
A Tool for Individualizing the Web

K.A. Oostendorp¹, W.F. Punch¹ and R.W. Wiggins²

¹ Intelligent Systems Lab, Michigan State University
E. Lansing, MI. , oostend2@cps.msu.edu, punch@cps.msu.edu

² Computer Center, Michigan State University
E. Lansing, MI., rwwmaint@msu.edu

Abstract. The increasing complexity of navigating the Internet is becoming one of the fundamental obstacles to its effective use. This is due to the nature of the Internet, principally, a disorganized collection of both sites and site documents whose exponential growth rate rapidly is outstripping any user's ability to master it. There are two ways to deal with this complexity: reorganize the structure of the Internet or give each user the ability to organize an individual perspective of the Internet. Although the former would produce more global benefit, the latter is both easier to accomplish and potentially more beneficial to any individual or group of users.

Our approach, therefore, is to create a navigation tool which copes with Internet complexity at the individual, rather than the organizational, level. This tool, PAINT (Personalized, Adaptive Internet Navigation Tool), allows the user to impose a hierarchical organization on Internet sites and documents of interest by creating categories under which to group sites. Such categorization can be used not only by an individual user, but also can be shared among groups of users with similar interests. PAINT will also provide **local** automatic classification based on user parameters and user behavior. That is, PAINT will record visited locations and categorize them according to past use. The user is then free to examine the automated organization, modify it, and make it a personalized view of the Internet. In our report, we will describe the PAINT tool, its use, and some preliminary investigations of local, automatic categorization.

1 Introduction

The loose conglomeration of network-connected computer sites known as the "Internet" is showing growth that is nothing short of astounding, comparable only to the growth in computer chip capabilities/size-reductions seen in the previous decade. In the decade from 1984 to 1994, the Internet has grown from approximately 1000 computer hosts to over 2,000,000, a growth of 2000 times! For each of these computer hosts, it is conservatively estimated that at least ten users are granted access (on average) through each host, giving 20,000,000 direct access to Internet. Moreover, this growth rate is continuing with the

number of hosts increasing by ten percent a *month*(pg.9-10, [Wiggins 1995]). Although recent research has shown these numbers to be an overestimate, the growth rate is still phenomenal and likely to continue.

This incredible growth has created a situation comparable to one in which many large-database administrators and users find themselves, the problem of *data-mining*. In databases, the database is a large resource that must be examined or “mined” for a specific “nugget” of information hidden in the plethora of irrelevant “rocks.” The Internet has a similar problem, in this instance termed “netmining.” Users of the Internet are faced with the ever increasing problem of finding a desired file on some host, that is the “nugget” in the network “mine.” When faced with a ten percent growth rate every month, how can the user’s job of mining the net can be made easier, or at least feasible?

Further problems are developing based on the evolution of the Internet organization and access. For example, the World Wide Web (WWW or web) is an information organization of internet sites based on a hypertext model, navigable by clients like NCSA Mosaic. WWW and its client navigators have made the net much more accessible, but also intensify the netmining problem by convoluting the organization of sites.

1.1 Approaches for Aiding Netminers

The many approaches being developed to aid netminers can be roughly divided into three categories: standardizing approaches to making the Internet intrinsically more accessible, navigation tools that make mining the net easier, and personalizing access routes to the Internet/WWW.

1.1.1 Standardization

A number of groups are working on Internet standards, such as Standards for Universal Resource Identifiers (URIs) in the WWW [RFC-1630] and Standards for Interchange of USENET Messages [RFC-1036]. There are also other working groups concentrating on standards to allow better access to and broader distribution of Internet and WWW resources. Without such standards, the growth of the Internet and WWW will soon outstrip any user’s ability to effectively navigate and absorb information simply because of incompatibilities.

1.1.2 Navigation Tools

Many Internet navigation tools have been developed over the years. Some of these include ARCHIE (to gather information on FTP sites), Veronica (to gather information on Gopher sites), and other resource tools such as WAIS (Wide Area Information Servers). These tools attempt to discover the pertinent resources available on the net and categorize them, thus serving as a “network index” for users. This type of approach is starting to be applied to WWW, for example GENVL[McBryan 1994] and Aliweb [Koster 1994].

1.1.3 Personalizing Access Routes

Finally, there are approaches that allow the user to personalize their access to the Internet/WWW. With these tools, users can catalog and organize sites according to their preferences. Furthermore, when it seems to be useful, they can make those organizations available to others. Such tools include the Prospero file system [Neuman 1992a, Neuman 1992b] and the Alex file system [Cate 1992].

