Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser

*Eser Kandogan and Ben Shneiderman**
Department of Computer Science,
Human-Computer Interaction Laboratory
*Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742
kandogan@cs.umd.edu, ben@cs.umd.edu

ABSTRACT
The World-Wide Web is becoming an invaluable source for the information needs of many users. However, current browsers are still primitive, in that they do not support many of the navigation needs of users, as indicated by user studies. They do not provide an overview and a sense of location in the information structure being browsed. Also they do not facilitate organization and filtering of information nor aid users in accessing already visited pages without high cognitive demands. In this paper, a new browsing interface is proposed with multiple hierarchical windows and efficient multiple window operations. It provides a flexible environment where users can quickly organize, filter, and restructure the information on the screen as they reformulate their goals. Overviews can give the user a sense of location in the browsing history as well as provide fast access to a hierarchy of pages.

Keywords
World-Wide Web, Window Management, Information Visualization, User Interfaces.

INTRODUCTION
The World-Wide Web (WWW) is becoming an invaluable source for the information needs of many users. By clicking on a link, users can easily access related information. The ability to access more information in such a quick way fascinates most users, however, after a while users typically have difficulty in remembering where they are coming from and in accessing previously visited pages.

The one-window-one-click interface is appealing in its simplicity, but for many tasks and many users a more powerful browser with multiple windows and multiple selection of links may speed task completion.

Studies on users' navigation strategies, and browsing-task analyses provide interesting results [6, 21, 7, 19]. However, current interfaces for browsing on the WWW are still primitive, in that they do not support many of the navigation needs of users, as indicated by these studies. They do not provide an overview and a sense of location in the information structure being browsed, nor facilitate organization and filtering of information. They provide only rudimentary means to aid users in accessing already visited pages. Recent research offers varied solutions including 3-dimensional, zooming, and metaphorical web browser interfaces [1, 23, 3, 5, 4].

In this paper, a new browsing interface is proposed with multiple window operations and hierarchical windows. It provides a flexible organization in which users can easily organize, filter, and restructure the information on the screen as they reformulate their goals. Overviews can be created for visited pages that may give the user a sense of location as well as provide fast access to a hierarchy of pages.

This paper begins with a review of the observations and analyses made on the problems in hypertext browsers. Next, studies examining users' browsing strategies are briefly described with the lessons derived from these studies listed. Then, our browsing interface is presented with its design principles and examined based on its compliance with the results of these user studies. Technical information regarding the implementation of the Elastic Windows browser is also discussed with reference to a more detailed description.

PROBLEM MOTIVATION
Conklin [9] identified the problems with hypertext as:

- **Disorientation:** The tendency to lose one's sense of location and direction in a nonlinear document.
- **Cognitive overhead:** The additional effort and concentration necessary to maintain several tasks or trails at once.

Disorientation, as Conklin argued, stems from the lack of knowledge of the current position in the whole information structure, but also of the path(s) to the desired destination position. Utting and Yankelovich [22] identified these as spatial and temporal contexts, respectively.

Utting and Yankelovich's further examination provided more details on the problems in current browsers:

- Hard to remember which documents are open
- Following every link is tedious
- Amount of information is unpredictable
- Difficult to get back to a point in the history

The cognitive overhead problem is more related to the user's browsing strategy. It is about how users seek a balance between the gains of added knowledge and the losses from increased distraction by following a link. Marchionini and Shneiderman [16] argue that in browsing, goals are not well defined and change dynamically as new information is encountered by the user.

Cruz [10] observed that current web browsers give little flexibility to users both in filtering out unwanted information and in the specification of the spatial and temporal layout. This limits user's browsing and organization capabilities.

The fundamental mechanism for organization is composition. However, Halasz [13] argues that the hypertext model lacks a composition mechanism, i.e. a way of representing and dealing with groups of nodes and links as unique entities separate from their components.

Rosenberg [19] pointed out that current WWW browsers provide a single window on the document, and when users click to follow a link, the new document is opened in place, replacing the former. Some systems allow another window to be opened for the new document. Browsers simply rely on the window manager to organize these open documents. However, current window managers fail to provide an organization which reflects the semantic relationship that exists among documents browsed on the WWW.

LESSONS FROM USER STUDIES

Although several studies provide demographical information on users and web-sites, our interest is on studies that examine users' navigational access patterns. Studies done on users' navigational strategies by Catledge and Pitkow [6] and by Tauscher and Greenberg [21] are particularly interesting in that they are done in open systems for long durations.

Catledge and Pitkow captured client-side user events of the XMosaic browser from a population of 107 users in the Georgia Institute of Technology's College of Computing for a three week period.

They characterized users' navigation strategies according to the average frequency of following a path with a certain depth. Average frequency was found to be linearly dependent on the path length with a slope of -0.24. Users' browsing strategies are classified as *serendipitous browser*, *general purpose browser*, and *searcher* based on users' average slope.

Besides this classification of the navigation strategies, they also observed:

- Tendency to browse in a small area
- Frequent use of backtracking: Back button usage 41%
- Shallow browsing: Rarely more than two layers
- Infrequent save, print, hotlist addition and retrieval

Tauscher and Greenberg analyzed 6 weeks of detailed usage data from 23 users with at least a year experience of browsing on an instrumented XMosaic browser. They analyzed recurrence of page visits, growth of URL vocabulary, visit frequency as a function of distance, frequency of URL accesses according to page types, locality, and length of repeated sequences.

In summary, their observations are:

- High page recurrence rate: 58% of the pages revisited
- Continued growth of URL vocabulary
- High recency of revisits
- Frequent visits only to very few pages: Personal and organization pages, search engines, etc.
- Browse in small clusters of pages
- Short sequences of repeated URL paths

ELASTIC WINDOWS WEB BROWSER

Hierarchical Page Organization

The Elastic Windows browser is a multi-window browser, where pages are organized hierarchically [14, 15]. Hierarchical organization of pages allows the user to see the context, while exploring further details lower in the hierarchy. Although syntactic information structure on the WWW is an arbitrary graph, presenting the information in a hierarchy can help users in their information seeking activities. While hierarchical organization might facilitate navigation, it can also give users a sense of location in the information structure.

