Annotating Digital Documents for Asynchronous Collaboration[*]

Alice Jane Bernheim Brush
Department of Computer Science and Engineering
University of Washington
ajb@cs.washington.edu

Technical Report 02-09-02
September 2002

[*] Note, this is a technical report version of my Ph.D. dissertation.

Abstract

Annotations are a natural way to record comments and ideas in specific contexts within a document. When people read, they often underline important parts of a document or write notes in the margin. While we typically think of annotating paper documents, systems that support annotating digital documents are becoming increasingly common. Annotations on digital documents are easily shared among groups of people, making them valuable for a wide variety of tasks, including online discussion and providing feedback.

This research explores three issues that arise when using annotations for asynchronous collaboration. First, I present the results of using a prototype annotation system, WebAnn, to support online discussions in an educational setting. In a field study in a graduate class, students contributed twice as much content to the discussion using annotations compared to a traditional bulletin board. Annotations also encouraged a different discussion style that focused on specific points in the paper being discussed. The study results suggest valuable improvements to the annotation system and factors to consider when incorporating online discussion into a class.

Second, I examine providing appropriate notification mechanisms to support online discussion using annotations. After studying notifications in a large-scale commercial system and finding them lacking, I designed and deployed enhancements to the system. A field study of the new notifications found that overall awareness of annotation activity on software specifications increased with my enhancements. The study also found that providing more information in notification messages, supporting multiple communication channels through which notifications can be received, and allowing customization of notification messages were particularly important.

Third, I explore how to anchor annotations robustly to documents to meet user expectations on documents that evolve over time. I describe two studies designed to explore what users expect to happen to their annotations. The studies suggest that users focused on how well unique words in the text that they annotated were tracked among successive versions of the document. Based on this observation, I designed the Keyword Anchoring algorithm, which locates an appropriate new position for an annotation using unique words in the text annotated by the user.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I have had the privilege of collaborating with David Bargeron, Alan Borning, JJ Cadiz, Jonathan Grudin and Anoop Gupta. Their collaboration and mentorship has been invaluable. With these collaborators I have previously published some portions of this research at CSCL 2002 [BBG+02a] (Chapter 3), CHI 2002 [BBG+02b] (Chapter 4), and CHI 2001 [BBG+01] (Chapter 5). The material included in this dissertation has been revised.

This research would not have been possible without all the participants that graciously agreed to take part in my studies. I am particularly grateful to the graduate students at the University of Washington and groups at Microsoft for participating in the field studies over several months. At the University of Washington, I also received generous support from the Catalyst group, in particular Mark Alway who provided access to the data from the EPost system. During the studies and evaluations, Molly Brown, Mike Brush, Duncan Davenport, Scott LeeTiernan, Cathy Marshall and Gina Venolia also provided valuable assistance.

This research was primarily done while I was an intern at Microsoft Research in the Collaboration and Multimedia Systems group. During my graduate career I experienced the best of both Microsoft Research and the Department of Computer Science and Engineering at the University of Washington with terrific support and interactions at both places.

At Microsoft Research, David Bargeron has been a terrific mentor and generously allowed me to build the WebAnn prototype on top of his Common Annotation Framework. David's friendship, assistance, feedback, guidance, and support have all been instrumental in helping me to complete this research. Other members of the Collaboration and Multimedia Systems group, particularly JJ Cadiz, Jonathan Grudin, Anoop Gupta, Gavin Jancke, and Gina Venolia, provided additional guidance and mentorship. Also at Microsoft I was fortunate to work with and learn from Cathy Marshall and Scott LeeTiernan.

In the Department of Computer Science and Engineering at the University of Washington, I am privileged to have Alan Borning as my advisor. He has served as a valuable sounding board on my path through graduate school and is always available with helpful advice and feedback.

My Ph.D. committee members: Alan Borning, Gaetano Borriello, Richard Christie, Jonathan Grudin, and David Notkin, have all been generous with their time and I appreciate their support. The members of my reading committee: Alan Borning, Jonathan Grudin, and David Notkin, have provided additional guidance and I am very grateful for their feedback both on research and writing.

Many colleagues have assisted me during this research, providing feedback on talks, answering questions, helping with studies, and providing general support and encouragement. I am very grateful to Corey Anderson, David Bargeron, Molly Brown, Mike Brush, JJ Cadiz, Edward Cutrell, Mary Czerwinski, Duncan Davenport, Michael Ernst, Andrew Fiore, Jonathan Grudin, Anoop Gupta, Jeff Hightower, Zack Ives, Gavin Jancke, Cathy Marshall, Scott LeeTiernan, Craig Kaplan, Kurt Partridge, Rachel Pottinger, Sean Sandys, Vibha Sazawal, Sarah Schwarm, Neil Spring, Tammy VanDeGrift, Gina Venolia, Steve Wolfman, Ken Yasuhara, and Doug Zongker.

I am particularly grateful to Rachel Pottinger for reading and critiquing a draft of this thesis. I also have had many helpful discussions with Sean Sandys.

I am grateful to a number of people who have advised me and served as mentors during my educational career. Duane Bailey at Williams College was an integral part of my undergraduate career and encouraged my interest in graduate school. Janice Cuny from the University of Oregon provided mentorship and inspiration during my summer participation in the CRA Distributed Mentor program. At the University of Washington, Larry Snyder and David Salesin advised me on earlier research and provided guidance.

My entire family has been an incredible source of support and encouragement throughout my education. Recently my mother, Linda Bernheim, and mother-in-law, Karen Brush, have both provided invaluable assistance with childcare. I also appreciate the support of my husband, Mike.

For Mike and Colin

Chapter 1

Introduction

As we read, particularly when we want to remember or call attention to something, making annotations, perhaps underlining, highlighting, or making a small note directly on an interesting object, is a natural way to record thoughts and ideas. Every day people annotate all sorts of media, from documents that are mostly text such as newspapers and books to media composed of images such as architectural plans and musical scores. While almost any type of media can be annotated (consider graffiti), in this work I focus on annotations made on text documents.

When reading, authoring, or providing feedback on a text document, annotations allow you to easily record thoughts and ideas in the context of the document. If you pick up a document you recently read carefully, chances are good that you underlined an important part, made a star in the margin, or even jotted down a comment or two. People make annotations on documents for a wide variety of personal and collaborative tasks.

For personal use, annotations can be valuable for summarizing a document or recalling which parts of it were important or interesting. Annotations are also a very natural way to collaborate on a document, as when providing feedback to co-authors. Whether making annotations for personal use or to share with others, annotating directly on a document allows the annotator to easily call attention to particular sections and give context to notes by their location. Having the context in the document for an annotation allows shorter comments and can increase the clarity of a comment for other readers, or even the original annotator after some time has passed.

While we typically think of annotating paper documents, as more and more documents exist primarily in digital form, support for annotating digital documents is becoming more common. Annotations on digital documents have a unique advantage over annotations made on paper. In particular, digital documents are more easily shared among people, facilitating asynchronous collaboration using annotations in a variety of scenarios. For example, when authoring a document with multiple co-authors, digital annotations provide a natural way to communicate ideas and feedback, even if the co-authors are geographically far apart. Students in a class can use digital annotations as a way to discuss a paper, so that all student comments are visible and linked to the points they wish to discuss in the paper. Annotations on digital documents can also be searched and indexed automatically.

Most digital document authoring systems (e.g., [Ado, MOWD, LFK88, NKC+94]) provide an annotation feature, sometimes called “comments.” These systems store the annotations inside the annotated document and use their knowledge of the document’s internal format to position the annotations. Annotating the document with comments or feedback requires permission to edit the document, and if the document is distributed for feedback (perhaps via email) each person giving feedback annotates their own personal copy. The document author seeking feedback then receives multiple copies of the document, each with annotations from one person.

One major advantage of storing the annotations inside the document is that the system can insert explicit positioning anchors for the annotations so they will remain positioned appropriately even if sections of the document are edited or changed in some way (as long as the positioning anchors remain intact). This positioning issue is unique to digital document annotations since a paper document will never change once it has been annotated.

A number of systems for annotating HTML web documents have been developed. Unlike document authoring systems, these systems typically store the annotations separately from the annotated HTML document. This complicates displaying an annotated document to the user, since the system must identify which annotations belong where in the document and then add them to the display. In particular, if an annotated document changes, the system must decide where to place the annotations in the modified version of the document.

However, storing the annotations apart from the document makes annotation more practical for tasks involving asynchronous collaboration such as group discussion and document review. Groups can easily discuss online documents they do not have permission to modify, such as published papers or working drafts of proposed standards, since the annotations are only added to the local version of the document displayed to the user (rather than the original document). Also, when asking other people to review documents, the author does not need to grant permission to modify the document and everyone reviewing the document will be able to see the comments.

Keeping the annotations separate from the documents also allows them to be easily tracked for notification purposes. When new annotations are made on a document, the system can alert users who have specified an interest in comments on that document. In addition, the annotations can be easily searched if users want to find comments they made previously, but can not remember which document they annotated or if they are interested in finding comments made by a particular person.

While potentially very advantageous for collaboration, there are still a number of open questions involving asynchronous annotation of documents. By building software prototypes and deploying them in laboratory and field studies, this work explores three issues surrounding the use of annotations for asynchronous collaboration:

- **Value for Discussion in an Educational Setting**: Is the ability to discuss a document using annotations, where the context of a comment will be clear to other readers, valuable in an educational setting?
- **Awareness of Annotations**: What is the best way to notify users that new annotations have been made on a document? What information should be included in the notifications to increase a user's awareness of annotations?
- **Anchoring Annotations**: What do users expect to happen to their annotations when a document changes? How should new positions for annotations on modified documents be found to meet user expectations?

1.1 Annotations for Asynchronous Discussion

I believe that annotating digital documents is a powerful tool for asynchronous discussion because the context of the annotation is clear to other readers. To test this hypothesis I conducted a field study in an educational setting comparing online group discussion of technical papers using WebAnn, a prototype system I wrote for digital annotation, and EPost [Epos], a high quality web-based threaded discussion board. The students in a graduate-level human computer interaction class alternated using WebAnn and EPost for the online discussion. I surveyed students on their experience and analyzed their contributions to the online discussions.

During the field study, online discussions easily exceeded the required participation level set by the instructor. For a variety of reasons, including access, the students slightly preferred EPost. However, students contributed almost twice as much to the online discussion using WebAnn, and it encouraged a different discussion style, focused on specific points in the paper, which some students preferred. The online discussion, particularly from WebAnn, was expected to serve as a starting point for in-depth discussions in the classroom. Instead it unfortunately often competed with the classroom discussion. From the study, I identified enhancements that will improve

WebAnn, including notifications, and also important process considerations for incorporating online discussions into a class.

1.2 Notification for Annotations

The WebAnn field study of online group discussions along with other research [CGG00] highlights the potential for notification to improve the value of document annotation for asynchronous collaboration. Using shared annotations allows readers to benefit from seeing comments in the context of the original document. However, each person annotates the document when they have time, making notifications critical to alert other interested parties to new annotations without forcing them to constantly revisit the document.

To understand user needs for notification, I designed and experimented with several different improvements to the notification system in Microsoft Office Web Discussions [MOWD], a commercial online discussion system. The enhanced notifications included more detailed email notifications and notifications using peripheral awareness. I then studied a group using document annotations to review software specifications.

The new notifications increased awareness of comments on documents and were generally well received by participants. In particular, they were considered an improvement over the existing notifications in Office Web Discussions. Field study participants used notifications for a variety of purposes ranging from very active monitoring of annotations on the document to more casual tracking. This variety of use highlights the importance of allowing users a choice of how notifications are delivered and supporting easy configuration of notification content.

1.3 Robustly Anchoring Annotations to Documents

While storing annotations outside the document facilitates discussion and document review, it complicates displaying the annotations in a document. Enough information must be saved with each annotation to link it to the position the user selected within the document in a way that is robust to changes in the document. Otherwise, when the online document changes, the system could lose track of the annotation’s proper position within the document, and it could become orphaned. Orphaned annotations are typically displayed to the user at the bottom of the annotated document or in some other special fashion. The user then must manually find a new location for the annotation or determine that it is no longer relevant due to the document changes.

Orphaning of annotations represents a serious problem. A study observing Microsoft Office Web Discussions use over a 10-month period found the key complaint was the orphaning of annotations when documents changed [CGG00]. I also heard many complaints about orphaning during my field study of notifications in Office Web Discussions.

Although several existing systems for digital annotation have algorithms to cope with document modifications and avoid orphans, none of these algorithms are based on user expectations. In a lab study I explored what users expected to happen to their annotations when the document changed. Then, based on these expectations I designed the Keyword Anchoring algorithm for robustly locating an appropriate new position for an annotation.

1.4 Contributions

This dissertation makes a number of contributions to understanding the use of annotations for asynchronous collaboration, and to the design of annotations systems to better support collaborative tasks.

The first contribution is the set of findings from the field study in an educational setting comparing discussions using asynchronous annotations with a threaded discussion board. The study found a number of issues that educators should consider when determining what type of discussion system would be appropriate for their class, including instructor and student workload, instructor role, the desired relationship between online and in-class discussion, avoidance of discussion overload, and the importance of universal access.

Experience during the study with the WebAnn prototype annotation system suggests many design implications for online annotation systems, such as (1) allowing users to annotate at every level of granularity (from individual words to general comments on the entire document), (2) the importance of filtering and notification, and (3) flexibly allocating screen space between annotations and the underlying document.

The second contribution is the set of findings from the investigation of notifications for annotations during document review. The study found that notifications need appropriate content to be valuable, that multiple mechanisms for notifications are important so that users can select their preferred method, and that appropriate notifications can improve a user's awareness of annotations on a document.

The third contribution is the set of user expectations I gathered regarding how annotations should be positioned when the document they annotate is modified. The study findings indicate that participants considered some parts of the text that they had annotated particularly important and focused on how well these “keywords” and phrases were found in the modified version of the document. I also found participants paid little attention to the text surrounding their annotations. Finally, even when some of the original text associated with an annotation was found, in certain cases it seemed participants would have preferred the positioning algorithm orphan their annotation.

Based on the user expectations for robust anchoring, I developed the Keyword Anchoring algorithm, my fourth contribution. The algorithm uses unique words in the vicinity of an annotation as the distinguishing characteristics for positioning the annotation. These unique words are tracked among successive versions of a document. By focusing on document content rather than on the underlying document structure, Keyword Anchoring requires no cooperation from the document, and can be used with a variety of digital document formats.

1.5 Outline

In the next chapter I discuss related work, outlining the features of several relevant commercial and academic annotation systems. In particular, I discuss previous research on the use of annotations for asynchronous discussion, and notification and anchoring mechanisms in existing annotation systems.

Chapter 3 describes the WebAnn prototype system for online annotation and its use for asynchronous discussion of technical papers in a graduate-level class.

Chapter 4 examines the value of notifications for annotations, describing the notifications prototype software from a field study, its use and evaluation by study participants, and design implications for notification systems.

Chapter 5 discusses two lab studies designed to elicit user expectations for robustly anchoring annotations to documents when the documents are edited.

Chapter 6 describes the Keyword Anchoring algorithm designed based on the user expectations found in the studies described in Chapter 5.

I conclude in Chapter 7 with remaining issues for using annotations for asynchronous collaboration around documents and how my contributions may generalize to using annotations on other media, including video and audio, for collaborative tasks.

Chapter 2

Related Work

Annotation is an important part of the reading process, particularly when the reader wants to engage with the document. Marshall's field studies [Mar97, Mar98] of annotations in college textbooks describe a number of ways students used annotation including: marking locations, interpretation, and tracing progress through challenging sections. While the students in Marshall's study annotated paper textbooks, support for annotating digital documents is becoming common. Many familiar document-authoring systems, such as Word [MWo], include annotation features, and the increasing popularity of the web has led to many systems for annotating web documents. Annotations on digital documents are often easily shared among several people, facilitating asynchronous collaboration using annotations in a variety of scenarios.

In this chapter, I first discuss examples of the wide variety of systems for annotating digital documents. I then present in more detail the systems and studies relevant to the annotation issues I focus on: online discussions in an educational setting, notification strategies, and techniques for positioning annotations on documents that change over time.

2.1 Systems for Annotating Digital Documents

A wide variety of annotation systems exist. Section 2.1.1 discusses examples of document authoring and document viewing software with annotation features. In Section 2.1.2, I focus on systems for annotating HTML documents, describing examples of the most common types. This section does not cover linking applications such as DLS [CDH+95] and Fluid Documents [ZCM98] developed in the hypertext community, since these applications focus on augmenting linking capabilities rather than mechanisms for creating text annotations or having discussions in the context of a document.

2.1.1 Software with Annotation Features

Many document viewing systems and document authoring systems include features to annotate a document. For example, Microsoft Word [MWo] supports highlighting and commenting on sections of text. Document viewing and authoring systems store the annotations made by a user in the document file and use specialized knowledge of the document format to position the annotations. Annotating or working with the document typically requires permission to edit the document. If the author distributes copies of the document for feedback, each person annotates their own copy.

Adobe Acrobat [Ado], the eBook reader [MER], and XLibris [SPG98, SGP98] are examples of document viewing systems that support annotation. Each allows annotations on a specific document format and does not allow the document to change. The systems are designed for single users and do not explicitly support discussion. XLibris, a research prototype, supported free form ink annotation on documents displayed on a notebook computer. After a document was annotated, the XLibris system could generate clippings of the annotated sections, allowing the user to review, sort and filter the clippings. XLibris was used for a number of studies of *active reading*, which Schilit et al. defines as combining reading with critical thinking and learning [SGP98]. In one study with members of a reading group, users did not find the clippings of the text they annotated particularly valuable [MPG99].

Microsoft Word is a good example of a commercial document authoring program that has an annotation feature and stores annotations in its own internal document format. Word's commenting feature allows users to annotate any part of a document with a comment. Word comments are displayed with author information in popups and also in a separate window. Since Word requires an annotator to have write permission before commenting, it can insert an anchor for the comment in the document, making it robust to any edits in the document.

Multivalent Documents [PW97, PW98, PW01], DIANE [BHB+97], and the Knowledge Weasel [LS93] each support annotation of documents in several different formats. The Multivalent Document (MVD) project models documents as layers of content that are assembled by the system. The goal of project is to separate functionality from document format and provide a very extensible platform. The user interacts with a document through dynamically loaded program objects called behaviors, which are used to implement any desired functionality. Annotations, implemented using behaviors, are a core part of the MVD architecture, and the system supports annotating spans of elements and geometric regions in HTML, ASCII, and scanned documents [PW97]. Section 2.2.3 discusses MVD's algorithms for positioning annotations so they are robust to changes to the underlying document.

In DIANE [BHB+97], a research prototype focused on multimedia annotations, users could also annotate multiple document types. Users recorded annotation sessions, in which they could make text annotations on a variety of different documents types, links documents and also add voice comments. The recorded sessions could then be sent to other people to be played back, allowing the receiver to view and hear the annotations in the order recorded by the author.

The Knowledge Weasel Hypermedia Annotation system [LS93] was another research system that allowed annotating different document types using free publicly available software. However, in the Knowledge Weasel, users were required to hand author special files to link their annotations to the data file being annotated. The Knowledge Weasel also supported searching for particular annotations using a query language with syntax similar to C. While the goal of supporting annotation on a variety of document types using freely available software is admirable, the authoring and querying environment of Knowledge Weasel system appears quite complex.

The PREP editor [NKC+94] and Quilt systems [LFK88] differ from the other research systems in this group because they are specifically designed for collaborative authoring, and support annotation as a part of the authoring process. Both systems are designed to help coordinate multiple co-authors, support different types of interactions based on a user’s role in the authoring process, and include notifications triggered by specific actions (e.g., moving to a different paragraph). However, like the other systems, the PREP editor and Quilt both require documents in a particular format and that all co-authors use that system.

While these software systems with annotation features are valuable for many tasks, they are not as useful for discussion or sharing annotations among large numbers of people, since the annotations are tied in most cases a particular version of a document. However, we will discuss relevant systems from this group briefly in future sections, looking specifically at notification mechanisms (Section 2.2.2) and robust anchoring (Section 2.2.3).

2.1.2 Annotating HTML documents

The increasing popularity of the web has led to an astonishing number of annotation systems for HTML documents. Most web annotation systems make it simple for groups of people to view and make annotations on a document, facilitating asynchronous collaboration. Typically the annotations are stored separately from the document and are added to it when the user browsers to the document. Because annotations are stored elsewhere, many systems allow users to annotate any HTML document, even if they do not have permission to modify it.

While a complete survey of all systems would quickly become outdated, as two such surveys from 1999 have [Gar99, HLO99], web annotation systems can be broadly divided into three groups based on their implementation: server-based, proxy-based, or extensions of a web browser. In a server-based approach, only HTML documents on a particular server can be annotated. In a proxy-based approach, the HTML document to be annotated is accessed through a proxy, typically a particular URL, which provides annotation capabilities. Finally, a large number of systems provide annotation features by extending a particular web browser with a small software program that the user installs.

Table 2.1 lists examples of annotation systems in each group. The table shows:

Status: Whether the system is a commercial product (*C*) or a research prototype (*R*)

Availability: Whether the system is currently available (*Yes/No*).

What can be annotated: Whether the system supports annotation only at a certain places in the document, for example at certain HTML tags (*Tags*), only at certain locations specified in advance by a user (*Predefined*), only on sections of the HTML document (*Section*) or on any text in the document (*Any*).

Features to support discussions: Whether the system includes explicit support for discussion by showing the author of a comment (*Author*), allowing replies to annotations (*Threaded replies*), having an index of comments (*Index*) and/or giving annotations specific types, like questions (*Types*).

Notification strategies: Whether the system has subscription based notification, which is typically done using email (*Subscribe*), or informational notification where users are informed or can discover what annotations are new when they visit the system (*Inform*).

Technique for positioning annotations after changes to the document: Whether the system ignores changes (*Ignore*), attempts to reposition annotations after documents changes (*Attempt*), has explicit anchor points for annotations, pushing the requirement to avoid editing the annotation anchors to the user (*Explicit*), or information about anchoring mechanism is not available (*Unknown).*

Server-based Approaches

Annotation systems that take a server-based approach require uploading a document to a particular server, and occasionally augmenting it with some special formatting before it can be annotated. The document and annotations are then stored on that server. Examples of server-based systems include PageSeeder [Pag], CaMILE [GT00], DocReview [Hen] and Col•laboració [Col].

Table 2.1: Examples of HTML document annotation systems. The systems are grouped by their implementation approach: server-based, proxy-based or extensions of a browser. For each system the table shows: whether it is a research prototype (R) or commercial system (C), whether the system is available, where annotations can be located, features to support discussion, notification mechanisms, and how the system handles changes to documents that are annotated. Annotations can be located at specific tags in the document (Tags), predefined positions (Predefined), on sections (Section) or anywhere (Any). The systems handle document changes either by ignoring them (Ignore), attempting to reposition the annotations (Attempt), pushing the responsibility to the user (Explicit), or the information is not available (Unknown).

					Discussion Features				Notification		
	System (year developed)	**Status**	**Available**	**Annotation Locations**	Author	Threaded Replies	Index	Types	Subscribe	Inform	**Handle Document Change**
Server-based	CaMILE [GT00] (1994)	R	Yes	Predefined	X	X		X			Explicit
	Col•laboració [Col] (1998)	R	No	Section	X				X		Explicit
	DocReview [Hen] (1997)	R	Yes	Section	X		X		X	X	Explicit
	PageSeeder [Pag] (~1998)	C	Yes	Tags, Predefined	X	X		X	X		Explicit
Proxy-based	Annotation Engine [Ann] (2000)	R	Yes	Any	X	X					Ignore
	Annotator [OAM99] (1999)	R	No	Any	X	X	X			X	Attempt
	CoNote [DH95b] (1994)	R	No	Predefined	X	X					Explicit
	CritLink [Cri] (1997)	C	Yes	Any	X			X	X		Attempt
	GrAnt [SMB96] (1996)	R	No	Any	X	X					Ignore
Browser Extensions	Annotea [KKP+01] (2001)	R	Yes	Any	X						Ignore
	ComMentor [RMW97] (~1996)	R	No	Any	X	X				X	Attempt
	E-Quill [EQu] (~2000)	C	No	Any							Ignore
	IMarkup [IMa] (~2000)	C	Yes	Any	X		X			X	Unknown
	Office Web Discussions [MOWD] (~2000)	C	Yes	Tags	X	X			X		Ignore
	Web Highlighter [Phi02] (2002)	C	Yes	Any							Unknown
	WebVise [GSØ99] (1999)	R	No	Any							Attempt
	Yawas [DV00] (2000)	R	Yes	Any	X		X				Ignore
	WebAnn (2001)	R	No	Any	X	X	X				Attempt

PageSeeder [Pag] is a commercial annotation system that allows users to comment at specific locations in a document. After uploading a document to the PageSeeder server, *seed locations* for comments are automatically added at the end of every paragraph, and the document author can also manually specify additional annotation locations. Users then make comments at a particular seed. Each comment has a type, either general, question, answer or reply. For each seed with comments, an index of subject lines is displayed; clicking on a particular subject line opens the comment in another browser window. Users can subscribe to receive new comments on a particular page in email, and then can send replies using email. To ensure appropriateness, PageSeeder allows pages to have a *moderator* who screens comments before they are added.

In CaMILE [GT00], a research system from Georgia Tech, comments can also be made only at specific locations. CaMILE has been used in educational settings to provide a forum for online class discussion. Professors typically add the anchor points for discussion to their web pages. Clicking an anchor opens a separate web page where students can participate in a threaded discussion related to the anchor point. Similar to PageSeeder, annotations made using CaMILE have a particular type. The system's default types are question, rebuttal, revision, comment, and new idea, but additional types can be added. CaMILE has been used for a number of different educational applications, including discussion of class assignments and giving feedback on student design projects. Studies of student usage of CaMILE will be discussed in Section 2.2.1.

