An Initial Examination of Ease of Use for 2D and 3D Information Visualizations of Web Content

Kirsten Risden[1], Mary P. Czerwinski[1], Tamara Munzner[2] and Daniel B. Cook[1]

[1] Microsoft Corporation

[2] Stanford Computer Science Department
Correspondance and requests for reprints should be sent to Mary P. Czerwinski, Microsoft Research, One Microsoft Way, Redmond, WA 98052, or marycz@microsoft.com.

Abstract

We present a discussion and initial empirical investigation of user interface designs for a set of three Web browsers. The target end user population we identified were experienced software engineers who maintained large web sites or portals. The user study demonstrated the strengths and weaknesses of two conventional 2D browsers for this target user, as well as that of XML3D, a novel browser that integrates an interactive 3D hyperbolic graph view with a more traditional 2D list view of the data. A standard collapse/expand tree browser and a Web-based hierarchical categorization similar to Yahoo!, were competitively evaluated against XML3D. No reliable difference between the two 2D browsers was observed. However, the results showed clear differences between XML3D and the 2D user interfaces combined. With XML3D, participants performed search tasks within existing categories reliably faster with no decline in the quality of their responses. It was informally observed that integrating the ability to view the overall structure of the information space with the ability to easily assess local and global relationships was key to successful search performance. XML3D was the only tool of the three that efficiently showed the overall structure within one visualization. The XML3D browser accomplished this by combining a 3D graph layout view as well as an accompanying 2D list view. Users did opt to use the 2D user interface components of XML3D during new category search tasks, and the XML3D performance advantage was no longer obtained in those conditions. In addition, there were no reliable differences in overall user satisfaction across the three user interface designs. Since we observed subjects using the XML3D features differently depending on the kind of search task, future studies should explore optimal ways of integrating the use of novel focus + context visualizations and 2D lists for effective information retrieval. The contribution of this paper is that it includes empirical data to demonstrate where novel focus + context views might benefit experienced users over and above more conventional user interface techniques, in addition to where design improvements are warranted.

1.0 Introduction

Web browsers and search engines provide users with a narrow keyhole onto the vast content of the World Wide Web. Most interfaces for accessing Web content employ minimally interactive two-dimensional techniques. There is promise in developing more effective browsers that incorporate richer three-dimensional interaction and visualization capabilities. Such browsers are becoming feasible for consumers because of the trend towards inexpensive high-performance graphics capabilities. In this paper, we compare a novel interactive browser with a 3D hyperbolic graph view to two legacy 2D browsers, and provide initial experimental evidence that the interactive 3D visualization of large amounts of information is indeed useful for certain tasks. However, our evidence also shows that users still prefer to use more familiar, 2D tools under certain task demands.

We shall focus on browser designs that were created to help organize, make sense of, and manage large collections of Web pages, files, or documents available on the Web. We are particularly interested in collections that are not strictly hierarchical in structure, in which categories may have one or multiple parents. We explore the domain of Web site content generation and maintenance, tasks that our target user population deals with on a daily basis, often with non-hierarchical data sets. In the empirical study reported in this paper, we analyze the usability of three different browsers in detail for this genre of tasks and data.

In this paper, we shall first provide key background and motivation for the design and development of XML3D, focusing on how this novel browser integrates an interactive view of the link structure of a site in 3D hyperbolic space with 2D list components. In addition, we will briefly cover the literature on graph-based visualizations. However, more detailed discussions of an earlier version of XML3D have been previously published (Munzner, 1997, 1998a, 1998b), and a review of graph-based information visualization is not the goal of the current paper. Since our goal is to communicate the initial ease of use findings from comparing XML3D to alternative browsers, we describe in detail the user interface of the XML3D as it applies to the tasks of Web search, content generation and maintenance. Then, we demonstrate the relative strengths and weaknesses of XML3D when compared with existing Web visualizations. We present a user study that compares the XML3D Web visualization to two alternative 2D visualizations of Web content: the traditional collapsible outline tree browser (e.g., the Windows

Explorer-like user interface) and the Web site http://www.snap.com. The Snap Web site is an edited, hierarchical structure navigated from an overview page to more detailed but less comprehensive pages, much like Yahoo! and many other directory search sites.

The contributions of this paper include a discussion of how to effectively integrate a 3D hyperbolic graph view with other 2D components based on empirical observation, and a detailed evaluation of the effectiveness of 3D hyperbolic layouts accompanied by 2D lists, particularly when compared to existing Web solutions such as collapsible tree browsers or standard Web portal site hierarchical layouts.

1.1 Motivation for 3D Hyperbolic Visualization

Although visualizing a users' recent activity on the Web has become a recurring theme in the information visualization literature (Card et al., 1999), the utility of showing end users the detailed hyperlink structure of their history list or search results is not yet proven. It is possible that some kind of higher-level grouping based on semantic analysis would be more useful for end users than the raw hyperlink structure typically provided today (Dumais and Chen, 2000). This is probably why "edited" portal sites such as Yahoo (http://www.yahoo.com) and Snap (http://www.snap.com) have gained such popularity in usage (Media Metrix, Nov. '98). However, there are still end users who are looking for more "expert" visualizations that support their tasks. Webmasters and content creators have a more compelling need to understand the detailed hyperlink structure of their site, as do Web developers. Thus, Web visualization for the special tasks involved with search, content creation and maintenance is the driving theme throughout this paper.

The World Wide Web can be considered as an instance of an abstract graph where nodes correspond to pages and edges correspond to hyperlinks. The Web is an interesting problem domain because although it is highly interconnected, the designer of a Web site often has a clear notion of which of the many incoming links to a page is the "best" parent for that page, with the other parents having secondary importance. This quasi-hierarchical characteristic of the data can be used to construct a spanning tree, which is generally what is shown in strictly hierarchical user interfaces like the collapsible tree browser used in the Internet Explorer history display. The fact that an intelligently constructed spanning tree gives some insight into the structure of a site is likely the reason that hierarchical interfaces are widely used---despite the fact that it would theoretically be quite ineffective if the Web were viewed as a completely general graph. Another reason for the popularity of strictly hierarchical interfaces is that tree layout is a much more tractable computational problem than general graph layout.

The H3 graph layout algorithm is a precursor to the algorithm used in XML3D that is computationally inexpensive. The H3 graph layout algorithm (Munzner, 1997) is a relatively recent technique that uses spanning trees as the layout backbone but also provides a way to show the non-tree links. The H3 layout is fast enough that it is not a computational bottleneck, and the H3Viewer drawing algorithm (Munzner, 1998b) provides guaranteed frame rate interaction with the structure. One of the distinctive features of the H3/H3Viewer systems is the use of hyperbolic space for both layout and navigation.

Methods of introducing deliberate distortion in order to show a large amount of contextual information in a given amount of screen area are collectively called "Focus+Context" (Card 1999, Chapter 4) or "Nonlinear Magnification" (Keahey & Robertson, 1997). The specific merits of using hyperbolic geometry as the distortion technique are discussed at length in several previous papers (Lamping et al, 1995; Munzner & Burchard, 1995; Munzner, 1997), and do not need to be repeated here. In brief, *hyperbolic space* is one of the non-Euclidean geometries where circumference and area increase exponentially instead of geometrically. Although hyperbolic space is infinite, we can project it into a finite ball of Euclidean space for a Focus+Context view.

We see an example of a hierarchical graph drawn in 3D hyperbolic space as the 3D graph to the left in Figure 1 below. When we lay out and move large graphs using hyperbolic distances, we can see details in a neighborhood around a node of current interest while retaining an overview of the larger structure. Although all the nodes are the same size in hyperbolic space, in the projection nodes that are far from the origin appear to be smaller than those at the undistorted center of the ball. The surface of the ball in Euclidean space corresponds to infinity in hyperbolic space, so it acts like a "vanishing surface" instead of the more familiar vanishing points of traditional perspective views. Although distant features are quite distorted, we see far more surrounding context than we could with a standard Euclidean 3D perspective projection. This is particularly important if and when we want to show the destinations of distal links, which may be quite far away from the originating node. In addition, the layout allows the user to manage the distortion somewhat so that the relative position of nodes (e.g., “left of”) is maintained. Not all hyperbolic browsers have this feature.