Figure 1 shows a user browsing the Human-Computer Interaction Lab (HCIL) web pages. Pages are hierarchically organized with the top level HCIL main page placed on the left. On the right, four pages (*Lab Description*, *Principal members*, *Students*, *Collaborators*, and *Research Project Description* pages) are opened as a group in the *About HCIL* window. Furthermore, five project pages are opened in the *Research Project Description* page. The hierarchy is created as a result of user actions, not prepared in advance.

Opening new pages In the Elastic Windows browser, a new page can be opened by clicking on the link. The window for the new page is opened "inside" the window of the parent page, placed on the right, using half of the space (Figure 2.a). Thus, when a link is followed, the context is preserved on the left, while the detail on a link is being examined on the right. By selecting a different link, the user might either replace the last selected link(s) or add the new link to the existing pages sharing the space.

Multiple links can be opened by the select operation with the left mouse button followed by a right mouse button click. All selected links are opened side by side, placed within their parents window on the right (Figure 2.b). Alternatively, vertical and tiled placement styles can be selected from the window menu. Links in a region can be selected by drawing a rectangle on the links with the left mouse button. Noncontiguous links can be added to the selection by pressing the control key with the left mouse pressed.

The conventional open-and-replace strategy is still available in the Elastic Windows browser by clicking the right mouse button with the Control key pressed. This way users can skip uninteresting intermediary pages when following a number of links to the desired information.

Multi-level context When browsing a large information structure, users might want to keep multiple levels of contexts on the screen at the same time. This might help users in their navigational strategy by greatly reducing the need to back up, and also lessens a possible disorientation.

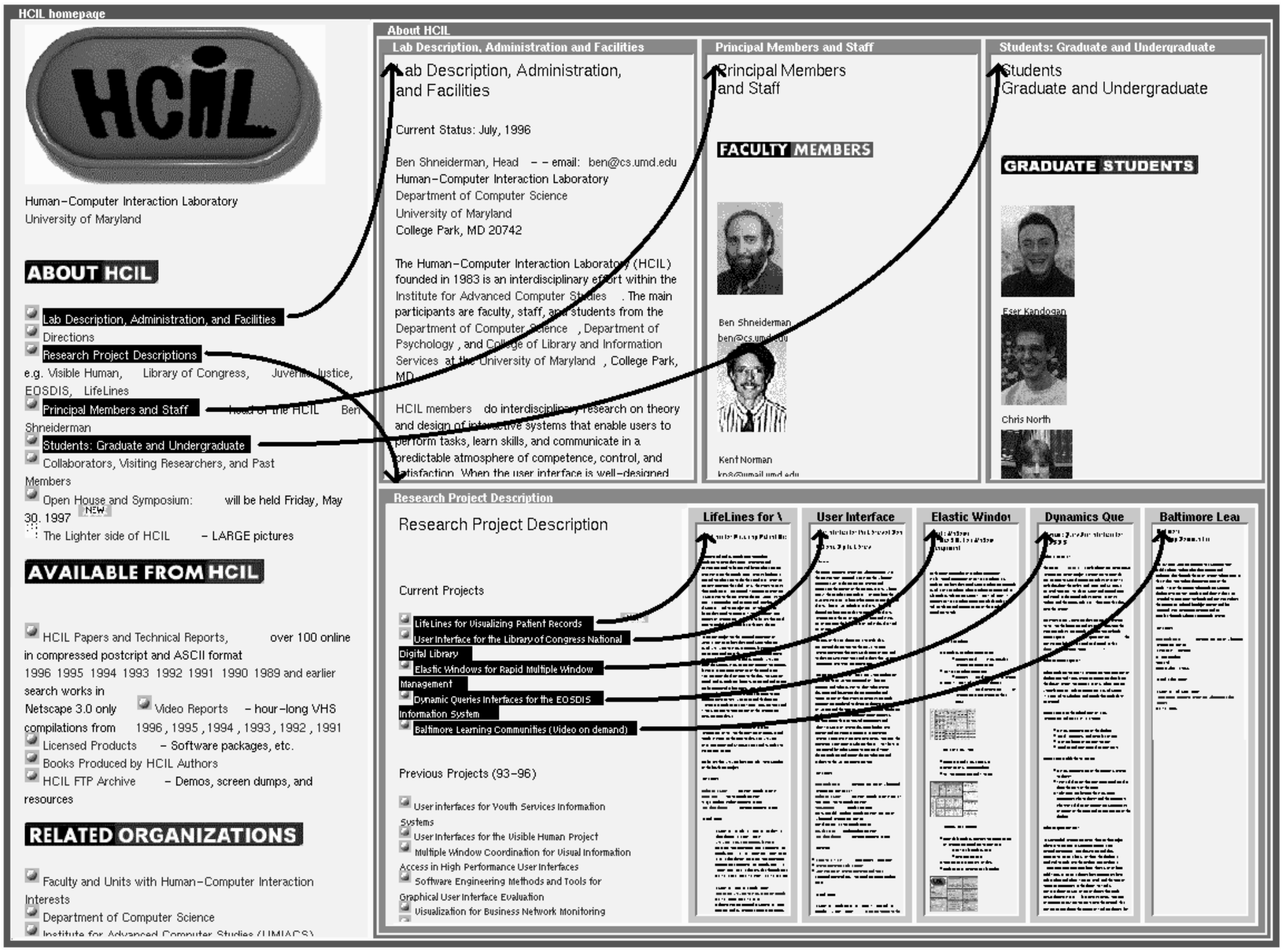

Figure 1: HCIL web pages: Main page on the left of the group window *About HCIL*, which contains *Lab Description*, *Principal members*, *Students*, *Collaborators*, and *Research Project Description* pages. Below, *Research Projects* page contains all five current project descriptions

