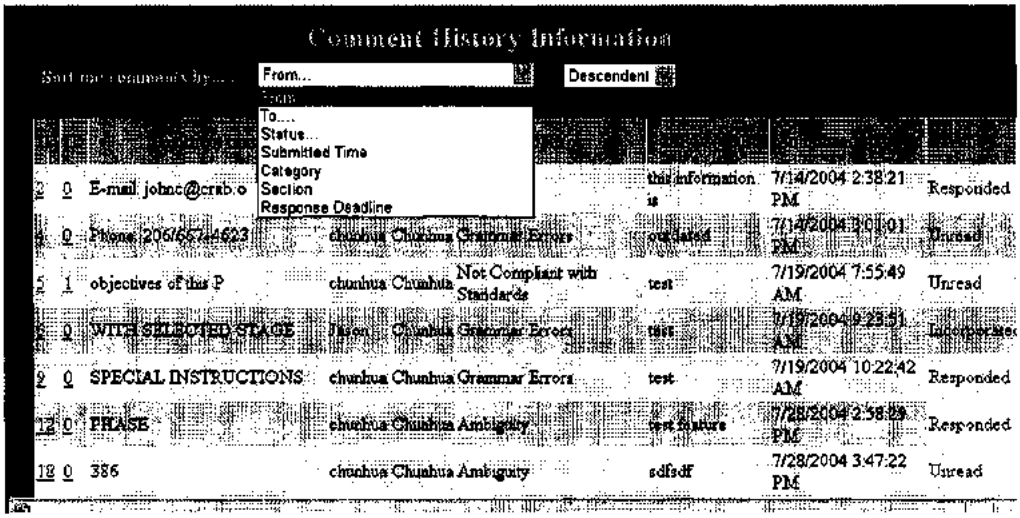


Figure 6.9 Annotation Status Browsing

(3) Development Progress Tracking

Figure 6.10 shows a screenshot for PCAT's progress tracking service. Protocol development progress is mapped onto the major milestones in the current SWOG protocol development workflow. Every SWOG protocol goes through the following sequential stages: concept submission, initial draft submission, draft submission to protocol review committee (PRC), draft submission to CTEP, and so on. Therefore, PCAT embeds a workflow model in the progress track service and integrates work progress capture and tracking.

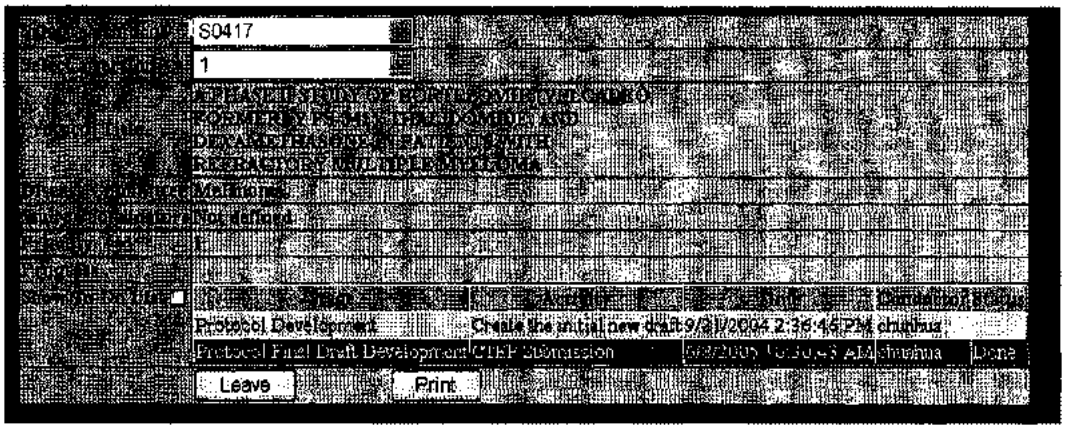


Figure 6.10 The PCAT Interface for Progress Tracking

In sum, PCAT provides integrated support for collaborative editing, collaborative reviewing, progress tracking, and group awareness support. In addition, to avoid users to switch applications during work, PCAT also support web-email services and interoperability with MS Word. Every protocol can be output as a Word document. If there is a protocol in word format, users can manually break into down twenty sections and load them into PCAT. Automation of loading word documents into the system is one of my future research plans that I will describe later in Chapter Eight.

6.4. Summary

In this chapter I described the features of the PCAT system. Based on the annotation design in Chapter Four, PCAT provides shared feedback for group activities on annotations and supports in-context discussions through threaded annotations. Its life cycle management for annotations and annotation

incorporation support are unique among existing collaborative writing systems. In Chapter Seven I will carry out evaluation studies to validate these designs.

Chapter 7: Evaluation Results

Evaluation of group work support technology is full of both social and technical challenges. I had been carrying out formative evaluations on the prototypes for the Protocol Collaborative Authoring System (PCAT) throughout the design process. In this chapter, I describe my evaluation results from these aspects: (1) quantitative results from a usefulness survey of the PCAT prototype (in Section 7.2); (2) qualitative results based on users' comments about the design of PCAT (in Section 7.3); and (3) a reflection of the participatory design, including how it complements qualitative fieldwork and what the social-technical gap for the design and evaluation of healthcare groupware could be (in Section 7.4).

7.1. Evaluation Design

The objective of my evaluation is two fold: (1) to assess the usefulness of the system features in PCAT, and (2) to understand users and their work better throughout the prototyping process. I took a staged evaluation procedure and used both qualitative and quantitative methods as I explained in Section 3.2.3 to address the research questions listed in Section 3.3.3. Below are some further implementation details.

As part of the formative evaluation process described in section 3.2.3, I demonstrated the prototypes under development to my participatory design users every couple of months between July 2003 and August 2004 to elicit user

feedback for the interface designs as well as how the system might fit into their daily work.

In addition to formative evaluations, I carried out two more case studies. One was a scenario-based case study (Jacobson 1995) in early 2004 by inviting a protocol design group to use the system to complete some protocol design tasks following a predefined scenario (see Appendix IV: An Example Evaluation Scenario Script). The scenarios defined a series of user interactions with the system in the context of work for assessing the system features that support these interactions. Four users in three roles participated in the evaluation for two days. The other was a field trial in a real work setting in late 2004. I asked a protocol design group to use the tool for about a week to support their daily protocol design tasks. I included eight users in four roles into this evaluation.

Throughout this whole evaluation process, I documented ten major formative evaluation meetings with written notes, excluding informal meetings with the SWOG users. For the two case studies, I conducted one-hour long semi-structured interviews with each participant. I partially transcribed and partially constructed notes for the interviews. During the second case study, I used Appendix III: A Likert Survey of System Usefulness to ask the participants to rate the usefulness of the system features. Likert survey was originally developed by Rensis Likert in the 1920's in an attempt to improve the levels of measurement in social research. It uses standardized response categories in survey questionnaires and is one of the most frequently used attitude measure in social sciences.

7.2. Quantitative Evaluation Results

My quantitative evaluation results consist of two parts: (1) the results from the Likert survey about the usefulness of the design of the PCAT prototype and the annotation model, and (2) the results from logging data in the PCAT prototype.

I surveyed eight participants in four roles, including two statisticians, two protocol coordinators, one data coordinator, and three PRC members, on the usefulness of the system features. They used a scale from 1 to 5, where 1 represents useless and 5 represents very useful. Table 7.1 shows the results.

Table 7.1 Ranked system feature usefulness based on average user ratings

Ranked System Features	All	PC	Stat	PRC
Pinpoint text for annotations	4.75	4.50	5.00	4.75
Highlight annotated text in color	4.63	5.00	3.50	5.00
Address annotations to people	4.13	5.00	4.50	3.50
Share annotations during reviews	3.88	4.00	5.00	3.25
Thread annotations for discussions	3.88	4.50	4.50	3.25
Generate print-friendly annotations	3.88	3.50	3.50	4.25
Respond to annotations	3.75	4.50	5.00	2.75
Browse annotation status	3.63	4.50	5.00	2.50
Email notification service	3.13	4.50	2.00	3.00
Sort annotations by properties	3.00	2.50	3.00	3.25
Filter annotations	2.75	3.00	3.50	2.25
Incorporate annotations	2.63	3.50	2.00	2.50
Set up response deadline	2.63	4.50	2.00	2.00
Categorize annotations	2.38	2.50	3.00	2.00

Table 7.1 shows that overall the top three most popular system features include pinpointing the text in a document for an annotation, highlighting annotated text in different colors for different reviewers, and enabling reviewers to address

annotations to certain people for coordinating group work. The least useful features are incorporating annotations during editing, setting up response deadlines for annotations, and categorizing annotations based on their content. Although based on a small sample and thus possibly influenced by individual preferences, the differences seem logically related to the roles of the respondents.

