Group Asynchronous Browsing on the World Wide Web

Kent Wittenburg, Duco Das, Will Hill, Larry Stead

Abstract

The goal of our Group Asynchronous Browsing (GAB) research is to provide tools for people to leverage the information hunting and gathering activities of other people or groups of people on the World Wide Web. To date we have focused on taking advantage of the personal subject indices that are being constructed today with bookmarks or hotlists of widely available browsers. We have also concentrated on monitoring URLs that may themselves serve as living resources on particular subject areas. In support of the former, we have created a server that collects and merges bookmark/hotlist files of participating users. The server can serve subsets of these merged bookmark files to either standard HTML client browsers or to a client built with the multiscale visualization tool Pad++. For the latter, we have built a tool called WebWatch that can monitor URLs of interest and alert users when significant updates appear.

Keywords: *Information retrieval, resource discovery, tools and browsers, information visualization, community-based navigation*

Introduction

Multitudes of intelligent agents are creating browsable information structures on today's World Wide Web. These intelligent agents are not semi-autonomous computer processes, but rather, people. One of the most effective ways to find information is to find the right person or people who are likely to know about what we are looking for and ask them. In fact, word of mouth and email may currently be the most effective resource discovery tools on today's World Wide Web. The goal of Bellcore's research in Group Asynchronous Browsing (GAB) is to provide tools for better utilization of people's knowledge of the World Wide Web by leveraging the efforts that individuals and groups are already putting forth as they browse. Today's World Wide Web browsers all have some sort of bookmark or hotlist facility with which users can take note of resources that are relevant to their interests and to which they expect to return. Some, such as Netscape's bookmark facility, offer the ability to structure relevant URLs into subject hierarchies. Furthermore, many individuals, groups, and institutions are making significant efforts in providing hand-built subject-oriented guides to the World Wide Web in the form of HTML files directly. For examples, see the Clearinghouse for Subject-Oriented Internet Resource Guides [2] or Yahoo [18]. These subject-oriented guides range from general purpose, for instance, Yahoo [18], to quite specific, for instance, a resource on visual languages and visual programming [16]. Among the issues we see with the utilization of such subject guides are (1) how to find them (the URLs themselves as well as the relevant subject heading(s) within the more general purpose guides), (2) how to combine information from different guides for effective and efficient browsing, and (3) how to keep abreast of changes to them in order to discover additional resources that the keepers of these guides might add. We tend to see the more formal general purpose guides and the more informal personal bookmark files as part and parcel of the same sort of activity.

Our main effort so far has revolved around the creation of a server that collects and merges personal bookmark files of participating users. Besides personal bookmark files, we have included one general purpose subject guide in our initial experiments as well, namely, Yahoo [18], whose role we will subsequently explain. Such a database combined with a World Wide Web server, which we call a Group Asynchronous Browsing (GAB) server, can then provide access to a merged subject tree structure in various ways. This collection of tools is intended to address the issue of how to utilize the browsing activities of others to discover resources, some of which themselves may be guides to further World Wide Web resources. Secondly, we have created a tool that can monitor resources of interest and alert users when significant updates appear. This tool, which we call WebWatch, serves the purpose of resource discovery by drawing users' attention to changes to a known document which itself may be a subject guide to other World Wide Web resources. These tools go part way towards meeting requirements for providing asynchronous, community-based browsing on the World Wide Web.

In the remainder of this paper we will first discuss a vision and the current status of our GAB server. Next, we will describe a client application we have created to visualize and browse over merged subject trees in the GAB database that is built on Pad++ [1, 13], a substrate for creating zoomable human computer interfaces. Then we will briefly discuss WebWatch and conclude with comments on limitations and future research.

The GAB Server

The obvious move for providing access to personal or general subject-oriented indices is to manually or automatically collect them into a database and then provide query or browse capabilities over this database. Yahoo, for instance, includes many entries that themselves are subject-oriented indices and one may search this index of indices. A few research projects have also investigated collecting personal hotlists [8] or history lists [10] into centralized servers. Following Furnas and Zacks [4], we have noted the potential benefits to resource discovery of a data structure called a multitree. For personal bookmark files, which we have so far focused on, a multitree structure can be created in which resources that appear across different subject-oriented bookmark files are shared in an internal graph structure. Browsers over this merged subject tree structure can then provide "backlinks" from shared resources to any and all subject headings which include them. The essential point to note with respect to information discovery is that, starting from some particular resource, new resources that have a good chance of being similar to it may be discovered by navigating "up" to any of the subject headings that include this starting resource and then navigating "down" from those subject headings to other, potentially unknown, resources. Assuming that participating users will construct bookmark files of general utility, we hypothesize that multitree browsing over merged bookmark files might be a cost-effective method for resource discovery since it so thoroughly leverages human judgment, the best arbiter of information quality and relevance.

Figure 1 is an abstract example of such a merged subject multitree, where each subject tree is indicated with a different color and the categorized URLs are indicated as capital letters at the trees' leaves. Subject headings would be attached to each of the nonterminal nodes of the trees. This example only shares tree structure at the terminal nodes of the trees, which is consistent with the sharing of links that occur across personal bookmark or hotlist files, but it should be noted that multitrees can share nonterminal nodes as well.