Unlike PageSeeder and CaMILE, annotations in the DocReview [Hen] and Col•laboració [Col] systems are made on entire sections of a document rather than at locations specified by the document author. In DocReview the document must be manually divided into *review segments*, typically a paragraph long, before it can be annotated. Then by clicking on a link labeled *W* at the beginning of the review section, users write annotations that are displayed in a separate browser window. Annotations can be read by clicking on the *R* link. DocReview includes two types of notifications. Users can subscribe to receive email notifications. The system also displays the number of comments on the document at the top of the page, informing users of how much activity has occurred.

Col•laboració also supports annotating sections of a document, but is designed for collaborative online authoring of documents rather than online discussion. In Col•laboració, users first create a document outline with section headers. They can then write and discuss different sections. Any person working on the document can save a version with an optional comment that can be revisited later. Col•laboració allows users to specify which other users should be sent an email notification about a particular comment. The system also includes a chat mechanism for synchronous collaboration.

Proxy-based Approaches

Proxy-based approaches allow users to annotate any HTML document on the web through a proxy server. Users typically access documents they are interested in through a specific URL that acts as an intermediary between the user and the web page. This allows the proxy server to add a user interface and merge existing annotations with the page before displaying it to the user. CritLink [Cri], Annotation Engine [Ann], the GrAnt prototype [SMB96], Annotator [Anno, OAM99] and CoNote [CoN, DH95a] are all examples of the proxy-based approach.

CritLink [Cri], accessed through crit.org, was one of the earliest systems to allow annotation on any web document. In CritLink, comments can be made on any text in the document. The user creates an annotation by copying the part of the document she wishes to annotate into a separate web form. The interface tells users to select enough text to annotate so that the text is unique in the document. CritLink's strategy for anchoring annotations will be discussed further in Section 2.2.3. After the user creates an annotation, icons denoting the its type (either comment, query, issue, or support) are then inserted inline around the text that was annotated, and the full comment is displayed at the bottom of the page. CritLink provides email notifications where users can sign up to receive messages when comments are made on particular document.

The Annotation Engine [Ann], another proxy-based system inspired by CritLink, was developed at Harvard Law School. Similar to CritLink, the user creates an annotation by copying the text she wants to annotate into the annotation creation dialog and then associates a note with it. The system also supports creating links from the document to other web pages. To display annotations, the Annotation Engine uses frames and shows a list of comments on the document in a frame on the left side of the browser. Users can reply to particular annotations to start a discussion.

In the GrAnt prototype [SMB96], as in CritLink and the Annotation Engine, the user explicitly cuts and pastes the section of the document she wants to annotate into a dialog box to create a new annotation. The designers focused on GrAnt being accessible from any browser, so they proposed a stream transducer architecture to intercept and alter the HTML document retrieved from the server to add the user interface and annotations. However, based on experience with the prototype, they noted problems with this method, in particular having the user interface only at the end of a document. They suggest that other methods for maintaining cross-browser support, such as a Java applet, might be more appropriate in order to have more control over the user interface.

Creating annotations in the Annotator system [Anno, OAM99], developed as part of the USC Brian project, requires more work than with the previous systems because the user must edit the document using Netscape Communicator. After editing, the user submits the document to a proxy server that parses the document, and removes and saves the annotations. The Annotator system includes a Java applet that displays an index of annotations on a document and the user interface for searching and managing annotations in a separate browser window.

The CoNote [CoN, DH95a] system differs from the previous proxy-based systems because it requires document authors to specify the annotation points that can be annotated. Each annotation point then has an index of links for any annotations made at that point. In CoNote each person has a role: either as a viewer, reader, user and author, and access control is based on role. Any annotations made by the document author are shown with a special icon to draw attention to them. CoNote has been used in numerous classes at Cornell University. Studies of CoNote will be discussed in Section 2.2.1.

Browser-Based Approaches

Many annotation systems extend a particular web browser to allow users to annotate any document on the web. With these systems, users install a small software program that adds a user interface for creating annotations to their web browser and then can make and view annotations as they read HTML documents. Similar to the proxy-based approach, most of the systems store annotation on a separate server facilitating sharing and allowing users to annotate documents they do not have permission to modify. However, by extending a particular browser the systems can often take advantage of non-standard features of that browser. In this section I survey some interesting examples of browser-based systems, from early research prototypes to commercial systems. WebAnn, the system I wrote, extends the Internet Explorer web browser and will be described in detail in Chapter 3.

Two early research prototypes, WebVise [GSØ99] and ComMentor [RMW97], both supported annotations by extending the browser. WebVise extended a number of Microsoft applications to support adding notes that look similar to post-its and creating links from text in a document to other web pages. The annotations could also be viewed using a Java applet or through a proxy server. The focus of the WebVise research was saving redundant location information for the annotations so they could be repositioned in edited documents. The WebVise anchoring approach will be discussed further in the section on robust positioning algorithms.

Researchers developed the ComMentor system by extending the Mosaic browser and they also experimented with placing the annotation functionality in a proxy server. ComMentor logically divided annotations into sets, where each set was stored on a particular annotation server and had certain access control restrictions. Annotation sets could be private, public, or visible to a group of people. Users designated the sets they were interested in and annotations from those sets were displayed when they browsed a web document.

While research on WebVise and ComMentor has ended, a number of browser-based annotation systems are currently being developed or available commercially. Annotea [Annot, KKP+01] is a research system under development by the W3C. One major goals of Annotea is to define an infrastructure to support shared annotations that reuses existing W3C technology, including RDF [RDF], XPointer [XPL], and XLink [XLi]. The Annotea prototype extends the Amaya [Ama] browser developed by the W3C to allow users to create annotations on any text in the document. The annotations are stored on a separate server to facilitate sharing. Annotea also focuses on performance and waits to download annotations until after the document loads. In addition the body of an annotation is only downloaded and shown in a separate window if the user clicks on that annotation's anchor point in the document.

Microsoft Office Web Discussions [MOWD] is a commercial annotation system available with Office 2000 that allows users to add annotations inline at the end of every paragraph in an HTML document. Annotations are stored on a central server so that everyone connected to the same server views each other's comments. Web Discussions includes threaded replies and email notifications. I worked with Web Discussions in my field study of notifications. The system is described in further detail in Chapter 4.

Two other commercial systems, IMarkup [IMa] and E-Quill [EQu], focus more on annotations drawn using a pen-like tool. Using the IMarkup's plugin to Internet Explorer users can make annotations that are free form drawings, text on sticky notes, highlights, or voice annotations. The product is primarily aimed at quick document review rather than discussing text, and therefore does not offer threaded replies. However users can filter comments based on author. E-Quill [EQu], which was purchased by Microsoft in 2001, supported similar annotation capabilities to

IMarkup. Using the E-Quill plugin, users could draw on web pages using a pen-like interface, attach notes similar to post-its, and highlight text. A copy of the page with the annotations was then stored on E-Quill's server and users could share their annotations by emailing the URL of the stored web page.

While the previous systems store annotations on a separate server to facilitate sharing, Web Highlighter [Phi02] and Yawas [DV00] are both extensions to Internet Explorer that store annotations in a local file. The Web Highlighter plugin allows the user to make highlights, notes, or personal links on any text in the document. While a user's annotations are only viewable on a particular computer, Web Highlighter includes an export feature so users can share annotations with other people using Web Highlighter. The Yawas system provides similar functionality to Web Highlighter, including exporting annotations to share them. In experimenting with Yawas, the designers investigated using annotations to improve information access, retrieval, and document clustering.

2.2 Using Annotations for Asynchronous Collaboration

As the previous section illustrates, digital annotation systems are very common. While the prevalence and variety of the systems highlights the value of annotations, there have been relatively few studies using annotations for asynchronous collaboration. This section discusses in more detail the systems and studies related to the annotation issues I focus on: online discussions in an educational setting, notification strategies, and techniques for positioning annotations on documents that change over time.

2.2.1 Online Discussion in an Educational Setting

Marshall's studies of annotations in textbooks show empirically that students already use annotations [Mar97, Mar98]. With the increase in support for annotating digital documents, shared annotations appear to be an effective method for online discussion because the context of a student's comments will be clear to other readers. Newsgroups and online discussion boards have long been used to supplement class discussions, and it seems that annotations might provide an even better way to engage students with class materials outside the classroom by allowing them to make comments in a specific context. However, studies of online annotation systems have focused primarily on using annotations to ask questions about class assignments or give feedback on student work rather than on discussion.

Students initially used CoNote [DH95b] to annotate lecture notes, problem sets, and handouts in a computer science course at Cornell University. Anecdotal evidence gathered by the course instructors suggests that the annotations provided a place for students stuck on the problem sets to ask questions quickly, and showed students that others were also confused. A more formal study [GSM+99] was done in a class in which students were required to use CoNote to comment on project sites constructed by other students. The study found that the majority of students felt it helped them make better websites, but surprisingly did not think CoNote helped them learn. In the study gender had an influence on the student's response, with a higher percentage of women responding that CoNote helped them learn and create better sites.

CaMILE [GT00] has also primarily been used to review class materials. One study compared the use of CaMILE and newsgroups in a large number of classes at Georgia Tech and found that the average thread length in CaMILE was significantly higher than in the newsgroups. This suggests perhaps CaMILE, in which discussion is anchored to a particular context, encourages longer discussions. However, since the course topics varied and each class used either CaMILE or a newsgroup, the comparison between the two types of discussion formats is very general, unlike my study described in Chapter 3. In my study students in a single class used both annotations and a discussion board.

While designing the PREP editor, Neuwirth et al. [NKC90, NKC+94] explored in depth using of annotations for collaborative writing. In particular they focused on giving feedback to students on paper drafts. Another PREP study [WNB98] explored the effect of different user interfaces on annotation behavior, finding that the interface affected the type and number of problems student writers identified in a manuscript. Wolfe [Wol00] also worked with students and explored the effect of positive and negative content in annotations by seeding materials with evaluative annotations, both positive and negative. The students then wrote essays using the annotated materials. The study findings strongly suggested that the annotations' content influenced the reader's perception of the materials. In contrast to this approach, in my study students used annotations to discuss papers rather than to inform their writing process.

In educational settings the focus has so far primarily been on using annotations for feedback and questions about assignments. This may be due to where students can comment on the documents. For example, both CaMILE and CoNote allow annotations only at specific locations, typically determined upfront by the instructor. This affords discussion of particular problems on a homework assignment or general feedback. In contrast, in WebAnn, the system I wrote, users can easily associate comments with a particular paragraph, phrase, or word of their choosing. While one solution may not be ideal for every setting, WebAnn allows exploration of the way people respond to the ability to identify precisely the context for a discussion.

One exception to this focus on feedback and questions about assignments is CLARE [WJ94]. CLARE was used to collaboratively analyze scientific papers online through the creation of a highly structured hypertext of labeled annotations describing portions of the text as problems, claims, evidence, theory, concepts, and so forth. It also allows more general annotations categorized as critiques, suggestions, and questions. The interface and process of use of CLARE is very different from that of WebAnn. Students first privately analyze the text, and then view each other's analyses, at which point they can comment by creating new links. However, CLARE resembles WebAnn in the way that it anchors annotations on online representations of core course content.

Anchoring and context in education are used in a different sense in the anchored instruction paradigm developed by the Cognition and Technology Group at Vanderbilt [CTG93]. They discuss using materials such as videos to anchor assignments in the context of students' daily lives. I focus instead on anchoring discussions in the context of the text being discussed.

2.2.2 Notifications Mechanism for Annotations

Notifications and other methods of maintaining awareness of changes to a document have long been recognized as important aspects of both synchronous and asynchronous document collaboration systems. A study of collaborative writing by Baecker et al. [BNP+93] stressed the importance of *mutual awareness*, which they defined as the knowledge of the state or actions of collaborators. Dourish and Bellotti [DB92] discuss the importance of *passive awareness*, "an understanding of the activities of others, which provides a context for your own activity" [DB92]. More recently a large-scale study of activity in BSCW [App01], a groupware system that supports shared workspaces, identified awareness features as the second most common group of operations used by frequent users.

Awareness of Document Activity

Document collaboration systems and document annotation systems support awareness in three main ways: by providing information about what has changed since the last visit (Informational), by allowing subscription to explicit change notifications (Subscription), and by passively displaying information about changes on the periphery of the user's display (Peripheral Awareness).

Informational

Informational methods update users on what has happened since their last visit to the system, but rely on use of the system to discover changes. A system can generate information about changes that have occurred since a person last visited automatically or by using comments explicitly entered when a change is made. In BSCW [App99], icons indicate recent document activity: reading, editing, or versioning. Clicking on an icon retrieves additional information about time and actor. Other document systems, such as Lotus QuickPlace [Lot], provide similar change information explicitly on a separate web page.

POLIwaC [Fuc99] also uses icons (and colors) for the lowest of its four intensity notification mechanisms. As the intensity levels increase, the user is notified with enlarged icons, scrolling messages, and dialog boxes. POLIwaC supports synchronous and asynchronous notifications. People in a shared workspace can be notified immediately or the next time they enter it.

The Annotator [OAM99] and ComMentor [RMW97] annotation systems allowed people to search the set of annotations made on a document. This provides information about new annotations, but requires additional work by the user.

In contrast to informational methods, the notifications discussed in Chapter 4 are subscription-based and inform users automatically of changes that have occurred.

Subscription

Many document collaboration and annotation systems that provide notifications (e.g., Quilt [LFK88], Crit.org [Cri], Web Discussions [MOWD], Intraspect [Int], BSCW [App99], Livelink [Liv]) allow users to subscribe to changes in documents, in folders, or specifically for document annotations. Users typically choose whether to be notified immediately or to receive a daily or weekly bulk notification. The notifications are primarily delivered using email. Quilt [LFK88] allowed users to specify the degree of change to a document -- for example, substantial -- that they wanted to be notified about. Users of Intraspect [Int], an enterprise collaboration system, can also be notified about changes via their personal web pages. It includes a "Tell People" function that allows a user to send email notifications directly to other people. Notifications in Web Discussions are described in further detail in Chapter 4 as part of my field study of notifications.

Peripheral Awareness

The goal of peripheral awareness is to make appropriate awareness information available to a user at a glance without being overly distracting or taking up too much screen real estate. Annotation systems have not traditionally used peripheral awareness for notification. However, other research has explored providing information using peripheral awareness. Dourish and Bellotti [DB92] discussed shared feedback that passively distributes information about individual activities to others in a shared workspace. For example, each user of ShrEdit, a multi-user text editor, has a cursor within a shared window and can thus see what others are doing. Gutwin et al. [GRG96] have studied "awareness widgets" such as miniature views in shared workspace groupware systems. BSCW provides an EventMonitor that can be used for realtime presence and activity awareness [KA98]. These systems focus on synchronous collaboration; Dourish and Bellotti suggest that information available peripherally might be valuable in systems that support both synchronous and asynchronous work modes.

In my study of notifications for an asynchronous document annotation system discussed in Chapter 4, I provide awareness through information that is always peripherally visible. This resembles the visibility at a glance available in the synchronous environments described above.

Studies of Notifications

A recent study of BSCW found that awareness features are very popular, particularly among frequent users of the system [App01]. The study's authors suggest that it takes time to adjust to the features used to co-ordinate asynchronous work.

Cadiz et al. [CGG00] observed the use of the Microsoft Office 2000 Web Discussions annotation system by about 450 people over a 10-month period. They mention the use of email notifications: Some users felt that they checked the document enough and did not need notification; others wanted notifications with more detailed information about the content of new annotations.

The prevalence of features to support awareness suggest its importance for collaboration around documents, but there are few studies of awareness features, and very few of notifications in shared annotation systems. The study described in Chapter 4 aims to redress this imbalance.

2.2.3 Robust Annotation Anchoring

Effectively positioning annotations in a digital document is a challenging problem. The exact document text related to an annotation is often ambiguous. For instance, Marshall [Mar98] suggests that people frequently place their annotations carelessly. The underlines and highlights they create (on paper in this case) often follow document structure or typographical characteristics rather than content. The positioning problem is even more difficult if the underlying document can be modified. In the Cadiz et al.[CGG00] study, while they observed many benefits of using annotations to comment on documents, a key complaint was the *orphaning* of annotations. That is, when the online documents changed, the annotations lost the link to their proper position within the document, and were presented at the bottom of the document.

In a more general context, the hypertext research community has extensively explored the issue of maintaining link integrity and annotations can be viewed as a special case of links. Open hypermedia systems such as Microcosm [HHD93], DLS [CDH+95], and HyperTED [VCH+94] store links, either user authored or computed by the system, externally from documents in separate linkbases [CHD99]. As Davis [Dav99] outlines, this model has several advantages including the ability to adapt third party applications, make links in read only data, and create generic links, where a link is available from every location of a particular item in the document. For example, every instance of a word might link to its dictionary definition. However, similar to orphaning issues for annotations, links also have what Davis [Dav99] terms the “editing problem” or the “content reference problem” where links lose their location if the document changes. Both Davis [Dav99] and Ashman [Ash00] enumerate ways hypermedia systems have tried to handle this problem, including publishing documents as read-only, using editing tools aware of links, supporting generic links, and attempting to correct broken links, the most relevant for robust annotation anchoring.

Existing annotation systems use three approaches for positioning annotations described in more detail in the following sections. The systems either ignore any changes to the document (Annotating Frozen Documents), restrict where users can annotate (Annotating Predefined Positions), or use more complex positioning algorithm to attempt to cope with changes (Adapting Annotation Positions).

Annotating "Frozen" Documents

Many annotation systems simply assume that annotated digital documents will never change. Adobe Acrobat Reader [Ado] and Microsoft eBook Reader [MER] are examples of this approach. Other systems such as NotePals [Not], Dynomite [WSS97] and XLibris [SPG98] have augmented traditional annotation of paper documents (which don't change) with computer support. In both types of systems, annotations are typically positioned using very simple means, such as character offsets, or page number plus an (x, y) position. The underlying document is never modified, so annotations never have to be repositioned.

Other systems do not explicitly require documents to remain unchanged, but work best when there are no modifications. In Table 2.1 these systems have *Ignore* in the Handle Document Change column. Users can typically create annotations on any web page, which are stored separately on a central server. The annotations are positioned by calculating a signature from some of the content on the page to which the annotation belongs. E-Quill [EQu], and Microsoft Office Web Discussions [MOWD] are commercial systems that have taken this approach; public web-scale architectures such as OSF [SMB96] and NCSA [LB97] do as well.

The hypertext community terms this approach the "publishing model" [Ash00, Dav98]. Any documents published to the system are read-only. While appropriate for some documents, such as published papers, in many important scenarios, such as accessing web pages, however, it is unrealistic to assume that documents will never change. If a document does change, these systems fail to properly position some annotations, and the annotations either silently disappear or are displayed in a separate window as orphans. Not surprisingly, this problem has been found to be particularly debilitating. In a study of the large-scale use of Microsoft Office 2000 Web Discussions, lost annotations was cited as the primary reason people stopped using the system [CGG00]. The robust anchoring work described in Chapters 5 and 6 aims to accommodate annotating documents that may be modified.

Annotating Predefined Positions

Some systems attempt to compensate for potential modifications in web pages by only allowing users to annotate predefined positions. These systems, labeled *Tags* in Table 2.1, limit the places where annotations can be placed to better control how annotations are positioned when the underlying page gets modified. For example, CoNote [DH95b] requires inserting special HTML-like markup tags before a document can be annotated. PageSeeder [Pag] only allows users to attach annotations to a small selection of HTML tags. One goal of my robust positioning work is to allow users to position annotations anywhere they would like on a digital document.

Adapting Annotation Positions

Several systems, labeled *Explicit* in the handle document changes column in Table 2.1, including Microsoft Word [MWo], the PREP editor [NKC+94] and DIANE [BHB+97], maintain documents in their own internal format, which allows them to insert explicit anchors for annotations into the document. These annotations then remain positioned appropriately when the document is edited as long as the anchor is not deleted. However, users may only annotate documents within the particular system. More importantly, users must have the appropriate access to the document to make annotations. I am interested in systems that allow annotation without these restrictions.

A number of systems, including Annotator [Anno, OAM99], ComMentor [RMW97], WebVise [GSØ99], and Robust Locations [PW00] (part of Multivalent Annotations [PW97]), allow annotations to be positioned anywhere within a web page. These systems, labeled *Attempt* in the handle document changes column in Table 2.1, typically save a combination of the text annotated by the user (*anchor text*), and text surrounding the user's annotation (*surrounding context*). The annotations created by these systems are robust to varying degrees. Each system can fail to correctly position an annotation in a modified document and orphan it. The systems have varying strategies for presenting orphans to the user, from separate popup windows [PW00] to placing them at the end of the document [RMW97].

In the rest of this section I describe three different approaches to positioning annotations: using information about both the document structure and content of the anchor text selected by the user, using only information about the selected content, or using the annotation's position relative to the underlying document structure.

Saving Structural and Content Information

Robust Locations [PW00] and WebVise [GSØ99] both save structural and content information and have the most detailed robust anchoring strategies. In Robust Locations, points in the document are identified redundantly using unique identifiers, including position in the HTML document tree structure, and context. Span anchors, which contain two robust points and the entire anchor text selected by the user, anchor the annotations in the document. To find an annotation's position in the document, Robust Locations first tries the unique identifier, then the tree walk information, and finally uses the context. Its developers describe one method of presenting orphans to the user and some initial testing.

WebVise is an open hypermedia service that supports collaborative annotation and allows creation of links to parts of web pages. WebVise uses a LocSpec that saves anchor information for a variety of media types. For a span of text the LocSpec contains a reference (ID or HTML target name), the anchor text, some surrounding text, the length of the anchor text, and its start position in the document. Unfortunately, their positioning algorithm is not described in detail, and consists primarily of alerting the user when exact match of the anchor text is not found.

The Keyword Anchoring algorithm described in Chapter 6 uses some of the same information as Robust Locations and WebVise. However, the algorithm focuses on meeting user expectations by using keywords instead of document structure. Keyword Anchoring could easily be used in conjunction with the approaches taken in Robust Locations and Webvise to provide additional robustness that may better meet user expectations.

Saving Only Content Information

Using only information about the content of the anchor text selected by the user makes the robust anchors independent of document format. ComMentor [RMW97], Annotator [Anno, OAM99], CritLink [Cri], and Annotation Engine [Ann] all use content information for robust anchoring. ComMentor and Annotator use a unique substring from the anchor text to search for a new position for the annotation in the document. HyperTED [VCH+94] and the Microcosm Universal Viewer [DKH94], both open hypermedia systems, also used unique substrings and search for anchoring links in documents. While this approach will be robust when the anchor text moves in the document, it is unclear how easily a new position will be found if the unique substring in the document has been modified or another instance of the substring has been added elsewhere in the document.

CritLink saves the entire anchor text and some amount of text surrounding the anchor text. The user interface in CritLink tells users to select enough text to annotate so that the text is unique in the document. To locate an anchor, CritLink searches for the entire anchor text in the modified document and uses the surrounding context to distinguish duplicates. The Annotation Engine's approach is similar to CritLink, but it does not save surrounding context. These methods will find the anchor text if it moves, but appear not to handle modifications to the anchor text.

In contrast to these approaches, Keyword Anchoring focuses on finding appropriate positions that meet user expectations precisely when the annotation's anchor text has been modified. The keyword approach is more flexible than the use of substrings, and keywords can be selected faster than a unique substring.

Saving Only Structural Information

Using information about the annotation location in the document's underlying structure is another method for anchoring annotations. Annotea [KKP+01] uses XPointers [XPL] to specify the position of an annotation. XPointers save structural information about the location of an annotation. The specification does not discuss robustness, aside from stressing that unique identifiers are most likely to survive document modifications. However, as the authors of the Annotea system observe, XPointer positions can be orphaned or incorrectly positioned if the document changes. Using identifiers as the primary anchoring method requires cooperation from document authors and reduces the number of documents that can be annotated compared with methods that save content information.

2.3 Summary

This chapter has described the wide variety of annotations systems. In my discussion, I have focused on web annotation systems because of their importance for facilitating asynchronous collaboration. These systems can be grouped based on their implementation approach, as either server-based, proxy-based, or an extension of a web browser. Table 2.1 outlined examples of each approach.

I have also discussed in more detail systems and studies related to the questions I explore in my research: the value of annotations for discussion in an educational setting, staying aware of annotations, and robustly anchoring annotations to documents. In an educational setting, annotations systems, such as CoNote, CaMILE, and PREP, have been used primarily for feedback on student work. In contrast, Chapter 3 describes my study of online discussion using annotations.

Several annotation systems have previously included some type of notification, typically either by providing information about what has changed since the user's last visit or allowing the user to subscribe to change notifications. However, there has been little study of user needs for notifications and other mechanisms for maintaining awareness of annotation activity on a document. In Chapter 4, I describe the software I built and study I conducted to explore the best way to notify users of new annotations.

Finally, positioning annotations in a digital document so that they are robust to change is a challenging problem. Current web annotation systems take one of three approaches, either ignoring any changes to a document, restricting where users can annotate in hopes of limiting the effects of document modifications, or using more complex techniques to attempt to adjust annotation positions. While these approaches are robust to varying degrees, no research has explored what users expect to happen to their annotations when the document changes. In Chapter 5, I describe two studies I conducted to help understand user expectations for annotations on documents that are modified. Chapter 6 introduces the Keyword Anchoring algorithm that was designed based on those expectations.