1.1.4 Which to Choose?

All three of the above approaches address important problems associated with better, easier to use, net access. However, in this paper, we address the third kind of approach, that of personalizing net access. Personalized access has the following aspects that make it a more likely candidate for further exploration.

- Personalized access is done on an individual level. The other techniques (standards and large-scale net indexing) require cooperation with other groups or other net sites. While those approaches might result in broader impact, they require cooperation at a level that could easily make the work go more slowly.
- Because personalized access does not require group cooperation, it can be idiosyncratic. Thus users across the net can set up their own personal “view” of the net, navigating it and using its resources. This seems more in tune with the character of the net, which is not an organized entity but a loosely connected set of sites each participating in the net while making their own choices as to what to present, how to present it, etc.
- The results of personal access modification affect no one but the person doing the changes and therefore would tend to be more readily accepted.

The rest of this paper will accomplish the following tasks.

1. Discuss how we plan to approach personalized access.
2. Describe an implementation of personalized access for Mosaic: the PAINT tool.
3. Discuss the possibility of *adaptive* personalization, which would provide personal modification to sites based on use as well as user preference.

2 Models for Personalized Access

2.1 An Existing Approach, Prospero and the Virtual System Model

As mentioned above, one of the systems addressing the personalized access approach to the net is Prospero [Neuman 1992a, Neuman 1992b]. Neuman’s thesis stated that existing approaches, based on a centralized server, were inappropriate for organizing access to an intrinsically distributed, large scale system. As such he developed the Virtual System Model, an approach that presented four principal features, listed below.

- Support for customizable name spaces
- Tools to support the construction of name spaces
- Support for synonyms in a name space
- Methods for selecting an appropriate name space when resolving the address of a name

In the construction of a name space – a mapping of a personal name and a network address – Neuman describes personalized access. However, the goals of Prospero and the Virtual System Model are much more than just personalized access. Prospero provides tools for: directory filtering, formal models of naming and other aspects of a distributed system, authentication and authorization, processes and processors and a host of other features.

2.2 Models for Personalized Access on the Web

Our main goals are in the same vein as those provided by Prospero and the Virtual System Model. However, we wish to focus on providing personalized access in as simple a form as possible, thereby making the implementation less cumbersome. Therefore, we will simply consider the mappings between names and addresses and their organization. By concentrating solely on personalized access, we can provide more features that serve personalized access. In this way, our work can be viewed as specializing a part of the work being done by Prospero and other systems that currently organize distributed systems.

In our work, we will focus on the following goals.

1. Create a scheme that supports the personalized mapping of name-to-address, along with some organization of that mapping.
2. Keep that scheme as simple as possible, thereby supporting easy sharing of personalized schemes and simplifying manipulation of the user's organization.
3. Provide user-tools, particularly GUI tools, that allow modification to both the name mapping and name organization.
4. Make the system available so that it can have immediate impact on the present user community, with a focus on WWW and Mosaic.
5. Investigate the possibility of having not only the user modify the organization, but also allow for automatic organization updates by the tool. This automatic organization would be under user control and based on the user's usage of sites and preferences. In this way, we incorporate *adaptive user interface* technology into the access model.

3 PAINT

PAINT (Personalized, Adaptable Internet Navigation Tool) is our first realization of the five goals delineated in Section 2.2.

We addressed Goal 4 first, namely how can we have the largest impact on the WWW community? Clearly the most usable, the most available tool presently for navigating the web is Mosaic. Furthermore, two other things are obvious based on a survey of web users provided by Pitkow [Pitkow and Recker 1994]. First, people are largely very satisfied with Mosaic as a tool. On a scale of 1(horrid)-9(excellent), 1081 respondents gave a mean rating of 8.086. Adding capability to a tool people seem most pleased with will increase chances of wide acceptance. Second, of 1079 respondents, 1006 (93.25%) were using a Unix version of Mosaic. Therefore our first implementation would be based on the latest Unix implementation of Mosaic (version 2.4).

Presently, Mosaic 2.4 does have a simple name-to-address mapping available in the *hotlist*. However, the hotlist provides simply a linear list of names gathered by saving the presently viewed page to the list. The user can then change the name by which the address is known.