One of our goals then is to explore World Wide Web services that might be based on such merged subject trees. Assuming that a server is involved, there are several issues that must be addressed. First, there is the data gathering problem, i.e., the server must acquire access to useful subject trees. To date, we have assumed that, with one exception, these are in the form of individual bookmark files that can be created with widely available browsers. For each service that a server then might offer, there is the issue of how to define a client request as well how to define a

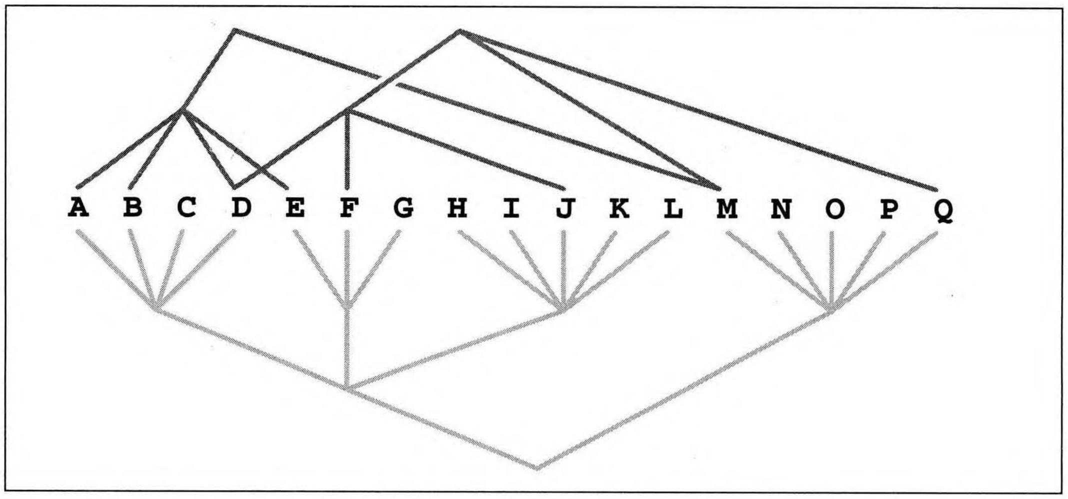


Figure 1: A multitree composed of three different subject hierarchies

server response. We have considered both browsing and querying services that make use of structural relationships in the merged subject trees. Various user interfaces are of course possible, some of which are afforded by existing World Wide Web clients and some of which would require customized browsers.

To date our implementation has focused on the server response side for standard World Wide Web browsers as well as for a customized client. The data gathering problem we have finessed so far by experimenting only with World Wide Web users within our own organization whose personal bookmark files are accessible within our local file structure. However, note that this local scenario may well be useful for intra-organizational information sharing that is deliberately kept proprietary. For wider public access, a forms interface such as can be seen in Webhound [17] could be used for users to submit their bookmark files to a GAB server. We started with bookmark files associated with the widely available Netscape Navigator. Our choice of Netscape is not to be taken as an endorsement of this product, however. Any tool for hierarchical organization of World Wide Web resources would do. Specialized clients that incorporate more sophisticated personal indexing facilities such as Paint [12] or Simon [8] would make our proposal easier to realize in one sense; however, choosing a commercial client that is in widespread use brings with it obvious advantages.

We have so far explored three services:

- In response to a request that specifies a subset of trees from the multitree database, generate an HTML document, usable by any standard World Wide Web client browser, that merges the specified trees and includes internal cross-referencing links for browsing.
- In response to a request as in (1), generate a script, useable by a Pad++ client that we have designed, that affords browsing over a zoomable treemaps visualization of the merged tree set.
- In reponse to a query-by-example in the form of one or more URLs, provide HTML text (that can be appended to any standard HTML document or offered as a dynamically generated page) that will provide links to resources in the GAB database that are related in a structural way to the one or more URLs submitted.

The first two services involve a "top-down" request on the client side that specifies which subject trees to include by simply choosing them from a list. Once this data is received on the client side, it may be browsed there. We envision that this sort of request, refined to include searches on other features of the participating user community and augmented to include the choice of specifying individual subject headings, might be useful for browsing over a relatively unconnected GAB subject tree. The hypothesis is that if one can choose a promising set of subject trees, perhaps by identifying a suitable group of people, it may be fruitful to simply browse over the merger of the trees looking for new resources. A possible indicator of quality of URLs in this scenario is the number of occurrences of a particular resource across the collected group. In other words, if a resource is indexed by many people, it may indicate that the resource is likely to be a better one compared to ones that are not so widely included.

