Navigating Large Hierarchical Space Using Invisible Links

Ming C. Hao, Meichun Hsu,
Umesh Dayal, Adrian Krug*
Software Technology Laboratory
HP Laboratories Palo Alto

HPL-2000-8
January, 2000

E-mail:(mhao, mhsu,dayal)@hpl.hp.com
akrug@hpugrca.grc.hp.com

invisible link, navigation, hierarchical space, cycles, cross links.

To date, many web visualization applications have shown the usefulness of a hyperbolic tree. However, we have discovered that strict hierarchical tree structures are too limited. For many practical applications, we need to generalize a hyperbolic tree to a hyperbolic space. This approach results in massive cross-links in a highly connected graph that clutter the display. To resolve this problem, an invisible link technique is introduced. In this paper, we describe the navigation in a large hyperbolic space using invisible links in some detail. We have applied this invisible link method to three data mining visualization applications: e-business web navigation for URL visits, customer call center for question-answer service, and web site index creation.

* Hewlett-Packard GmbH, Pempelfurt Strasse 1, 40882 Ratingen, Germany

Navigating Large Hierarchical Space Using Invisible Links

Ming C. Hao, Meichun Hsu, Umesh Dayal, Adrian Krug
Hewlett Packard Research Labs

ABSTRACT

To date, many web visualization applications have shown the usefulness of a hyperbolic tree. However, we have discovered that strict hierarchical tree structures are too limited. For many practical applications, we need to generalize a hyperbolic tree to a hyperbolic space. This approach results in massive cross-links in a highly connected graph that clutter the display. To resolve this problem, an invisible link technique is introduced. In this paper, we describe the navigation in a large hyperbolic space using invisible links in some detail. We have applied this invisible link method to three data mining visualization applications: e-business web navigation for URL visits, customer call center for question-answer service, and web site index creation.

Keywords: Invisible Link, Navigation, Hierarchical Space, Cycles, Cross Links.

1. INTRODUCTION

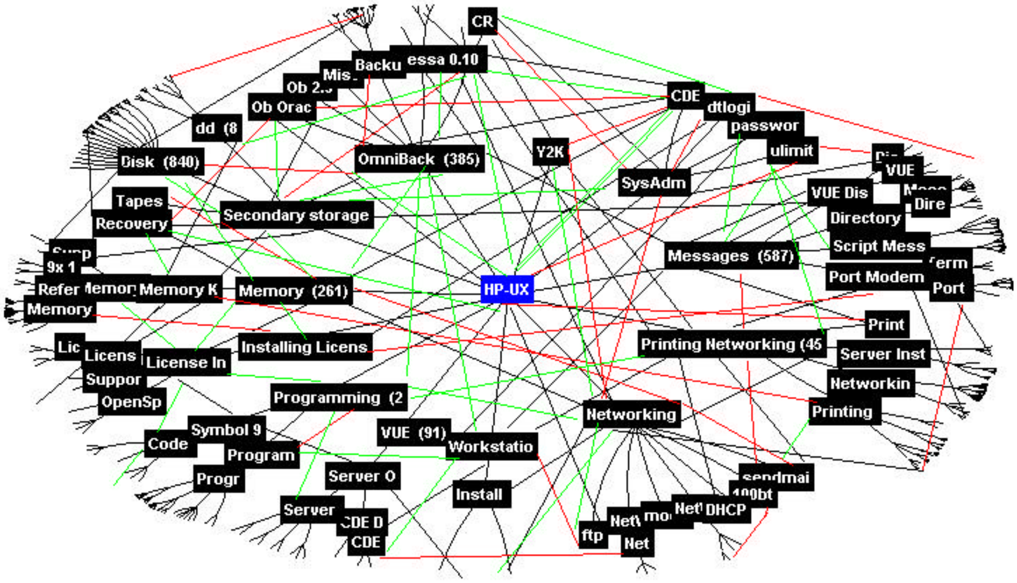
To date, many web visualization applications have shown the usefulness of hyperbolic trees [1, 2, 3, 4, 5, 6, 8]. Hierarchies, such as organization charts or directory structures can be represented in a hyperbolic tree. Also, there are other applications, such as navigating E-Business URL paths and file system links in which relationships need to be linked from different branches of a tree. For example, we can map an e-business application's structure including URL's and their subsequent links into a hyperbolic space. But we cannot put all the URL links into a strict directed tree. Usually there are several cross-links, and they might even form cycles. In a customer support web application, hyperbolic trees can be used to organize large numbers of questions and answers in a hierarchical structure. It is still sometimes necessary for an answer to link to a set of questions/answers in another group that does not directly belong to the hierarchical hyperbolic tree. To expand its use in a wide variety of visual navigation applications, it is necessary to generalize a hyperbolic tree into a hyperbolic space.