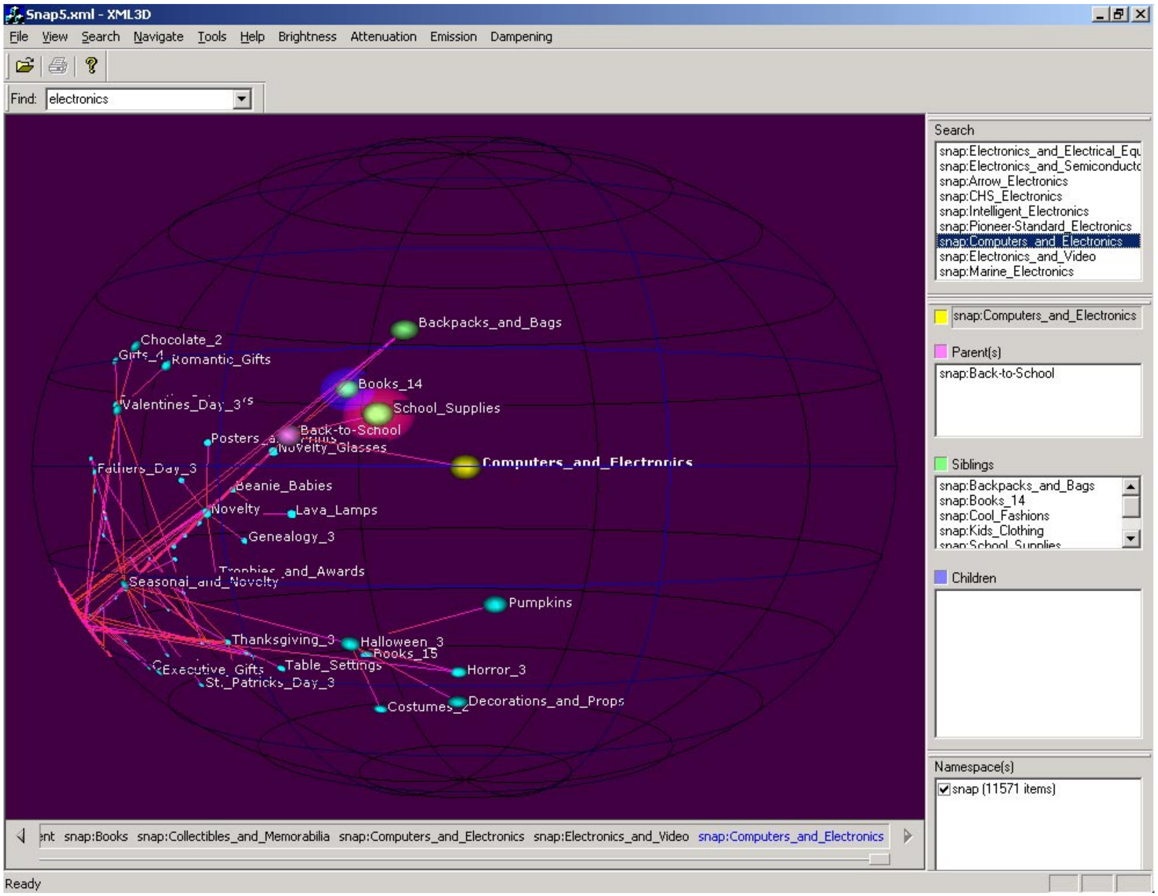

Figure 1. A screen shot of XML3D, including the hyperbolic graph.

1.2 XML3D

One of the key features of XML3D is its ability to handle multiple-inheritance hierarchies. Since XML has natively a single-inheritance structure, a method was needed for describing the extra parent-child relationships that a node is involved with other than the one explicitly implied by XML tree itself. This was handled by adding attributes to certain XML tags: either a "name" or a "ref" attribute to define either end of a directed link between two nodes. Put simply, XML3D reads in “multiple-inheritance” hierarchical relationships represented in schema-like XML text files and shows them graphically. The interface consists of two linked components, the H3Viewer described above and several 2D lists (of parents, children, and siblings) which dynamically update based on the selected graph node. We started from the premise that drawing graphs as nodes connected by edges is a compelling visual metaphor. The idea that the H3Viewer is most powerful when effectively integrated with other interface components was asserted without proof in previous work (Munzner, 1998a). XML3D was built to explicitly test this assertion in a particular task domain (that of Web content developers and maintainers, our target audience). Figure 2 shows an example of the XML3D application.

When an XML file is first loaded into XML3D, the 'root' node occupies the focus point at the center of the ball. The user navigates from one node to another directly via mouse selection, or via keyboard equivalents. Selecting a new node triggers an animated transition so that the node moves to the focus point at the center of the ball. The motion is a combination of hyperbolic translation and a Euclidean rotational component so that the parents of a node are on the left and the descendants are on the right. Explicit lists of parents, children, and siblings appear to the right of the graph in a 2D area of the screen known as the InfoBar. Items in the InfoBar are 'hot'; clicking on one of them causes the node representing that item in the graph to move to the center of the ball. Nodes in the graph, such as the currently selected node, its parents, sibling and children, are differentiated by color.

Without changing the selected node, the user can alter the perspective on the graph in six degrees of freedom: translation along the x, y, and z-axes, and rotation about those axes. A standard mouse supports these movements; we have seen in previous design work in 3D that it was important to keep the main input device interaction familiar, even if new functionality is available with non-standard device operations (Robertson et al., 1998). It remains to be seen via user study if users will learn to utilize the six degrees of freedom in their mouse usage. More precise movements are possible through keyboard shortcuts for more experienced users.

Browsing a graph implies that a history list can be maintained. XML3D displays the path of nodes traversed by the user in getting to the currently selected node. (A path is defined as a set of nodes logically connected by subtype arcs). This is supported by the History list at the bottom of XML3D's window (see Figure 1, bottom). The history list merely scrolls from left to right as the user traverses nodes, without showing any relationship information. The selected node's label typically occupies the right-most position in the linear list control. Clicking on a label in the history list causes the 3D graph to be re-centered on the appropriate node, and also scrolls the relevant label to the far right of the history list. Previously visited nodes are not erased from the list until the user selects an "unrelated" node (a node not logically connected by subtype arcs). If the user selects a node that is unrelated to the current home node, the old, unrelated path information is discarded. This is a way of starting a new search while browsing the graph, and emptying the history list to display only nodes from this point forward in time. The user may manually discard the path information from the View menu at any time. The 'back' and 'forward' buttons on the history list are analogous to the Back and Forward operations in a Web browser: the graph is re-centered on a previously visited node for each click of these buttons. The history list also includes a slider bar, akin to a horizontal Windows scrollbar, which allows the user to scroll forward or backward through the horizontal history list.

XML3D features a Find toolbar, normally located just below the menu bar. All items that match the search text are changed to the 'selected' node color. Navigation between members of the search results set is supported in the Search/Find Next and Find Previous menu items, as well as via keyboard shortcuts or selection via mouse. The results of a search also appear in the Search Results toolbar (see Figure 2). Items in the Search Results toolbar are selectable as well.

