Asynchronous Collaborative Writing through Annotations

Chunhua Weng
Biomedical and Health Informatics
University of Washington
1959 NE Pacific St. Box 357240
Seattle, WA 98195
cweng@u.washington.edu

John H. Gennari
Biomedical and Health Informatics
University of Washington
1959 NE Pacific St. Box 357240
Seattle, WA 98195
gennari@u.washington.edu

ABSTRACT
Annotation is central to iterative reviewing and revising activities in asynchronous collaborative writing. Currently most digital annotation models and systems assume static context information and provide far less functionality than physical annotations. We extend prior annotation research by Marshall and Cadiz and design an activity-oriented annotation model to mimic the rich functionality of physical annotations for an enhanced collaborative writing process. In this model, we define an annotation life cycle and support annotation version control. We implement a collaborative writing system that supports improved in-situ communication and cross-role feedback based on our annotation model.

Categories and Subject Descriptors
H.5.3 [**Information Interfaces and Presentation, HCI**] Group and Organizational Interfaces – *Asynchronous interaction*, *Computer-supported cooperative work*

General Terms
Design, Human Factors

Keywords
Asynchronous Collaboration, Annotation, Collaborative Writing

1. INTRODUCTION
Asynchronous collaborative writing is ubiquitous but fraught with many problems. An essential part of iterative reviewing and revisions in collaborative writing is annotation – where text is marked up with comments and meta-information by participants in the collaboration. Annotation has been well recognized as an important approach to document-centric discussions [2]. In collaborative writing, however, it is often poorly supported. Much time delay, communication overhead, and cognitive burden is added to collaborative annotation and annotation incorporation [6].

Prior research conducted by Marshall and Cadiz has motivated our research. Marshall identified a rich set of functionalities of physical annotations [8]. These functions include: 1) annotations as bookmarks 2) annotations as in-situ locations for problem-working 3) annotations as a record of interpretive activity 4) annotations as a visible trace of the reader's attention 5) annotations as incidental reflections of the material circumstances. However, many of these annotation functions, such as problem solving and procedural signaling, are neither supported in current digital annotation frameworks nor in any collaborative writing system. We think Marshall's results imply that digital annotations empowered with these functions could facilitate in-situ communication and decision-making (item #2), evolving design rationale capture (item #3), and progress tracking (item # 3, 4).

Besides having fewer functions than physical annotations, existing digital annotation systems also encounter many real-world challenges in collaborative writing activities. Cadiz conducted a large case study of digital annotation uses among a group of software specification document writers [3]. He identified the following major problems that impede the uses of digital annotations in collaborative writing processes, including 1) technical orphaning of annotations in evolving documents 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. One major reason behind these problems is that existing annotation frameworks or systems only support static annotations and assume that the referenced artifact will never change. This is not true in collaborative writing because shared drafts evolve over time and some annotations become irrelevant after they have been incorporated.

Taken together, we believe that the results of Cadiz and Marshall suggest a need for work that improves digital annotation in collaborative writing settings. Annotations are a central media that connect different roles and evolving artifacts in a collaborative writing process and they are more than static labels. We hope to empower digital annotations with the rich functionality of physical annotations and to solve the problems identified by Cadiz et al. Therefore, we have designed an annotation model and a collaborative writing system based on this model. We believe that a collaborative writing system with appropriate annotation support can help us achieve more effective group awareness, in-situ communication, and group coordination in asynchronous collaborative writing. This paper details our design and briefly describes our initial implementation of such a system.

2. REAL-WORLD PROBLEMS
Many technology-driven collaborative writing systems fail to match user needs in real-world collaborative writing processes. To avoid this pitfall, we have conducted a thorough qualitative study of the collaborative clinical trial protocol writing processes in the Southwest Oncology Group (SWOG) using interviews, observa-

CSCW'04, November 6–10, 2004, Chicago, Illinois, USA.

tions, and artifact collections; the details of this study are presented in [6]. The writers in our study are distributed across the country and do not have groupware support. We found four major types of communication and coordination problems.

First, collaborative writers communicate with each other in a variety of ways, such as emails, phone calls, and face-to-face meetings. Therefore, decision-making and communication are often disconnected from writing itself. Valuable information may be buried in these communications and cannot easily be recovered later. There is also significant communication overhead since participants have to reconstruct the context for any retrieved communication.