A common method for visualizing all connected nodes in a hyperbolic space is to draw lines. This introduces many lines and intersections. Also, there are broken lines for far away nodes that are off the screen. As illustrated in Figure 1, the hyperbolic space becomes very cluttered and difficult to navigate.

In this paper we discuss an "invisible link" [9] technique with link tables and navigation path histories for linking a large highly connected graph without cluttering the display. Only the primary (direct parent-child, no-cross) links are shown on the screen. All other cross-links are invisible to the user.

* Correspondence: Email: (mhao, mhsu, dayal) @hpl.hp.com; akrug@hpugrca.grc.hp.com

Figure 1: A Conventional Method: include highly connected links



2. INVISIBLE LINKS & LINK TABLES

2.1 Invisible Links

To unclutter the display and ease navigation in a large highly connected hyperbolic space, we employ an invisible link technique. This technique only shows the parent-child path of each node. All other interconnected links are defined in each node's link table. They are invisible to the user. These invisible links only appear after the user focuses on the node.

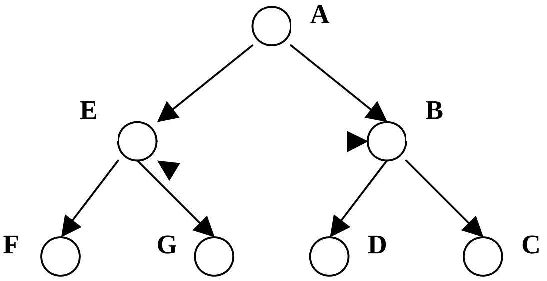
When the user clicks on a node that contains a number of invisible links, these invisible links will be drawn from that node. Each invisible link represents a path to a node in a different branch of a hierarchical space. The user can dynamically select any one of the paths. After the user makes the selection, the selected node with its children will be automatically mapped from their original structure to the current focused node. At this time, the selected nodes will be unmapped from their original structure. Placeholders are used to save/restore links for mapping and unmapping operations. Also, placeholders are used to track cycles.

This technique retains the original simplicity of a hyperbolic tree without having extra lines and intersections.

2.2 Link Tables

The Link Table is defined at hyperbolic space initialization time. There is one Link Table that contains one entry per node in a hyperbolic space. Figure 2 illustrates the contents of a link table.

Figure 2: Link Table



primary incoming links:	i.e.	B->D (for node D)	A->B (for node B)	A->E (for node E)
secondary incoming links:		none	E->B	D->E
immediate out-going link:		none	B->C; B->D	E->F; E->G
invisible link: (1-n cross links)		D->E	none	E->B

For example: node B, node D, and node E form a cycle BDE

2.3 Link History Log

The link history log records previous link paths of a node while it is in focus. It contains information for mapping a cross node and its children that are pointed by an invisible link from their original structure to the current focused nodes. The link history log contains two parts: the link (path) and operation (actions)

