

Co-Authoring with Structured Annotations

Qixing Zheng, Kellogg Booth, and Joanna McGrenere

Department of Computer Science, University of British Columbia

{qzheng, ksbooth, joanna}@cs.ubc.ca

ABSTRACT

Most co-authoring tools support basic annotations, such as edits and comments that are anchored at specific locations in the document. However, they do not support meta-commentary about a document (such as an author's summary of modifications) which gets separated from the document, often in the body of email messages. This causes unnecessary overhead in the write-review-edit workflow inherent in co-authoring. We present document-embedded structured annotations called "bundles" that incorporate the meta-commentary into a unified annotation model that meets a set of annotation requirements we identified through a small field investigation. A usability study with 20 subjects evaluated the annotation reviewing stage of co-authoring and showed that annotation bundles in our high-fidelity prototype reduced reviewing time and increased accuracy, compared to a system that only supports edits and comments.

Author Keywords

Collaborative writing, collaborative authoring, structured annotation, usability study, workflow

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI)

INTRODUCTION

Co-authoring academic papers, books, business reports, and even web pages is common practice [18]. Word processors and other tools provide some support for collaborative authoring, but not as effectively as we might desire. Much of the effort in collaborative writing is spent reviewing and editing drafts [14]. Typical workflow involves co-authors annotating drafts and passing them back and forth. Basic annotations are edits (insertions and deletions) and comments on specific parts of the document, but co-authors also communicate at a meta level about a document, for example, by making suggestions to change the document tone, clarifying previous annotations, or responding to other co-authors' document-related questions. We use the term "co-authoring" to refer to this entire writing-reviewing-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. *CHI 2006*, April 22-27, 2006, Montréal, Québec, Canada. Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

editing cycle. While the purpose of annotations ranges from strictly personal [11] (fine-grain highlighting to aid memory) to more communal (comments or questions for co-authors), Neuwirth suggests that the most important purpose of shared annotations is the fine-grained exchanges among co-authors creating a document [13]. We present a novel framework for co-authoring that fully integrates annotations into a document and introduces structured annotations that explicitly support workflow management within the co-authoring cycle.

Co-authors often make basic annotations using their word processors and then send the revised document to fellow co-authors via email as attachments, pathnames in a shared file system, or URLs on the web. This is usually done asynchronously, one author at a time. The annotate-and-email sequence is repeated until the document is completed. The meta-commentary often takes place outside the document, usually in the bodies of the emails that are used to send the drafts back and forth. This is problematic because it requires that co-authors maintain collaboration artifacts in different places (word processor files and emails) with no formal association between the two, which unnecessarily complicates workflow. Valuable information can be buried and easily forgotten or misplaced [19]. Even if the appropriate emails are located, depending on the nature of information communicated, it can be difficult to navigate between the email and the document content.

The focus of our research is small, distributed groups collaborating asynchronously during the editing and reviewing stages of co-authoring, working on documents with a large volume of embedded annotations. We began with a small field investigation of document annotation requirements. We then designed document-embedded structured annotations called "bundles" that incorporate the meta-commentary into a unified annotation model that meets the requirements we identified. A usability study with 20 subjects evaluated the annotation reviewing stage of co-authoring and showed that annotation bundles in our high-fidelity prototype reduced time and increased accuracy, compared to a system that only supports edits and comments.

RELATED WORK AND TERMINOLOGY

Annotations have evolved from paper-based to digital. The term "annotation" itself carries many different meanings. Marshall [12] classified paper-based annotations into four categories depending on their content type and locations. Similarly, Brush et al. [3] defined digital annotations to be

markings made on a document at a particular place, with each annotation having two components: an *anchor* and *content*. Fish et al. defined annotations to be hypertext nodes [8] that are linked to the base document. Ovsianikov et al. [15] proposed the idea of “clumps,” which are comments that can anchor at multiple places in the document, similar to a type of bundle that will be discussed later. None of these definitions extend beyond simple editing (insert, delete, or replace) or comment annotations.

There is no standardized annotation model, especially no agreed-upon convention for structuring annotations. Previous research [5, 9, 16] has identified various attributes for annotations such as class, type, title, context, id, time, annotator, status, and priority. Weng and Gennari developed an eleven-attribute annotation model [19] that uses annotation *status* to support awareness of in-progress reviewing and revision activity. Unlike previous annotation data models for describing websites’ metadata [9], their annotation model is process-oriented, designed to streamline workflow by keeping track of what has been done. It is the only one we are aware of that allows annotations to be anchored to the entire document; most models assume that annotations will be anchored at a particular location within the document.

Various tools support collaborative authoring. Brush [2] reviewed some of these annotation systems with a focus on issues such as online discussion in educational settings, notification strategies, and annotation re-positioning techniques in an evolving document. We review systems from the point of view of how well they support collaborative authoring workflow.

Noël and Robert studied 42 users in May 2001 [14] and found that most used individual word processors and e-mail as their main co-authoring tools. Eighty-three percent of the subjects used Microsoft Word. Word integrates editing and comment annotations into the document and assigns attributes automatically. Annotations can be filtered by author or by type (formatting changes, comments, insertions, or deletions). All annotations are listed in a reviewing pane below the document pane, ordered by position in the document. Word incorporates an annotation into the document once it is accepted by one co-author, so other co-authors might not know it existed after the document is saved. Word has a Web Discussion function for collaboration; however, Cadiz notes that it is limited in terms of where annotations can be anchored [4].

In contrast to Word, annotations in Adobe Acrobat Professional do not alter the document. This must be done manually after reading the annotations. Status indicators and more sophisticated filtering by annotation type, reviewer, or status are provided. The reviewing pane in Acrobat uses a threaded display, not simple document order, so replies to an annotation are listed indented and below the original annotation.