Our approach was to replace the hotlist in Mosaic with the PAINT program. PAINT would be developed as a separate program, but interact through the hooks used by Mosaic's hotlist facility. This would minimize interactions between the two programs and allow for quicker development.

We then addressed Goals 1 and 2, namely developing a simple scheme for mapping name-to-address sites that also maintained some organization for those sites. The scheme we chose was a tree organization for the name-to-address map that would store arbitrary text descriptions of the node along with a name and an address. If a node in the tree was a leaf, that node represented a network address. If the node was not a leaf, it was considered an organizational node, or the parent of a set of leaves. As such, organizational nodes will never have associated addresses.

An example is shown in Figure 1. This figure is a text-file representation of an organization of WWW URL's. The first node in the file, the Root node, is the basis of the entire tree. All other nodes are labeled by four fields: address (for leaf nodes), name, parent node, text description (optional). The use of a text file allows the user (or another program) to easily manipulate the organization of the map. Furthermore, this text representation of the map could easily be shared between groups.

Meeting Goal 3, constructing a GUI that would provide easy interaction with such a map, was the next goal. The actual PAINT tool that manipulates the underlying text scheme of Figure 1 is shown in Figure 2.

The four consecutive panes in the top of PAINT represent a directory structure viewer. This viewer is the same type of viewer used in the NeXTStep® and Smalltalk environments. Each pane represents the nodes contained in a particular level of the tree. The nodes displayed in a pane are those nodes which are the children of the particular node (which is darkened) selected in the pane to its direct left. Furthermore, the selected node in each pane is displayed above the pane. In Figure 2, the leftmost panel displays the Root node, the pane to the right of it displays the children of the Root node, and the next pane to the right displays the children of the Web Resources node. If the selected node is a leaf node, then no children will be displayed.

Directly below the directory structure viewer are two text fields and another viewer pane. These display information about the presently selected node. Continuing with Figure 2, The NCSA Mosaic Home Page is the most recently selected node (the deepest in the tree). Asso-

Root

A Directory

GA Pages

Root

A Directory

Web Resources

Root

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html>

NCSA Mosaic Home Page

Web Resources

THE Home page, from the people who brought you Mosaic

<ftp://ftp.rpi.edu/pub/communications/internet-cmc.html>

Information Sources: the Internet and Computer-Mediated Communication

Web Resources

<http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html>

The World-Wide Web Virtual Library: Subject Catalogue

Web Resources

Figure 1: An excerpt from our example scheme for organizing name to address maps using a tree organization

ciated with it are its URL

(<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html>), a user-selected and modifiable name (NCSA Mosaic Home Page), and an arbitrary text description entered by the user. If the selected node is not a leaf node, then the address is not active and cannot be filled by the user, though the name and description can be modified. Any modifications made in PAINT are updated in the associated text file when the user exits the program. In accordance with Mosaic's hotlist, this file is called `.mosaic-paint`. If no such file exists, PAINT will read the current `.mosaic-hotlist` file and create a `.mosaic-paint` file which has all the hotlist entries as children of a default Root node.

In keeping with the theme of simplicity, there are only five commands available to the user, represented as buttons at the bottom of the PAINT tool. **Add Node** allows the user to create a new node, which is added as a child to the presently selected node. Nodes *cannot* be added to a leaf node (a node containing an address). **Remove Node** removes the presently selected node. Nodes with children *cannot* be removed – this is a safety feature in the tool. **Send URL** sends the URL to Mosaic so that Mosaic may access the network site. **Quit** dismisses PAINT.

Finally, the **Move Node** button allows the user to move a node, and its associated sub tree (if any) to other parts of the structure. The intent of **Move Node** is to make the presently

