Hunter Gatherer: Interaction Support for the Creation and Management of Within-Web-Page Collections

m.c. schraefel,[1] David Modjeska,[2] Daniel Wigdor,[1] Yuxiang Zhu[1]

[1]Dept. of Computer Science
University of Toronto
Toronto, Canada
{mc |dwigdor |yuxiang}@dgp.toronto.edu

[2]Faculty of Information Systems
University of Toronto
Toronto, Canada
modjeska@fis.toronto.edu

ABSTRACT

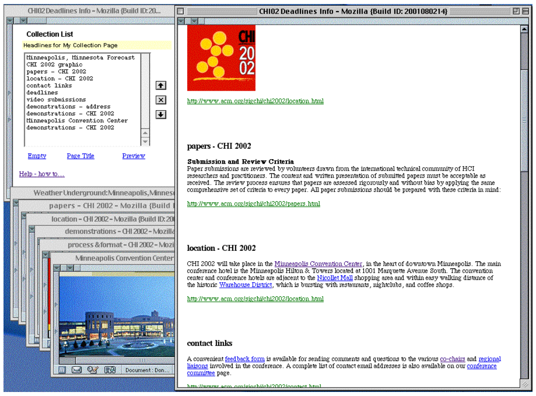
Hunter Gatherer is an interface that lets Web users carry out three main tasks: (1) collect components from within Web pages; (2) represent those components in a collection; (3) edit those component collections. Our research shows that while the practice of making collections of content from within Web pages is common, it is not frequent, due in large part to poor interaction support in existing tools. We engaged with users in task analysis as well as iterative design reviews in order to understand the interaction issues that are part of within-Web-page collection making and to design an interaction that would support that process. We report here on that design development, as well as on the evaluations of the tool that evolved from that process. These evaluations confirm the effectiveness of Hunter Gatherer for facilitating within-Web-page collection making and management. Aside from the practical aspects of the tool, the interaction supported by Hunter Gatherer also has implications for Intellectual Property in terms of facilitating the concept and practice of users repurposing Web data.

KEYWORDS

Interaction design, information gathering, information management, information sharing, attention, collections

INTRODUCTION

Studies of Web-based information interaction such as [2][4][8], have generally dealt with a Web page as the smallest unit of consideration. Task analysis carried out in a user study reported in [9] indicates that users, however, regularly need to deal with smaller units, that is, information components from *within* Web pages. The study found two things: (1) that Web users want to be able to make collections of information found from within Web pages, but that (2) users only infrequently make such collections, in large part because of poor interaction support for this activity. For instance, bookmarks, referencing entire pages often capture more than the desired data; this forces users first to load and then to sift through multiple pages to attempt to find the desired material. Text editors cause users to shift attention between the information gathering task in the browser and the information management task with the editor. With editors, users often forget or neglect to label the collected component with a title or the URL of the source page, making later access to the original material difficult, degrading the value of the collection over time.


CHI02 Deadlines Info - Mozilla {Build ID:20...
CHI02 Deadlines Info - Mozilla {Build ID: 2001080214}
Collection List
Headlines for My Collection Page
Empty
Page Title
Preview
Help - how to...
Weather Underground:Minneapolis,Minnes
papers - CHI 2002 - Mozilla {Build I
location - CHI 2002 - Mozilla {Build ID:20
demonstrations - CHI 2002 - Mozilla
process & format - CHI 2002 - Mozi
Minneapolis Convention Center
Document : Don...
CHI
20
02

```
Minneapolis, Minnesota Forecast
CHI 2002 graphic
papers - CHI 2002
location - CHI 2002
contact links
deadlines
video submissions
demonstrations - address
demonstrations - CHI 2002
Minneapolis Convention Center
demonstrations - CHI 2002
```

http://www.acm.org/sigchi/chi2002/location.html

papers - CHI 2002

Submission and Review Criteria
Paper submissions are reviewed by volunteers drawn from the international technical community of HCI researchers and practitioners. The content and written presentation of submitted papers must be acceptable as received. The review process ensures that papers are assessed rigorously and without bias by applying the same comprehensive set of criteria to every paper. All paper submissions should be prepared with these criteria in mind:

http://www.acm.org/sigchi/chi2002/papers.html

location - CHI 2002

CHI 2002 will take place in the Minneapolis Convention Center, in the heart of downtown Minneapolis. The main conference hotel is the Minneapolis Hilton & Towers located at 1001 Marquette Avenue South. The convention center and conference hotels are adjacent to the Nicollet Mall shopping area and within easy walking distance of the historic Warehouse District, which is bursting with restaurants, nightclubs, and coffee shops.

http://www.acm.org/sigchi/chi2002/location.html

contact links

A convenient feedback form is available for sending comments and questions to the various co-chairs and regional liaisons involved in the conference. A complete list of contact email addresses is also available on our conference committee page.

http://www.acm.org/sigchi/chi2002/contact.html

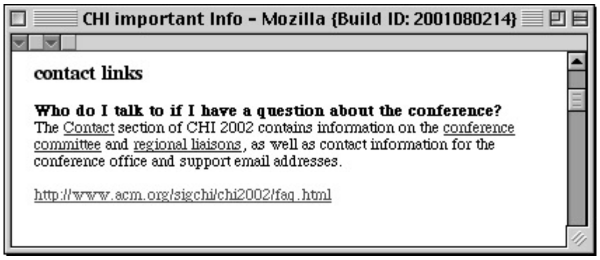
Figure 1. Hunter Gather. The Collection Window (foreground) presents all elements in the current collection; the List of collected components appears upper left; lower left are the discrete pages from which the components in the collections have been selected.