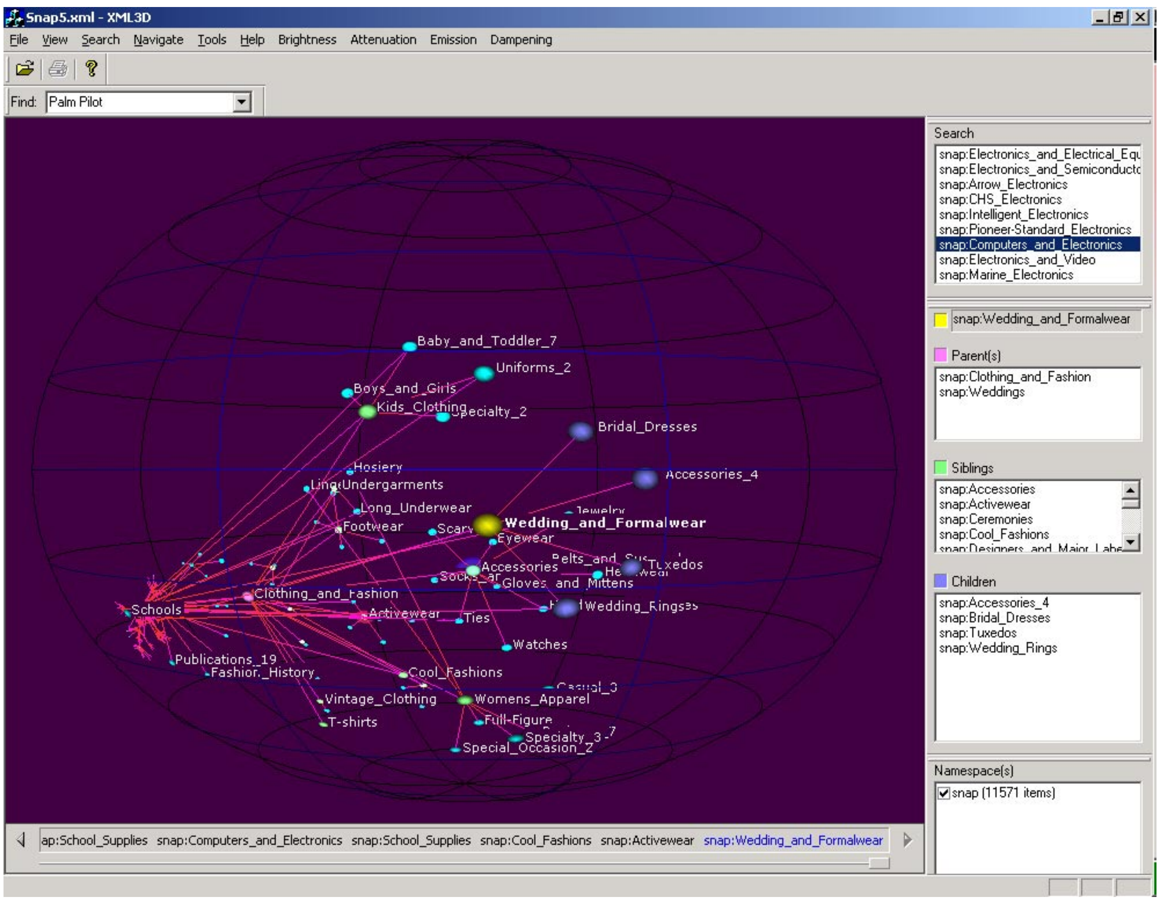

Figure 2. A screen shot of XML3D, with the Wedding_and_Formalwear node selected by the user.

1.3 Previous Empirical Evaluations

There have been a few published empirical studies of 2D versus 3D information visualizations, but results are mixed and far more work in this area is needed. Wickens (1994a, 1994b, 1995) and his colleagues have done the most work in this domain, largely focusing on aeronautical applications, but also including graphs, maps and other displays. For instance, Wickens, Liang, Prevett and Olmos (1994a) report on two experiments that contrast egocentric vs. exocentric features of 3D or 2D electronic map displays for both local and global tasks. Pilots used these displays for a simulated landing along a curved approach, through a terrain-filled region. In Experiment 1, a rotating vs. fixed-map display was experimentally crossed with a 2D vs. 3D view. Rotating displays supported better performance overall, and 3D displays provided a slight advantage for lateral guidance but a substantial cost for vertical control, because of the ambiguity with which perspective viewing depicted precise altitude. In Experiment 2, pilots flew with the rotating 2D display, and with an improved version of the rotating 3D display, using color coding to reduce the ambiguity of altitude information. Pilots' vertical control improved as a result of the 3D display improvements. However, a map reconstruction task revealed marginally better performance for terrain awareness with the 2D map. The authors conclude that there are costs and benefits of presenting information in 3D, ego-referenced format for both local guidance and global awareness tasks.

In a separate study, Wickens, Miller and Tham (1994b) had air traffic control subjects perform a simulation in which they were required to evaluate pilot flight plan change requests. Some requests were reasonable and safe, whereas others would cause a mid-air conflict with another plane. Subjects received the requests either in a conventional 2D display or a 3D perspective display. Requests were issued either as voice messages or displayed visually on the screen. The results indicated that performance was generally equivalent between the 2D and the 3D display and was best with the voice message transmission format. So for this task, no performance benefits for 3D visualizations were demonstrated.

Wickens, LaClair and Sarno (1995) compared standard 2D data graphs with 3D graphs and color graphs for presenting 3-dimensional data. The data was formatted visualizing the effects on Y of 4 levels of X and 4 levels of Z. Z was represented by a line type in the 2D display, and by space (depth) in the 3D display, and Y was represented by color in the color graphs. Subjects answered questions about the graphed data that varied in the degree to which they required focused attention from a single data point to having to perform an an integration across the entire data set in the graphs. The results indicated that subjects' performance was slowest and least accurate for the 3D graph for the focused attention questions, but this performance disadvantage lessened as questions became more integrative. Performance with the color display suffered badly in both speed and accuracy with the integrative questions. The 2D display performed well in both speed and accuracy across all question types. Again, it was demonstrated that there are benefits and costs associated with presenting visual information in 3D, depending on the users' tasks.

As alluded to earlier, there are many examples of graph-based visualizations, both in the information visualization literature and in the hypertext literature (e.g., Anderson, 1995; Crampes, 1997; Furnas & Zacks, 1994; Mackinlay, Rao & Car, 1995; Noik, 1993; Poulovassilis & Levene, 1994; Salminen, Tague-Sutcliffe, & McClellan, 1995). Unfortunately, these graph-based visualizations have rarely been subjected to empirical evaluation in the reports. For this reason we felt strongly that an important contribution to the field would be to evaluate XML3D's design strengths and weaknesses in real world tasks that would be performed by our target end users.

Figure 3. The http://www.snap.com homepage.

1.4 User Study Overview

We wanted to empirically compare the usability of XML3D to available 2D interfaces, such as the collapsible tree browser, as well as a traditional user interface for a portal Web search engine. We had access to the 12,000 node, Snap.com hierarchy, and ported its contents into XML3D and a 2D collapsible tree browser. We used the live Snap.com hierarchy, (http://www.snap.com), as the second 2D alternative during the study. We ran the study over an internal T1 line (a very fast connection) so that performance across the browsers was comparably fast (no users commented

