

Notable: At the Intersection of Annotations and Handheld Technology

Michelle Baldonado¹, Steve Cousins¹, Jacek Gwizdka², and Andreas Paepcke³

¹ Xerox PARC, 3333 Coyote Hill Road
Palo Alto, CA 94304 USA

{michelle, cousins}@parc.xerox.com

² University of Toronto, 5 King's College Road
Toronto, Ontario M5S 3G8 Canada
jgwizdka@acm.org

³Digital Libraries Lab (InfoLab), Stanford University
Stanford, CA 94305 USA
paepcke@db.stanford.edu

Abstract. The Notable annotation system enables users to annotate paper documents using handheld devices in a mobile environment. This paper describes the design issues and solutions that arose in creating Notable, with a particular focus on design challenges at the intersection of annotations and handheld technology. Novel design strategies include separating the annotation writing platform from the document viewing platform, providing search as the method for document selection, offering context-sensitive phrase completion and icon-based graphical pinning for fine-granularity annotation anchoring, and including some support for coordinating group annotation activity.

1 Introduction

The Post-It™ note celebrates its twentieth birthday this year. The ubiquitous yellow sticky squares of paper can be found attached to a variety of objects, from paper documents to computer monitors. Among other reasons, people find that Post-Its™ are useful because they can be attached easily to physical objects (and subsequently removed), because they are small, and because they are highly visible. Yet the notes are not perfect—they can be hard to share with multiple people (photocopies are often problematic because the notes may obscure underlying text), they are difficult to attach to specific locations within documents, and the notes are easy to lose.

Using a Post-It™ note to attach a comment to another object is one example of annotation, an activity that takes many forms [1]. Examples range from highlighting words in a text to writing comments in the margins of a paper to assigning metadata to a literary work. More formally, we define an annotation as follows:

Definition of annotation: An annotation is a commentary on an object that:

- the annotator intends to be perceptually distinguishable from the object itself; and
- the reader interprets as perceptually distinguishable from the object itself.

Our observation that the screen size of the widely available Palm™ handheld device is about the same size as a Post-It™ note was one of the inspirations for creating Notable, a handheld-based system for annotation. We hypothesized that a handheld-based analog to the Post-It™ note would be valuable because it would afford sharing and finding annotations after their initial creation, while keeping intact the desirable size and portability features of the Post-It™. Currently, one of the most popular techniques for sharing handwritten annotations is to rewrite them in another medium (e.g., write comments in the margins of a paper, then summarize those comments in an e-mail message). Our goal was to eliminate the need for this intermediate step. While rewriting can often be of value during a writer's sensemaking process, it requires additional time and effort. Copiers make sharing without rewriting plausible, but few office workers have a personal copier; copiers are typically available in a shared common space rather than in an individual office or cubicle. Furthermore, copiers produce paper documents that must be distributed physically. Scanners and digital cameras offer more streamlined approaches to sharing handwritten comments, but the state-of-the-art of optical character recognition is still not advanced enough to make searching for handwritten comments reliable.

Digital annotation systems (see Sect. 5 for a brief review of annotation systems) have already proven useful for sharing annotations, but they typically require that the user sit at a desktop computer or use a laptop. Although desktop computers and laptops offer increased screen real estate, network capabilities, and processing power in comparison to handheld devices, they fall short of handhelds in portability and startup time. We believe that a small device that is easy to carry and “ready to hand” is likely to be a good fit for mobile work practices.

In this paper, we focus on the design issues and solutions encountered in designing a handheld-based annotation system. We begin with a usage scenario in Sect. 2, which shows how our myriad design decisions come together in a working system. Sect. 3 examines several specific design issues that arise in creating annotations, linking annotations and documents, and viewing and sharing annotations. Sect. 4 gives a brief overview of the Notable system architecture. Sect. 5 relates Notable to previous work. Finally, Sect. 6 summarizes the features of Notable and outlines some possibilities for future work.

2 Scenario

This section describes a scenario that shows how the Notable system might be used in practice. Historically, this scenario was a key element in the Notable design process. The scenario centers around a researcher who needs to review a journal submission. She has the document in paper form, and has also scanned the document into a searchable repository to create a digital identity for the document. Alternatively, she might have the document in digital form already, in which case she is likely to print it out for later reading.