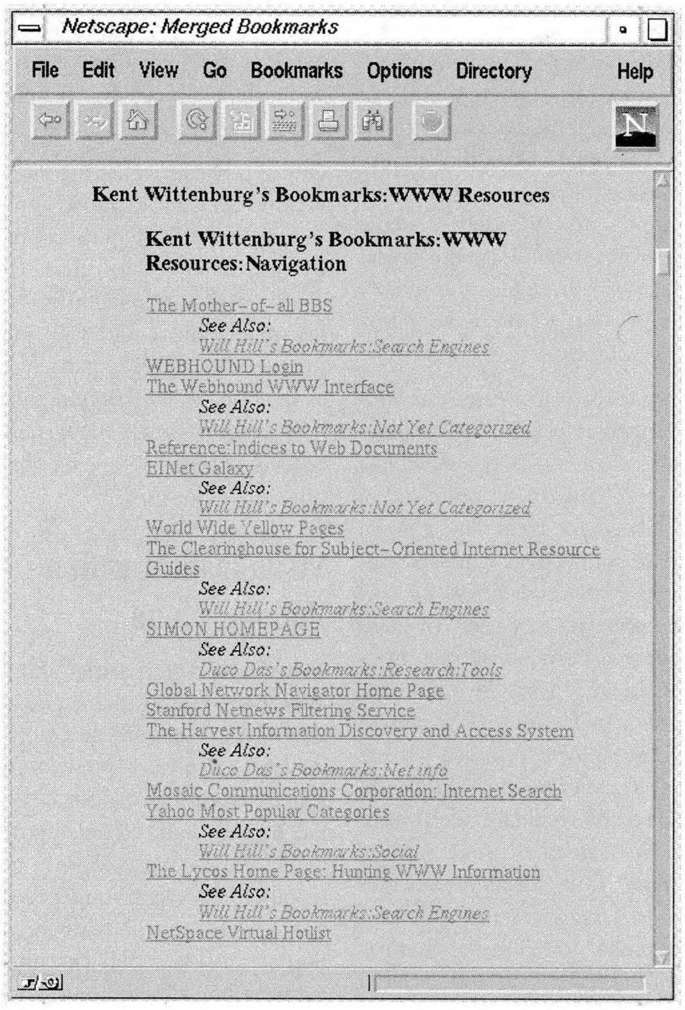
Figure 2 shows a screen dump of an HTML document, being viewed through a commercial World Wide Web browser, that is the result of a query to the GAB server to include a set of participating users' bookmark files (along with Yahoo). The subject headings are indicated as paths from the root of the bookmark tree files. URLs that appear in other GAB subject trees are evidenced as "See Also" crosslinks that appear indented below the URLs in question. The user can then browse to potentially related URLs by clicking on a cross reference link that will jump to that portion of the document that includes other subject listings that also included that same URL. So, for example, note that the URL titled "The Mother-of-all BBS", categorized under Kent Wittenburg's bookmarks with the subject path "WWW Resources: Navigation" also appears under Will Hill's bookmarks under the category "Search Engines." A click on the latter link will jump to that category of Will Hill's, where related resources may then appear.

One of the weaknesses of this browsing method is that users can easily become disoriented about where they are in the information space—an intra-document instance of the well-known "lost in hyperspace" problem. Our informal observation is that this phenomenon invariably occurs when one makes an abrupt jump to another location in the document that is off screen initially. While there are various ways to address this navigation issue in the context of dynamic generation of sets of HTML documents, the Pad++ multitree browser, discussed in the next section, is motivated in part as a visually-oriented solution to this problem.

An alternative to the top-down query method for a GAB server is for the server itself to do the navigating and return just those resources which are presumed to be the best for the client's purposes. Here the client's request comes in the form of one or more URLs, and the GAB server responds with the "close relatives" of those URLs. This is a query-by-example scenario where the measure of relevance is defined structurally. Our starting point for the relevant relation over nodes in a multitree is the sibling relation, defined as follows:

> With respect to some node *X* and some multitree *MT*, the siblings of *X* in *MT* = { *Z* | *Z* is an immediate child of some immediate parent node of *X* in *MT*, and *Z* does not equal *X*}.

The siblings of some information resource are those resources that have been been placed in at least one common subject category as the original. A procedure for collecting siblings would make the navigation moves mentioned earlier: from a starting URL, move up the subject multitree to all subject headings that contain this URL; then move down and collect any new URLs that are contained within these subject headings. For example, in Figure 1, the siblings of A are {B, C, D, E}. The siblings of D are {A, B, C, E, F, J}. Augmentations to this basic procedure might, for instance, filter out any siblings already known to the client if the user's history list is accessible.



Kent Wittenburg's Bookmarks:WWW Resources

Kent Wittenburg's Bookmarks:WWW Resources:Navigation

The Mother-of-all BBS
See Also:
Will Hill's Bookmarks:Search Engines
WEBHOUND Login
The Webhound WWW Interface
See Also:
Will Hill's Bookmarks:Not Yet Categorized
Reference:Indices to Web Documents
EINet Galaxy
See Also:
Will Hill's Bookmarks:Not Yet Categorized
World Wide Yellow Pages
The Clearinghouse for Subject-Oriented Internet Resource Guides
See Also:
Will Hill's Bookmarks:Search Engines
SIMON HOMEPAGE
See Also:
Duco Das's Bookmarks:Research:Tools
Global Network Navigator Home Page
Stanford Netnews Filtering Service
The Harvest Information Discovery and Access System
See Also:
Duco Das's Bookmarks:Net info
Mosaic Communications Corporation: Internet Search
Yahoo Most Popular Categories
See Also:
Will Hill's Bookmarks:Social
The Lycos Home Page: Hunting WWW Information
See Also:
Will Hill's Bookmarks:Search Engines
NetSpace Virtual Hotlist

Figure 2: An HTML document generated by the GAB server representing a subject tree merger

A slightly more distant relation over nodes in a multitree is the cousin relation. It is essentially the transitive closure of the sibling relation, but we find it convenient to distinguish siblings from cousins and so define it as follows:

> With respect to some node *X* and some multitree *MT*, the cousins of *X* in *MT* = { *Z* | *Z* is a sibling of *Y*, *Y* is a sibling of *X*, and *Z* is not a sibling of *X*.} The cousins of an information resource are those resources that have at least one subject category in common with a sibling of the original resource. For example, the cousins of A in the multitree of Figure 1 are {F, G, J}.