Recently, numerous web-based collaborative authoring tools have been developed [2]. XMetal Reviewer, a new product by Blast Radius Inc. [20], is a representative system. Designed for reviewing XML documents, it combines many of the advantages of Word and Acrobat. Basic annotations are integrated with the document, and global comments appear at the top of the document. Insertions and deletions can be incorporated into the document rather than kept as annotations, but this can always be reversed. This makes annotations persistent, unlike Word where accepted changes lose their identities as annotations.

XMetal Reviewer facilitates discussion by letting co-authors reply to each other’s annotations in real-time and in-context to reduce miscommunication. Annotations can be filtered by type, author, or status. XMetal is server-based to support collaboration among a large group of people. This could be a drawback for small groups that want a lightweight solution.

In all three systems, annotations can only be grouped using system-defined filters such as filter-by-author or filter-by-status. Comments about some specific aspect of a document may be scattered throughout the document, so it would be useful to gather them together. In a similar vein, there is only a partial record of the co-authors’ annotating processes. Some systems keep track of editing sessions, but do not otherwise capture ordering or relationships between individual annotations. This was identified by Weng [19], who noted that “[a]nnotations should be activity oriented.”

Annotation systems fit within the broader research area of collaborative writing. The classic collaborative writing systems such as PREP [13], Quilt [8], and SASSE [1] all support basic annotations, but do not support annotation grouping. In contrast, the recent Anchored Conversations system [6] allows text chats to be anchored into documents so that co-authors can have conversations within their work context. Although this is a real-time conversation tool rather than a shared annotation tool, it is an attempt to integrate meta-commentary with the document.

A SMALL FIELD INVESTIGATION

We investigated the email exchanges of three small groups of academics (3 to 5 people). Each group had co-authored a conference paper, approximately 8 pages in length. There were a total of 158 email exchanges analyzed, across the three groups. Many of the emails included document attachments (Microsoft Word or LaTeX files). While we did look at the documents to understand the relationship between the text content of the email and the annotations, our analysis focused on the email content. Below we categorize the most frequently-occurring content, and provide the percentage of the 158 emails to which each category and sub-category apply. Note that these are not exclusive categories. Most emails fell into more than one category.

To-do item(s) describe what remains to be done, or what should be done next (89%). The ordering of the items implicitly prioritizes the work, and sometimes co-authors give explicit direction on priorities. These often include collaborators' available times to work on the paper.

Summaries of edits that a co-author has just made to the document (92%) often appear together with to-do lists. Co-authors also summarize edits about issues that arise at multiple places in the document (78%), such as global word replacements or spelling changes throughout a document.

Discussions about the document often include parts of the text copied into an email to provide context (64%). Discussions include two subcategories: *questions* are sometimes directed at a particular co-author (53%); *general comments* (41%) pertain to the entire document (e.g., comments on the tone of the document or suggestions on document structure).

Comments-on-comments are comments about one or more previous comments. These most often concern comments that have not yet been addressed (31%) or advice to co-authors on how to process the referred to comments (34%).

The six task descriptions used in our evaluation (below) provide concrete examples of some of the above categories.

The information expressed as text embedded in email constitutes what we referred to at the outset as "meta-commentary." Co-authors devote a lot of effort to describing how annotations relate to each other because text is inefficient for expressing annotation location, type, or context, especially when an issue arises at multiple places in the document. Currently, co-authors must describe associated annotations by writing comments (either internal to the document or externally in email). There is no way to annotate multiple annotations directly. Recognizing this, we gathered requirements to build an annotation model that would unify all document-related communication by adding structure to annotations.

REQUIREMENTS FOR STRUCTURED ANNOTATIONS

Based on our literature review and our field investigation (FI), we have derived eleven design requirements for annotation systems that reflect co-authoring workflow. The last four requirements address communication that is currently happening outside the document.

- R1.** Support basic annotations such as edits and comments with specific anchors [1, 6, 19].
- R2.** Provide an easy way to incorporate changes specified in annotations into the document [1, 8, 19, 20].
- R3.** Preserve the original annotations in case co-authors want to later refer back to them [1, 20].
- R4.** Support both a separate annotation list view as well as viewing annotations embedded in a document [2, 15].
- R5.** Support annotation status to assist co-authors keeping track of the reviewing process [1, 19].

- R6.** Support document-related discussion with threaded annotations [2, 8, 15, 19].
- R7.** Support flexible filtering to allow co-authors to review more focused sets of annotations [15, FI].
- R8.** Allow annotations to be directed to specific co-authors [8, 20, FI].
- R9.** Support general comments that anchor to the entire document [15, 20, FI].
- R10.** Allow users to prioritize annotations [19, FI].
- R11.** Support the annotation of groups of annotations [FI].

We evaluated the three systems discussed in the Related Work section (Microsoft Word 2003, Adobe Acrobat Professional 7.0, and XMetal Reviewer) against these requirements. The results are summarized in Table 1, which suggests that current tools do not support some of the requirements.

	Word	Acrobat	XMetal
R1: basic anchors	Yes	Yes	Yes
R2: incorporated	Limited	Limited	Yes
R3: reversible edits	Limited	No	Yes
R4: dual views	Yes	Yes	Yes
R5: status	Limited	Yes	Yes
R6: discussions	Yes	Yes	Yes
R7: filtering	OR only	OR only	OR only
R8: specify receiver(s)	No	No	Yes
R9: general comments	No	No	Yes
R10: prioritization	No	Limited	No
R11: grouping	No	No	No

Table 1. Evaluating current co-authoring systems with respect to the eleven requirements.

