Informal Information Gathering Techniques for Active Reading

Ken Hinckley, Xiaojun Bi, Michel Pahud, Bill Buxton
Microsoft Research, One Microsoft Way, Redmond, WA 98052
{kenh, mpahud, bibuxton}
{kenh, mpahud, bibuxton}@microsoft.com; xjunbi@gmail.com

ABSTRACT
GatherReader is a prototype e-reader with both pen and multi-touch input that illustrates several interesting design trade-offs to fluidly interleave content consumption behaviors (reading and flipping through pages) with information gathering and informal organization activities geared to *active reading* tasks. These choices include (1) relaxed precision for casual specification of scope; (2) multiple object collection via a visual clipboard; (3) flexible workflow via deferred action; and (4) complementary use of pen+touch. Our design affords active reading by limiting the transaction costs for secondary subtasks, while keeping users in the flow of the primary task of reading itself.

Author Keywords
active reading; informal interaction; pen; touch; tablets.
ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: Input

INTRODUCTION
This note explores the theme of informal information gathering techniques in the context of a prototype e-reader application with both multi-touch and pen input. Our belief is that slates, e-readers, and booklets represent a new class of digital appliance— one targeted for *casual use*: that is, opportunistic, low-overhead interaction with loosely structured tasks surrounding the consumption, curatorship, and lightweight creation of content, as well as informal organization of materials [3,9,12,17]. With the growth of this market we have the opportunity to embrace a different approach to interaction— one that is in keeping with the casual intent and context of such usage and which will complement, rather than replace, interfaces that support more formal and structured activities.

Our goal is to support both casual reading and the subtasks demanded by active reading in the same user experience. Reading is a primary activity for many knowledge workers [1], yet it remains unclear how to best support activities such as annotation [3,9], collection of encountered information [12], non-linear navigation [15], and informal organization for quick storage and retrieval [17]— in a way that keeps users "in the flow" [2] of reading.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

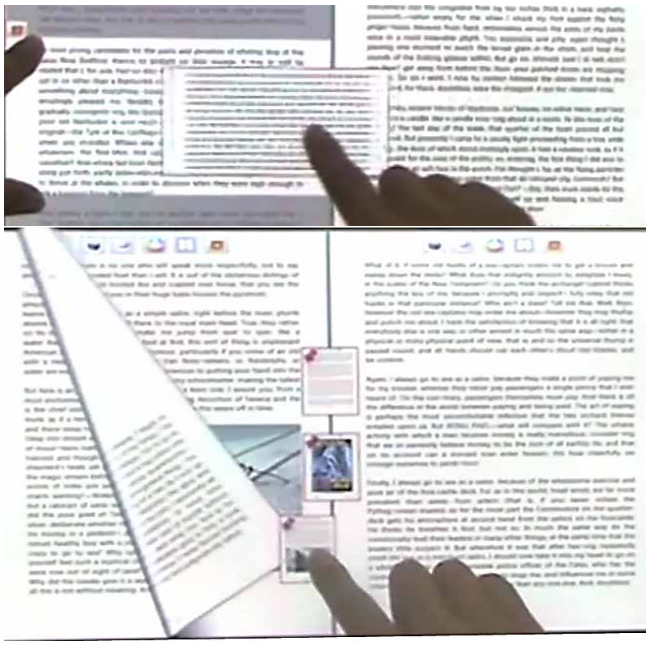

Fig. 1. **Top:** Framing part of a page & dragging it to the *Pocket*. **Bottom**: Collecting entire pages by dragging them to the center.

GatherReader uses both pen and multi-touch interaction to support fluid transitions between reading and writing tasks. We contribute techniques for grabbing pieces of content, collecting entire pages, temporarily storing items in a floating visual clipboard known as the *Pocket*, reading side-by-side with writing, automatic two-way cross-referencing of notes with source material, and post-hoc interpretation of pen gestures. Such interactions afford a middle ground where content consumption and lightweight content creation feed into one another, a virtuous cycle where the user can browse, collect, aggregate, annotate, and of course read content using a combination of pen and touch inputs.