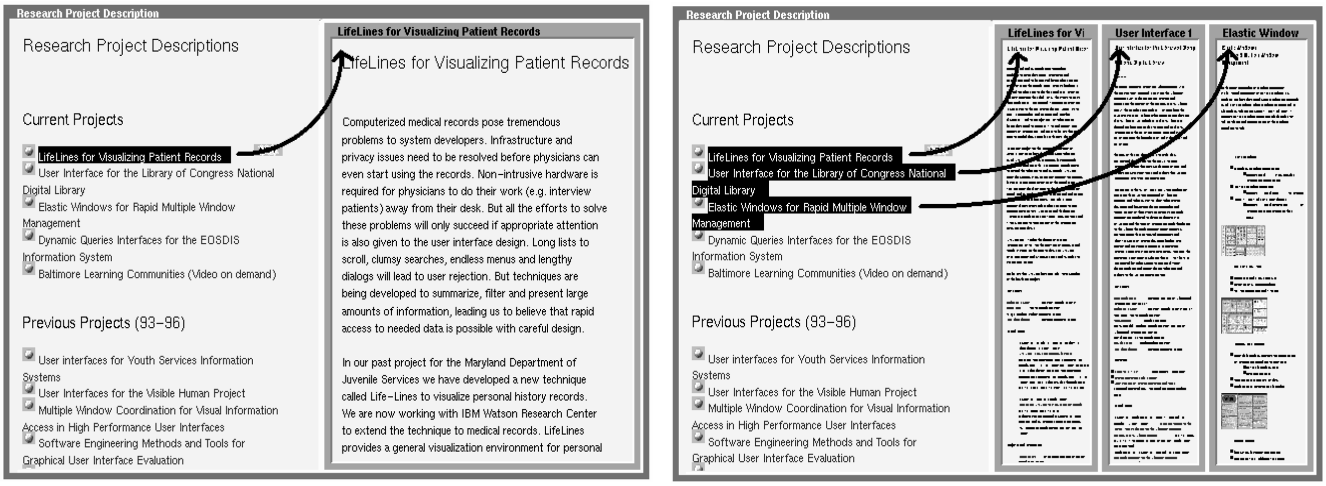

Figure 2: Opening new pages: a) Click to open a single page b) Select and click to open multiple pages

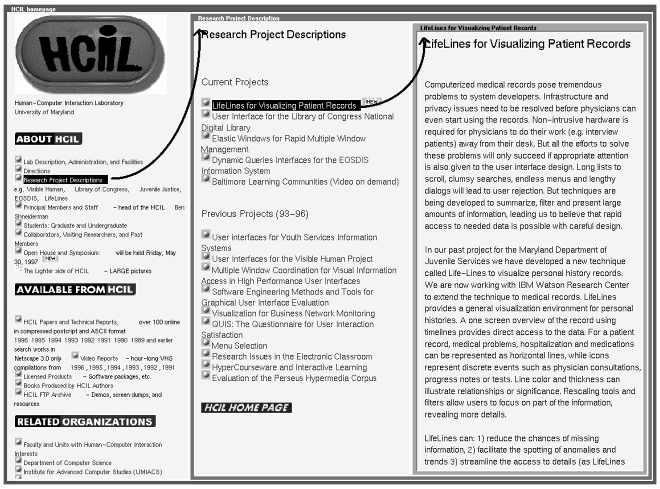

Figure 3: Keeping three levels of context in view: *HCIL Main page*, *Research Project Description*, and *Lifelines* project page

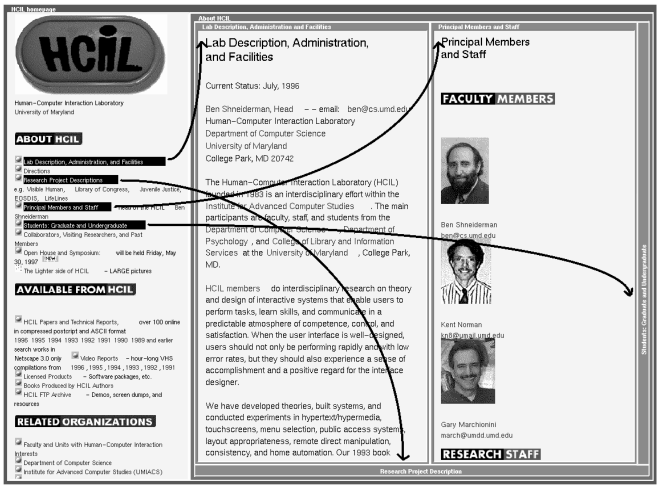

Figure 4: Information hiding: *Research Project Description* and *Students* pages are packed on the layout shown in Figure 1

In Figure 3, the user keeps both the HCIL main page and the *Research Project Description* page in view, while exploring the Lifelines project. Keeping the project description page allows the user to switch to other project pages easily. Keeping the HCIL main page gives users orientation as well as allows them to switch to other interesting links related to HCIL. In the Elastic Windows browser users can also follow more than one trail at the same time. Selecting another link from any page starts a new trail.

Multi-level focus While keeping multiple levels of context helps preserve orientation, at times users may need to focus on a particular set of pages deeper in the hierarchy. In the Elastic Windows browser, users might maximize a subhierarchy of pages at any level to full screen, allowing more detail to be displayed. For example, a user might initially focus on all research projects in the HCIL page, then on a particular research page, and then on the participants' pages of that project. Users can also skip multiple levels and directly focus on a deeper subhierarchy. The maximize operation can be invoked from the window menu. Users can come back to the previous hierarchy by the return operation also selectable from the window menu. The return operation is more powerful than the conventional back operation in that users can skip levels and see breadth as well as depth.