Categorization of annotations had been a focus of the participatory design process for a few months, but was rated as the least useful feature by the participating users. Both users and I thought such categories would be very useful to help protocol designers classify the common problems in protocols and help with filtering review comments. However, filter annotations by category was used less than expected. Our interviews with the users showed that it is not easy to pick a category. This result confirms Grudin's comment that when introducing a new feature, it is important to make sure that no extra hard work is added and that the person who does the job really benefits from the feature (Grudin 1988).

Some features have a low overall average rating but a high average rating for some individual role. For example, "set up response deadline" is not generally welcomed by all users but is highly preferred by statisticians. Feature ratings also vary from role to role. Both statisticians and protocol coordinators prefer to share annotations while reviewing protocols, but PRC members do not like this feature. Also, statisticians prefer to get email notifications to monitor the status of their discussions in threaded annotations; but protocol coordinators dislike this feature because most incoming emails simply remind them of new editing tasks.

The protocol in my particular field study received 100 annotations in 58 threads in total. Below in Figure 7.1 I illustrated the distribution of these annotations across different categories, as defined in Section 4.6.1. Comparing the data in Figure 4.9 and Figure 7.1, I found that there is a similarity between the annotation distribution among categories for protocols reviewed without using PCAT and the annotation distribution among categories for the experimental protocol reviewed in the PCAT prototype. "Revision Suggestions" and "Uncertainty" are the two most common categories both with and without PCAT being available. Incompleteness, Ambiguity, and Inaccurate References are the remaining common problems. After using PCAT, protocol reviewers found fewer "Inconsistency" problems than before using PCAT.

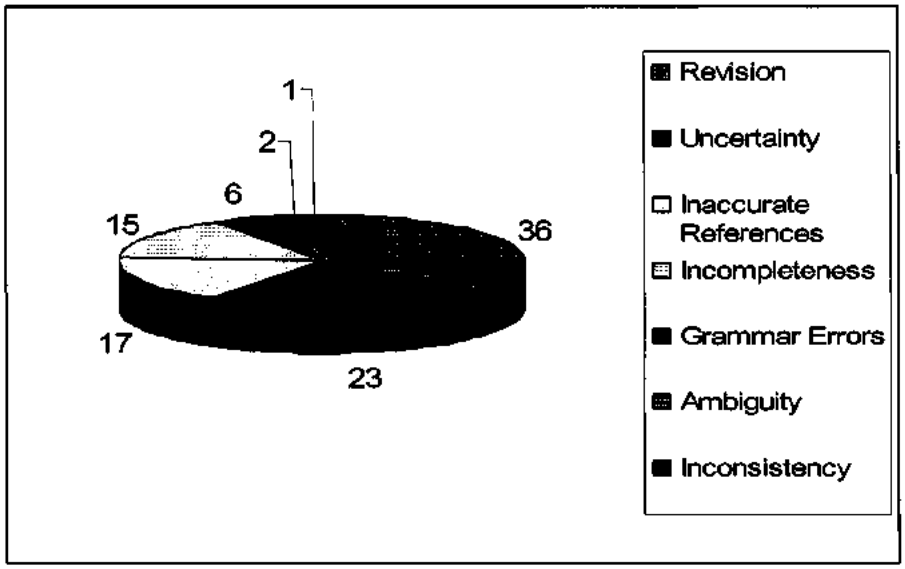


Figure 7.1 Distribution of Annotations among Different Categories

Figure 7.2 illustrates the annotation distribution among different sections in the experimental protocol. For convenience, I only picked the first nine most annotated sections, which are ordered as "Eligibility Criteria", "Treatment Plan", "Background", "Stratification Factors", and so on as shown in Figure 7.2. By comparing this figure to the data collected through my field studies as shown in Figure 4.8, I found that "Eligibility Criteria" and "Treatment Plan" are consistently ranked as top problematic sections, which are more complex and needs more computing support than other sections. The "Study Calendar" section should also be a heavily annotated section; however, PCAT could not capture the annotations for this section at all during my field study because PCAT does not support annotation for both web documents and Microsoft Excel files. I also explained this technical limitation in Section 7.4.1.

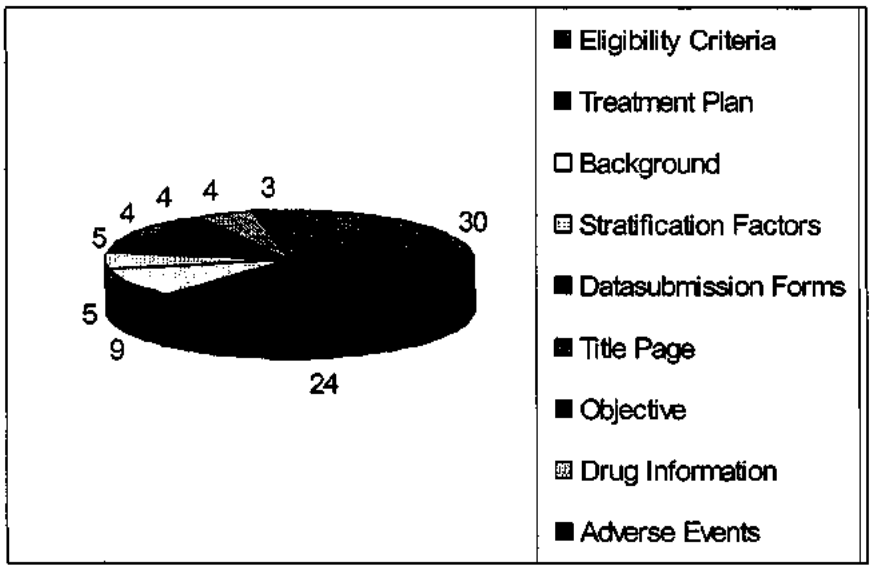


Figure 7.2 Distributions of Annotations among Different Protocol Sections

Moreover, in Section 4.6.1 I mentioned that the average number of comments for protocols during PRC meetings is 47 comments. This protocol reviewed in PCAT received significantly more annotations compared to the above average of 47 comments, 100 in total. There could be multiple reasons for explaining this phenomenon: (1) the protocol could be more problematic and contained more errors than average protocols; (2) the user interface design of PCAT made it easy and interesting for the users to add more annotations than usual; and (3) the threaded discussion feature encouraged the users to add more annotations to carry out in-context communication with other group members.

Future studies are needed to evaluate how my PCAT system could improve quality control of protocols by identifying more problems in the protocols and by

facilitating more discussions among collaborative clinical researchers through threaded annotations.

7.3. Qualitative Evaluation Results

The above section provides quantitative evaluation results about the usefulness of the design. In this section, I describe the qualitative evaluation results that help us better interpret the above quantitative results. Here are two representative examples of the system's impact on users.

(1) Does sharing annotations help or not?

One of the PCAT's design goals is to enable protocol reviewers to share review annotations online and hence alleviate some group members' effort to distribute annotations to the group. However, this design introduced a tradeoff between enhanced information sharing versus increased workload for some users. It makes it easier to access group annotations but changes the old division of labor among group members. Some of the quotes from the users in the study follow. Comments and annotations refer to the same concept.

A PRC member: "*Overall, I do not believe this is a useful tool for PRC members. It requires too much time and effort for the individual reviewers. Each person has to wade through other people's comments to make sure that they are not repeating something. And then you have to take the time to type out what you*

want to say. And each comment requires you to go through a series of menus and options to select, which is time consuming."

A PRC member: "*This tool is shifting the workload. Traditionally, I just need to talk about our comments on the face-to-face meeting. It is the responsibility of the leading statistician to compose all the comments for us. Now I have to do the job, and he just needs to read the annotations in the system.*"

A protocol coordinator: "*Is it helpful to look at other people's comments? It is sort of a double-edged sword. By looking at other people's comments, you can see how other people phrase a comment in a better way so that you can learn. However, I cannot stop myself from looking at other people's comments; therefore, I have to spend more than double the time on the system. I cannot get my job done in the limited time. Also, not every comment applies to me.*"

(2) Is it easy to do the review online?

The users expected a system that supports a smooth transition between physical and digital documents, but our current technology does not support this well.