We discuss several design trade-offs exemplified by our prototype. These include relaxed precision for casual specification of scope; multiple object collection via a visual clipboard; flexible workflow via deferred action; and distinct roles for pen and touch input. These collectively minimize the transaction costs (that is, the round-trip time-motion and attention costs) of the active reading subtasks that we support, and thereby reflect, augment, and respect user's hard-won existing skills for interacting with documents with both hands, multiple fingers, and a stylus.

RELATED WORK
Studies of knowledge workers have repeatedly shown that reading goes hand-in-hand with writing [1], cross-

referencing information between documents [1,11,13], non-sequential navigation [15], gathering pieces of information [12], spatial layout [10], and so on. All of these active reading activities are subtasks of the primary activity of reading, browsing, or otherwise skimming through pages, notes, and piles of documents. As such, active reading should support these activities with lightweight interactions at a reasonable price in terms of the transaction costs, and without distracting users from their primary task: reading.

Several efforts have investigated informal interfaces and lightweight approaches to organization. Studies of knowledge workers' activity with paper documents [3] and notes [9] indicate need for unstructured ways to gather information. Informal organization tools should support rapid storage and retrieval, without forcing users to name or categorize information [17]. LiquidText [16] uses multi-touch with scrollable documents to support active reading, including collection of document snippets and arranging cross-referenced excerpts side-by-side with the document. GatherReader explores new ways to support such task workflows. It adopts the central design principle of division of labor between pen and touch: *the pen writes and touch manipulates* [8]— which LiquidText, while a multi-touch-only system, nonetheless explicitly notes may benefit active reading tasks [16]. We also contribute a new perspective on informal information gathering in the context of e-reading.

GATHERING CONTENT

Traditional copy-paste offers one way to gather content. Copy-paste is a command phrase with scope, verb, and indirect object: *Copy* (**verb**) <*this*> (**scope**) *to* <*new location*> (**indirect object**). Effectively interleaving the gathering of content with reading requires reducing the steps— the individual, atomic interactions— in each part of this command phrase. It also must maintain a flexible workflow that permits the user to quickly grab content, while deferring secondary decisions of what to do with it.

Here, we contrast our approach with that of LiquidText because it shares many motivations with our own work, it is a good example of a recent paper applying modern multi-touch interaction techniques to this problem, and because contrasting the specific actions and gestures employed by LiquidText shows clearly the trade-offs we have made to optimize the task workflow of GatherReader to afford the casual and relatively unstructured collection of encountered information [12]. The workflow supported by LiquidText is also similar to, but more streamlined than, the transactions required to perform similar tasks on the iPad; gathering a passage of text requires the step-by-step selection of individual words. The steps in LiquidText [16] are:

1. Tap on an opening word boundary;
2. Touch down and drag to extend the selection; and
3. Lift at the closing word boundary.

To gather (extract) the text selection, the user then must:

4. Hold the selection with the nonpreferred hand; and
5. Drag the selection to the canvas (a freeform arrangement area side-by-side with the document).

Note here that LiquidText already simplifies one step of the traditional copy-paste workflow as exemplified by the iPad: the *Copy* command itself is made implicit by the direct manipulation of steps 4-5 so that no "command" selection *per se* has to be made.

Relaxed Precision for Casual Specification of Scope

GatherReader uses a *Frame-and-Drag-to-Pocket* technique to support gathering passages of text. The user frames the desired passage with thumb-and-forefinger at the margin of the document to specify the desired passage of text. Note that this relaxes the precision required by steps 1-3 above. In GatherReader this framing gesture accentuates the framed region, while a subtle alpha-transparent gray mask deemphasizes the rest of the page (*Fig. 1, top*).

This approach lowers the transaction costs for grabbing a passage of text in two ways: first, it combines steps 1-3 into a single chunk [4]; and second, it relaxes the precision with which the user must specify the selection in the first place. The user simply frames the *approximate* start, and the *approximate* end, by bringing thumb and forefinger to the screen. This integrates steps 1-3 into a single multi-touch gesture while also affording a casual specification of scope.

