Proceedings of the International Conference on Information Technology (ITCC 2004), Las Vegas, April 5-7, 2004.

Toward Dialogues With Documents: MultiBrowser

Daniel Berleant, Jinghao Miao, Zhong Gu, and Dongping Xu
Iowa State University
2215 Coover Hall
Ames, Iowa 50011, USA
(515) 294-3959
{berleant, jhmiao, zhonggu, xdp}@iastate.edu

ABSTRACT

When several information items are presented simultaneously, the potential exists for users to interact with information in a rich and rewarding way. The MultiBrowser system supports such interaction via an information foraging style of hypermedia browsing. Multiple windows, colored bars that provide visual cues, and automatically inserted hyperlinks among the paragraphs in a document set contribute to the structure of repositories created by MultiBrowser. These repositories are browsable over the Web using an ordinary Web browser.

Keywords

Browsing, foraging, multidisplay, repositories, visualization.

1. INTRODUCTION

Consider a vision for interacting with texts: a repository of text that, together with a suitable inference engine and user interface, acts as a discussion partner for the user. The user asks a question or makes a comment, and the system response is the most appropriate passage in the repository. The system response might instead be derived from the user interface rather than directly from the text itself, such as a request for clarification or some other meta-comment. A response could also contain both a passage and a meta-comment, such as a description of the relevance of the passage to the question. This would be useful for example if the relevance was analogical. The system would allow a user to have a "conversation" with a text.

Taking this idea further, a system for dialoguing with documents could provide several responses to a user's input simultaneously. Why be limited to just one? Using such a system would be roughly analogous to having a discussion with a board of experts. In this case the expertise resides in the combination of the repository and the document dialogue system.

Although it seems unlikely that a system attaining this vision will be constructed soon, it can serve now as motivation for experimental systems that move in the direction of that goal. This report explains an architecture and an implemented system that provides a starting point. The system, MultiBrowser, responds to user actions with multiple passages from a repository of hypertexts. This strategy is a potentially useful way to support the preference of readers of on-line text to scan and pick out passages rather than reading in a traditional sense (Nielsen 1999 [20]). The style of interaction that occurs in MultiBrowser is information foraging (Pirolli and Card 1995 [22], Pirolli 1998 [21], and Wexelblatt and Maes 1999 [28]), rather than information retrieval. Information foraging is a relatively undirected kind of browsing. Foraging is an appropriate model of behavior when goals are vague, a situation that often occurs in practice, as when one "surfs" the Web with a "browser," "skims" a book, or "browses" library shelves. Such foraging activities can play an important role because the traditional information acquisition model of information retrieval with highly focused goals may not always be the most appropriate. In MultiBrowser, users browse by foraging within an automatically generated hypermedia structure that links paragraphs that are automatically determined to be similar.

2. BACKGROUND: MULTI-WINDOW DISPLAYS

A key goal mentioned earlier, for a system to provide multiple responses simultaneously, can be met with a multiple window architecture. The term "collage" for such a display seems to have originated with Kaltenbach and Frasson (1989 [11]), who describe a system for presenting mathematical proofs which was later extended to presenting hypertext (Kaltenbach et al. 1991 [12]). In MultiBrowser, clicking on a hyperlink brings up a collage of several non-overlapping, hypermedia-containing windows (Figure 1). Thus hyperlinks are conceptually one-to-many. Each collage provides a view of the repository.

Unlike the Avant Browser (www.avantbrowser.com), the MultiBrowser windowing interface is defined strictly within the repository itself, using standard HTML commands. HTML <frameset...>...</frameset> commands are used to specify windowed text for display in a standard browser. Also unlike the Avant Browser, windows in MultiBrowser displays are non-overlapping.

At least two studies suggest that the non-overlapping, space-filling windows strategy is preferable to overlapping windows for displaying related texts. In one, Bly and Rosenberg (1986 [1]) showed that such a windowing strategy supports faster identification of task-relevant paragraphs than an overlapping window strategy when it is possible to present the bulk of the source material without overlapping. In another, Kandogan and Shneiderman (1997 [14]) investigated completion times for 21 task conditions, finding that their "Elastic Windows" windowing method enabled faster task completion than overlapping windows.

Collage-based displays are supported by a long tradition in hypertext systems. The NoteCards system (Halasz 1988 [7]) and the Sun version of Hyperties (Shneiderman 1987 [23]; Lifschitz and Shneiderman 1987 [16]) are among the early such systems.

MultiBrowser uses collages that are tiled (space filling and without overlaps). Some notable other works using such collages include the Krakatoa Chronicle online newspaper (Kamba et al. 1995 [13]), the VOIR interface which, like Krakatoa, is based on the newspaper metaphor (Golovchinsky and Chignell 1997 [6]), and Elastic Windows (Kandogan and Shneiderman 1997 [15]).