A PRC member: "*computer-based commenting interface implies a formal process, not as comfortable as on paper. On computer, I feel that I have to know what I have to say, but on paper I can feel more relaxed. I don't like to read protocols online. I will print a hard copy, read it and type comments online.*"

A data coordinator: "*I often do cross-section checking. Therefore, I have to look at multiple sections together to identify inconsistency. It is much easier for me to lay out paper copies for these sections on a table and read them together. On computer, I have to switch between windows and it is not quite easy. By the way, Can I add a comment for multiple sections? Can I change the anchor text for an annotation?*"

A PRC member: "*A protocol contains excel spreadsheets, word documents, as well as PDF files. Your system does not support commenting on these entire file formats. It is not comfortable for us to launch multiple applications to do separate reviews.*"

These two examples demonstrate the tradeoffs in sharing and digitizing information. Managing these tradeoffs is an active research area in CSCW and HCI. Providing a solution to such tradeoffs is beyond the scope of my dissertation; however, these results show an example gap between what we can support technically and what we have to support socially.

7.4. A Methodological Reflection

As described in Chapter Three, I intertwined fieldwork and participatory design in this research to achieve a synergy of both. After looking back at my participatory design process and my field studies, here I summarize some lessons I learned from using this method.

7.4.1. Unstable User Feedback over Time

Throughout the prototyping process in this work, user feedback was unstable and my system designs had to change frequently because of the changing user requests over time. Sometimes I had to revert to an earlier design. Below are two examples.

(1) Is formatting important or not?

In March 2003, I consulted the participatory design users on what file formats should be supported in the system. Those users said:

"*The content of a protocol was more important than format; as long as I can lay down texts so everybody can see the text and give comments, it would be fine. I can leave the formatting to the very end of the design.*"

Therefore, I did not put much consideration to the formatting during the design of PCAT. However, later I encountered some obstacles for evaluations in the field trial because of the heterogeneous information sources that protocol designers use. Protocol reviewers have to review content in multiple file formats including Word, Excel, and PDF files. Both they and the protocol coordinators got frustrated about the insufficient formatting support, as indicated in these quotes.

"*People are accustomed to commenting on formatting. The PIs might not like to comment on not well formatted text.*"

"I think formatting is important to understand the content of the protocol in some sections. The loss of formatting on the web view makes review difficult in some sections, such as the study calendar table."

I adopted the open source web editor called *htmlarea* as part of my design. This editor does not have formatting features fully consistent with Microsoft Word, especially for tables and footnotes. Since my participants were very familiar with Word and needed rich formatting support as offered in Word, they complained that this editor was not fully interoperable with Word and had some reluctance during the evaluation studies. Although my evaluation goal is to test the usefulness of the annotation design and the prototype system PCAT and interoperability seemed to be just an engineering problem, this unsolved technical problem did hinder my users from using PCAT to some degree. This lesson assured me the importance of carrying out effective evaluations in the real work context, as Kaplan suggested (Kaplan 1997).

(2) Should there be a single or multiple editors?

In 3.2.2, I described my iterative prototyping and the major iterations that I went through. In fact, some of the iterations were driven by the changing user needs. At an early stage of the prototyping process, my users told me that only one editor, the protocol coordinator, should be allowed to revise the protocol after reviews. Later my users felt encouraged by the group support features and requested the editing rights be given to all the protocol writers so that they could edit different

sections of a protocol concurrently. I made corresponding changes and designed section-based concurrency control and version control.

However, when I demonstrated the revised design to the participatory design users including one manager of all protocol coordinators, this manager started to hesitate and told us some of her other concerns. In their current system without PCAT, the protocol coordinator in a protocol writing group is the only person who can write content into a development protocol. The person in this role knows exactly what exists in the old draft, what new input is needed, and what information is authorized to be put into the protocol. When there are different opinions among different protocol writers, the protocol coordinator often waits until the negotiation among these protocol writers is resolved. In contrast, with such an open collaborative writing environment as PCAT, where everybody can write freely, she was worried about the consequence of removing the centralized group control of protocol coordinators. For the same section, different protocol writers may try to edit it in different ways. The whole work may lose the advantage of having the centralized coordination by protocol coordinators. Protocol writers may have to communicate directly within the group through their writings in the shared protocol, rather than through comments. This may further cause the inconsistency of the document.

In addition, other participatory design users in other roles did not object to the manager's concern. They also indicate that it would be easier to let the protocol coordinator do their job as before and take the exclusive editing control of the

document. Therefore, eventually they switched back to the single editor mode and I reverted to one old solution accordingly.

One possible explanation for this example could be that the users prefer explicit communication through comments to less explicit communication through writings. Another possible explanation for this example could be that users were worried about losing the control of group work by giving group members too much freedom.

The above two examples in this section are also representative of challenges for requirements engineering and change management during an iterative design process, where users have difficulty in articulating their needs precisely.

7.4.2. The Social-Technical Gap

The evaluation results of this research reveal a divide between what I must support socially and what I can support technically, which is a social-technical gap defined by Ackerman (Ackerman 2000). Next, I elaborate on the socio-technical challenges that form this social-technical gap.

(1) Technical Challenges

There is a severe tension between the capabilities of available technologies and people's needs for support of their habits, ideals, and nuanced behaviors. In this setting, the users expected digital documents to support their contextualized reviewing behaviors on physical documents, such as annotating multiple sections

together and creating annotations anchored to multiple texts in a document. In addition, the users are used to MS Word and expect the online editor to provide formatting features consistent with MS Word. Moreover, users are used to reading paper copies while making comments but current technology does not well support smooth linking or blending of physical and digital documents. These challenges hinder further system uses and evaluations.

(2) Expanding User Needs

Our initial design was centered on the iterative review and revision support. Our final design includes progress tracking, web email, user management, and many other features. All of them have been developed toward the elicited user needs. In our annotation design, I defined "unread", "read", "responded", "incorporated", and "resolved" statuses for an annotation. In our recent field trial, our users suggested adding another status: "agreed" so that some reviewers can simply acknowledge existing comments without creating new comments. I foresee that if I implement this feature, I still need to define "disagree" and support some voting mechanism. In a participatory design process, users naturally grow mature and savvy at the same pace as the design. But as users become savvier, it is more difficult to satisfy their changing and expanding needs.

(3) Lack of Incentive for Certain Users

Throughout our designs and evaluations, I did not get sufficient access to one major role, the PI, in the group work of protocol design because of two challenges.

First, PIs are unlike statisticians or protocol coordinators, who belong to the same organization. PIs are affiliated with various organizations. They come and go because designing clinical trial protocol is not their profession, but a once or twice experience in their life. To understand the user needs of this fluid population of PIs can be a big research project on its own. Second, PIs seem to have little incentive to use a group work support technology. According to the user feedback, the PIs are currently at the high level in the group work. To avoid overwhelming PIs with too much information, protocol coordinators often send composed and filtered messages to PIs. According to our field study, some PIs do not know the tedious work that other group members do and few of them have a clear picture about the whole group work process. It is hard to motivate the PIs to participate in a technology design that may threaten their power.

(4) Group Work among Non-peers

The collaborative work within SWOG is essentially among non-peers. Protocol coordinators are supposed to support other people. They do most of the work, but they do not get the authorship. Similar to other healthcare settings, the people involved in protocol designs have disparate training backgrounds and sit at different levels in the hierarchical healthcare research system. This situation creates confusion for system designers. Should our groupware technology support the lower level workers or those higher level PIs? Should I increase the difference in work load or should I strive for a balance? In these groups, some roles are subordinate to other roles. It appears there will be constant conflicting needs from

these different roles. Group work support technology often subtly changes the group structure and interpersonal relationships, which is one of the eight classic challenges for CSCW designs identified by Grudin (Grudin 1988).

(5) Subtle Organizational Nuances

Healthcare settings are complex and full of nuances. Our evaluations reveal some subtle feelings that some users have. A data coordinator says, "*I am a senior protocol developer, I definitely welcome shared annotations. But if I were a new protocol developer, I would not like my comments to be seen by others. I do not want other people to think my comments are stupid and I know nothing. I can read other's comments, but I would not like to share mine.*" Protocol writers have disparate training backgrounds and do not have shared cultures. They have subtle concerns toward the concept of group learning. Another example, our system tries to facilitate direct communication among group members to minimize unnecessary efforts for manual information relay, but the evaluation results show that the users think direct communication may impose commitment on receivers. They prefer the currently loosely-coupled collaboration mode instead. Nuanced behaviors like these are often hard to perceive. Designers need to thoroughly understand users and their work with insight and over time.

(6) Changing Organizational Structure

Not until our recent field trial did I learn that most protocol coordinators do not make changes directly on protocols, but a new role, word processor, does the job.