The cousins {F, J} are related through A's sibling D. The cousins {F, G} are related through A's sibling E.

An issue we encountered early on in our experiments was the problem of sparse connectivity in a merged subject-tree database. If the original subject trees share no common resources, it is difficult to leverage the collective subject trees with structural relations in the ways we have been describing. We conducted some informal experiments to determine how much sharing of URLs there was across actual user bookmark files in our organization. It turned out that the amount of sharing was surprisingly low. Even within our local computer graphics and interactive media group at Bellcore, a group who all use the World Wide Web regularly and who would presumably have common interests, we found a very low incidence of shared resources.

As a reponse to this problem, we came up with the idea of using the cousin relation in conjunction with one or more general purpose subject indices to provide the glue that connected a group of personal bookmark files. The general idea here is to increase connectivity across a relatively sparsely connected GAB database by taking advantage of the presumably larger coverage of one or more general purpose subject trees (which can be merged into a single multitree). The tree colored green in Figure 1 is meant to suggest the role of the general purpose subject multitree. It will tend to cover more of the World Wide Web than any relatively small collection of personal bookmark files, which are suggested by the red and blue trees. Thus, even though one may find no common linkages across the personal bookmark file collection directly, there is still a means of indirectly finding related resources by going through the green tree. Our initial experiments used Yahoo [18] in the role of general subject multitree.

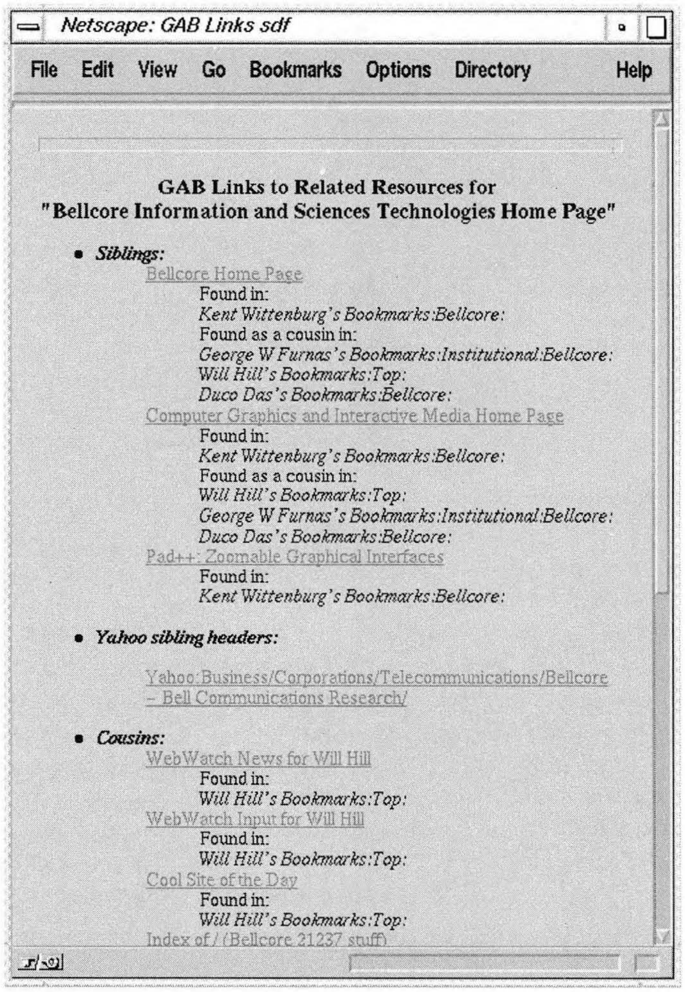
Figure 3 is a screen dump of a page automatically generated by our GAB server in response to a query containing the URL whose title is "Bellcore Information and Sciences Technologies Home Page." There were siblings found in only one user's bookmarks, but there were a greater number of cousins from three of the participating users. The implication is that a subject category of Yahoo contained a link to the resource "Bellcore Information and Sciences Technologies Home Page" and that other resources that appeared under this category were shared in the subject trees of the three other users. We include a link to the mediating Yahoo subject category as well, since users may be interested in jumping there directly for futher browsing.

Our hypothesis is that the more general "cousins" relation will afford more connectivity in a GAB server, and this hypothesis has been borne out by our initial observations. Ultimately, the right choice of which relation to use may have to be fine-tuned and could well differ across specific instances of GAB databases. Combining other evaluations of the usefulness of some potentially new resource with these purely structural relations is also possible, e.g., community-based recommendations [6, 17].