Information Hiding In the Elastic Windows browser, a hierarchy of pages can be packed into a small horizontal or vertical bar, giving more space to other pages. The pack operation can be selected from the window menu. This operation not only saves screen space but also facilitates information hiding. In Figure 4, the *Research Project Description* page containing five project pages and the *Students* page are packed into a horizontal bar at the bottom and into a vertical bar on the right, respectively, from the initial layout in Figure 1. Packed pages can be quickly restored to their previous sizes with a single click on the bar.

Overviews In the Elastic Windows browser, a hierarchy of visited pages can be collapsed into a hierarchicon providing an overview of those pages (Figure 5). A hierarchicon is an active thumbnail image of hierarchies of pages, which allows selection of any subhierarchy to be displayed on the screen. This facilitates fast access to a hierarchy of pages (Figure 6). Clicking on the same region more than once changes the depth of the selected subhierarchy.

At any time during exploration, a hierarchicon can be created for a hierarchy of pages by selecting from the window menu. As a result, a hierarchicon is created with the scaled image of the hierarchy of pages, added to the overview window. The selected subhierarchy is displayed in the same window as the top-level window of the hierarchicon replacing its contents. The overview window can hold more than one hierarchicon.

Hierarchicons best serve their purposes when the content of pages is rich with images or the text is well-structured. However, spatial characteristics (e.g. location, shape) of pages can also be used in the selection of the active group, though with more cognitive effort.

Since hierarchicons are only for visited pages, they do not serve as an overview of *all* the information in a site. The purpose of hierarchicons is to enable fast switching within a hierarchy of visited pages. However, sites might provide information in the form of images with URL addresses that allows the Elastic Windows browser to automatically build hierarchicons for accessing pages not yet visited.

Restructuring Pages

The structure of the information on the World-Wide Web is authored by individuals or groups of designers. Designers enforce a structure on the information by providing links between pages and pieces of information within a single page that indicate a certain relationship among information units. This structure guides the users in browsing the information and affects their navigational strategy directly.

However, the authored structure may not always match the structure desired by the users browsing the information for some tasks. Users with different backgrounds browse in-

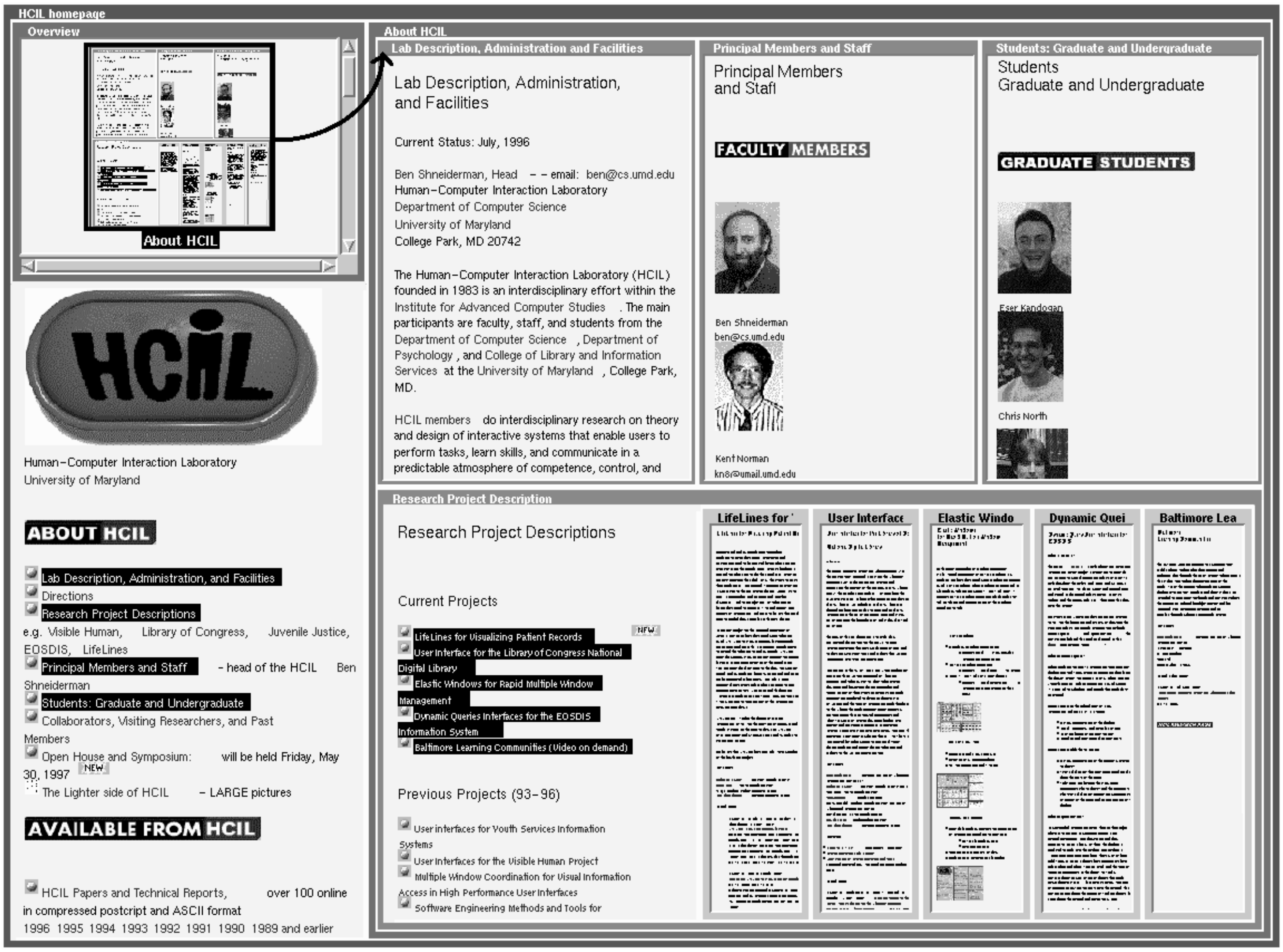

Figure 5: Hierarchicons: A hierarchicon for the *About HCIL* window is created as shown in the overview window