Chapter 3

Annotations for Asynchronous Discussion

Tools that support asynchronous discussion allow discussion to occur when and where it is convenient for users. In the educational domain, newsgroup and online discussion boards have long been used to supplement class discussions. However, the use of these systems has often been limited and optional, because not all students were assumed to have access to a system or the skills to use it. Today these restrictions are fading away, and the increasing ubiquity of online access allows instructors to incorporate asynchronous discussion directly into their classes. For example, students can begin discussing a reading assignment while they are reading it, instead of taking notes and waiting until they get to the classroom to express their reactions.

Annotations seem to be a very natural way to support asynchronous discussion, since the context of an annotation will be clear to other readers. A shared annotation system that allows a discussion to be anchored directly on a reading assignment potentially offers many advantages over other systems, such as newsgroups or discussion boards, in which the discussion is divorced from the context of the document.

To evaluate the use of annotations for asynchronous discussion I performed a field study of online class discussion comparing WebAnn, a shared annotation prototype system I wrote, to EPost [Epos], a web-based threaded discussion board. During the field study, online discussions easily surpassed the required participation level, particularly with WebAnn where students contributed almost twice as much to the discussion. For a variety of reasons, including access, the students slightly preferred using EPost. The study identified enhancements that would improve WebAnn and also important considerations for the process of using it in an educational setting.

In the rest of this chapter I present the WebAnn system and describe the study comparing WebAnn and EPost. Section 3.1 describes the WebAnn system. Section 3.2 discusses the field study and Section 3.3 presents the study results. Important issues and options to consider when incorporating online discussion into a class are discussed in Section 3.4. Section 3.5 concludes with some reflection on the potential for anchored online discussion using annotations.

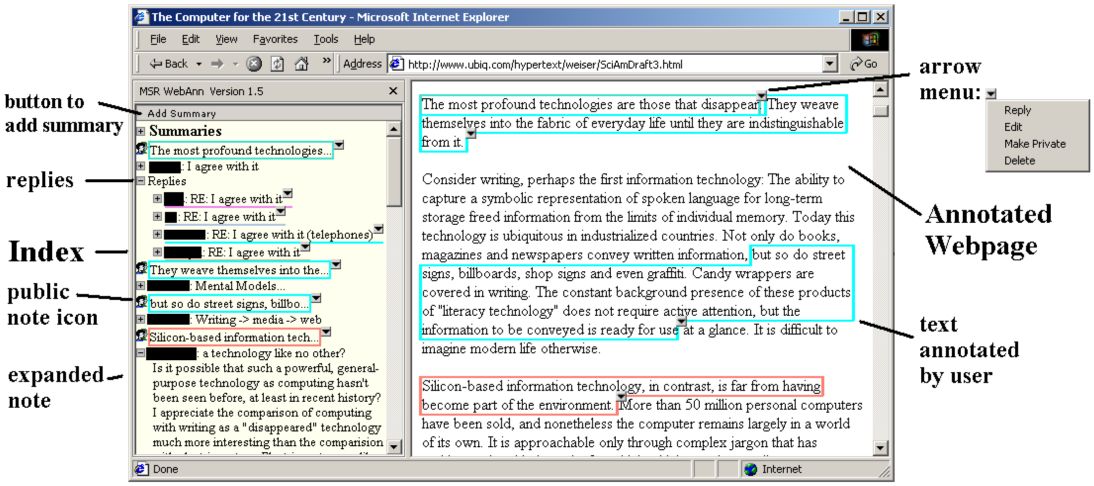


Figure 3.1: WebAnn interface embedded in Internet Explorer. On the right is the webpage being annotated, on the left is the index of notes and replies. Student names are blacked out to provide anonymity.

3.1 WebAnn

In a traditional online discussion board, a student’s post is divorced from the context of the assignment. To contribute a comment or question, students must manually reconstruct the context of their remarks before making them. That is, they must identify not only the paper or document being discussed, but perhaps also the section, paragraph, sentence, or word. Only after this is done can a discussion thread ensue.

WebAnn takes a different approach. It supports fine-grained annotation of text on web pages, so that students’ remarks can be made and seen in the context that inspired them. Furthermore, annotations are shared, and can serve as anchors for threaded discussions. In this way, discussions around class material outside of class are captured for all to see, *and* they are directly linked to the materials—and the precise location within the materials—that they reference.

The process of use envisioned for WebAnn is as follows. A student reads a paper online, and can at any point identify some text and type in a comment or question. It can either be a personal note or an entry in a public class discussion. The student will also see annotations left in the class discussion by previous readers, and can reply to those. With this facility questions can be asked or answered, opinions made known, issues identified, and discussions started. Using the WebAnn system, students can more easily participate in discussions of class materials, and discussion outside the classroom will flourish.

3.1.1 User Interface

WebAnn is embedded in Microsoft Internet Explorer. As shown in Figure 3.1, the web page being annotated is displayed on the right, and discussions on the page are shown in the index window on the left. To create a new note, a user selects the text to be annotated and chooses to “Add a Note” from a popup menu. A dialog box appears, into which the user types the note. Notes can optionally be made private so they are visible only to the user that authors the note. The new note is added to the index, and the text to which it corresponds in the page is outlined with a color unique to that user.

Once an annotation has been created, it is available as a navigational aid in both the index and the page, so that clicking on it in the index, for instance, scrolls the web page until the outlined text is in view. Later on, the user can go back and edit or delete his or her notes (provided they do not have replies). To add a global note that applies to the entire web page, the user clicks the “Add Summary” button and follows the same procedure.

Threaded discussions grow when users reply to existing notes. To reply to a note, a user clicks the arrow menu next to the note (either in the index or the web page) and chooses “Reply.” Replies are added to the index directly below the original note.

3.1.2 System Implementation

WebAnn has a client/server architecture and can be easily installed on any computer running Windows 2000. WebAnn is adapted from a system I built to experiment with robust annotations on web pages that will be discussed further in Section 5.3.1. The WebAnn client provides the features described in the previous section by extending the Internet Explorer browser with a Custom Explorer Bar. Users first install a small program that registers WebAnn with their browser, and then Internet Explorer lists WebAnn as an Explorer Bar in the View Menu of the browser. Users then select the WebAnn menu option to display and make annotations. Other common browser functionality such as search and history are also implemented as Explorer Bars.

Annotations are specific to a particular URL, so once the user turns on WebAnn, the client contacts the server to retrieve any annotations for that URL. The server is part of the common annotation framework (CAF) implemented by David Bargeron, which provides persistent annotations in an internet-based store. More details regarding the CAF architecture are available [BG01].

After retrieving annotations for the current web page, the WebAnn client positions the annotations in the page and outlines the text that has been annotated in the user's local version of the web page. WebAnn displays the content of the annotations (the comments made by the users) in its pane on the left side of the browser, as shown in Figure 3.1. Annotating web pages using WebAnn does not modify the page, since any user interface additions are made to the local copy of the page and the annotations are stored on the CAF server. This means users can annotate any text on any page on the web.

To position annotations in the web page, the client uses the annotation's anchor information. Each anchor contains an Internet Explorer bookmark for the text associated with the annotation as well as additional information to help position the annotation if the page has changed. Chapter 6 describes in detail the exact anchoring information saved by my Keyword Anchoring algorithm when the user creates a new annotation.

When the user creates a new annotation, the client gathers the appropriate information, including the user's comment and positioning information, and automatically saves the annotation on the server. If the user browses to a previously annotated web page, the client retrieves and displays any annotations for that page.

3.2 Field Study of Online Discussions

To examine the tradeoffs between discussions anchored in-context and traditional discussion boards, I compared the use of WebAnn to the EPost [Epos] threaded discussion board system in a graduate-level Human Computer Interaction (HCI) class taught at University of Washington during Spring Quarter, 2001. My advisor, Alan Borning, taught the class and I served as the teaching assistant (TA).

EPost, a University of Washington (UW) "Catalyst" tool, is a high-quality web-based threaded discussion board similar to a traditional newsgroup browser. EPost discussion boards are widely used at UW, and can be accessed using any HTML browser. As shown in Figure 3.2, the left side of the EPost interface lists the original messages and replies, while the contents of the selected message are displayed on the right. To post a message, the user clicks on the "Post New Message" link at the top left, and to reply on the "Reply to Message" link. EPost supports several

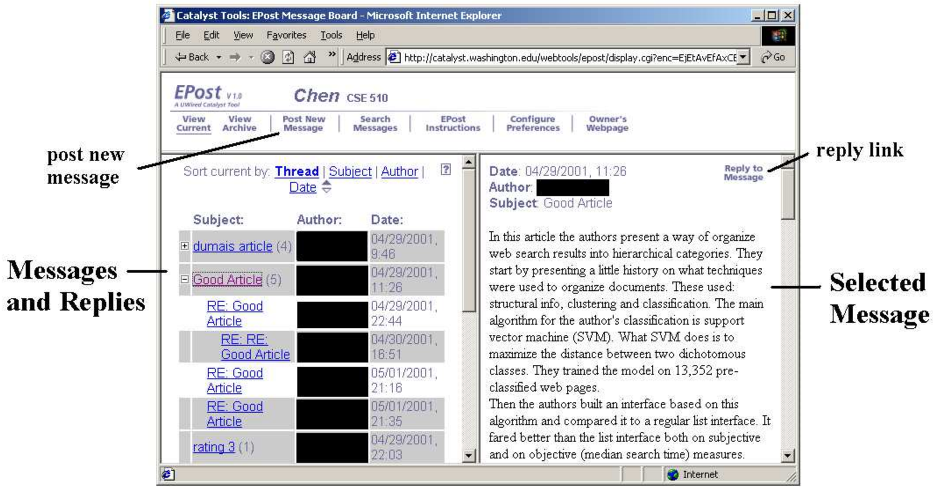


Figure 3.2: EPost [Epos], a threaded discussion board from the UW Catalyst toolkit. The left pane is the index of posts and replies. The right pane is the text of the selected message. Student names are again blacked out.

useful features, including filtering and notification options. For the class I studied, an EPost discussion board was created for each reviewed paper.

3.2.1 Study Design

The goal of the study was to assess the efficacy of WebAnn for promoting educationally valuable discussion outside of class as compared with EPost. Also, I was interested in how online discussion affected subsequent discussion in class. Would the use of WebAnn increase the overall amount and quality of discussion and participation? Other studies have found that anchored discussions lead to longer threads and greater participation [GT00]. Beyond this, would WebAnn engage students more by encouraging more specific and pointed comments and stimulate more engaging discussion in class? I expected that the online and in-class discussions would be complementary, since the online discussion could be used to inform the in-class discussion.

During the class students were assigned to read 2 or 3 papers per week. For each paper, students wrote a brief review with two parts: a summary, and some personal reactions to the paper. Students were also required to respond to at least one other student's comments for each paper. The total length of each review was expected to be equivalent to two or three paragraphs, and students could skip up to five reviews during the quarter. Reviews were 25% of a student's course grade, and it was expected that all students would receive full credit for them. Previous offerings of the course had a similar requirement, except for the requirement to post a reply to another student's comment.

This assignment format is a particularly good one for discussion systems such as WebAnn and EPost. To submit paper reviews using EPost, students posted a message containing both their summary and comments, and then replied to other students' posts. With WebAnn, students used the "Add Summary" button for the summary, and then anchored their comments and reactions throughout the paper by adding note annotations and at least one reply.

The class met Mondays and Wednesdays for 80 minutes each day. Student reviews were due on Tuesday at noon, and responses were due before class on Wednesday. Originally the plan was to make the reviews and responses due before the Monday class, but the inability of some students to access WebAnn from home forced the later due dates. Six women and five men enrolled in the class. Four students were from Computer Science; the others were from the Information School, Psychology, and Medical Informatics. One student was an employee of the UW library system.

The class alternated between using EPost and WebAnn in two-week blocks. During the 10 week quarter, students spent at least 4 weeks using each system, with an extra week on EPost at the beginning of the quarter and a final week of presentations when there were no paper reviews. For WebAnn weeks, all papers were available in html to enable online annotation.

I fixed basic WebAnn problems and released two new versions during the first two weeks of its use, focused on improved annotation loading and rendering speed. For the final two weeks of WebAnn use, I introduced a version supporting direct editing of annotations (rather than having to delete and re-create a note manually to change it).

During the quarter the online discussion forums were archived and I surveyed the students weekly about their experience. An initial survey collected background information, and a final survey asked about their overall experience with in-class and online discussion. Two colleagues, Jonathan Grudin and Cathy Marshall, interviewed eight students at the end of the quarter. Cathy Marshall and I have published another study that explores the relationship between the student's comments on paper and those in WebAnn in [MB02].

3.3 Field Study Results

Online discussions easily exceeded the required participation level. WebAnn activity differed substantially from EPost discussion. One significant observation is that all students printed out

Table 3.1: Student participation using EPost and WebAnn.

Method	Number of Papers	Number of Messages	Messages Per Paper	Average Messages Per Author Per Paper	Average Replies Per Author Per Paper	Average Character Contribution Per Author Per Paper
EPost	13	299	23	2.23	1.15	2485
WebAnn	12	470	39.2	4.71	1.58	4401

and read paper versions of papers. This removes important potential advantages of WebAnn and affects its use and reception. Despite this, WebAnn was used more than EPost, and the pattern of use provides guidance to the design and use of anchored annotation systems.

3.3.1 More participation using WebAnn

As expected, the ability to anchor comments precisely led to more comments in WebAnn weeks, given that students would be likely to combine comments in EPost threads. Table 3.1 shows key per-author participation statistics using the systems. Using WebAnn, there was an average of 39 comments per paper, with EPost 23. Several students also remarked on the increase in survey responses. For example, "*I made more comments with WebAnn. With EPost I was disinclined to make minor or very contextual points—the kind it is easy to do with WebAnn. I also think I wrote more replies in WebAnn because it was easy to do so (and easy to see the context).*"

Students also replied more when using WebAnn. With EPost, the average number of reply messages per author in each discussion board was 1.15 (most students made only the one required response per discussion forum). Using WebAnn, authors averaged 1.58 replies per paper. A paired-samples t-test showed the averages were significantly different with $p < 0.03$[1]. In fact, 8 out of the 11 students made more replies using WebAnn than using EPost. One student averaged a remarkable 3.33 replies per paper using WebAnn, and only 1.5 with EPost.

While I thought WebAnn annotations might be short, since students would not need to reconstruct the context for each comment, I was not sure how the total participation per student would vary since students would probably make more posts using WebAnn. I found that students wrote almost twice as much with WebAnn. Each student wrote an average of 4,401 characters per paper using WebAnn, compared to 2,485 characters per paper using EPost. These are

[1] The p-value essentially measures the probability that the t-test might be returning incorrect information. Therefore smaller p-values indicate higher confidence in the results of the t-test. A p value less than 0.05 is generally considered significant [EM].

significantly different based on a paired t-test ($p < 0.001$). Although increased participation in discussion does not necessarily imply enhanced learning, grades in this class included participation and there were no exams, so increased participation was considered a positive outcome.

3.3.2 General vs. Specific Discussion

The two systems support very different types of discussions. EPost discussion boards are completely separated from the paper being discussed, while WebAnn discussions are anchored directly to specific parts of the paper. As expected, these differences affected the type of online discussion that occurred in the two systems, and this was reflected in student survey responses. For instance, one student observed that with WebAnn it was "*More difficult to make high level comments about [the] paper, [and] discussions usually focused on sentences or paragraphs …*" and another noted that with EPost "*It's definitely harder to make pointed comments about these papers.*" In response to the final survey, one student said "*I think the comments were at a higher level in the E-Post system and more general opinions were presented*" and another said "*…the comments were more specific and numerous [with WebAnn]. I think this is because I could transfer notes I'd made on paper and section of text I'd highlighted directly to the annotation software*."

Although the preference for more general or more specific discussions varied, many students observed that WebAnn led to more thoughtful, involved discussions. For instance, one student observed "*More scattered, but more insightful comments in WebAnn,*" while another saw "*More involved discussion—more back and forth,*" and a third said "*I think the quality of annotations and online discussion [with WebAnn] was better than with E-Post*."

3.3.3 Student Preferences

Table 3.2 shows median student ratings on several key questions from the weekly survey. The ratings are on a 6 point Likert scale where 1 is "Strongly Disagree" and 6 is "Strongly Agree." Table 3.3 shows the median student ratings for key questions from the final survey, also on a 6 point scale where 1 is "Low" and 6 "High," except for question 6 where 1 is "Disagree" and 6 "Agree." Only the ratings for amount of time spent on software in Table 3.3 (concerning software trouble) were significantly different between the two systems based on a paired sign test ($p < 0.02$).

Table 3.2: Median student ratings on questions from the weekly surveys.
(1 is Strongly Disagree, 6 is Strongly Agree). Numbers of students who responded to a question are in ()'s. N/A* There was not yet a basis for comparison. +Only 4 students participated in the online discussion this week, which may have impacted the ratings.

Week	1	2	3	4	5	6	7	8	9
System	EPost	EPost	EPost	WebAnn	WebAnn	EPost	EPost	WebAnn	WebAnn
1. Discussions in class were valuable	5 (11)	5 (11)	5 (11)	5 (10)	5 (10)	5 (10)	5 (11)	4 (9)	5 (6)
2. Online Discussions outside of class were valuable	4.5 (10)	5 (11)	4 (11)	4 (10)	5 (9)	4 (11)	4 (11)	4 (10)	3 (5)+
3. The review method [software] was beneficial to my learning	4 (9)	5 (11)	4 (11)	3 (10)	4 (9)	4 (11)	3 (11)	4 (10)	3 (5)+
4. I prefer this reading review method	N/A*	N/A*	N/A*	2 (8)	3 (8)	4 (11)	4 (11)	3 (10)	5 (6)

Table 3.3: Median student ratings on questions from the final survey. For the first 5 questions 1 is Low, 6 is High. For question 6, 1 is Disagree, 6 is Agree. *The only significant difference is for question 5.

System	1. "The quality of the online discussion was"	2. " My satisfaction with this method of online discussion"	3. "The quality of the in-class discussion was"	4. "My satisfaction with the in-class discussion"	5. "The amount of time I spent on problems with the software"*	6. "Overall I prefer this method"
EPost	4	5	4	4	1	4
WebAnn	4	4	5	4	4	4

Value of Discussion

In general, students gave high ratings to the value of discussions both in class and online throughout the course, and there is little quantitative distinction in the value of discussion supported by the two systems. In surveys and interviews students commented more specifically on the value of online discussion. Some examples: *"[online discussion] made the discussion [in class] a lot more interesting," "since they [online comments] are written, they are definitely more composed," "[through online discussion I] got to know people's opinions, the people who aren't as vocal in class,…having the online discussion encouraged everyone to participate in-*

class as well," and *"there were a couple of people who often dominated the class conversation, but they wouldn't dominate the online discussion because everyone got a chance to talk."*

Finally, two interesting ratings from Table 3.2 are the 3's given to WebAnn in week 9 for questions 2 and 3. In this week, most students used their paper skips and only 4 students participated in the online discussion. This affected satisfaction with online discussion. One student commented: *"It's really boring when no one says anything."*

System Preference

Based on their subjective ratings, students preferred EPost slightly overall. However, with only 11 students the data are inconclusive, and individual student preferences varied. Table 3.2, question 4, illustrates that WebAnn preference ratings started low and rose over time. This may reflect the improved versions of WebAnn that were introduced. Table 3.3, question 6 shows that on the final survey both EPost and WebAnn received the same median rating, despite having encountered more technical and access problems with the WebAnn software. However, comparing a particular student's ratings of the two systems, for example, if the student rated EPost a 4 and WebAnn a 3, I obtained more information: 5 students preferred EPost, 3 preferred WebAnn, and 3 had no preference. In this regard it is useful to keep in mind that by reading printed copies of papers, students lost the advantages of annotating and seeing comments of others as they first read a paper and were thus reacting primarily to the discussion features.

Comments on the final survey indicated that preferences for a particular method were based on a range of factors, including access and perceived quality and granularity of the discussions. Favoring the EPost system, one student said, "*I didn't have [a]* preference. *[The] only issue was that I could use EPost at home*." Another expressed a "*slight preference for EPost because it allowed for more articulation of complete ideas/thoughts.*" A third observed that "*It was easier to understand other student's opinions by reading all their comments in a single message. Also, I think the comments were at a higher level in the EPost system and more general opinions were presented.*"

In favor of WebAnn, one student said "*I prefer WebAnn (later versions) over EPost. I think the quality of annotations and online discussion was better than with EPost. WebAnn allowed us to comment on specific portions of text, which was nice…*" and another observed that WebAnn was useful because students "*can comment on particular parts of the paper easily…*"

3.4 Issues and Options

Based on the survey ratings and comments, most students felt that online discussion helped the live discussion start quickly and gave it focus. The online discussion space also provided an outlet for students who said less in class, and overall increased class participation. In addition to these successes, though, I learned a number of important lessons about incorporating online discussion into a class. First I describe some of the major issues encountered and then discuss potential changes in technology and process that would help address them.

Student and Instructor Workload: In general, incorporating online discussion into the class created more work for the students and instructor by requiring everyone to keep track of and participate in the online discussion at some level.

Although some students felt WebAnn led to more thoughtful online discussions, and clearly it resulted in more extensive discussions, WebAnn required students to do more work to post their reviews. As noted above, although all papers for WebAnn discussion were made available in HTML format, all students printed them out to read. To enter WebAnn comments they had to go back and annotate the papers online. One student commented: *"I found WebAnn much more time-consuming to use, perhaps because I prefer to read on paper, and then had to go back through to do the annotations."*

Should professors or teaching assistants participate online beyond reading or leave it as a space for students? In this class, both types of interaction occurred and each has advantages and disadvantages. In the study the instructor and TA generally participated very little (3 posts in EPost, 5 in WebAnn) beyond reading all messages. This was less work for the instructors and allowed students to take the lead, but meant questions could go unanswered or issues left undeveloped. One guest lecturer in a WebAnn week addressed most questions and issues students raised online, an approach used successfully for design reviews of student assignments using CoWeb [GRK00]. Students seemed to appreciate the responses. One advantage of having the instructor, the TA, or even an expert in the field respond to students is that it may encourage students to go back and read through the comments. On the other hand, students may avoid controversial points if an author or known expert will reading them. In this case, one student deleted or edited one or more of his comments when he realized the paper author (the guest lecturer) might read them.

Online and in-class discussion: Before the study, I saw the online discussion as a complement to the in-class discussion, leading to a more engaging classroom discussion focused on issues raised online. Each week after student reviews and comments were due, the instructor and I, as the TA, read all the comments and replies. I also created a list of interesting issues and comments to start in-class discussion if necessary. Some students found this helpful, but others commented:*"…[it would] be more effective if there were some way to better integrate online discussion with in-class discussion,"* and *"[class time] was redundant."* Smoothly integrating the two was more challenging than expected. In a sense, in-class and online discussion competed with one another.

Integrating online and in-class discussions was complicated by the timing of the online discussion and the differing amounts of participation in the online discussion (both posting and reading comments). Because weekly reviews were due Tuesday and replies were due Wednesday just prior to class, the time for students to read through responses was limited, a problem exacerbated by the fact that some could access the system only from one of home or work. If a reply was added shortly before class it was unlikely that many students would read it. This negatively impacted the in-class discussion. A student who made a long or complicated reply online might not want to repeat it in class, even when asked to by the professor. As a result, interesting replies were not always picked up in class.

Differences in time commitments and interest levels led to varying student participation online, which in some cases took a fair amount of time. Students who participated online often seemed uninterested in continuing that discussion for those who had not participated online. In one instance, following a spirited WebAnn discussion among six students, the professor tried to bring up the issue in class for further discussion. One student said there had already been a *"pretty good discussion on [the] board."* This comment, along with others like it, ended the classroom discussion on the topic. The students who had participated online saw no need to discuss the topic further.

Global and specific comments: With WebAnn it was easy to make or understand focused comments, but awkward to make general notes about large sections or even long paragraphs. Conversely, EPost required considerable context to comment on a particular point. Each tool readily supported one type of comment. The ideal tool would facilitate comments at multiple levels. In the next section I discuss improvements to WebAnn to support more general comments.

Allocating Display Space: Online discussion systems face a tradeoff between focusing attention on threaded comments or on the document being discussed. CoNote [DH95a, DH95b] places links to content in threaded discussions; CaMILE [GT00] places links to discussion in content. WebAnn splits the focus between comments and document. (With EPost, only comments are viewed.) In the interviews, some students noted this tradeoff and suggested that more space be devoted to comment threads. When reading the document and making comments the document might be the focus, as was the default WebAnn setting, but these students did most reading on paper. When reviewing others' comments and replying, the comments might better be the focus. In fact, students could adjust the size of the frames in WebAnn, but did not discover this.

Discussion overload: When examining student participation in the online discussion, I found students contributed much more during the WebAnn weeks. While this suggests that anchored discussions in WebAnn encourage students to participate more, some students remarked that the number of comments and replies was overwhelming. Clearly, this could become an even larger problem with bigger classes. EPost discussions could also be problematic with large numbers of participants. This tradeoff between encouraging student participation in online discussions and keeping online discussions a "manageable size" has also been noted by Guzdial & Turns [GT00]. In Section 3.5.2, I outline process changes that could help avoid discussion overload.

Convenient universal access: As other studies have found, convenient access is critically important [HGT97]. I was initially concerned with making sure all students had *some* access to WebAnn, which runs only on Windows 2000. However, it turned out that *where* students had access was also important. With EPost, 9 of the 11 students had access both at school and home. Using WebAnn, although all students had access either at home or school, only two students had access in both places. With access in only one location, students were limited in when and where they could do their reviews and participate in the online discussion. In the interviews, several students commented that having limited access to WebAnn was frustrating.