Visualization with a Pad++ Client

As we have noted earlier and has been remarked elsewhere [11, 3], there is a reason to suppose that effective visualizations of complex hypermedia structures may attenuate the "lost in hyperspace" syndrome. In our GAB work we have so far focused on visualizing the GAB database itself, which we suppose may be useful for resource discovery under conditions of sparse connectivity and perhaps for other tasks as well. Furnas and Zacks [4] describe several interfaces for browsing of multitrees, and these may indeed be of interest for GAB. Here we offer a new multitree browser that leverages the infinite zooming and navigational features of Pad++. It uses treemaps [7] as a starting point for the layout.

Pad++ [1, 13] and its forebears [14] provide a substrate for creating interactive graphics on an infinitely scalable surface with smooth zooming and panning. One hypothesis is that hierarchical relations can be naturally captured with scale: information that is deeper in the hierarchy can be represented as being smaller on the Pad++ surface. Treemaps [7], a method for visualization of hierarchical information based on spatial containment, seemed to us to be the most promising lay-



GAB Links to Related Resources for "Bellcore Information and Sciences Technologies Home Page"

- ***Siblings:***
  - Bellcore Home Page
    - Found in:
    - *Kent Wittenburg's Bookmarks:Bellcore:*
    - Found as a cousin in:
    - *George W Furnas's Bookmarks:Institutional:Bellcore:*
    - *Will Hill's Bookmarks:Top:*
    - *Duco Das's Bookmarks:Bellcore:*
  - Computer Graphics and Interactive Media Home Page
    - Found in:
    - *Kent Wittenburg's Bookmarks:Bellcore:*
    - Found as a cousin in:
    - *Will Hill's Bookmarks:Top:*
    - *George W Furnas's Bookmarks:Institutional:Bellcore:*
    - *Duco Das's Bookmarks:Bellcore:*
  - Pad++: Zoomable Graphical Interfaces
    - Found in:
    - *Kent Wittenburg's Bookmarks:Bellcore:*
- ***Yahoo sibling headers:***
  - Yahoo:Business/Corporations/Telecommunications/Bellcore – Bell Communications Research/
- ***Cousins:***
  - WebWatch News for Will Hill
    - Found in:
    - *Will Hill's Bookmarks:Top:*
  - WebWatch Input for Will Hill
    - Found in:
    - *Will Hill's Bookmarks:Top:*
  - Cool Site of the Day
    - Found in:
    - *Will Hill's Bookmarks:Top:*
  - Index of / (Bellcore 21237 stuff)

Figure 3: GAB server response for related resource query

out strategy for zoomable multitree data. Most other alternatives that occurred to us required some form of dynamic changes to object positions during user interactions. Smooth animation for arbitrary object movement or generalized warping was not yet well-supported in Pad++ at the time we did this research. Also, dynamically shifting layout brings with it a host of other issues involving cognitive continuity that we supposed might be handled by general panning and zooming features alone if reasonable static layouts could be found.

Figure 4 shows a screen capture of our Pad++ multitree browser together with a standard HTML client that may be controlled through Pad++. The Pad++ client is automatically invoked as a helper application from the file returned by the GAB server, which it tags with a Pad++ MIME type.

Each user subject tree in the Pad++ visualization is assigned a unique color hue. The coloring algorithm chooses points of equal distance from the continuum of hues proportional to the number of users. Color saturation is bound to hierarchical depth; it is reduced at each lower level of

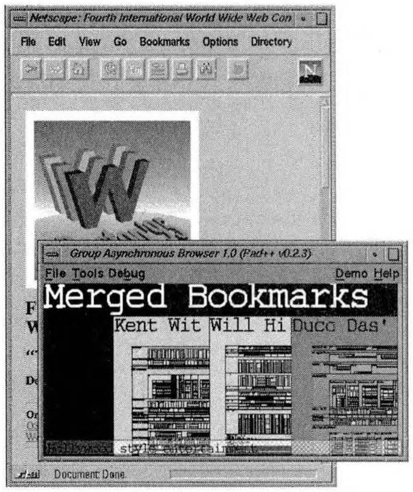

Netscape: Fourth International World Wide Web Con
File Edit View Go Bookmarks Options Directory
Group Asynchronous Browser 1.0 (Pad++ v0.2.3)
File Tools Debug
Demo Help
Merged Bookmarks
Kent Wit
Will Hi
Duco Das'
Document: Done

Figure 4: The Pad++ multitree browser as helper app

the tree. Both intermediate and terminal nodes of the multitree are represented as rectangles. Their names are displayed in the translucent status bar. A double click with the left mouse button will cause the system to smoothly zoom/pan to the rectangle of interest. In addition to system-controlled zooming and panning, the user is also able to control panning and zooming directly. As the user zooms in, nodes higher up in the hierarchy slowly dissolve while revealing more of their descendants.

Figure 5 is suggestive of the system behavior during a zoom in to a subject category of one of the users. The actual application simply narrows the frame as the view is smoothly zoomed towards its destination. If a user wants to look at the contents of a resource itself, she double-clicks on its name and a message to open that resource is conveyed to the standard World Wide Web browser.

