

8.0 Appendix II Save file format

The read routine ignores all lines contain the sharp sign '#' as the first non-space character. Pixmaps are not saved as a part of the file.

```
< DEPTH=num URL="url" TITLE="title" >
```

num :a number from 0 to N where N is the depth of the tree.

url :a URL in a format recognizable to NCSA Mosaic 2.4

title :The title of the document.

```
< DEPTH=0 URL="http://www.galcit.caltech.edu/~ta/cgi-bin/
asylhome-ta" TITLE="The Asylum" >
```

```
< DEPTH=1 URL="http://www.galcit.caltech.edu/~ta/lb/lb.html"
TITLE="The Asylum's Lite-Brite" >
```

```
< DEPTH=2 URL="http://www.galcit.caltech.edu/~ta/lb/top-
posts.html" TITLE="The Asylum's Lite-Brite Gallery - Staff
Favorites" >
```

```
< DEPTH=1 URL="http://www.galcit.caltech.edu/~ta/cgi-bin/
cclock-ta" TITLE="WWW Cuckoo Clock" >
```

```
< DEPTH=1 URL="http://www.galcit.caltech.edu/~ta/renstimp/
rsvote.html" TITLE="Ren & Stimp Feedback Page" >
```

```
< DEPTH=2 URL="http://www.galcit.caltech.edu/~ta/cgi-bin/
rsvote.pl" TITLE="Feedback Button Results" >
```

```
< DEPTH=3 URL="http://www.galcit.caltech.edu/~ta/renstimp/
slobber.slb" TITLE="Untitled, URL http://www.gal-
cit.caltech.edu/~ta/renstimp/slobber.slb" >
```

```
< DEPTH=1 URL="http://www.galcit.caltech.edu/~ta/inmates/
inmats.shtml" TITLE="The Asylum's Inmates" >
```

```
< DEPTH=1 URL="http://www.galcit.caltech.edu/~ta/core-
dump.html" TITLE="The Asylum's Core Dump" >
```

```
< DEPTH=2 URL="http://www.galcit.caltech.edu/~ta/cgi-bin/
bwall?main" TITLE="The Graffiti Wall has Moved" >
```

```
< DEPTH=3 URL="http://cybersight.com/cgi-bin/cs/gw?main"
TITLE="CyberSight Graffiti Wall: Let's Go" >
```

```
< DEPTH=2 URL="http://www.galcit.caltech.edu/~ta/revdoor/
oldurls.shtml" TITLE="The Asylum: Old Rev Door URL's" >
```

```
< DEPTH=3 URL="http://aurora.york.ac.uk/~steer/htdocs/
bships.html" TITLE="404 Not Found" >
```

```
< DEPTH=0 URL="http://www.cc.gatech.edu/grads/a/Eric.Ayers/"
TITLE="Index of
/grads/a/Eric.Ayers/" >
```

7.0 Appendix I

Additional X Resources:

These resources are defined in addition to Mosaic's resources. They all control settings the Graphic History view and share the same 'hist' prefix.

| Resource | Type | Default | Description |
|-----------------------|---------|--|---|
| histNodeColor | Pixel | Black | Specifies the color of the tree nodes. |
| histArrowColor | Pixel | Black | Specifies the color of the arrows that represent links in the tree. |
| histLineColor | Pixel | Black | Specifies the color of the lines that represent links in the tree. |
| histNodeWidth | Int | 75 | Specifies the default width of a node in the tree if the pretty drawing algorithm is disabled. |
| histNodeHeight | Int | 75 | Specifies the default height of a node in the tree if the pretty drawing algorithm is disabled. |
| histMaxNodeWidth | Int | 10000 | Specifies the maximum allowable width of a tree node. |
| histDisplayTitle | Boolean | True | Specifies whether the tree should display titles for each node by default |
| histDisplayUrls | Boolean | False | Specifies whether the tree should display titles for each node by default |
| histNodeFont | Font | "-*-helvetica-medium-r-normal-*-10" | |
| histDisplayThumbnails | Boolean | False | Specifies whether the tree should display thumbnail images for each node by default |
| histHelpUrl | String | "http://www.cc.gatech.edu/grads/a/Eric.Ayers/MosaicG/MosaicG_1.0_help.html" | Specifies the URL to lookup when the help menu item is invoked from the Graphic History View. |
| histAboutUrl | String | "http://www.cc.gatech.edu/grads/a/Eric.Ayers/MosaicG/MosaicG_1.0_about.html" | Specifies the URL to lookup when the about menu item is invoked from the Graphic History View. |
| histThumbnailX | Int | 64 | Specifies the width of a thumbnail image. |
| histThumbnailY | Int | 64 | Specifies the height of a thumbnail image. |
| histThumbnailScale | Int | 2 | Specifies the scale by which a thumbnail should be reduced. |