Second, traditional version control methods require users to communicate or explain changes, especially design rationales to group members explicitly. In iterative reviewing and revising processes, it is hard to manually trace how review suggestions are incorporated into a new version of the evolving draft. There is little or slow responsiveness across the roles in the collaboration. Authors and editors can easily develop misunderstandings since authors often have no idea why editors change the document in a particular way. This observation also matches work carried out by Farkas [5].

Third, group discussion and coordination are poorly supported. Sometimes two reviewers provide conflicting revision suggestions; editors may overlook such conflicts until after they have made changes to the document. Group writers often cannot get timely and appropriate notifications of group progress.

Finally, we find that writers use comments for a wide variety of purposes, such as asking questions or suggesting a change to the draft. Comments can lead to discussions or revisions. However, because these writers do not have a good annotation tool, they write comments in a separate note and email the note to other group members. Hence many emails are also sent that report on the problem-solving progress for issues raised in comments. Therefore, communication overhead around comments is large. When participants develop multiple versions of the source document, they can easily get lost about which comments apply to which version.

We claim that existing annotation tools do not sufficiently address the above problems because their underlying annotation models do not record changes to annotations. For example, a representative annotation system, Web Discussion, does not support version control and has the annotation orphaning problem [3]. All existing annotation data models, including RDF [1] and Dublin Core [4], only define static properties of annotations and do not capture relevant information about activities on annotations and consequent changes on annotations throughout the collaborative writing process.

Based on our fieldwork, we designed a new annotation model oriented to collaborative writing with the following features to address the above problems:

1. ***Support in-situ communication and decision-making*** We hope to reduce the cognitive overhead in communication among collaborative writers by integrating communication and decision-making via threaded annotations in the context of the document. These threaded annotations also help users detect conflicting opinions and encourage group discussions.
2. ***Improve cross-role awareness*** Efficient group work relies on smooth transitions or interactions between work done by people of different roles, such as reviewer and author. Collaborative writers need to get timely cross-role responsiveness to achieve effective interactions.
3. ***Provide a rational version control mechanism*** In collaborative writing, writers' understanding of the problem and design rationale evolves together with the shared draft. Annotations represent opinions of writers and convey revision rationales. Capturing integrated version history and annotations leading to revisions could help group members make better sense of how the draft evolves.
4. ***Improve shared workspace and group awareness*** Writers should concentrate on content and writing. If more information about progress could be shared in the group automatically, less communication effort would be needed.

Next, we describe our design of such an annotation model and an asynchronous collaborative writing system based on this model.

3. AN ANNOTATION MODEL

Although annotations have been used widely, they are often viewed as static labels. To support group dynamics, we think that it is important to define dynamic properties in an annotation model. Specifically, an annotation object should include dynamic context and status properties. First, an annotation is attached to certain text in an evolving draft, and the context, the "anchor text", may change over time. If the annotation is part of a discussion, then the communication context may change while the discussion is evolving.

Second, we think that an annotation in a design or writing process has 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, writers might carry less cognitive overhead by using status information when tracing annotation incorporation progress. Finally, status could facilitate annotation management. It is unnecessary to robustly anchor every annotation to a new draft in collaborative writing if an annotation has been incorporated; instead, a system should know that which annotation is irrelevant. Therefore, status information could help manage "annotation overflow" problem in an evolving design process and changing information space.

In greater detail, our annotation model includes 11 features:

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*. Especially in iterative writing processes, it is vital to specify in which version of a shared document an annotation applies. The second property is *the text context* within the document. It helps readers pinpoint the surrounding text for an annotation easily. It can also be used to support "document-centered discussions": if annotations are discussions, writers can see discussions and document content side by side. The third property is *the discussion context*, which references previous annotations that may have triggered this new annotation. This information helps position an annotation in a threaded dis-

cussion and helps collaborative writers understand conversational contexts.