Multitrees are of course not just trees, which treemaps are designed to support, and so a solution for adapting treemaps to the more general graph structure of multitrees was required for this visualization. For purposes of layout, we created a tree out of our original multitree by (1) adding a root node whose immediate children were the root nodes of each of the subject trees in the GAB database and (2) for each shared node, adding an additional level in the tree consisting of cross references to the subject headings containing those shared nodes. This is a graphical rendition of a structure that is also evident in the HTML nested lists of Figure 2. The cross references are then color-coded to the subject trees which contain them. Besides the participating users subject trees, we also include cross-references into Yahoo, which are colored red.

Consider the visualization encoding exemplified in Figure 5. "Navigation" is a subject heading. It is assigned a rectangle colored by its host subject tree. This green rectangle contains several other green rectangles that represent the URLs categorized as navigation resources. Those URLs that appear elsewhere in the multitree structure under different subject categories enclose additional rectangles of different colors, one for each cross-referenced subject heading.

As for navigating, clicking on a cross-reference rectangle invokes a system-generated pan/zoom that smoothly zooms out until both the point of origin (i.e., the cross reference) as well as the destination are in the view and then zooms in on the destination, after which the shared node

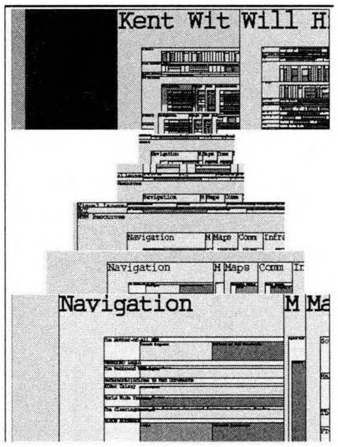


Figure 5: Zooming in to see more detail in a subject tree

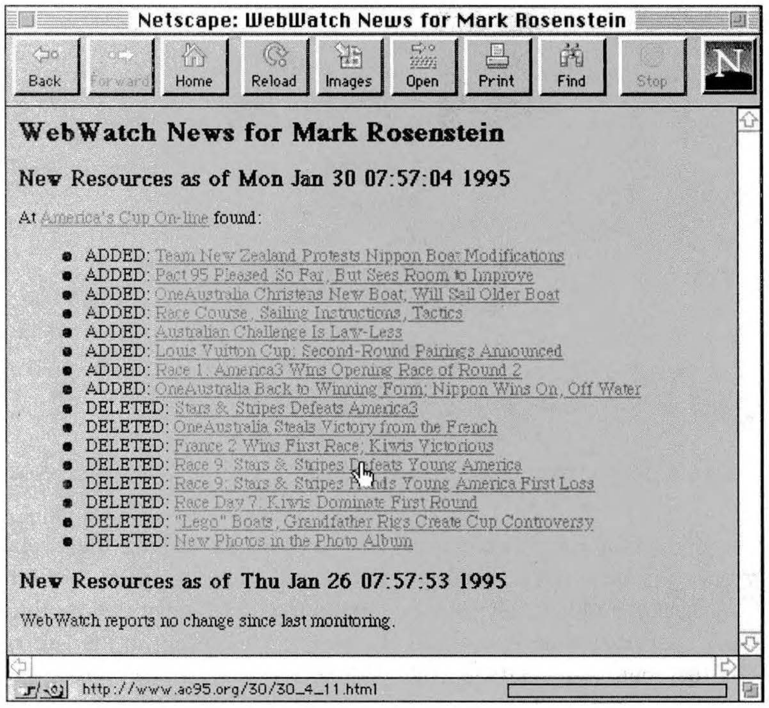
blinks. We bel'eve such automatically computed pan/zoom trajectories [5] are a very significant feature of Pad++ navigation and uniquely characterize this browsing and visualization approach. The hypothesis is that one can overcome the "lost in hyperspace" feeling garnered from abrupt transitions in hyperspace by always getting an automatically generated bird's eye view as a transition when making large jumps from one part of the space to another. Note that this hypothesis is only sustainable if the user is in fact able to acquire some familiarity with the global layout—thus the importance of maintaining continuity through relatively static spatial positioning.

Our Pad++ client has other features as well. With a slider the user is able to rescale the treemap, which has the following sigificance. A value of 100% yields the classic no-offset treemap (where only terminal nodes are afforded real estate), whereas a value of, for example, 40% yields a layout where adjacent tree levels differ in size by orders of magnitude. Imagine being able to stretch and compress a tree arbitrarily as one is looking down from the top. As the tree stretches, the levels that are lower in the hierarchy get smaller and further away. This leaves more screen real estate that might be used to display information about the local levels one can see clearly. It also diminishes the load on the perceptual system by making the overall display less cluttered. There is a need for further research in exploring these layout options through "semantic zooming," a concept in the Pad community that signifies that the system might choose to display information differently depending on how much screen real estate is afforded a particular object.

Monitoring Through WebWatch

Requirements for group asynchronous browsing include not only the need to find and browse subject-oriented indices in a community of interest but also the need to stay abreast of changes to those indices. The task of keeping informed of new, relevant, and significant developments in their areas of responsibilities is becoming an important part of the job responsibilities of today's knowledge workers.