6.0 Bibliography

1. Andreessen. NCSA Mosaic Technical Summary. Technical report, National Center for Supercomputing Applications, 1993.

2. Doemel, Peter, "WebMap - A Graphical Hypertext Navigation tool", Second International WWW Conference, 1994.

<URL:<http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/doemel/www-fall94.html>>

3. Moen, Sven. "Drawing Dynamic Trees", IEE Software, July 1990.

4. Mukherjea and Foley (To appear), "Visualizing the World-Wide Web with the Navigational View Builder", Computer Networks and ISDN System, Special Issue on the Third International Conference on the World-Wide Web '95, April 1995, Darmstadt, Germany.

<URL:<http://www.cc.gatech.edu/gvu/people/Phd/sougata/Nvb.html>>

documents on the server and the user could then cheaply browse the documents in the space without the expense of downloading all the text and images of each page.

Another shortcoming of the Graphic History View is related to the restriction of most color workstations to display only 256 colors at a time. This limitation imposes severe restrictions on the thumbnails. The images contained in this paper were captured from an X Server running with 24 planes of color. Although it is often possible to determine the content of an image whose colormap has been changed, it is not always easy and the image is usually not visually appealing. Any user that regularly uses a WWW Browser under the X Window System on a color workstation with 8 bit color has likely run across problems with the colormap when trying to run two or more applications or trying to view two documents simultaneously. As graphic information becomes more pervasive and more easily accessible via the Web, the need to display many images simultaneously will grow, as will the perceived inadequacy of 8 bit color displays. Hopefully, 24 bit color will be a standard offering for future generations of workstations as it is already becoming for personal computers. An intermediate solution to this problem might be to create a private colormap for each thumbnail and swap the colormaps as the cursor is dragged over each thumbnail, or creating a private colormap for the thumbnail images in general, and attempting to generalize that color palette to fit all thumbnails.

In some preliminary reviews of MosaicG, users have expressed interest in having more power to manipulate the documents and tree structure in general. It has been suggested that a user might want to re-parent a node as the root of a tree, or erase branches of a tree completely. It has also been suggested that the Graphic History View concept serve as a model for a graphic hotlist, allowing node representations to be dragged and dropped between views and even applications. Such behaviors would be even more desirable in an object based document environment, such as Microsoft Windows' OLE specification. A formal usability study of the Graphic History view is in progress as of the writing of this paper.

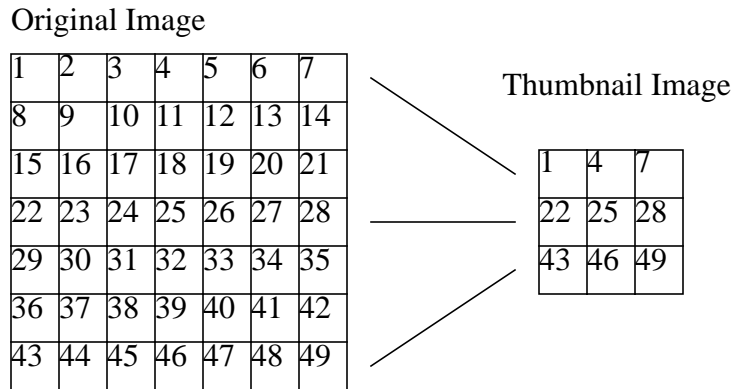


FIGURE 5. Demonstrating how pixels from an original image are taken to produce a thumbnail image assuming a scaling factor of 3.