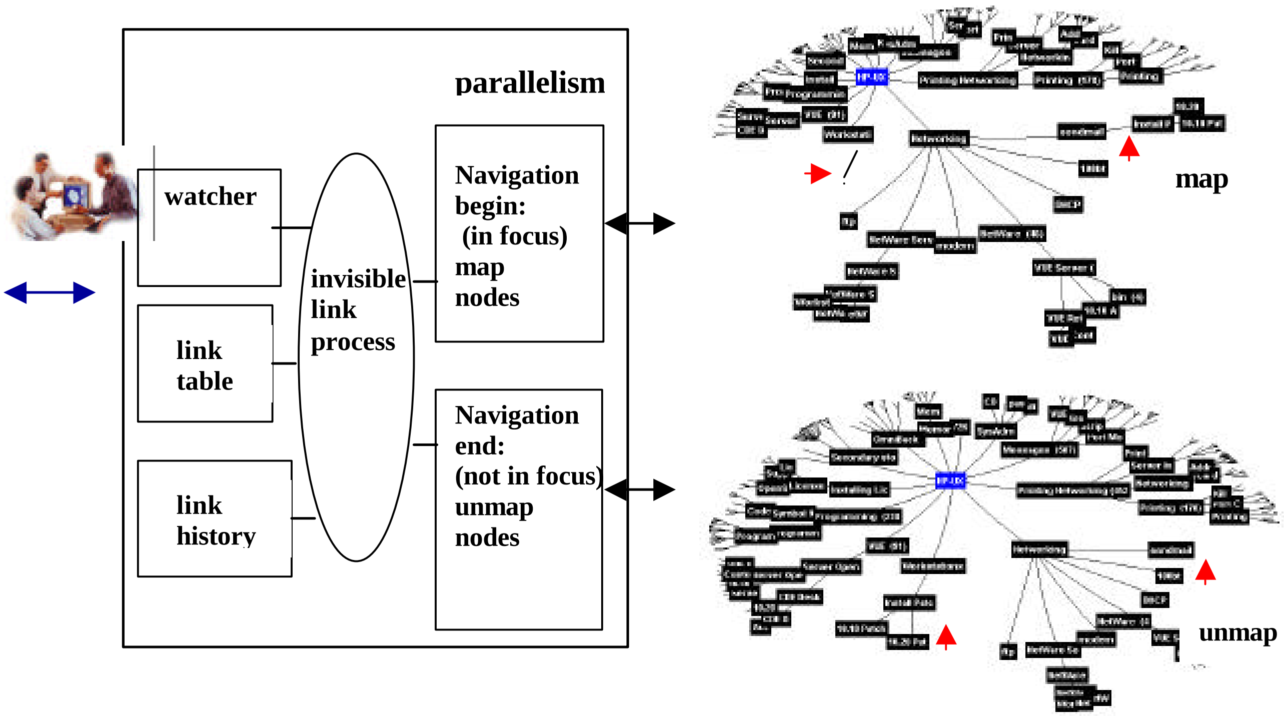
Link	**Operation**
navigation (in focus)	if the current node has invisible links ->display all invisible paths
map	map the node and its children connected by a selected invisible path
unmap	unmap the node and its children from their original structure
placeholder	hold the places of the node in their original structure
navigation end (not in focus)	unmap the node from the current node and re-map the nodes back to their original structure pointed by the placeholder
cycle	track the current node's ancestors for managing loops

2.4 Process Flow

Figure 3 illustrates the process flow using the invisible link technique. This technique contains the following elements:

1. Use an invisible link to automatically map nodes from different branches of a hierarchical space to the current focused node.
2. Use the link table to define the parent-child relationships and interconnection of each node.
3. Use the link history log to track navigation activities and to maintain cycles.