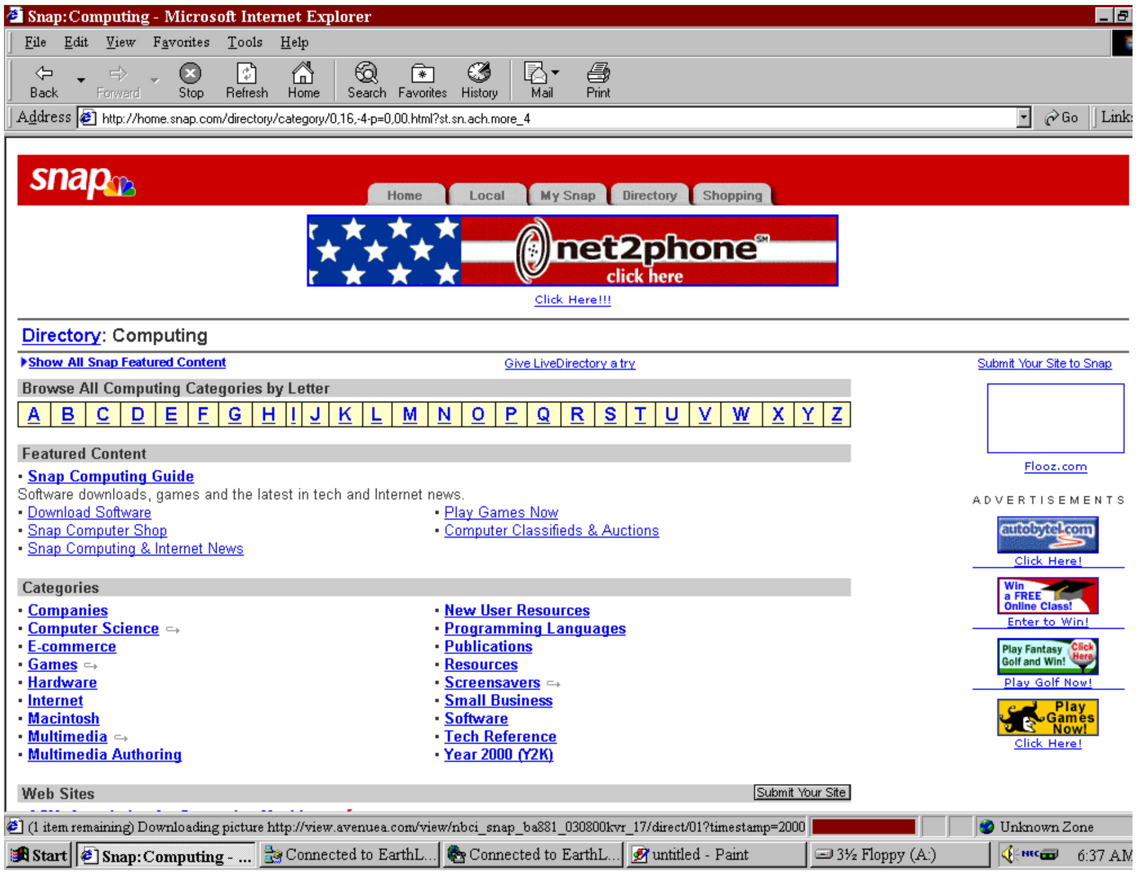

Figure 4. User browsed from the Computing directory from the snap homepage.

The three browsers differed with regard to how they provided access to content and the extent to which they revealed the structure of that content in relation to the directory as a whole. This is important because it may have an impact on the results obtained in the study. Both XML3D and www.snap.com "bubble-up" content from disparate places in the hierarchy. XML3D does this by displaying links to the parents, siblings and children of selected nodes and by displaying links to nodes that have labels that match search queries (see Figure 2). XML3D reveals semantic relationships by explicitly listing parents, children and siblings in both a list view and a graph view. The website www.snap.com (see Figure 3) reveals content from disparate places in the hierarchy by using keyword matching of search queries and reveals semantic structure by making a list of the user's current location with each of the parents up to the top node. Children are the links listed on the page at www.snap.com. Children that have an arrow beside them are cases of multiple inheritances in which the child "lives" with another parent. Siblings and other parents of a selected node are not displayed at www.snap.com. Figure 4 shows an example of searching on the word, "Computers", and Figure 5 shows what happens if the user searches on "Computers" and then chooses the "Computer Science" category. The snap website shows the path, the children and via the gray arrows shows those children that have multiple inheritance.

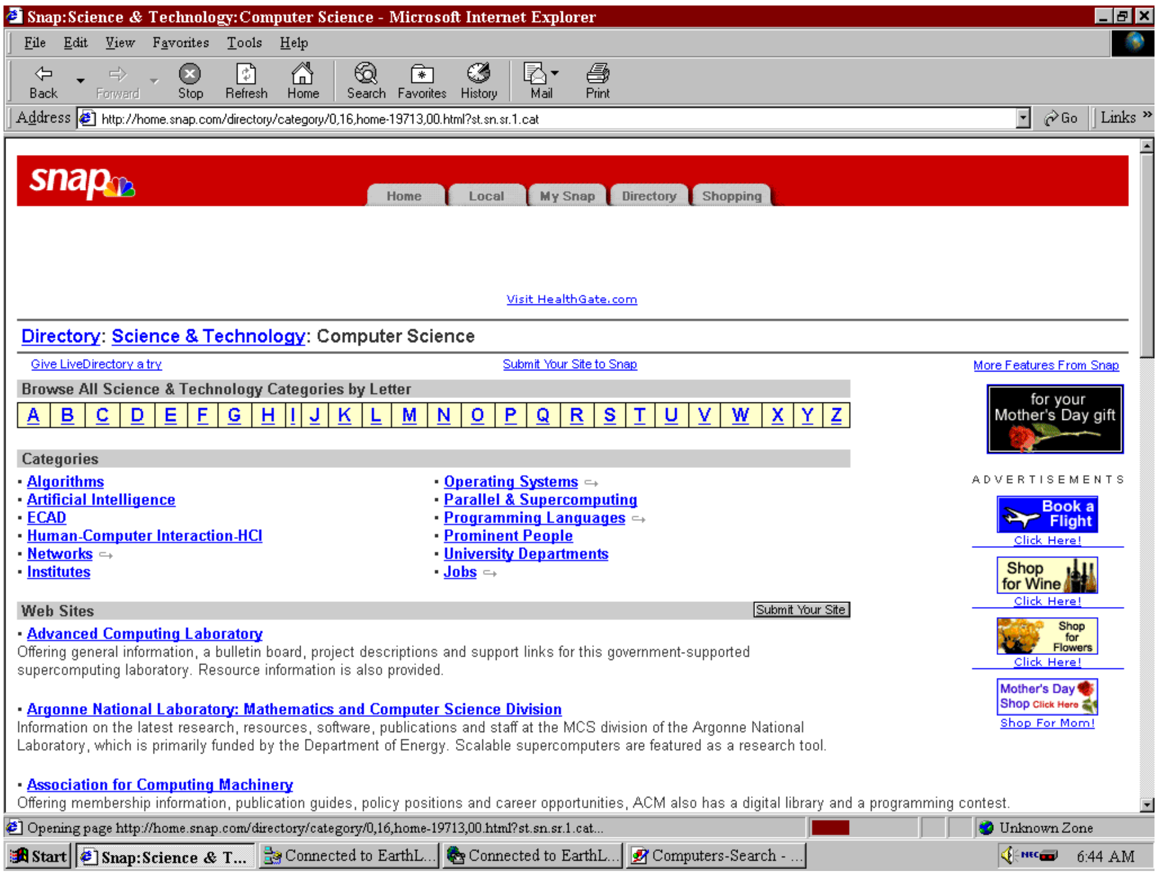

Figure 5. User searched on Computers and then clicked the Computer Science category.

In the 2D collapsible tree browser, the "bubbling up" of related content is limited to opening the tree to the first node that matches a search query. The 2D collapsible tree browser reveals parent, sibling and child relationships through indenting, but only for one parent at a time (see Figure 6 for an example). In both of the 2D user interfaces, we anticipated the possibility that either not showing all global relationships in context (tree browser) or not showing all related siblings and parents for a node (www.snap.com) would negatively affect search performance in the study.