The thumbnail images are generated by slightly modifying the HTML widget that Mosaic uses in the Document View. This generation of a thumbnail image adds a constant amount of processing time whenever a new document is accessed. The user can specify through an X resource file, the scaling factor to use for generating thumbnails, and the maximum width of a thumbnail image in pixels. After a document is retrieved, the widget performs a layout routine and renders an image of the document to the screen for the Document view. To generate the thumbnail, an off screen pixmap is created and the widget re-display routine is called, substituting the off screen pixmap for the visible window as the target of all Xlib drawing routines. The image is then scaled down by simply copying every Kth pixel of the pixmap where K is the scaling factor to use in generating the thumbnail as shown in Figure 5. Every Kth pixel in the horizontal direction is copied until the maximum thumbnail width is attained. Thus, the thumbnail image is a reduction of the top-left corner of the original document which represents the first $(K * \text{max_width})$ by $(K * \text{max_height})$ pixels in the original image. Note that no attempt is made to factor in the contribution of intermediate pixels to the thumbnails. This approach was taken to reduce problems with colormaps, although colormap problems are still not completely resolved. Using an anti-aliasing algorithm to sample groups of pixels would produce a smoother thumbnail image, but doesn't seem practical given the limitation of most workstations to 1 bit and 8 bit color.

5.0 Shortcomings

As mentioned earlier, the “exploratory” approach is expensive for both clients and servers. The “reflective” approach, on the other hand, has the disadvantage of only being able to show documents that have already been viewed. A happy medium between the exploratory and reflective approach could be achieved if some meta information about the structure of a server's document space were maintained by the server and made accessible via a single request. In this way, the server could quickly return the relationships between all

to the structure of the web at any other time. Servers may become unavailable, documents may change location, or anchors in a document may change.

4.0 Implementation

The Graphic History view was added to the Mosaic 2.5 source in as unobtrusive a manner as possible. Although NCSA Mosaic allows communication between separate processes through a shared file, the browser view needed to be able to access the graphic output of the Document View and be able to track the difference between following a link, typing a URL, pressing the back or forward button, or selecting a URL from the hotlist. For these reasons it was decided to write the view to be compiled in with the source code and not implemented as a separate application. In spite of the integration of the Graphic History source code into the NCSA Mosaic source, re-incorporating the browser code into a new release of Mosaic (from 2.4 to 2.5) took less than an hour.

History information is stored in a hash table and a tree structure that is separate from NCSA Mosaic's internal data structures. Care was taken to adhere to the same style of interface presented by Mosaic visually, while trying to keep the changes to the existing Mosaic source to a minimum. The view does modify the existing code slightly to inform the graphic history view when certain actions take place in the Document View, and modifications were made to the HTML widget to allow the widget to generate a copy of the Document View. The Graphic History view also relies on Mosaic's "Xmx" wrapper for Motif.

When a document is visited, a hashing function is applied to the URL to assign the document a slot in the hash table. If an identical URL has already been cached, no new node is created, but an indication of a new link is added to the nodes at the source and destination of the link. The browser makes no attempt to determine if two different URLs reference the same document, so sometimes the same document can appear more than once in the Graphic History View.

The tree layout algorithm is from Sven Moen's *Drawing Dynamic Trees* article published in IEEE Software, July 1990[3]. This algorithm draws ideas from other "tidy tree" drawing algorithms, optimizing them for drawing text. To describe the algorithm briefly, an outline is calculated for each node in the tree. The algorithm operates by recursively calculating an outline around each subtree from the leaves to the root of the tree. As the algorithm moves toward the root, it packs the nodes vertically by joining the outline of children to the outline of the parent. Children of sibling nodes are adjusted so that the outlines of subtrees do not cross. The end result is that the children of a node are packed in height and width, lining up the x coordinates of all children of a node, and centering the children of a node vertically about the node.