The word processors were not ready to use the system both because they were less trained and they were occupied with editing tasks. I was told that the organization introduced this role to alleviate the work load of protocol coordinators, but this change interfered with our evaluation plan. The feature "incorporate annotations" was lowly rated as shown in Table 7.3, possibly because the feature was never used. Therefore, ongoing organizational changes can have great impact on designs. System designers must be sensitive to various organizational changes and prepare proactive plans.

The above results exemplify some classic challenges for group work support (Grudin 1988), such as disparity in work and benefit (e.g., work load shifting), disruption of social processes (e.g., difficult to motivate and engage PIs), and difficulty of evaluation.

7.4.3. Complementing Fieldwork and Participatory Design

A descriptive fieldwork and a participatory design process complemented each other effectively in my dissertation research.

One challenge for fieldwork research in general is that a field often contains unlimited knowledge. It is difficult for a system designer to identify a meaningful focused area. Therefore, it is tempting for a system designer to first come up with a design idea and then focus a field study on this design idea. Then the fieldwork would be driven by a particular design. In this dissertation research I did not take

this approach. I benefited from my descriptive field work in that I was not biased toward any design idea at the beginning.

During the early stage of my participatory design, when I inquired about what informatics tool support SWOG might need for protocol development, the participants from SWOG unanimously said that they needed better knowledge-based management for protocol standards and version control for developing protocols. They were aware of the problems in protocols such as inconsistency and ambiguity, and they imagined that knowledge-based systems would be the solution. No one mentioned that group communication was actually a big challenge for them.

Later, after I observed the group communication via email and the collaborative protocol reviewing process, especially after I identified the discussions carried out through protocol review comments, I realized that iterative reviewing and revising of developing protocols was a more severe problem that needed to be addressed first. After I explained this idea to the participants at SWOG and shared with them the field study results, they agreed on the importance of supporting iterative reviewing and revising of developing protocols and were willing to put this area as a top priority for the research. Only from this point on did I start to focus my system design on supporting the iterative reviewing and revising process.

The above example shows how my descriptive field study inspires a useful idea and complements my participatory design. The following example shows how my participatory design corrected a misunderstanding I had in my field study.

A common goal of technology designs is to support or improve the work practices of users. One challenge for introducing new technology is change management, as I describe in 7.4.2. A lesson that I learned in this research is that a system designer needs to know what should be changed as well as what could not be changed when supporting work practices of users.

During the participatory design process, I once provided a "protocol information retrieval" design in the PCAT prototype. I came up with this protocol standard management design because I observed this work practice at SWOG during the fieldwork. I thought this work practice needed to be supported in the PCAT prototype. The users at SWOG looked at this design and told me:

"Yes, this is how we do the work now. But we were not happy about it. This is not the right way to manage protocol standards. Protocol standards change very frequently. Copying and pasting standards from previously designed protocols introduced a lot of errors. We need to change this."

I learned from this example that we may be able to see how users do the job, but we may not see how they feel about it. There is much tacit knowledge in the field. Here this information is impersonal so this user spoke up. At some other times, it may be hard to collect tacit knowledge from the field, such as interpersonal

relationships, consultation of users to get their feedback for a design through participatory design would be helpful.

Overall, I found it helpful to share the field knowledge with SWOG participants down the road for a couple of reasons. It helped make the group work visible to these domain experts, who were immersed in their work all the time hardly knowing the big picture. Knowing more about their work and their group workers helped these users better understand the feasibility of their design suggestions. Moreover, it helped me collect feedback from users to validate my interpretation of field study results.

7.5. Summary

In this chapter, I describe the evaluation results for my research and contribute an understanding of a social-technical gap for healthcare group work support in the Southwest Oncology Group. The social-technical challenges include expanding user needs, lack of incentive for group members, difficulty for collaboration among non-peers, and a gap between what is technically feasible and what users require socially. The study further strengthens my belief in the importance of using social-technical approaches to design healthcare information systems.

Multidisciplinary healthcare group work is common in various healthcare settings. For instance, electronic medical records are integrated results by physicians, nurses, lab result analysts, and many others. Such group work may share similar challenges as collaborative clinical trial design. In Chapter Eight I will elaborate

on my ideas of future work that may extend this research to other groups in medicine domains.

Chapter 8: Summary and Conclusions

In this chapter, I summarize my understandings of collaborative clinical trial protocol development and my annotation design to support collaborative writing (Section 8.1), discuss the contributions of my research (Section 8.2), report on the limitations of my current approach (Section 8.3), and present avenues for future work (Section 8.4).

8.1. Supporting Collaborative Writing through Annotations

This dissertation contributes early knowledge about how clinical trial protocols are developed through group work at the Southwest Oncology Group (SWOG). In response to the research questions in section 1.3.1, I described the major roles and their responsibilities for protocol development at SWOG. I illustrated the current work practice at SWOG including the centralized document control and group work coordination. I also explained the challenges for the loosely-coupled protocol development from multiple perspectives: the subtle interpersonal relationships in the hierarchical healthcare system, the diverse incentives among multidisciplinary group members, and the limitations in email-based group communication and coordination. On this basis, I focused on the iterative reviewing and revising process during collaborative protocol development. I described how comments are used by group writers, the major categories of comments, as well as the distribution of comments in different sections and in different categories. I also identified four major roles of review comments in the

collaborative writing process, which are communication artifacts, group learning artifacts, activity justification artifacts, and progress tracking artifacts.

Informed by the field knowledge, this dissertation also defines a new annotation model that supports collaborative writing. This annotation model first supports version control for both annotations and evolving documents and supports life cycle management for annotations on evolving documents.

The formative evaluations and the two field case studies of the prototype system PCAT at SWOG demonstrated the potential usefulness of some aspects of this design. Some participants at SWOG are willing to adopt this tool in their daily protocol development work. Some annotation features in my design received unanimously positive feedback from the users, such as linking annotations to specific areas in the document precisely and highlighting annotations in colors for different reviewers. These evaluation studies also revealed an example socio-technical gap in group work support. This gap consists of the social-technical challenges, such as to support information sharing while avoiding workload shifting within the group, to help users become more savvy about the technology design while trying to meet the expanding user needs, to address the subtle organizational nuances, and to support group work among non-peer workers.

As I describe the limitations of my research in section 8.3, my evaluation studies involved only a small number of participants for relatively short periods. More evaluation studies are needed to justify broader claims. Nevertheless, these initial

results suggest that by supporting in-context communication among collaborative writers and by providing life cycle management for annotations in an evolving information space, PCAT has some potential usefulness for the group work of collaborative clinical trial protocol writers.

8.2. Contributions

This dissertation has contributions to three disciplines: medical informatics, computer supported cooperative work, and clinical medicine.

8.2.1. Contributions to Medical Informatics

The primary contributions of this dissertation to medical informatics include (1) an understanding of the group work of clinical trial protocol writing (2) a collaborative clinical trial protocol writing system PCAT and a new annotation design for evolving documents (3) an implementation of the grounded design method during the design of PCAT and the annotation model.

No previous studies have described the protocol development as a group work among multidisciplinary clinical researchers. The information I collected by studying this group work at SWOG provides useful field knowledge for future medical informatics researchers who are interested in developing tools to support this kind of group work.

Research in Medical Informatics started from artificial intelligent methods applied to medical domains. Group work support technology in this domain has been very

rare. Prior to my study, protocol developers that I have studied have not recognized the problems in their group work; instead, they viewed the lack of knowledge-based support as the major problem. I identified the most problematic part of protocol development as the iterative reviewing and revising process that involves challenging group communication and coordination. I also uncovered that group protocol writers do not follow a strict workflow model; instead, their work is loosely coupled. This understanding implies that a rigid workflow support system may not work well for collaborative workers in this setting.

Moreover, I designed and implemented the prototype system PCAT to support the collaborative work among SWOG protocol developers. My evaluation studies validate the usefulness of some features of PCAT and reveal example socio-technical challenges for group work support in medical domains. Such insight can help informatics researchers to account for the social-technical gap in group work support research for multidisciplinary healthcare researchers.

Positive user feedback in my studies led to an organizational decision to make the prototype of PCAT a usable tool for regular protocol development activities. This practical impact on SWOG's daily work practice is a valuable contribution.

8.2.2. Contributions to CSCW

The primary contributions of my work to CSCW include (1) an understanding of a lengthy collaborative writing process in the clinical research domain (2) the development of a new annotation model that supports the iterative reviewing and

revising process during collaborative writing (3) an insight into the challenges involved in group work among loosely coupled healthcare researchers.