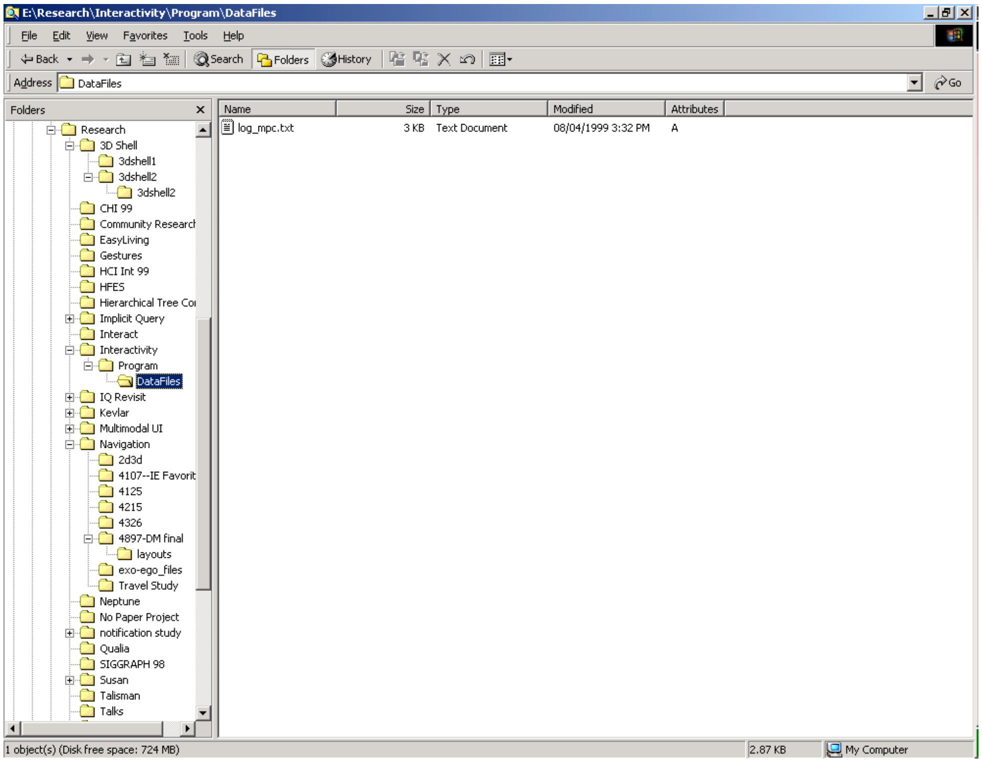

Figure 6. An example of the collapsible tree browser, Windows Explorer, with many folders opened.

Therefore, we hypothesized that the XML3D browser, which incorporates both of these features, would be a superior user interface for Web search tasks. However, due to the fact that it incorporates a novel, 3D graph visualization, it may not perform as well as predicted in comparison to more conventional, 2D user interfaces. Recent studies have demonstrated how the particular form of categorizing search results or the match of the user interface to the task demands can influence search performance in 2D as well as 3D (Chen & Dumais, 2000; Sebrechts, Vasilakis, Miller, Cugini & Laskowski, 1999). It is for all of these reasons that we felt an empirical evaluation across the three user interfaces was warranted for our task set.

As stated earlier, we chose tasks that dealt with directory management for our study, since this is what our target population of users cites as their primary task. Directory management for a Web site involves adding content to a directory scheme. Appropriate categories for the content may already exist in the scheme, in which case finding the category amounts to a targeted search. At other times, no appropriate category exists and a new one must be created and added to the scheme in order to accommodate the content. This requires comprehensive browsing. Both existing and new categories may have single or multiple parents depending on the structure of the scheme. Trying to determine where new content should be organized in these instances requires targeted search, revisitation of previously seen sites, and comprehensive browsing. Therefore these directory management tasks cover a broad range of search behaviors and allowed us to explore ease of use and usefulness of the three browsers competitively across different levels of task complexity.

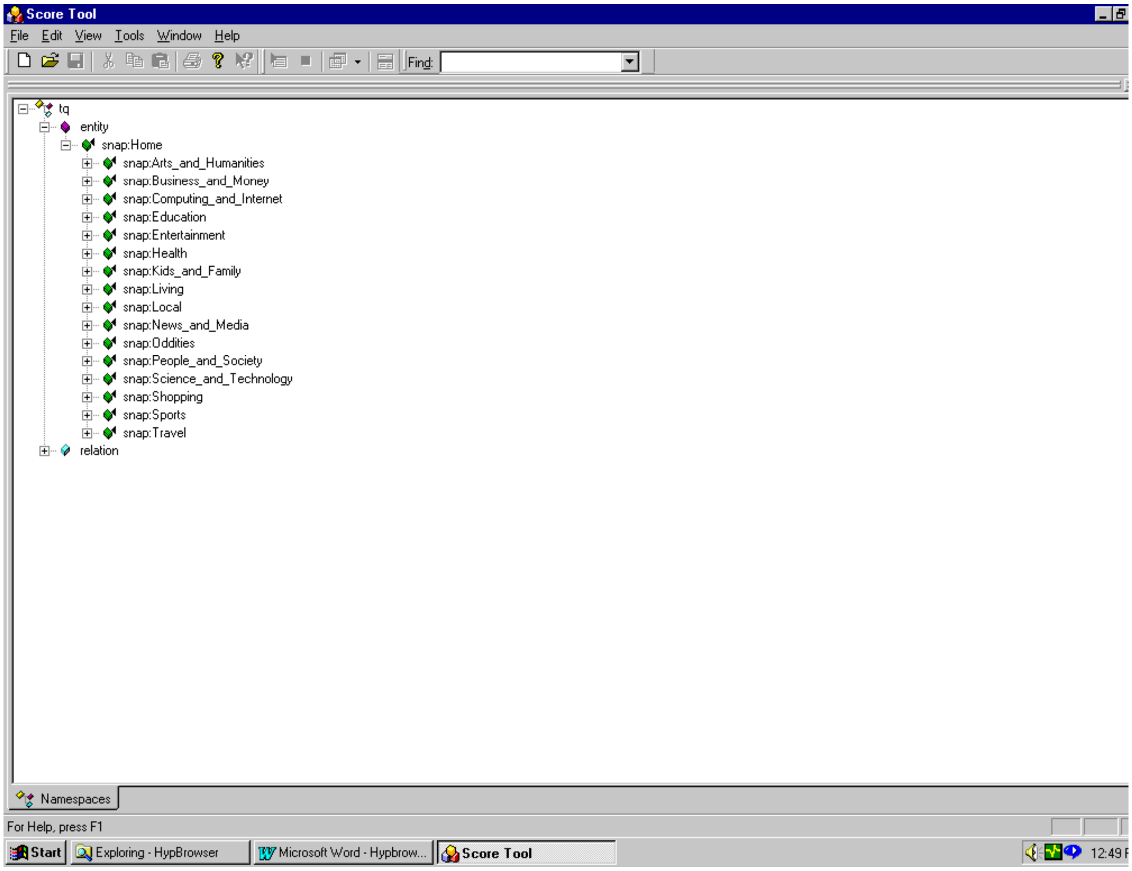

Figure 7. The collapsible tree browser used in the study.

2.0 User Study

2.1 Method

2.1.1 Participants.

Sixteen adult males from the Puget Sound region participated in this study in exchange for a software gratuity. Because the product team we were working with was targeting a technical end user, we recruited comparatively sophisticated study participants. All of the participants had worked as programmers for at least two years and were familiar with at least one of the following programming environments: Java, C++, Unix, Visual Basic, Interdev. One participant's data was excluded from the analysis because he did not understand English well enough to complete the tasks. This subject was not replaced.

2.1.2 Tasks

Three different interfaces were used to present the www.snap.com directory content. Users could either browse each user interface by interacting directly with the items on the screen, or they could type in key words in the search fields provided with each interface.

Tasks with different levels of complexity were created by varying 1) whether the subject was asked to find an existing category or add a new category to the directory scheme and 2) whether the target category and requested response involved a single parent/path or multiple parents/paths. The tasks varied in kind as well as complexity across levels. Table 1 provides an overview and examples of each type of task. Task sets were formed by creating six tasks for each task type and randomly assigning three of each type to one set and three of each type to the other set. Within sets, tasks were ordered in terms of complexity from Task Type 1 through Task Type 4 (see Table 1). Two of the tasks from Task Type 1 served as a practice task for each interface used in the session (which is why two task sets were required). Across participants, the two tasks sets were used equally often in each interface condition and their order was counterbalanced.