The design of MultiBrowser builds on the multiple windows idea with novel features. Each window in a MultiBrowser display is associated with a color bar. The color bar summarizes the location, in the space of all documents in the repository, of the document for which the window provides a view. The motivation for color bars is to provide quickly and easily perceived cues about what documents are most similar to what other documents. Another novel feature is the hyperlink associated with each paragraph which, when clicked, brings up a display of windows. This hyperlink is annotated to show how many of the target windows will be views into documents that are in a *kernel* of core documents in the repository, how many are *close to* the kernel, and how many are *peripheral*.

The next section gives some details on how MultiBrowser works. The following section analyzes the multiple windowing approach in MultiBrowser. The subsequent section examines the color bar technique used and the user studies that were performed. The hyperlink annotations are discussed in detail in another report (Miao and Berleant, submitted [18]).

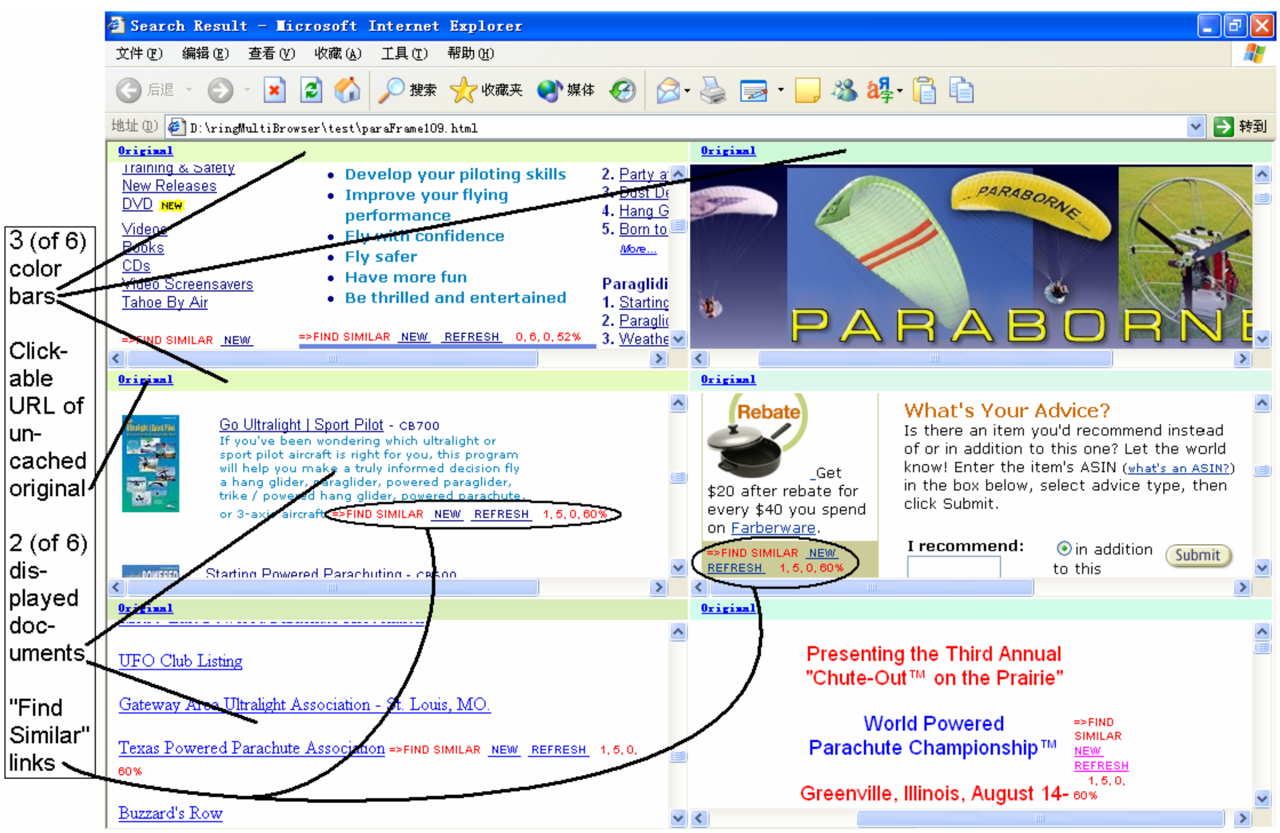

Figure 1. Sample MultiBrowser screen from a repository created from documents obtained by the Web search engine query "powered parachuting" (no endorsement of any company is intended). KEY: the 4 numbers associated with each FIND SIMILAR **link state, for its associated paragraph, how many of the six windows it points to will contain documents with, respectively, many incoming paragraph links, a modest number (*n*), few (*f*), and a composite "%" ranking equal to** 100-8*n*-16*f*.

3. DIRECT MULTIDISPLAY IN MULTIBROWSER

MultiBrowser constructs hypermedia repositories that are structured to support information foraging using a standard Web browser. MultiBrowser takes as input either a list of URLs, or a search engine query which it passes to a Web search engine to get a list of URLs. It then constructs a repository from the listed documents, in two phases as follows.