Collaborative writing has been an active research area in the CSCW community for years. Collaborative clinical trial protocol writing is a unique example of lengthy collaborative writing processes in the healthcare arena. My work on this process contributes to the collaborative writing research area by showing that group writers can carry out the work in spite of their diverse views of the group work (Chapter 4).

In addition, annotation research has traditionally focused on annotation for static documents. I designed an annotation model that has these new properties:

(1) a dynamic activity status for annotations during collaborative writing processes that facilitates life cycle management of annotations and group awareness support for annotation-centered activities

(2) version information consistent with the evolving document that partly ensure the correct document context for annotations

(3) annotation addresses and temporal properties for annotation notification support with a purpose to reduce meta-communication, which is introduced in section 4.5.3

I identified the roles of comments during collaborative writing (Chapter 4) and provided implications for in-context communication support for collaborative

writers in an evolving information space. I suggested that such support should integrate research from multiple areas including design rationale capture, collaborative writing support, and annotation interface design.

Furthermore, I identified some challenges for group work support in the particular domain of clinical research due to the multidisciplinary nature of the group work and the loosely coupled group members. Protocol development is a great example of loosely coupled activities that has great research value for both medical informatics researchers and CSCW researchers.

8.2.3. Contributions to Clinical Medicine

Clinical trial protocol development is important for clinical research but barely receives appropriate technology support. Previous research has primarily focused on knowledge-based approaches for protocol design support and has paid little attention to augmenting the natural work process of clinical researchers.

This dissertation studied protocol development from CSCW perspectives and supported it using CSCW methods. My field studies helped clinical researchers understand their group work better and resulted in the design of PCAT. This system could potentially improve the iterative reviewing and revising of clinical trial protocols. Eventually, this research may lead to clinical trial protocols with better quality control, may speed up protocol development, and may ultimately improve the discovery process of disease treatments. My dissertation contributes to the ground work for collaboration support in the domain of clinical medicine.

8.3. Limitations

There are a few limitations of this research, including unsolved technical problems in PCAT and the scope of the field studies and the evaluation studies. I mentioned the unresolved technical problems in PCAT as engineering challenges in section 7.4.2; therefore, here I analyze the limitations in the scope of both the field studies and the evaluation studies.

8.3.1. Scope of the Field Studies

Protocol development involves multiple major roles including study coordinators (or principle investigators), protocol coordinators, and biostatisticians. During my field study and participatory design, I had good access to both biostatisticians and protocol coordinators but limited access to study coordinators. In Section 7.4.2, I described the possible reasons that make it difficult to engage study coordinators in the study. Study coordinators are a dynamic body of protocol developers, who may only write one or two protocols in their career and then switch to other commitments, and most of them do not share a consistent view of the protocol development process.

My research has a small sample size because the widely distributed organizational architecture imposes challenges for participant recruitment and most protocol writers do not often write protocols. Despite this, my work provided some baseline data for a larger scale of field study to fully understand the user needs of study coordinators in the future.

Last, my study has been limited to a single organization: SWOG. However, because SWOG is representative of NCI cooperative groups, I believe that some of my results can be generalized to other cooperative groups.

8.3.2. Scope of the Evaluation Studies

Groupware evaluation is complex in essence. In my work, I took a staged evaluation plan. Within the scope of this dissertation research, I focus my evaluations on the usefulness of the system designs. My field studies to evaluate the designs of the annotation model and the prototype system only spanned no more than a week; therefore, I did not study how the system prototype would be used in protocol developers' daily work. Adoption evaluation of group work support technology is still an open research topic, especially in the medical domains where social-technical challenges are complex. I hope to study the adoption of the designs as part of my future work.

8.4. Future Work

My research on collaborative clinical trial protocol writing support provides a basis for a series of research projects. Below I discuss several such projects.

8.4.1. Expanded Field Studies

SWOG is one of the cooperative groups funded by NCI. It is still unknown how different cooperative groups may carry out the protocol development process differently. Moreover, besides cooperative groups, there are lots of

pharmaceutical protocol development organizations, which also involve a lot of collaborative clinical trial design efforts and greatly demands group work support technology.

I would like to carry out more expanded field studies of protocol design in other cooperative groups or in other pharmaceutical protocol authoring agencies and to identify their similarities and differences. I would also like to carry out more longitudinal field evaluations of PCAT with more protocol developers from different protocol design groups in the future. Such studies would extend the usability evaluations in my current work and provide insights into the adoption of healthcare groupware technologies. They would help us understand the work support needs of protocol developers when we analyze their uses of the system, what they like and dislike, and the reasons.

8.4.2. Further Annotation Studies and Design

Annotations have very broad applications and great potential to support all sorts of collaborative activities. Physicians use annotations to give diagnosis comments on patient health records; researchers use annotations to label medical images or gene sequences, software engineers use annotations in programs, and so on. I plan to evaluate the annotation design from my dissertation in broader application settings and design more annotation-driven collaborative technologies. For example, I will investigate how physicians and nurses review and annotate electronic patient records as well as what major problems or challenges exist

when it comes to use these annotations. I plan to apply and improve my annotation designs for electronic medical record systems and to support information flow and collaboration among a variety of users.

8.4.3. Collaborative Environment Design

During my research, I got the opportunity to introduce the design of my annotation model and the PCAT prototype system to researchers in the field of technical communication and geographic information systems. They all liked the design and encouraged me that this system could be a good collaborative environment for students and researchers.

Both these fields produce well-structured documents such as research grant proposals or technical manuals that have strict formatting requirements similarly to those of clinical trial protocols. They also have similar document quality control processes to protocol designs through iterative reviews and revisions. One of my future plans is to extend the design on PCAT and apply it to facilitate group learning in classrooms or among distributed geographic informatics researchers.

8.5. Concluding Remarks

Collaboration among multidisciplinary healthcare professionals is common and calls for more groupware technology support. I have carried out a study to understand how collaborative clinical researchers work as a group and have

designed a new annotation technology to facilitate their loosely-coupled distributed collaboration.

My research in dissertation bridged research on annotations and collaborative writing and provided a basis for future collaborative medical informatics research.

List of References

Ackerman, M. S. (2000). "The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility." Human-Computer Interaction 15(2): 179–203.

Armitage, J., chair (2003).http://cis.nci.nih.gov/fact/1_4.htm

Bardram, J. (1996). The role of workplace studies in the design of CSCW systems. Proceedings of IRIS 19.613-629

Bargeron, D., A. Gupta, et al. (1999). Annotations for streaming video on the web. Conference on Human Factors in Computing Systems, Pittsburgh, Pennsylvania.278 - 279

Beck, E. E. (1993). A Survey of Experiences of Collaborative Writing: 9-28.

Beck, E. E., Bellotti, V.M.E. (1993). Informed Opportunism as Strategy: Supporting Coordination in Distributed Collaborative Writing. Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work (ECSCW'93).219-231

Beyer, H. and K. Holtzblatt (1999). "Contextual design." J interactions 6(1): 32-42.

Blomberg, J., L. Suchman, et al. (1996). "Reflections on a Work-Oriented Design Project." Human-Computer Interaction 11(3): 237-265.

Brickley, D. and R. Guha (2004). "Resource Description Frame-work (RDF) Schema Specification 1.0." W3C recommenda-tion, World Wide Web Consortium.

Brush, A. (2002). Annotating Digital Documents for Asynchronous Collaboration, PhD. Dissertation, University of Washington, Seattle, Pages. 119

Brush, A. J., Bargeron, D., Grudin, J., Borning, A., and Gupta, A. (2002). Supporting Interaction Outside of Class: Anchored Discussions vs. Discussion Boards. online proceedings of Computer Support for Collaborative Learning (CSCL)

Brush, A. J. B., D. Bargeron, et al. (2001). Robust annotation positioning in digital documents. Proceedings of the SIGCHI conference on Human factors in computing systems, Seattle, Washington, United States.285 - 292

Burrow, A. L. (2004). Negotiating access within Wiki: a system to construct and maintain a taxonomy of access rules. Proceedings of the fifteenth ACM conference on Hypertext/hypermedia.77-86

C. Sun, X. J., Y. Zhang, Y. Yang (1997). REDUCE: a prototypical cooperative editing system. Proceedings of the 7th International Conference on Human-Computer Interaction, San Francisco, USA.89-92

Cadiz, J., A. Gupta, et al. (2000). Using Web Annotations for Asynchronous Collaboration Around Documents. Proceedings of CSCW 2000, Philadelphia, PA.309-318

Campbell, D. G. (2002). The Use of the Dublin Core in Web Annotation Programs. Int. Conf. on Dublin Core and Metadata for e-Communities

Catlin, T., P. Bush, et al. (1989). InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration. Proceedings Applications in Writing.365-378

Churchill, E. F., J. Trevor, et al. (2000). Anchored conversations: chatting in the context of a document. CHI 2000.454-461

Cockton, G. (1999). Grounded Design: Integrating Models and Evaluations. Interacting in the large: Developing a framework for integrating models in HCI, ACM CHI'99 Workshop, Pittsburgh, USA

Conklin, E. J. (1996). Capturing Organizational Memory, White Paper, TouchStone Consulting Group, Inc.