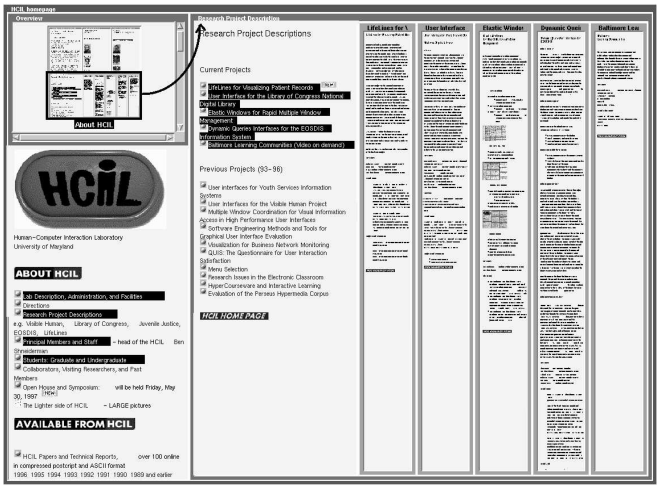

Figure 6: Hierarchicons: The hierarchicon for *About HCIL* window is used to switch the view to *Research Project Description* page and five projects

formation with a variety of goals and browsing strategies. Besides, in browsing goals are not well-defined and they may change based on the information collected. Changes in goals may necessitate changes to the structure of the information. Browsers should provide facilities that allow users to restructure the information to fit their needs.

The Elastic Windows browser allows users to restructure the information on the screen with efficient multiple window operations. Operations can be applied to a hierarchy of windows where effects of the operation are propagated down the hierarchy recursively. Thus, grouping, filtering, and restructuring of information can be done very efficiently.

Grouping Users can create groupings of pages, that are originally in different locations in the authored information structure. This allows users to gather interesting related information and operate on them as a group.

The Elastic Windows browser allows users to open a container window at any point in the hierarchy. A container window can be opened by double-clicking on the border of a window. Selecting links, dragging and dropping them inside the container window opens the pages for the selected links

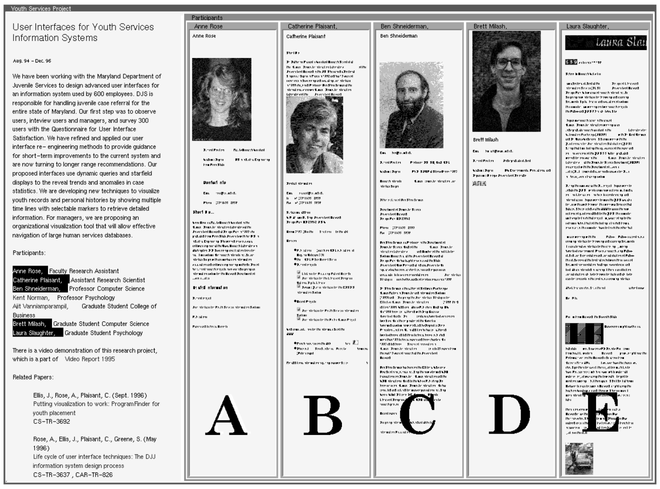

Figure 7: Grouping of participants pages (A, B, C, D, and E) in the HCIL working on the Youth Services project

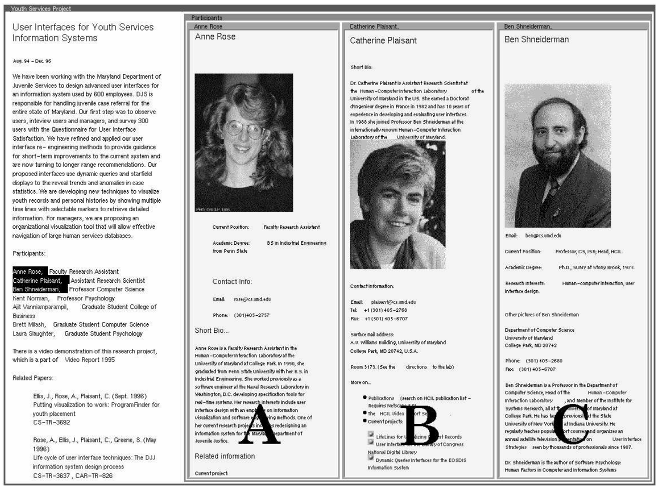

Figure 8: Filtering of the two programmers' pages (D and E) from the participants pages group of the Youth Services project from Figure 7

grouped in the container window. Pages can be added to or removed from an existing group at any time.

Figure 7 shows a grouping of pages of the participants that work on the Youth Services project in HCIL. Having opened them in a group allows the user to understand their responsibilities on the project, and compare their perspectives, backgrounds, educational levels, etc. Groupings can also be established from widely separated pages as well. One such example could be grouping of pages from projects related to Information Visualization.

Filtering When presented with many pieces of information, users typically desire to filter out uninteresting ones. This allows users to have a better focus on the interesting pieces of information by giving them more screen space.

The Elastic Windows browser allows users to filter out multiple pages in a group quickl by unselecting the corresponding links from their parent page. Alternatively, pages can also be filtered out from their window menus one by one. When a window higher in the hierarchy is closed all its children are closed recursively as well. This way a hierarchy of pages can be filtered out with a single operation.

Figure 8 shows filtering of the two programmers' pages from the group that contains all the participants' pages in the Youth Services project by unselecting the corresponding links. This way users can easily focus on the researchers' pages, eliminating the programmers' pages.

Modifying Hierarchy In browsing, goals are not well defined and may change as new information is received [16]. A change in goals might necessitate a change to the structure of the information. Facilities that allow users to change the information structure efficiently might help them perform their tasks by reorganizing the screen space.