In either case, before the reviewer leaves her office for the day, she uses a desktop application called NoteDesk (Fig. 1) to locate the article's digital handle via a search that specifies salient keywords for the article. Once the handle has been located,

NoteDesk creates a digital surrogate of the document, which contains sufficient information about the document to allow annotations to be attached to specific locations within the article. The surrogate is transferred to the reviewer's handheld device when she synchronizes it with her desktop computer as her last step before heading home.

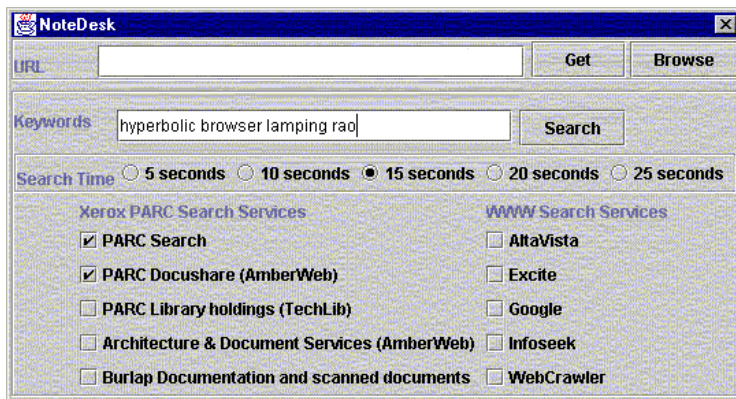


Fig. 1. Using NoteDesk (a desktop application) to search for an object's digital surrogate, which will be downloaded to the user's handheld device at synchronization time—thus enabling annotation

In the evening, she reads the paper copy of the document and makes corresponding digital annotations on her handheld (Fig. 2). For each annotation, she can specify whether it applies to the entire document (Option 1), to a particular region on a page (Option 2), or to a specific phrase in the document (Option 3). For region-level anchoring, she needs only to specify the page number and indicate on a graphical icon where the annotation belongs. For phrase-level anchoring, she uses context-sensitive phrase completion to identify the specific phrase with minimal text entry. Back at the office the next day, the annotations she has made are transmitted to a shared document repository (at synchronization time), and linked to the document. From her desktop computer, she can now access her annotations for inclusion in her final review of the article. For example, she can cut-and-paste annotations into her word processor.

In the design process for Notable, we have explored several variants on this scenario, including annotating for personal sensemaking and note taking on specific document types (e.g. calendars, agendas, etc.). We have also explored the possibility of annotating objects other than paper-based documents, such as museum objects and devices that require regular maintenance (e.g. office machines). More abstractly, our approach generalizes to almost any scenario in which a user needs to assign metadata to a physical object (with a digital surrogate) in a mobile context. Assigning grades to school essays, ratings to movies, and captions for pictures are all possible scenarios that Notable could support.

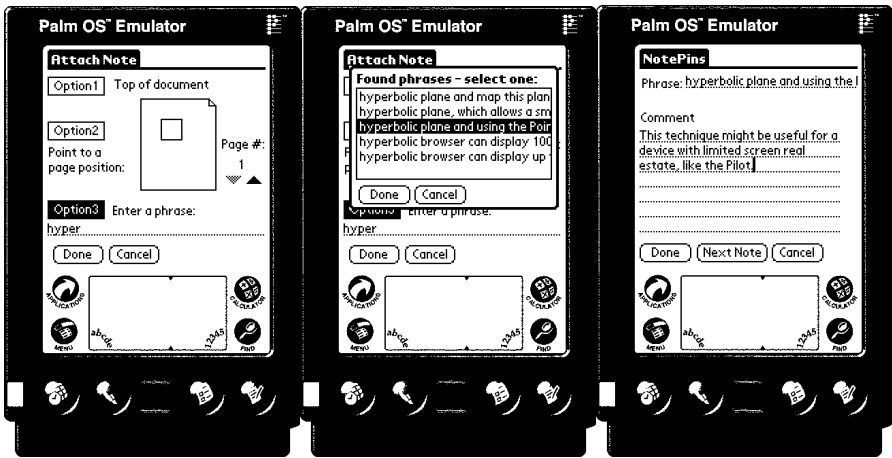


Fig. 2. Annotating a document with Notable: (a) choosing the level of granularity for the annotation; (b) attaching the annotation to a particular phrase; and (c) composing the annotation

3 Design Issues

Historically, many design challenges have lurked at the intersection of a task and a technology. Examples of fruitful intersections include financial modeling and personal computers, personal information management and handheld devices, and person-to-person communication and networked computers. In this section, we outline the design problems and solutions that arise in creating annotations, linking annotations and documents, and viewing and sharing annotations on the interactive platforms that comprise Notable—paper, handheld device, and desktop computer (the latter is optional if the handheld device has a wireless modem).