2.1.3 Procedure

Since we were primarily interested in the usability of the more novel XML3D user interface for our target user population, all users used the 3D Hyperbolic browser and only one of the alternative 2D browsers. Half of the participants were assigned to the 3D Hyperbolic condition first and half of the participants were assigned to one of the 2D interfaces, either the tree browser shown in Figure 8. or www.snap.com first, in order to counterbalance browser order to control for training effects. Participants were asked to imagine that, as part of their job, they managed the content directory for the www.snap.com website. They were told that this entailed determining how to fit content submitted by domain experts into the existing Snap directory scheme. They were advised that for the purposes of the study they would only be locating where to put the content and answering questions about their choices rather than taking the additional steps to actually add the content to the scheme. They were asked to work as quickly as possible while keeping the quality of the scheme in mind and were advised that we would stop them, ask them to complete whatever answer they were working on and then go on if any task time exceeded 3 minutes.

Participants were instructed to collect their own task times, as the experimenter was not blind to the browser they were using and times were not collected by the browser's software automatically. To do this they went into an observation log tool, selected the task they were about to begin from a list and clicked a button to start a timer. When they completed the task, they went back to the observation log tool and clicked the same button to stop the timer. The difference between the initial and final button clicks was logged as the task completion time. The steps for timing their tasks were provided for

	One Parent	**Multiple Parents**
Existing category	Task Type 1. You want to add content about Pope John Paul to an existing category by the same name. Find the Pope John Paul category.	Task Type 2. You want to add "buying photographic art supplies" to the existing Photography category. Find the category. Then figure out what other paths people can take to get to this content.
New category	Task Type 3. Create a new category for content about "Elementary Schools" and find a logical place to put it. Determine if there are other paths people could take to the content. Take those paths into account if necessary when coming up with the new category label.	Task Type 4. Create a new category for "Software tools for creating effective e-commerce Web sites". Find a logical place for this category. Determine at least one other path that you would create to this category. Take this into account if necessary when creating your new category.

Table 1. Descriptions of information retrieval tasks used in study.

reference on each task page. Participants were asked to finish answering the question they were working on and to move to the next task if their time on task exceeded 3 minutes, and the experimenter used a "countdown" tool to notify them when this happened.

Since participants had very little experience with the XML3D browser, a small amount of training was provided prior to the use of all three of the user interfaces to complete experimental tasks. Participants were given an overview of the user interface they would be using and completed practice tasks that involved exposure to keyboard controls before beginning any experimental tasks.

3.0 Results

A tool for managing content should allow users to do tasks quickly and with consistency (e.g., different people should arrive at the same decision). Therefore, time on task and level of subject agreement (i.e., consistency) on responses to task questions were compared across interfaces and task types. An overall Analysis of Variance (ANOVA) comparing which tool (user interface) was used, which order it was used in, whether or not it contained 2D only or 3D elements, the number of parents a search target had and whether or not it came from an existing category was carried out on the task time data. Initial analyses showed no effects of which 2D tool was used (e.g., there was no main effect for whether or not tasks were completed with www.snap.com or with the collapsible tree browser user interface, $F(1,11)=1.34$, $p>.05$). An overall ANOVA of the consistency data resulted again in there being no main effect of which 2D tool was used in comparison to XML3D. As a result, the data from the tree and Snap.com browser interfaces were pooled into a single 2D condition and the time and consistency data were submitted to a 2 (XML3D versus 2D) X 2 (existing versus new category) X 2 (single versus multiple parents) mixed analysis of variance (category type and number of parents were within-subject variables). We did this in order to increase statistical power, given our very small number of subjects. This might enable us to better examine the strengths and weaknesses of the conventional versus the novel user interface components.

3.1 Task Times

Although we used a deadline procedure of 3 minutes, task time data were trimmed by removing observations that exceeded cell grand means by three standard deviations or more in order to reduce the variance in the data. (In these cases a user had an especially hard time finding a solution relative to the other participants). In addition, there were two cases in which task times less than 10 seconds were recorded. These were considered to be recording errors and were also eliminated from the data set. Missing values accounted for 1.6% of the data.

Analysis of variance on mean task completion times revealed main effects of user interface condition (XML3D v. 2D), $F(1, 11) = 10.19$, $p < .01$, and category (new v. existing), $F(1, 11) = 37.76$, $p < .001$. Participants completed tasks faster using XML3D than using the more conventional interfaces. In addition, participants were faster to complete tasks involving an existing as opposed to a new category. The main effect of category was qualified by a significant interaction with the parent variable, $F(1, 11) = 60.84$, $p< .001$. Follow-up analyses showed that the difference between existing and new category tasks was larger when a single parent was involved, t (14) = -9.96, p < .001, than when multiple parents were involved, t (13) = -1.07, p > .05. Neither the main effect of parent nor any of the other interactions were significant. However, because we were particularly interested in understanding the kinds of tasks that benefit from XML3D, separate comparisons of XML3D and the 2D interfaces were conducted for the four task types. Participants completed existing category tasks with both one and multiple parents faster using XML3D than the 2D interfaces, t (8) = 2.87, p < .05 and t (8) = 2.34, p < .05, respectively[2]. There was no difference in task times for new category tasks with either one or multiple parents, t < 1 for both task conditions. These results are shown in Figure 9.

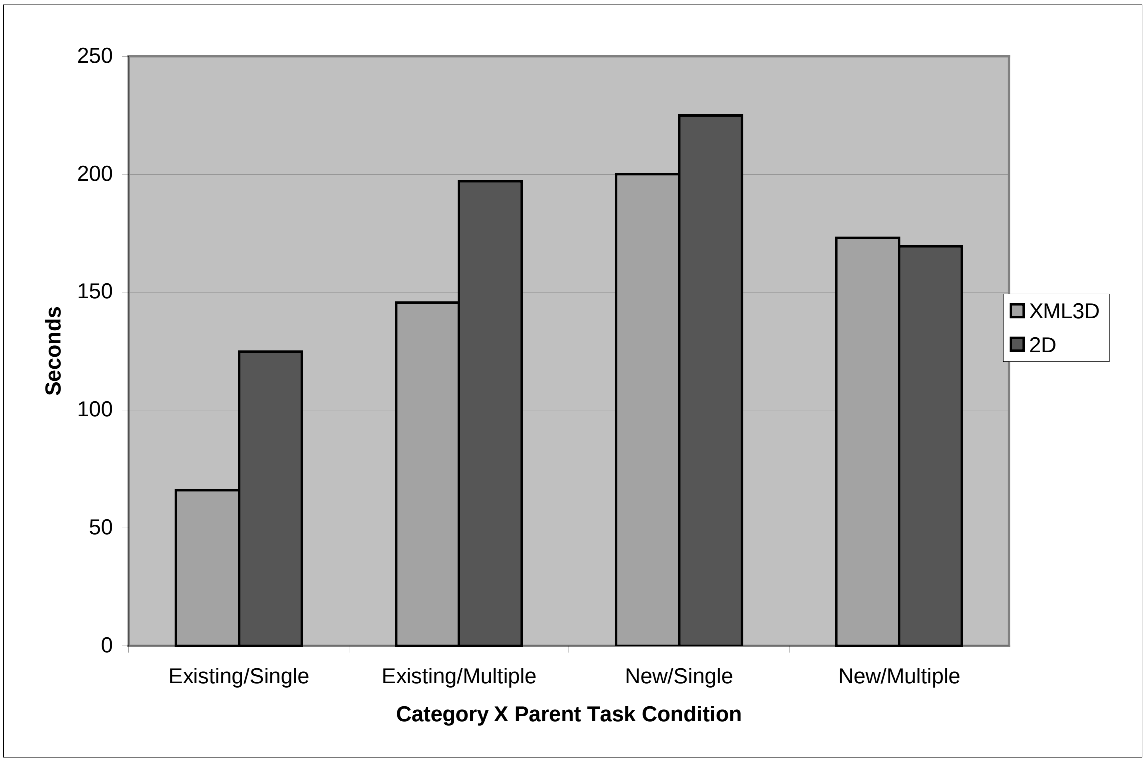

Figure 9. Average task times for XML3D v. the 2D conventional tools combined.