Figure 3: Navigating Large Hierarchical Space Using Invisible Links

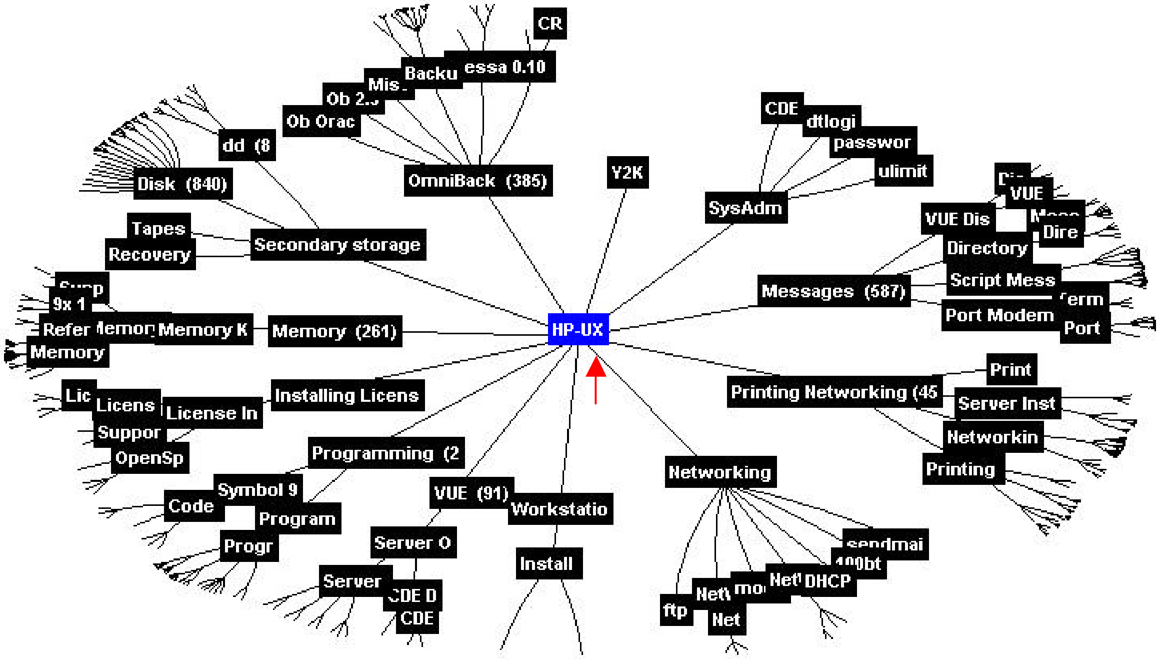


3. NAVIGATION IN A LARGE HYPERBOLIC SPACE

The following example illustrates the navigation paths in a large hot topic hierarchy for web document navigation. The hot topic hierarchy consists of a tree structure to represent the parent-child relationships plus some cross links and cycles. The goal of the navigation is to find a solution to the problem of “why can the user not send mail on his HP-UX system? “

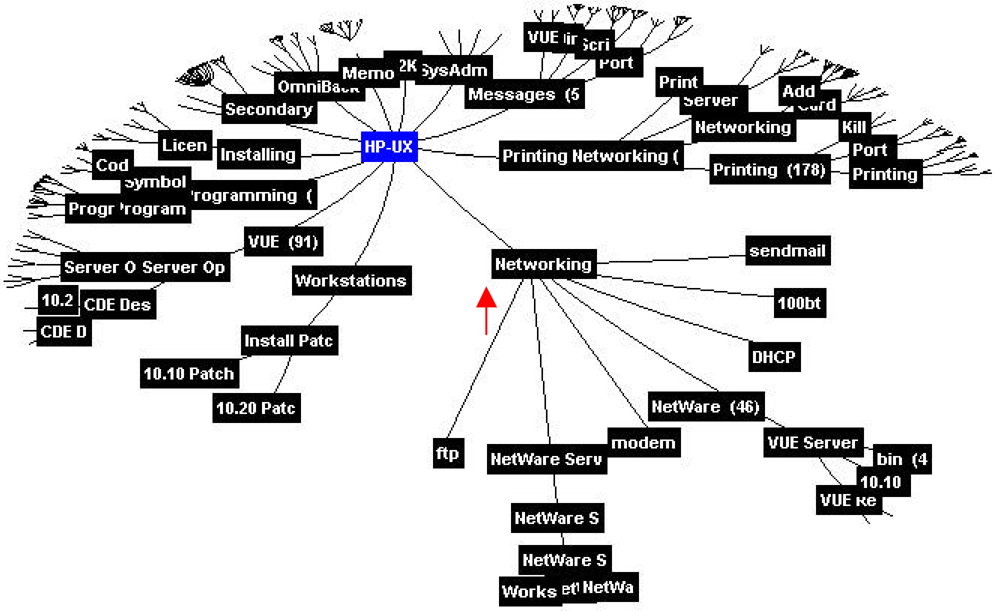
3.1 Step 1:

Navigation starts from the root "HP-UX" node and moves toward "Networking" node
An arrow is used to trace the current navigation path.



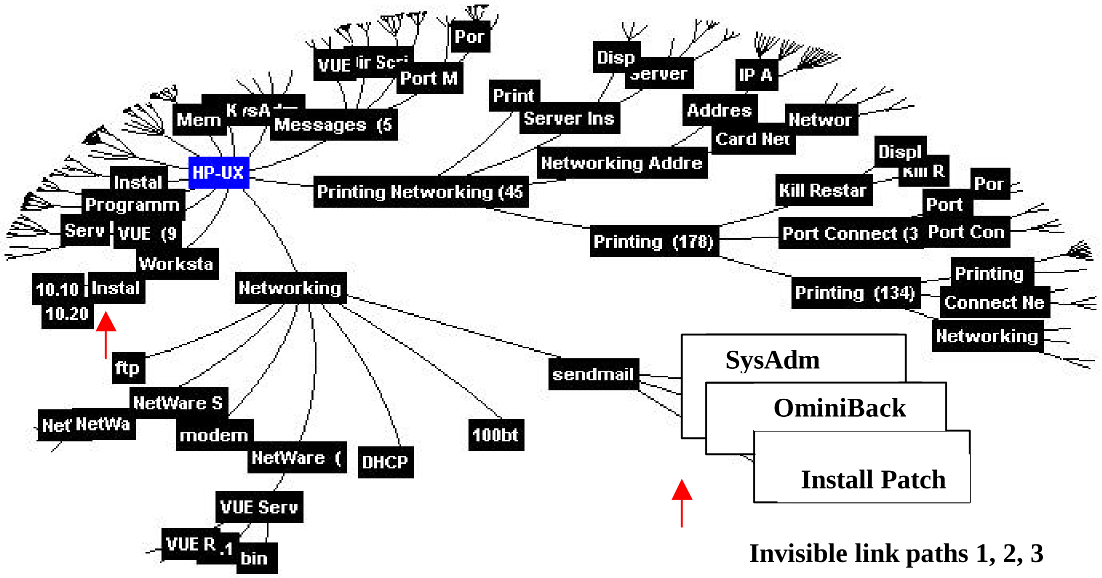
3.2 Step 2:

From "networking" node, find the path to the "sendmail" node.



