



# Institute for Software Research

University of California, Irvine

## User Experiments with Tree Visualization Systems



Alfred Kobsa  
Univ. of California, Irvine  
kobsa@ics.uci.edu

November 2003

ISR Technical Report # UCI-ISR-03-11

Institute for Software Research  
ICS2 210  
University of California, Irvine  
Irvine, CA 92697-3425  
[www.isr.uci.edu](http://www.isr.uci.edu)

[www.isr.uci.edu/tech-reports.html](http://www.isr.uci.edu/tech-reports.html)

# **User Experiments with Tree Visualization Systems**

Alfred Kobsa  
Institute for Software Research  
University of California, Irvine  
Irvine, CA 92697-3425  
kobsa@uci.edu

ISR Technical Report # UCI-ISR-03-11

26 Nov. 2003

## **Abstract:**

This paper describes between-subjects experiments with five well-known tree visualization systems, and Windows Explorer as a baseline system. Subjects performed tasks relating to the structure of a directory hierarchy, or to attributes of files and directories. Task completion times, correctness and user satisfaction were measured. Significant system and task type effects and an interaction between system and task type were found. We analyze the reasons for these observed differences and present several design recommendations and conclusions.

# User Experiments with Tree Visualization Systems

Alfred Kobsa\*  
University of California, Irvine

## Abstract

This paper describes between-subjects experiments with five well-known tree visualization systems, and Windows Explorer as a baseline system. Subjects performed tasks relating to the structure of a directory hierarchy, or to attributes of files and directories. Task completion times, correctness and user satisfaction were measured. Significant system and task type effects and an interaction between system and task type were found. We analyze the reasons for these observed differences and present several design recommendations and conclusions.

**CR Categories 1998:** H.5.2 [User Interfaces]: Graphical User Interfaces (GUI) H.1.2 [User/Machine Systems]: Human factors I.3.6 [Methodology and Techniques]: Interaction Techniques

**Keywords:** information visualization, experimental comparison, task performance, accuracy, user satisfaction

## 1 Introduction

Relative comparisons between information visualization systems can provide valuable information about the effectiveness and the ease of use of such systems. An analysis of the reasons behind observed differences in system usage can help identify success factors for the design of visualization systems.

Only few such studies were carried out so far (see Section 6 for an overview). In this paper, we describe a between-subject experiment where we compared five well-known tree visualization systems, and Windows Explorer as a baseline system. Section 2 gives a brief overview of these systems. Section 3 describes the methods that we used in our study. Section 4 presents the overall quantitative results with respect to correctness, speed of task performance and user satisfaction. Section 5 discusses the results and implications individually for each system. Section 6 reviews related studies, and Section 7 summarizes conclusions from this study and proposes implications for future experiments.

## 2 Visualization systems used in this study

### 2.1 Treemap 3.2<sup>1</sup>

Treemap [Johnson and Shneiderman 1991; Shneiderman 1992] is a space-filling visualization that depicts subordination of tree

nodes through recursive subdivision of rectangles that represent these nodes. The size of each rectangle is proportional to the size of the node, and users can choose among several partitioning algorithms. Labels and color inside the visualization indicate selectable information about files. Filtering of files by size and age grays them out, and filtering by depth of nesting removes undesired substructures. Figure 1 shows our test hierarchy when viewed with Treemap.