Despite these shortcomings, those surveyed still expressed a need to create collections from material within Web pages. Scenarios for such collections are easy to imagine: a journalist might want to build a collection of different newspaper coverage of the same story. A student might build a heterogenous collection to reflect her current term, including courses, professors, gym hours and so on.

We developed Hunter Gatherer (HG) both to support this kind of within-Web-page collection making and to investigate how this novel interaction design might affect Web-based information practices. Hunter Gatherer (Figure 1) blends the transparency of bookmark capture for component selection, with the support of an editor for revising collections. The tool also automates the inclusion of a contextual, editable header for each component, and grabs the URL of the source page for that component (Figure 2), so that users can return to the source document at any time. Our goal for Hunter Gatherer's interaction design is to let users, rather than the tool, determine which information activity they wish to focus on: gathering, management or contemplation of the collection.

The location for this technical report, via anonymous FTP
ftp://ftp.cs.toronto.edu/csrg-technical-reports/437/

Hunter Gatherer is the result of an iterative process of user-based design, surveys and evaluation. This paper describes the most recent version of the artifact, the associated interaction design, and its evaluations. We begin with a discussion of related work in Web-based collection management, and illustrate where this work does not address the interaction problem most relevant to within-Web-page collection making: shifting focus between information capture and post-capture information management. We follow this with a discussion of our prototype tool development and evaluation over several iterations. Finally, we report on lessons learned from these evaluations, and describe how the results have helped to refine our understanding of the tasks we hope to support.



contact links

Who do I talk to if I have a question about the conference?
The Contact section of CHI 2002 contains information on the conference committee and regional liaisons, as well as contact information for the conference office and support email addresses.

http://www.acm.org/sigchi/chi2002/faq.html

Figure 2. Close up of single component in a collection. Figure shows collection title in the window name; automatic addition of both element header and URL back to component's source page.

RELATED WORK

Our research investigates the problems faced by Web users who wish to carry out two related tasks: to gather information components from a variety of Web sources and to manage that gathererd information. When we focus on information gathering on the Web, we foreground the process that Marshall *et al*. refer to as "information triage," the act of moving through a variety of sources to determine quickly whether they are of potential worth. The sticking point occurs when, on making such a determination, we wish to capture the component identified for retrieval. When users are engaged in information triage, they currently lack a method for putting the identified components into a collection *without* needing to make the collecting activity a foreground task. While there has been much work done on the management of Web-based document collections (which we discuss below), there has been less work on the interaction activity of placing the identified information from the source into the collection. Therefore, our work has focused especially on the latter process.

Bookmarks and Visualization

Our design model for the kind of transparent interaction that we wish to emulate has been bookmark-making. Bookmarking is well integrated with most Web browsers. The user engages a simple command key sequence, or makes a menu selection, and the current page is added to a list of bookmarks. With slightly more concentration on the bookmark task, users can shift focus to more specific information management tasks: many bookmark tools, for instance, support adding bookmarks directly to specific folders within the bookmark list. Such interaction supports a gradient of task focus, from peripheral attention to main focus. While bookmarking supports this multiple attention level for interaction, its failure to help users retrieve information effectively from bookmarks has been well discussed in Abrams *et al*. [1]. To deal with the shortcomings of bookmarks for retrieving information, several research and commercial applications have been developed. While not completely applicable to our research, there are related findings from that work which inform ours.

Card, Robertson and York's WebBooks [2] is an early example of an application for bookmark visualization. In this work, the entire Web page is always available, eliminating the requirement for a user to load each interesting bookmark iteratively. Collections of pages are visualized as books, where pages in the collection can be quickly "flipped through." While the WebBook eliminates the need for users to load pages, it still focuses on a complete Web page as the artifact of value.

More recently, Robertson *et al*. developed the Data Mountain tool to let users arrange bookmarks as page of thumbnails on an inclined plane. Compared with Internet Explorer's Favorites bookmark tool, participants were able to retrieve pages more quickly and with fewer errors [7]. Czerwinski *et al*. extended this work; they demonstrated that the name and the location of a bookmark on the plane were the two factors most important for successful retrieval; a page's thumbnail image was less important [4].

Amento, Terveen, Hill and Hix's TopicShop work [2][11] draws particularly on the Data Mountain research for letting users manage collections of sites on a given topic. In this case, an algorithm developed for TopicShop captures candidate sites, which become available to a user in a multi-paned window. In the site profile pane, for instance, a list of sites shows miniature thumbnails of the page, along with relevant site characteristics, such as name and number of links in and out of the page. This information helps users decide if they wish to visit the site. Users can then drag chosen sites into a "work area." The site is represented here as a thumbnail. Thumbnails can be "piled" into groups; groups are in turn reflected in the site profile window. Evaluation participants found this multi-view approach to evaluating and organizing collections to be TopicShop's most effective feature.