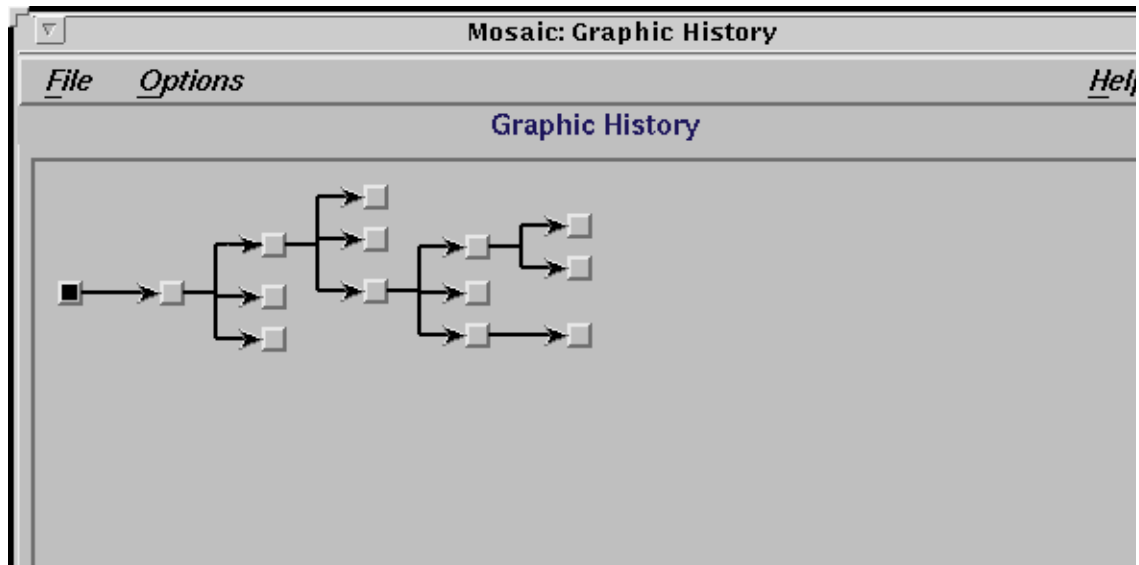


FIGURE 4. The Zoom Out feature.

The Graphic History View allows the user to zoom out to get an overview of the structure of the documents that have been visited as shown in Figure 4. Each node is collapsed to a small square. When the user places the mouse pointer over a node in the tree, the title and URL of the document are displayed at the bottom of the window. The view also allows the user to collapse a portion of the tree into a smaller representation by clicking on an arrow head pointing to a node.

The way the tree is built depends on the way documents are accessed. Documents that are accessed by selecting anchors in a document are added as child nodes in the tree relative to the node that represents the document where the source anchor is found. Documents that are opened by choosing a title from the hotlist or by entering a URL manually are added as the root of a new tree. Pressing the “forward” and “backward” buttons in the Document View changes the highlighted node in the Graphic History View but does not add new nodes to the tree.

One of the most difficult problems in visualizing the structure of documents on the web arises from the N to N relationship between documents. A tree is meant to show hierarchical relationships, but hypertext documents are traditionally more complex. The challenge for MosaicG was to present a view that is simple enough to understand at a glance without cluttering the image with extra lines and arrows. When MosaicG encounters a document that is the destination of more than one link, a short arrow appears to the left of the node. By positioning the mouse over this arrowhead, the other nodes in the hypertext that contain links to this document are highlighted.

The user may save a browsing session as a file using the “Save” command. The history tree is saved in a text file of the user's choice. The thumbnail images are not preserved when the tree is saved, but will be updated if the node is revisited. Note that there are no guarantees that the structure of hypertext web as constructed at one time will correspond

...” menu item opens another window that will display the sequence of documents visited by the associated Document View window. A separate Graphic History view is created for each Document View, each of which displays a different history. The Graphic History View allows the user to display different information as a part of the tree. The user can selectively display document titles, URLs, or a thumbnail image for each node. When the mouse is placed over a node in the tree, the title and URL of the document appears in the two text fields at the bottom of the Graphic History View Window. A user can recall a document in the tree by double clicking on a node in the Graphic History View window.