STRUCTURED ANNOTATION MODEL

Using the requirements as a guide, we constructed a comprehensive model of annotations that encompasses the behaviors we observed in the field investigation. Every annotation has a set of attributes. Depending on the purpose of the annotation, some of the attributes can be empty.

Mandatory attributes are the *creator* of the annotation, a *timestamp*, reviewing *status* (unread/read and accepted/rejected), and an *anchor* (the annotation's location and range relative to the document content). Multiple non-contiguous ranges are permitted as the anchor for a single annotation. As a special case, the anchor can be the entire document.

Optional attributes are the *name* of the annotation (a short text string), a list of *recipients* (those co-authors who can view the annotation), a free-form text *comment*, *modification* (insertion, deletion, or a replacement of text), a *priority*, and *substructure* (a list of other annotations that this annotation refers to).

Each annotation must have at least one of the name, comment, modification, or substructure attributes in addition to the four mandatory attributes.

We classify annotations into two categories: *single* annotations that have no substructure, and *bundled* annotations that have substructure. The latter are called *bundles*.

Identifying types of annotations

There are some common annotation types that correspond to traditional annotations. An *edit* populates the modification attribute. A *comment* has only the comment attribute with an anchor into specific document content, and a *general comment* has an anchor that is the entire document. These are all single annotations (Figure 1).

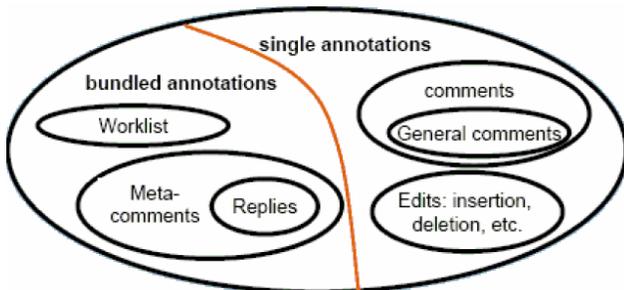


Figure 1. A Venn diagram illustrating how different types of annotations fit into the annotation model.

A number of interesting new types of annotation arise from our model. A *meta-comment* is a comment that has substructure, indicating the list of annotations to which it refers, and these in turn may have anchors into the document. Meta-comments can have their own document anchors (in which case they are not “pure” meta-comments). This nesting can be as many levels as desired, leading to the notion of inherited anchors by which a meta-comment is recursively associated with the anchors of its substructures. A *reply* is a special meta-comment that refers to just a single previous annotation.

Another special type of bundle is a *worklist*. An example would be a bundle having the name “Check spelling” and comment text that says “I am not sure how we spell some of the people’s names in our report —please make sure I have them right?” The recipient list would indicate who is to do the spell-check, and the anchor would indicate all of the places in the document where the names in question appear.

Linking bundle creation to the co-authoring process

The spell-checking bundle just described could be created manually, but we envision it being created automatically as a side effect of running a document processor’s spell-checking command. Realizing that the misspelled words are all names of people, a user could indicate that the selected words form the multi-location anchor for a new bundle by clicking on a button in the spell checker’s dialogue box, and then manually adding the name and comment attributes by

typing text into the appropriate fields. Recipients would be selected from a pull-down menu of co-authors.

Another application for automatic bundling as a side effect is the “track changes” feature in Word. A user should be able to turn on tracking that automatically bundles all new edits (and comments too, if desired) so that at the end of a session there is a ready-made bundle that can be turned into a worklist so the user can review just the changes from the current session, or highlight the changes for another co-author to review.

The full power of structured annotations lies in the interplay between normal workflow (editing, commenting, and reviewing) and the ability to capture that workflow and use it to manage future workflow. In the annotation model proposed by Weng and Gennari [19] users can assign only one pre-defined category, such as “question” or “reply,” to each annotation. Our model allows users to define their own categories and bundle relevant annotations into the substructure of a new annotation whose name attribute identifies the category. Moreover, any annotation can be assigned to multiple categories because the bundling substructure has no restrictions other than the requirement that it be acyclic.

The addition of optional user-defined annotation attributes may still be necessary. It would be an easy extension to our model.

THE BUNDLE EDITOR

Based on our model for annotations, we implemented a high fidelity prototype called the “Bundle Editor,” which has a number of functions designed to support structured annotations (Figure 2). The main features are a two-pane window with the *document pane* above and annotations below in the *reviewing pane* (much like Word, Acrobat, and XMetal), filtering operations to select annotations, a mechanism for grouping annotations into bundles, and the ability to annotate previous annotations. Every annotation has attributes (some empty) including an anchor into one or more locations in the document, and references to a set of previous annotations to which it applies. Grouping and references provide the structure that is absent in other co-authoring systems, as well as providing explicit representation of user-defined workflow.

The reviewing panel is a multi-tabbed pane. Each tab pane contains different information. Two tabs in the reviewing panel are permanent. The first tab is “All Annotations,” which contains all single annotations (inserts, deletes, replacements, and comments). General comments initially appear at the top of the list, the rest appear in the order in which their anchors occur in the document.

The second permanent tab is “All Bundles.” It lists all bundles, the annotations that have substructure. The last bundle listed in this tab is named “all other annotations.” It is maintained automatically by the system. It contains all the single annotations that do not belong to other bundles.

Functional description of the Bundle Editor

We describe some of the important functionality with links to the specific requirements in parentheses.