Once again, the Web page is the entity of value. This makes sense in the case of TopicShop, as the entire page or site is desired overall, since, by design, the pages collected are themselves either all "on topic" (e.g., a fan site) or are collections of links to such sites. It is not clear if the TopicShop algorithm could be extended to capture, for instance, a more heterogeneous notion of topic, as in the preceding student scenario. There, "My Term" as a topic might reflect an associative set of components such as courses and student loan information, rather than clusters of similar information.

Editors

Some editors such as Microsoft Windows' Front Page and Netscape Navigator's Communicator are better integrated for the within-Web-page collection process than basic text editors or even some word processors. Both applications let users open a blank, editable page into which they can drag content, including images, from the browser to the editor. Users can then edit the collected information in any way they wish. Unlike bookmark managers, the editor page makes all the collected components readily apparent to a user looking at the file. The file can be saved to a server via the editor's integrated FTP support. Users can also access the URL of any collected image. The same cannot be said, however, for any collected text. Unless the URL is specifically grabbed, that information is not captured. Similarly, the user must label the content themselves, since no page information (such as page title) travels with the copied content. Word processors such as Microsoft Word support drag and drop of both text and images from Web pages into files; plain text editors support text capture.

Hybrids: Spatial Hypertext

In Spatial Hypertext, which predates the emergence of the Web, the notion of the page, *per se,* does not exist. Documents are always already collections of data objects, like one's own notes on a topic, or references to other works. These data objects are manipulated in a 2D visualization space, so that the space in which a user creates a hypertext is also the space in which that document is viewed. This is a more elastic version of hypertext than what the Web currently supports. By way of intermediary, Mark Bernstein's Web Squirrel,[1] is a tool that attempts to bring some of the data object vs. Web page approach to Web practice, though its main use is for annotating bookmarks. Web Squirrel lets users create and copy information (such as URLs) into a Web Squirrel file. The Data is represented as squares to be directly manipulated in a 2D space. The objects can then be arranged and annotated. Agents sift through information in a collection (or "farm" in Squirrel parlance) and suggest connections among collected objects. Like bookmark lists, which only reveal a page title, not the page content, the Web Squirrel boxes hide annotation/link information attached to them. Also, only one box's information can be revealed at a time. As well, while users copy and pastes text information from a Web page into Web Squirrel, the source URL for that text is lost unless the user also grabs the URL and drops that into the application. This URL will then show up as a distinct box from the text. Finally, Web Squirrel does not capture images or other media.

Overview

With the exception of a hybrid tool like Web Squirrel and the Spatial Hypertext work that informs it, Web-based research has focused on managing whole Web pages and sites, rather than on the discrete content within a Web page. Even in Spatial Hypertext with its emphasis on capturing one's own annotations, however, there is little consideration of the interaction of getting content from one context to another. We wish to expand the research to consider this interaction aspect of the movement among information gathering, capture and reflection.

HUNTER GATHERER DESIGN PROCESS

Our main goal for Hunter Gatherer has been to support the collection making interaction process for collecting within-Web-page components. To determine how best to do this, we carried out the task analysis, tool comparison and an initial prototype design review [9].

Goals

From our tools and task analysis, and prototype design review, we determined 3 requirements for Hunter Gatherer.

- First, to make the addition of components to collections as transparent as highlighting text.
- Second, the interaction must support user-determined focus shift among component selection, addition, monitoring, and management.
- Third, the collected components must automatically capture enough contextual information for the collection to be immediately valuable for the user.

In the following sections, we present an overview of the artifact to support this process, and its evaluation in terms of these three goals.

Description of the Tool

First, Hunter Gatherer is a browser-based, not a stand-alone application. By integrating Hunter Gatherer with the browser in a manner similar to bookmarks, we are able to minimize the forced divided attention [12] introduced by shifting between browser and editor, gathering and management. Our approach is also proxy based. This means that the user does not have to download additional software to access the tool. While not perfect, the proxy approach also lets us support multiple operating systems and browsers simultaneously. Further, our interest is in the potential impact of supporting within-Web-page collection making on Web information practices. Multiple OS support lets us deploy the tool over a wide user space for this assessment.

Second, Hunter Gatherer does not copy data into a collection; it creates references for the components inside the Web page instead. Thus, a collection is built as a string of URLs. The resulting Aggregated URL (AURL) makes collections highly transportable: an AURL can be bookmarked or emailed. For instance, http://[anon proxy server] /examples/servlet/Collection_b?aurl=http%3a%2f%2fwww%2eacm%2eorg%2fsigchi%2fchi2002%2fpapers%2ehtml%23P%231%231%23**papers**%20%20**CHI**%20**2002**%7chttp%3a%2f%2fwww%2eacm%2eorg%2fsigchi%2fchi2002%2flocation%2ehtml%23P%233%231%23**location**%20%20**CHI**%20**2002&pagetitle=Chi**%20**Info** represents an AURL with 2 components (its headers and page title are in bold).