3.3 Step 3:

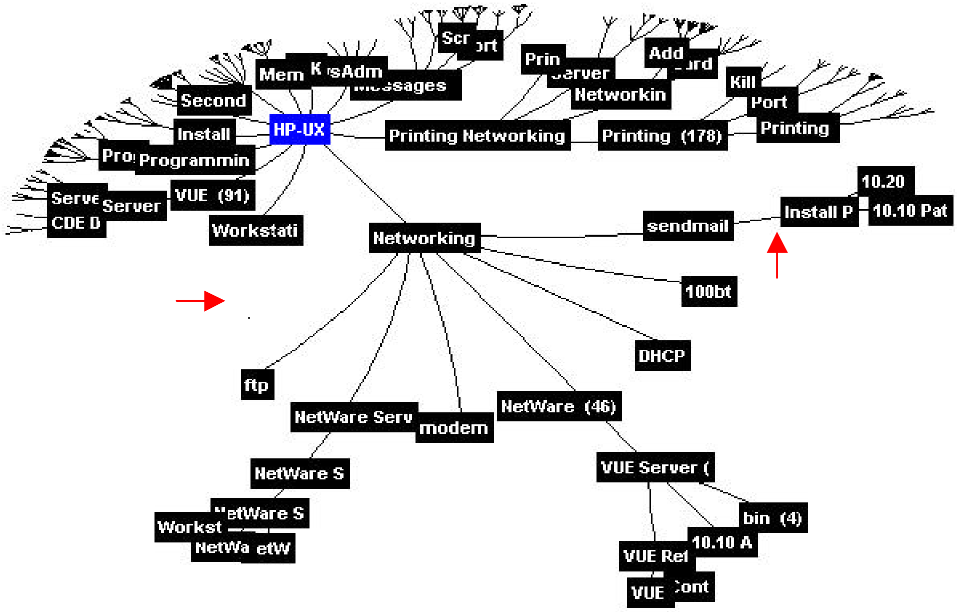
When the user clicks on the "sendmail" node, the user will be notified by the "sendmail" node to select one of the invisible link paths; the user selects the "Install Patch" node.



Invisible link paths 1, 2, 3

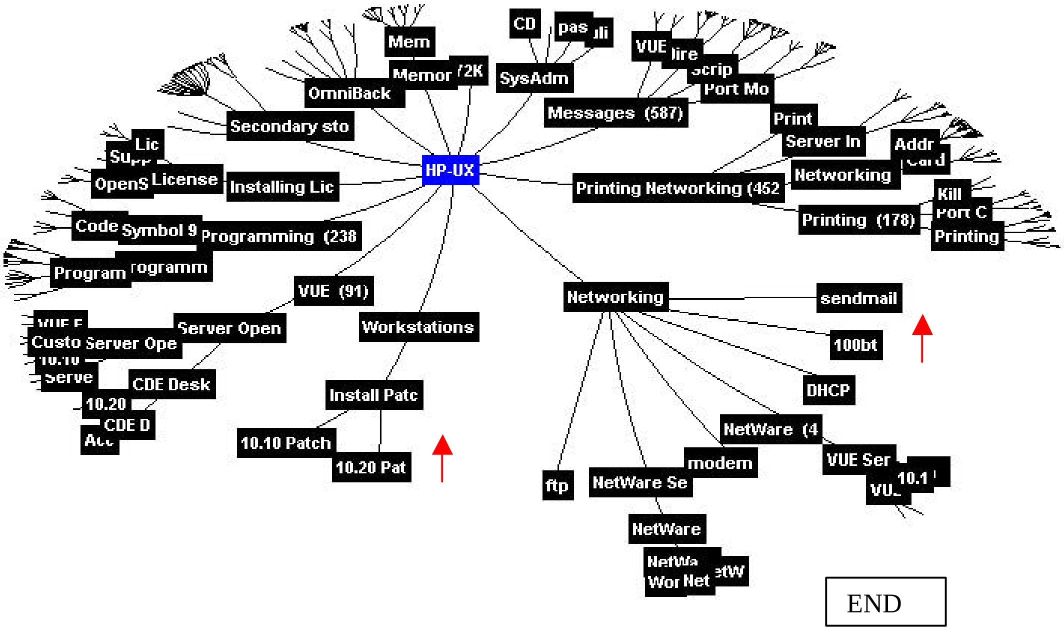
3.4 Step 4:

After the user selects an invisible link path – "Install Patch", the "Install Patch" node and its children nodes (Patch Install, 10.10 Patch, 10.20 Patch) will be mapped from the original structure (Workstation) in the hyperbolic space to the "sendmail" node. A placeholder is placed on the node's original structure (Workstation)



3.5 Step 5:

After navigating on “Patch Install”, “10.10 patch”, and “10.20 path” nodes, the user discovers that missing the " 10.20 patch” causes his “send mail” problem. The user ends his navigation session by clicking on the “end” button. At this time, the three nodes linked by the invisible link will be moved from their new structure (at “sendmail” node) back to their original structure (at “Install Patch” node).



4. APPLICATIONS

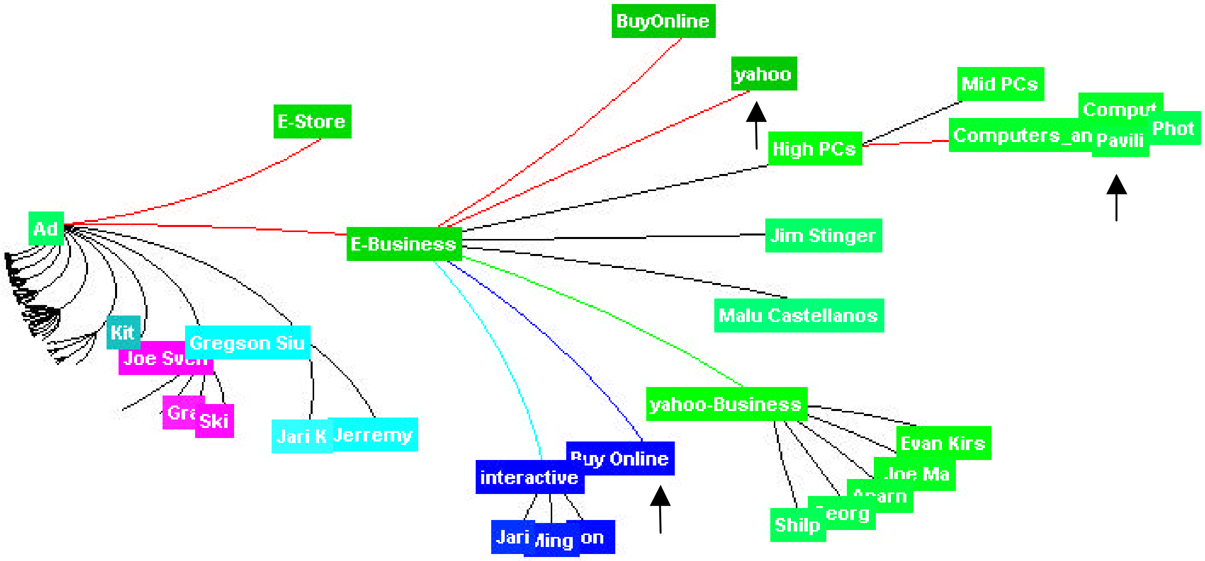
The invisible link technique together with the link table and link history logs in a hyperbolic space is built on a Java-based server-client model. To achieve rapid display, the visualization construction and rendering are done locally in the client site. The data mining and computation execute on the server. The server is integrated with the data mining engine and data warehouse. We have applied this invisible link technique to three data mining applications: Web e-business URL path navigation, customer call center document navigation, and Web site index creation.

4.1 Web E-Business URL Path Navigation

One of the common problems electronic store managers want to solve is how to target advertisements and increase product sales. At HP Laboratories, we constructed a large highly interconnected graph with cycles in a hyperbolic space to visualize the pages that a customer clicking through on the electronic store’s web site. The nodes are web pages and the edges are URL links. We incorporated the invisible links technique to navigate nodes from other branches of the hyperbolic space. We use information from the link table and the link history log to track and visualize cross-links and cycles.

Figure 4 illustrates a web URL path navigation for analyzing the effectiveness of advertisement. For example, the store manager can easily follow the paths from Yahoo, BuyOnline, and other referring sites to find out how many customers visited the ads on the referring sites and how many of these visits resulted in sales. To graphically represent the contents of the node, each node provides information including visiting time, access statistics. Color-coding is now used to show the sales over a period of time, e.g. the top sales are colored red.