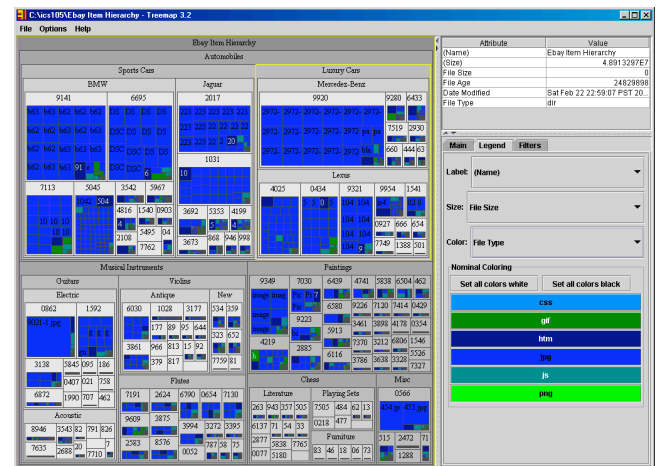


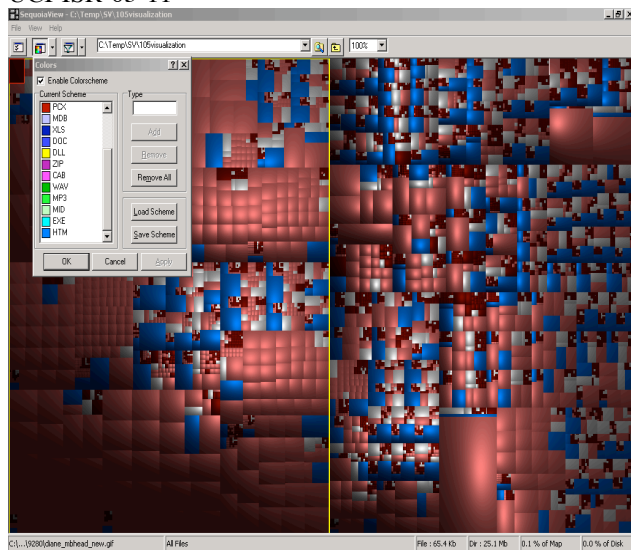
Figure 1. Test tree seen with Treemap  
(different colors denote different file types)

### 2.2 SequoiaView 1.3

SequoiaView (formerly Cushion Treemaps [van Wijk and van de Wetering 1999]) visualizes trees in a similar manner as Treemap. SequoiaView however offers the option to also add shading and spotlighting as extra cues to emphasize the tree structure through 2 1/2D appearance. It also provides additional functionality for filtering, sorting, highlighting and explicit inclusion or exclusion of files. Filtering does not grey out but omits the filtered files. Figure 1 shows our test hierarchy when viewed with SequoiaView.

\* E-mail: kobsa@uci.edu, WWW: <http://www.ics.uci.edu/~kobsa>. This research has been supported by a grant of the Center for Research on Information Technology and Organizations (CRITO). The author wishes to thank Frank van Ham, Catherine Plaisant, Ramana Rao, Ben Shneiderman and Jarke van Wijk for making their systems available and/or providing valuable input on the experimental tasks.

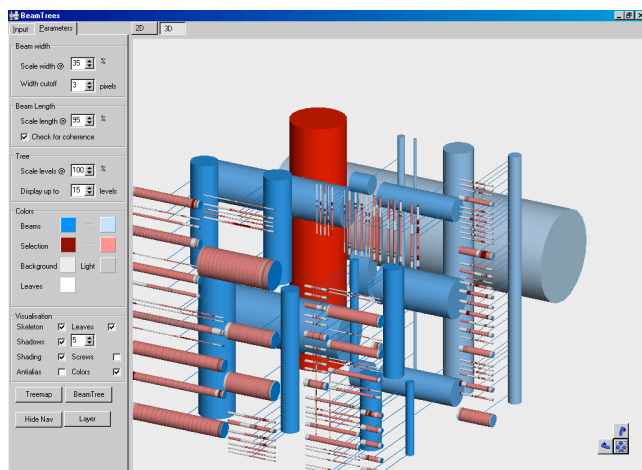
<sup>1</sup> A more recent release, Treemap 4.0, was not yet available at the time of this experiment.