Each user can view and non-destructively edit the collection,

[1] http://www.eastgate.com/squirrel/FAQ.html

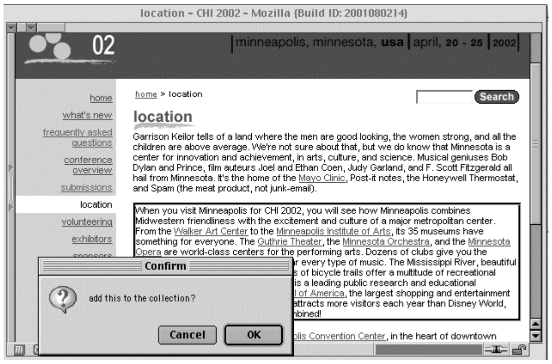
since editing only changes an AURL, and one user's changes to an AURL have no impact on someone else's. Versions of collections can be readily shared while working on an information-gathering project, either as a link in a collaborative Web page, or sent via email.

We note briefly two side effects of the AURL approach. First, HG references within-page-components by converting an HTML page to XHTML, an XML-compliant form of HTML. This process lets us use XML's Document Object Model to reference the elements marked up within a page. The referenced-based approach to collections makes collections dynamic. If a user includes a component for the local weather, each time the page is loaded, the user will see the latest forecast. In some cases, it may be necessary to construct methods to let users identify which components are important to be set as static and which can remain dynamic. For now, we are interested particularly in refining the interaction between component selection, capture and management rather than considering the long-term archival properties of a collection. Second, as a side effect of referencing components by AURLs rather than by coping content, Hunter Gatherer embodies a version of Nelson's Transclusions [7]. Translusions propose creating and publishing hypermedia documents by reference in part so that authors can control both private and public organization and publication of information resources. One could imagine a method of extending Hunter Gatherer to support authorizing Web sites/pages/components for publication within public or private collections

Interaction Technique

Selection

There are three steps to collect a page component in Hunter Gatherer: (1) select the component to be collected (Figure 3); (2) with that component selected, press the "a" key; (3) a dialog box appears (Figure 3) asking if the user wishes to add the component or not. The user can click "ok" or press the return key to approve the collection. We plan to make this last step part of a user's tool preferences, since in our design reviews, some users wish to be asked to confirm a selection; others do not. The current default is to ask. The user can continue to add components in this manner. Any component that can be displayed in a Web page can be added to a collection, from images to applets.



location - CHI 2002 - Mozilla {Build ID: 2001080214}

location

Garrison Keilor tells of a land where the men are good looking, the women strong, and all the children are above average. We're not sure about that, but we do know that Minnesota is a center for innovation and achievement, in arts, culture, and science. Musical geniuses Bob Dylan and Prince, film auteurs Joel and Ethan Coen, Judy Garland, and F. Scott Fitzgerald all hail from Minnesota. It's the home of the Mayo Clinic, Post-it notes, the Honeywell Thermostat, and Spam (the meat product, not junk-email).

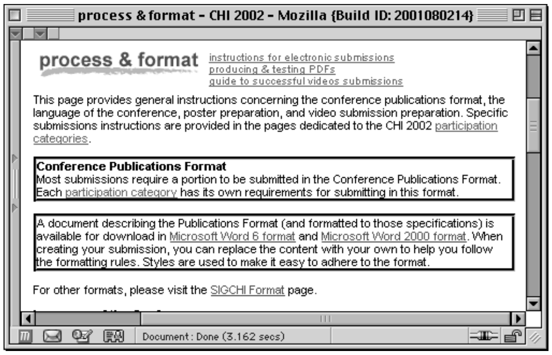
When you visit Minneapolis for CHI 2002, you will see how Minneapolis combines Midwestern friendliness with the excitement and culture of a major metropolitan center. From the Walker Art Center to the Minneapolis Institute of Arts, its 35 museums have something for everyone. The Guthrie Theater, the Minnesota Orchestra, and the Minnesota Opera are world-class centers for the performing arts. Dozens of clubs give you the r every type of music. The Mississippi River, beautiful s of bicycle trails offer a multitude of recreational is a leading public research and educational l of America, the largest shopping and entertainment attracts more visitors each year than Disney World, nbined!

add this to the collection?

lis Convention Center, in the heart of downtown

Figure 3. The user has selected a component (indicated by the border around the selection) and hit the "a" key to add the component to a collection. The dialog box then appears to ask the user to confirm the addition.

The selection and add process is relatively transparent. It does not require the user, after selecting a component, to shift attention from the browser to an editor application, paste content into that application's file, go back to the browser, copy the URL, go back to the editor, paste the URL, add a note to contextualize the component, save the file, go back to the browser and refocus on hunting for the next component. The user simply identifies a component to be added; the system manages the other steps so that users can focus their main attention on a task until *they decide* to shift that focus to a different task.



process & format - CHI 2002 - Mozilla {Build ID: 2001080214}

process & format
instructions for electronic submissions
producing & testing PDFs
guide to successful videos submissions

This page provides general instructions concerning the conference publications format, the language of the conference, poster preparation, and video submission preparation. Specific submissions instructions are provided in the pages dedicated to the CHI 2002 participation categories.