We developed WebWatch as a portable service to help users keep abreast of new developments in their favorite World Wide Web pages. WebWatch makes it easy to monitor any Web page in HTML format. On a daily basis, WebWatch alerts its users to new Web resources as they appear at monitored locations. WebWatch outputs an HTML formatted page which reports added and/or deleted resources at any monitored location in the Web. For example, Figure 6 shows the result



WebWatch News for Mark Rosenstein

New Resources as of Mon Jan 30 07:57:04 1995

At America's Cup On-line found:

- ADDED: Team New Zealand Protests Nippon Boat Modifications
- ADDED: Pact 95 Pleased So Far, But Sees Room to Improve
- ADDED: OneAustralia Christens New Boat, Will Sail Older Boat
- ADDED: Race Course, Sailing Instructions, Tactics
- ADDED: Australian Challenge Is Law-Less
- ADDED: Louis Vuitton Cup: Second-Round Pairings Announced
- ADDED: Race 1: America3 Wins Opening Race of Round 2
- ADDED: OneAustralia Back to Winning Form; Nippon Wins On, Off Water
- DELETED: Stars & Stripes Defeats America3
- DELETED: OneAustralia Steals Victory from the French
- DELETED: France 2 Wins First Race; Kiwis Victorious
- DELETED: Race 9: Stars & Stripes Defeats Young America
- DELETED: Race 9: Stars & Stripes Hands Young America First Loss
- DELETED: Race Day 7: Kiwis Dominate First Round
- DELETED: "Lego" Boats, Grandfather Rigs Create Cup Controversy
- DELETED: New Photos in the Photo Album

New Resources as of Thu Jan 26 07:57:53 1995

WebWatch reports no change since last monitoring.

http://www.ac95.org/30/30_4_11.html

Figure 6: An example page generated by WebWatch

of a WebWatch report for the resource "America's Cup On-line."

Reported are anchors that are "new for me here since the last time I checked." This type of service is most useful in a large, rapidly changing, and unpredictable information environment.

How does it work? Users of WebWatch use Xcut&paste to copy and paste in anchors into their WebWatch subscription page. This is an HTML page whose anchors point at resources to be monitored for the user. Entry of an anchor into a WebWatch subscription file activates the monitoring of the selected Web location. WebWatch fetches the last modified date (if available) of the resource and compares that date with its own page records. If the resource has been modifed more recently than WebWatch records, or if WebWatch has no record, WebWatch fetches the document to *diff* against the last copy it fetched.

Nightly, a chron file, which is an Emacs batch-mode program, wakes up WebWatch. WebWatch relies upon Emacs W3 functionality. It fetches and stores new copies of all monitored HTML World Wide Web resources. Old and new versions are *diff*'d to identify additions and deletions. These are analyzed for anchors pointing at resources. New anchors added to monitored documents will contain URLs pointing at new resources. Deleted anchors will contain old URLs that no longer appear as resources in the monitored resource. Having collected these newly added resources and newly deleted resources, WebWatch generates an HTML page which encodes a tree whose leaves are added and deleted resources and whose upward paths are HTML heading levels from monitored resources. Only changes show up in each tree.

When is a new anchor really new? What constitutes a real change is a semantic question. Currently, WebWatch reports new links if the anchor in question has both a new reference location and name. If an anchor has only a new reference location and not a new label, we consider the change merely a house-keeping change. If an anchor has only a new label in the document but points to exactly the same reference location as last time, we consider the change only a cosmetic name change and do not report it as a new resource.

WebWatch appears to work well for the simple task for which it was designed. Monitoring locations for changes has been useful to the small group of current users. One user was able to purchase Van Halen tickets in a timely manner by monitoring the Van Halen concert schedule page and then ordering as soon as a nearby concert was announced. Two other users monitored news from the America's Cup races. It seems evident that there is a demand for some monitoring functionality on the World Wide Web, and we expect future browsers and/or servers to contain resource monitoring features that take into account personal interests, responsibilities, and history of interaction. As is discussed by LaLiberte and Braverman [9], there are still issues involving scalability that must be addressed when "what's new" services become commonplace on the World Wide Web.

Conclusion

We believe that our research on Group Asynchronous Browsing shows promise for attacking the resource discovery problem on the World Wide Web. As far as we know, this research brings a unique three-pronged approach to the problem: a GAB server for merged bookmarks and general subject indices, a Pad++ client for visualizing the GAB database, and a tool for monitoring resources of interest. Others have looked at pooling hotlists or annotations (e.g., [9, 15]), but we believe our approaches to browsing over merged subject trees involving the sibling and cousin relations to be novel. On the visualization side, Mukherjea and Foley [11] have also built a prototype that uses treemaps for visualizing Web structures. Our major contribution here is to add zooming to treemaps that incorporates scaling options as well as navigation through crosslinking and automatic pan/zoom trajectories formulated in [5]. Immediate future work includes improving the client-side interface to the GAB server as well as fielding the GAB server more widely. We also expect to be doing some empirical experiments to test the validity of the visualization work, and there are many extensions that we would like to see including visualizations of web structures more generally.