Basic Functionality

The Bundle Editor has all of the basic functionality that a typical document editor has, such as insert, delete, and comment (R1). It also has specific functions to create a bundle (R11). Bundles are stored with the document and linked to various places in the document or to other annotations. Co-authors can add/remove annotations from bundles by selecting the annotations of interest and then clicking on the add/remove button. An annotation can be in more than one bundle and bundles can be in other bundles (Figure 2). Co-authors annotate a group of annotations by including a comment in the appropriate bundle and directing the bundle to a particular set of co-authors (R8).

The filtering function in the Bundle Editor is more flexible than the filtering functions in existing tools (R7). It allows co-authors to select annotations based on multiple attributes such as “all of Jennifer’s and Brad’s comments”. The result of filtering is a new bundle that is a subset of annotations in the reviewing pane to which the filter was applied. The result can either replace the bundle in the reviewing pane or appear in a new reviewing pane.

Reviewing progress can be tracked by assigning a status to individual annotations (R5). Depending on co-authors’ reviewing activities, the system tracks some of the annotation status automatically so an “unread” annotation becomes “read” by a co-author when it has been selected. Co-authors can always over-ride a system-assigned status.

The Bundle Editor has functions for replying to annotations, which encourages discussion (R6), and allows co-authors to make general comments to each other without leaving the document (R9).

Working with Bundles

Various techniques help users maintain a mental model of the document and its annotations. In order to capture the structure of annotations, we employ a threaded list of annotations in the reviewing panel (R6). Users can expand or collapse any bundle to view or hide the annotations belonging to it. A right-click on any annotation within the document or the reviewing panel gives users the option to view the bundles to which it belongs (R11).

Users can select multiple bundles at a time and perform operations (such as setting the reviewing status) on all the selected annotations. If a bundle is selected, all its sub-annotations will be highlighted in the document (Figure 2). Users can have several bundles active at the same time, each in separate tabs of the reviewing pane, and switch between them. Each tab can be sorted according to author, date, order-within-document, or various other attributes. Co-authors can prioritize annotations in a bundle using drag-and-drop techniques (R10).

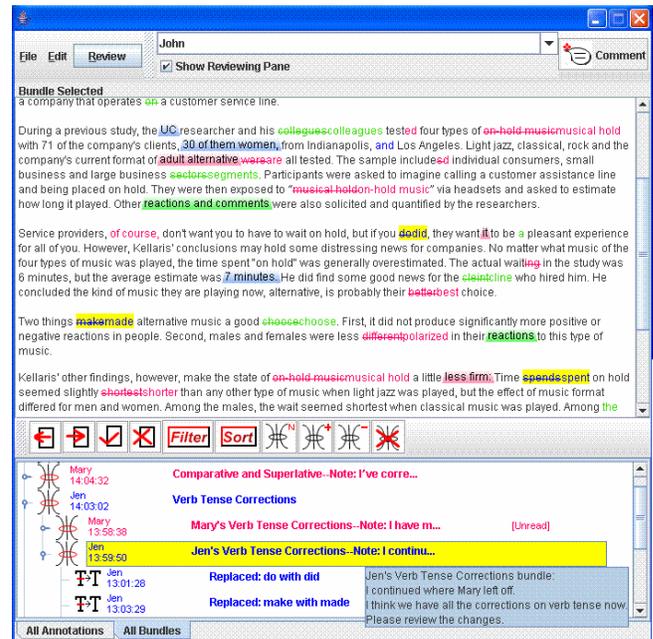


Figure 2. Bundle Editor with document & reviewing panes. The user highlights a bundle in the reviewing pane, which highlights all its sub annotations in the document pane. The bundle “Verb Tense Corrections” has two sub-bundles. One was created by Mary and one was created by Jen.

The Four Primary Ways of Creating Bundles

Bundles can be created manually while annotating the document. For example, if Jennifer finds recurring problems in a document, she can create a bundle by explicitly selecting all of the relevant annotations.

Temporary or working bundles are created by filtering and other operations. They can be saved as permanent bundles just by clicking. For example, Jennifer might want to look at the comments made by Brad. She can create a working bundle by filtering on “Brad” and “comment” and save the result as a bundle for later reviewing.

Working bundles can also be created by normal editing commands, such as “Find/Replace.” Brad may want to replace all the occurrences of “Jennifer” with “Angelina” and then save the results as a bundle so that other co-authors can manipulate all of the annotations in a single operation, such as setting the status to “reject”.

A bundle is created automatically at the end of every reviewing session. So after Jennifer finishes her session, all her new annotations from that session form a bundle that other co-authors can review unless she elects to not save it. This mechanism generalizes the “track changes” functionality in current editors and provides a uniform way to capture reviewing history.

Implementation

The Bundle Editor is implemented using Java Swing. The most important underlying component of the editor is the Annotation class. The Annotation class encapsulates all the annotation features, in particular the attributes and structure

of an annotation. It also encodes various operations that can be performed on annotations such as adding/removing annotations to/from sub-structures. The two main interface components are the document panel and the reviewing panel, which govern different annotation displays.

EVALUATION OF STRUCTURED ANNOTATIONS

We conducted an experiment to investigate the effect of structured annotations on reviewing workload and quality.

Participants

A total of 20 people (8 females) participated. They were undergraduate and graduate students recruited through online mailing lists and newsgroups. They were paid \$20 for their time. All spoke English as their native language. Seventeen used a word processor (mainly Microsoft Word) every 2-3 days, and 3 did so once a week. All felt very confident about using their word processor, although 5 had never used any annotation functions. They had all been involved in collaborative authoring, some fewer than 5 times and some more than 10 times.