2. ***Message Body***. This is the actual text message of a comment.
3. ***Annotation Creator***. This feature describes information about the annotation creator. In addition to recording the name of the creator, we propose capturing their role and status. Some experts' annotations could be more important than others'. Therefore, information about the role could help identify authority or prioritize multiple user opinions.
4. ***Annotation Recipient***. This feature represents information about people to whom an annotation should be delivered. Systems can use this information to provide automatic notification services through emails or reminder messages.
5. ***Annotation Time***. This feature can be used not only to indicate when an annotation is created.
6. ***Response Deadline***. This feature specifies the time by which the annotation should be addressed. It could be used by notification service to facilitate prioritization of annotations.
7. ***Responses***: This is a multi-value feature, as an annotation can receive multiple responses. This information is useful for forming threaded discussions.
8. ***Status***. The transitions of annotation statuses enable collaborative writers to know what happened 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 our collaborative writing support purposes, we define three statuses for an annotation: "unread", "responded", and "incorporated". With status information, reviewers can stay aware how their opinions are received by other reviewers, writers, and editors; writers can indicate their work progress by changing the status of annotations to "incorporated". There would be improved cross-role feedback among them.
9. ***Category*** This feature captures the characteristics of the problems brought up in an annotation. Example categories are "incompleteness", "inconsistency", "question", etc.
10. ***Rating*** An annotation can be labeled as either "positive feedback" or "negative feedback".
11. ***Urgency*** This property sets up the priority for annotations. Collaborators may want to process only the most important annotations in a limited time period.

Most prior annotation schemes or models use annotations to support static activities such as information retrieval or integration. Compared to existing annotation data models such as RDF and Dublin Core Metadata Element Standard, our model has the following major additions: 1) annotations are associated with a life cycle with status information, which supports progress tracking and provides cross-role feedback among reviewers and authors; 2) annotations have extended activity-oriented properties such as rating, category of problems, response deadline, etc; and 3) annotations have a richer context information, including not only a versioned physical text anchor in the document but also contextual threaded discussions. Through these features, our annotation model could solve the "annotation orphaning" problem and removes irrelevant annotations for evolving drafts.

4. SYSTEM DESIGN

4.1 System Infrastructure

Figure 1 depicts our system infrastructure. We provide a two-level logical structure for an asynchronous collaborative writing system based on our annotation model. The bottom level is the content layer, which includes co-evolving annotations and documents. The top level is the service layer, which supports in-situ discussion & decision-making, rational version control, awareness, and progress tracking. All collaborative writers access content through the service layer.

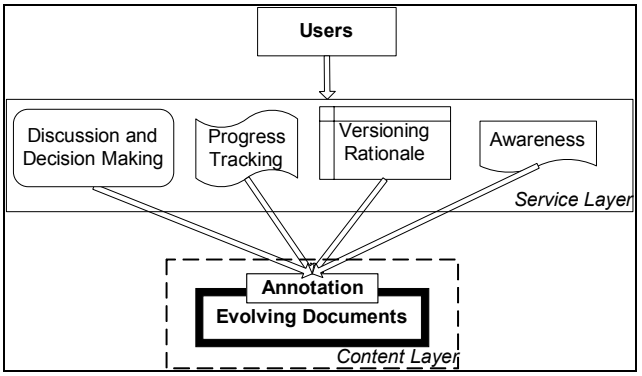

Figure 1. System Infrastructure

4.2 Collaboration through Annotations

To increase the accessibility of the system, we implement the system using web-based .NET services. The implementation and evaluation details of this system are presented in [9]. Our goal with this system is to make it a document design environment that integrates communication, version control, and project management features. We provide a rich-text web editor, a web-based annotation system, and other utilities such as web-email and user access control. Below are system highlights that support asynchronous collaboration through annotations.

4.2.1 In-situ Discussion and Decision-Making

A user can open a web document, freely select some text and create an annotation, filling in values for the properties listed in section 3. Some information such as annotation creator and context is automatically populated by the system. We store annotations and documents separately; therefore, annotations can be attached to or detached from documents flexibly. When annotations are attached to documents, their anchor text is highlighted in color. Each writer can select a unique color to highlight his annotations. A user can reply to annotations made by others; annotations then form threaded discussions, which help avoid or detect conflicting opinions.

4.2.2 Progress Tracking

Reviewers make annotations and authors or editors incorporate annotations. Therefore, the statuses of annotations indicate the reviewing and revising progress. The system also includes a process model with a list of to-do tasks. Completed reviewing and versioning activities will be recorded by the system and mapped to the to-do list; every group member can track the project progress by browsing the statuses of to-do tasks on the list. In addi-

tion, group writers can check the status for any annotation and check where it is in its life cycles. Our participatory design users tell us that this is very important because it helps them easily stay aware of how their comments are received by others without sending emails to nudge their colleagues explicitly.