In the Elastic Windows browser, a hierarchy of pages can be copied from and moved to different locations in the structure with efficient multiple window operations. Once the copy (move) operation is selected from the window menu, the cursor shape changes to indicate the operation. Then, clicking on the new location copies (moves) the selected hierarchy of pages to their new location.

Figure 9 shows a restructuring scenario. While the user is initially focused on the participants pages (A, B, and C) for the Youth Services project and then on Catherine's page (B), other projects by Catherine (1 and 2) become interesting. Although these two projects pages are initially opened within Catherine's page context, they are moved to a higher level for further examination. The new location of these pages is at the same level with the Youth Services project page.

Personalization Restructuring of visited pages and being able to save this structure allows users to use their personalized structure in later sessions. Users might find pages not only using the hierarchical structure but also recalling their spatial and visual characteristics. Thus, our approach might be superior to a hierarchical textual bookmark list.

Layout Dynamics

The Elastic Windows browser uses a space-filling tiled strategy for window placement. When a window is opened, closed, resized, or packed, the screen space is proportionally allocated for each window according to its previous size. Groups of windows stretch like an elastic material as they are being resized, and other windows shrink to make space. The extent of a window operation is the group window. The results of the operations are propagated to windows inside that group recursively.

Changes in the size of windows can cause automatic packing of windows which fall below a threshold size. Windows at any level might get packed, opening more space. Windows are packed into either horizontal or vertical bars appropriately. Packed windows are reopened as space becomes available. In Figure 10, initially the *Research Project Description* page is resized automatically packing the *Lab Description*, *Principal members*, and *Students* pages. Then, the Elastic Windows project page is resized, packing other research project pages.

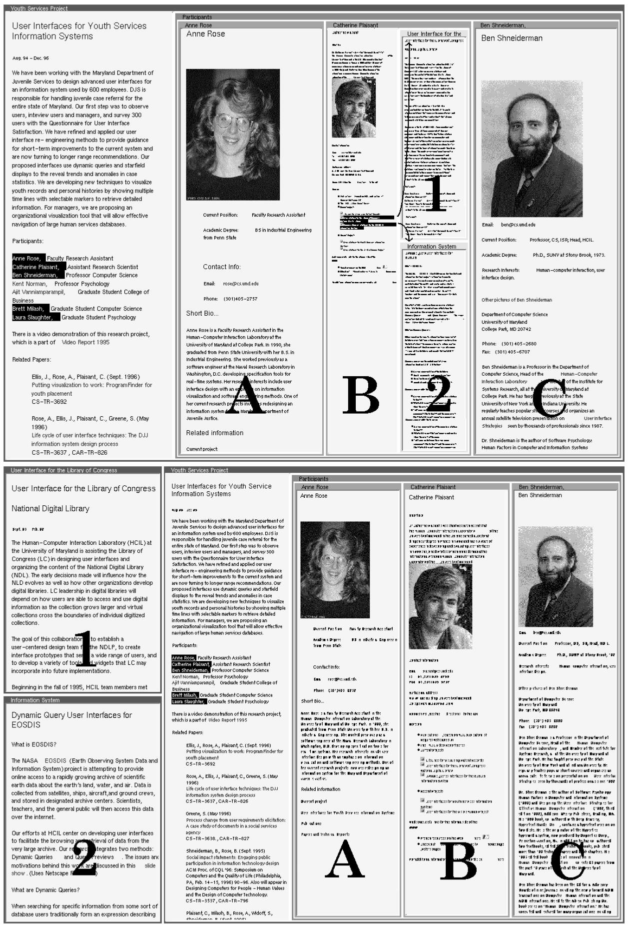

Figure 9: Restructuring: Two project pages (1 and 2) within Catherine's page (B) moved to the same level as the Youth Services project page

Tiled approaches may have an advantage in that they avoid wasted space and disturbing overlaps. On the other hand, in space-filling tiled approaches, window contents may not always conform to different window sizes and small windows may not provide sufficient view of contents. To provide better visibility in the Elastic Windows browser, contents are scaled according to the width of the window. However, for windows above a threshold width, contents are reformatted rather than scaled to display more information. We believe that scaling of contents preserves visual attributes of information which can provide context to help orient the user, and also provide an overview of its contents. It also allows easier recall for later accesses. However, highly demagnified pages might not be very useful especially for text-only pages.

Scaling of contents coupled with the automatic packing feature makes the Elastic Windows browser spatially-scalable. Users can add more and more pages without degrading the screen space utilization. However, more training might be needed for effective usage.

In the current implementation of the Elastic Windows browser, when multiple links are opened, pages corresponding to these links are opened by default side by side and placed within their parent`s window, sharing half of the space. Since the

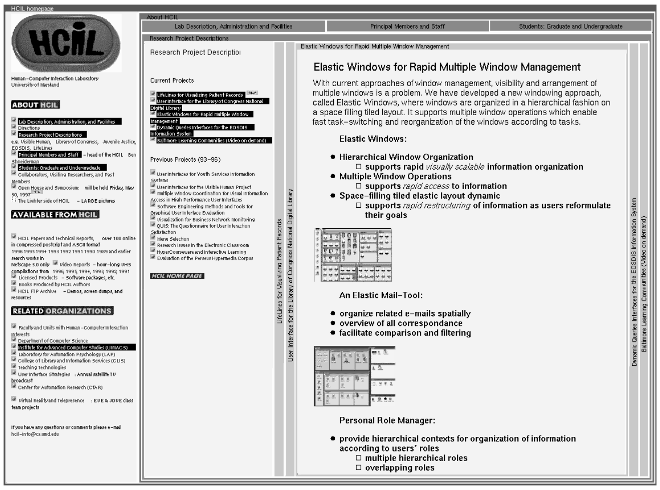

Figure 10: *Research Project Description* page and the Elastic Windows project page are resized automatically packing *Lab Description, Principal members*, and *Students* pages and the remaining research project pages on the layout in Figure 1.

contents of these pages are scaled, it gives users an overview of possible interesting pages. Alternatively, vertical or tiled placements can be selected. Space can be partitioned in different proportions, giving more space to new pages.