Figure 4: Web URL Path Navigation



4.2 Customer Call Center Navigation

The second example shown is a Hewlett Packard internal web based troubleshooting application called InterView [6]. InterView was developed as an interactive web based troubleshooting system, which guides a user by a question answer dialog to the right solution of the problem. InterView's knowledge base can be best described by a huge decision graph of nodes.

The Knowledge base is organized in domains, knowledge areas that cover a similar topic, while each domain consists of several hundred interconnected nodes. Due to the fact that the knowledge base is more general than a strict decision tree, the invisible link extension of the hyperbolic space is needed to visualize the knowledge base correctly.

We developed a hyperbolic space viewer in InterView for the following purposes:

1. Navigation and overview of the domain
   The users as well as the authors are using the hyperbolic space visualization to navigate through the network of nodes to get the right solution. This allows especially experienced users to identify and find the right solution from the knowledge base quickly.

2. Authoring and maintenance of the domain
   By enabling the hyperbolic space viewer with some additional functions like copy, cut, link, the knowledge authors are able to maintain the structure of even very big domains easily.

3. Color based statistical information mapping
   To allow efficient content and knowledge management, the InterView application collects statistical data, like access, knowledge success/failure counts etc.

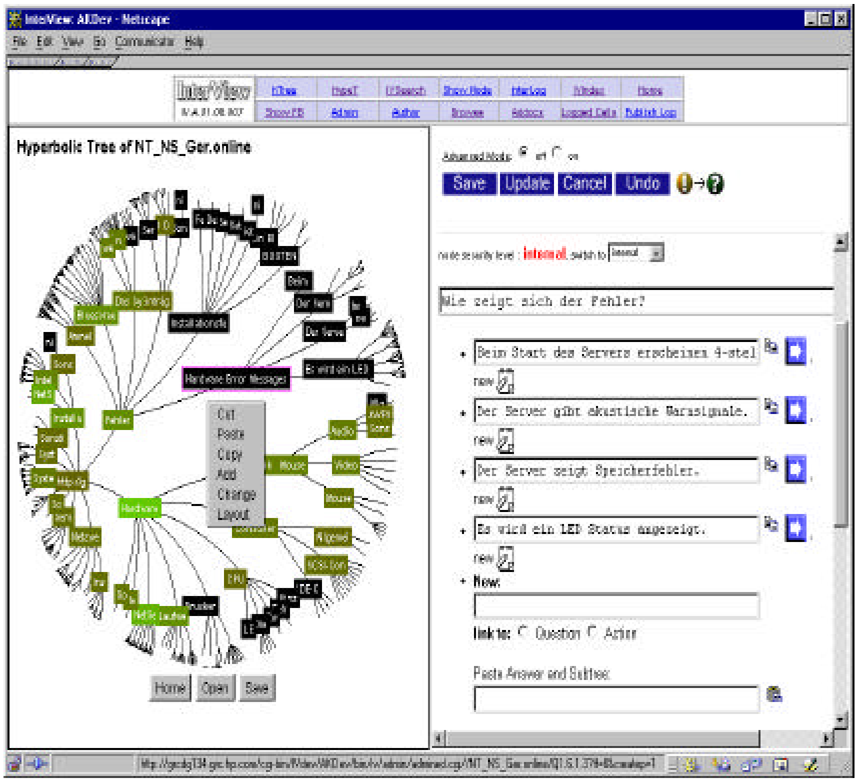
To allow authors an easy way to interpret this statistical data in the context of the knowledge itself, we are using colored nodes in the hyperbolic space to represent the data; e.g. the more often a node was used or accessed, the lighter the color is applied.

This representation allows authors easily to identify areas in the knowledge base where content maintenance or changes have to be made.

Without the invisible link technique the hyperbolic space viewer could only be used in a very limited way to represent InterView's knowledge base. The invisible links allow the users as well as the authors to focus fully on the hierarchy of the content and not on the complexity of the network.

Due to the fact that the developed Java based hyperbolic tree viewer allows switching between visible and invisible links, it is still possible for experienced authors to review and change the complete structure of the knowledge decision graph, while novice users prefer the strict hierarchical view.

Figure 5: An InterView Application Sample


InterView: AIDev - Netscape
File Edit View Go Communicator Help
InterView
Hyperbolic Tree of NT_NS_Ger.online
Hardware Error Messages
Cut
Paste
Copy
Add
Change
Layout
Home
Open
Save
Advanced Mode
Save
Update
Cancel
Undo
node security level : internal switch to
new
New:
link to: ○ Question ○ Action
Paste Answer and Subtree:

Wie zeigt sich der Fehler?