3.4.1 Improvements to WebAnn

The subjective ratings and comments suggest the majority of students had a small preference for EPost, even though they contributed more using WebAnn. Factors including access, workload, software use, and different types of discussion seem to influence this preference. This section proposes technical improvements to WebAnn, or any other online discussion system, that might address issues raised by the field study.

Access: Making access as universal as possible by supporting more operating systems and browsers would enable students to review the discussion more often from more locations, and might improve participation in making and reading comments. Adding an offline mode would also help students with slower internet connections.

A more sophisticated solution might allow comments and replies to be sent through email in addition to being added as web page annotations. When students did not have access to the annotation system, they would still receive annotations in email. Replying to the email would add their response to the online discussion. Although this raises the question of how much context to include in the notification, the MRAS video annotation system found this approach successful in a number of studies without including context from the video in the email notification [BGG+01].

Filtering and Notification: Several students suggested adding filtering options, which exist in EPost, to WebAnn: including author-based filtering, and identifying notes and replies that are new. Mechanisms that assist in quickly finding replies to a person's comments and highlight potentially interesting discussions based on collaborative filtering, perhaps by allowing students to rate each others' posts, could further reduce discussion overhead and student workload, making it easier to keep up with the online discussion.

Although EPost can notify students of the presence of new posts, only 3 students subscribed to this for one week of the study, suggesting a need for improvement. Notifications could summarize the comments made that day, rather than just alerting a student to the fact that comments were made. This could provide a sense of the ongoing discussion and encourage checking online for the full comments. Notification messages could include clickable links to take a student directly to the online discussion [BGG+01]. Finally, an optional feature that notifies a note's author when someone replies could encourage more back and forth discussions.

Advanced notifications features might allow students to follow the online discussion more easily. This could support easier integration of online and in-class discussions, reduce student and instructor workload, and help students deal more effectively with discussions containing a large number of comments.

Supporting General Comments: Students wanted to add comments at many different levels, from general comments about an entire paper to specific comments on a particular issue. To better support online discussions, WebAnn needs a mechanism for easily commenting on larger document units, including paragraphs, sections, and the entire paper. Softening the display of anchors in the web page, perhaps with vertical lines in the margin instead of outlining the text, might make users more willing to overlap comments. More ambitiously, mechanisms for clearly supporting comments at every level of the document could be provided, perhaps through menu items that specify “comment on this document,” “comment on this section,” and so forth.

Allocating Screen Space: As noted above, the interface should clearly indicate that the annotation and document frames can be resized to accommodate a focus on threaded comments or a focus on the content.

3.4.2 Process Changes

Along with technological improvements, careful consideration of the process of use might smooth the experience of combining online and in-class discussions.

More time for reviews: It would have benefited students to have all online discussion before the first class of the week, and to have the review and replies due earlier to provide more time before class discussion for reading and responding. Scheduling class meetings for the end of the week could also address weekend access issues.

Summarize online discussion in-class: A short summary of the online discussion at the beginning of class might help cope with the different levels of online participation and frame an in-class discussion that builds on the online experience. Explicitly acknowledging students who took part in the online discussion could encourage other students to participate.

Consider instructor role: The pros and cons of active instructor participation online were noted above. On the whole, if instructors join the discussion late, it can provide an incentive for students to contribute to and review discussions. If online participation were not required—and some students felt mandatory replying to others’ comments was artificial—this might be essential to motivate discussion.

Adjust the number of papers discussed online or in class: Using the online discussion for addressing fewer papers in more depth rather than for all the papers would reduce the amount of work. Dividing the papers into those discussed only online and those that are dealt with only in class would also reduce workload. Alternatively, classroom time could be used more for other activities, such as demos and discussing student projects.

Reduce the number of students participating at any one time: To combat discussion overload, reduce workload, and help integrate online and in-class discussion, the number of messages students produce or read could be reduced. Students could be asked to comment on fewer papers, or participation could be made optional for large classes. Alternatively, students could be divided up into discussion groups, and each discussion group could briefly summarize in class what was discussed online, greatly reducing the number of messages a student must read.

Reduce assignments: Another approach for reducing student and instructor workload is to limit the number of assignments the students have, or to more dramatically reduce the time that is spent in-class. The broader issue is to consider what classes are best served by the technology. Possibly classes with less reading (which students may be more likely to do online) or classes taught using distance learning (that have no in-class discussion) would better exploit the value of anchored discussions. Or perhaps discussions could revolve around assignments and projects, which might have shorter blocks of text, as in Guzdial & Turns [GT00].

3.5 Conclusion

Online anchored discussions hold great potential for extending in-class discussion beyond the classroom door. In the study, online discussions allowed the less vocal students to contribute equally and made in-class discussions more interesting, but integrating the online and in-class discussions was challenging. Rather than serving as a starting point for in-class discussions, the online discussions often competed with the classroom discussion. Students who participated frequently online seemed uninterested in addressing the same issues in-class with the rest of the students.

Because students in this class uniformly printed and read assignments on paper, many potential advantages in annotating in context were lost. Nevertheless, WebAnn led to more discussion, even while requiring the greater effort of a second pass to add comments. With improvements in technology, including better annotation systems, universal access, and displays that facilitate reading online, as well as appropriate process modifications, anchored discussions are an exciting avenue for distributed education and a viable tool to supplement classroom instruction.

Chapter 4

Notification for Annotations

Shared annotations on digital documents are an attractive means of asynchronous discussion. However, as the students participating in the field study of online discussion observed, without notifications, people must continually revisit a document to see the latest comments and participate in the discussion. One way to address this problem is to integrate a notification mechanism into a shared annotation system. When a new annotation is added, interested parties are notified (e.g., by email) and can revisit a document to read more, add a reply, or contribute new comments. Many systems (e.g., [App99, Cri, Int, LFK88, Liv, MOWD]) have used this approach with varying degrees of success.

Although notification mechanisms in shared annotations systems are common, there has been little study of user needs and little exploration of design tradeoffs. To better understand notifications and their role in discussion I studied a large product development team at Microsoft Corporation that reviews internal software specification documents using Microsoft Office Web Discussions [MOWD], a shared annotation system with a closely-integrated email notification mechanism.

First, I gathered data about the team's experience using Office Web Discussions, in particular its email notifications. Then, informed by that data, I designed improvements to the Office Web Discussions notification mechanism including more detailed email notifications and notifications using peripheral awareness. I deployed the enhanced notifications in a field study over three months and found that providing more information in notification messages, supporting multiple communication channels through which notifications can be received, and allowing customization of notification messages, were particularly important. Overall awareness of annotation activity on software specifications increased with the notification enhancements.

Section 4.1 describes Microsoft Office Web Discussions and its standard notifications. Then Section 4.2 discusses the current usage of Office Web Discussions. In Section 4.3 I describe the enhanced notifications and Section 4.4 discusses the field study and results. Issues raised by the study that developers should consider when designing a notification system are outlined in Section 4.5. Section 4.6 concludes with directions for future research.

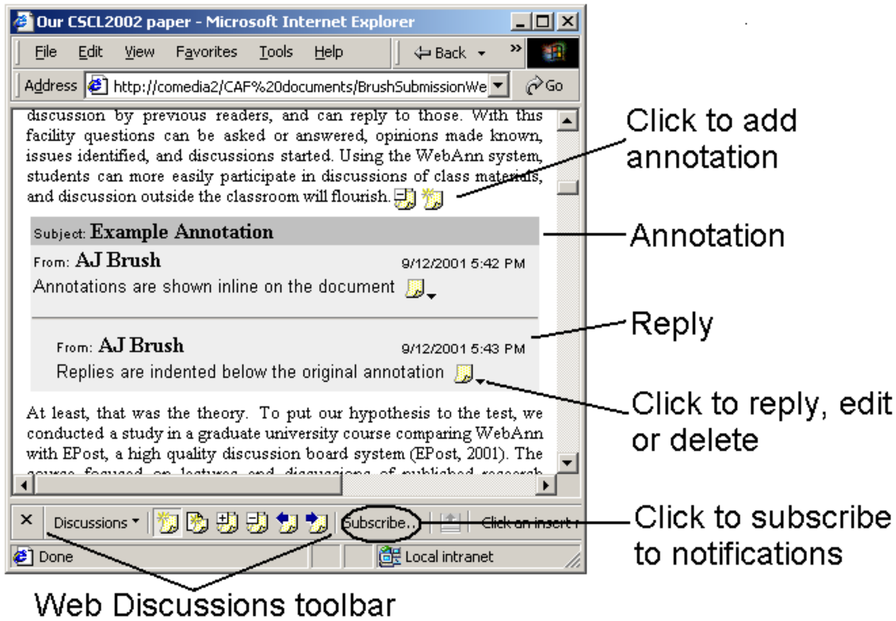


Figure 4.1: A web page annotated with Microsoft Office Web Discussions. Annotations can be made at the end of every paragraph and are displayed inline on the page.

4.1 Microsoft Office Web Discussions

The Microsoft Office Web Discussions annotation system [MOWD] is part of Microsoft Office and allows annotation at the end of paragraphs on a web page. An annotated web page is shown in Figure 4.1. The annotations are displayed inline in the page and replies are indented. Annotations are created by clicking a button in the Web Discussions toolbar at the bottom of the browser window. This displays icons on the page where annotations can be added. Clicking on an icon brings up a dialog box where a user can type in an annotation. Users reply to an annotation by clicking on the icon at the end of an annotation. Using the Web Discussions toolbar users can also expand or collapse all the annotations in a document, and navigate forward and backward through the annotations in the context of the document.

Using Web Discussions does not modify the original HTML version of the web page. Annotations made using Web Discussions are stored on a separate annotation server that resides on an organization's intranet. When a user with appropriate server permissions browses to a web page with Web Discussions turned on, the annotations for that page are downloaded. Web Discussions then inserts each annotation into the local version of the web page at the appropriate

The following change(s) happened to the document
http://server/Notify.htm

Event:	Discussion items were inserted or modified in the document.
By:	colinb
Time:	9/12/2001 3:20:24 PM

Click here to stop receiving this notification.

Figure 4.2: Web Discussions current email notifications. The notifications contain the name of document, author of the comment, and time the comment was made.

paragraph using anchoring information stored in the annotation. If the document has changed and the matching paragraph is not found, the annotation is orphaned and displayed in a separate window at the bottom of the browser.

4.1.1 Notification Mechanism

For this research I focused on notifications in Web Discussions. The system includes a simple default notification mechanism where by clicking on the “subscribe” button in the Web Discussions toolbar users can receive email when annotations on the document are made or modified. Users can have email sent for each change or select to receive a daily or weekly summary. An example of the notification email is shown in Figure 4.2

Cadiz et al. [CGG00] found several significant drawbacks to this mechanism. For instance, it does not identify which annotations have been added or make it easy to follow-up on the discussion. Subscribers cannot control notifications based on who made annotations (e.g., someone replying to an annotation made by the subscriber, or the document author). And it does not inform annotators as to who will be notified automatically of annotations.

4.2 Usage of Web Discussions Notifications

To better understand current practice, I surveyed a subset of users in a large software product development group about their experience with Office Web Discussions in particular their experience with the default notification mechanism. A colleague, David Bargeron, also analyzed annotation usage logs from a six-month period.

The product group uses Web Discussions to comment on software feature specification documents, or “specs.” Program managers are responsible for writing specifications that are subsequently commented on by the developers and testers who will implement and test the features. Others, including documentation and usability specialists, also comment on the specifications. The focus of this study was on notifications in Web Discussions. An earlier study by Cadiz et al. [CGG00] reports in more depth about general use of Office Web Discussions for specification review.

4.2.1 Usage Survey

To understand how specifications are reviewed, awareness of comments, and satisfaction with the default Web Discussions email notifications, I surveyed a subset of current Web Discussions users. I contacted approximately 250 people and received 98 responses to the online survey from testers (38%), program managers (29%), developers (16%), and others (17%) including documentation and usability specialists.

Reviewing Specifications

The primary methods respondents reported using to comment on specifications are email (84%), Web Discussions (81%), face to face at specification review meetings (80%) and face to face with specification authors (63%). As Table 4.1 shows, participants are most likely to use Web Discussions for comments if they do not need a response before the next specification review meeting (there are usually two such meetings per spec) or for a couple of days.

Awareness

Survey respondents agreed it was important to stay aware of comments on specifications for

Table 4.1: Methods used to comment when timely responses are needed (98 respondents).

	Immediate Response	Within a few hours	Within a day or two	Until next review	Total
Face-to-face	53	8	1	0	62
Use email	22	73	43	13	151
Use Web Discussion	5	10	45	55	115
At spec review meeting	6	0	1	21	28
Other	12	7	8	9	36

features they are responsible for and those they are interested in. (Median response was “Agree.” All questions were on a 5-point scale from “Strongly Disagree” and “Strongly Agree.”) When asked if it was easy to stay aware of comments for specifications they were working on, the median response was “Strongly Agree.” In contrast, the ease of following comments on specifications they were interested in received a median response of “Neutral.”

Existing Notifications

Eighty-four respondents (86%) had used Web Discussions for specification reviews. The median was “Agree” that using Web Discussions for specification reviews works well. Forty-three respondents had subscribed to the existing email notifications. They typically subscribe to notifications for specifications they are working on but did not author (84%), and they are less likely to subscribe to specifications they author (44%) or review (40%). Satisfaction with email notifications was quite low: The median was “Disagree” for “I am satisfied with the current email notifications for Web Discussions.”

I asked respondents to comment on what they liked and disliked about email notifications. Most positive comments stressed that notifications saved them from repeatedly checking the document for changes and a few commented that they appreciated choosing when to be notified. Many negative comments focused on the lack of helpful content in the notifications and on email overload.

4.2.2 Usage Analysis

For additional insight into the use of annotations, David Bargeron analyzed Web Discussions usage logs for a six-month period from February through August of 2001. During this time, 466 users made 13,780 annotations on 851 documents. Each user created an average of 29.6 annotations on 4.9 documents. Each document had an average of 16.2 annotations made on it and 1.35 subscriptions for email notification of Web Discussions events (adding comments, deleting comments, modifying comments, “resolving” a comment, and so on).

Users and Notifications

Users of Web Discussions notifications fall into three groups: 348 made annotations but did not subscribe to notifications, 118 annotated and subscribed, and 48 subscribed but did not annotate. Thus, 68% of users did not receive notifications.

The 118 users who both annotated documents and subscribed to notifications signed up for a total of 562 notifications subscriptions on 415 different documents, an average of 4.76 subscriptions per user. 234 of these 415 documents were annotated. Daily subscriptions were preferred. 328 (58%) of the 562 subscriptions were for daily notifications, 224 (40%) were for immediate notifications and 10 (2%) were for weekly notifications.

The 48 users who received notifications but did not annotate the document averaged 4.9 subscriptions. Collectively they held 237 subscriptions to 200 documents. Daily subscriptions were again the most popular, comprising 138 (58%) of the 237 subscriptions with 98 (41%) immediate subscriptions and 1 weekly subscription. In the Cadiz et al. study [CGG00] the preference for daily notifications (the default) was even more apparent. Overall 70% of the subscriptions in that study were for daily notifications.

4.3 Notification Enhancements

Inspired by the study of current usage and previous research I implemented enhancements to Web Discussions notifications and performed a field study of their use. I explored the design tradeoffs using two methods: improving existing email notifications and implementing notifications using peripheral awareness.

This is an automatic notification. More information...
Click here to update your notification settings.

The changes that just occurred are:

On http://server/Notify.htm

duncanbb added a comment on **9/12/2001 9:47 AM**
test annotation
This is a test

> **colinb** added a reply to a comment by duncanbb on **9/12/2001 3:20 PM**
> **RE: test annotation**
> This is the text of an example annotation.

Click to update your notification settings.

Figure 4.3: Enhanced email notifications. The new notifications include the annotation's content and indicate if it is a reply. Clicking on the subject line of the annotation opens the document to that annotation.

4.3.1 Detailed Email Notification

To provide more information in notification messages I implemented an email notification service for Web Discussions. As shown in Figure 4.3, the notifications include the content of new annotations and indicate when an annotation is a reply to an existing annotation. During the field study described in the next section, I added a direct hyperlink from a comment in email to its location in the document to allow users to easily follow-up on annotation activity.

Using a simple web form, users select to have the email notifications about new annotations on a document delivered immediately, daily, or weekly. In addition to these standard options, users who sign up for daily or weekly emails can ask for immediate notification messages to be sent for replies to their annotations. To reduce the amount of notification mail a user receives, users are not notified about annotations they create. Also, if a user subscribes for daily or weekly notifications on multiple documents, all notifications about new annotations on those documents are sent in one email message.

The email notifications are implemented as a Microsoft Windows service. The service maintains a database containing the notification subscriptions on the specification documents. The service then polls the Microsoft Office Web Discussions annotation database to determine which documents have new annotations. If any users are subscribed for immediate notifications on a document with new annotations, the service composes and sends the appropriate email notification to that user. Daily summary emails are sent in the early morning for annotations made on documents the previous day, and weekly summary mails are sent Monday morning for annotations made the previous week.

4.3.2 Peripheral Notifications using Sideshow

Email is commonly used for notification; however it seems heavyweight for maintaining continuous awareness. Constantly tracking the annotations on a document could result in many messages. To explore another channel for notifications I implemented notifications using the Sideshow [CVJ01] peripheral awareness system.

The Sideshow system uses a small amount of screen real estate for its peripheral awareness sidebar. The Sideshow sidebar sits on the side of the screen and contains items called *tickets*. Each ticket displays information from a particular source. Examples of Sideshow tickets include an inbox ticket that displays information about the user’s email inbox, and a “traffic ticket” that

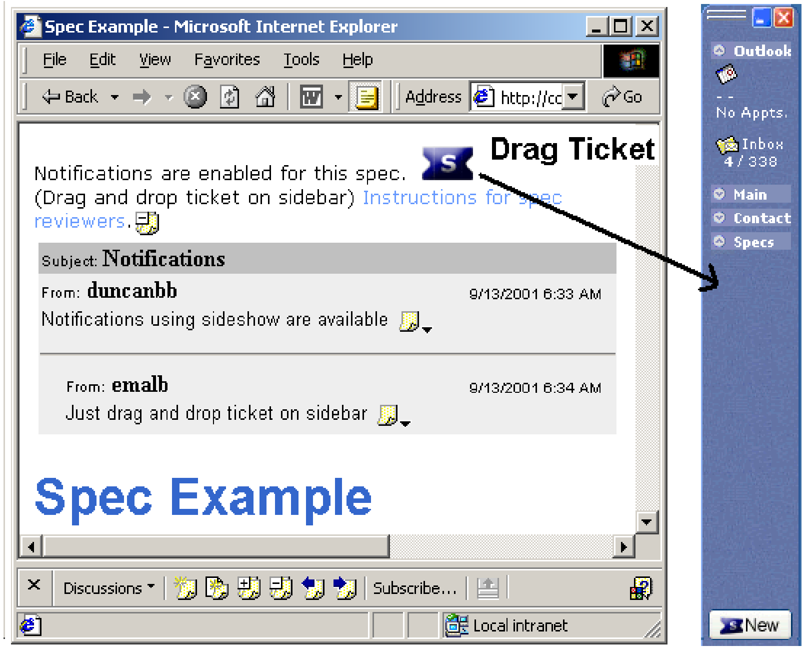


Figure 4.4: Subscribing to Sideshow notifications. To subscribe to notifications for a specification document the user drags the ticket icon from the document to the Sideshow sidebar.

monitors traffic congestion from local traffic cameras. The user can often customize the tickets displayed on the sidebar. Sideshow also supports designing new tickets.

For annotation notifications, I implemented a Web Discussions Sideshow ticket that polls the Web Discussions annotation database and displays general information about the number and contents of annotations on a particular document. In order to cope with the large number of specification documents involved in the field study described in Section 4.4, I also created a web form for customizing the basic Web Discussions ticket to create a ticket for a particular specification document. This allowed program mangers to easily create tickets for their own specifications.

Figure 4.4 shows a document with a Sideshow ticket on it. To subscribe to annotation notifications, a user simply drags the ticket from the document and drops it on the sidebar. Thereafter the user can see current information about annotations made on the document by glancing at the ticket on the sidebar.

The ticket, shown in Figure 4.5, displays the total number of annotations and annotations that are new today. By default, annotations made on the same day are considered “new,” but the user can easily customize this either to annotations made since the current time or all annotations

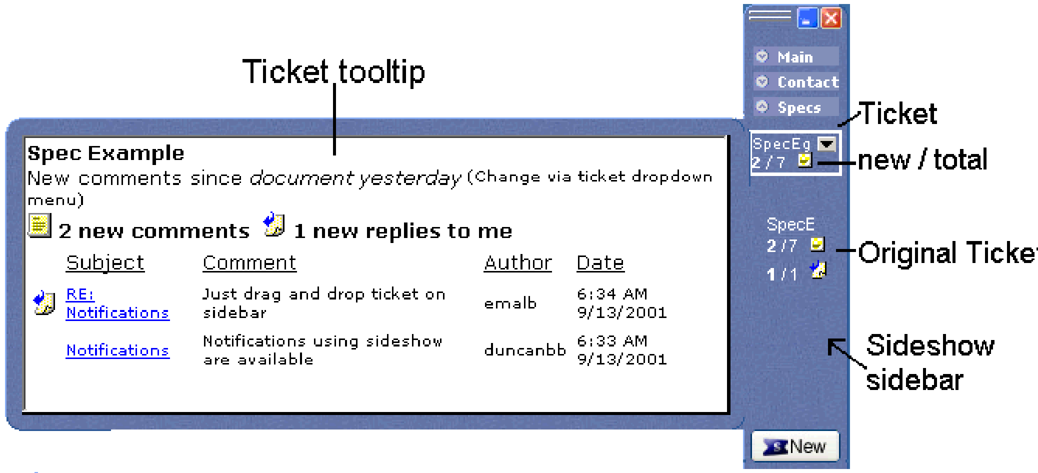


Figure 4.5: Web Discussions Sideshow Ticket. The ticket displays number of annotations and replies. The tooltip window shows details when a user mouses over the ticket.

made on the document. The original ticket used in the first half of the field study had a second line that displayed the total number of replies to comments made by the user running Sideshow, and the number of new replies.

When the user mouses over a ticket, the tooltip window shows more detail about new annotations, including the author, creation time, and contents. In the second half of the field study, the tooltip also included direct hyperlinks that opened the document directly to new annotations.

4.4 Field Study of Notification Enhancements

To study the effectiveness of the enhanced notifications, I deployed them to a subset of users in the product group for use in their specification review process from August to November of 2001.

4.4.1 Study Methodology

In mid-August 2001 I approached program managers in three groups using Web Discussions and asked them to identify specification documents that would be reviewed soon. Program managers, developers, testers and others, including documentation and usability specialists, review specifications over a period of a few weeks to a few months. For each specification at least two meetings are also scheduled where people meet face-to-face to discuss issues with the specification and go over the Web Discussions comments made on it.

I added Sideshow tickets to specifications identified by the program managers and encouraged people reviewing the documents to try the detailed notifications. I also contacted everyone who had previously signed up for the default Web Discussions notifications and asked them to try the notifications.

Integrating the notification mechanisms did not alter the specification review process for the teams that tried it. They continued to use Web Discussions for commenting on their specifications, and could still elect to use the default Web Discussions notifications, but they had the added option of using the more detailed notifications instead.

Before trying the notifications, participants filled out the survey of current usage discussed previously. Some users filled out the current usage survey but did not subscribe to the enhanced notifications. In general these users either did not need notifications to stay aware of specifications, or currently had no specifications they needed to stay aware of.

On September 10th 2001, I surveyed current users for feedback and interviewed six users in depth. At this time, 39 people were subscribed to the enhanced notifications: 22 of them were using Sideshow tickets, 10 were subscribed to the email notifications, and 7 people were using both. This feedback survey received 22 responses, primarily from program managers (41%), and testers (36%).

I then introduced some improvements based on the feedback and recruited additional participants. On November 26th 2001, I again surveyed current users and conducted two additional interviews. By this point, 90 people had used the notifications, 60 had Sideshow tickets, 18 were subscribed to email and 12 were using both. The final survey received 31 responses from program managers (39%), developers (23%), testers (19%), and others (19%). Twelve people answered both the September and November surveys.

4.4.2. General Experience

188 people made 4,221 annotations on 98 documents involved the field study. 57 (30%) of the 188 people adopted the enhanced notifications exclusively, 30 (16%) adopted Web Discussions notifications exclusively, and 16 (9%) used both systems. 85 (45%) annotators did not subscribe to any notifications. Each annotator created an average of 22.3 annotations on an average of 2.4 documents, and each document had on average 43 annotations.

Surveys and interview data indicate field study participants were positive about the new notifications. Participants particularly appreciated the fact that the enhanced notifications allowed them to stay aware of annotation activity without opening a spec. One participant said "*[Sideshow] kept me up to date about what discussions were occurring about my specs,"* while another said *"[the email notifications] keep me up to date."*

The two primary uses of the notifications during specification review were active monitoring of annotations and more casual tracking of annotation activity. Active monitoring was primarily done using Sideshow tickets. One program manager interviewed watched until the ticket showed five or six comments, then dealt with them all at once.

Participants also used both Sideshow and email to track annotations passively. One manager used Sideshow to notice when not enough comments were being made (previously he did the same tracking by opening the spec). Another person kept the email notifications around until he had time to visit the spec.