**Figure 2. Test tree seen with SequoiaView**  
(different colors denote different file types)

### 2.3 BeamTrees

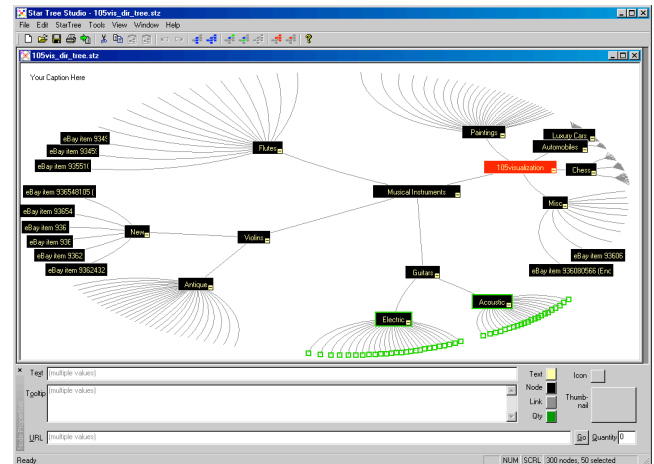
BeamTrees [van Ham and van Wijk 2002] features both a space-filling Treemap-like visualization, and a 3D node-link visualization which is depicted in Figure 3. Each individual beam represents a directory, and the slices on the beams represent files. Different file types have different colors. Users can rotate and zoom the display, and brush files and folders to obtain some information about them.



**Figure 3. Test tree seen with BeamTrees**  
(different colors denote different file types, and selection)

### 2.4 Star Tree Studio 3.0

Star Tree (formerly Hyperbolic Browser [Lamping et al. 1991, 1995; Lamping and Rao 1996]) presents trees in a node-and-link fashion and thereby follows a distortion-based visualization paradigm. Specifically, it uses fish eye distortion as the basis for a focus + context presentation and interaction. Users can shift the focus by dragging operations, and obtain information about folders and files by clicking on them. There is also a “Find and Replace” function where user can choose which field of the node properties to search. The results will be highlighted in the tree. Figure 4 shows our test tree with Star Tree.



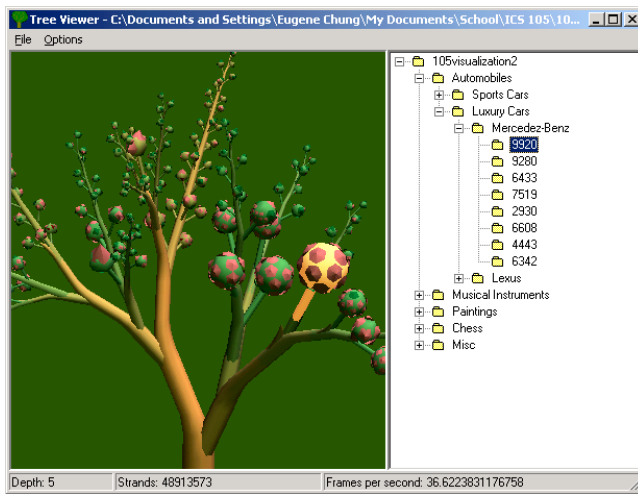
**Figure 4. Test tree seen with Star Tree**  
(different colors denote highlighting and selection)

### 2.5 Tree Viewer

The TreeViewer [Kleiberg et al. 2001] visualizes trees in a form that closely resembles botanical trees (see Figure 5). The root directory is represented by the tree stem. Subdirectories are branches (multiple subdirectories branch off one by one). Terminal directories are the “bulbs” at the end of the branches, and files are discs-shaped “fruits” on top of the bulbs. Files and directories at the same level are displayed in the same color. The size of a directory is represented by the thickness of the corresponding branch or bulb, and the size of a file by the size of its fruit.

The user interface is divided into two parts: the left panel displays the 3-dimensional visualization, and the right panel contains an Explorer-like presentation of the same tree. Clicking on a directory in the left-hand panel will highlight the corresponding visualization (in Figure 5, the visualization of directory ‘9920’ is highlighted in yellow). Users can move and rotate the tree, and zoom in and out. They can also change the colors of the tree, background, leaves and branches, and change the general appearance of the tree.