Throughout this section, we appeal to a framework that represents annotation systems as five interrelated components [2]: annotation writing platform (the platform on which annotations are created), annotations (the commentary), annotation target (the object to which the annotation applies), annotation correspondence (the link between the annotation and the annotated object), and annotation reading platform (the platform on which the annotation is read).

3.1 Creating Annotations

A Post-It™ note is authored on a piece of paper that is separate from the object to which it becomes attached. Our decision in Notable to separate the document reading platform (paper) from the annotation writing platform (the handheld device) owes much to our observations of how Post-It™ notes are used. Given the complementary nature of paper and handhelds, decoupling reading and writing is of particular value for a handheld-based annotation system. On the one hand, paper is a good medium for reading because it supports quick navigation, flexible spatial layout, and inline

annotation (though as we have already observed, readers of paper documents often also create annotations on separate pieces of paper, including Post-It™ notes) [3]. Furthermore, most authors prepare their work for the familiar 8.5"x11" or A4 format, and make graphical decisions based on this assumption. For example, figures are often designed to make maximal use of the paper's width. A paper version of a document is thus perhaps more likely to reflect the author's intended presentation. On the other hand, handheld devices promise to form a good medium for annotation creation because of their connectivity (either via modems or via synchronization with a desktop computer), which facilitates annotation sharing and retrieval. Furthermore, this situation is likely to improve as advances are made in wireless connectivity.

However, connectivity alone does not ensure that users will be able to find the right annotation at the right time. The original Post-It™ model suggests that an annotation can only be found when its parent document is found. In Notable, we require that annotations be entered using Graffiti™, which means that the individual characters and words comprising an annotation are fully recognizable by the system. The consequence of this decision is that annotations can be located by keyword searches that refer either to the annotation text or to the parent document text. If annotations were instead entered in digital ink, finding annotations on their own would be more difficult (given the current state of handwriting recognition). On the other hand, digital ink affords easier annotation entry (text entry on handheld devices can be awkward) and allows for the possibility of annotation drawings. Digital ink also suggests the possibility of a gesture-driven annotation interface. Accordingly, a next generation version of Notable might allow users to combine these two methods of annotation entry.

Although Notable is designed under the assumption that users will prefer to read documents on paper, we expect that there will be some occasions when it is satisfactory to read a document on the handheld device itself. Accordingly, in cases where the text of the document is available in digital form, we allow the user to read the document directly on the handheld device if desired. We accomplish this for a variety of document formats by invoking a text conversion service (available via an online service developed in another research project in our lab) at the time the user selects a document for annotation. The plain text version of the selected document is downloaded to the handheld for viewing.

3.2 Linking Annotations and Documents

Given that the document reading and annotation writing platforms are separated in Notable, a digital equivalent for "stickiness" is critical to the design. Notable enables the user to link an annotation and an object by storing together in an annotation repository the text of the annotation and the digital identifier of the object—thus virtually attaching the annotation and object to each other. The indirect nature of this annotation correspondence can be exploited in cases where an annotation refers to multiple targets. For example, a reader might want to create an annotation that briefly compares the main thrusts of two related documents. Annotation systems that require inline, in situ annotations make this style of annotation difficult. In contrast, although cross-target annotation is not a supported feature in our current implementation of

Notable, our choice of annotation correspondence mechanism allows for this feature in the future.

Before annotation, Notable users perform searches for the digital surrogates of the objects to be annotated. Associated with each surrogate is a digital identifier—the piece of information that the system uses to associate annotations and their targets. In our first iteration on the Notable design [4], search took place directly on the handheld device, which was supplemented with a wireless data modem. After finishing the initial implementation, we reviewed this choice and concluded that it was important to support handheld owners without wireless modems as well. Accordingly, we developed a second version, in which search took place on the desktop and document identifiers were propagated to the handheld via synchronization. In both versions, search is by keywords, though a later version could support fielded search, in which a user can specify additional bibliographic metadata constraints, e.g., author's name.