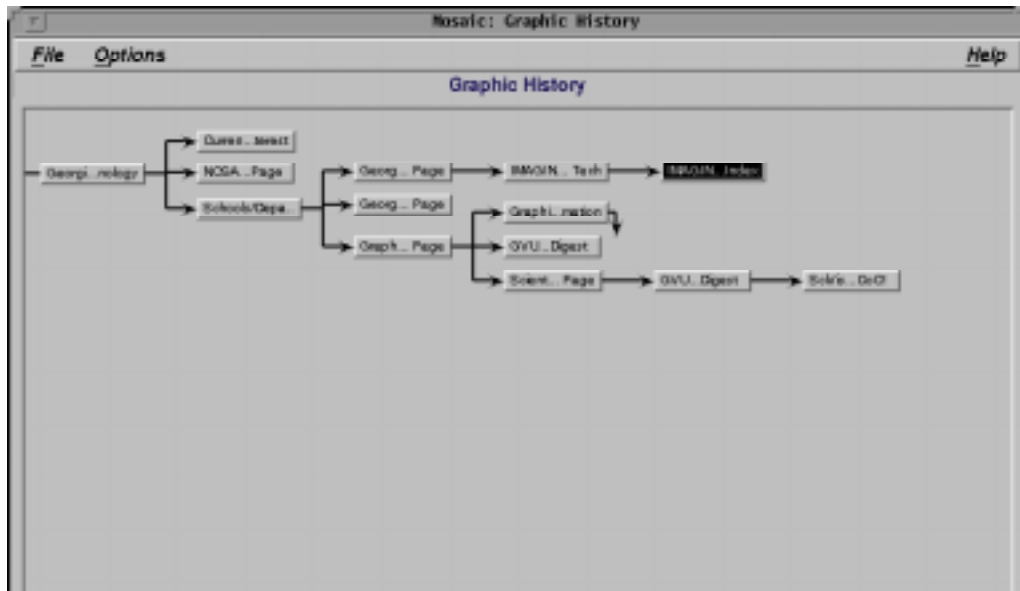


FIGURE 3. Demonstrating the title shortening algorithm

A daunting task for any visualization system is the management of screen real-estate. In the graphic history view, horizontal screen space is the most critical resource, as the trees grow from left to right. This layout seems the most natural for English text strings, but we find that the tree quickly grows off of the right side of the page. To conserve horizontal screen space, the Graphic History View shortens the titles of the documents as the tree grows to the right as demonstrated in Figure 3. The amount of abbreviation can be manually controlled by a scale on the Graphic History View's "Preferences" dialog box. By default, the program will try to increase the amount of abbreviation as the tree grows. This behavior can be controlled by enabling or disabling the "auto resize" menu item. The title shortening algorithm tries to preserve whole words in the title so the abbreviated title will make sense. It also tries to preserve whole words at either end of the title. It builds the abbreviated title back and forth from the beginning and end of the title, adding as many whole words as will fit at either end, and then adding characters to the title until the length of the title fills the width allotted for the node.

Graphic History software, it would be worth exploring this issue further and incorporating the option of displaying of additional types of information, such as relative distance between servers, relationships between servers in the same DNS namespace, or the frequency in which a page was visited.