Crabtree, A. (1998). Ethnography in Participatory Design. Proceedings of the Participatory Design Conference PDC'98, Seattle, WA.93-105

Crabtree, A. (2003). Designing Collaborative Systems: A practical guide to Ethnography.ISBN: 1-85233-718-4, Pages. 178

Davis, J. and D. Huttenlocher (1995). Shared Annotations for Cooperative Learning. Proceedings of Computer Support for Collaborative Learning '95, Indiana University, Bloomington, IN

Dourish, P. and V. Bellotti (1992). Awareness and coordination in shared workspaces. Proceedings of the 1992 ACM conference on Computer-supported cooperative work,, Toronto, Ontario, Canada.107-114

Ehn, P. and M. Kyng (1991). Cardboard computers: mocking-it-up or hands-on the future. Design at Work: Cooperative Design of Computer Systems. J. Greenbaum and M. Kyng: 169-195.

Ellis, C. A., S. J. Gibbs, et al. (1991). "Groupware: some issues and experiences." Communications of the ACM 34(1), Jan. 1991: 39-58.

Emery, F. E. and E. Trist (1960). "Socio-technical Systems." Management Science, Models and Techniques Vol. 2: 83-97.

Erickson, T., D. N. Smith, et al. (1999). Socially translucent systems: social proxies, persistent conversation, and the design of "babble". Proceedings of the SIGCHI conference on Human factors in computing systems, Pittsburgh, Pennsylvania, United States, ACM Press.72-79

Farkas, D. (1991). Collaborative Writing, Software Engineering, and the Universe of Collaborative Activity. Collaborative Writing in Industry. M. M. Lay and W. Karis, Baywood Publishing Company: 13-30.

Farkas, D. K. (1987). "Online Editing and Document Review." Technical Communication 34(3): 180-83.

Farkas, D. K., Poltrock, S.E. (1995). "Online editing, mark-up models, and the workplace lives of editors and writers." Professional Communication, IEEE Transactions on Volume: 38(Issue: 2): 110-117.

Fazi, P. (2002). WITH: a system to write Clinical Trials using XML and RDBMS. Proc of AMIA Symposium of 2002, San Antonio, TX.240-244

Fazi, P., D. Luzi, et al. (2001). Supporting writing and diffusion of clinical trials. Proc of 12th International Conference on Information and Intelligent Systems

Fischer, G. and B. Reeves. (1992). "Beyond intelligent interfaces: Exploring, analyzing, and creating success models of cooperative problem solving." Journal of Applied Intelligence 1: 311–332.

Fish, R. S., R. E. Kraut, et al. (1988). Quilt: a collaborative tool for cooperative writing. Conference on Supporting Group Work.30 - 37

Fox J, J. N., Rahmanzadeh A (1998). "Disseminating medical knowledge: the PROforma approach." Artif Intell Med. 1998 Sep-Oct;14(1-2): 157-81.

Franciosi, L. G., B. A. MacLeod, et al. (2001). A Web-Based Clinical Trial Expert System (WebCTES). ISAS, Information System, Analysis, and Synthesis Conference Proceedings of 2001

Garcia, A., H. Howard, et al. (1993). Active Design Documents: A New Approach for Supporting Documentation in Preliminary Routine Design, Stanford, Calif.

Garfinkel, H. (1996). "Ethnomethodology's Programme." Social Psychology Quarterly 59 (1): 5-21.

Glaser, B. G. and A. L. Strauss (1967). The discovery of grounded theory: Strategies for qualitative research. Chicago: AldinePages.

Green, R. (1997). Web review: a Web-based documentation review tool. Proceedings of the 15th annual international conference on Computer documentation, Salt Lake City, Utah, United States.93-98

Green, S. and D. Lowry (2004). Protocol Review Guidelines, Seattle, Washington. Pages 38

Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. Proc. CSCW 88, ACM Press.85-93

Grudin, J. (1996). Evaluating Opportunities for Design Capture. Design rationale: concepts, techniques, and use. Thomas P. Moran and J. M. Carroll. Mahwah, NJ, Lawrence Erlbaum Associates, Inc.

Haake, J. M. and B. Wilson (1992). Supporting collaborative writing of hyperdocuments in SEPIA. Proceedings of the 1992 ACM conference on Computer-supported cooperative work, Toronto, Ontario, Canada.138-146

Handschuh, S. and S. Staab (2002). Authoring and Annotation of Web Pages in CREAM. Proceedings of WWW conference.462-473

Hartswood, M., R. Proctor, et al. (2002). "Co-realisation: Towards a principled synthesis of ethnomethodology and participatory design." Scandinavian Journal of Information Systems 14(2): 9-30.

Horstmann, T. and R. Bentley (1997). "Distributed authoring on the Web with the BSCW shared workspace system." StandardView: 9-16.

Hughes, J., V. King, et al. (1994). Moving out from the control room: ethnography in system design. Proceedings of the 1994 ACM conference on Computer supported cooperative work.429-439

Jacobson, I. (1995). The use-case construct in object-oriented software engineering. Scenario-based design: envisioning work and technology in system development. J. Carroll, John Wiley & Sons, Inc. New York, NY,: 306-339.

Kahan, J., M.-R. Koivunen, et al. (2001). Annotea: An Open RDF Infrastructure for Shared Web Annotations. Proc. of the WWW10 International Conference, Hong Kong, ACM Press

Kaplan, B. (1997). "Addressing organizational issues into the evaluation of medical systems." J Am Med Inform Assoc. 1997 Mar-Apr; 4(2): 94-101.

Karasti, H. (1997). Bridging the analysis of work practice and system redesign in cooperative workshops. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques.185-195

Kensing, F. and J. Blomberg (1998). "Participatory Design: Issues and Concerns." Computer Supported Cooperative Work 7: 167-185.

Kim, H.-C. and K. Eklundh (2002). Collaboration between Writer and Reviewer through Change Representation Tools. Proceedings of the 35th Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society.38-47

Kim, H.-C. E. (2002). From Comments to Dialogues: A Study of Asynchronous Dialogue Processes as Part of Collaborative Reviewing on the Web. Proc

of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35'02), Big Island, Hawaii, IEEE Computer Society.116-125

Kim, H.-C. E. and K. S. Eklundh (2001). "Reviewing Practices in Collaborative Writing." Computer Supported Cooperative Work (CSCW) 10(2): 247 - 259.

Kunz, W. and H. W. J. Rittel (1970). "Issues as Elements of Information Systems." Institute of Urban and Regional Development, University of California, Berkeley, California Working Paper No. 131.

Lee, J. and K. Lai (1991). "What's in design rationale." Human-Computer Interaction 6(3-4): 251-280.

Lowry, P., Albrecht, CC, Lee, JD, Nunamaker JF (2002). Users experiences in collaborative writing using Collaboration, an Internet-based collaborative work. Proceedings of the 35th Annual Hawaii International Conference on System Sciences

Lunsford, A., Ede, Lisa (1992). Singular Texts/Plural Authors: Perspectives on Collaborative Writing. Carbondale: Southern Illinois UPPages.

Markus, M. (1983). "Power, politics, and MIS implementation." Comm ACM. 26(430-44).

Marshall, C. C. (1997). Annotation: from paper books to the digital library. Proceedings of the second ACM international conference on Digital libraries

McDonald, D. W., C. Weng, et al. (2004). The multiple views of inter-organizational authoring. Proceedings of the 2004 ACM conference on Computer supported cooperative work.564-573

Millen, D. R. (2000). Rapid ethnography: time deepening strategies for HCI field research. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techni.280-286

Mitchell, A., I. Posner, et al. (1995). Learning to write together using groupware. Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems (CHI'95).288-295

Modgil, S. and P. Hammond (2003). "Decision support tools for clinical trial design." Artificial Intelligence in Medicine 27(2): 181-200.

Muller, M. J. (2002). Participatory Design: the Third Space in HCI. The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, Lawrence Erlbaum Associates, Inc: 1051 - 1068.

Musen, M., J. Rohn, et al. (1987). "Knowledge engineering for a clinical trial advice system: uncovering errors in protocol specification." Bull Cancer 74(3):: 291-6.