Conference Publications Format
Most submissions require a portion to be submitted in the Conference Publications Format. Each participation category has its own requirements for submitting in this format.

A document describing the Publications Format (and formatted to those specifications) is available for download in Microsoft Word 6 format and Microsoft Word 2000 format. When creating your submission, you can replace the content with your own to help you follow the formatting rules. Styles are used to make it easy to adhere to the format.

For other formats, please visit the SIGCHI Format page.

Document : Done (3.162 secs)

Figure 4. Component Selection. As the user holds the control key and drags the cursor over the page, available components are indicated by borders appearing around them. By holding the control and shift key, users can select multiple components.

When the user first selects a component to be added to a collection, a small window, the List/Edit view, will open (Figure 4). This window displays a list of the components in the collection and allows a user to monitor the growth of that collection. Because the window can be moved behind other windows, or arranged to be peripherally available, the user can determine the degree to which they wish to monitor the collection: each time they add a component after a collection has been initiated, the List/Edit View window does not come to the front, but stays where placed. Indeed, in the first design review of the prototype with 26 evaluators [9], the ability to adjust the "focus" of List/Edit View to monitor collection state was seen to be essential. As an example, Figure 1 shows the List/Edit view visible beside the main browser window, making the state of the current collection peripherally available. By having the list available in this way, the user can be confident that another component has been added to the collection by a fast glance at the window.

Prototype Selection Note. Because we use the XML Document Object Model to locate and reference components within a Web page, we have 2 methods to identify components for selection: one is by page element, such as a paragraph, indicated by the XHTML tags like `<p></p>`. The other is to identify entities *within* elements, so that in `<p>some text </p>` a user can select, for instance, the last "e" of "some" and the first "t" of text. This latter method emulates the act of highlighting a portion of a Web page for copying. In the current iteration of Hunter Gatherer, we have had success with selecting and addressing entire components, like paragraphs or table cells. We have discovered a number of incompatibilities across systems for within-element text selection, so have temporarily taken this approach off line. This is why the visual feedback for selecting a part or parts of a Web page is indicated by borders around elements rather than by highlighting. As users, we are used to interpreting highlighting as something that can be edited to a fine-grained level. Since we cannot yet support this degree of selection fully, we have opted to use borders to indicate what is selectable, since they are less likely to be interpreted as being as refinable as highlighting. When we have character selection back on line, we will evaluate whether we should keep both modes of indicators: highlighting and bounding boxes, or simply highlighting.

Collection Interaction

If the user wishes to move task focus from adding components, to the collection, to dealing with the collection itself, they can do so via the List/Edit View (Figure 5). This window for monitoring collection state also acts as the editor palette for the collection. Users have several editing options available: they can rename a component, sort components in the list, delete components from the list, give the collection a title and preview the collection in a browser window.

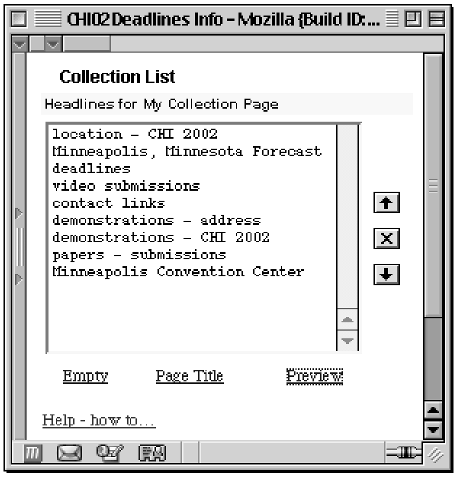
Collection View

When the user selects Preview from the List/Edit View, a new browser window opens, displaying each of the components represented by the list, in the order in which they are displayed in that list. With both List/Edit View and Collection View open, as in TopicShop, users have two ways to visualize the collection. As shown in Figure 2, each component appears with an automatically generated header: the title of the component's source Web page. The component also appears in the collection with the source page URL as a link. At any time, the user can click that link to open the source page for that component. Likewise, any links within the captured component behave just as they would in the component's source page.

Direct Manipulation. At present, users manipulate collections via the List/Edit view. We do not believe that this is optimal, and certainly related research, especially the TopicShop and Spatial Hypertext work, indicates that for the collection management part of the interaction, direct manipulation of components has value over list views *alone*. We are developing a way to integrate direct manipulation into the tool.

Gradations of Interaction: Focus

Throughout the collection making process with Hunter Gatherer, the user can move among hunting for sources, selecting components from those sources, adding those components to a collection, editing the content of a collection, previewing the collection, and saving a version of the collection (by making a bookmark, for instance, of the current collection AURL). If the user at a later point wishes to return to a collection, they load its AURL, which may be done by selecting a bookmark for a collection or by pasting the AURL from an email message into the browser's Location. To edit the collection further, the user clicks the "edit" link from the collection page, and a List/Edit View window of that collection opens, listing all its components. The user can continue to view or revise that collection. By having all views as browser windows, the user determines which part of the collection making activity they wish to foreground, keep in the background or have peripherally available, simply by arranging the browser's windows.