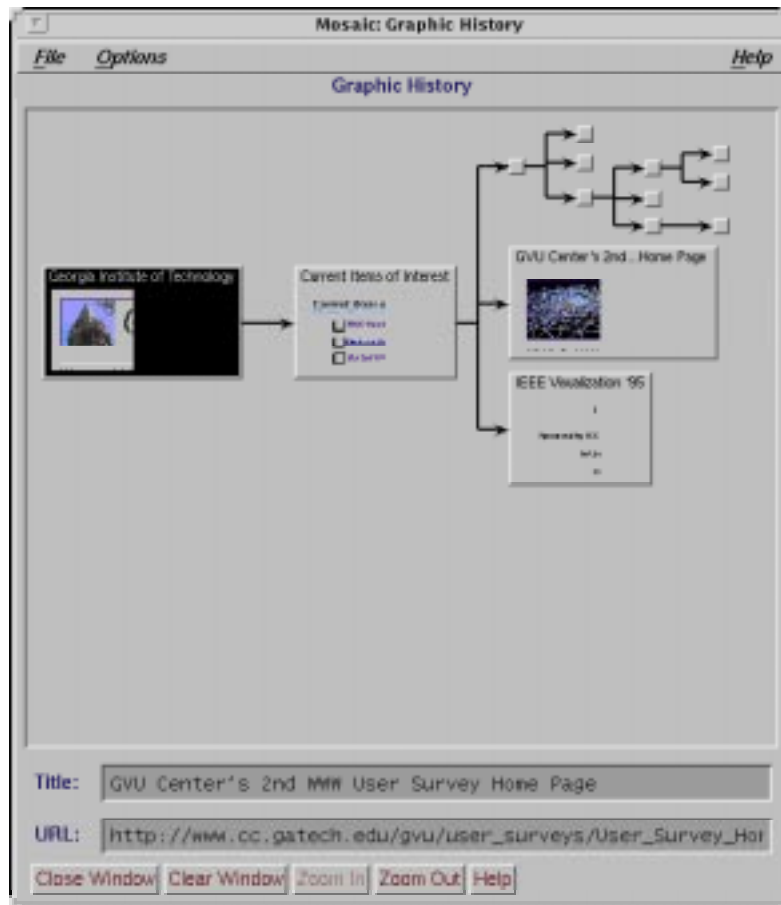


FIGURE 2. A portion of the tree is collapsed.

A primary consideration in the design of the browser was scalability. To eliminate visual clutter during a long browsing session, the view allows the user to condense branches of the tree which are no longer of interest to the user as shown in Figure 2. This not only saves screen real estate, but it also allows the user to easily focus on a smaller part of the visualization. The user may also get an overview of the visited documents by zooming out for a smaller representation of all documents in the tree.

3.0 Features

The menu of the MosaicG Document View is identical to the distributed NCSA Mosaic browser, with the addition of one menu item in the "Navigate" menu. The "Graphic His-

on a server or set of servers must be queried as a batch job to determine the structure of documents before the user attempts to browse the document space. This approach allows the user to visualize the space without having to visit any documents in the space, but comes at the price of many time consuming and resource intensive queries for the server or servers involved. These batch jobs must be re-run frequently to maintain an accurate representation of the document space because users are continually adding new documents to the server, and the contents of the documents themselves tend to be volatile.

In comparison, the approach of MosaicG might be described as a “reflective” approach, in that the representation is built passively as a user browses the document collection. The application makes no a priori assumptions of the structure of the space and builds the visualization only as new documents are encountered. Thus, the resulting visualization is customized to each session and is built to represent the way a user explores the hypertext.

In MosaicG, the history of the session is displayed as a two dimensional tree built from left to right. Documents are represented as nodes in the tree, and link between documents are represented as arrows in the tree. A document at the source of an arrow contains the source anchor, and the document pointed to by the arrow contains the destination anchor. Since a document can be both the source and destination of many links, the visualization includes special arrows to indicate that a document is the destination of more than one hypertext link.

The visual quality of most WWW documents is such that most pages have a distinctive look and feel. MosaicG uses thumbnail images of the documents to allow the user of a browser to quickly recognize a page or set of pages in the tree. A quick glance at a the thumbnail representation is often enough for a user to recognize a previously visited page. By building the structure to represent the history of browsing and by providing thumbnail images, the history browser attempts to lessen the burden of on the user to recall titles of pages by facilitating quick recognition.

As a part of designing the Graphic History View, an informal survey was presented to a small sample of the WWW user population. One part of the survey asked users about their browsing habits. Almost all users said they frequently used the backtracking navigation buttons provided by the Mosaic Document View. This is important because it is this type of browsing that produces the branching layout of nodes in the Graphic History view. Several respondents noted the problem of navigating back to documents that had been previously visited in a session, but none of them reported having used the text based history list to find a previously viewed document.

Another part of the survey also presented approximately 10 different options for displaying the tree of documents. Sample references to documents and a corresponding layout of nodes in a graph were presented with variations on layout direction (horizontal or vertical), ways to display nodes that are referenced as links from more than one page, and the use of color in the visualization. The responses to these different options varied widely. No particular layout or presentation strategy emerged as a clear favorite. Several of the respondents sketched novel schemes for creating such a visualization, while others wanted to see different types of information displayed. In preparing for a future release of the

ward” and “backward” buttons on the NCSA Mosaic Document View) to allow the user to return to the most recently visited documents in the list.