Phase 1: Document-Scale Processing

MultiBrowser begins by clustering the documents into three groups, using the standard k-means algorithm. The distance metric used is the cosine measure in the space of 5-grams, strings of 5 characters (Damashek 1995 [4]). Mayfield and McNamee (1998 [17]) compared this metric with use of full words and found that 5-grams were competitive (fig. 2 of [17]).

For each document, the inverses of its distances to the centroids of the three clusters, arbitrarily labeled the red, green, and blue clusters, are used to compute intensities of red, green, and blue (RGB) components of a composite color. The composite color is presented in a color bar directly above the document when it is displayed in a window (Figure 1). The color bar provides a summary of the document's location in the space of documents. That in turn allows the user to visually compare its similarity to documents in other windows by comparing their color bars, which they may wish to do in deciding which window to read next. Each color bar also contains a hyperlink that, when clicked, loads the corresponding document into the full browser frame for a closer look. (Further discussion regarding color bars and related work is given in Section 5.2.)

Phase 2: Paragraph-Scale Processing

After a color bar has been computed for each document, the documents are segmented into paragraphs and the set of paragraphs is processed. As in the case of documents, each paragraph is mapped to a normalized point in 5-gram space, and points are compared using the cosine similarity metric. For each paragraph, the most similar other paragraphs are identified, regardless of what documents they are in, so that later the display can show the paragraph and the other paragraphs most similar to it, each in its own window. This will occur in response to a user click on a "FIND SIMILAR" link (Figure 1).

Immediately preceding each paragraph an anchor (HTML <a name=…> tag) is inserted. Immediately after each paragraph a "FIND SIMILAR" link is inserted which points to a unique HTML file containing a <frameset...> tag specifying three, four or six windows, depending on the value of the *windowNum* configuration parameter provided to the system at repository creation time. Figure 1 shows the case of six windows. The number of windows is specified because presumably the optimal number is not the same under all circumstances; research is needed regarding the optimal number of windows given the display and information environment. Each window displays a document starting at one of the anchors preceding a paragraph in the document. The top left window redisplays what was in the window containing the "FIND SIMILAR" link that was just clicked. The other *windowNum*-1 windows display documents starting at the anchors preceding each of the *windowNum*-1 paragraphs in the repository that were rated as most similar to the one whose "FIND SIMILAR" link was clicked. Thus each link in essence has *windowNum* targets. The value of multi-tailed links was explored as early as the 1980s [25]. Each window is scrollable, allowing users to move to anywhere in its document, and at its top is the color bar computed for the document.

The system is designed to be accessed using a standard Web browser. If the user clicks "NEW" in the FIND SIMILAR link, a new set of windows is shown in a new browser frame that pops up over the current browser frame, thereby caching the previous browser frame on-screen behind the new one for easy access later. If instead the user clicks "REFRESH" in the FIND SIMILAR link, the new set of windows replaces the current display, which is moved to the browser's history list. (History navigation has been shown to be valuable [3][26].)

A repository cannot currently be created in real time. However, it can be created overnight, over a lunch break, etc., so that in cases where this delay is acceptable a custom repository can be created. Alternatively, WWW *alcoves*, pre-computed repositories on specialized topics, can be created and placed on the Web. Ultimately, real-time repository creation would be useful. One approach to this would be to allow a user to start foraging after just one or two documents have been incorporated into the repository, while simultaneously adding more documents at the same time as the user continues foraging.

4. FORMALIZING THE INTERFACE

This section formalizes the intuition that simultaneous display of *n* items rapidly increases the information retrieval precision of the overall display for even modest values of *n*. The concept of information prefetch is then used to further analyze the MultiBrowser user interface design.

4.1 Multidisplay, Monodisplay, and Precision

Define *multidisplay* as *a non-overlapping, space-filling collage of two or more presented information items.*

Define the opposite of multidisplay as monodisplay:

a *monodisplay* is *the presentation of one item.*

While multidisplays show more than one item at a time, monodisplays can show more of a single item in the same display area. Each can be preferable depending on circumstances, hence MultiBrowser's support of multidisplay and monodisplay (Section 3).

4.1.1 Formalizing multidisplay

Let P_1 be the probability that the most likely item to be of interest to the user, is. For example, if the items are displayed because they match a query, given a reasonable ranking algorithm this would be the one with the highest rank. The probability that this item is *not* of interest is therefore $1-P_1$. Similarly, let P_n be the probability that the nth most likely item to be of interest, is. The probability that it is not is thus $1-P_n$. Then

$p(\textit{no displayed item is of interest}) = (1-P_1)(1-P_2)...(1-P_n)$.