CHI02 Deadlines Info - Mozilla {Build ID:...

Collection List

Headlines for My Collection Page

```
location - CHI 2002
Minneapolis, Minnesota Forecast
deadlines
video submissions
contact links
demonstrations - address
demonstrations - CHI 2002
papers - submissions
Minneapolis Convention Center
```

Figure 5. List/Edit View of a Collection. This collection contains components from multiple Web sites. Users can sort, delete or rename components listed, and rename or preview the collection.

EVALUATION

In order to asses how Hunter Gatherer meets the requirements for collection, focus shift and continued value, we initiated 3 sets of related evaluations: a survey to better quantify our understanding of existing Web-based information-management practices, an experiment to assess the tool's efficiency, and a field study to gain insight into how a new way of working with Web-based information may fit into daily practice.

Survey

Design and Methodology

In the survey, we asked participants to report on their Web use in three main areas: creating bookmarks, copying and pasting from the Web, and printing Web pages. In each case, we asked about perceived frequency of practice and under what circumstances users initiated the activity. In the copying/pasting section, we asked about copying and pasting from Web pages into email and into text editors, and under what circumstances each was done. In questions asking for frequency of practice, we used a Likert scale, ranging from "never" to "always;" in other questions, such as kinds of pages considered for bookmarking, we presented a list of predetermined topics from entertainment to research. We also collected demographic data about the participants.

We ran the survey with 77 participants. The participants were from a first year Computer Science summer session course. Most participants were 18-24 years old and had comparable sophistication with Web use. All but one participant had a computer at home. Having this commonality in background let us evaluate differences in Web practice *within* a Web-savvy group. Analysis of this constrained demographic sample will help us refine an iteration of the survey for a random sample population, as well as provide benchmark results for that iteration.

Significant Results Overview

While more complete results and copies of the survey are available at [10], we report on some of the most significant effects below with respect to Hunter Gatherer.

Bookmark Use. Out of 77 participants, only 5 reported spending some or all of their Web time looking at pages being visited for the first time. Similarly, 51 participants reported spending most of their Web time looking at pages visited frequently. 32 participants reported spending most of their time on the Web for the purpose of entertainment. The second most popular Web activity was personal research, with 16 participants reporting this as their most frequent activity.

Gender. In a one-way ANOVA, we saw that for the subjects who reported copying URLs into e-mail, there was a significant effect of gender (F = 5.564, p < .021). A significantly larger proportion of men (30%) than women (4.8%) reported copy/paste of URLs for “pages that may be of interest, even if the content changes,” as opposed to “pages that contain specific information you wish to share.”

Analysis

As the Abrams *et al.* work shows [1], users make enough bookmarks to make collections unwieldy. Additionally, our study shows that, whatever the size of the complete bookmark set, users report visiting only a constrained and regular subset of those bookmarks. More particularly, only infrequently do users visit “new” pages. This raises questions about how a new bookmark or new site becomes part of that regular subset, how this practice interacts with search engine use, and how Hunter Gatherer may enhance that process through collections which represent, in one page, a set of valuable components from multiple sources.

The finding that men are more likely than women to mail URLs referencing general, possibly changing, information rather than URLs for specific content indirectly supports the rationale for HG’s design for saving and sharing collections in AURLs. First, the finding implicitly indicates that users *do* use URLs in email to share information. Second, it shows that URLs are used to share at least two kinds of information. Based on this practice, Hunter Gatherer’s AURL approach extends rather than changes existing practice for information sharing. HG Aggregated URLs reference *collections* rather than an URL’s single page. AURLs can also blend references to both specific information (e.g. quotations) and to more general, dynamic information (e.g. stock reports). The finding with respect to information practices suggests that gender may also be reflected in tool use. We will evaluate for this effect in future studies.

Experiment

Design and Methodology

We set up a 2x2, within-subjects study to test the efficiency of Hunter Gatherer compared to an editor for creating collections. To reduce learning curve noise in the data for the editor-based collections, we choose Microsoft Word as the most familiar editor among participants. The first factor in the experiment was tool (Hunter Gatherer vs. Word); the second factor was data set (Web pages on a Chemistry program; Web pages on a Physics program). We first ran a pilot study with five participants, refined the protocol, and ran the formal experiment with 12 participants, representing a mix of technical and non-technical undergraduate and graduate students at the University of [Anonymous].

At the start of the evaluation, users were given 15 minutes training time with Hunter Gatherer. Users were then asked to build two collections, each from a given set of bookmarks *to be clear enough to be used by someone else.* This direction was motivation to use the tools’ editing capability to create the most effective collection possible within the time constraints. We alternated which tool a participant would use first, Word or Hunter Gatherer. To reduce potential learning effects, we prepared two similar collections of bookmarks, one on the Chemistry program and one on the Physics program at University [Anonymous]. The pages for each set were taken from the same general Web sites, so that pages were similar but for content. Participants were given 5 minutes with each set of 3 bookmarks to familiarize themselves with the content of the pages. Participants were then given 15 minutes to build a collection from the bookmarks that would explain how to get a minor in the given subject, list and describe the required courses, and show the course instructors for the term. The experiment let us test HG in terms of our 3 requirements: (1) the efficiency of component addition (2) the effectiveness of HG in the complete collection making cycle (3) the immediate legibility of the resulting collection.