**Figure 5. The test tree seen with Tree Viewer**  
(same colors denote same hierarchy level)

## 2.6 Windows Explorer

Windows Explorer is the integrated file browsing and management tool for the Windows operating system. It employs an expandable tree structure to represent the directory hierarchy, folder icons to represent directories, and different icons to represent files. For file management, Windows Explorer provides functions such as sorting, searching, and file detail viewing.

The window is divided into two panes. The left pane displays a view of the folder structure. The right pane shows details of the files and folders within a particular selected folder. The row of buttons above these two panes is the toolbar, which provides shortcut methods for controlling Explorer features. The area along the bottom of the window provides information about objects currently selected within the folder and file detail panes. A menu bar at the top of the window provides access to the full set of commands for controlling explorer features.

## 3 Methods

The aim of the experiment was to determine whether solving tasks in the six systems would differ with respect to task completion times, accuracy, and user satisfaction.

### 3.1 Tasks employed

Users had to solve 15 tasks in a tree-like product hierarchy that was taken from Ebay. Tasks were generated and selected by the experimenters in an iterative brainstorming process based on whether or not they were interesting and would naturally occur in the analysis of the respective data sets by a hypothetical Ebay seller. In some cases, questions had to be rephrased using a technical terminology in order to make them unambiguous. The following are the 15 tasks that were eventually selected:

- Q1. What is the maximum depth of the Ebay hierarchy?
- Q2. Is the tree balanced or unbalanced? (A tree is unbalanced if its depth in one branch is at least two more than in another branch)
- Q3. Find the name of the parent directory of the directory "BMW".
- Q4. Which directory has the greatest number of immediate subdirectories?
- Q5. Which file was modified most recently?

- Q6. Find the total number of bottom-level directories under "Automobiles".
- Q7. Which bottom-level directory of the Ebay item hierarchy has the largest number of files?
- Q8. Find a file of type .css and write down its name
- Q9. Find the directory that contains the most .png type files.
- Q10. Find all files added on Feb. 19, 2003. Write down their file names.
- Q11. Locate the file labeled 1990.htm.
- Q12. What is the name of the largest file in the Ebay items hierarchy?
- Q13. Find the number of directories WITHOUT a file of type .js.
- Q14. Which directory includes a deeper hierarchy: "Flutes" or "Guitars"?
- Q15. Find the two duplicate occurrences of the directory "1869". Indicate their pathnames.

Questions 1-4, 6, 7, 9, 14 and 15 refer to the structure of the tree hierarchy, while Questions 5, 8 and 10-13 refer to attributes of subdirectories or files. In the following, we will refer to these two different task types as "structure-related tasks" and "attribute-related tasks".

### 3.2 Subjects

48 subjects participated in the experiment. They were students with a major or minor in Information and Computer Science or Engineering who had at least one year of experience working with computers. Prior studies with the adoption of information visualization systems by administrative data analysts [Gonzales and Kobsa 2003a, b] indicated that the subject population was reasonably representative for the target user group, due to the analytical capabilities they have in common. The results of one subject were discarded since this person did not follow the instructions thoroughly enough.

### 3.3 Experimental Design

The experiment took place in a small laboratory on the campus of the University of California, Irvine. A between-subjects design was used, with the visualization system and the task type as the independent variables. The 48 subjects were randomly assigned to each condition. Groups of 2-4 students received half an hour of instruction on the visualization system they were assigned to, and familiarized themselves with the Ebay hierarchy. Thereafter they solved controlled practice tasks for another twenty minutes. During this practical training they received additional instruction from 2-3 experimenters.

Subjects then began the experiment. They had to answer each of the 15 questions, write down the answer(s), and self-time the duration of task completion. Subjects were instructed to abort unsolved tasks after 5 minutes and to note down this fact. Their interaction was recorded by screen capture software. At the end of the experiment, they completed a final questionnaire for about 10 minutes.