Define P as

$$P = p(\textit{at least one displayed item is of interest}) = 1 - (1-P_1)(1-P_2)...(1-P_n) \quad (1)$$

P rapidly approaches 1 as n increases, under the assumption that all of the displayed items have a relevance probability that is not near 0. Call this the non-small precision assumption. The non-small precision assumption will usually hold for P_1 because content is usually viewed because of its likely interest. On the other hand if few relevant items exist, then P_n might be close to 0 for n above some value k. Then a multidisplay of over k items would not be helpful.

Under the non-small precision assumption, the multiplicative character of equation (1) implies that increasing n by even a little moves P, the chance that at least one displayed item will be of interest, substantially closer to 1. This will be significant for users, unless P is already near 1 (which is a desirable situation).

Example 1. Suppose $P_1=0.4$, $P_2=0.35$, $P_3=0.3,\ldots$. Then by equation (1), increasing n from 1 to 2 increases the probability of an item being of interest from .4 to .61, over 50%, while increasing n from 6 to 7 has little to recommend it since this would only slightly increase the probability of an item being of interest, from .861 to .875 or just 1.6%.

Conclusion. The formalization, and *Example 1*, show that displaying a modest number of items is sufficient for most values of P_n.

Potential application. Search engines could improve the usefulness of their returned lists by varying the number of returned items that are displayed based on their quality. In cases where fewer items are displayed, more information about each could be presented.

4.2 Direct & Indirect Display, and Prefetch

Define *direct display* as *the presentation of actual content.*

Direct display contrasts with indirect display:

define *indirect display* as *the presentation of metadata.*

Metadata, a library science term, refers to brief characterizations whose principal use is facilitating access to more detailed information (Smith 1996 [24]). An example of metadata is the title+passage material in a Web search engine return list. A similar concept in the IR domain is sometimes described as *document surrogate material* (Hearst 1999 [9]). The direct and indirect display concepts can be extremes of a continuum, as shown by the following examples.

- If only part of an information item, such as a long document, can be displayed then that direct display is also an indirect display describing the rest.
- Blurring the distinction between direct and indirect display may be a design objective, as in Zellweger et al.'s (1998 [29]) fluid links.
- User needs can vary so that, for example, the display of a bibliographic reference is indirect if one needs the material itself, but direct if one merely needs to know its author.

Typically direct display gives immediate access to content, while indirect display provides abstraction and summarization.

4.3 Direct Display and Multidisplay Combined

A display can be mono- or multi-, and simultaneously direct or indirect, making four combinations (Table 1).

	Direct	**Indirect**
Mono-	Typical editing of a document	"Press Ctrl-Alt-Delete to begin."
Multi-	Displays of several windows	Directories; presentations of lists of links

Table 1. Examples of direct monodisplay, indirect monodisplay, direct multidisplay, and indirect multidisplay.

Direct multidisplay is most useful when there is both a limited number of items to display, and sufficient room to display them. Increasing display capacity, which makes it possible to show more and larger items, is possible because ongoing trends of improvement in network bandwidths and display hardware allow ever higher resolutions at lower cost [27]. For example, electronic paper has recently been marketed and electronic wallpaper is becoming plausible as a future technology [10]. Trends in display hardware capacity are significant because even modest %/year increases imply availability of displays with several times the capacity of present displays in the not-too-distant future. Thus the appetite of direct multidisplay for display space will become less of a problem over time. At the same time the summarizing function of the metadata in indirect displays, which is cognitively important, is feasible to retain: it takes relatively little space, so simply present it along with the direct display it corresponds to. These considerations suggest a trend toward use of direct multidisplay, at least until human perceptual capacity becomes a bottleneck.

4.4 Direct Multidisplay and Prefetch

One use for direct multidisplay is reducing the number of interface actions. Actions such as moving and clicking a mouse take time and attention that could otherwise be put to other uses, and a multidisplay requires fewer such actions than a monodisplay. For example, in common Web search engines, accessing the content of some URL in a return list requires, (1) inferencing about the content from the metadata in the list, and (2) another click to view the item chosen. To view a second item requires (3) clicking the "back" button on the browser to return to the list, then repeating steps (1) and (2). After viewing a few items, returning to a previously viewed one that needs to be viewed again, an event familiar to many readers, annoyingly requires either remembering which one it was, or rechecking one or more to find it again. The need for such interface actions should be minimized. The following analysis explores *prefetch*, one way that direct multidisplay can conserve user interface actions.

Suppose a system displays at least two items and, after the user views one, views a second. Let this be called a *prefetch,* because the second viewed item was displayed before the first was read. Prefetch can only occur when direct multidisplay is used, because with monodisplay a second item is not even displayed, and with indirect display no item(s) at all are displayed, only their metadata. We wish to characterize the likelihood of successful prefetch in a direct multidisplay.