3.2 Consistency

The level of agreement in participants' decisions regarding placement of content in the directory was assessed. To do this we identified the most common answers to questions about category placement and category parents. The percentage of answers matching the majority response was calculated for each subject in each condition as a measure of consistency. Missing values (when subjects did not write down a search task answer) accounted for 5.3 % of the data. Analysis of variance on the percentage scores revealed no main effects or interaction effects for any of the

[2] All t tests are based on the assumption of unequal variance in this set of analyses. Use of this assumption is warranted particularly when conditions have unequal numbers of observations and the ratio of variances is more than 2 for small sample sizes.

variables. This indicates that there was no speed accuracy tradeoff associated with faster task completion times using the XML3D browser. Separate comparisons for each task were consistent with this finding. Although XML3D resulted in greater consistency in participants' answers for all but the existing category- single parent task condition, none of the comparisons reached traditional levels of significance. For the existing category – single parent task t (12) = -1.69, p > .05. For the existing category- multiple parent task t (12) = 2.02, p > .05. For the new category – single and –multiple parent tasks t < 1.

3.3 Feature Usage

Type	Frequency	Frequency	Proportion
3D graph	Find Next	167	0.113
3D graph	Find Previous	7	0.004
3D graph	Go To Child	20	0.013
3D graph	Go to Next Sibling	4	0.002
3D graph	Go to Parent	55	0.037
3D graph	Go to Previous Sibling	12	0.008
3D graph	Go to Top	14	0.009
3D graph	Node Selected	199	0.135
History	History Selection	91	0.061
2D lists	Node Selected from Child List	117	0.079
2D lists	Node Selected from Parent List	137	0.093
2D lists	Node Selected from Search Result List	326	0.221
2D lists	Node Selected from Sibling List	65	0.044
Search	Search Text Field	255	0.173
	Totals	1469	1

Table 2. Total usage frequency of XML3D elements across all experimental tasks.

The task time and consistency analyses indicate that tasks (particularly those involving existing categories) can be completed more quickly and with the same level of consistency with XML3D as that achieved using more traditional 2D interfaces. A question that arises is which aspects of XML3D are responsible for this advantage? Does use of the 3D graph, the 2D lists, or some combination of all of the XML3D features lead to enhanced performance? To determine the answer to this question the log files for the eight participants who used the 3D Hyperbolic Browser XML3D were analyzed for how frequently the graph and the list features were used across the various types of tasks, and the overall total usage results are shown in Table 2. As can be seen from the table, the 3D graph is used about 32.5% of the time during the search tasks, while the 2D list elements are used around 43.9% of the time overall. Other elements used during tasks include the search field (17.4% overall) and the history list (6% overall). It was observed that the search field was primarily used at the beginning of a search trial, as expected,

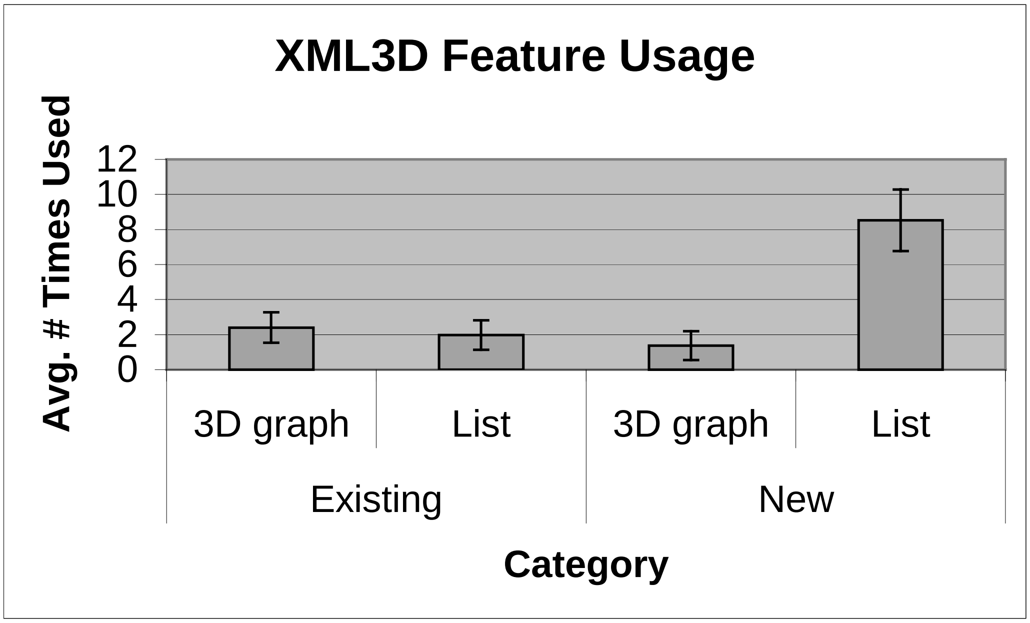

Figure 10. Average XML3D feature usage by task type.

and the history list was used only sparingly. In other words, after the initial search, participants were primarily using either the 3D graph or the 2D lists. Therefore, an isolated comparison of the average number of 3D graph elements versus the average number of 2D list elements used, for specific search task types, is provided in Figure 10.

Within the XML3D user interface, the average number of times the 3D graph versus the 2D list features were each used was analyzed in a 2 (3D graph versus list feature) X 2 (Category—new or existing) X 2 (Parents—one or more) analysis of variance with repeated measures on all variables. The analysis showed a significant main effect of Category, $F(1, 7) = 11.41$, $p < .05$, qualified by significant interactions with both parent (one or multiple), $F(1,7) = 13.27$, $p < .01$, and user interface feature (3D graph or 2D list), $F(1,7) = 19.38$, $p < .01$. The overall pattern indicated that 2D list use was greater, in general, for new category tasks than for existing category tasks, specifically in the single parent condition, $t(7) = -5.58$, $p < .001$. (The t-test for category within the multiple parent condition was less than 1, and so not significant.). Whereas there was no difference between the rate of graph and list use for existing category tasks, $t(7) < 1$, that difference was quite pronounced for new category tasks, $t(7) = -3.09$, $p < .05$. We speculate that, in the more complex, new category search tasks, users tended to rely more on usage of the 2D list to navigate the hierarchy because it was more familiar to them, since all of our users had been using these sorts of user interface controls for many years. When performing a simple, targeted category search, users relied equally on usage of either the 3D graph layout or the 2D list view.

While any discussion as to why users tended to use the 2D components for some tasks, and the 3D components for others is of course purely speculative, it deserves our attention and will be a subject of further study. In particular, it would be advantageous to think of a better implementation of the 3D hyperbolic graph that would support the kinds of searching required in new category tasks.

3.4 Subjective Ratings

The conventional 2D user interfaces were rated slightly higher on user satisfaction than was XML3D (4.85 v. 4.5, respectively, on a 7-point scale; 1=low, 7=high). This difference was not reliable, $t(13)=-0.5$, $p=0.30$. XML3D did score higher than the 2D tools when users were asked whether or not the software was pleasantly surprising (average rating =4.6 v. 3.07 for the 2D tools; again, this difference was not reliable $t(13)=1.12$, $p=0.14$).

5.0 Discussion

In order to address a lack of user evaluations across a wide variety of information visualizations of web content, an empirical evaluation of three different user interface tools was presented. Using the Snap.com hierarchy content, simulated web developer tasks were given to target end users. These tasks included searching for categories that had one or more parents in the information space, as well as tasks for which no parent or category currently existed. The results of the study revealed an interesting pattern of strengths and weaknesses for the three different user interfaces. Specifically, a performance advantage in terms of task times emerged for the novel browser XML3D when existing category tasks were being performed; but this advantage dissipated for new category tasks. When existing category tasks were presented to users in the XML3D user interface, the users tended to utilize the 3D graph and the 2D list user interface components about equally often. However, the 2D list components of XML3D were used much more frequently when new category tasks were presented to users.