The correctness of users' task performance and the task completion times were measured based on their answers in the answer sheet. The user satisfaction data were taken from the final questionnaire. A Chi square test was performed to measure the effect of the system used on task correctness, and an ANOVA (with Fisher's PLSD) to analyze the system effect on task completion times and user satisfaction. All significant differences found will be discussed below.

## 4 Quantitative results

### 4.1 Correctness

Figure 6 shows (from bottom to top) the number of correct answers for each system, the number of incorrect answers, the number of cases in which subject indicated having exceeded the 5-minute timeout, and the number of cases when subjects gave no answers.

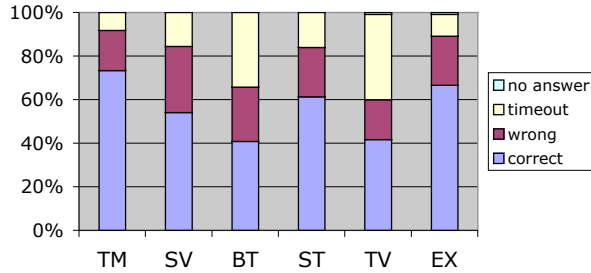


Figure 6: Correctness of answers

TreeMap yielded the highest number of correct answers (73.3%), while the correctness of BeamTrees (40.8%) and Tree Viewer (41.7%) was the lowest. The timeout rate was highest for Tree Viewer (39.2%), and lowest for Treemap (8.1%) and Windows Explorer (10%). A pair-wise Chi Square test found that the following differences in the count of correct answers were significant at the 5% level (<) or the 1% level (<<):<sup>2</sup>

BT << TM, SV, ST, EX

TV << TM, ST, EX

TV < SV

A Chi Square test reveals a significant difference between the structure-related and the attribute-related tasks regarding the number of correct responses ( $p < 0.001$ ). We therefore break down the correctness results by task type.

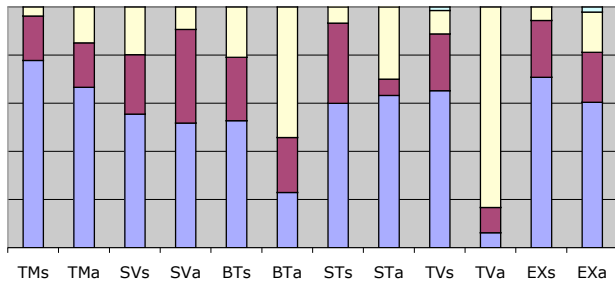


Figure 7: Correctness broken down by structure (s) and attribute-related tasks (a). See Figure 6 for the legend.

Figure 7 shows the data of Figure 6, broken down by structure-related tasks (s) and attribute-related tasks (s). The differences between answer correctness for structure and attribute is significant for BeamTrees ( $p=0.001$ ) and Tree Viewer ( $p<0.0001$ ).

A Chi Square test comparing the answer correctness by task type reveals the following significant differences:

### Structure-related tasks

BT << TM

< EX

SV << TM

< EX

ST < TM

### Attribute-related tasks

BT << TM, SV, ST, EX

TV << TM, SV, ST, EX

< BT

### 4.2 Speed of task performance

Figure 8 shows the average total performance times for the 15 tasks, per system. Whenever subjects indicated having exceeded the 5-minute limit, the task performance time was set to 300 seconds. The few cases where subjects gave no answer were disregarded. Windows Explorer and Treemap had the shortest task performance times (101.2 and 106.5 seconds per task), while BeamTrees had the highest (188.4 seconds per task).