Let n be the number of items in a multidisplay. *Prefetch* occurs when two things are true: (1) some displayed item is of interest now, and (2) upon finishing with that item one of the remaining $n-1$ displayed items will be of interest. Given some item currently of interest in a multidisplay, define Q_2 as the probability of the most likely of the remaining $n-1$ items to be of interest after the viewer finishes the current item. The subscript "$_2$" is used instead of "$_1$" because a reasonable guess for the value of Q_2 would be that it is close to P_2 as defined in Section 4.1.1. Define Q_3 similarly as the probability of the second most likely of the remaining items to be of interest after the viewer finishes the current item, and analogously for $Q_4 \ldots Q_{n-1}$. Let

P_{none}
$=p(\textit{none of the remaining } n-1 \textit{ items will be of interest next})$
$=(1-Q_2)(1-Q_3)(1-Q_4)\ldots(1-Q_n)$.

Therefore

$p(\textit{prefetch occurs} \mid \textit{a first item is of interest})$
$=1-P_{none}$
$=1-(1-Q_2)(1-Q_3)(1-Q_4)\ldots(1-Q_n)$. (2)

The last step is to remove the condition that a first item is of interest, which has probability P. This is done by multiplying by P (see Equation 1), giving Equation (3).

$p(\textit{prefetch occurs})$
$=P[1-(1-Q_2)(1-Q_3)(1-Q_4)\ldots(1-Q_n)]$ (3)

As in equation (1), the non-small precision assumption implies that increasing the value of n offers initially large but rapidly diminishing returns.

Example 2. Suppose P=0.861 (the result of *Example 1*). Suppose also that Q_2=0.3, Q_3=0.25, Q_4=0.2,....
For n=2, $p(\textit{prefetch occurs})$=0.861[1−(1−0.3)]=0.26.
For n=3 however,
$p(\textit{prefetch occurs})$=0.861[1−(1−0.3)(1−0.25)]=0.41.
This is a sizeable improvement of 58%. By comparison, the improvement from increasing n from 6 to 7 is only 2.4%.

Since increasing n to high values will usually not result in much improvement, even a lengthy list of links, using indirect display to cram as many as possible onto a screen, would likely have only incremental benefit, and even then the prefetch is quite incomplete since only links and not content are presented.

The issue of space available per item. The more items displayed the less space is available for each. When this space is too small to display the interesting part of an item, or worse, enough of it to determine if it is in fact of interest, the gain of displaying more items competes with the loss of not displaying enough of each, making the display an *in*direct multidisplay, at least for that user.

One way to alleviate this problem is to be judicious about what in the item is displayed. For example the Google search engine displays only a phrase or so of each retrieved item, but these phrases are chosen for relevance to the query that retrieved them. Another approach would be to vary the size of the different windows depending on content so as to display enough to meet the user's needs. If the size of the portion of a document that should be displayed could be determined, then a generalized form of equation (1) could be stated and optimized. One approach to determining the portion to display would use automatically derived theme shifts [19] that define natural boundaries of document segments.

5. EXPERIMENTS

The first experiment was to check on user impressions of the MultiBrowser system. Then, since the purpose of the color bars is to support document similarity judgments, we investigated the ability of subjects to use color bars to make such judgments.

5.1 User Impressions

Checking whether users could comprehend and use the MultiBrowser interface architecture could be addressed with a small group of six volunteers. None had prior experience with MultiBrowser. Half browsed a MultiBrowser repository on powered parachuting and half one on vegetarian cooking. Repositories were configured to display with six windows. Each subject spent several minutes browsing at leisure following a short training session, then wrote free form answers, later content analyzed, to questions on their impressions.

Results. Subject responses were content analyzed, a standard social sciences method (Evans 2003 [5]), by examining them for key words and phrases relating to three questions (Tables 2-3). Table 2 shows that subjects tended to consider the density of information displayed by MultiBrowser to be comparatively high. Table 3 shows that all subjects considered MultiBrowser's "FIND SIMILAR" links to be an efficient way to find relevant information though we present no objective evidence it actually is).

Density of information	# Ss
Too high	2
Above normal	2
Acceptable	2
Below normal or too low	0

Table 2. Analysis of responses to the request, "Comment on the density of relevant information appearing, on average, on the screen when MultiBrowser is used vs. when ordinary display is used."

Efficiency	# Ss
High, or above normal	4
Efficient	2
Low, or below normal	0

Table 3. Analysis of responses to the request, "Comment on the efficiency of finding relevant information using "FIND SIMILAR" links."

5.2 Experiments on Color Bars

In an information foraging environment, automatic assessments of text similarity can help when an interesting text inspires the desire to read other, similar texts. The paragraph hyperlinks in MultiBrowser support navigating among similar paragraphs. MultiBrowser also supports navigating among similar *documents* by including a color bar for each window (Figure 1). Clicking the color bar of a window brings up the document in a large, single-window display. The user can also condition a decision about what paragraph hyperlink to click by taking the color bar of the containing document into account. Because the purpose of the color bars is to facilitate judgments of similarity among documents, the speed and consistency of such judgments were investigated.