Empirical Results

A one-way within-subjects ANOVA showed a significant effect of tool type (collection time (F = 5.730, p < .040) in comparing average component collection time using HG and Word. Participants required an average of 6.70 seconds using HG and an average of 10.9 seconds using Word (Figure 6).

Observations

General Observations. First, despite practice with the Hunter Gatherer tool in which we also demonstrated that each component captured contained a default header and source page URL, only 3 participants, when using Word to build a collection, included the URL of the source page for a given component. The collections, on average, had over a dozen components. The participants who included URLs did so for only a few components, and each of them had used Hunter Gatherer as their first collection making tool. *Word-specific Observations.* In creating collections in Word, many participants over-captured the information required from the Web page, and then edited the extra material out from the collection file. Also in editing, Word was more efficient than Hunter Gatherer for revising component headers. Headers in Word could be edited directly in the collection, whereas Hunter Gatherer requires participants to move to the List/Edit view to enter a dialog box to make a change.

HG-specific Observations. In the post-evaluation questionnaire, most users reported that they would prefer highlighting components to collect them, in addition to having the bounding box as the only method for component selection. Participants also commented that sorting components in collections was "easier" in Hunter Gatherer than in Word. Similarly, in being asked what the best feature of Hunter Gatherer is, 10 out of 12 users reported it as the automatic capture of the component's URL.

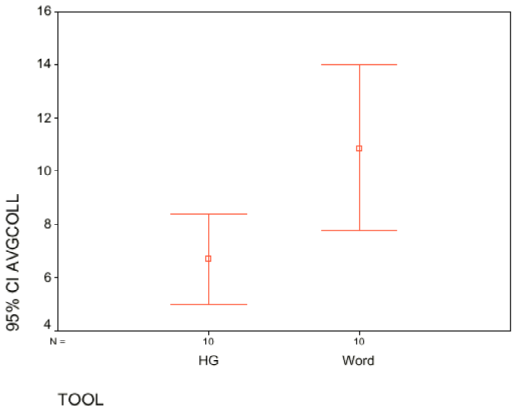


Figure 6. Hunter Gatherer significantly more efficient in component addition than Word.

Analysis

We have met our first design requirement to make the addition of a component as efficient as copying text from a browser. Though participants expressed a desire to have highlighting as a selection method, HG selection performance was significantly better than with Word. The Hunter Gatherer method is also more effective than Word for component addition, since HG automatically adds both a header for the component and the URL for the source page, the latter addition indicated by users as the most valuable attribute of the tool.

We have only partially met our second design requirement to support user-determined focus shift among collection tasks. Header editing in HG forces users to concentrate on the tool, rather than the task: double clicking a header title in List/Edit view and ok'ing a change in a header dialog box is less transparent than editing the header directly in a file. While we would need to test this particularly, our observations also indicate that users want to be able to make a first pass at component selection, and then edit the components further, after they have been collected. We may be requiring users to focus too much on precise component selection when they would rather be focusing on faster, more general initial "information triage" as described above, and edit further, later. In our Future Work section, we propose several modifications we wish to test to address these issues.

With the automatic capture of the URL, we have partially met our third design requirement for automatic capture of enough information for the collection to continue to be useful to the user. Hunter Gatherer's default component headers, however, need refinement: they are currently only the title of the source Web page. When components come from multiple pages, the title is sufficient to identify the component for the user. In our test, with many components coming from a handful of pages, we saw that, in List/Edit view especially, one component became indistinguishable from another. This lack of distinction renders parts of the collection immediately less useful. We address this concern in the Future Work section as well.

Field Study

Design and Methodology

Andrew Dillon, discussing Process, Outcome and Affect as alternative evaluation measures to Effect, Efficiency and Satisfaction, suggests that the *affect* of a design – whether a user experiences the interaction as empowering or frustrating – is critical for understanding and improving the interaction design [5]**.** With Hunter Gatherer, we are interested to know if, given an efficient and effective interaction, the tool itself will become an *affective* part of Web information practices. To begin to answer this question, we have followed Dillon's suggestions for evaluating affect: we have given the tool to users to explore "free style." Participants have been asked to try the tool over a month, to answer a weekly set of questions about their tool use, and to share example collections made during that time. Participants learned how to use the tool via a Web page, describing its features and known bugs. There were 14 participants from a wide variety of disciplines. Each identified themselves as "tolerant" of alpha software.

Overview of Results and Analysis

All but one participant reported that they like the tool and make collections with it. Consistent with the findings in the survey which suggest that users rely on a frequently visited constrained set of bookmarks, participants did not make many random collections, but made a few deliberately, for specific purposes. Those purposes are as distinct as the users. One participant has made a collection of components from a variety of financial information sites, which, he states, he consults daily, since the components are dynamic and he wants only current financial information. Another participant in Medicine has a collection on a particular disease profile that he wishes to publish for participants at an upcoming conference. Another has gathered components for course lecture notes. The intent of these collections indicates that they will have a relatively long shelf life and possibly high return use. Only one participant, a reporter for a national television network, indicated making collections for shorter term purposes, to collect background sources for upcoming stories. We will quantify both collection making and return to collection rates in the follow-up study.