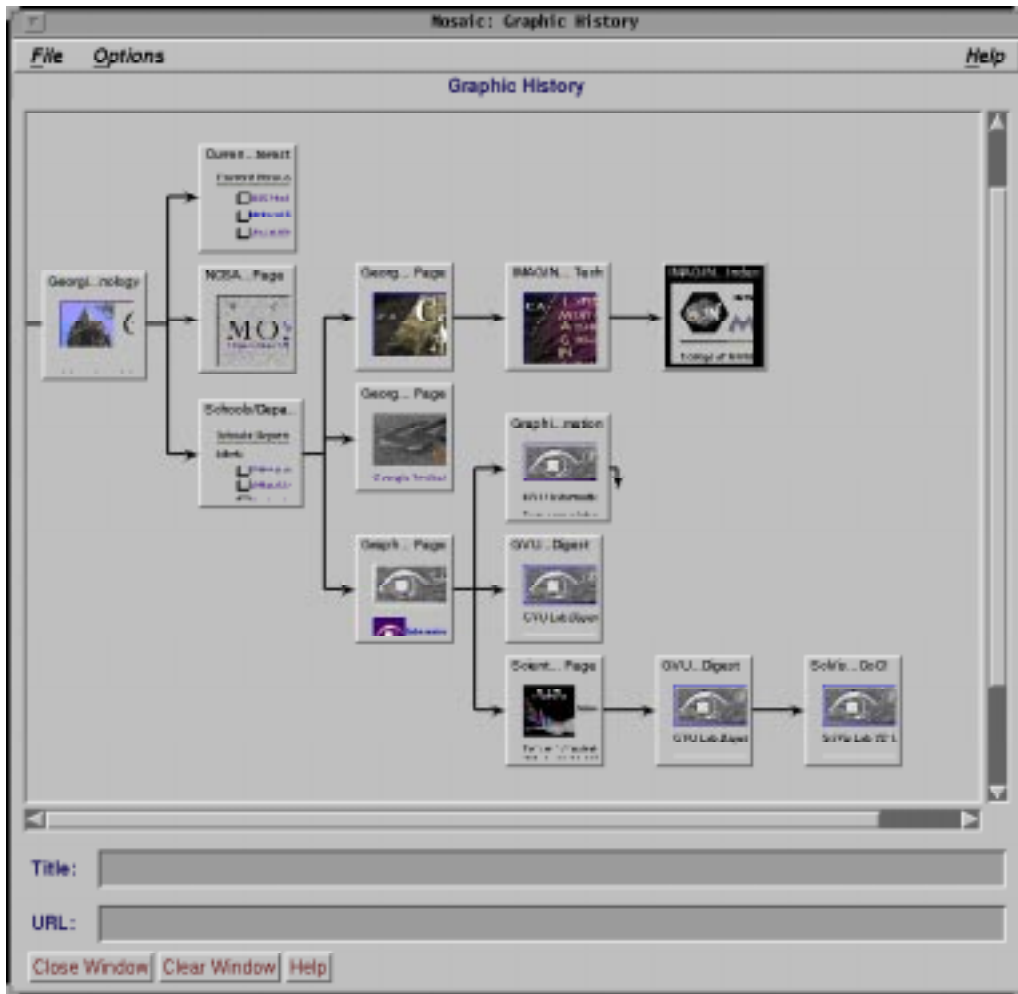


FIGURE 1. An overview of the Graphic History View

Although hypertext links can create a web-like structure of hypertext documents, many hypertext documents are arranged hierarchically. A user visits the document at the top level, traverses down the tree to read one subject in depth, and then backtracks up to the top node to find another subject. When the session history is viewed as a linear list of accesses, the top level document may appear in the list many times. This represents the frequency of access to the document, but does little to convey the hierarchical organization of the document space. MosaicG attempts to create an easy to understand visual representation of these types of browsing patterns to match the user’s mental model of the relationships between documents.