But to be clear, this is a trade-off: favoring imprecise specification of scope means we forego the ability to gather word-by-word selections. Furthermore, our technique is geared towards selecting a single line up to a paragraph or two, but not arbitrary selections that might span multiple pages. We could add mechanisms to handle these cases, but this might make the techniques more complex, distracting, or inefficient and thereby inhibit the informal information-gathering behaviors we wish to support in the first place.

Integrating Selection and Gathering

But what of steps 4-5 outlined above? GatherReader uses a two-finger framing gesture to specify the selection (steps 1-3), as well as to hold a temporary state (mode) where a single-finger drag tears off a copy of the selected region (*Fig. 1, top*). Thus our technique enables users to articulate steps 1-5 as a continuous two-handed gestural phrase, which makes it feel more like a single action [4].

However, we observed (during informal user tests, reported later in this note) that many users prefer to articulate this unimanually— that is, they frame the desired passage with thumb and forefinger of their preferred hand, and then they extract the selection by dragging it away with a single finger, again with their preferred hand. Hence, our technique supports framing passages from either the left margin or right margin, and furthermore the selection frame, once established, persists until the user drags it off (or taps again to dismiss it). Supporting this one-handed task flow is critical for slates, which users often hold in the nonpreferred hand while interacting with their content.

Multiple Object Collection via a Visual Clipboard

Step 5 of the LiquidText workflow above, dragging an extract to the canvas, supports an important capability: that of moving pieces of content to a secondary surface where

they can be freely arranged or grouped by proximity, annotated, inserted in the context of other notes, etc. However, one weakness of the approach is that the split-screen canvas view is always visible in LiquidText— meaning that the document itself is not the sole focus of attention at times of intense reading— and furthermore it means that the indirect object of the resulting copy-paste command phrase implicitly *requires the user to decide immediately where to put the extract*— in effect, how to categorize and organize it— at the moment he gathers it.

Our approach keeps attention focused on the document even as the user gathers multiple pieces of content. The *Pocket*, inspired by the knitted-elastic straps on moleskin notebooks, is an area that always remains visible (i.e., it floats above whatever document, notes, or application happens to be displayed below), where users can tuck content away (*Fig. 1, bottom*). Our prototype uses the spine to pocket items (because our original motivation stemmed from dual-screen devices [7]), but feedback from test users suggests the bottom edge of the screen would be a better design choice for our current single-screen prototype.

The Pocket serves as an unnamed and uncategorized place to *temporarily* place one or more objects, as well as a visually salient reminder of what has been collected. Unlike traditional copy-paste, it is not a single-buffer model, so the user can freely gather multiple items while continuing to read. Once the user frames part of a page, the user places it in the Pocket via a single-touch drag from the selected region to the spine between the pages. The selection then snaps to the spine and shrinks to thumbnail size, stored for later use. Multiple items may be dragged into the pocket or stacked on top of one another, if desired.

Analogs of the Pocket have a long history in human-computer interaction, such as van Dam's *Attic* [5] and Roberson's *Fix-and-Float* [14], but our work extends these concepts to a visual clipboard in the modern context of direct input on tablets and e-readers. The Pocket was originally conceived in the context of the unpublished Courier project at Microsoft; here we articulate the intent and utility of this approach for gathering content.

Flexible Workflow via Deferred Action

Significantly, the Pocket also supports deferral of the decisions as to what to do with the collected items, where to put them, or how to classify them; in short, it offers a place to stash an object until its future role and significance are better understood [3,9]. The user may drag items out of the Pocket at a later time to drop them into a different document, notebook page, application, or web page, as inputs for subsequent content creation. At present the GatherReader prototype leaves off here where other efforts, including our own previous work on the Codex [7] and pen + touch [8], show how to support further handwriting recognition, search, and richer content creation. By offering an ever-present and immediate place to put things with a minimal transaction cost, GatherReader explicitly seeks to support low-overhead collection of content without disrupting the core experience of reading itself.