Method

We compared two annotation systems: the bundle system, which supported structured annotations, and a simple system, which supported edits, and comments. The simple system was intended to be representative of current co-authoring systems. Both systems were created by modifying our Bundle Editor, so they differed only in their annotation functions. The content in bundled annotations that could not be included in the simple system's annotations was displayed in a separate simulated email window, beside the system interface. Table 2 summarizes the differences between the two systems.

	Bundle System	Simple System
Interface components	document panel, multi-tabbed reviewing panel	document panel, single pane reviewing panel
Single annotations (excluding general comments)	embedded in the document and listed or grouped in the reviewing panel	embedded in the document and listed in the reviewing panel
General comments	listed at the top of the "All Annotations" tab in the reviewing panel	shown in the simulated email window
Group of related Annotations	listed in the "All Bundles" tab in the reviewing panel	shown in the simulated email window
Filtering functions	AND, OR filtering on all or a subset of annotations	OR filtering on all annotations

Table 2. Bundle System vs. Simple System

Participants were asked to assume the role of co-authors of two documents and to review annotations related to the documents made previously by other co-authors.

Participants were instructed that reviewing annotations meant accepting the annotations they agreed with and rejecting the others, according to a prescribed task. Each participant saw both systems, with a different document for each system. The two documents used were chosen from ScienceDaily [17]: *docB* (528 words, 7 paragraphs) is about the growth of black holes, and *docM* (535 words, 7 paragraphs) is about customer reaction to "on-hold music" when calling a customer service phone line. The two documents have almost identical reading difficulty level, as determined by the Kincaid Readability Test and the Flesch Reading Ease Score. A third document was used during two practice sessions. Because this was common to all experimental configurations, we were not concerned with its similarity to the other two documents.

Tasks

There were six tasks to complete for each document. The annotations for all tasks were present from the outset. We controlled for the number, type, and authorship of annotations in the documents: 52 basic annotations (8 insertions, 5 deletions, 25 replacements, and 14 comments); Jennifer, John, and Mary made 15, 15, and 25 annotations respectively. In addition we controlled for reviewing difficulty with respect to the amount of context required to accept/reject an annotation; some could be processed by reading a single sentence, whereas others required two sentences or a full paragraph.

All tasks were representative of the types of tasks we saw in our field investigation, where authors connected meta-commentary in email with lower-level document-embedded annotations. However, the difficulty in our tasks was primarily for the user to find/navigate to the right set of annotations to review, which was our main focus. Because the subjects were not authors, our tasks minimized the difficulty of deciding whether to accept/reject annotations.

Task 1 – Location Pointers. (Summaries of edits)

Instruction. *docB*: review annotations on quantifying words (e.g., "at least," "at most"). *docM*: review annotations on comparative and superlative forms of adjectives. (5 task-relevant annotations from 1 co-author distributed in each document.)

Presentation. Bundle system: a bundle with a comment attached containing all relevant annotations. Simple system: an email message containing location pointers for relevant annotations.

Expectations. Better performance for both speed and accuracy in the bundle system.

Task 2 – Localized Annotations. (General comments)

Instruction. Review all annotations in a paragraph. (5 task-relevant localized annotations from multiple co-authors.)

Presentation. Bundle system: a general comment describes which paragraph to review. No relevant bundle created. Simple System: an email message describes which paragraph to review.

Expectations. Similar performance in both systems.

Task 3 - Spelling Edits. (Summaries of edits)

Instruction. Review spelling edits in the document. (6 task-relevant annotations from 1 co-author distributed in the document.)

Presentation. Bundle system: a bundle with a comment attached containing all relevant annotations. Simple system: an email message describing relevant annotations.

Expectations. Better performance for speed in the bundle system. Similar performance for accuracy in both systems because spelling edits are easy to review.

Task 4 – Multiple Co-authors Annotations. (To-do items)

Instruction. Review all verb tense edits in the document. (8 task-relevant annotations from 2 co-authors distributed in the document.)

Presentation. Bundle system: a bundle with two bundles (created by 2 co-authors) in its substructure, each containing task-relevant annotations and comments attached to each bundle. Simple system, two email messages (from 2 co-authors) are shown (one is a reply to the other) describing the relevant annotations.

Expectations. Better performance for both speed and accuracy in the bundle system.

Task 5 – Global Replacements. (Summaries of edits)

Instruction. docB: review all the replacements between “grow” and “growth.” docM: review all the replacements between “on-hold music” and “musical hold.”

Presentation. Bundle system: a bundle with comment attached containing relevant annotations. Simple system: an email message describing relevant annotations.

Expectations. Better performance for speed in the bundle system. Similar performance for accuracy in both systems because these replacements are easy to identify.

Task 6 – Unaddressed Comments. (Comments-on-comments)

Instruction. Review a co-author’s comments that have not been accepted or rejected.

Presentation. Bundle system: a general comment describes which co-author’s comments to review. No relevant bundle created. Simple System: an email message describes which co-author’s comments to review.

Expectations. Filtering functions are likely to be used in both systems. Better performance for both speed and accuracy in the bundle system because of multi-attribute filtering.

Measures

Our main dependent variables were speed and accuracy. Speed consisted of total completion time per task, which was the aggregate of navigation time and decision time. Accuracy was assessed with three measures: the number of task-relevant annotations reviewed (accepted/rejected), the number of task-relevant annotations reviewed *correctly*, and the number of non-task related annotations reviewed. We also recorded the number of times the filtering function was used. Self-reported measures captured through questionnaires included ease of finding annotations, ease of

completing tasks, confidence in performing task, ease of use, ease of learning, and overall system preference.

Design

The experiment was a within-subjects 2x6 (system type x task) factorial design. Document type was a within-subjects control variable, and both system and document presentation orders were between-subject controls. A within-subjects design was chosen for its increased power and because it allowed us to collect comparative comments on the two systems. To minimize learning effects we counterbalanced the order of presentation for both system type and document, resulting in four configurations.