Musen, M., S. Tu, et al. (1996). "EON: A Component-Based Approach to Automation of Protocol-Directed Therapy." JAMIA 3: 367-388.

NCI (2003).http://cis.nci.nih.gov/fact/1_4.htm

Neuwirth, C. M., D. S. Kaufer, et al. (1990). Issues in the Design of Computer Support for Co-authoring and Commenting. Proceedings of CSCW'90, ACM 1990 Conference on Computer-Supported Cooperative Work., Toronto, Canada.183-195

O'Day, V. L., D. G. Bobrow, et al. (1996). The Social-Technical Design Circle. Proceedings of the 1996 ACM conference on Computer supported cooperative work, Boston, Massachusetts, United States.160-169

Olson, J. S., G. M. Olson, et al. (1992). How a Group-Editor Changes the Character of a Design Meeting as well as its Outcome. Proceedings of the 1992 ACM conference on Computer-supported cooperative work, Toronto, Ontario, Canada.91-98

Ovsiannikov, I., M. A. Arbib, et al. (1999). "Annotation Technology." Int. J. Human-Computer Studies 1999(50).

Pankoke-Babatz, U., G. Mark, et al. (1997). Design in the PoliTeam project: evaluating user needs in real work practice. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques.277-287

Posner, I. and R. Baecker (1992). How People Write Together. Proceedings 25th Hawaii International Conference on System Sciences,.127-138

Pratt, W., M. Reddy, et al. (2004). "Incorporating ideas from computer-supported cooperative work." J Biomed Inform 2004 Apr;37(2): 128-37.

Reeves, B. and F. Shipman (1992). Supporting communication between designers with artifact-centered evolving information spaces. Proceedings of the 1992 ACM conference on Computer-supported cooperative work.394-401

Roscheisen, M., C. Mogensen, et al. (1997). Shared Web Annotations As A Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples. Technical Report CSDTR/DLTR

Sackett, D., R. Haynes, et al. (1991). Clinical Epidemiology: A Basic Science for Clinical Medicine. Boston, Little Brown Co.Pages.

Shapiro, D. (1994). The limits of ethnography: combining social sciences for CSCW. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW), Chapel Hill, NC, ACM Press.417-428

Sharples, M., Ed. (1993). Computer Supported Collaborative Writing. Computer Supported Cooperative Work, Springer-Verlag, Pages. 222

Suchman, L. A. (1987). Plans and situated actions: the problem of human-machine communication, Cambridge University Press,New York, NY, USA.ISBN:0-521-33137-4, Pages. 203

Suchman, L. A. (1995). "Making Work Visible." Communications of the ACM 38(9): 56-64.

Sumner, T. and S. Buckingham Shum (1998). From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. Proceesings of Human Factors in Computing Systems (CHI `98), Los Angeles, ACM Portal

SWOG (2004). How An Idea Becomes a Protocol: A Statistician's Perspective, Seattle, WA. Pages 3

SWOG (2004). SWOG Clinical Trial Protocol Development Guideline, San Antonio, TX. pages 28

Treese, W. (2004). "Open systems for collaboration." netWorker: 13-16.

Trigg, R. H., L. A. Suchman, et al. (1986). Supporting collaboration in notecards. Proceedings of the 1986 ACM conference on Computer-supported cooperative work.153-162

Tu, S. and M. Musen (1996). The EON model of intervention protocols and guidelines. Proc AMIA Annu Fall Symp. 1996.587-91

van der Lei, J. (2000). What's in a protocol. An Invitational Workshop: Towards Representations for Sharable Guidelines, Position Paper.

Vera Zaychik, W. C. R. (2002). "Capturing Communication and Context in the Software Project Lifecycle." Research in Engineering Design.

WALSH, J. P. and G. UNGSON, R. (1991). "Organizational Memory." Academy of Management Review 16: 57-91.

WebOpedia (2005).http://www.webopedia.com/TERM/A/annotation.html

Wolfe, J. (2002). "Annotation technologies: A software and research review." Computers and composition 19: 471-497.

Wolfe, J. L. and C. M. Neuwirth (2001). "From the margins to the center: The future of annotation." Journal of Business and Technical Communication; Thousand Oaks Vol. 15(Iss. 3): 333-339.

www.htmlarea.com (2005).

Appendix I: Questions in Semi-Structured Interviews

Questions about the work practice

1. Could you please describe the role you play? What do you do with developing protocols? What are your responsibilities?

2. How many people working as the same role are there in SWOG? How many protocols do they work simultaneously?

3. How and at what time stages are you involved in the protocol development process?

4. Could you please use a diagram to capture your view of the protocol development process? What I want are the major steps, people and tasks involved in each step.

5. What sections do you participate in reviewing and writing?

6. In your view, do you see any problems or challenges in the current protocol development model? If yes, what are they?

Questions about communication

7. Who in the protocol development team do you contact? Are you co-located? Are you familiar with each other?

8. What are the major communication channels with other protocol developers? How often do you use email, phone calls, face-to-face meetings, etc?

9. In this profession, how do people view different communication channels? Which one is the most formal one? Which one is the mostly used one?

10. Do you have a preference for a specific communication channel for a particular circumstance? For example, would you prefer email rather than phone calls most of time or sometimes?

11. How do you like email communications? Can you talk about its pros and cons for the reviewing tasks? Similarly, how about face-to-face meetings or phone calls?

Questions about Review Comments

1. Could you please describe your protocol review process? Suppose one minute ago you receive a protocol for you to review. Could you just talk about your review process and how you share your comments with others?

2. After comments are sent out, what happened to these comments? I am interested in all iterative activities in the review process.

3. Does it happen that people may give different comments on the same issue? How do you resolve that? Who are the decision makers?

Appendix II: An Sample Change Management Document

Things that will change only minimally

- The workflow among statisticians, study coordinators and protocol coordinators
- Working styles: Dealing with comments, putting together documents, email & telephone habits

Things that will change

- Handling reviewers comments
    - *Protocol reviewers can add comments directly to the protocol documents without specifying in detail where the comment message applies; the system will automatically capture the anchor for every comment. This is also helpful for detecting conflicting opinions sent from different reviewers*
    - *Comments will be made available immediately online to all the protocol writers. There is no need to send comments in emails, which is hard to track*
    - *Comments can be organized as threaded discussions*
    - *Reviewers can specify particular people that they want the comment to be read and responded. This is support for "direct message"*
    - *Changes made to the protocols can be related to comments so that reviewers can understand how their comments are received by protocol authors and what effect they have on the document*

- Improved progress tracking for protocol manager
    - *The system is going to support progress tracking. She can overview the status of comments and the version history of the development protocol*
    - *The system can send automatic reminders to related persons who are behind the work*
    - *Ideas? We're not sure what is needed in this area*

- Smaller workload for protocol coordinator
    - *Comments can be addressed to a specific person as a "directed message"; the protocol coordinator will be saved from relaying questions and answers as a middle player*

- *Protocol authors such as PIs or statisticians have the option to write or insert their sections online, instead of emailing the content in word document attachments to the P.C.*
- *Keeping track of different document versions is easier and will be taken care of by the system. But we need help understanding what method for versioning would be best for PCs!*

- Benefits for the whole group
  - *Most communications will change from personal emails to web-based and document-centered discussions*
  - *Every comment has a life cycle going through "newly added", "responded" and "resolved". This allows writers to understand how comments are received and where new versions come from*
  - *Everybody shares the same version of online protocol; there will be less chance for people to work on different versions*

Where we are now? Web prototype demo, A complete workflow demo, which includes the following activities:

- *A protocol coordinator creates a new protocol based on a default template online*
- *The leading statistician or the study coordinator adds input for their specific sections for the new protocol online*
- *The protocol coordinator makes sure the protocol draft is ready for review and sends notifications to the group*
- *People use the collaborative reviewing tool and add in-line comments to the current version of the protocol. People can respond to one another's comments.*
- *The protocol coordinator takes resolved comments, makes revisions to the draft, and saves it into a private version. Once there are no further comments for a given draft and all the comments have been incorporated into the revision, the protocol coordinator creates and releases a new version online. (This is a proposal to be confirmed by Dana: Please tell us if this approach works for PCs. Currently everybody can make revisions online)*
- *Reviewers can continue to add comments on the new versions. They can also load old versions. However, old versions are Read-Only to reviewers. (This functionality is under construction)*
- *The protocol manager can browse the version evolutions; they can also check the status of all comments such as "how they have been processed" and "which comments still remain unresolved".*

Comparison with DocuMart

- They don't have concrete solutions for communication support
- They do have materials and ideas for template construction and management

Appendix III: A Likert Survey of System Usefulness

Please put a '*" in the cell where you think it would fit.