When participants performed tasks faster with XML3D (existing category tasks), no decline in the quality of their responses was observed. In other words, the benefit in performance was not offset by a lack of agreement across users as to where web content should be placed.

The benefits of using the 3D Hyperbolic Browser for existing category tasks appears to be related to a complementary use of the 3D graph and the 2D list features. Benefits were observed only when the graph and lists were used equally often. In contrast, participants used the 2D list user interface components more for tasks in which a new category had to be found within the existing scheme, and no reliable benefits of the XML3D were observed in these instances. Future studies should explore optimal ways of integrating the use of novel 3D graph layouts and 2D lists for effective information retrieval, so that other kinds of tasks might benefit from the new designs.

One obvious explanation for the XML3D advantage during existing category tasks lies in the way in which XML3D provides more accessible focus and context. The XML3D browser reveals the overall structure and the local neighborhood content similarities within one integrated view (the 3D graph layout). In addition, the InfoBar to the right hand side integrates structure and semantic relationships via a 2D list of features. The other two conventional user interfaces do not do this as effectively (as more focus information is provided, global context information begins to scroll out of view). There are 2D designs that provide more of a focus + context view by using distortions such as "fish-eye views" (e.g., Furnas, 1986), but no published user studies are available at this time for comparison purposes. The XML3D design may have proven beneficial during some tasks primarily because current users can best leverage the 3D hyperbolic graph layout when it is accompanied by a 2D list, or vice versa. It might also simply have been the case that the 2D list provided better global transitioning capabilities for new category tasks, and hence users opted for it in these circumstances. Future studies will need to examine usage patterns for novel graph visualizations v. 2D elements for different types of tasks to better distinguish these possibilities. However, giving users the option to use a novel visualization in addition to a traditional 2D list view appears to be a powerful design solution, at least until alternative user interfaces become more pervasive.

Future work will focus on better incorporating intelligent search functionality into our novel browser designs. In the version of XML3D used for the study, only text string matching was supported. It will be important to enrich the search algorithm used for matching in future designs. Recent studies in our group (Czerwinski et al., 1999) have also shown that visualizing implicit query matches to currently selected items via subtle highlighting techniques is a powerful information management technique. Implicit query could be especially useful to Web directory managers, by indicating where to best place existing content, as how many related categories might exist. In addition, we observed participants in the current study clearly wanting to perform searches on simple subsections of an area of the hyperbolic graph in XML3D. This would be a nice feature to implement in future versions of our browser designs.

We have described a novel Web browser, XML3D, and empirically evaluated it against two more conventional 2D user interface designs for visualizing hierarchical Web data. We found that the combination of a 3D hyperbolic graph view and a 2D list view is a viable user interface for Web directory management tasks. The empirical results of the user study supports some, but not all of our intuitions about the power of providing a visualization of linked views in the context of an interactive 3D hyperbolic graphics system. Iterative test and redesign of the user interface to better support tasks for new categories remains as future work.

6.0 Acknowledgements

We would like to thank Doug Szabo for his development expertise and work on XML3D.

7.0 References

ANDERSON, P. (1995). Collapsible Highgraphs: A Folding Paradigm for Hypertext Visualisation**.** *Proceedings of OZCHI'95, the CHISIG Annual Conference on Human-Computer Interaction***,** p.114-117.

CARD, S., MACKINLAY, J. and SHNEIDERMAN, B. (1999). Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann.

CRAMPES, M. (1997). Auto-Adaptive Illustration through Conceptual Evocation. In *DL'97: Proceedings of the 2nd ACM International Conference on Digital Libraries,* p.247-254.

CZERWINSKI, M., DUMAIS, S.T., ROBERTSON, G.G., DZIADOSZ, S., TIERNAN, S. and VAN DANTZICH, M. (1999). Visualizing implicit queries for information management and retrieval. In *Proceedings of CHI '99, Human Factors in Computing Systems*, (Pittsburgh, May 17-20, 1999), ACM press.

DUMAIS, S.T. & CHEN, H. (2000). Bringing order to the Web: Automatically categorizing search results. To appear in *Proceedings of CHI 2000, Human Factors in Computing Systems*, (The Hague, April 1-6, 2000), ACM press.

FURNAS, G.W. (1986). Generalized fisheye views. *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems,* p.16-23.

FURNAS, G.W., ZACKS, J. (1994). Multitrees: Enriching and Reusing Hierarchical Structure. *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems,* v.1 p.330-336.

KEAHEY, T. A., and ROBERTSON, E. L. (1997). Nonlinear magnification fields. In *Proceedings of the IEEE Symposium on Information Visualization*, IEEE Visualization, October 1997.

LAMPING, J., RAO, R., and PIROLLI, P. (1995). A Focus+Content technique based on hyperbolic geometry for viewing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems,* New York, May 1995. ACM.

LOWTHER, K. & WARE, C. (1996). Vection with large screen 3D imagery. *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems,* v.2, p.233-234.

MACKENZIE, I. S. & WARE, C. (1993). Lag as a determinant of human performance in interactive systems. *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems,* p.488-493.

MACKINLAY, J.D., ROA, R. and CARD, S.K. (1995). An organic user interface for searching citation links. *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems,* v.1 p.67-73.

MUNZNER, T. (1997). Laying Out Large Directed Graphs in 3D Hyperbolic Space. *Proceedings of the 1997 IEEE Symposium on Information Visualization*, 2-10.

MUNZNER, T. (1998a). Exploring Large Graphs in 3D Hyperbolic Space. *IEEE Computer Graphics and Applications,* Vol. 18, No. 4, pp.18-23, July/August 1998.

MUNZNER, T. (1998b) Drawing Large Graphs with H3Viewer and Site Manager. Proceedings of Graph Drawing '98, Montreal, Canada, August 1998, *Lecture Notes in Computer Science,* 1547, pp. 384-393, Springer-Verlag.

MUNZNER, T. and BURCHARD, P. (1995). Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. *Proceedings of VRML '95* (San Diego, California, December 14-15, 1995), special issue of Computer Graphics, pp 33-38, ACM SIGGRAPH, New York, 1995.

NOIK, E.G. (1993). Exploring large hyperdocuments: Fisheye views of nested networks. *Proceedings of ACM Hypertext'93,* p.192-205.

POULOVASSILIS, A. and LEVENE, M. (1994). A nested-graph model for the representation and manipulation of complex objects. *ACM Transactions on Information Systems*, v.12 n.1 p.35-68.

ROBERTSON, G.G., CZERWINSKI, M., LARSON, K., ROBBINS, D., THIEL, D. & VAN DANTZICH, M. (1998). Data Mountain: Using spatial memory for document management. In *Proceedings of UIST '98, 11th Annual Symposium on User Interface Software and Technology*, pp. 153-162.

SALMINEN, A., TAGUE-SUTCLIFFE, J. and McCLELLAN, C. (1995). From text to hypertext by indexing. *ACM Transactions on Information Systems*, v.13 n.1 p.69-99.

Sebrechts, M.M., Vasilakis, J., Miller, M.S., Cugini, J.V., & Laskowski, S.J. (1999). Visualization of search results: A comparative evaluation of text, 2D and 3D Interfaces. *Proceedings of SIGIR '99*, 3-10.

WANG, W. and RADA, R. (1998). Structured hypertext with domain semantics. *ACM Transactions on Information Systems*, v.16, n.4, p.372-412.

WICKENS, C.D., LACLAIR, M. and SARNO, K. (1995). Graph-task dependencies in three-dimensional data: Influence of three-dimensionality and color. *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting,* v.2, p.1420-1424.

WICKENS, C.D., LIANG, C., PREVETT, T and OLMOS, O. (1994a). Egocentric and exocentric displays for terminal area navigation. *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting,* v.1, p.16-20.

WICKENS, C.D., MILLER, S. and THAM, M. (1994b). The implications of data-link for representing pilot request information on 2D and 3D air traffic control displays. *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting*, v.1, p.61-65.