Apparatus

The experiment was conducted on a single Linux machine running SUSE 9.0 with an Intel® Pentium® 4 CPU and 512 MB of RAM. The software for both of the experimental systems was written in Java 1.5.1.

For each task, a task instruction screen was shown first. Some tasks also had task background information to refresh participants on basic English grammar or words used in the document. For example, task 1 in docM explains what the comparative and superlative forms are for adjectives. For each document, the same task instructions were given for both the simple and bundle systems, but the two documents had different (but similar) task instructions.

Once a participant read the task instruction screen, s/he clicked on the “Start Task” button. The system loaded and the data logging and timing function started. After the participant finished a task, s/he clicked “End Task” and the next task instruction appeared.

Procedure

The experiment was designed for a single two-hour session. A questionnaire was administered to obtain information on past computer and writing experience. Participants were then shown a training video on general concepts such as collaborative authoring and how to use the first system, followed by a practice session of six reviewing tasks using the first system. The practice tasks were similar to the experimental tasks described previously, but in a different order and on a practice document different than either of the test documents. Participants were next asked to read the original version of the task document (i.e., with no annotations), after which they had to perform a list of six tasks in the order they were given. A second questionnaire was administered to collect feedback on the first system. Participants were given a 5-minute break and were then shown a video on how to use the second system followed by six practice tasks using the same practice document, then six experiment tasks for the second document. A final questionnaire solicited feedback on the second system and asked the participants to directly compare the two systems. A short de-briefing was conducted with some of the participants based on their questionnaire data.

Hypotheses

Our main hypotheses were as follows:

H1. The bundle system will reduce the time participants spend navigating to relevant annotations. Some tasks (as identified above) will be more affected than others.

H2. Participants will perform more accurately in the bundle system than the simple system. Some tasks (as identified above) will be more affected than others.

Results

Here we report on both the quantitative data captured through software logging as well as the self-reported data from our questionnaires.

Before testing our hypotheses, we checked to make sure that there was no effect of document. Investigation of an interaction effect between document and task on total time ($F(4,64) = 4.706, p = .002, \eta^2 = .227$) revealed that task 1 was more difficult in docB than in docM. Our goal had been to create two documents that were as equal in difficulty as possible, and so we removed task 1 from our remaining analyses, and focus exclusively on tasks 2 through 6.

To test our hypotheses we ran a 2 systems x 2 order of documents x 5 tasks ANOVA for our speed and accuracy measures. System and tasks were within-subjects factors, and orders of system and document presentation were both between-subjects factors. For our secondary analyses, a series of two-tailed t-tests were used to investigate performance differences between the two systems for each of the tasks. Along with statistical significance, we report partial eta-squared (η^2), a measure of effect size, which is often more informative than statistical significance in applied human-computer interaction research [10]. To interpret this value, .01 is a small effect size, .06 is medium, and .14 is large [7].

Testing Hypotheses

Total navigation time (across all 5 tasks) was significantly less in the bundle system ($p < .001$). Participant's decision time was not impacted by the two systems ($p = .336$). The large navigation time effect was sufficient to influence the total completion time, which was also significantly lower in the bundle system ($p < .001$). The means are given in the top half of Table 3.

Speed (seconds)	Mean		F	Sig.	η^2
	Bundle	Simple			
Navigation	39.3	58.3	40.1	<0.001	0.715
Decision	60.8	64.5	0.98	0.336	0.058
Completion	100.2	122.8	22.9	<0.001	0.589
Accuracy (number of annotations)					
Reviewed	5.25	5.01	19.53	<0.001	0.550
Reviewed correctly	4.84	4.61	59.02	0.018	0.306
Errors	0.05	0.65	7.05	<0.001	0.787

Table 3. Speed and accuracy measures across five tasks. Df = (1,16). N=20.

As hypothesized in H1, and as Figure 3 shows, some tasks required less navigation time than others, and this differed by system (an interaction between task and system, ($F(4,64) = 16.09, p < .001, \eta^2 = .354$)). T-tests revealed that tasks 3, 4, and 5 were all significantly faster in the bundle system (all $df=19, p < .001$). There were no differences detected for tasks 2 and 6.

Consistent with hypothesis H2, accuracy was also significantly better with the bundle system. Across all 5 tasks, participants reviewed more task-relevant annotations ($p < .001$), they correctly processed more task-relevant annotations ($p = .018$), and they made fewer identification errors, meaning they reviewed fewer non-task related annotations ($p < .001$) in the bundle condition. Means for these errors are shown in the bottom half of Table 3.

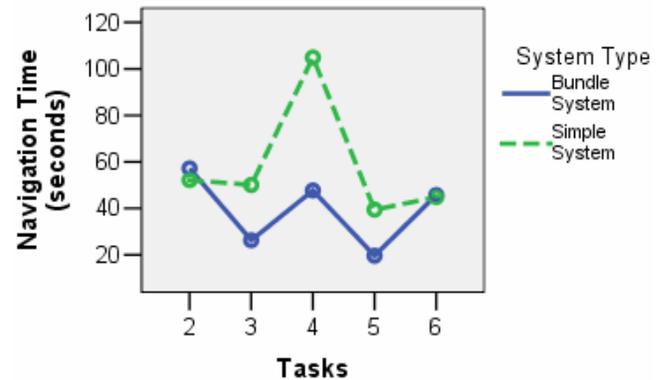


Figure 3. Mean navigation times per task in the two systems. N=20.