4.2.3 Rational Version Control

A version control system assigns a version id to every document and annotation. When a user wants to track changes across versions, he can see annotations that are incorporated into version N; the revision suggestions contained in these annotations can be used as a "revision rationale" for version N. In this way, the system keeps not only version history, but also the rationale behind changes. This versioning rationale also accounts for "process knowledge", which helps group writers understand why a particular opinion may not have been accepted.

4.2.4 Group and Cross-Role Awareness

A floating monitoring window provides peripheral awareness of group activities to all users. Users can peek at recent activities on documents and annotations. In addition, users can subscribe to email notifications for the following events: 1) status transitions of specific annotations; 2) annotations made by a specific user; 3) newly added annotations in the past 24 hours; 4) newly created versions; and 5) annotations that define "me" as a recipient. A stored procedure in the annotation database sends out notifications based on the above different filters. It also sends out regular notifications to recipients of those annotations that have not been taken care of for two weeks or longer and to authors when they receive new annotations for their writings. When a new version is created, an email notification from the system is sent to group members with explanations of what revisions are made to create this version based on a summary of resolved annotations associated with the prior version. We hope that these awareness features will save users' effort by avoiding explicit email communication.

5. DISCUSSION

Current annotation design research falls into two major categories: one is the development of annotation data schema, such as [1], and the other is the design of annotation systems with focus on annotation for static information, such as [2]. 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. This research tries to fill in this gap by designing an activity-oriented annotation model. We have grounded our design on a qualitative study of asynchronous collaborative writing processes. Therefore, we hope to integrate disconnected research on asynchronous collaborative writing and research on annotation.

Rather than building application-level capabilities, we provide a data-level structure: our annotation model. This model includes task-specific features such as category, life cycle status, and changing context. These features help answer the call for better design of a generic annotation framework by Grudin [7]. We define life cycles status and dynamic context information for annotations; we also support in-situ communication and problem solving, evolving design rationale capture, and progress tracking with this annotation model. We hope this model bridges the gap between the functionality of digital and physical annotations.

Collaborative writing needs effective group coordination and progress tracking support. Our system collects information about annotation status and group activity history and combines them with manually reported activities to generate a progress report. Moreover, we provide cross-role feedback so that reviewers can check how their annotations are received by others. In addition, we hope 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. We hope to alleviate human cognitive overhead when they search for information. A physical annotation can be thrown away easily once we do not need the message on it. In our system, we 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.

In sum, we hope that our annotation model can increase the richness of digital annotation and lead to systems that facilitate enhanced collaborative activities in an evolving information space.

ACKNOWLEDGMENTS

We particularly thank SWOG users involved in this research and Dr. John Crowley for funding this research over the past year.

6. REFERENCES

[1] Brickley, D., Guha, R. (2004). Resource Description Framework (RDF) Schema Specification 1.0. W3C recommendation, World Wide Web Consortium.

[2] Brush, A.J. (2002). Annotating Digital Documents for Asynchronous Collaboration, PhD. Dissertation *Department of Computer Science and Engineering*, University of Washington, Seattle, 119.

[3] Cadiz, J.J., Gupta, A., Grudin, J. (2000). Using Web Annotations for Asynchronous Collaboration Around Documents. *Proceedings of CSCW 2000*, Philadelphia, PA. 309-318.

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

[5] 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.

[6] Gennari, J.H., Weng, C., McDonald, D.W. (2004). An Ethnographic Study of Collaborative Clinical Trial Protocol Writing. *Proc of MedInfo'04*, San Francisco, CA. *(in press)*

[7] Grudin, J., Bargeron, D. (2004). Communication and Collaboration Support Systems. in K. Okada, T.H., and T. Inoue (Eds.) ed., Ohmsha. *(in press)*

[8] Marshall, C.C. (1997). Annotation: from paper books to the digital library. *Proc of the 2nd ACM international conference on Digital libraries*, 131-140.

[9] Weng, C., Gennari, J.H., McDonald, DW. (2004). A Collaborative Clinical Trial Protocol Writing System. *Proc of MedInfo'2004*, San Francisco, CA. *(in press)*