Experiment A: response speed. This experiment compared the color bar metadata with a typical example of text metadata, the short title+passage descriptions provided by AltaVista. In this experiment two document sets, of ten each, were identified. One set was from the powered parachuting repository and the other was from the vegetarian cooking repository. Eighteen subjects were tested, nine on each set. For testing purposes all document content was removed, leaving only the color bars. Then for each color bar *b* (representing a document) in a set, subjects attempted to identify which two of the other color bars in the set were most similar to *b*.

The AltaVista-provided title+passage metadata were used as a control condition under the assumption that they were representative of non-query-dependent textual metadata representations of documents. While viewing the metadata subjects were asked, as in the color bar condition, which two others in the set seemed most similar to *b* for each metadata item *b*.

Experiment A results. Every subject was faster on the color comparison task than on the title+passage task ($p<0.00001$, two-tailed sign test). On average subjects completed the color comparison task 2.9x faster.

Experiment B: response consistency. Consistency across subjects makes sense as an important aspect of the quality of similarity judgments. Thus this experiment assessed the consistency across subjects of their similarity judgments about color bars. A tendency for subjects to agree that particular documents are similar based on presented metadata suggests that the metadata (either color or the title+passage control condition) supports similarity judgments that have objective reality. On the other hand, lower agreement across subjects suggests less ability of the metadata to support objective similarity judgments, forcing subjects to either make subjective judgments or at best to rely on subjectively chosen criteria.

As in the response speed experiment, each comparison problem involved choosing the two color bars most similar to the color bar of a document *b*, or (the control condition) picking the two most similar AltaVista-provided title+passage excerpts to that of *b*. Each of 9 subjects performed both the experimental and control tasks for each comparison problem they solved. None of the subjects had used MultiBrowser before.

Experiment B results. In 30 of the 32 comparison problems subjects had a higher rate of agreement with each other about which two color bars were most similar to a given color bar than they did about which two title+passages were most similar to a given title+passage. Thus color bars support judgments of document similarity that are more consistent across subjects than those supported by title+passage excerpts ($p<0.0001$, two-tailed sign test). There are however a few subtleties in these results. These are discussed next.

5.3 Discussion: Color Bars

The color bars technique is not limited to MultiBrowser. It could also be used, for example, as a clue in search engine return lists and other situations where it is desirable to show similarities among members of a set visually. Although color bars are different from the standard title+passage data, both are alternative ways of representing documents with metadata. Therefore a comparison of their abilities to represent documents is desirable. Despite their large differences, it is possible to do such a comparison for well-defined measures. The two measures we tested were (1) the speed, and (2) the consistency across users, of similarity judgments. Color bars were found to support similarity judgments among documents that were both faster and more consistent across subjects than the traditional title+passage excerpts provided by AltaVista.

The issue of consistency across subjects of similarity judgments needs further comment. The fact that subjects tended to agree with each other more on similarity judgments of color bars than of title+passages does not prove that their similarity judgments were more correct for color bars. In principle, they could agree while being incorrect. To resolve this question would require a gold standard for correct similarity judgments. The cosine measure on document 5-gram vectors, used for measuring text similarity in the work reported here, is often considered reasonable but other reasonable measures could be proposed that might give significantly different results. Indeed, assessments of document similarity that account for the range of cognitive processes that could legitimately underlie a judgment of similarity probably need to provide answers that are no more specific than a range of possibilities. Thus, it is unclear how to obtain a gold standard that would be able to legitimately classify any given subject's similarity judgment as incorrect.

We also did not address the questions of whether some other method of generating text metadata would lead to different results, or of whether color blindness, present in different varieties and in different degrees in a significant minority of the population, was present in any subjects or affected the results. Nevertheless, the results do suggest color bars as a design space alternative to the typical title+passage metadata for the purpose of supporting fast, consistent document similarity judgments by users.

Although color bars out-performed text metadata in supporting speed and consistency in similarity judgments, the fact that the color assignment method requires the document set to be divided into three clusters, one for each of the three primary colors, deserves further comment. In particular, what happens if using the most natural clustering, however that might be defined, would result in other than three clusters?

Except in mathematically unusual cases, if the most natural grouping is one cluster, then some reasonable way to partition that cluster into two or more clusters can be chosen. If there are two clusters, then one of them can similarly be segmented into two or more new clusters, making a total of three or more. If there are four or more, then some pair of clusters is at least as reasonable to merge into one as any of the other pairs, enabling reducing the total number of clusters. Repeating the process outlined will eventually yield three clusters as desired.