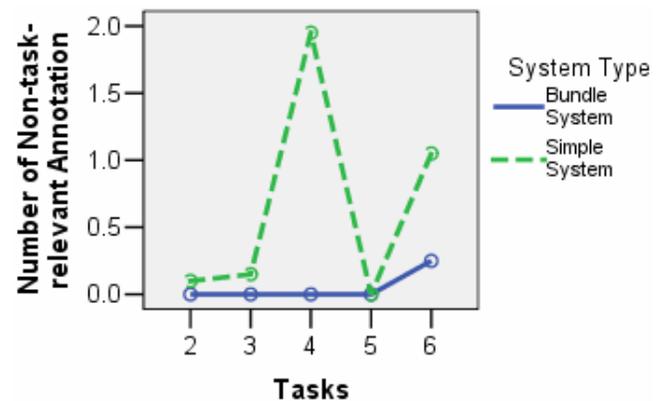


Figure 4. Mean number of non-task related annotations reviewed. N=20.

There was an interaction between task and system on the number of non-task related annotations reviewed ($F(1, 16) = 21.93, p < .001, \eta^2 = .578$). T-tests showed that there were significantly more non-task related annotations reviewed in the simple system for task 4 and task 6 (both $df=19, p < 0.001$). These differences are apparent in Figure 4.

Other Effects

In addition to the main effect of system type, we also found a main effect of task across all measures. This was expected

because we designed each task to match a particular type of annotation activity; some activities are inherently more difficult and time consuming than others.

We also found a number of multi-way interactions involving task, and system and document presentation orders. Systematic investigation of each of the interactions revealed no clear interpretation of the interactions.

Not surprisingly, participants used the filtering functions more in the simple system than in the bundle systems ($F(1, 16) = 39.42, p < 0.001, \eta^2 = 0.711$).

Self-reported Measures and Other Feedback

We ran the Wilcoxon Signed-Rank Test on the questionnaire data. Consistent with our navigation and accuracy findings, analysis of the self-reported measures showed that with the bundle system participants found it easier to find annotations ($p = 0.002$), easier to complete tasks ($p = 0.012$), and were more confident in their answers ($p = 0.014$). They also had an overall preference for the bundle system ($p < 0.003$). But there was no significant difference in the ease of learning ($p = 0.667$) or ease of use ($p = 0.26$) between the two systems. When asked which of the two systems they would prefer to continue using, 18 out of the 20 participants (90%) chose the bundle system.

Participants provided free-form comments at the end of the questionnaire about what they liked and disliked about each system. For the simple system, although not actually integrated with the system, most participants indicated that they liked the email window, which provided them more information to complete tasks. Interestingly, many participants who used the simple system first indicated they liked the filtering function; however, of those participants who had first been exposed to the bundle system, almost all disliked the comparatively limited filtering functions in the simple system. For the bundle system, participants noted the time saved using bundles and were surprised how easy it was to learn to use bundles. They also liked the flexible filtering provided in the bundle system. One suggestion for improvement in the bundle system was the small size of the reviewing panel; participants felt it required too much scrolling.

Summary of Results

To summarize, the bundle system allowed participants to navigate among annotations significantly faster for tasks 3, 4 and 5. Participants were also significantly more accurate with the bundle system; for example, they reviewed significantly fewer non-task related annotations for tasks 4 and 6. Overall, 90% of participants preferred the bundle system.

Discussion

Bundle Concept is Intuitive: All participants developed the strategy of using a bundle list as their guide for completing tasks. They searched first for an existing bundle related to the current task description before directly searching for

annotations in the document. Based on their interaction sequences with the prototype and their feedback, it was clear that the bundle concept, and its fit within the task workflow, was intuitive.

Bundles Reduce Navigation Time: Once participants found a relevant bundle, locating each annotation in the document was a short single click away. By contrast, in the simple system, most of the navigation time was spent searching through the document for the next relevant annotation, which was time consuming. Bundling reduced the navigation time for tasks 3, 4, and 5, none of which were amenable to basic filtering. For task 6, filtering was a good strategy in both systems. Even though the bundle system had the advantage of filtering on both the comment and author attribute, it was easy in the simple system to filter on author and then identify the comments. So it was not surprising that task 6 did not show a difference. As one would hope, there was no difference in navigation time for tasks that were localized within the document (task 2).

Bundles Improve Accuracy: Once the correct bundle was found, users were guaranteed to find the task-relevant set of annotations. This minimized the number of extra annotations reviewed, and allowed users to concentrate on reviewing the actual annotations. The biggest difference was found in task 4 where 39 extra annotations were reviewed across all participants in the simple system, but no extras were reviewed in the bundle system. The cause of this was users mistakenly identifying annotations to be verb tense changes; for example, in docB replacements between “grow” and “growth” were treated as verb tense changes. This was quite surprising, given that all our participants were native English speakers. But it shows that bundling can overcome even basic misunderstandings of the English language.

Users Group Annotations: Participants filtered significantly more often in the simple system than in the bundle system. They did so to reduce the number of annotations under consideration for a task. Participants were effectively creating their own *temporary* task-based annotation groups. Not only might there be cost overhead to having the reviewer do the grouping (see discussion on cost below), but current systems do not allow users to store filter results for subsequent usage. Bundling supports the easy creation and reuse of annotation groups formed through filtering.

Scalability of Bundles: Our target context for bundles is heavily annotated documents. We chose simpler documents for our experiment in order to keep the tasks manageable. We speculate, however, that a comparison between the bundle and simple systems for sophisticated documents would be even more dramatic. As a document increases in length, causing relevant annotations to be spread further apart, navigation time will increase without bundles.