Acknowledgments

This work has been supported by ARPA grant N66001-94-C-6039 under the ARPA Human Computer Interaction initiative. The Beyond Imitation Project, which has participation from University of New Mexico, New York University, and Bellcore is directed by Jim Hollan of the University of New Mexico. We are grateful for this support. The Pad++ team, Ben Bederson in particular, has been instrumental in this effort through the creation and support of the substrate for our visualization work. Thanks to Mark Rosenstein for his contributions to the GAB server and also to George Furnas for his many insightful criticisms and suggestions as well as for his inspiring research. Finally, thanks to David Filo and Jerry Yang for making a dump of the Yahoo database available for our research purposes. ■

References

1. Benjamin B. Bederson and James D. Hollan, *Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics*, Proceedings of ACM UIST '94 (Marina Del Rey, CA), ACM Press, 1994, pp. 17-26.
2. Clearinghouse for Subject-Oriented Internet Resource Guides, *http://www.lib.umich.edu/chhome.html*
3. Peter Doemel, *WebMap—A Graphical Hypertext Navigation Tool*, Second International conference on the World Wide Web (Chicago, IL), 1994, pp. 785-789, *http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/doemel/www-fall94.html*

4. George W. Furnas and Jeff Zacks, *Multitrees: Enriching and Reusing Hierarchical Structure*, Proceedings of CHI '94 Human Factors in Computing Systems (Boston, MA), ACM Press, 1994, pp. 330-336.

5. George W. Furnas and Benjamin B. Bederson, *Space-Scale Diagrams: Understanding Multiscale Interfaces*, Proceedings of CHI '95 Human Factors in Computing Systems (Denver, CO), ACM Press, 1995, pp. 234-241.

6. Will Hill, Larry Stead, Mark Rosenstein, and George Furnas, *Recommending and Evaluating Choices in a Virtual Community of Use*, Proceedings of CHI '95 Human Factors in Computing Systems (Denver, CO), ACM Press, 1995, pp. 194-201.

7. Brian Johnson and Ben Shneiderman, *Tree-maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures*, IEEE Visualization '91, 1991, pp. 284-291.

8. Mark J. Johnson, SIMON HOMEPAGE, *http://www.elec.qmw.ac.uk/simon/*

9. Daniel LaLiberte and Alan Braverman, *A Protocol for Scalable Group and Public Annotations*, Computer Networks and ISDN Systems 27: 911-919, 1995, *http://www.igd.fhg.de/www/www95/proceedings/papers/100/scalable-annotations.html*

10. Jong-Gyun Lim, *Using Coollists to Index HTML Documents in the Web*, Second International conference on the World Wide Web (Chicago, IL), 1994, pp. 831-838, *http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/lim/coollist.html*

11. Sougata Mukherjea and James D. Foley, *Visualizing the World-Wide Web with the Navigational View Builder*, Computer Networks and ISDN Systems 27:1075-1087, 1995, *http://www.igd.fhg.de/www/www95/proceedings/papers/44/mukh/mukh.html*

12. K.A. Oostendorp, W.F. Punch, and R.W. Wiggings, *A Tool for Individualizing the Web*, Second International conference on the World Wide Web (Chicago, IL), 1994, pp. 49-57, *http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/oostendorp/oostendorp.html*

13. Pad++: Zoomable Graphical Interfaces, home page created by Jonathan Meyer, *http://www.cs.unm.edu/pad++/begin.html*

14. Ken Perlin and David Fox, *Pad—An Alternative Approach to the Computer Interface*, Proceedings of ACM SIGGRAPH 1993 (Anaheim, CA), ACM Press, 1993, pp. 57-64.

15. Martin Roescheisen, Christian Mogensen, and Terry Winograd, *Beyond Browsing: shared comments, SOAPs, trails, and on-line communities*, Computer Networks and ISDN Systems 27: 739-749, 1995, *http://www.igd.fhg.de/www/www95/proceedings/papers/88/TR/WWW95.html*

16. The World Wide Web Virtual Library: Visual Languages and Visual Programming, *http://cuiwww.unige.ch/eao/www/Visual/Visual.Programming.biblio.html*

17. The Webhound WWW Interface, *http://rg.media.mit.edu:80/projects/webhound/www-face/*

18. Yahoo, *http://www.yahoo.com*

About the Authors

Kent Wittenburg
[*http://community.bellcore.com/kentw/home-page.html*]
Bellcore, Room MCC 1A-332R
445 South St.
Morristown, NJ 07962-1910
kentw@bellcore.com

Duco Das
[*http://community.bellcore.com/duco/home-page.html*]
Stevens Institute of Technology & Delft University of Technology
Julianalaan 132
2628 BL Delft, The Netherlands
duco@bellcore.com or
afstc029@IS.TWI.TUDelft.NL

Will Hill
Bellcore,AT&T Bell Labs, Room 2B-402
600 Mountain Ave.
Murray Hill, NJ 07974
willhill@research.att.com

Larry Stead
Bellcore, Room MCC-1A348R
445 South St.
Morristown, NJ 07962-1910
lstead@bellcore.com