Survey respondents felt using the notifications affected their behavior. On both surveys, when asked about their awareness of online comments on specifications where they had the notifications, the median response was that they were "more" aware. When asked about how fast they responded to other comments, the median response was that they responded "faster."

Respondents felt there was no change in the amount of online discussion, or in the number of comments they made, or the speed with which other people responded to comments on specifications with the enhanced notifications. This is perhaps understandable since not everyone involved with a particular specification subscribed to the notifications.

Respondents also answered more specific questions for the enhanced notifications types that they tried.

4.4.4 Email Notifications

Nine of the respondents (40%) on the September survey and eleven of the respondents (35%) on the November survey were subscribed to email notifications. On both surveys, participants signed up for email notifications "Agreed" that enhanced email notifications were useful and they provided enough information about new comments. (All questions were on a 5 pt. scale from "Strongly Disagree" to "Strongly Agree.")

Table 4.2: The most valuable information in email.

Information in Email	1st	2nd	3rd	Total
Content of new comments	7	1	1	9
Author of the comment	2	2	3	7
Hyperlinks to open spec to comment	1	4		5
Context from spec around comment		2	2	4
Whether comment was reply			3	3
Subject line of the comment		1	1	2

Subscriptions

30 people made 131 subscriptions to the enhanced email notifications on 122 different documents. Similar to the usage of the default Web Discussions notifications, there are more subscriptions (86, 65%) for daily notifications compared to immediate (44, 34%) and weekly (1, 1%) notifications. I found it somewhat surprising that only nine of the daily subscriptions (10%) asked for immediate emails for replies to comments made by the subscriber. In Section 4.5.5, I discuss how interview data suggests the annotations in Web Discussions are used more to track major issues in the specifications rather than have discussions. This type of use may make special reply notifications less important for this task.

Design Improvements

On the September survey participants “Agreed” that direct hyperlinks from the comment to its location in the specification would also be useful for email notifications. With assistance from David Bargeron, I added hyperlinks to the email notifications, so that clicking on an annotation’s subject line opened the specification directly to that annotation. On the final survey in November I asked participants to rate the three most valuable pieces of information in the email notifications. The ratings, shown in Table 4.2, highlight the value respondents placed on the hyperlinks.

In the final survey I asked participants about whether it would be valuable to have context from the specification around the comment in the notifications, even though this feature was not currently implemented. Adding context to notifications is a non-trivial task and I wanted to gauge whether users considered it important before attempting it. The survey data and interviews indicated that including context information in email would be an interesting direction for future

Table 4.3: The most valuable information in the ticket tooltips.

Information on Ticket Tooltip	1st	2nd	3rd	Total
Content of new comments	5	4	2	11
Num. of comments & replies to me	6	2	2	10
Subject line of new comments	5	4		9
Hyperlinks to open spec to comment	2	3	4	9
Author of new comments		2	7	9
Date of comment		2	1	3
Context from spec around comment		1	1	2
Reply to me icon			1	1

work. Other design suggestions included a clearer visual distinction between replies and new annotations, and including the text of annotations that were replied to.

4.4.3 Notifications Using Sideshow Tickets

Sixteen of the respondents (73%) on the September survey and twenty of the respondents (65%) on the November survey had used sideshow tickets, typically for 3-5 specifications.

Ease of Use

On both surveys, respondents' median response was to "Agree" that Sideshow tickets were easy to install and use. Respondents also "Agreed" that the tickets provided enough information about the comments on the specifications. Based on interview data, subscribing to notifications using Sideshow was very easy.

The ability to customize the ticket to change which comments were considered "new" and shown in the ticket tooltip was used by some participants. Half the November survey respondents (10) had customized a ticket's settings to change which comments were considered "new" and shown in the ticket tooltip. (Note, this question was not asked on the September survey.) In interviews participants also discussed changing the "new" setting, and the preferred setting seemed related to the rate of comments on the spec.

Design Improvements

The interviews and September survey data identified several ways to improve the tickets. In order to facilitate tracking a large number of specifications, participants thought tickets needed to be much smaller. They felt the title and the number of new annotations were most important to display on the ticket and "Agreed" that hyperlinks that opened the specification directly to a comment would be useful.

I introduced the smaller version of the ticket (shown in Figure 4.5) in mid-October, and also added hyperlinks to the comments in the tooltips. On the final November survey, 14 respondents (70%) preferred the second version of the ticket. I also asked respondents to rate the three most valuable pieces of information in the tooltip. As shown in Table 4.3, the new hyperlinks were quite popular. I also asked about including the context of a comment in the ticket tooltip, even though it was not currently implemented. Based on the survey data, participants using Sideshow tickets were less interested in seeing context of the comments than those using email notifications.

For the future, interview data suggests design improvements to try, including a ticket that summarizes several specifications, filtering comments in the tooltips, and visual changes to a ticket when new comments occur.

4.5 Discussion

The enhanced notifications were generally successful. Field study participants reported that both Sideshow and the detailed email notifications were useful, particularly in contrast to the dissatisfaction with default Web Discussions email notifications found by the initial usage survey. The field experience points to several critical issues to consider in designing other annotation notification systems.

4.5.1 Different Uses of Notifications

Annotation notification mechanisms generally need to be flexible enough to support both active monitoring and more casual tracking, as well as other uses. I found providing notification via different communication channels, with Sideshow and email, critical so that users can choose the delivery mechanism that best fits their needs. Providing detailed information in the notifications is also helpful, allowing users who are actively monitoring or passively tracking to make informed decisions about the importance of an annotation that has been made.

4.5.2 Roles and Notifications

The number of specifications a person is responsible for, as well as that person's role, affects the value and usage of notifications. More study is needed, but the data suggest that notifications become more useful as the number of specifications a user is responsible for increases. People responsible for many specifications, such as managers and tech writers, assessed notifications as being more valuable. In particular, notifications provide a way to monitor activity on the specification and decide when to revisit.

The interview and survey data also suggest very different opinions about notifications among authors of specifications. Some authors felt they already checked on their own specifications frequently enough and did not need notifications, while others appreciated the notifications as a way to track comments.

Notification may generally be more valuable for tracking specifications that cover related features or are from other work groups. Several users wanted to read and respond to more related specifications. Perhaps if notifications are more informative, users will subscribe to them to track related specifications.

4.5.3 Cultural Considerations

Prevailing group culture may affect notification usage as well. Based on the initial usage study and interviews, groups use several different methods to communicate feedback on specifications. The value of notifications may be heavily influenced by the amount a group relies on annotations for feedback compared to other options such as email or face-to-face meetings.

In addition, Cadiz et al. found that some notification email may be redundant anyway, since some users tended to send email directly when timely notification of a comment was important [CGG00]. Providing meta-awareness of who is subscribed to notifications may increase the value of automatic notifications, since with meta-awareness users could see who was receiving automatic notifications and then avoid sending duplicate notifications. I discuss issues related to supporting meta-awareness in future work in Section 7.2.2.

4.5.4 Configuration and Subscription

Due to the range in interest levels and rate of comments made on specifications, easy configuration of notifications is critical. People generally agreed about the content of the notification messages, but opinions varied about whether users preferred email notifications daily, weekly or immediately. For Sideshow tickets, users also had different preferences for which annotations should be considered “new.”

The field study reinforced the importance of making subscribing convenient. One advantage of a Sideshow ticket was the ease of dragging it from the specification document over to the Sideshow sidebar. For email subscriptions the participants had to go to a separate web page. This may be why fewer people tried the email notifications.

Although default Web Discussions email subscriptions can be done directly from a spec, they still require user action. Users may favor an automated approach in which they are subscribed to daily notifications when they first comment on a document. Opt-out mechanisms can be frustrating for users, but if notifications contain enough information and are easy to unsubscribe or filter, this could be a popular feature.

4.5.5 “Replies to me”

I initially thought informing people of replies to their annotations would be particularly valuable. However, it appears Web Discussions are used less as a place for quick conversation and more for issues to be tracked. Knowing about replies to your comments may be interesting, but less important to know immediately.

That said, on other tasks reply notification may be more important. For example, in the field study described in Chapter 3 it appeared some of the students would have appreciated reply notifications. Also, some users in this field study did sign up for immediate email for replies to their comments, so supporting this capability does seem worthwhile. Furthermore, wider use of notifications may lead to quicker response times, and could make features such as specialized reply notifications more valuable.

4.5.6 Notifications about document changes

The notification enhancements focused on making people aware of annotations made using Web Discussions. Many people said that they wanted similar detailed information about updates to the specification document. The existing notification mechanism can notify people of document changes, but the notification messages do not contain much information. In interviews, users indicated that knowing that the specification changed and perhaps some measure of the amount of change (e.g., small, medium, major) would help. A first step would be to integrate the time of the most recent file content change into the annotation notifications.

4.6 Conclusion

An effective and useful notification mechanism is an important part of a shared document annotation system. This study of a commercial annotation system found that it failed to meet user's notification needs. After a large-scale field study and incorporating feedback, the enhanced system shows promise. In particular, providing more information about new annotations, supporting multiple communication channels through which notifications could be received, and allowing customization of notification messages were popular. Overall awareness of annotation activity on specifications increased with the enhancements.

With the increasing use of online documents, annotation is an active focus of research and development. This study has identified several important considerations for designers of annotation notification systems. Users want notifications to provide as much detail as possible while requiring minimal effort to subscribe to or monitor. Within the context of one task, users have different preferences for notification settings. The usage of configuration options highlights the importance of making customization easy. Multiple channels to deliver notifications proved valuable to support different styles of use. Although understanding these design considerations is valuable, there remain several areas for future research including providing meta-awareness of who is subscribed to notifications on a document and exploring the value of contextual information in email.

Chapter 5

User Expectations for Anchoring Annotations

Annotations, as the last two chapters illustrated, facilitate discussions about particular sections of a document and can be a very valuable way to collaborate. As described in Chapter 2, systems that support annotating web documents by multiple people, such as Microsoft Office Web Discussions [MOWD], typically store the annotations separately from the documents being annotated. In these systems, when users create new annotations, the system saves information about which document the annotation belong to and the position in the document where the user made the annotation. When a user browses to an annotated document, the annotation system can add the annotation to the local copy of the document in the proper position. While storing the annotations separately complicates the display of the annotations, it makes them more practical for asynchronous discussion. For example, users no longer need permission to modify a document before annotating it, so document authors can post a document and easily collect feedback on it from a large group of people. Also, groups can ensure their annotations are kept confidential and only seen by trusted group members, even while commenting on public documents such as working drafts of proposed standards.

When annotations are used in reviewing documents, as for the software specifications described in Chapter 4, collecting the feedback is only one step in the document review process. After gathering feedback, the document author revises the document. However, changing the document can have unintended effects on the annotations. Often, if modifications to the document change the text that was annotated, the annotation system will lose track of the annotation's proper position in the document. This *orphaning* of annotations, where annotations lose their link to their proper position in the document, was a key complaint when Cadiz et al.[CGG00] observed the use of Office Web Discussions [MOWD] by roughly 450 users over a 10-month period. Participants in the notifications field study described in Chapter 4 also discussed their frustration with orphaned annotations in Web Discussions during interviews.

The problem of orphaning is unique to annotations on online documents, as paper-based documents do not change underneath the annotator. As more documents appear online and as other traditionally paper-based document processes become increasingly digital (such as editing and revision), “robust annotations” that remain associated with the correct portion of the document across modifications will become crucial.

Correctly positioning annotations in a revised document is a difficult problem. Some annotation systems try to work around the problem by limiting where an annotation can be placed [DH95b, MOWD]. Other researchers have begun to explore algorithms for robustly saving an annotation’s position and finding it in a modified version of the document [GSØ99, PW00]. However, focusing solely on algorithmic approaches to this problem neglects a crucial step. No one has asked users what they expect an annotation system to do when a document changes.

This chapter describes two studies designed to take that step. My belief was that observing how people placed annotations in modified document would help me understand user expectations and design a robust positioning algorithm that meets those expectations. In the initial study, participants transferred existing annotations to a modified version of a document and also rated annotations positioned automatically in the same modified version. In this study, I found it was unexpectedly difficult for participants to work with annotations that they had not made.

Participants in the second study made their own annotations, and then rated how well a simple algorithm positioned their annotations in a modified version of the document. The results indicate that participants considered some parts of the text that they had annotated especially important, and focused on how well these “keywords” and phrases were found in the modified version of the document. Participants also seemed to pay little attention to the text surrounding their annotations. Finally, even when some of the original text associated with an annotation was found, in some cases it seemed participants would have preferred that the algorithm orphan their annotation.

Section 5.1 lays out a framework for annotation position information and types of document modifications. Sections 5.2 and 5.3 describe the methodology of the two studies and their results. In Section 5.4 I discusses how to use these results to construct better robust positioning algorithms. Section 5.5 concludes with suggestions for future research to continue investigating user expectations.

5.1 Framework

Approaching the annotation positioning problem requires understanding two key components: how digital annotations work and how documents may be modified.

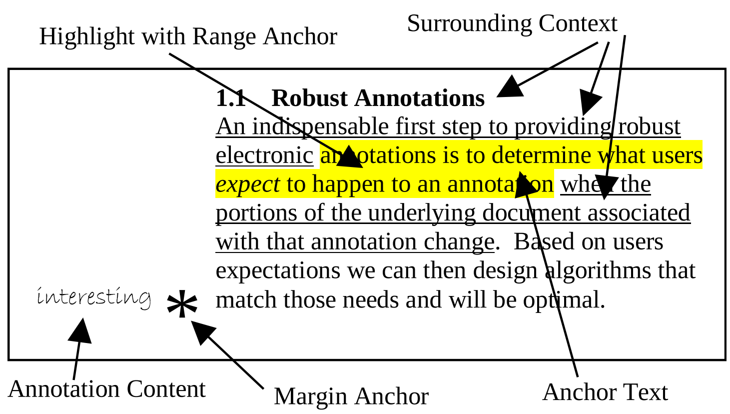


Figure 5.1: Annotation example showing anchor types and surrounding context. In the user studies I focused primarily on annotations with range anchors.

5.1.1 Annotation Definition

An *annotation* is a marking made on a document at a particular place. Each digital annotation is composed of two items: Some *content* (for example, a user comment or highlighter ink), and an *anchor* (the information used to position an annotation in the document).

Marshall [Mar98] has classified paper-based annotations into 4 groups, based on whether the annotation content is *explicit* to another reader (e.g., a scribbled note) or *implicit* (e.g., yellow highlighter ink implying importance), and whether the annotation's anchor is a *margin* anchor (e.g., asterisks, a note scribbled to the side of a paragraph) or a *range* anchor (e.g., highlighted text, circled word). Figure 5.1 illustrates the two anchor types. The highlight annotation has a range anchor and implicit content, and the asterisk annotation has a margin anchor and explicit content.

5.1.2 Robust Anchor Representation

The content and anchor information for digital annotations is often stored separately from the annotated document. This strategy allows people to annotate documents even if they do not have permission to modify them. However, this also requires high quality anchor information. Without a good anchor, a system cannot position annotations correctly in a document for display to users.

To insure correct annotation positioning in a document even when the document changes, a system needs to use *robust anchors.* Robust anchors could potentially use two types of information to identify an annotation's location:

- **Anchor text information**: The complete text marked by the user (see Figure 5.1).
- **Surrounding context information**: Text and document structure near the annotation, but not explicitly selected by the user (see Figure 5.1).

One goal of the studies was to determine the relative value of both types of information to users when trying to position annotations in a modified document.

Anchor Text Information

The key role of anchor text information is to identify the annotation's position in a document uniquely and efficiently. As discussed in Chapter 2, numerous strategies exist to address this problem: storing simple character offsets, keywords, or the entire text string selected by the user. These methods only work when a user explicitly marks text. Margin annotations do not mark any text explicitly, instead relying on proximity to suggest which text they are associated with. For example, the asterisk in Figure 5.1 could relate to just the last few words, or the last sentence, or the complete paragraph.

Surrounding Context

The surrounding context is the text that is near the annotation, but not explicitly selected by the user. For example, the underlined text in Figure 5.1 can be considered part of the surrounding context for the highlight annotation. More generally, we can think of the surrounding paragraph, subsection, section, and meta-information, such as HTML markup tags, as part of the surrounding context.

Surrounding context is important for several reasons. First, it is the only way to identify where margin annotations should be positioned. Second, surrounding context can be used, as in Robust Locations [PW00], to verify that the correct position for the annotation anchor has been located. Third, the range of text specified by the reader may not be carefully chosen [Mar98]. For digital annotations, this may mean that people expect annotations to remain intact if the surrounding context remains, even if large changes occur in the anchor text information.

Table 5.1: Annotation anchor modification types. The table presents different types of modifications that an annotation's anchor text may undergo in the document modification process. I use this classification in the study to understand users' expectations for robust annotation positions.

Modification Type	Modification	Description
Delete	Minor Delete	Between 1 character and half of the anchor is deleted.
	Medium Delete	More than half of the anchor is deleted.
	Total Delete	Entire anchor is deleted.
Reword	Minor Reword	Between 1 character and half the anchor is reworded.
	Medium Reword	More than half the anchor is reworded, reorganized, or split into multiple pieces.
	Total Reword	Complete anchor is reorganized. Typically only a few key words remain.
Move Anchor Text	Anchor Text Indirect	Anchor text itself doesn't change, but the text around it does.
	Anchor Text Direct	Anchor text moves within the paragraph or changes paragraphs.
Move Paragraph	Paragraph Indirect	The paragraph in front of the annotation's paragraph changes.
	Paragraph Direct	The paragraph containing the annotation moves forward or backward.

5.1.3 Document Modifications

Documents may be modified for different reasons and in a variety of ways. It is important to differentiate between modifications made to address annotations and modifications made independently of annotations.

A modification may be made in response to an annotation. For example, a sentence may be highlighted with "please reword" written in the margin next to it. If the author rewords the sentence, it is difficult to know whether a system should try to position and show the annotation in the modified document. I do not focus on robust positioning of these editing annotations in this research. A solution based on a "resolve button" is discussed elsewhere [CGG00].

Modifications may also be made independently of any annotation. For example, an author may generate a new draft of a document while a colleague marks up a previous draft. This is the case I focus on here.

Table 5.1 shows my modification classification scheme. A piece of text can experience three main types of modifications: deletes, rewords and moves. Note that a single piece of text may undergo several of these modifications at once.

Although delete and reword modifications are obvious to a reader, move modifications are more complicated. For example, if the paragraph prior to the annotation is deleted, the surrounding context of the annotation changes without any change to the actual text that the annotation is anchored to.

5.1.4 Study Focus

I chose to focus on a limited number of common annotation and modification types in these studies. First, because the majority of digital annotations use range anchors, not margin anchors – it is easier to highlight text with a mouse than it is to draw an asterisk in a margin – I focused on annotations with range anchors.

Second, I focused on annotations that were made during active reading of text documents, similar to those studied by Marshall [Mar97], instead of examining editing annotations. Annotations made during active reading are often meant to persist for future reference, perhaps as part of an asynchronous discussion or for giving feedback to the author. In this case the context of the annotations is particularly important, for example to contextualize the discussion or provide detailed feedback. Thus active reading annotations are precisely the type of annotation that needs to survive document modifications.

5.2 Pilot Study: Annotations on Paper

To examine user expectations for robust annotation positions, I conducted two user studies. The main goal of the pilot study was to explore what users perceive as annotation context. I did this by isolating the task from user interface design concerns and having participants perform the task for which I was trying to design an algorithm. Participants transferred annotations from an original document to a modified version (on paper). The hypothesis was that observing the thought processes people use to identify the context of an annotation and place it in a modified document would help create a software algorithm that does what people expect.

5.2.1 Experimental Method

Eight people who had at least basic computer knowledge participated in the study. All were either college educated or college students, and all read for at least 30 minutes on average every day. Participants received a gratuity for spending two hours participating in the study.

Participants performed three main tasks. First, they looked at a pre-annotated document and told me what they thought the context for each annotation was. The document was a news article with a variety of types of annotations on it (a selection of highlights, underlines, margin notes and symbols created by four coworkers). Second, they transferred the annotations from the original document to a version modified by a technical writer. Third, participants compared the original annotated document with a modified version in which “a computer” had positioned the annotations. The annotations were actually placed in the modified version by a person using an algorithm similar to the method reported in [PW00]. Participants rated how well the computer did using a 7-point Likert scale.

5.2.2 Lessons Learned

Instead of obtaining data about the cognitive processes people use to transfer annotations, I learned that making explicit the context of annotations and then transferring them is a difficult task. Problems seemed to stem from the fact that people were asked to work with annotations that they did not make. I had consciously designed the task this way so that I could control what type of modifications each annotation was subjected to in the altered version of the document. However, if a participant did not understand (or agree) with an annotation, it negatively affected that person’s ability to specify its context and to transfer it. One participant quipped, “*Again we have another star here. That’s a horrible annotation right there.*” Another said, “*I don’t see how it [another annotation] applies, but I guess it does.*” One participant even refused to transfer annotations that were “*someone’s opinion that I didn’t agree with.*”

Rating the computer’s transfer of the annotations was also difficult because participants were not working with annotations that they had made. Instead of rating the new position of the annotation in the modified version, several participants rated how valuable they thought the annotation was. Also, because the task was done on paper (where it was clear that a person had marked up the document), people had a difficult time understanding that I was pretending a computer had positioned the annotations. I applied the lessons learned in this pilot study when designing my second study.

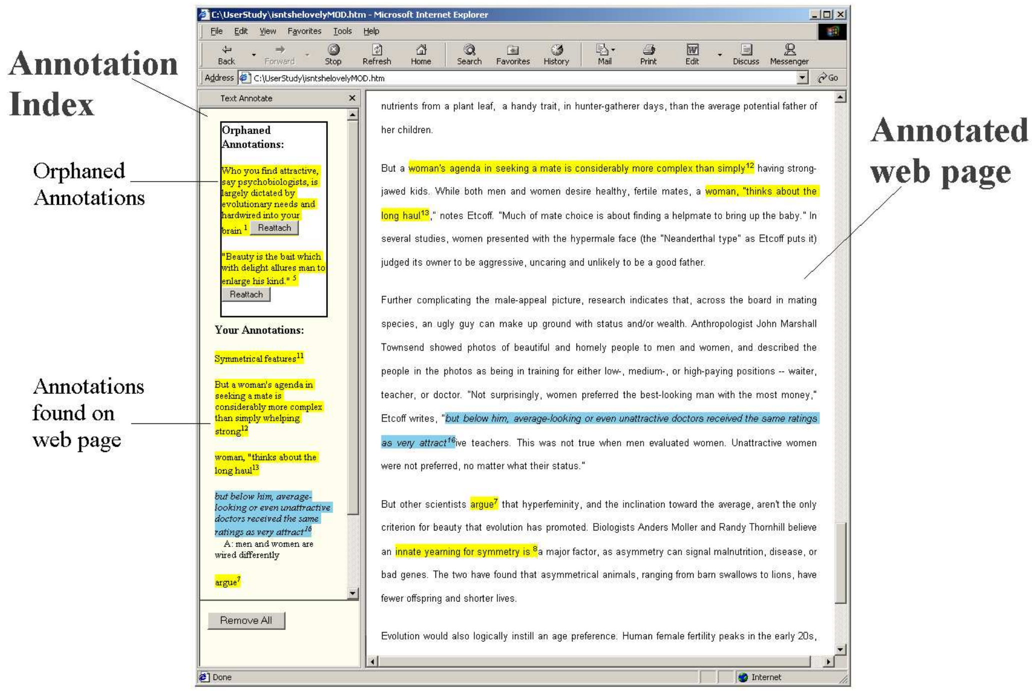


Figure 5.2: Text annotation software used in the second study to create notes and highlights on a web page. The annotation index lists the annotations for the current page, including the orphaned annotations that could not be placed on the page. The annotations shown are characteristic of those made by participants.

5.3 Second Study: Digital Annotations

Based on the experience from the pilot study, I conducted a second study in which participants created their own annotations on a digital document. I narrowed the focus to examine user ratings of annotation positioning done by a computer. The primary goal for this study was to gauge users' reactions to a relatively simple repositioning algorithm, especially when it failed.

5.3.1 Annotation Software

For this study, I extended Microsoft Internet Explorer, as shown in Figure 5.2, to allow people to highlight and make notes on web pages. This software was later developed into the WebAnn prototype described in Chapter 3.

A user makes an annotation by using the mouse to select a portion of text on a web page, and then left-clicking the selection. A menu pops up from which the user can choose to highlight or attach a note to the selected text. Highlighted text is displayed with a yellow background, and text with a note attached is displayed with a blue background. The annotation index window on the left shows a list of all annotations for the web page and displays the contents of any note annotations. Participants could delete annotations by left-clicking on an existing annotation and selecting “delete” from the menu. While this is a very simple interface and it was sufficient for the study, Wojahn et al.[WNB98] suggest that aspects of the interface may affect the annotations people make. However, this was not the focus of the study.

Annotation Positioning Algorithm

I included a simple algorithm to reposition annotations if an annotated document was modified. The algorithm was similar to the context method reported in [PW00]. The algorithm saved the text selected by the participant as the anchor and then used text matching to find the anchor position in the modified version. If the entire original text was not found, the algorithm alternated cutting words off the front and back of the anchor text while looking for the shorter text in the modified document until it found a partial match, or until the length of the anchor fell below 15 characters. If the algorithm could not find a location for the annotation, it orphaned the annotation. Orphaned annotations were displayed at the top of the annotation index (see Figure 5.2).