1. **It is very useful (or I can foresee its usefulness after the editor is consistent with Word)**
2. **Sometimes useful**
3. **Occasionally useful**
4. **I have not used it at all**
5. **I won't use it**

Feature	1	2	3	4	5	Other comments
Reply to other's comments						
Browse comment status						
Incorporate a comment						
Filter comments						
Output print-friendly comments						
Address comments to certain people						
Set up comment response deadline						
Link comment to certain text in the document						
Comment categorization						
Urgent email notification for high priority comment						
Highlight comments in color on the text						
See other people's comments during my review						
Sort comments on different criteria: submission time, user, category, etc.						
View comments side by side to text						
View comments as threaded discussions						

Appendix IV: An Example Evaluation Scenario Script

Roles: Deepa: a real protocol coordinator. **Tasks to carry out**: Group activity tracking, group communication, and comment making and replying, protocol revising based on comment, protocol version control

Deepa logs into the Electronic Protocol System. She browses group activity history and finds that her group members have added some input to different sections. Then she wants to complete some sections based on input from the PI and stats. However, she runs into several questions. She notices that both of them are not online at the moment; therefore, she communicates her questions to the PI or the stats by using either the web mail or adding annotations to the text written by them. After ten minutes, the PI comes back and logs into the system, she checks the "comment history" and finds that there are newly added comments, addressed to her from Deepa. The PI knows the answers to these questions and quickly responds to the comments. Deepa immediately detects responses by noticing status transitions of her early comments, from "unread" to "responded". With these answers, she edits corresponding sections. While editing, she can view comment details and change the status of comments to "incorporated" once she make the change. She saves changes after she is done with each section. Then she saves a new version for the whole protocol and logs out of the system.

Vita

Chunhua Weng

Research Interests

I am broadly interested in Biomedical Informatics research that supports the clinical research life cycle, user-centered information system design, and annotation-driven biomedical applications.

Education

University of Washington, Seattle, WA August 2005
Ph.D. in Biomedical and Health Informatics
Advisor: Dr. John H. Gennari. Other committee members: Dr. Jonathan Grudin, Dr. Ira Kalet, Dr. David W. McDonald, and Dr. David Farkas.

University of California, Irvine, CA June 2002
M.S. in Information and Computer Science

Nankai University, P.R. China. July 1997
B.S. in Computer Science

Journal Publications

1. J.H. Gennari, C Weng, DW. McDonald, Asynchronous Communication among Clinical Researchers: A Study for System Designs, International Journal of Medical Informatics.(To Appear)
2. C Weng, DW. McDonald, JH Gennari, Participatory Design of a Collaborative Clinical Trial Protocol Writing System. International Journal of Medical Informatics. (To Appear)

Conference Publications

(A) PCAT Collaborative Writing System (part of Chapters Six & Seven)

3. C Weng, Why is it hard to support group work in distributed healthcare organizations: empirical knowledge about of the social-technical gap, Proc of AMIA 05, Washington DC, October, 2005. (Accepted)
4. C Weng, J.H. Gennari, DW. McDonald, A Collaborative Clinical Trial Protocol Writing System, 11th World Congress on Medical Informatics (MedInfo'04), San Francisco, CA. November 2004, 1481-6.

(B) An Annotation Model (part of Chapter Five)

5. C Weng, JH. Gennari, Asynchronous Collaborative Writing through Annotations, Proc of ACM Conference on Computer Supported Cooperative Work (CSCW'04), November 2004, Chicago, IL, 578-81.
6. C Weng, Annotation and Asynchronous Collaborative Writing, CSCW'04 Doctoral Consortium, November 2004, Chicago, IL, Conference Supplement.

(C) Field Knowledge about Protocol Development (part of Chapter Four)

7. C Weng, J Benedetti, D Sparks, J McCoy, JH Gennari, Understanding the Group Work of Creating a Clinical Trial Protocol, 26th anniversary meeting of the Society for Clinical Trials, May 22-25, 2005, Portland, OR, abstract accepted.
8. DW. McDonald, C Weng, J.H. Gennari, The Multiple Views of Inter-organizational Authoring, Proc of ACM Conference on Computer-supported Cooperative Work (CSCW'04), November 2004, Chicago, IL, 564-73.
9. JH. Gennari, C Weng, DW. McDonald, J. Benedetti, S Green, An Ethnographic Study of Collaborative Clinical Trial Protocol Writing, 11th World Congress on Medical Informatics (MedInfo'04), San Francisco, CA. November 2004, 1461-5.

(D) Participatory Design (part of Chapter Three)

10. C Weng, DW. McDonald, J.H. Gennari, Participatory Design of A Collaborative Clinical Trial Protocol Writing System, IT in Health Care: Socio-technical Approaches 2nd International Conference, 13-14 September 04, Portland, Oregon.
11. C Weng, JH. Gennari, DW. McDonald, Scenario-based Participatory Design of a Collaborative Clinical Trial Protocol Authoring System. Proceedings of the American Medical Informatics Association Fall Symposium (AMIA'03). Washington DC. November, 2003, 1051. (Abstract)

(E) Knowledge-based Approach to Protocol Development Support

12. C Weng, M Kahn, JH Gennari, Temporal Knowledge Representation for Scheduling Tasks in Clinical Trial Protocols. Proceedings of the American Medical Informatics Association Fall Symposium (AMIA'02). San Antonio TX. Nov. 2002, 879 – 883.

Professional Presentations

1. Supporting Collaborative Clinical Trial Protocol Writing through an Annotation Design, General Exam Defense at UW. Jan. 2005
2. Asynchronous Collaborative Writing through Annotations, conference presentation for [5], Nov. 2004
3. Annotation and Asynchronous Collaborative Writing, conference presentation for [6], Nov. 2004
4. A Collaborative Clinical Trial Protocol Writing System, conference presentation for [7], Sep. 2004
5. Design the right system right: A Participatory Design, conference presentation for [11], Sep. 2004
6. Mapping Informatics into Various Medical Fields: Opportunities and Challenges, research presentation at the biomedical informatics program at UW, Aug. 2004
7. Collaborative Information Integration through Annotations, research presentation at the biomedical informatics program at UW, Mar. 2004
8. Annotation: Uses, Users, and Technology, Dissertation Literature Review at UW, December. 2003
9. Temporal Knowledge Representation of Clinical Trial Protocols, conference presentation for [10] Nov. 2002

Research and Development Experiences

Research Assistant in Biomedical and Health Informatics at University of Washington, Seattle, partially sponsored by the Southwest Oncology Group. Apr. 02 – Sep. 04
Carried out a qualitative research on the collaborative clinical trial writing process within the Southwest Oncology Group, designed and implemented a new annotation model and a collaborative clinical trial protocol writing system.

Research Assistant in Biomedical and Health Informatics at University of Washington, Seattle. Jan. 02 – Apr.02
Studied the Foundational Model of Anatomy (FMA), proposed an evaluation plan for this large knowledge base, and explored ontology alignment approaches.

Research Assistant at Information & Computer Science at UC-Irvine. Aug. 00 – Dec.01

Designed a temporal knowledge representation for clinical trial protocols using Protégé 2000, and developed a Protégé plug-in that schedules patient visits based on study calendar information from clinical trial protocols.

Research Assistant at Intelligent Robotics Lab, Nankai University. Aug.98 – Jun.00
Designed a statistical model to measure the average command transmission delay of TeleRobots due to unstable Internet traffic.

Research Assistant at Computer Science Department, Nankai University. Jul.96 – Jul.97
In collaboration with three other students, designed a computer-based examination system for The Adult Education Committee of Tianjin, P.R.China., responsible for module integration and user interface design. In June 1996, the system was successfully used for testing the computer literacy of about 1,000 adult learners in Tianjin.

Software Developer at Nankai Sun Co. Ldt, Tianjin, P.R.China. Jul. 97 – Aug. 98
Involved in the design of commercial web-based information systems, mainly responsible for database transaction processing and user interface design.

Honors

Selected to participate in the Doctoral Consortium of ACM CSCW'04 Conference
Outstanding student honor and scholarship awarded by Nankai University 94—97

Professional Service

Reviewer for ACM SIGGROUP'05 conference 2005
Reviewer for ACM SIGCHI'05 conference 2005
Reviewer for ACM CSCW'04 conference 2004
Admission Committee Member of BHI Graduate Program at UW 2003

Professional Affiliations

Member of American Medical Informatics Association (AMIA) 2000—present