More specifically, we use URLs (uniform resource locators) as our document identifiers. URLs are also used to link annotations and documents in Web annotation systems, e.g. ComMentor [5] and ThirdVoice™. URLs can be located easily by interacting with search facilities available from numerous sources (Web search engines, digital libraries, intranet searching tools, etc.). In addition, URLs are type-independent; Notable users can annotate any document with a URL, regardless of its format or its access restrictions. In addition, URLs can refer to the digital surrogate of a coffeepot just as well as to the digital surrogate of a document. Accordingly, we can envision future versions of Notable that allow users to annotate sculptures, office machines, or places.

One drawback of URLs is that they impose a new constraint on our system: each target needs to have an established URL before it can be annotated. Thus, if a user wishes to annotate an object that is not locatable by our search engines, a URL for the object must be established and it must be registered in a known document repository. For paper documents, we have established an additional requirement: they must be scanned so that fine-grained annotations can be supported. In many of our scenarios, we expect that the documents will already be available online, have URLs, and be easy to find with a simple search. Nevertheless, this strategy of asking users to scan in documents and register them in a repository allows us to realize the scenario of Sect. 2. A further potential problem with URLs is that they may change or become obsolete. We address this problem to some extent by copying the target data into a local repository. Widespread use of URNs would allow us to solve this problem more satisfactorily.

Of course, URLs alone allow only for target-level annotation. In the first iteration of the Notable design, we felt that target-level annotation was sufficient. We were inspired by the analogy to Post-Its™ and influenced by a study [6] showing that independent annotations, such as comments on a paper left on voice mail, tend to be higher-level comments than those written directly on a paper. However, all of the potential users who saw the system asked for finer granularity anchoring. We responded by deciding to support both phrase-level and region-level annotation anchoring. We quickly found that the separation of annotation writing platform from document viewing platform, the lack of screen real estate on the handheld device, the limited computational power (speed and memory) on the handheld, the time required

for text entry on the handheld, and the high latency of the network link coincided to make the addition of fine-grain annotation a serious design challenge.

For digital documents, the supported level of granularity for annotations is usually intertwined with the governing document model. Structurally, annotations might be at a chapter level, section level, or paragraph level. Graphically, annotations might be at a page level or region level. Textually, annotations might be at a sentence level, phrase level, or word level. Variations abound. Given that our users were creating their annotations in a medium separate from the one in which they were viewing the documents, we needed techniques for the users to specify in more detail where the annotation should be attached regardless of the underlying digital document model. For inspiration, we looked at how users typically respond to this challenge in non-handheld environments. The most influential example we found was in e-mail—in particular, e-mail conventions for providing comments on articles. Typically, these comments include textual meta-information about where the annotation belongs. Fig. 3 is an excerpt from an e-mail message employing this approach.

** page 3, col 2, "By architecting our constructor tool...": Maybe this sentence (or paragraph) should be moved earlier; this makes it clear that we are talking about our approach...

** page 4, Figure 4 would not print on my printer, but I assume it prints elsewhere. In any case, Fig 4 seems to have too much "stuff" and I am not even sure it will print OK in black and white. If we keep it we need to add explanation about what the different patterns mean.

Fig. 3. An example of e-mail conventions in article reviewing

Note that the author of these comments is using two different techniques to “anchor” the annotation. He anchors the first comment by pinpointing the phrase to which it applies. He helps the reader to locate that phrase more quickly by identifying the page and region in which the phrase occurs. He anchors the second comment solely by identifying the page and region to which the annotation applies.

A direct translation of these techniques to the handheld is possible, but suffers from two problems. First, the user needs to employ a standard vocabulary for the anchoring meta-information (e.g., “** page 3, col 2”) in order for the system to anchor the annotations accurately (the references must undergo computer interpretation rather than human interpretation). In a handheld environment, it is tricky to convey such a vocabulary to the user. Second, the user needs to enter a fair amount of text in order to specify the location fully, which is a high cost for a short annotation. On a handheld device with awkward text input, the technique is error prone and unacceptably demanding of the user. Accordingly, we have developed two specialized techniques for anchoring annotations to fine granularity locations in documents: context-sensitive phrase completion and icon-based graphical pinning. These two techniques can be combined to further decrease the input burden on the user.