Other bar-based document visualizations have also been described. TileBars (Hearst 1996 [8]) is a well-known example. The speckled scroll bar visualization of Byrd (1999 [2]) is another example. These visualizations contain internal structure representing facts about the internal structures of documents they represent. In our lab we have shown the feasibility of combining these concepts in a browsing system (Figure 2). The result is a TileBar in shades of gray, augmented with colored speckles – a speckled TileBar. To combine that with the color bar concept as well is clearly possible: the gray levels in a speckled TileBar would need to be replaced with brightness levels of the color assignment algorithm of Section 3.

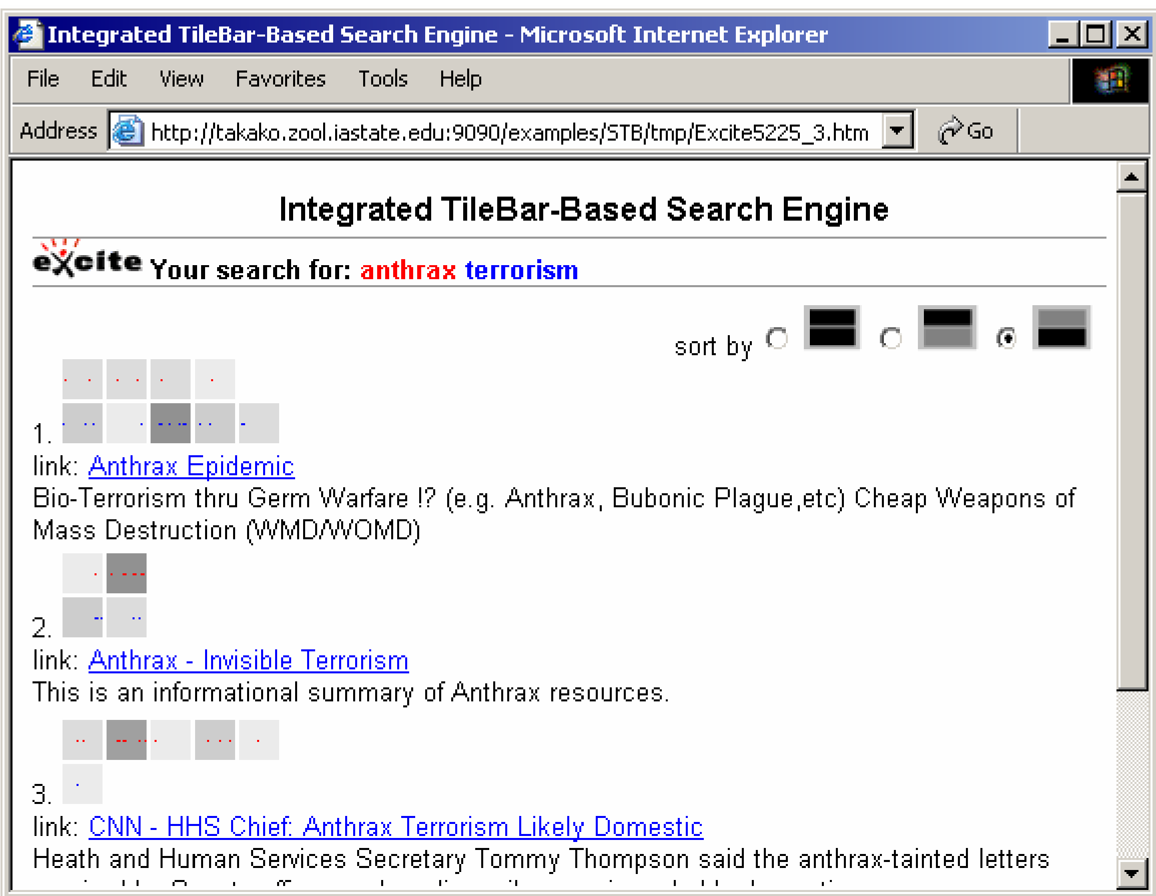

Figure 2. Hybrid of TileBars and a simplified version of Byrd's speckled scroll bar visualization.

6. FUTURE WORK AND CONCLUSION

There are at least two goals for further work in addition to the research on visualizations noted above. One is real-time operation, important because it can be inconvenient to have to wait for the system to process a set of documents into a repository. An algorithm for doing this would process documents into the repository one at a time, and allow the user to browse the repository at the same time as it is gradually increasing in size. The same idea could also apply to adding a newly available document to a mature repository. Another need is to allow updating of a repository by replacing a document in it with a new version of the same document. One way to approach this would be to decompose it into a document deletion operation preceding the document addition just described.

MultiBrowser is an information foraging system as well as a first step toward a much more futuristic vision: a system that supports written conversational dialogues between a user and a repository in accordance with the "panel-of-experts" metaphor.

ACKNOWLEDGMENTS

We thank Matthew Chalmer for comments made on an earlier version of the manuscript.

REFERENCES

[1] Bly, S. and J. Rosenberg. A Comparison of Tiled and Overlapping Windows. *Proc. CHI '86*, 101-106.