A Further Streamlined Approach to Gather Pages

Our *Frame-and-Drag-to-Pocket* technique allows gathering pieces of content, but to push the idea of relaxed precision even further, we implemented *Pocketing an Entire Page* (*Fig. 1, bottom*). This integrates page flipping itself with a *Drag-to-Pocket* gesture: the user starts flipping the page but keeps pulling, similar to tugging at the corner of a physical page, to "tear the page out of the book."

Here, the gesture begins as an ordinary page flip— articulated by swiping across the edge of a page— but if the user continues dragging instead of lifting, a thumbnail of the current page appears under the user's finger, as if the user had torn out a copy of the page. We use a simple movement distance threshold and 500ms time-out to separate this gesture from page flip gestures (i.e., short, quick drags or finger taps at the margin of the page).

A page flip implicitly refers to a scope of one page. By extending page flip to our pocketing concept, we offer users an extremely low-overhead way of "saving" a page for later reference, and we again leverage the Pocket so that the decision of what, if anything, to do with the page can be put off. This affords collection of pages by eliminating as many ancillary costs as possible, and indeed integrates the method of collecting pages with the act of flipping through the document itself. In this way, the design offers a low-resistance path for users to move from simple browsing behaviors (flipping pages) to the informal information gathering behaviors that our design seeks to encourage.

Annotation and Notes Side-by-Side with Reading

We use pen input to support freeform annotation and note-taking while reading [1]. We believe specialization of the roles of the devices [8] is critical to support seamless mark-up. Systems with only touch (or a capacitive stylus) require mode switching to toggle between mark-up and direct manipulation. Our system also supports split-screen viewing of the document side-by-side with notes by tapping on a *split-screen view* icon. This opens a notebook on the right with independent page navigation [7]. The pen always writes, whether the user employs it to mark-up the document or take separate (split-screen) notes.

Automatic Two-way Cross-Referencing of Notes

Any notation that the user jots in the notebook is automatically linked back to the document page that currently appears on the left-hand side of the screen. A subtle animation of a blue cone links the notation to the source page as soon as the user finishes writing a passage. If the user reviews his notes later, tapping on the resulting hyperlink icon flips to the source page for the notation in the document; tapping it again brings the user right back to where he's currently working. This supports working with multiple documents, or parts of a document, in one working context. The automatic hyperlinks are two-way: tapping a *Comments* button at the bottom of the document page

brings the user to any notations associated with that page. Furthermore, both note and document pages can contain cross-refrences to several different pages. Animated page flips help make the link traversals salient and unambiguous.

Our use of pen and touch to support rapid interleaving of notations with flipping through the document (via touch), plus our automatic hyperlinking mechanism, affords low-overhead creation of notes. It also affords quick navigation from notes to related content, and vice versa. Because we automatically capture these cross-links, we keep the user focused on the text and their notations: the user does not have to explicitly form links or create bookmarks to ensure that he can get back to the referenced content. This approach automates much of the page-by-page navigation that the user would otherwise have to engage in: links are automatically created, and link traversal provides a way to easily flip back and forth between parts of a document.

Post-hoc Pen Gestures

By default our system always treats pen strokes as mark-up. The user can give a mark a post-hoc interpretation as a gesture by touching it. This interprets the pen stroke as a specification of scope (e.g. lasso, underline, or scratch-out), and brings up an associated context menu with options such as *Copy*, *Highlight*, and *Delete*. For example, the user can activate a region that he has previously circled with the pen by touching the ink stroke, and then drag the resulting non-rectangular clipping to the Pocket. This lightweight approach to mode switching is another example of *flexible workflow via deferred action* that builds on previous work [6,18]. The virtue of our approach is that the user can initially focus on reading while marking up a text, and then later infuse those marks with computation, if desired.

PROTOTYPE IMPLEMENTATION AND EVALUATION

Our prototype runs on Windows 7 using a Wacom 21" FS5 pen + touch display tablet. We use Wacom's custom multi-touch driver to support simultaneous multi-touch and pen events. Slates with pen and multi-touch (e.g. ASUS Eee Slate EP121 PC) are coming to market as of this writing.