Employing the first technique, a user anchors an annotation by beginning to write the phrase to which the annotation applies. After several characters have been entered, the system displays a pop-up menu with a list of the phrases in the document that begin with those characters. The user can choose the appropriate phrase for anchoring and proceed to create the annotation. The phrase list is of a manageable size due to

our dual-pronged strategy of: 1) requiring the user to enter several characters before presenting matching phrases in the document, and 2) presenting only phrases that appear in the document (rather than phrases that are drawn from a dictionary or from the user's input history). A variant of this approach would be to compute a new matching phrase list after each keystroke by the user. In either case, an anchor phrase can be selected with minimal text entry.

Our context-sensitive phrase-completion technique requires that the full text of the document be available to the annotation system. This need coincides with our design decision (outlined in Sect. 3.1) to convert selected documents to plain text and download the text to the handheld. Space savings could be obtained by eliminating stop words from the downloaded text, although we do not do so at this time (given that the text is also available for viewing). This technique could also be of use at annotation creation time. Very often, an annotation includes some of the same words as the existing phrases. Thus, we hypothesize that context-sensitive phrase completion could potentially save users time both during annotation linking and during annotation creation.

The second technique allows the user to anchor an annotation by indicating a region on a page icon for the annotation. Moving a small rectangle over the icon specifies the region. The exact page is indicated by clicking on a scroll button until the desired page number appears (for long documents, an alternative direct page number entry area is clearly necessary). This low-cost graphical page icon is generic; it has the same appearance for every document the user annotates. An alternative would be to use full thumbnails of the page, thus allowing the user to point to a specific figure, etc. We deemed the computational cost of generating such thumbnails to be too high on our handheld devices; similarly, we deemed the required bandwidth too high to transmit such thumbnails to the device. The critical insight behind our regional anchoring solution is that a Notable user has the paper document at hand (our regional anchoring strategy is not applicable in the case where the annotator is viewing the plain text version of the document directly on the handheld device); thus, a perfect thumbnail is not necessary. For more precise anchoring, however, future work could include compromise strategies, such as "icon stationery," in which the visual appearance of the icon would be tailored to the known genre of the document. For example, two-column documents or letters might have associated specialized icons.

Our current implementation allows users to choose between context-sensitive phrase completion and icon-based graphical pinning as anchoring strategies (if finer granularity annotations are desired). We envision that the two techniques could be coupled to produce an even more efficient approach to annotation anchoring. The region and page selected by the user could constrain the list of possible phrase matches that is generated in response to user text input. This combined technique would require that the handheld device have a mapping from text to page position, but would allow for a handheld-specific counterpart to the first example in Fig. 3.

3.3 Viewing and Sharing Annotations

In addition to specifying the platform on which annotations are created and the platform on which the annotation target is viewed, the design of an annotation system must also specify where the annotations themselves can be read. In Notable, annotations can be read on the handheld device and can also be viewed on the desktop machine. An important design question concerning the annotation reading platform is: what is the technique used to display annotations? Are annotations displayed inline in the target document; are they overlaid on the document; are they independent of the document? The handheld view of annotations in Notable necessarily must show annotations independently from the document, since our assumption is that the document is viewed on paper. Users can navigate from one annotation to the next using on-screen arrow symbols. Each annotation displays its own meta-information, including its author and the specific location to which it applies. On the desktop, however, Notable need not be constrained in this way. Although currently annotations can only be read on the desktop by using the viewer built for the document repository in which the annotations are stored (anchoring information is available only as a declarative property of an annotation), future work could use the anchoring information associated with annotations to display annotations inline. The MultiValent Document (MVD) architecture [7] might provide a good substrate for creating such a view for Notable annotations, as has proven true for NotePals notes [8]. One of the challenges in displaying regionally anchored annotations at the correct place is to ensure that the digital version and paper version of the document both utilize the same graphical layout. For example, the annotation system might need the user to provide extra information to align the two versions (e.g., what is the last word on page 1 in the paper version?). Animated techniques for revealing annotations via progressive disclosure are also an interesting avenue to explore [9].

Decisions about annotation features play an important role in determining the work practices that evolve around an annotation system. For example, willingness to publish annotations is highly sensitive to the degree of access control that is associated with annotations. The ability to make simple distinctions among personal, group, and public annotations can have a large impact on the usability of the system in organizational settings. Notable records for each annotation who its author is so that access control will be possible in future versions. The ability to annotate an annotation is also significant in determining the degree of conversation that revolves around annotation. In Notable, we have built in the ability to annotate an annotation from the start. Our approach is to model annotations as documents in their own right, thus allowing for cascading annotations to emerge.