[2] Byrd, D. A Scrollbar-Based Visualization for Document Navigation. *Fourth ACM Conference on Digital Libraries* (*DL '99*), 122-129.

[3] Catledge, L.D. and J.E. Pitkow. Characterizing Browsing Strategies in the World-Wide Web. *Computer Networks and ISDN Systems* (1995) **27** (6): 1065-1073.

[4] Damashek, M. Gauging Similarity With N-Grams: Language-Independent Categorization of Text. *Science* (10 Feb. 1995) **267**: 843-848.

[5] Evans, W. Content Analysis Resources. http://www.car.ua.edu/ (as of 11-17-03).

[6] Golovchinsky, G. and M.H. Chignell. The Newspaper as an Information Exploration Metaphor. *Information Processing & Management* (1997) **33** (5): 663-683.

[7] Halasz, F.G. Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems. *Communications of the ACM* (July 1988) **31** (7): 836-852. Revised from *Hypertext '87*, 345-365.

[8] Hearst, M. TileBars: Visualization of Term Distribution Information in Full Text Information Access. *Proc. CHI '95*, 59-66.

[9] Hearst, M. User Interfaces and Visualization. Chapter 10 in R. Baeza-Yates and B. Riveiro-Neto, eds., *Modern Information Retrieval*, ACM Press, 1999, pp. 257-339.

[10] E Ink Corp. http://www.eink.com.

[11] Kaltenbach, M. and C. Frasson. Dynaboard: User Animated Display of Deductive Proofs in Mathematics. *International Journal of Man-Machine Studies* (1989) **30**: 149-170.

[12] Kaltenbach, M., F. Robillard, and C. Frasson. Screen Management in Hypertext Systems with Rubber Sheet Layouts. *Proceedings Hypertext '91*, 91-105.

[13] Kamba, T., K. Bharat, and M.C. Albers. The Newspaper on the Web. *Proc. WWW4* (Boston MA, November 1995). http://www.w3.org/Conferences/WWW4/Papers/93/.

[14] Kandogan, E. and B. Shneiderman. Elastic Windows: Evaluation of Multi-Window Operations. *Proc. CHI '97*, 250-257.

[15] Kandogan, E. and B. Shneiderman. Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser. *Symposium on User Interface Software and Technology* (*UIST '97*), ACM, Oct. 1997.

[16] Lifshitz, K. and B. Shneiderman. Window Control Strategies for On-Line Text Traversal. http://www.cs.umd.edu/hcil/mem bers/bshneiderman/umlpapers/articles.html.

[17] Mayfield, J. and P. McNamee. Indexing Using Both N-Grams and Words. NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7), 1998, 419-424. http://trec.nist.gov/pubs/trec7/t7_proceedings.html.

[18] Miao, J. and D. Berleant. From Paragraph Networks to Document Networks. Submitted. http://class.ee.iastate.edu/be rleant/home/.

[19] Miller, N.E., P.C. Wong, M. Brewster, and H. Foote, TOPIC ISLANDS – a Wavelet-Based Text Visualization System. *IEEE Visualization '98*.

[20] Nielsen, J. User Interface Directions for the Web. *Communications of the ACM* (Jan. 1999) **41**: 65-72.

[21] Pirolli, P. Exploring Browser Design Trade-Offs Using a Dynamical Model of Optimal Information Foraging. *Proc. CHI '98*.

[22] Pirolli, P. and S. Card. Information Foraging in Information Access Environments. *Proc. CHI '95,* 51-58.

[23] Shneiderman, B. User Interface Design for the Hyperties Electronic Encyclopedia. *Proc. Hypertext '87*, 189-194.

[24] Smith, T.R. The Meta-Information Environment of Digital Libraries. *D-Lib Magazine*, Corp. for National Research Initiatives, July/August 1996. www.dlib.org/back.html.

[25] Stotts, D. and R. Furuta. Petri-Net-Based Hypertext: Document Structure with Browsing Semantics. *Transactions on Information Systems* (Jan. 1989) **7** (1):3-29.

[26] Tauscher, L. and S. Greenberg. Revisitation Patterns in World Wide Web Navigation. *Proc. CHI '97*, 399-406. Expanded version: http://ijhcs.open.ac.uk/tauscher/tauscher-nf.html.

[27] United States Display Consortium. *Display Trends Newsletter*. http://www.usdc.org/newsroom/newsletters_do wnloads/displaytrends2.PDF and http://www.usdc.org/newsr oom/newsletters_downloads/ppdisplaytrends.PDF.

[28] Wexelblat, A. and P. Maes. Footprints: History-Rich Tools for Information Foraging. *Proc. CHI '99*, 270-277.

[29] Zellweger, P., B.-W. Chang, and J. Mackinlay. Fluid Links for Informed and Incremental Link Transitions, *Proc. Hypertext '98*, Pittsburgh, June 20-24, 50-57.