Other projects similar to MosaicG include WebMap[2] and The Navigational View Builder[4]. In these tools, the tool or an agent for the tool actively queries a set of documents and builds a representation of the relationships between these documents. This approach might be described as an “exploratory” approach. The structure of documents

1.0 Introduction

The use of the World Wide Web (WWW) has increased dramatically in just a few years. Its presence on the Internet has attracted the attention of researchers and the popular media alike. The availability of browsers for multiple computing platforms, many of them distributed for no cost, combined with new avenues for accessing the Internet allows even novice computer users with limited resources to make use of the wide range of services and information available on this global computer network. As electronic publishing on a large scale emerges, the advantages and drawbacks pertaining to hypertext systems have become familiar to a large user population.

To help users orient themselves in a hypertext web, browsers often provide a list of the documents a user has visited, a way to move forward and backward along previously traversed links, and a quick way to return to a home document. These navigation aids are essential in helping users manage the huge store of information available on the WWW. Hypertext links encourage users to explore related topics and references to other works from within a document. Although the backtracking aids and history list are helpful navigation tools, users often have trouble re-visiting a page that was previously viewed in a session. This problem becomes acute after many invocations of the backtracking shortcuts. Users of hypertext systems often find themselves eagerly following hypertext links deeper and deeper into a hypertext web, only to find themselves “lost” in the sense that they are unable find their way back to previously visited pages. This difficulty in re-visiting previously viewed pages may discourage users from engaging in such exploratory behavior. It is hoped that the addition of the graphic history view will encourage exploratory behavior and help users navigate the WWW more easily in general.

MosaicG is derivative work of the National Center for Supercomputing Applications’ (NCSA) Mosaic[1] Web browser, version 2.5. It enhances the history keeping facility of the browser by providing a two dimensional view of the documents a user has visited in a session, as shown in Figure 1. It is intended as an easy to use aid in navigating a collection of hypertext documents. By presenting titles, Uniform Resource Locators (URLs) and thumbnail images of the documents a user has visited in a session, the Graphic History View allows a user to easily recognize a previously visited document and provides an easy way for the user to re-visit that document and analyze the structure of a set of hypertext documents.

2.0 Overview of the Graphic History View

For the purposes of this paper, a browsing session is defined to be the history of document accesses during a particular invocation of a browsing application. Opening a new browsing window delimits the beginning of a new browsing session. Closing a browsing window delimits the end of a browsing session. Within a browsing session, users often revisit documents. Traditional hypertext browsing applications have maintained a list of documents visited during a session in the order in which they were traversed. Another session specific navigation aid that is often provided is a backtracking mechanism (i.e. the “for-

Using Graphic History in Browsing the World Wide Web

Technical Report GIT-GVU-95-12

May 1995

Eric Z. Ayers and John T. Stasko
Graphics, Visualization and Usability Center
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
zundel@cc.gatech.edu, stasko@cc.gatech.edu

keywords: WWW, navigation, hypertext, Mosaic, history

Abstract

Users of hypertext systems often find themselves eagerly following hypertext links deeper and deeper into a hypertext web, only to find themselves “lost” and unable find their way back to previously visited pages. As navigation aids to help users orient themselves in the web, browsers often provide a list of the documents a user has visited, a way to move forward and backward along previously traversed links, and a quick way to return to a home document. Still, users often have trouble re-visiting a page that was previously viewed in a session, especially after many invocations of the backtracking shortcuts.

MosaicG is derivative work of NCSA Mosaic[1] version 2.5 which enhances the history keeping facility of the browser by providing a two dimensional view of the documents a user has visited in a session. It is intended as an easy to use aid in navigating a collection of hypertext documents. By presenting titles, URLs and thumbnail images of the documents a user has visited in a session, the Graphic History View allows a user to easily recognize a previously visited document and provides an easy way for the user to re-visit that document and analyze the structure of a set of hypertext documents.

A postscript copy of this paper is available on-line:

<URL:file://ftp.gvu.gatech.edu/pub/gvu/tech-reports/95-12.ps.Z>