The scenario depicted in Sect. 2 was motivated by a common occurrence for researchers—a need to participate in the group review of an article. In our original version of Notable, coordination of groups was handled out-of-band from the annotation application. Individual users each had to search for the appropriate document handle, then proceed to annotate the document. In our second design iteration, we recognized that there was an opportunity for amortizing the cost of searching across a group, thus reducing the amount of interaction required of most users. The current design allows for an individual user to specify at document selection time that the document is destined for group annotation. The consequence of

this declaration is that all members of the selected group automatically receive the document (and any associated annotations) the next time they synchronize their handheld devices. In the current implementation, group information is statically defined, but we see clearly how to extend this concept to arbitrarily defined groups. Future versions might allow users to have more fine-grained control over which annotations of the group should be downloaded (e.g., annotations written by anyone in the group on this document, but only from this category).

4 System Architecture

Fig. 4 depicts the current Notable system architecture, which is composed of six major components: handheld annotation application, annotation conduit, desktop search application, format conversion service, document/annotation repository, and search service. A simplified version of this architecture may be used if the available handheld devices are wirelessly networked (the “desktop” components are no longer necessary). In this section, we give a brief overview of the data flow among components and also describe each component’s functionality in more detail.

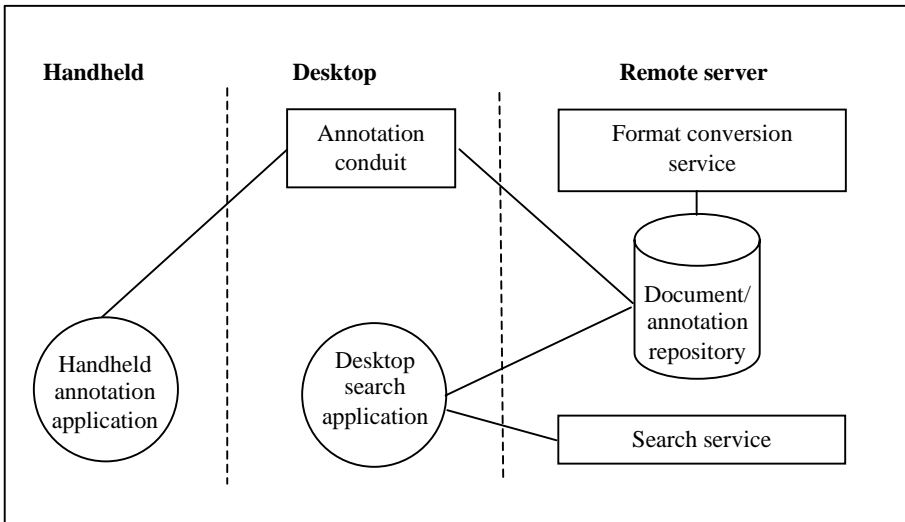


Fig. 4. Notable architecture (version from non-wireless handhelds)

4.1 Handheld Components

Data flow between the handheld annotation application (implemented in C++ for the Palm™) and the annotation conduit takes place at synchronization time. During synchronization, the annotation conduit sends to the annotation application the necessary meta-information (possibly including full text) for the documents the user has selected for future annotation. Any previously existing annotations are sent along with the document metadata. In the opposite direction, the annotation application sends to the conduit all annotations created on the handheld device since the previous synchronization event, together with linking and anchoring information for the annotations.

4.2 Desktop Components

The annotation conduit (implemented in Java) is primarily an intermediary between the handheld annotation application and the document/annotation repository. The conduit locates selected documents by performing a search of the document/annotation repository for documents flagged for download by the user currently performing the synchronization (note that a document might be flagged either because the user selected the document for annotation or because someone in the user's group flagged the document for group annotation).

The conduit is also responsible for sending created annotations to the document/annotation repository for storage. In Notable, annotations are modeled as documents in their own right. This design decision simplifies the activities of annotating annotations and searching for annotations, since annotations differ little from other documents in the repository (they have some special properties, but share many properties in common with other documents).

The desktop search application (also implemented in Java) presents a user interface whereby documents may be located and selected for annotation. The search application sends the entered search parameters to a search service that operates on a remote server. It can receive results from the search service incrementally.