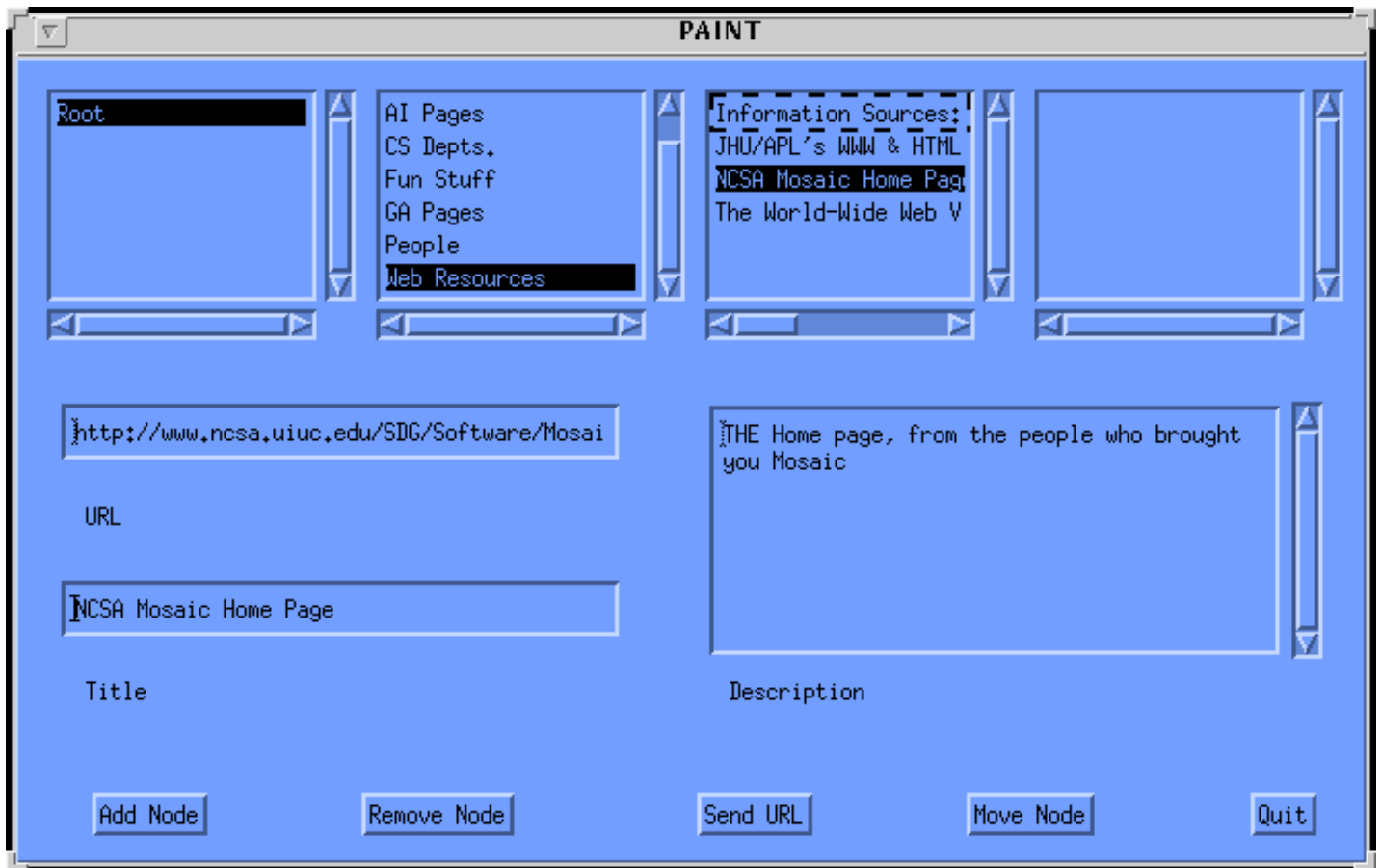


Figure 2: The PAINT GUI

selected node a child of some other part of the directory structure. Thus, when **Move Node** is selected, a list of all available parent nodes is presented to the user. The user selects a node from the list and the presently selected node becomes a child of that node. The directory structure viewer then redisplay the new structure.

3.1 Implementation

PAINT was developed on a Sparc20 under Solaris, interacting with Mosaic 2.4. PAINT was developed as a separate program using iXBUILD, a Motif development package. It was then linked with Mosaic, to make the final product.

4 Adaptable Personalized Access

Work in the area of *adaptive* or *intelligent* interfaces has been a research focus in human-computer interaction area for some time. However, in recent years some of this work has become the focus of computer science researchers as well [Schneider-Hufschmidt et.al. 1993,

Sullivan and Tyler 1991]. Defining adaptive interfaces is difficult, especially given the interdisciplinary nature of the field, but Schneider-Hufschmidt [Schneider-Hufschmidt et.al. 1993] note four general goals that adaptive interface researchers address: adapting a system being used by users with different requirements, adapting a system being used by a user with changing requirements, adapting a system to a user working in a changing environment, and adapting a system to a user working in different system environments.

For our purposes, we are focusing on the second approach: adapting an interface to a user with changing requirements. From the point of view of creating an adaptive interface for Web/Internet browsing, we see two general approaches.