- Beim Start des Servers erscheinen 4-stel
- Der Server gibt akustische Warnsignale.
- Der Server zeigt Speicherfehler.
- Es wird ein LED Status angezeigt.

4.3 Web Site Index Creation

Some existing sites on the Internet provide an index, which describes how the web site is structured. Very often this index already has a hierarchical structure and represents directly the page hierarchy of the site. This hierarchical index structure provides a good help for experienced as well as novice users to navigate quickly to the right content page of the web site. Unfortunately such an index is not available for every existing web site. Creating such an index automatically based on a given URL would enable the user to browse quickly the information structure of a web site, even though he is completely unfamiliar with this site. In general, web sites can contain several hundreds of different web pages. A presentation of such an index as a hyperbolic tree enables the user to focus on a special subtopic of this hierarchy as well as navigate quickly through the index.

One potential problem to solve is that general web sites very often do not consist of strict hierarchical structures. This means that several web pages are referenced from different places on the web site itself (e.g. the index web page itself might be referenced from nearly every page's navigation bar.)

In these cases the proposed invisible link technology becomes very helpful. The user can still focus on the hierarchical structure to navigate the web site index but does not miss any cross-links between the different web pages.

We developed a prototype of a service that allows users to create automatically a web site index, which is visualized as a hyperbolic tree for navigational purposes.

The user provides an initial URL as well as a maximal depth to scan the given web sites URL. Based on the given information a web robot scans the given web site and derives the hyperbolic tree index hierarchy from the given web site's hierarchy. The depth of the hierarchical index is determined based on the given depth value. In the hyperbolic tree visualization each node represents one web page. The label of the node is derived based on several rules and heuristics from the title, headers or images names of the original HTML source of the pages. Clicking on the node in the hyperbolic tree hierarchical visualization opens the original web page in the web browser of the user.

In our experiments we discovered that this visualization method allows users easily to navigate and discover the structure of new or unknown web sites. Even experienced users very often prefer to navigate a web site using the hyperbolic tree index because it allows quickly moving or switching between different pages of the web site.

Currently we have plans to establish an internal service to allow other HP internal web sites to use this system to create their hyperbolic tree based indexes automatically. This approach is especially beneficial for web sites with frequently changing content or dynamically built structures.

5. CONCLUSION

The hyperbolic space with invisible links enables a user to easily navigate through different links without being overwhelmed with a large number of nodes and paths. Our technique could be extended to a 3D graph hyperbolic space. In addition, the invisible link technique could be applied to the area of e-commerce, e-services, such as product associations, customer Internet Shopping behavior analysis, and a web site indexing service.

ACKNOWLEDGEMENT

Thanks to Sharon Beach, Craig M. Wittenbrink, and Martin Griss from HP Research Laboratories for their encouragement and suggestions; to Graham Pollock from Agilent Laboratories for review and comments. Also thanks to Jeff Holmbeck, Harold Shinsato from "Inxight Software, Inc." for their technical support and for allowing us to use their toolkits in our experiments.

REFERENCES

1. Tamara Munster, "Exploring Large Graphs in 3D Hyperbolic Space" IEEE Computer Graphics. Vol. 18, Number 4. 1998

2. Emmanuel Frecon, Gareth Smith, "WebPath – A Three Dimensional Web History", InfoVis 98, Oct. 1998

3. The Hyperbolic Tree Toolkit is a product from Inxight Software for exploiting large amounts of information. 1998

5. John Lamping and Ramana Rao, and Peter Pirolli. "A focus + context technique based on hyperbolic geometry for visualizing large hierarchies". ACM SIGCHI Conference. ACM, May 1995.

6. George G. Roberson, Jack D. Mackinlay, and Stuart K. Card, “Cone Trees: Animated 3D Visualization of Hierarchical Information”. 1991, ACM

7. Adrian Krug. “InterView: Knowledge Content & Program Architecture”, Hewlett Packard SW-Support Delivery Engineering, 1998

8. Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. “Information Visualization Using Vision to Thank”, Morgan Kaufmann Publisher, Inc. 1999

9 Ming C. Hao, Meichun Hsu, Umesh Dayal, Adrian Krug, “ Visual Mining Large Web-based Hyperbolic Space Using Hidden Links “, The Third International Conference on The Practical Application of Knowledge Discovery and Data Mining, PADD’99, April.