We conducted informal usability studies with 11 knowledge workers. Each user was given specific tasks to accomplish with each technique, and also given time to freely experiment. Users particularly appreciated the use of multi-touch in combination with pen for mark-up, and split-screen notes side-by-side with reading. Users were enthusiastic about the ability to quickly grab content using *Frame-and-Drag-to-Pocket*, although most users preferred to articulate the gesture with one hand, as mentioned previously. Users liked the quick, approximate method of selection and did not ask for more precision. The *Pocket Entire Page* interaction also tested well but didn't resonate as strongly with users, because users often just wanted to grab a specific passage. For *Post-hoc Pen Gestures*, users found tapping a pen stroke a clear way to activate it. However, better palm rejection is needed to prevent inadvertent touch gestures while writing.

DISCUSSION

To fully support active reading in knowledge work, the designer must carefully weigh the transaction costs and workflows afforded by traditional approaches, and where warranted, reconsider tradeoffs between *precise vs. relaxed specification of scope*, *immediate vs. deferred action*, and *multi-touch vs. stylus interaction*, to name a few.

We have demonstrated techniques that enable users to shift from browsing content, to collecting and organizing content, without imposing any a priori overhead (e.g. naming, tagging, structuring) on the reading experience. Explicit organization or "saving of things" is never required, yet because they are available in a low-overhead and informal way, they quietly pervade the experience. Likewise, pen input as a way to quickly interleave annotations and mark-up encourages another lightweight form of content creation. It is noteworthy that the working patterns afforded by our approach are difficult to effectively articulate in the traditional single-clipboard copy-paste model of content collection, or in single-modality multi-touch interfaces where no pen is available for annotation.

We believe the techniques and design approaches we advocate here offer a constructive way of framing the debate about whether casual-use devices should be for *content consumption* or *content creation*. Embracing both within the same workflow, in a way that minimizes the transaction costs, offers the potential to support knowledge work in a way that embraces human ways of consuming, collecting, and creating content.

REFERENCES

1. Adler, A., et al. A diary study of work-related reading design implications for digital reading devices. *CHI'98*.
2. Bederson, B., Interfaces for Staying in the Flow. Ubiquity. **5**(27).
3. Bondarenko, O., Janssen., R. Documents at Hand: Learning from Paper to Improve Digital Technologies. *CHI'05*.
4. Buxton, W. Chunking and Phrasing and the Design of Human-Computer Dialogues. *Proc. IFIP World Computer Congress*, 1986.
5. DeRose, S.J., van Dam, A., Document structure and markup in the FRESS hypertext system. Markup Languages, 1999. **1**(1): p. 7-32.
6. Hardock, G., Kurtenbach, G., Buxton, W. A Marking Based Interface for Collaborative Writing. *UIST'93*.
7. Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., Balakrishnan, R. Codex: A Dual-Screen Tablet Computer. *CHI'09*.
8. Hinckley, K., et al. Pen + Touch = New Tools. *UIST 2010.*
9. Kidd, A. The marks are on the knowledge worker. *CHI'94*. 1994.
10. Kirsh, D., The intelligent use of space. Artificial Intelligence, 1995. **73**: p. 31-68.
11. Liao, C., et al., Papiercraft: A gesture-based command system for interactive paper. ACM Trans. Comput.-Hum. Interact, 2008. **14**(4).
12. Marshall, C., Bly., S. Saving and Using Encountered Information: Implications for Electronic Periodicals. *CHI'05*.
13. Price, M.N., Golovchinsky, G., Schilit, B.N. Linking by inking: trailblazing in a paper-like hypertext. *ACM HYPERTEXT '98*.
14. Robertson, G.G., Card., S.K. Fix and float: object movement by egocentric navigation. *UIST'97*.
15. Tashman, C., Edwards, W.K. Active Reading and Its Discontents: The Situations, Problems and Ideas of Readers. *CHI'11*.
16. Tashman, C., Edwards, W.K. LiquidText: A Flexible, Multitouch Environment to Support Active Reading. *CHI'11*.
17. Voida, S., Mynatt, E., Edwards, W.K. Re-framing the desktop interface around the activities of knowledge work. *UIST'08*.
18. Zeleznik, R., et al., Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. *UIST 2010*.