USER STUDIES REVISITED

- **High page recurrence rate:** Since a considerable percentage of pages are revisited according to studies, browsers should support fast access and easy recall of visited pages. Current browsers use only temporal properties for intra-session revisits (e.g. Back button, History lists), and textual hierarchical listings (e.g. Bookmarks) for inter-session revisits. The Elastic Windows browser employs a hierarchical page organization which preserves spatial and visual properties of pages and the location in the hierarchy. Thus, each of these properties may improve recall of a visited page and thus facilitate faster access. However, in the Elastic Windows browser at high (de)magnification factors, the visual properties might not be very useful, especially for text-only pages.

- **Continued growth of URL vocabulary:** According to the user studies, users continually visit new web pages, thus browsers should provide fast open for new pages. While current browsers facilitate fast open for a page with a single click, they fail to provide facilities that enable fast retrieval of previously visited pages. Approaches should be scalable not only in terms of space allocation, but also retrieval. In the Elastic Windows browser only a single action (e.g. click) is needed to open a single page, and two actions (e.g. select and click) are needed to open multiple pages providing fast access. Hierarchical organization, scaling of contents, and automatic packing make the Elastic Windows browser scalable in terms of space allocation. While the Elastic Windows browser offers more powerful screen management facilities, it might require more user training than the current one-window-one-click style browsers. The

Elastic Windows browser is also more scalable in terms of retrieval, since not only the location of pages but also their visual and spatial properties can be used.

- **High recency of revisits / Frequent use of backtracking:** According to the studies people tend to revisit pages just visited. In Elastic Windows, when a link is followed, it is possible to keep the parent page on the screen. This can improve user performance since the need to go back and forth between the parent page and children pages is eliminated. However, since two (or more) windows are on the screen at the same time, less space is given to each page. Scaling of contents is used to increase the amount of information displayed at the expense of smaller fonts and images, thus it might be harder to read page contents.

- **Frequent visits only to very few pages:** Since the Elastic Windows browser is a multiple window browser, frequently accessed pages such as personal, organizational pages and search engines can simply be left open in windows, preferably packed. Thus, while allowing easy access, it does not occupy much of the screen space.

- **Browse in small clusters of pages:** While studies found that users only browse in small clusters of pages, Elastic Windows browser is capable of clustering any number of pages in a group, facilitating comparison. However, as the number of windows increase the readability can decrease. Pages can be resized, scaling page contents, making them more readable at the expense of more screen management. It is also possible that this result is due to with the one-window-one-click browser used in the observation.

- **Short sequences of URL paths / Shallow browsing:** Studies indicate that users generally do not traverse deep nestings, however, this result might also be related to the browser used in the observation. The Elastic Windows browser supports arbitrarily deep traversals and the maximize operation facilitates focus at any level. However, the maximize operation might lead to disorientation when traversing deep hierarchies, though users can reorient themselves by returning to the previous hierarchy.

- **Infrequent save, print, hotlist addition and retrieval:** Although hotlist addition (save) is rather infrequent, there is a high possibility that pages in the hotlist are used as an intermediate step to reach other pages. The low percentage of these operations does not imply their usefulness. The Elastic Windows browser allows users to save visited pages for later sessions with efficient hierarchical window save operation that recursively saves all pages in a hierarchy.

IMPLEMENTATION

We implemented Elastic Windows using the Galaxy C++ application environment by Visix Software Inc. It runs as an X Windows application under Solaris. Modified NCSA Mosaic 2.5 libraries were used for retrieving and formatting hypertext documents.

Spatial organization of hierarchical windows is kept in an ordered tree with variable number of children at each node. Each level of the tree corresponds to a division of the available space in alternating horizontal and vertical directions. In order to overcome this strict alternating division imposed by ordinary ordered trees, a special type of node is used which functions as a space holder in the structure avoiding the division at that level. Packed windows are marked to indicate that their subwindows will not be drawn on the screen.

In order to allow direct access to nodes at any level of the tree, an array with links to the nodes in the tree is maintained. When initiating operations on windows, references to the nodes are made from this array, thus avoiding unnecessary traversals. Changes in the upper levels of the tree are propagated down the subtree to lower level nodes recursively, also avoiding unnecessary node traversals. More details on the window operations and layout dynamics are described in [14].

Scaling of page contents is done based on the window widths. Windows wider than a threshold value are reformatted keeping the same scaling factor. Galaxy libraries used in the implementation yielded sufficiently rapid graphics performance.

RELATED WORK

Research on web browsers provided many approaches. While some are add-on visualizations to browsers, others are standalone browsers. These approaches can be classified as:

- **2-D Graph:** WebMap [11], Graphic History View [2], Navigational View Builder [18]
- **3-D:** Harmony [1], HyperSpace [23]
- **Hierarchical Windows:** IGD [12], VIKI [17], Elastic Windows [14]
- **Zooming:** Pad++ [3]
- **Metaphorical:** WebBook [5], DecScape [4]

The above list is not complete, but presented here to give an idea of alternative approaches in a classification. The Elastic Windows browser has an advantage over the other hierarchical approaches in that multiple window operations increase user performance and flexibility in arranging pages on the screen. 3-D approaches might have a disadvantage in that valuable information might be occluded, though it is possible to transform the structure in such a way that brings the occluded information into view. Although metaphorical approaches follow concepts familiar to users, they may be limiting in terms of interaction and screen utilization.

Although many solutions are provided, a taxonomy of the approaches is needed to have a better understanding of the relative advantages and disadvantages. Such a taxonomy should address issues related to navigation, presentation, interaction, querying, (re)structuring, composition, and tailorability.