This algorithm is fairly simple. It does not take into account surrounding context or search for the anchor text in a more sophisticated manner, and it weighted the center words of anchor text more heavily than the words toward the beginning and the end. I decided to use this algorithm to gather observations of user expectations before developing a more complicated algorithm. I expected the algorithm to fail often, alerting me to the scenarios where participants were most unhappy with the algorithm’s performance.

5.3.2 Experimental Method

For this study, 12 participants were recruited in the same manner as the first study. Participants were first given a brief training task to familiarize themselves with the system, and then given the task of annotating a document so that it “could be skimmed quickly by a busy executive.” The document was a general interest news article from the web. Next, participants were told that an updated version of the document was available, but that rather than repeating the task of annotating the document, they would have the computer transfer their annotations from the old to the new document. Participants then examined each annotation and rated its position in the new document on a 7-point Likert scale where 7 was “perfect”, 4 was “ok”, and 1 was “terrible.”

In this study, because participants made their own annotations, I needed to create an updated version of the document before the study with modifications that would affect participant’s annotations. To do this, I asked a few pilot study participants to annotate the document (on paper). Then I made changes in the original document in places where people tended to make annotations. A colleague unfamiliar with the annotation positioning algorithm created a second updated version. If participants quickly finished the rating task using the first updated version, I had them repeat the task for the second updated version.

5.3.3 Results

The main purpose of this study was to examine participant satisfaction with the algorithm’s attempt to reposition annotations in the updated document. The 12 participants made a total of 216 annotations and then rated their satisfaction with how each annotation was positioned in the first updated version. Half the participants also rated the positions of their annotations in the second updated version. A total of 302 position satisfaction ratings were collected.

I present participant position satisfaction ratings in the following sections by breaking down the set of 302 ratings into three logical groups based on the changes made to an annotation’s anchor text:

- **Same**: Annotations anchored to text that did not move or change.
- **Move**: Annotations anchored to text that was moved from one portion of the document to another, but that otherwise did not change.
- **Complex**: Annotations anchored to text that was changed and possibly moved.

I expected high satisfaction ratings for the transfer of annotations in the **Same** group because the algorithm finds all such annotations. For annotations in the **Move** group I still expected fairly high ratings, since the algorithm also finds these annotations. However, I believed that if the anchor text moved significantly in the document, this would change its surrounding context, and perhaps render it irrelevant. In this case, participants might prefer the annotation to be orphaned.

For annotations in the **Complex** group, I expected lower scores due to the simplicity of the algorithm. I expected instances where participants would be unsatisfied with how much of an annotation’s anchor text the algorithm found or that an annotation had been orphaned. I also believed that participants would always rate non-orphaned annotations higher than annotations that were orphaned, except when the orphan was caused by deletion of the entire anchor text.

Same: When the Anchor Text Does Not Change

Although the positioning algorithm is simple, it is guaranteed to find annotations attached to unique text that does not move or change. 47 out of 302 position ratings fell into this category. As I expected, the median participant rating for these annotation positions was a perfect 7.0. When the anchor text remains the same and the system finds it in the new document, participants are satisfied.

Move: When the Anchor Text Moves

121 of the position ratings were for annotations attached to anchor text that was moved in the updated document, but not otherwise changed. I focused on move modifications noticeable to a human reader. For example, a paragraph might have been moved from one page to another. 100% of annotations attached to text that only moved were found in the updated document. This was due to my algorithm's use of simple text matching to find an annotation's anchor text and because participants attached their annotations to unique sections of text. The median participant rating for these annotation positions was 7.0.

The high ratings given for these annotation positions surprised me somewhat. I expected that if the text to which an annotation was attached moved significantly, there would be times when an annotation would lose relevance and need to be orphaned. However, the data indicate that this is not the case. Thus, perhaps the surrounding context of an annotation is of lesser importance when considering factors that contribute to keeping participants satisfied with automated annotation positioning. It would be interesting to explore whether users feel the same way about the surrounding context for editing and margin annotations.

Complex: When the Anchor Text is Modified

134 of the position ratings were for annotations attached to text that was changed in some way in the updated document. Of these annotations, the algorithm successfully transferred 71 and orphaned 63. Note that a piece of text may have been both changed and moved, but since data in the previous section indicate that ratings are independent of moves, I focus primarily on how the anchor text changed.

To analyze this set of annotations, I classified the changes that were made to an annotation's anchor text. Sometimes just one word was changed, and sometimes the entire sentence was rewritten. Changes were coded using the six "delete" and "reword" categories outlined in Table 5.1, and these encodings were used to compute a modification score for each annotation. Minor rewords and minor deletes were given one point and medium rewords and medium deletes were given two points. Using this scheme, higher scores indicated more drastic changes, with a highest possible combined modification score of 3. Total deletes were treated as a separate category and automatically given a score of 4. Total rewords were eliminated from the analyses because only one such case occurred.

Reliability of these classifications was verified by having a colleague not involved with the research code a representative sample of the anchor text changes. Inter-rater reliability for the modification score was high ($\alpha = .90$)[2].

When Annotations are Orphaned

Table 5.2 shows the median position ratings for annotations that were orphaned in cases where the text changed. As I expected, the table shows that participants gave the lowest ratings when little modification occurred to the text and the annotation was not found. The table also suggests that ratings increased as more modifications occurred, to the point where participants gave the highest ratings to orphaned annotations when all of an annotation's anchor text was deleted.

Comments that participants made while rating orphaned annotations also support the hypothesis that as the amount of text change increases, people are more satisfied with orphaning the annotation. For one annotation, a participant remarked that the document "*changed around*

Table 5.2: Median participant position satisfaction ratings, on a 1 to 7 Likert scale, for annotations where the anchor text changed and the annotations were not found (orphaned). As the amount of modification to the anchor text increased, participants seem more satisfied that the annotation had been orphaned.

Modification Score	Rating (number of annotations)
1	1.50 (12)
2	3.0 (18)
3	3.0 (7)
4 (total delete)	7.0 (25)

[2] α ranges from 0 – 1.0 and indicates the amount of agreement between raters. 0 denotes no agreement and 1.0 denotes perfect agreement [EM].

enough and the keywords left out of the second article, I could see it might not find that." Of another annotation, a participant observed that the modifications "*redid [it] entirely…makes sense they [the algorithm] didn't find that one.*"

When Anchor Text Changes and Annotations are Found

Table 5.3 shows the median position ratings for annotations that were found in cases where the anchor text changed. Note that a successful annotation transfer includes cases where only part of an annotation could be transferred. Suppose a person made the following highlight annotation:

The quick brown fox jumped over the lazy dog.

Below is an example of modified text and the partial anchor text the algorithm would have found:

The quick fox jumped away from the dog.

To take into account partially found annotations, I also examined this set of annotations by looking at what percentage of the annotation anchor text was found in the modified document. These percentages are listed in the columns of Table 5.3.

The data in Table 5.3 suggest two trends. First, not surprisingly, the greater the percentage of the annotation anchor text found, the more satisfied people are (bottom row of Table 5.3, read left to right). Second, and somewhat counterintuitive, the more drastic the modifications to the anchor

Table 5.3: Median participant position satisfaction ratings for annotations where the anchor text changed and some percentage of it was found. Participant satisfaction appears related to the amount of anchor text found and inversely related to the amount of modification that occurred to the anchor text. Number of annotations in each case is in ()'s.

Modification score	1 to 24% found	25 to 49% found	50 to 74% found	75 to 100% found	Overall
1	3.0(3)	3.0 (13)	3.0 (19)	6.0 (18)	4.5 (53)
2	2.0 (9)	3.0 (6)	4.0 (1)	5.0 (1)	3.0 (17)
3	-	3.0 (1)	-	-	3.0 (1)
Overall	2.5 (12)	3.0 (20)	3.5 (20)	6.0 (19)	4.0 (71)

text, the less satisfied people were when the annotation anchor was found (right column of Table 5.3, read top to bottom). This was unexpected. I thought that participants would be more impressed when the system was able to find part of the annotation's anchor text even when significant changes occurred.

Finally, somewhat surprising was the participant's median ratings of 3 for both found and orphaned annotations with modification scores of 2 & 3 (see Tables 5.2 & 5.3). I had expected found annotations to always be rated higher than orphans not caused by a total delete of the anchor text.

5.4 Discussion

The results from the studies provide valuable insight for designers of annotation systems.

5.4.1 Surrounding Context is Less Important

As noted previously, robust anchors can be created by storing an annotation's surrounding context and anchor text information. I was surprised when the studies indicated that users might not consider surrounding context very important for annotations with range anchors.

I observed rather casual text selection where annotation boundaries were influenced by document formatting (for example, ends of lines) similar to Marshall's observation for annotations on paper [Mar98]. I thought this might cause participants to expect the annotation transfer algorithm to perform a more sophisticated search for related text when the anchor text was deleted or moved, but the data do not support this. Participants gave very high position ratings for annotations that were orphaned due to the original text being deleted and for annotations attached to text that was significantly moved.

This does not necessarily mean that robust positioning algorithms should not save surrounding context. Rather, users may not consider it very important, so it should perhaps be weighted less heavily in algorithms that employ it. Future research should examine whether this finding was due to the focus on annotations made during active reading instead of other types of annotations, such as ones made during editing.

5.4.2 Focus on Keywords

When examining the particular cases in which participant ratings were low, I found that participants often expected the system to do a better job locating key words or phrases. Comments included:

- *"The key words are there, it should have been able to somehow connect that sentence [in the modified version] with the original"*
- *"Should have gotten that one, at least the quote."*
- *"Should have at least highlighted the name."*
- *"Doesn't pick up a change in wording that means essentially the same thing."*

Thus, when designing robust positioning algorithms, it may be helpful to pay special attention to unique or "key" words in the anchor text, as the ComMentor [RMW97] system does. Participants also appear to consider names and quotations as particularly important. A simple thesaurus or grammar parser may also be useful to recognize when simple rewords have occurred that do not change the sentence's semantics.

5.4.3 Orphan Tenuous Annotations

Based on Tables 5.2 and 5.3, two trends seem to emerge. First, if an annotation is found, users initially assign the highest rating and then move down the satisfaction scale based on how much of the annotation anchor text the algorithm found and how many modifications occurred. For orphaned annotations the process works in reverse. Participants start with the lowest rating and then move up the scale as more modifications are noticed, or when they realized the entire anchor text has been deleted.

These trends suggest that there may be a point at which, even though an algorithm may be able to find a highly likely location for an annotation in a modified document, the participant would be more satisfied if the annotation were orphaned. Testing this hypothesis further is a good area for future research.

5.4.4 Include User Intervention

If indeed systems choose to orphan some annotations even when they have a relatively good guess as to where annotations should be positioned, it may be helpful to provide users with a "best guess" feature that shows them where orphaned annotations might be located. This feature may also be helpful for situations where users need to insure all annotations are moved to a modified version of the document. Some of the system's "best guesses" may not be correct, but they may provide enough information for a user to easily reattach orphaned annotations.

5.5 Conclusion

As more documents exist digitally, annotations are poised to play an increasing role in document processes such as editing and providing feedback. However, to truly realize the potential for digital annotation, an annotation system must be able to robustly anchor annotations to text that is modified. A critical first step in designing robust anchoring algorithms that support users is understanding what they expect to happen to their annotations when the document is modified. The studies described in this chapter begin that exploration. The framework for annotation positions and types of document modifications also provides a basis for ascertaining the relative importance to users of different kinds of anchor text information.

For the types of annotations studied, the results suggest that participants paid little attention to the surrounding context of an annotation, and algorithms may want to give the surrounding context relatively little weight when determining an annotation's position. As for anchor text information, participants' comments stressed the importance of key words, proper names and quotations. I also found in certain cases, even when part of the annotation's anchor text is found, users may prefer that the positioning algorithm not place it in the modified document. The detailed data I collected are useful for determining potential thresholds for orphaning annotations.

While the results reveal valuable information about user expectations and have helped in designing the keyword robust annotation positioning algorithm described in the next chapter, it would be valuable to gather additional information about user expectations. In particular, determining user expectations for anchoring of other types of annotations, such as editing and margin annotations, as well as annotations used in other tasks and on different document types.

Chapter 6

Robustly Anchoring Annotations Using Keywords

Robust anchors that allow annotations to remain associated with the correct portion of the document through modifications are crucial. Several existing annotation systems have methods for locating positions within a document when the document changes [GSØ99, KKP+01, OAM99, PW00, MOWD, RMW97, Yee]. While these algorithms are robust to varying degrees, none of them take users' expectations into account.

In this chapter I introduce Keyword Anchoring, a robust anchoring method designed based on the user expectations gathered in the studies described in Chapter 5. Keyword Anchoring primarily uses unique words from the annotated document to anchor and re-position annotations, and it ignores any specific internal document structure. This allows it to reflect user assumptions about the document and to be used with many different digital document representations. Keyword Anchoring is complimentary to existing robust anchoring methods including Robust Locations [PW00] and WebVise [GSØ99]. In Section 7.3.2, which describes future directions for robust anchoring, I outline how Keyword Anchoring could be used in conjunction with these methods to provide additional robustness that may better meet user expectations.

In addition to presenting the Keyword Anchoring algorithm, I focus on issues that arise when positioning annotations in a modified document. In particular, results from the initial user assessment of the keyword algorithm suggested the value of showing annotations that have been re-positioned in a modified document with moderate confidence as guesses, rather than representing them as either correctly positioned or orphaned.

This chapter is organized as follows. Section 6.1 describes the Keyword Anchoring algorithm, including what information the algorithm saves for an annotation and how it locates a new position for the annotation if document modifications affect the annotation's original anchor. Section 6.2 discusses results from a user study of the algorithm that compared Keyword Anchoring to the simple algorithm used in Chapter 5. Section 6.3 explores the general implications of my work for robust anchoring algorithms and ideas for extending the Keyword Anchoring approach to handle complex anchors and different media types such as images and video. Section 6.4 summarizes how Keyword Anchoring can be used to improve annotation tracking when documents are modified.

6.1 Keyword Anchoring

Robustly anchoring annotations requires two steps: deciding what anchoring information to save for an annotation, and then using the anchor to find the annotation in the document. Based on user feedback, when designing the Keyword Anchoring algorithm I focused on unique words in the anchor text, put less emphasis on the surrounding context of the annotation, and experimented with different thresholds for when to display the anchor positions found by the algorithm in the modified document.

Keyword Anchoring is designed for situations where the annotations are not stored in the document, allowing users to annotate documents they do not have permission to modify. The algorithm also assumes the document is unaware of the annotation and the document author has not provided identifiers or other markers to support annotation. Finally, the Keyword Anchoring algorithm does not assume any particular document structure, document object model or language, only that the document contains words.

6.1.1 Creating Keyword Annotation Anchors

The user expectations found in the studies described in Chapter 5, in particular the focus of the users on keywords in the anchor text, greatly influenced the information saved in keyword anchors. The anchors are specialized for ranges of text rather than a particular point in the document.

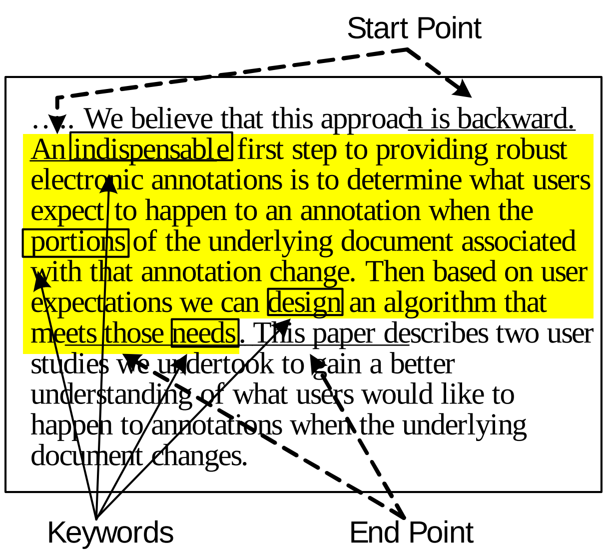
As shown in Figure 6.1, keyword annotation anchors contain:

- **An HTML bookmark for the selection**: An Internet Explorer specific string used to anchor annotations quickly in documents that have not changed.
- **Offset from start of document**: The number of characters from the beginning of the document to the anchor text.
- **Length of the anchor text**: The length, in characters, of the text selected by the user.
- **Information about the start and end points of the anchor text**: A small amount of text from the document surrounding the start and end of the anchor text. As shown in Figure 6.1, in the current implementation 15 characters of content is saved both to the left (*left content*) and right (*right content*) of the start and end points.

- **Information about keywords in the anchor text**: A list of unique words from the anchor text and their locations within the anchor text. The current implementation saves at least three keywords if possible.

The HTML bookmark is used in the current implementation to find an annotation's anchor when the document has not changed. While specific to Microsoft Internet Explorer, any browser is likely to have some method of uniquely marking a section of text (e.g., ID, bookmark, or character offset) and the appropriate data for a particular implementation could be used in the anchor. Alternatively this information could be ignored to ensure the anchor is completely independent of the document format.

Saving the keywords from the anchor text is the crucial part of the algorithm. The keywords are determined by selecting the words in the anchor text that are most unique with respect to the rest of the document. For a particular document, the algorithm initially calculates a map of word frequencies. For example, in a document "the" may occur 500 times, while "purple" occurs only twice. When the user creates an annotation, the algorithm select as keywords all words in the anchor text that only occur once in the document. As Figure 6.2 illustrates, for each keyword selected, the algorithm saves the word and also its distance from the start and end points of the anchor text. Saving keywords, rather than the entire anchor text as some systems do, reduces



Bookmark: IE specific opaque string
Offset from document start: 3712
Length: 298
Start Point Information
Left Content: "h is backward. "
Right Content: "An indispensabl "
End Point Information:
Left Content: "ets those needs"
Right Content: ". This paper de"
Keywords:
indispensable, distance from start: 3
distance from end : 295
portions, distance from start: 141
distance from end: 157
design, distance from start: 256
distance from end: 42
meets, distance from start: 281
distance from end: 17

Figure 6.1: A highlight annotation and its keyword anchor information

storage requirements. It also makes the anchors more robust in the face of wording changes.

Tradeoffs between space, speed, and accuracy may arise when deciding how many keywords the algorithm should save, particularly if many or no words in the annotation's anchor text are unique. In the current implementation all unique words in the anchor text are saved. If there are fewer than three words in the anchor text that occur only once in the document, the algorithm select words with increasing frequency (those that occur twice, those that occur three times, etc.) until at least three keywords have been found. All words in the anchor text are selected if it is shorter than three words.

In future versions of the algorithm it would be interesting to experiment with different ways of determining the minimum number of keywords that are saved. Perhaps basing the number of keywords on the length of the annotation's anchor text, the length of the document or the number unique words in the document.

6.1.2 Finding Keyword Annotation Anchors

After the user creates an annotation, the Keyword Anchoring algorithm uses the annotation's anchor to find its appropriate position any time annotations are viewed on the document. Similar to other methods, the Keyword Anchoring algorithm initially assumes the document has not changed and tries to find the anchor using only bookmark and offset information. While positioning the annotation quickly when the document has not changed is important, in this section I focus on the more interesting problem of finding an appropriate position when the document has been modified.

First, I discuss how the positioning phase of the algorithm finds and considers *candidate anchors* for the annotation in the modified document, and then how the *confidence score* is computed for the candidate anchor. Next, I describe how the presentation of the candidate anchor with the highest confidence score varies based on the score. The positioning phase of the Keyword Anchoring algorithm handles a variety of document changes including movement of anchor text, reordering of keywords, deletion of some of the anchor text (even keywords), and rewording of the anchor text to a certain extent.

Building Candidate Anchors

When the positioning algorithm does not find an annotation's anchor using the bookmark, it looks for the keywords from the anchor in the modified document. The basic Keyword Anchoring

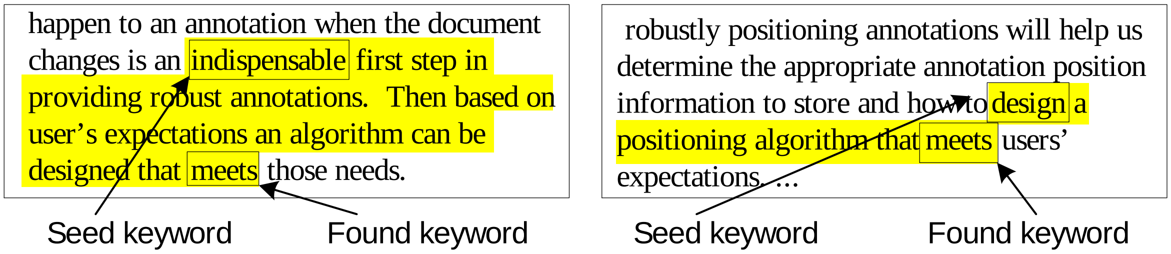


Figure 6.2: Two candidate anchors in a modified document after the keyword expansion step for the annotation from Figure 6.1.

algorithm is conservative and investigates all possible candidate anchors. The positioning phase of the Keyword Anchoring algorithm is as follows:

1. Locate a seed keyword: For each instance of a keyword the positioning algorithm finds in the document, it creates a candidate anchor containing that keyword as the *seed* keyword. The positioning algorithm then extends the candidate anchor by looking for other keywords and the start and end points from the original anchor.

2. Look for other keywords: The positioning algorithm searches for each of the other keywords in the original anchor after the current seed keyword. First, the positioning algorithm checks if the current candidate anchor already contains the new keyword. If so, it moves on to the next keyword. Otherwise, if the positioning algorithm finds the keyword after the current candidate anchor, it extends the candidate anchor to include the keyword as long as that does not make the candidate anchor unreasonably long. The current implementation requires that the candidate anchor not grow larger than twice the length of the original anchor. Figure 6.2 shows two of the candidate anchors the positioning algorithm finds in a modified version of the document for the annotation from Figure 6.1.

3. Look for the start and end points: After including as many keywords as possible in the candidate anchor, the positioning algorithm uses information about the start and end points of the original anchor. As shown in Figure 6.1, each point contains text to the left of the point, the *left content*, and text to the right of the point, the *right content*. Initially, the positioning algorithm only looks at the content from the start and end points that was explicitly part of the anchor text selected by the user, the right content of the start point and the left content of the end point.

The positioning algorithm looks in front of the current candidate anchor for the right content of the start point. If it finds part or all of the content, the positioning algorithm extends the

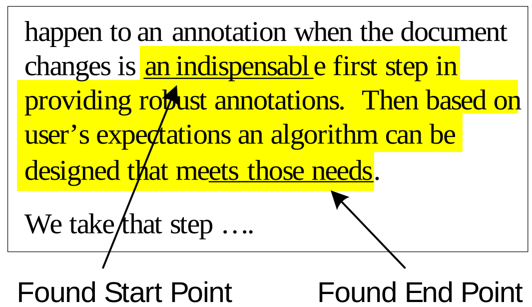


Figure 6.3: The complete candidate anchor for an anchor from Figure 6.2 including context found from the start and end points.

candidate anchor to include the found text if that would not make the candidate anchor too long. The positioning algorithm performs a similar check looking for the left content from the end point of the original anchor after the current candidate anchor. Figure 6.3 shows a candidate anchor that has been extended to include content from the start and end points.

Scoring a Candidate Anchor

After building the candidate anchor, the positioning algorithm then scores it based on its similarity to the original anchor. The features currently used to compute the confidence score for a candidate anchor are the number of keywords, start and end points, length, position in the document, and surrounding context. The surrounding context information helps handle cases in which the anchor text appears multiple times in the document. Although I describe how the candidate anchor is scored separately for clarity, in the implementation the positioning algorithm interleaves building and scoring the anchor.

Figures 6.4 and 6.5 contain the pseudo-code of the algorithm used for scoring a candidate anchor. The ComputeWeight function, shown in Figure 6.4, is used repeatedly in scoring the candidate anchor to determine how much of the weight associated with a particular feature should contribute to the confidence score. The more a feature in the candidate anchor differs from the original anchor, the less the weight associated with that feature will contribute to the confidence score. Table 6.1 gives the value for the weights used in the current implementation. These weights were influenced by user expectations and my implementation experience. They could benefit from additional empirical testing and tuning for particular applications.

```
ComputeWeight(DifferenceAmt, MaxDifference, Weight) returns a double

   //normalize DifferenceAmt to between 0 – 1 using MaxDifference parameter

   DifferenceNorm = (MaxDifference- DifferenceAmt)/MaxDifference

   //use the amount of difference to adjust how much the weight contributes
   //to the score

   ComputeWeight = DifferenceNorm*Weight
```

Figure 6.4: ComputeWeight function. ComputeWeight determines how much of the Weight parameter should contribute to the confidence score based on the size of the DifferenceAmt parameter. As DifferenceAmt increases, the function returns smaller fractions of the Weight parameter.

Table 6.1: Weights used in the current implementation when scoring the candidate anchor.

Weight for	Value
Keyword	100
Start and Endpoint (SEPointWeight)	50
Length	50
Position	20
Surrounding Content (SurContextWeight)	10

Figure 6.5 shows the ScoreCandidateAnchor function, which calculates the confidence score of a candidate anchor. In Step 1, the function examines the keywords in the candidate anchor. Each keyword in the candidate anchor besides the seed keyword increases the candidate anchor's confidence score depending on the relative change in distance between the seed keyword and the keyword from the original anchor to the candidate anchor. Based on user expectations gathered in the study described in Chapter 5, keywords contribute the most points to the candidate anchors confidence score.