4.3 Remote Server Components

The Notable document/annotation repository is a distributed, property-based system (implemented in Java) that provides transparent document access and remote property-triggered service invocation [10]. Access to all documents stored in the repository is controlled at two levels. At the first level, authentication is necessary for access to the repository. At the second level, property-based access control is possible, though our current implementation does not make use of this feature.

We have defined a schema for the documents and annotations that are created and accessed via the Notable system. By defining a schema (and an accompanying API for submitting and retrieving documents that match the schema), we have built a degree of interoperability into Notable. New annotation reading and viewing platforms can be easily added. Examples of schema properties for documents include type, title, state (indicating if a document is ready for download and by whom), and annotation time. For annotations, the schema defines link and anchor properties in addition to the standard document properties.

The format conversion employed by Notable is invoked by the documents themselves, through the use of active properties. The format conversion service resides outside of the repository on a general-purpose document service bus. This service bus provides numerous services for transforming documents, including summarization, translation assistance, and printing, in addition to format conversion. The format conversion service is invoked by the active property via its Java API.

The search service employed by Notable, also external to the document repository, is capable of performing meta-search across a variety of heterogeneous sources—including the Web as well as local document repositories. The search service satisfies search requests by utilizing a unified interface to numerous existing search engines through the use of proxies [11].

5 Related Work

Annotation systems have been developed for a variety of platforms and media types. Examples include the ComMentor [5] and ThirdVoice™ applications for annotating Web pages, Tapestry [12] features for commenting on e-mail, Microsoft Word™ features for marking up and commenting on text documents, Dynamite [13] features for annotating multimedia documents, XLibris [14] features for marking up documents viewed on a tablet-based computer, and the NotePals [8] application for meeting note taking on both handheld devices and tablet-based computers. More systems are surveyed and analyzed in [2, 15].

Of the annotation systems we have examined, NotePals is the most similar to Notable, even though the focus of NotePals is on supporting note taking at meetings and not comments on documents. Users of NotePals author ink-based notes on a meeting and assign those notes to a particular category and time span that together identify the meeting. At synchronization time, all notes from the group that have the same category and time span are merged together and made accessible to the group. Accordingly, members can take turns authoring notes and can easily share ink-based notes. In addition, NotePals annotations can be linked to particular presentation slides, audio segments, and agenda items based on timestamps. In contrast to NotePals, Notable is tailored to annotating user-selected documents (rather than meetings), uses search for document selection (rather than established category names), relies on keystroke-based annotations (rather than ink-based annotation), and supports phrase-based and region-based anchoring (rather than timestamp-based anchoring).

At the more detailed level of anchoring strategies, there is related work in both the textual and graphical domains. [16] outlines a method for using phrase completion on handheld devices in order to facilitate text entry of Japanese characters. Matches come from a dictionary and are ranked by a variety of factors, including term frequency. Although there is no initial context sensitivity, adaptation to the current document context occurs by virtue of the ranking algorithm giving precedence to previously selected words. [17] uses site-specific auto-completion to inform users about the selectivity of their chosen keywords. Outside of the realm of handheld devices, context-sensitive phrase completion has been used in a variety of products, including text-editing software (e.g., emacs) and personal finance applications (e.g., Quicken™). Similar in spirit to icon-based graphical pinning, Adobe Acrobat Reader™ allows users to select an area of a document for magnification by manipulation of a small rectangle on a thumbnail of a document page.

6 Conclusions and Future Work

A handheld-based annotation system, Notable permits users to author sharable and reusable comments on paper documents in a mobile environment. Key design decisions for Notable include separating the annotation writing platform from the document viewing platform, providing search as the method for document selection, offering context-sensitive phrase completion and icon-based graphical pinning for

fine-granularity annotation anchoring, and including some support for coordinating group annotation scenarios.

Our work on Notable has suggested numerous directions for future research. First, we recognize the importance of involving more users in our design process. Conversations with potential users have already resulted in a second iteration of the Notable design. At this point, we believe that a set of qualitative and quantitative user studies would be of great benefit. Questions that we would like to investigate include the usability of the system, the range of scenarios in which Notable is of value, and the effectiveness of our anchoring techniques.

For usability, the design must be extended to allow for the effective viewing of the created annotations and documents on other devices in the user's computing environment. Hybrid graphical/textual/structural anchoring strategies and the development of "icon stationery" are also likely to improve usability. Designing for usability does not stop at interface design. Annotation is part of a wide range of group processes. Accordingly, we foresee the introduction into Notable of active annotations and the integration of better group support.