When asked specifically if the concept of making collections from within-Web-pages had become a technique that was now part of their way of thinking about managing Web-based information or not, most users responded that the tool/concept had indeed become part of their way of thinking about gathering information on the Web. Only one participant reported discovering that he did not find a need to make within-Web-page collections. Indeed, many of the participants regularly emailed design suggestions that would make the tool more effective for them. These reports reiterated our experiment findings: better header support and highlighting for component selection. Further, we completed our survey after initiating our field study. Based on the unexpected significance of gender found in that survey, we plan to ensure that the next field study is gender balanced.

CONCLUSIONS AND FUTURE WORK

The data collected so far, both empirically and anecdotally, indicates that Hunter Gatherer is a promising concept. User response has been positive. We wish to improve the HG interaction in terms of our second and third design goals, in particular: user-determined attention shift and viability of default collection representation. On the basis of the above results, we are in the process of making revisions to Hunter Gatherer in the following areas:

- *Selection*: Highlighting at both component and within-component levels; supporting non-contiguous selection.
- *Headers*: For the default component header - page title *and* key words from the component.
- *Editing*: Editing headers directly in the collection view; deleting elements of a component in the collection view.

These revisions should help to satisfy our first research goal: to support within-Web-page component collection making. Our second research goal, as stated above, has been to consider how supporting this interaction may affect users' Web interaction practices. The evaluations reported here have primarily helped with the first goal. To address the second, after the proposed revisions are completed, we plan to run a gender-balanced field study over several months, specifically to evaluate interaction affect for both individual and, what we have not looked at so far, collaborative Web information management practices. In parallel, we are also investigating how Hunter Gatherer's approach to document-creation-by-reference may be used as part of collaborative digital rights management.

REFERENCES

1. Abrams, D., Baecker, R., Chignell, M. Information archiving with bookmarks: personal Web space construction and organization in Conf. Proc. on Human Factors in Computing Systems, 1998, pp. 41–48.

2. Amento, B, Terveen, L., Hill, W., Hix, D. TopicShop, enhanced support for evaluating and organizing collections of Web sites in Proc. of the 13th Annual ACM symposium on User interface software and technology, 2000, 201–209.

3. Card, S K., Robertson, G. G., York, W. The WebBook and the Web Forager: an Information Workspace for the World Wide Web. Conf. Proc. on Human Factors in Computing Systems (Vancouver, Canada, April 13-18, 1996), 111.

4. Czerwinski, M., van Dantzich, M., Robertson, G.G., Hoffman, H. The contribution of thumbnail image, mouse-over text and spatial location memory to Web page retrieval in 3D. Sasse A. & Johnson, C., Eds. HCI--Proc of Interact '99 (Edinburgh, Scotland, 1999), IOS Press, 163-170.

5. Dillon, A. Beyond Usability: Process, Outcome and Affect in Human-Computer Interaction. Presentation to Faculty of Information Studies University of Toronto, 23/03/01.

6. Marshall, C. C. Shipman, F.M., Coombs, J.H. VIKI: Spatial Hypertext Supporting Emergent Structure. Proceedings of the 1994 ACM European Conf on Hypermedia Technology, 1994, 13–23.

7. Nelson, T. H. "A Literary Structure with Two Fundamentally Different Means of Connection." Xanalogical Structure, Needed Now More than Ever: Parallel Documents, Deep Links to Content, Deep Versioning and Deep Re-Use. http://www.sfc.keio.ac.jp/%7Eted/XUsurvey/xuDation.html.

8. Robertson, G., Czerwinski, M., Larson, K. Robbins, D., Thiel, D., van Dantzich, M. Data Mountain: Using Spatial Memory for Document Management. Proc. of UIST '98, 11th An Sym on User Interface Software and Technology, ACM Press, 153-162.

9. schraefel, m.c. and Zhu, Yuxiang. Preliminary Requirements Gathering for the Design of User-determined, Within-page, Web-based Collections. Tech Report, CSRG-433, DCS, U of Toronto, 2001.

10. schraefel, m.c., Wigdor, D., Modjeska, D. Results of a Survey on URL Saving and Sharing. Tech Report, forthcoming, DCS, U of Toronto, 2001.

11. Terveen, L., Hill, W., Amento, B. Constructing, Organizing, and Visualizing Collections of Topically Related Web Resources. ACM Trans. Comput-Hum. Interact. 6, 1 (Mar. 1999), 67-94.

12. Wickens, C. D., Hollands, J.D. Engineering Psychology and Human Performance, 3rd Ed., Prentice Hall, 2000.

Note,

There is also a demonstration video, schraefel_HG01.mov, in the same directory as this report.

It is a Quicktime video, requiring the free Apple Quicktime player (http://www.apple.com/quicktime/download/).

The location for this technical report, via anonymous FTP
ftp://ftp.cs.toronto.edu/csrg-technical-reports/437/