We believe that approaches should be evaluated thoroughly by user studies based on observations examining users' browsing tasks and navigational strategies. Among the web browsers, only a few are evaluated by user experiments [3]. We have conducted a user study comparing the Elastic Windows approach to the independent overlapping windows approach. Results indicated faster user performance for the Elastic Windows interface among expert users in the context of personal role management tasks [15].

CONCLUSION AND FUTURE WORK

We presented a hierarchical multi-window WWW browser, which we believe supports users in their navigation strategies. Hierarchical window organization and multiple window operations allow users to organize web pages, and efficiently restructure the information on the screen. We are working to improve the implementation with smooth transitions on layout updates, leading to decreased user disorientation.

We are planning to observe users' navigational strategies. Although recording of every user action in a browser would yield important information on navigation strategies, observations might be dependent on the browser used. Besides, users might have made compromises on their strategies that can not be seen in the data collected. Thus, such a data collection should also be supplemented by a think-aloud observation, providing more information on users' goals, strategies, and the compromises they made due to the browser interface. These observations will allow us to devise more realistic task sets for effective comparisons among approaches.

ACKNOWLEDGMENTS

We are grateful to Chris North for his comments on this work at various times, and Gary Marchionini and Catherine Plaisant for their review of the draft of this paper. Special thanks go to Visix Software Inc. for their donation of the Galaxy Application Environment. This material is based upon work supported by the National Science Foundation under Grant No. NSF IRI 96-15534, and by IBM.

REFERENCES

1. Andrews, K., Visualizing Cyberspace: Information visualization in the Harmony Internet browser, *Proc. Information Visualization'95*, Computer Society Press, (1995), pp. 97-104.

2. Ayers, E. and Stasko, J., Using graphic history in browsing the World Wide Web, *Proc. Fourth International World Wide Web Conference*, http://www.w3.org/pub/Conferences/WWW4/Papers2/270/, (1995).

3. Bederson, B. B., Hollan, J. D., Stewart, J., Rogers, D., Vick, D., Ring, L., Grose, E. and Forsythe, C., A zooming Web browser, To appear in *Human Factors in Web Development*, Ratner, Grose, and Forsythe, (Editors), (1997).

4. Brown, M. H. and Shillner, R. A., DecScape; An experimental Web browser, *Computer Networks and ISDN Systems 27*, 1, (1995), pp. 1097-1104.

5. Card, S., Robertson, G. and York, W., The WebBook and the Web Forager: An information workspace for the World-Wide Web, *Proc. CHI'96 Conference - Human Factors in Computing Systems*, (1996), pp. 111-117.

6. Catledge, L. and Pitkow, J., Characterizing browsing strategies in the World-Wide Web, *Proc. Third International World Wide Web Conference*, http://www.igd.fhg.de/www/www95/papers/, (1995).

7. Carmel, E., Crawford, S. and Chen, H., Browsing in hypertext: A cognitive study, *IEEE Transactions on Systems, Man, and Cybernetics 22*, 5, (1992), pp. 865-883.

8. Cockburn, A. and Jones, S., Which way now ? Analysing and easing inadequacies in WWW navigation, *Int. Journal of Human-Computer Studies 45*, 1, (1996), pp. 105-129.

9. Conklin, J., Hypertext: An introduction and survey, *IEEE Computer 20*, 9, (1987), pp. 17-41.

10. Cruz, I., Talk in Multimedia User Interfaces panel at *Advanced Visual Interfaces '96 Conference*, (1996).

11. Doemel, P., WebMap - A graphical hypertext navigational tool, *Proc. Second International WWW Conference*, http://www.ncsa.uiuc.edu/SDG/IT94/IT94Info.html, (1994).

12. Feiner, S., Seeing the forest for the trees: Hierarchical display for hypertext structure, *Proc. Conference on Office Automation Systems*, (1988), pp. 205-212.

13. Halasz, Frank G., Reflections on NoteCards: Seven issues for the next generation of hypermedia systems, *Proc. ACM Hypertext'87*, (1987), pp. 345-365.

14. Kandogan, E. and Shneiderman, B., Elastic Windows: Improved spatial layout and rapid multiple window operations, *Proc. Advanced Visual Interfaces '96*, ACM, (1996), pp. 29-38.

15. Kandogan, E. and Shneiderman, B., Elastic Windows: Evaluation of Multi-Window Operations, *Proc. CHI'97 Conference - Human Factors in Computing Systems*, ACM, New York, (1997), pp. 250-257.

16. Marchionini, G. and Shneiderman, B., Finding facts vs. browsing knowledge in hypertext systems, *IEEE Computer 21*, 1, (1988), pp. 70-80.

17. Marshall, C., Shipman, F., and Colombus, J., VIKI: Spatial Hypertext supporting emergent structure, *Proc. ECHT'94, European Conference on Hypertext*, ACM, (1994), pp. 13-23.

18. Mukherjea, S. and Foley, J. D., Visualizing the World-Wide Web with the navigational view finder, *Computer Networks and ISDN Systems 27*, 1, (1995), pp. 1075-1087.

19. Rosenberg, J., The structure of hypertext activity, *Proc. Hypertext'96*, (1996), pp. 22-29.

20. Rivlin, E., Botafago, R. and Shneiderman, B., Navigating in Hyperspace: Designing a structure-based toolbox, *Communications of the ACM 37*, 2, (1994), pp. 87-96.

21. Tauscher, L. and Greenberg, S., How people revisit web pages: Empirical findings and implications for the design of history systems, To appear in *International Journal of Human Computer Studies*, Academic Press, (1997).

22. Utting, K. and Yankelovich, N., Context and orientation in hypermedia networks, *ACM Transactions on Information Systems 7*, 1, (1989), 58-84.

23. Wood, A., Drew, N., Beale, R., and Hendley, B. HyperSpace: Web browsing with Visualization, *3rd International World-Wide Web Conference*, (1995), pp 21-25.