More generally, we see Notable as an example of an application genre that is rapidly evolving: remote user interfaces that allow users to interact with physically separate objects. With Post-It™ size display areas, keystroke-based text entry systems, and programmable infrastructure, today's handheld devices offer designers the opportunity to orchestrate new interactions between users and the objects they are viewing. At the same time, the limitations in screen real estate, input technology, memory, and bandwidth of these devices (in comparison to desktop computers) impose a complex set of design constraints. Translating desktop applications to this new environment is an increasingly significant design challenge.

7. Acknowledgments

We thank Brian Lee for his implementation work on the first version of the Notable system and Polle Zellweger for her contribution of the idea behind icon-based graphical pinning. We are also grateful to Ken Pier, Mark Stefik, and the HUC2k reviewers for their thoughtful feedback on earlier drafts of this paper.

References

1. Marshall, C.: Toward an Ecology of Hypertext Annotation. In: Proceedings of ACM Hypertext '98. Pittsburgh, PA (June 1998) 40-49
2. Cousins, S. B., Baldonado, M., Paepcke, A.: A Systems View of Annotations. PARC Tech Report P9910022 (April 2000)
3. O'Hara, K. and Sellen, A.: A Comparison of Reading Paper and On-Line Documents. In Proceedings of CHI '97. Atlanta, GA (March 1997) 335-342
4. Baldonado, M., Cousins, S., Lee, B., Paepcke, A.: Notable: An Annotation System for Networked Handheld Devices. In: Proceedings of CHI '99. Pittsburgh, PA (May 1999) 210-211

5. Röscheisen, M., Mogensen, C., Winograd, T.: Interaction Design for Shared World-Wide Web Annotations. In: Proceedings of CHI '95, Vol. 2. Denver, CO (May 1995) 328-329
6. Chalfonte, B. L., Fish, R. S., Kraut, R. E.: Expressive Richness: A Comparison of Speech and Text as Media for Revision. In: Proceedings of CHI '91. New Orleans, LA (April-May 1991) 21-26
7. Phelps, T. A. and Wilensky, R.: Multivalent Annotations. In: Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries. Pisa, Italy (September 1997)
8. Davis, R. C., Landay, J. A., Chen, V., Huang, J., Lee, R. B., Li, F., Lin, J., Morrey III, C. B., Schleimer, B., Price, M. N., Schilit, B. N.: NotePals: Lightweight Note Sharing By the Group, For the Group. In: Proceedings of CHI '99. Pittsburgh, PA (May 1999) 338-345
9. Zellweger, P., Chang, B., Mackinlay, J.: Fluid Links for Informed and Incremental Link Transitions. In: Proceedings of Hypertext '98. Pittsburgh, PA (June 1998) 50-57
10. Dourish, P., Edwards, K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D. B., Thornton, J.: Extending Document Management Systems with User-Specific Active Properties. *ACM Transactions on Information Systems* (in press)
11. Paepcke, A., Cousins, S. B., Garcia-Molina, H., Hassan, S. W., Ketchpel, S. K., Röscheisen, M., Winograd, T.: Using Distributed Objects for Digital Library Interoperability. *IEEE Computer Magazine* 29:5 (May 1996) 61-68
12. Nichols, D., Oki, B. M., Goldberg, D., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM* 35:12 (1992) 61-70
13. Wilcox, L. D., Schilit, B. N., Sawhney, N.: Dynamite: A Dynamically Organized Ink and Audio Notebook. In: Proceedings of CHI '97. Atlanta, GA (March 1997) 186-193
14. Schilit, B. N., Golovchinsky, G. Price, M.: Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In: Proceedings of CHI '98. Los Angeles, CA (April 1998) 249-256
15. Ovsianikov, I. A., Arbib, M. A., McNeill, T. H.: Annotation Technology. *International Journal of Human-Computer Studies*, 50:4 (1999) 329-62
16. Masui, T.: An Efficient Text Input Method for Pen-Based Computers. In: Proceedings of CHI '98. Los Angeles, CA (April 1998) 328-335
17. Buyukkokten, O., Garcia-Molina, H., Paepcke, A.: Focused Web Searching with PDAs. In: Proceedings of WWW 9. Amsterdam, The Netherlands (May 2000) 213-230