Cost/Benefit Tradeoff: Our experiment only evaluated the annotation reviewing stage of authoring. Bundles shift some of the effort that is traditionally spent on annotation

reviewing to annotation creation. At first glance this might appear to be a zero sum game, because effort is only being shifted within the authoring workflow. We argue that authors are currently communicating a large amount of information through email, and that *manually* creating bundles should be more efficient than the overhead incurred in the inefficiencies of email. *Automatically* generated bundles should clearly be faster than email communication. A tradeoff to explore, however, is between the value of bundles and the increased overall complexity that they bring to the annotation system. Evaluating bundle creation, and the impact of bundles on the complete co-authoring workflow, is an obvious next step in our work.

Bundles Provide a More Pleasant User Experience: When participants were asked which system they preferred, 90% stated that it was the bundle system. The elements of the simple system they liked the most were the email message and filtering function. We note that the experiment design provided a single email message per task, with clear instructions, which underestimates the workload in real situations when users need to locate the relevant email, and possibly an entire email thread describing the task. The two participants who favored the simple system were both experienced Microsoft Word users, but neither had used the annotation functions. They were excited by the functionality in the simple system, and they found the bundle system to be complex and confusing. However, they both recognized the advantages of bundles and thought that after becoming accustomed to basic annotation functions, they might desire more complex ones.

CONCLUSION AND FUTURE WORK

We have presented a structured annotation model, which includes annotation groups called bundles. Bundles are designed to improve co-authoring workflow. We have implemented a preliminary prototype called the Bundle Editor and compared it to a system that only offers basic annotation functions. Our study focused on annotation reviewing and showed that structured annotations can reduce the time it takes to navigate between task-relevant annotations and can improve reviewing accuracy. Now that there are confirmed benefits on the reviewing side, our next step will be to investigate the usability of bundle creation and, more generally, how bundles support the full co-authoring workflow. Broader issues still remain. These include investigating how bundles might be extended to support version control and synchronous co-authoring, which are both classic problems in the collaborative writing literature.

REFERENCES

1. Baecker, R. M., Nastos, D., Posner, I. R., and Mawby, K. L. The user-centered iterative design of collaborative writing software. In *Proc. CHI 1993*, ACM Press, (1993), 399-405.
2. Brush, A. J. B. Annotating Digital Documents for Asynchronous Collaboration, *Ph.D. dissertation*, (2002). University of Washington.
3. Brush, A. J. B., David, B., Anoop, G., & Cadiz, J. J. Robust annotation positioning in digital documents. In *Proc. CHI 2001*, ACM Press, (2001), 285-292.
4. Cadiz, J. J., Anop, G., & Jonathan, G. Using Web annotations for asynchronous collaboration around documents. In *Proc. CSCW 2000*, ACM Press, (2000), 309-318.
5. Catlin, T., Bush, P., & Yankelovich, N. InterNote: extending a hypermedia framework to support annotative collaboration. *Proc. Hypertext 1989*, ACM Press, (1989), 365-378.
6. Churchill, E. F., Trevor, J., Bly, S., Nelson, L., & Cubranic, D. Anchored conversations: chatting in the context of a document. In *Proc. CHI 2000*, ACM Press (2000), 454-461.
7. Cohen, J. Eta-squared and partial eta-squared in communication science. *Human Communication Research*, 28, Oxford Journals, (1973), 473-490.
8. Fish, R. S., Kraut, R. E., & Leland, M. D. P. Quilt: a collaborative tool for cooperative writing, *Conference Sponsored by ACM SIGOIS and IEEECS TC-OA on Office information systems*. ACM Press, (1988), 30-37.
9. Kahan, J., & Koivunen, M.-R. Annotea: an open RDF infrastructure for shared Web annotations. *Proc. 10th International Conf. on World Wide Web*. (2001), 623-632.
10. Landauer, T. Chapter 9: Behavioral research methods in human-computer interaction. In M. G. Helander, T. K. Landauer and P.V. Pranh (Eds), *Handbook of Human-Computer Interaction* (2nd ed., pp. 203-227). Amsterdam: Elsevier Science B.V (1997).
11. Marshall, C. C. Annotation: from paper books to the digital library, *Proc. of the 2nd ACM International Conference on Digital libraries*. ACM Press (1997), 131-140.
12. Marshall, C. C., & Brush, A. J. B. From personal to shared annotations. *Ext. Abstracts CHI 2002*, (2002), 812-813.
13. Neuwirth, C. M., Kaufer, D. S., Chandhok, R., & Morris, J. H. Issues in the design of computer support for co-authoring and commenting, *Proc. CSCW 1990*, (1990), 183-195.
14. Noël, S., & Robert, J.-M. Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, (2003), 13(1), 63-89.
15. Ovsiannikov, I. A., Arbib, M.A., & McNeill, T. H. Annotation technology, *International Journal. of Human-Computer Study*, (1999), 50(4), 329-262.
16. Resnick, M. M. a. D. Third Voice: Vox Populi Vox Dei? http://firstmonday.org/issues/issue4_10/margolis/index.html.
17. ScienceDaily. <http://www.sciencedaily.com/>
18. Sharples, M., Goodlet, J.S., Beck, E.E., Wood, C.C., Easterbrook, S.M. and Plowman, L. Research issues in the study of computer supported collaborative writing. In M. Sharples (Ed.), *Computer Supported Collaborative Writing* (pp. 9-28): Springer-Verlag, (1993).
19. Weng, C., & Gennari, J. Asynchronous Collaborative Writing through Annotations, Notes. *Proc. CSCW 2004*, ACM Press, (2004), 564-573.
20. XMetal Reviewer. http://www.xmetal.com/en_us/products/xmetal_reviewer/index.x