In Step 2, the function determines if the candidate anchor contains the original anchor's start and end points. If the start or end point is present in the candidate anchor, the function compares how much the point's position has changed relative to the seed keyword. Step 3 compares the length of the candidate anchor to the length of the original anchor to determine how much the length weight should contribute to the confidence score. Step 4 examines the change in position of the candidate anchor in the modified document relative to the position of the original anchor in the original document. Step 5 checks for the surrounding context of the original anchor around the candidate anchor. Although currently the surrounding context is always scored, it might be

```
ScoreCandidateAnchor(CandidateAnchor, OriginalAnchor) returns an integer

//Step 1: Score keywords in candidate anchor
   totalScore = 0
   For each keyword k, (besides the seed keyword, s)
       DistanceChange = abs((distance between s and k in candidate anchor)
                        - (distance between s and k in original anchor))
       totalScore += ComputeWeight(DistanceChange, MaxDistChange, KeywordWeight)


//Step 2: Score Start/End Points
   if candidate anchor includes the right content of the start point
       StartDistChange = abs((distance between s and start in candidate anchor)
                         - (distance between s and start in original anchor))
       totalScore += ComputeWeight(StartDistChange, MaxDistChange, SEPointWeight)

   if candidate anchor includes the left content of the end point
       EndDistChange = abs((distance between s and end in candidate anchor)
                       - (distance between s and end in original anchor))
       totalScore += ComputeWeight(EndDistChange,  MaxDistChange, SEPointWeight)

//Step 3: Score Length of candidate anchor
   LengthChange = abs(length original anchor – length candidate anchor)
   totalScore += ComputeWeight(LengthChange, MaxLenChange, LengthWeight)

//Step 4: Score Position
   PositionChange = abs(location of original anchor –
                          location of candidate anchor)
   totalScore += ComputeWeight(PositionChange, MaxPosChange, PositionWeight)

//Step 5: Score Surrounding Context
   if the left content of the start point is in front of the candidate anchor
             totalScore += SurContextWeight
   if the right content of the end point is behind the candidate anchor
             totalScore +=SurContextWeight

//Step 6: Normalize the score to between 0-100
   maxScore = (KeyWordWeight*(number of keywords in original anchor-1))
             + 2*SEPointWeight + LengthWeight + PositionWeight
             + 2*SurContextWeight
   ScoreCandidateAnchor = 100 * (totalScore/maxScore)
```

Figure 6.5: ScoreCandidateAnchor function. ScoreCandidateAnchor computes the confidence score for a candidate anchor based on its similarity to the original anchor using the ComputeWeight function shown in Figure 6.4. The confidence score is based on: the number of keywords in the candidate anchor and how much they have changed in position, the presence and change in position of the start and end points, the change in length of the anchor, the annotation's change in position, and the presence of the surrounding context from the original annotation.

better in the future to consider the surrounding context only if the anchor text appears repeatedly in the document.

The last step, Step 6, normalizes the confidence score to a value between 0 and 100 by dividing by the maximum possible score for the original anchor. While the goal of the scoring

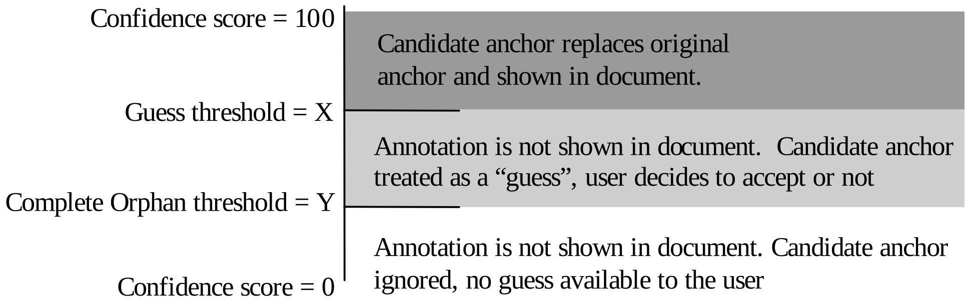


Figure 6.6: Thresholds used to interpret the confidence score of the candidate anchor. One of the goals of the user study described in Section 6.2 was to determine appropriate values for the Guess and Complete Orphan thresholds.

function is to give keywords the most weight, in the current implementation if there are fewer keywords, the relative value of the end points, length, position, and surrounding context increases.

Presenting the Best Candidate Anchor

After considering all possible candidate anchors, the positioning algorithm uses the one with the highest confidence score to replace the original anchor. The confidence score of the candidate anchor denotes how closely the new anchor matches the original and determines how to present the anchor to the user. As shown in Figure 6.6, if the confidence score exceeds the Guess threshold then the candidate anchor replaces the original anchor and the positioning algorithm places the annotation in the document at the new anchor. With a confidence score above the Guess threshold, the algorithm has found a place for the annotation that should meet user expectations. Although not currently implemented, users should be able to easily modify the found anchor to allow them to make adjustments if any of the anchors do not meet their expectations.

When the confidence score of the final candidate anchor falls below the Guess threshold, the positioning algorithm orphans the annotation and does not place it in the document. The algorithm instead makes the candidate anchor available to the user as the algorithm’s best guess. The user can then specify the new position of the annotation choosing to accept the guess, modify the guess, select a different anchor text or delete the annotation. If the confidence score of the anchor is below the Guess threshold, even though the algorithm found a possible location, user interaction is necessary to ensure the new anchor meets user expectations. One of the primary

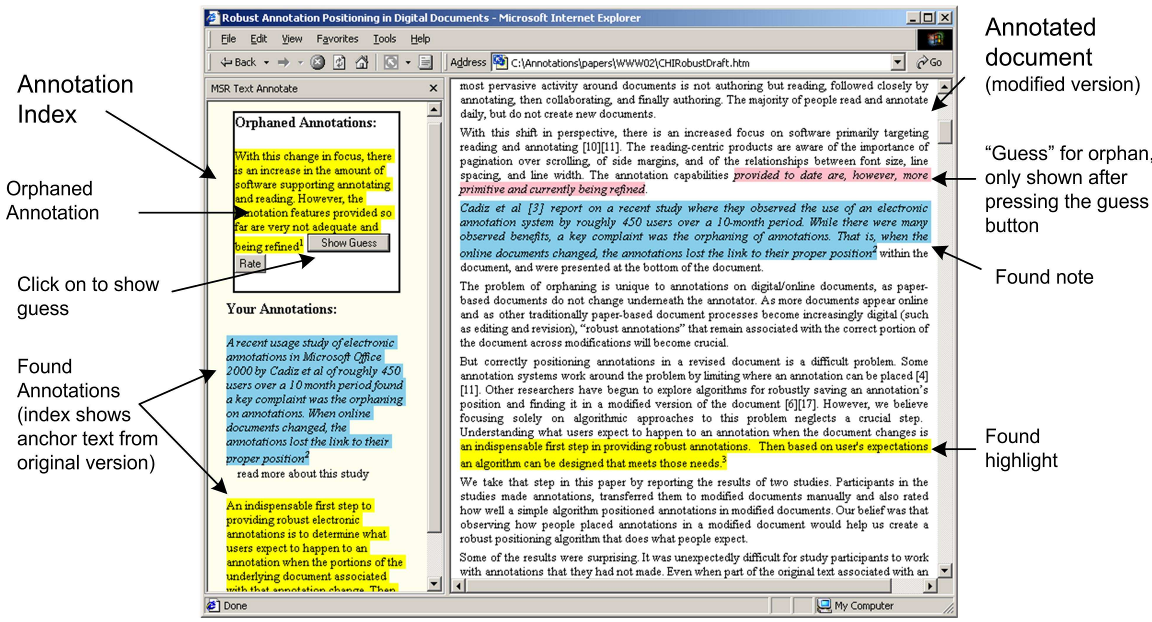


Figure 6.7: Text annotation software used by participants in the user study. Annotations were positioned in the modified document by the Keyword Anchoring algorithm with a guess threshold of 50. Clicking on the guess button for an orphan (if available) shows the guess in the document. The index shows the text selected by the user in the original document so participants could easily rate the new position found by the anchoring algorithm.

goals of the user study described in the next section was to determine an appropriate value for the Guess threshold.

After the user study, I added a Complete Orphan threshold to the Keyword Anchoring algorithm as shown in Figure 6.6. When the confidence score falls below this value, even if the positioning algorithm identified a candidate anchor algorithm, it is not shown as a guess. Scores below the Complete Orphan threshold mean that the vast majority of the anchor text has been deleted from the document and the candidate anchor is unlikely to meet user expectations.

6.2 User Assessment

To conduct a user study, I added Keyword Anchoring to the annotation prototype system described in Chapter 5. A user makes an annotation by selecting text on a web page, and then left-clicking the selection. A menu pops up, from which the user can choose to highlight or attach a note to the selected text. Highlighted text is displayed with a yellow background, and text with a note attached is displayed with a blue one. Annotations can be deleted by left-clicking on an existing annotation and selecting "delete" from the menu. Annotations and anchors are stored on a server using the Common Annotation Framework (CAF) [BG01]. I later developed this prototype into the WebAnn system described in Chapter 3.

Figure 6.7 shows the interface after a user's annotations have been positioned in the modified version of a document. The annotation index window on the left displays a list of all annotations for the web page and shows the anchor text selected by the user in the original document so participants could easily rate the position found in the modified document. Any orphaned annotations are listed at the top of the index. Clicking on the "guess" button for an orphaned annotation (if available) highlights the algorithm's best guess in the document.

6.2.1 Experimental Method

Using the prototype, I performed a lab study to verify that the keyword approach met user expectations and to gather some data to assist in choosing a suitable Guess threshold. Eight people recruited from the general public participated in the study for a gratuity. Participants were either college educated or college students and read at least 30 minutes on average every day.

Participants performed the same task as in the user study described in Chapter 5. After reading and annotating an article, they rated where their annotations were placed in a modified version of the document. The ratings were done on a 7 point Likert scale, where 7 was "perfect", 4 was "ok" and 1 was "terrible." A colleague unfamiliar with the Keyword Anchoring algorithm created the modified version before the study.

For comparison purposes, participants rated the positions of their annotations for both the keyword algorithm (Key) and a simple text search algorithm (Simple) described in Section 5.3.1. For positions found by the simple text search algorithm, an anchor's confidence score was the percentage of the original anchor found in the modified document. The simple text search algorithm is similar to robust positioning algorithms used in a number of systems, in particular the context method reported in [PW00].

Participants rated the results of different algorithms in positioning their annotations in a modified document. Half the algorithms showed all the annotation anchors they found in the document regardless of their confidence score (Key0, Simple0), and the other half showed annotation anchors that scored below 50 points as orphans with guesses available (Key50, Simple50). So there were a total of four different configurations: Key0, Key50, Simple0 and Simple50. For a particular algorithm, the position found for the annotation in the modified document was the same for both versions (e.g., Key0, Key50). However, if the confidence score was less 50, then in one version (e.g., Key50) the annotation would be shown as an orphan with the candidate anchor available as a guess, and in the other (e.g., Key0) the annotation would be placed in the document at the candidate anchor.

Unfortunately, asking participants to rate their annotations repeatedly had an unintended side effect. Based on their comments, participants appeared to become fatigued and less concerned about their ratings by the end of the study. While participants' ratings support their comments made during the study, I have placed less emphasis on numerical analysis of the ratings because of this fatigue effect. As discussed in Section 6.3.1, a field study would be valuable to gain more information about user expectations while avoiding rating fatigue.

6.2.2 Study Findings

The 8 participants in the study made 115 annotations, averaging 14.4 annotations per person. The annotations were mostly highlights (76%) with some notes (24%). Based on comments and ratings, the Keyword Anchoring algorithm appears to meet user expectations. Participant comments made while rating new positions found by the keyword algorithm included: "*sentence changed, but computer did a good job of finding whole thing including main point,*" "*good job to pick that out, very similar, a few minor changes with words more concise,*" "*found words that were pretty much the same, so ended up getting the jist of it even though it was very different,"* and *"they did a lot of editing on this one but the computer's guess is exactly what I would have highlighted."*

In contrast, participant comments highlight some issues with the positions found by the simple text search. Comments included: "*there's a lot extra that it missed, word rearrangement stuff,*" "*it should have found more,"* "*it seemed like it would have picked out some of the keywords that exist farther on,*" and "*lost crucial part of the highlight, did not find key words.*"

Comparing participants' ratings of the new anchor positions found by the algorithms also shows the preference for the keyword algorithm. A Friedman test to evaluate the difference in median ratings was significant (χ^2 (3, N =8) = 9.813, p = .02), with a Kendall coefficient of concordance equal to 0.409. The median ratings and some of the significance values from a follow-up pairwise comparison using the Wilcoxon test are shown in Table 6.2. The median rating for positions found by Key0 was significantly higher than both of the simple text algorithm's median ratings at the p=0.05 level, while the median rating for Key50 was significantly higher than the median ratings for the simple text algorithm at the p=0.1 level. While

Table 6.2: Significance values from the Wilcoxon follow-up test to compare median ratings for annotation positions found by the algorithms.
*The median rating for the keyword algorithm with no guess threshold is significantly higher than the two simple text match versions at the p=0.05 level.
+The median rating for the keyword algorithm with the guess threshold at 50 is significantly higher than the simple text match versions at the p=0.1 level.

Algorithms	**Simple50**, Median = 5.5	**Simple0**, Median = 5.0
Key0, Median = 6.5	$p < 0.04^*$	$p < 0.05^*$
Key50, Median = 6.5	$p < 0.07^+$	$p < 0.08^+$

I hesitate to put too much emphasis on the numerical ratings due to a small number of participants and some issues with rating fatigue, the numerical ratings support participant comments and reinforce that participants preferred the keyword algorithm.

Experience with Keyword Algorithm

From repeated participant comments, one simple modification to the Keyword Anchoring algorithm, namely extending anchors to sentence or phrase boundaries, would improve satisfaction. Some example comments include: "*not selecting the whole sentence, it would be nice if it did that,*" "*why didn't it highlight the rest of the sentence,*" and "*should have grabbed stuff just before.*" Some participants also commented on the importance of names in the anchor text they selected, saying "*should have found author's name,*" and "*the computer missed the name.*" Treating proper nouns, any capitalized words within a sentence, or other words with special formatting, as keywords by default even if they occur multiple times would also be a reasonable extension of the algorithm.

One of the initial goals for the study was to develop general guidelines for the appropriate Guess threshold value. I was looking for guidelines that did not depend on the scoring algorithm as currently implemented. For example, whether finding 50% of the anchor's keywords in the modified version meant the new position was likely to meet user expectations. Instead, the user study made apparent the need to add an additional lower threshold in the keyword algorithm and reinforced the importance of user interaction.

When the annotation's anchor text dramatically changed in the modified version, occasionally the keyword algorithm still found a candidate anchor, with a very low confidence score. It quickly became apparent that the low scoring candidate anchors were completely unrelated to the original annotation, particularly when the candidate anchor contained only one keyword. Two participants who experienced this situation commented: *"found only one word, guess was really bad*" and "*only found one word, bad guess.*" Not surprisingly the median rating was 1.0 ("terrible") for the six cases where the keyword algorithm's new anchor contained only one keyword. As discussed previously, due to this experience I added a lower Complete Orphan threshold to the Keyword Anchoring algorithm.

The importance of user interaction with the algorithm was reinforced when I looked at potential values for the Guess threshold. Individual preference for when to place a found anchor in the modified document instead of showing it as an orphan with a guess varied dramatically. Comments included: *"[about the guess option] quite accurate, sooner have them left them in the document,*" and *"[algorithms] should have been picky, last one [Key0] wasn't picky at all and it should have been.*" Other participants expressed a desire to have annotations with lower confidence scores placed directly in the document at the guessed candidate anchors, but shown in a different color. In Section 6.3.2, I discuss options for allowing users to interact and modify the thresholds values.

6.3 Discussion

My experience with the Keyword Anchoring algorithm and user study highlights a number of general issues that robust anchoring methods should address.

6.3.1 User Interface Refinements

Determining the appropriate user experience remains a challenge for robust anchoring algorithms. Most current systems, including [KKP+01, PW00, MOWD], alert the user when an annotation has been orphaned, but do not provide much additional assistance. The user study demonstrated the value of presenting potential anchors as guesses, but to understand fully the best way to help users cope with orphaned annotations, more studies are necessary.

In particular, since orphaned annotations occur as the user attempts to accomplish a task, such as providing feedback, field studies would be very valuable to gather more information on user expectations. Providing assistance to reposition orphaned annotations might be particularly important in some situations, for example, if other users wanted to verify their feedback had been incorporated in the revised document. In other situations, users might want orphaned annotations deleted or easily dealt with using a resolve button as suggested in [CGG00].

As described in Chapter 3, I have extended the prototype annotation interface to support threaded discussions and multiple authors. In the future, it could be deployed in a field study to gather more data.

6.3.2 Setting Threshold Values

One goal of the user study was to gather data to assist in determining appropriate values for the Guess and Complete Orphan thresholds in the Keyword Anchoring algorithm. As discussed in Section 6.2.2, users had very difference preferences when the algorithm should place an annotation in the document at the candidate anchor found or show the annotation as orphaned with a guess. This makes clear that there are no set values that will work for either the Guess or Complete Orphan threshold.

While users should be able to easily modify the Guess and Complete Orphan thresholds to fit their tastes, it may also be valuable for the algorithm to automatically adjust the thresholds. For example, the algorithm could lower the Guess threshold based on the guess the user accepts without modifications and raise the orphan threshold based on which guesses the user completely rejects.

To collect additional data about where to set the default thresholds, future field studies should start initially with a very high Guess threshold so that more annotations are orphaned with guesses and then observe when users accept the guesses as the new position. I believe that starting with a high Guess threshold and presenting the newly found positions for the annotations as suggestions runs less risk of annoying the user then directly placing annotations in the document at candidate positions with low confidence scores. These candidate anchors with low confidence scores, as long as they are above the Complete Orphan threshold, are still very valuable. In the user study, I observed that providing a guess had value even when participants did not necessarily agree with the guess, since it dramatically sped up rating the annotation. In a real situation this could be analogous to users taking advantage of the guess to determine quickly how to handle orphaned annotations. Ideally, the guess would suggest a reasonable starting point for reattaching the annotation or deciding it should be deleted.

6.3.3 Complex Anchoring Situations

The current prototype, as well as most other annotation systems, only allows users to annotate continuous ranges of text. In the user studies, participants have asked for the ability to anchor one comment to multiple pieces of non-contiguous text in the document. This raises numerous user experience questions and is an interesting direction for more research.

How do users specify that they would like a comment anchored to multiple places in the document? How should robust anchoring methods handle this? Are there multiple anchors that are found independently and only linked together for display? Or should the robust positioning algorithm use knowledge about the relationship of the two anchors in the original document? Users might also like to have annotations with more implicit anchors robustly anchored, such as annotations that are drawings or marks made in the margin of the document.

Robust anchoring algorithms perhaps also should to handle instances in which document modifications have split the original anchor text into multiple pieces. This is potentially challenging for the anchoring algorithm, which needs to handle finding two possible locations for an annotation containing different parts of the anchor, and also the user interface that displays the results. Another open question is the best way to show that one annotation anchor in the original document has now been separated in the modified version.

6.4 Conclusion

Online documents are frequently modified, and this can cause annotations to lose the link to their proper position in the document. Keyword Anchoring is a robust anchoring algorithm that was designed based on what users expect to happen to their annotations when the underlying document changes.

Keyword Anchoring uses unique words from the text selected by the user to anchor an annotation, and does not assume cooperation from the document or knowledge of the underlying document structure. Saving keywords makes the anchoring very flexible and extremely robust to document modifications while ignoring document structure allows the algorithm to be used with any document format. Keyword Anchoring could easily be used in conjunction with other robust anchoring methods to provide increased robustness that may better meet user expectations.

The user study of Keyword Anchoring suggested that the algorithm meets user expectations better than a simple text search algorithm and highlighted some improvements that could enhance Keyword Anchoring. The study also provided insight on how systems can present annotations to the user based on the positioning algorithm’s confidence in the location found in the modified version. While Keyword Anchoring is a step toward addressing the robust anchoring problem in a way that meets user expectations, additional work remains. Future studies, particularly field studies, are needed to refine robust positioning algorithms to ensure they meet user needs and determine the appropriate user experience.

Chapter 7

Conclusions and Future Work

This work has explored the use of annotations for asynchronous collaboration. By building software prototypes and deploying them in field and laboratory studies, I have investigated the value of annotations for discussing documents in an educational setting, methods for increasing awareness of annotations, and ways of robustly anchoring annotations to meet user expectations. In this chapter I conclude by revisiting the contributions of my research, described previously in Section 1.4, and outlining areas where the research could be extended.

7.1 Annotations for Asynchronous Discussion

In my field study in a computer science graduate class, I compared the online discussion of technical papers using the WebAnn annotation system to that using EPost, a threaded discussion board. The key difference between the two types of discussions is presence or absence of context. Student comments in the annotation system tied directly to sections of the paper being discussed, while in the discussion board the student comments had to stand on their own. My belief was that context provided by annotations would increase the overall amount and quality of discussion, as well as stimulating a more engaging discussion in-class of issues brought up online. Although students contributed more content using WebAnn, they slightly preferred using the EPost discussion system. I also found that contrary to my expectations, online and in-class discussion sometimes competed with each other, rather than being complementary.

7.1.1. Contributions

The study findings tell a mixed story for annotations. Using annotations, students contributed almost twice as much content using WebAnn and replied more often. The students found that annotations promoted a particular discussion style in which they could easily make comments on specific sections or issues in a paper. However, overall students slightly preferred using EPost, the threaded discussion board, for a variety of reasons, including access issues, amount of work, and the more general discussion style.

The study highlighted a number of issues to consider when deciding what type of online discussion system would be appropriate for a particular task. The type of discussion an instructor wants to promote is critically important. Annotations can be very useful for focused comments on sections of a document, but do not lend themselves to more general discussion of the content (at least in current implementations). Workload should also be a consideration. Using WebAnn was more work for students because they all printed the papers to read them and then had to go back and skim the online version to locate the correct position for their comments. Using EPost students did not need to attach their comments to a particular location in the document. Chapter 3 outlined further process changes that might better integrate annotations in a classroom setting, including allowing students more time to comment, summarizing any online discussion in the classroom, addressing workload by limiting the number of papers discussed online, and reducing discussion overload by limiting the number of students contributing to the online discussion at any one time.

Using the WebAnn system for the study suggested several features that all online annotation systems should include to support asynchronous discussion smoothly. Students wanted to be able to make general comments easily on larger sections or on the entire document. They also stressed the importance of notification when new annotations where made and filtering so they could easily find comments by a particular person.

7.1.2 Future directions

The field study and experience with the WebAnn system highlights a number of directions to explore in future research. There are technical questions, such as how to support more general discussions using annotations, as well as process questions to pursue.

The most prevalent observation by the students was that the two online discussion tools supported very different types of discussions. While WebAnn supported specific discussions using annotations, and EPost excelled at supporting general discussions, students wanted a single tool that could do both. I believe that a number of interface changes to WebAnn, including softening the display of anchors on the text so that students might be willing to select longer sections to comments on, and menu items that explicitly allow commenting on paragraphs or sections, would improve support for general discussions using annotations. Implementing these and perhaps other interface changes to WebAnn and then performing an additional study, would be valuable to discover whether annotations can be used to support both general and specific discussions.

When using the online discussion systems, one surprising finding was the degree to which the online and in-class discussions competed rather than complementing each other. Further investigation into techniques for better integrating the online and in-class discussion would be very valuable. While adding notifications to WebAnn might help students more easily stay aware of online discussions, instructors may still need to make an effort to integrate online and in-class discussion or explicitly decide they will be completely separate. This issue is particularly relevant in large classes, precisely the situations where online discussions might be very valuable, to allow everyone to participate in some type of discussion. In the study, even with only eleven students, they complained of overload trying to keep up with the discussion. In a larger class staying on top of the discussion could be even more difficult, and the gulf between the online and in-class environment could easily widen.

Another area for future research is investigating the ways in which the instructor's role influences or changes the online discussions. In the study, the instructor and TA primarily only read comments without responding, although one guest lecturer did respond to comments made on his papers. Understanding the tradeoffs and influence on the types of discussion that occur based on how the instructor or teaching staff choose to participate in the discussion would be very valuable when educators explore integrating online discussion into their classes.

7.2 Notification of Annotations

Using WebAnn for asynchronous discussion reinforced the importance of making users aware of new annotations. To explore user needs for notifications and to try different notification mechanisms, I conducted a field study of notifications in Microsoft Office Web Discussions with a large software product development group using annotations to discuss software specification documents. I designed and deployed improvements to the existing notifications in Microsoft Office Web Discussions and experimented with notifications using a peripheral awareness system. Participants felt that using the new notifications increased their awareness of annotations on the documents and they considered them an improvement over the existing notifications.

7.2.1 Contributions

Participants in the notifications field study used the new notifications while writing, revising, and reviewing software specification documents. This real life setting provided valuable findings about user needs for notifications. In particular, users felt very strongly about what information the notifications should contain. Their rankings of the most valuable information in the notifications, including the content of new comments, author information, hyperlinks to the comment in context, and the number of new and total comments, will be helpful for designers of future notification systems.

The study highlighted the importance of supporting multiple notifications mechanisms so that participants could select the mechanism (either email or the Sideshow peripheral awareness system) that best met their needs. During the study, participants used the notifications in many different ways, ranging from very active monitoring of comments on a document to casual tracking of new comments. This variety of uses reinforces the importance of making the choice of delivery mechanism and configuration of information in the notifications very easy, so that users can customize the notifications to their needs.

7.2.2 Future directions

My experience in the notifications field study suggests several areas that would be interesting to explore further. In particular, it would be useful to investigate whether additional content could improve the notifications, whether meta-awareness allowing users to see who is subscribed to notifications would be helpful, and whether additional ways to make subscription and configuration of notifications easier would also be helpful.