1. We could adapt the interface itself based on how the user interacts with the interface. That is, we could make certain functions easier to access, such as moving a function from being activated as an item in a pull-down menu to being activated by a push-button. Such a move makes that function more directly accessible. Similarly, rarely used functions could be moved to less directly accessible interface points. The interface is adaptable because the interface changes based on user action over time. Those functions used most often become more accessible, the rarely used functions remain accessible but do not occupy precious direct-access real-estate on the interface. Furthermore, if the actions of the user change, such as a new function being used more often, the interface can adapt.
2. We could adapt the *organization* of the information the interface presents to the user. In the case of a tool such as PAINT, this could mean that the tree of web/internet nodes could be modified. The questions to be answered are: what kind modifications would be most useful and how does the user's action trigger and determine these modifications?

We have considered both options for the PAINT tool but have focused mostly on the later, adapting the organization of the tree of internet/web nodes. In fact, our original plan was to work on the former approach (actually changing the interface itself), but there are a number of problems with this approach that make it more difficult. Some of these problems include controlling the rate of change such that the user sees some effect but is not swamped by a constantly changing interface and integrating user preferences into the interface changes.

Given that our focus, at least for the short term, is on changing the tree organization, we return to the questions of the kinds of modifications to impose on the tree and how the user triggers these changes. Our first cut at this, not yet implemented in the PAINT tool itself, is to use the frequency and sequence of visitation to determine the categories under which leaf nodes – representing actual internet/web sites – are placed. For example, an upper level tree node, representing the highest level of division of leaf nodes, could be created such that it separates often used, lower-level nodes from less used nodes. Furthermore, nodes that are often “co-visited,” that is nodes that are typically visited within a small period of web visitations, would be candidates for being placed in the same category.

This process, while simple, might suggest to the user some new organizations of the tree of internet/web nodes that would make relocating important sites easier. Suggesting is the key word, since we cannot allow the interface to usurp control from the user. Thus PAINT

will not modify directly the user's organization, but propose a new organization and allow the user to select those parts, if any, s/he wishes to incorporate into the tree.

5 Conclusions

PAINT is a tool that allows a user to organize a personal view of the internet/web. Our first implementations of PAINT are incorporated into Unix-Mosaic as a replacement for the hotlist facility. We are presently working on expanding the capabilities of PAINT to include techniques of adaptive user interfaces, giving the user some outside support in analyzing how they use the net and making recommendations on how to do so more efficiently.

References

- [RFC-1630] T. Berners-Lee (CERN), "Universal Resource Identifiers in WWW, A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", Network Working Group, RFC-1630, June 1994.
- [RFC-1036] M. Horton (AT&T Bell Labs), R. Adams (Center for Seismic Studies), "Standard for Interchange of USENET MESSAGES", Network Working Group, RFC 1036, December 1987.
- [Cate 1992] V. Cate, "Alex—A Global Filesystem", *Proc. Usenix File System Workshop*, pp. 1-11, Ann Arbor, MI, May 1992.
- [Koster 1994] M. Koster, "Aliweb—Archie-Like Indexing in the Web", 1st International Conference on the World Wide Web, Geneva Switzerland, May 94.
- [McBryan 1994] O.A. McBryan, "GENVL and WWW, Tools for Taming the Web", 1st International Conference on the World Wide Web, Geneva Switzerland, May 94.
- [Neuman 1992a] B.C. Neuman, "The Virtual System Model: A Scalable Approach to Organizing Large Systems", Ph.D. Dissertation, University of Washington, 1992.
- [Neuman 1992b] B.C. Neuman, "Prospero: A Tool for Organizing Internet Resources", *Electronic Networking: Research Applications and Policy*, 2(1), pp 30-37, Meckler Publication, Spring 1992.
- [Pitkow and Recker 1994] J.E. Pitkow and m. Recker, "The Results from the First World Wide Web user Survey", 1st International Conference on the World Wide Web, Geneva Switzerland, May 94.
- [Schneider-Hufschmidt et.al. 1993] M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (Editors), Adaptive User Interfaces, Elsevier Science Publishers, 1993.
- [Sullivan and Tyler 1991] J.W. Sullivan and S.W. Tyler, Intelligent User Interfaces, ACM Press, Frontier Series, 1991.

[Wiggins 1995] R.W. Wiggins, The Internet for Everyone, McGraw-Hill, 1995.