Experience in the field study highlighted several additional types of information that might be valuable to include in notifications. Most existing notifications, including the ones deployed in the field study, do not include any context information indicating where the new annotations were made in the document. It would be interesting to explore whether including context in the notifications, perhaps several lines from the document around where the new annotation was positioned, would help users in deciding whether or not they needed to visit the document due to the new annotation.

In addition to including context information, participants also suggested other types of information that could help make notifications more valuable. Participants using email notifications suggested that if the new annotation were a reply, the notifications should include the text of the original annotation. They also thought that making a clearer visual distinction between new annotations and replies in the notifications would be helpful. Participants using the Sideshow notifications asked for the ability to filter comments, tickets that summarized new annotations on several specifications, and more noticeable visual changes to a ticket when new comments occurred.

Looking more generally at notification systems, there is usually no way for a user to learn which other users are subscribed to notifications. This lack of meta-awareness about who will be notified can cause problems. For example, if a user adds an important new comment to the discussion, he or she might choose to send an explicit email to member of their group alerting them to the new comment. Any group member already subscribed to notifications on the document will then receive two notifications, one from the system and one from the person who made the comment. One approach would be for the notification system to allow users to choose whether or not their subscription information was publicly available. Future research could explore the best way to present who is subscribed to notifications and whether users find meta-awareness information valuable.

Current notification mechanisms are also typically opt-in, so that the user explicitly subscribes to notifications on a particular document or object, and chooses how and when the notifications will be delivered. While this guarantees that the user is interested in receiving the notifications (or was at one time), this pushes the requirement for managing notification subscriptions and configurations to the user. It is then easy for the user to forget to cancel notifications that are no longer valuable or to forget to subscribe to notifications on documents that could be important.

Another area for future research would be exploring how a notification system could take a more active role in assisting users in managing their notifications, such as simply remembering the previous configuration and delivery setting for the user, suggesting other documents that the user might want to receive notifications for, or automatically subscribing the user to notifications on documents they read regularly. Formally studying the effects of these options will be critical in finding ways to assist users rather than frustrating them.

7.3 Robustly Anchoring Annotations

Systems that support using annotations for asynchronous discussion typically store the annotations outside the documents, so that users can annotate documents they do not have permission to modify. The systems can also then provide access control for the annotations and allow different groups to maintain their own sets of annotations. Unfortunately, storing the annotations separately complicates their display and can cause problems if an annotated document is edited. While many annotation systems include algorithms to cope with positioning annotations in documents that have been modified, there had been no previous research exploring what users expect to happen to their annotations when a document changes.

In a laboratory study, I investigated what users expected to happen to their annotations by asking them to rate their satisfaction with annotations positioned by a very simple algorithm. Then, using the user expectations gathered, I developed and tested Keyword Anchoring, an annotation positioning algorithm designed to meet user expectations.

7.3.1 Contributions

In the laboratory study of user expectations, I found that participants considered some parts of the text they had annotated to be particularly important. When rating their satisfaction with the new positions found for their annotations in the modified document, they focused on how well these “keywords” and phrases that they annotated were found. The study results also suggested that participants paid little attention to the text surrounding their annotations and sometimes might have preferred that the positioning algorithm orphan their annotations rather than place them in the document when only a small amount of the original text associated with the annotation was found.

Based on the study findings, I developed the Keyword Anchoring algorithm. The algorithm positions annotations using unique words in the text annotated by the user. By focusing on document content and tracking the unique words in successive versions of the document, Keyword Anchoring remains independent of the underlying format of the document and requires no cooperation from the document.

7.3.2 Future Directions

Much research remains to be done on how to anchor annotations robustly in a way that meets user expectations. Promising directions for future research include exploring user expectations for a wider variety of annotation types on text documents, integrating the Keyword Anchoring algorithm with existing robust anchoring methods, and exploring robustly anchoring annotations on other media such as images and video.

In my studies I have focused on annotations made during active reading of text documents. It would be valuable to explore whether users have different expectations for annotations made for different tasks, such as providing editing feedback. I also focused on annotations made on ranges of text, since those are typically the most common in online annotation systems. With the increasing support for digital ink, it would be interesting to explore robustly anchoring ink annotations. This may be particularly challenging since ink annotations are not initially explicitly anchored to a particular location in a document.

Keyword Anchoring robustly anchors annotations using keywords from the anchor text annotated by the user. A number of other robust anchoring methods exist, including Robust Locations [PW00] and WebVise [GSØ99], and another direction for future work would be integrating Keyword Anchoring into existing methods. The Robust Locations framework in particular could benefit from being extended to include Keyword Anchoring. Robust Locations uses several different strategies for anchoring, starting initially with character offset (a bookmark), then document structure, and finally using context. Using Keyword Anchoring instead or in addition to the context method could provide Robust Locations with additional guidance that may better meet user expectations when the annotated document has been changed.

Exploring user expectations for robustly anchoring annotations to video and images is another promising direction for future work. The Keyword Anchoring approach may extend to these media types very naturally. For example, while a simple image anchoring algorithm might use the x, y position of the user's annotation in the image, a "keyword" approach could save anchor information about particular features or objects at the location in the image annotated by the user. The positioning algorithm could then attempt to locate the feature or object in the modified image, and would be robust in the presence of common image modifications such as cropping and scaling. For video, the keyword approach corresponds even more directly to the notion of key frames. Selecting the key frames from the section of video annotated by the user could help robustly anchor an annotation to the video.

While not designed for handling changes, the approach of the MAVIS project [LDG+96], an extension of the Microcosm hypermedia system for implementing generic links from non-text objects, such as images and video, could be used to support robust anchoring. In the Microcosm architecture, the anchor for a generic link is defined once, and then the system adds the link anywhere in any document where the anchor occurs. To locate anchor points for the non-text generic links, MAVIS used fuzzy matching; that is, the original defined anchor only needed to be similar to the original anchor. MAVIS supported different methods for computing similarity, including rotation, scaling, and translation of images. These similarity methods could be used for robustly anchoring annotations to images and videos that have changed.

7.4 Concluding remarks

Annotations are a natural tool for asynchronous collaboration. By supporting commenting in the context of a document or other media object, annotations allow users to easily record and share thoughts and opinions. As the number of documents, images, and videos available continues to grow, so will our desire to annotate them. While this work has shown the potential for annotations to support discussions, methods of increasing awareness of annotations, and has introduced an algorithm for robustly anchoring annotations to meet user expectations, continued research to explore how to best support asynchronous collaboration using annotations is critically important.

Bibliography

[APK+99] Gregory D. Abowd, Maria da Graca Pimetel, Bolot Kerimbaev, Yoshihide Ishiguro, and Mark Guzdial. Anchoring Discussions in Lecture: An Approach to Collaboratively Extending Classroom Digital Media. In *Proceedings of CSCL 1999*, pages 11-19. Palo Alto, CA, December 1999.

[Ado] Adobe Acrobat. http://www.adobe.com/acrofamily/main.html

[Ama] Amaya Browser. http://www.w3.org/Amaya/

[Ann] Annotation Engine. http://cyber.law.harvard.edu/projects/annotate.html

[Anno] Annotator. http://www-hbp.usc.edu/Projects/annotati.htm

[Annot] Annotea. http://www.w3.org/2001/Annotea/

[App01] Wolfgang Appelt. What groupware functionality do users really use? Analysis of the usage of the BSCW system. In *Proceedings of the 9th Euromicro Workshop on PDP*. http://bscw.gmd.de/Papers/PDP2001/PDP2001.pdf

[App99] Wolfgang Appelt. WWW Based Collaboration with the BSCW System, In *Proceedings of SOFSEM 1999*, Springer Lecture Notes in Computer Science 1725, pages 66-78. http://bscw.gmd.de/Papers/SOFSEM99/sofsem.pdf

[Ash00] Helen Ashman. Electronic Document Addressing: Dealing with Change, *ACM Computing Surveys*, 32(3): 201-212, September 2000.

[BNP+93] Ronald M. Baecker, Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby. The User-centered Iterative Design of Collaborative Writing Software. In *Proceedings of INTERCHI 1993*, pages 399-405. Amsterdam, The Netherlands, April 1993.

[BG01] David Bargeron and Anoop Gupta. A Common Annotation Framework. Microsoft Tech. Report, MSR-TR-2001-108, 2001. http://www.research.microsoft.com/scripts/pubs/view.asp?TR_ID=MSR-TR-2001-108

[BGG+01] David Bargeron, Anoop Gupta, Jonathan Grudin, Elizabeth Sanocki, and Francis Li. Asynchronous collaboration around multimedia and its application to on-demand training. In *Proceedings of HICSS-34 Conference*. http://www.research.microsoft.com/research/coet/MRAS/HICSS34/paper.pdf

[BHB+97] Sandford Bessler, Michael Hager, Hartmut Benz, Rolf Mecklenburg, and Steffen Fischer. DIANE: A Multimedia Annotation System, In *Proceeding of ECMAST 1997,* Milan, Italy, May 1997.

[BB01] A.J. Brush and David Bargeron, Robustly Anchoring Annotations Using Keywords, Microsoft Technical Report, MSR-TR-2001-107, 2001. http://www.research.microsoft.com/scripts/pubs/view.asp?TR_ID=MSR-TR-2001-107

[BBG+01] A.J. Brush, David Bargeron, Anoop Gupta, and JJ Cadiz. Robust Annotation Positioning in Digital Documents. In *Proceedings of CHI 2001*, pages 285-292, Seattle, WA, March 2001.

[BBG+02a] A.J. Brush, David Bargeron, Jonathan Grudin, Alan Borning, and Anoop Gupta. Supporting Interaction Outside of Class, In *Proceedings of CSCL 2002,* pages 425-434, Boulder, CO, January 2002.

[BBG+02b] A.J. Brush, David Bargeron, Jonathan Grudin, and Anoop Gupta. Notification for Shared Annotation of Digital Documents. In *Proceedings of CHI 2002*, pages 89-96, Minneapolis, MN, April 2002.

[BB91] Christine V. Bullen and John L. Bennett. Groupware in Practice: An Interpretation of Work Experiences, 1991. Reprinted in Baecker, R.M. (Ed.) *Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann, pages 69-84, 1993.

[CGG00] JJ Cadiz, Anoop Gupta, and Jonathan Grudin. Using Web Annotations for Asynchronous Collaboration Around Documents. In *Proceedings of CSCW 2000*, pages 309-318, Philadelphia, PA, December 2000.

[CVJ01] JJ Cadiz, Gina Venolia, Gavin Jancke, and Anoop Gupta. Sideshow: Providing Peripheral Awareness of People and Information, Microsoft Technical Report 2001-83. http://research.microsoft.com/scripts/pubs/view.asp?TR_ID=MSR-TR-2001-83

[CDH+95] Leslie Carr, David De Roure, Wendy Hall, and Gary Hill. The Distributed Link Service: A Tool for Publishers, Authors and Readers, In *Proceedings of the Fourth International World Wide Web Conference*, Boston, MA, December 1995.

[CHD99] Leslie Carr, Wendy Hall, and David De Roure. The Evolution of Hypertext Link Services, *ACM Computing Surveys*, Vol. 31, Number 4es, December 1999.

[Col] Col•laboració. http://www.nada.kth.se/~henrry/Collaboracio.html

[CoN] CoNote, Tools for Collaborative Leaning. http://www.cs.cornell.edu/home/dph/annotation/annotations.html

[Cri] Crit.org. http://www.crit.org

[CTG93] CTGV, Anchored Instruction and situated cognition revisited. *Educational Technology*, 33(3) pages 52-70, 1993.

[Dav98] Hugh Davis. Referential Integrity of Links in Open Hypermedia Systems, In *Proceedings of HyperText 1998*, pages 207-216, Pittsburgh, PA, June 1998.

[Dav99] Hugh Davis. Hypertext Link Integrity, *ACM Computing Surveys*, Vol. 31, Number 4es, December 1999.

[DKH94] Hugh Davis, Simon Knight, and Wendy Hall. Light Hypermedia Link Services: A Study of Third Party Application Integration. In *Proceedings of the 1994 ACM European Conference on Hypermedia Technology*, pages 41-50, September 1994.

[DH95a] James R. Davis and Daniel P. Huttenlocher. CoNote System Overview. 1995. http://www.cs.cornell.edu/home/dph/annotation/annotations.html

[DH95b] James R. Davis and Daniel P. Huttenlocher. Shared Annotation for Cooperative Learning. In *Proceedings of CSCL 1995*, pages 84-88, October 1995.

[DLC+] Richard C. Davis, James A. Landay, Victor Chen, Jonathan Huang, Rebecca B. Lee, Francis C. Li, James Lin, Charles B. Morrey III, Ben Schleimer, Morgan N. Price, and Bill N. Schilit. NotePals: Lightweight Note Sharing by the Group for the Group. In *Proceedings of CHI 1999*, pages 338-345, Pittsburgh, PA, May 1999.

[DV00] Laurent Denoue and Laurence Vignollet. An annotation tool for Web browsers and its applications to information retrieval. In *Proceedings of RIAO 2000, Sixth Conference on Content Based Multimedia Information Access*, Paris, France, April 2000. http://133.23.229.11/~ysuzuki/Proceedingsall/RIAO2000/Wednesday/16AP1.pdf

[DB92] Paul Dourish and Victoria Bellotti. Awareness and Coordination in Shared Workspaces. In *Proceedings of CSCW 1992*, pages 107-114, November 1992.

[EM] Valerie J. Easton and John H. McColl, Statistics Glossary v1.1, HTML editing by Ian Jackson, http://www.cas.lancs.ac.uk/glossary_v1.1/main.html

[Epos] EPost. UW Catalyst Toolkit, http://catalyst.washington.edu/tools/

[EQu] E-Quill. http://www.e-quill.com/ *Acquired by Microsoft in 2001

[Fuc99] Ludwin Fuchs. AREA: A Cross-Application Notification Service for Groupware. In *Proceedings of ECSCW 1999*, pages 61-80, 1999.

[Gar99] Jon Garfunkel. Web Annotation Technologies, 1999. http://ps.pageseeder.com/ps/ps/papers/annot/jongar/jongar.pshtml

[GSM+99] Geri Gay, Amanda Sturgill, Wendy Martin, and Daniel Huttenlocher. Document-centered Peer Collaborations: An Exploration of the Educational Uses of Networked Communication Technologies. *Journal of Computer-Mediated Communication*, 4(3), March 1999. http://www.ascusc.org/jcmc/vol4/issue3/gay.html

[GSØ99] Kaj Grønbæk, Lennert Sloth, and Peter Ørbæk. Webvise: Browser and Proxy Support for Open Hypermedia Structuring Mechanisms on the WWW, In *Proceedings of the Eighth International World Wide Web Conference*, pages 253-267. Toronto, Canada, May 1999.

[GRG96] Carl Gutwin, Mark Roseman, and Saul Greenberg. A Usability Study of Awareness Widgets in a Shared Workspace Groupware System. In *Proceedings of CSCW 1996*, pages 258-267, Cambridge, MA, November 1996.

[Guz97] Mark Guzdial. Information Ecology of Collaborations in Educational Setting: Influence of Tool. In *Proceedings of CSCL 1997*, pages 83-90, December 1997.

[GRK00] Mark Guzdial, Jochen Rick, and Bolot Kerimbaaev. Recognizing and Supporting Roles in CSCW, In *Proceedings of CSCW 2000*, pages 261-268, December 2000.

[GT00] Mark Guzdial and Jennifer Turns. Effective discussion through a computer-mediated anchored forum. *Journal of the Learning Sciences*, 9(4) pages 437-470, 2000.

[HHD] Wendy Hall, Gary Hill, and Hugh Davis. The Microcosm Link Service. In *Proceedings of Hypertext 1993*, pages 256-259, November 1993.

[HLO99] Rachel M. Heck, Sarah M. Luebke, and Chad H. Obermark. A Survey of Web Annotation Systems, http://www.math.grin.edu/~rebelsky/Research/Summer1999/Papers/survey_paper.html

[Hen] Charles Hendricksen. Asynchronous Collaboration by Document Review, http://students.washington.edu/veritas/papers/DRpaper/basedoc.html

[HT99] Jim Hewitt and Chris Teplovs. An Analysis of Growth Patterns in Computer Conferencing Threads. In *Proceedings of CSCL 1999*, pages 232-241. Palo Alto, CA, December 1999.

[HGT97] Cindy E. Hmelo, Mark Guzdial, and Jennifer Turns, Computer-support for collaborative learning: Learning to make it work. American Educational Research Association 1997 Annual Meeting, Chicago, Illinois, March 1997. http://guzdial.cc.gatech.edu/papers/aera97/mbl.html

[Int] Intraspect. Enterprise Collaboration Management with Intraspect, http://www.intraspect.com/

[IMa] IMarkup. http://www.imarkup.com

[LB97] Daniel Laliberte and Alan Braverman. A Protocol for Scalable Group and Public Annotations. *1997 NCSA Technical Proposal*, http://www.hypernews.org/~liberte/www/ scalable-annotations.html

[LS93] Darryl Lawton and Ian E. Smith. The Knowledge Weasel Hypermedia Annotation System, In *Proceedings of Hypertext 1993,* pages 106-117, November 1993.

[LFK88] Mary D. P. Leland, Robert S. Fish, and Robert E. Kraut. Collaborative Document Production Using Quilt. In *Proceedings of CSCW 1988*, pages 206-215, September 1988.

[LDG+96] Paul H. Lewis, Hugh C. Davis, Steve R. Griffiths, Wendy Hall, and Rob J. Wilkins. Media-based Navigation with Generic Links, In *Proceedings of Hypertext 1996*, pages 215-223, Washington, DC, March 1996.

[Liv] Livelink. Functional Assessment of Open Text Livelink. http://www.opentext.com/livelink/

[Lot] Lotus QuickPlace Release 2.0 Reviewer's Guide. http://www.lotus.com/home.nsf/welcome/quickplace

[KKP+01] José Kahan, Marja-Riitta Koivunen, Eric Prud'Hommeaux, and Ralph R. Swick. Annotea: An Open RDF Infrastructure for Shared Web Annotations, In *Proceedings of the Tenth World Wide Web Conference,* pages 623-632, Hong Kong, May 2001. http://www.w3.org/2001/Annotea/Papers/www10/annotea-www10.html

[KA98] Thomas Koch and Wolfgang Appelt. Beyond Web Technology- Lessons Learnt from BSCW. In *Proceedings of the 1998 IEEE WEI ICE Workshop*, http://bscw.gmd.de/Papers/wetice98/index.html

[Mar97] Catherine C. Marshall. Annotation: from paper books to the digital library, In *Proceedings of Digital Libraries 1997*, pages 131-140, Philadelphia, PA, July 1997.

[Mar98] Catherine C. Marshall. Toward an ecology of hypertext annotation, In *Proceedings of HyperText 1998*, pages 40-48, Pittsburgh, PA, June 1998.

[MB02] Catherine C. Marshall and A.J. Brush. From Personal to Shared Annotations, In *Proceedings of CHI 2002*, pages 812-813, Minneapolis, MN, April 2002.

[MPG99] Catherine C. Marshall, Morgan N. Price, Gene Golovchinsky, and Bill N. Schilit. Introducing a digital library reading appliance into a reading group, In *Proceedings of Digital Libraries 1999*, pages 77-84, Berkeley, CA, August 1999.

[MER] Microsoft Ebook Reader. http://www.microsoft.com/reader/

[MIEB] Microsoft Internet Explorer, Creating Custom Explorer Bars, Tool Bands and Desk Bands, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/Shell/programmersguide/shell_adv/bands.asp

[MOWD] Microsoft Office 2000 Web Discussions. http://office.microsoft.com/assistance/2000/wWebDiscussions.aspx

[MWo] Microsoft Word. http://www.microsoft.com/office/word/using.htm

[NKC+94] Christine M. Neuwirth, David S. Kaufer, Ravinder Chandhok, and James H. Morris. Computer Support for Distributed Collaborative Writing: Defining Parameters of Interaction, In *Proceedings of CSCW 1994*, pages 145-152, Chapel Hill, NC, October 1994.

[NKC90] Christine M. Neuwirth, David S. Kaufer, Ravinder Chandhok, and James H. Morris. Issues in the Design of Computer Support for Co-authoring and Commenting. In *Proceedings of CSCW 1990,* pages 183-195, Los Angeles, CA, October 1990.

[Not] NotePals. http://guir.berkeley.edu/projects/notepals/

[OS97] Kenton O'Hara and Abigail Sellen. A Comparison of Reading Paper and On-Line Documents, In *Proceedings of CHI 1997*, pages 335-342, Atlanta, GA, March 1997.

[OEG+95] D. Kevin O'Neill, Daniel C. Edelson, Louis M. Gomez, and Laura D'Amico. Learning to Weave Collaborative Hypermedia into Classroom Practice. In *Proceedings of CSCL 1995*, pages 255-258, Bloomington, IN, October 1995. http://www.covis.nwu.edu/info/papers/pdf/oneill-cscl-95.pdf

[OAM99] Ilia A. Ovsiannikov, Michael A. Arbib, and Thomas H. McNeill. Annotation Technology, International Journal of Human Computer Studies, 50, pages 329-362, 1999.

[Pag] Pageseeder. http://ps.pageseeder.com/ps/topic/ps

[PW01] Thomas A. Phelps and Robert Wilensky. The Multivalent Browser: A Platform for New Ideas, In *Proceedings of Document Engineering 2001*, Atlanta, GA, November 2001.

[PW97] Thomas A. Phelps and Robert Wilensky. Multivalent Annotations, In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries,* Pisa Italy, September 1997. http://www.cs.berkeley.edu/~phelps/Multivalent/papers/edl97-abstract.html

[PW98] Thomas A. Phelps and Robert Wilensky. Multivalent Documents: A New Model for Digital Documents, Technical Report, CSD-98-999, University of California, Berkeley, 1998.

[PW00] Thomas A. Phelps and Robert Wilensky. Robust Intra-document Locations, In *Proceedings of the Ninth World Wide Web Conference*, Amsterdam, May 2000. http://www9.org/w9cdrom/312/312.html

[Phi02] Patrick Phillippot. Where Did I Read That?, PC Magazine (online), April 9, 2002 http://www.pcmag.com/print_article/0,3048,a=24007,00.asp,

[RDF] Resource Description Framework, http://www.w3.org/RDF/

[RMW97] Martin Röscheisen, Christian Mogensen, and Terry Winograd. Shared Web Annotations as a Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples, Technical Report CSDTR/DLTR, Stanford University, 1997.

[SMB96] Matthew A. Schickler, Murray S. Mazer, and Charles Brooks. Pan-Browser Support for Annotations and Other Meta Information on the World Wide Web. In *Proceedings of the Fifth International World Wide Web Conference*, Paris, France, May 1996.

[SPG98] Bill N. Schilit, Morgan N. Price, and Gene Golovchinsky. Digital Library Information Appliances, In *Proceedings of Digital Libraries 1998*, pages 217-226, Pittsburgh, PA, June 1998.

[SGP98] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations, In *Proceedings of CHI 1998*, pages 249-256, Los Angeles, CA, April 1998.

[Thi] ThirdVoice web annotations. http://www.thirdvoice.com/

[VCH+94] Adrian J. Vanzyl, Branko Cesnik, Ian Heath, and Hugh Davis. Open Hypertext Systems, An Examination of Requirements, and Analysis of Implementation Strategies, comparing Microcosm, HyperTED, and the World Wide Web. http://wwwlinf-wiss.uni-konstanz.de/Res/openhypermedia.html

[Ver00] Janet Verbyla. Unlinking the Link, *ACM Computing Surveys*, Vol. 31. Number 4es, December 1999.

[WJ94] Dadong Wan and Philip M. Johnson. Computer Supported Collaborative Learning Using CLARE: The Approach and Experimental Findings. In *Proceedings of CSCW 1994*, pages 187-198, Chapel Hill, North Carolina, October 1994.

[WSS97] Lynn D. Wilcox, Bill N. Schilit, and Nitin “Nick” Sawhney. Dynomite: A Dynamically Organized Ink and Audio Notebook, In *Proceedings of CHI 1997*, pages 186-193, Atlanta, Georgia, March 1997.

[WNB98] Patricia G. Wojahn, Christine M. Neuwirth, and Barbara Bullock. Effects of Interfaces for Annotation on Communication in a Collaborative Task, In *Proceedings of CHI 1998*, pages 456-463, Los Angeles, CA, April 1998.

[Wol00] Joanna L. Wolfe. Effects of Annotations on Student Readers and Writers, In *Proceedings of Digital Libraries 2000*, pages 19 – 26, San Antonio, TX, June 2000.

[XLi] XML Linking Language. http://www.w3.org/XML/Linking

[XPL] XML Pointer Language. http://www.w3.org/TR/xptr/

[Yee] Ka-Ping Yee. Draft RFC for text-search fragment identifiers used in CritLink, http://crit.org/http://crit.org/draft-yee-url-textsearch-00.txt

[ZCM98] Polle T. Zellweger, Bay-Wei Chang, and Jock Mackinlay. Fluid Links for Informed and Incremental Link Transitions, In *Proceedings of HyperText 1998*, pages 50-57, Pittsburgh, PA, June 1998.

Vita

Alice Jane (A.J.) Bernheim Brush has a B.A. degree from Williams College, in Williamstown, MA and a M.S. degree from the University of Washington in Seattle, WA. She has previously worked as a software developer for Tripos, Inc in St. Louis, MO, and interned at WRQ, Microsoft, and Microsoft Research in Seattle, WA. Her main research interest is human computer interaction with a focus on computer supported collaborative work.