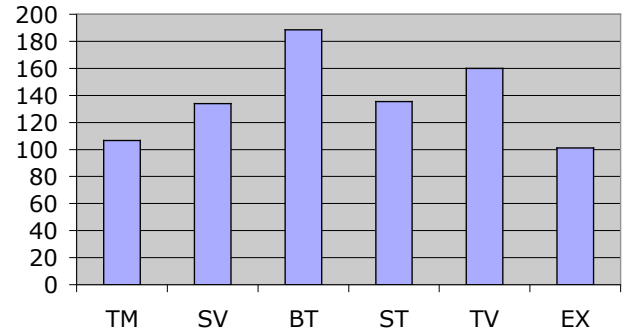


Figure 8: Average task completion times (in seconds)

An ANOVA revealed a significant effect of the system used ( $p < 0.0001$ ). Fisher's PLSD showed that the following differences were significant at the 5% level (>) and at the 1% level (>>):

BT >> TM, SV, ST, EX

BT > TV

TV >> TM, EX

TV > SV

SV > TM, EX

ST > TM, EX

The effect of task type and the interaction between system used and task type is also highly significant ( $p<0.0001$ ). We therefore break down the average task completion times by task type (see Figure 9).

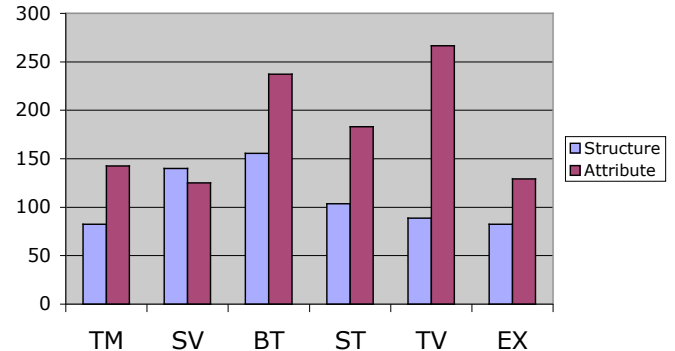


Figure 9: Average task completion times by task type and by system used (in seconds)

<sup>2</sup> The system on the left-hand side of the < and << operators is the slower system.

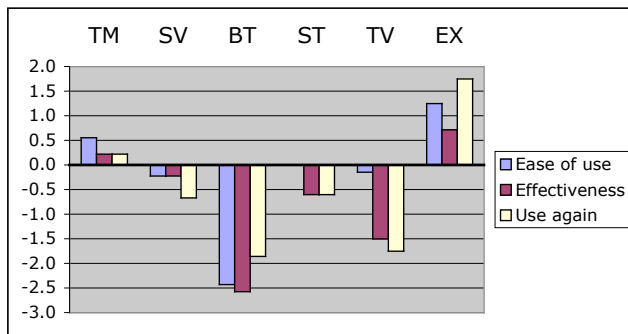
The differences of the task completion times between task types are significant at the 1% level for all systems except SequoiaView. An ANOVA with Fischer's PLSD yields the following statistically significant differences between the task completion times per task type.

Structure-related tasks	Attribute-related tasks
BT >> TM, TV, EX	BT >> TM, SV, EX
BT > ST	BT > ST
SV >> TM, TV, EX	TV >> TM, SV, ST, EX
SV > ST	ST >> SV
	ST > EX

As can be seen, Tree Viewer and StarTree change their relative positions considerably depending on whether structure or attribute related tasks are performed.

### 4.3 User satisfaction

Figure 10 shows subjects' responses to questions relating to the ease of use and effectiveness of the system they had worked with, and to the question whether they would use the system again for analyzing data. Scales ranged from -3 to 3.



**Figure 10: Perceived easy of use and effectiveness, and replies to "use system again?"**

The following differences in the responses to these usability questions are significant at the 5% or 1% level:

Ease of use	Effectiveness
BT << TM, SV, ST, TV, EX	BT << TM, SV, EX
	BT < ST
EX > SV, TV	TV << TM, EX

#### Use system again?

BT << EX  
BT < TM  
EX >> SV, TV  
EX > ST  
TM > TV

## 5 Individual observations and discussion

### 5.1 Treemap

Treemap turned out to arguably be the best visualization system overall. Subjects who used it excelled both in the correctness of their answers and in their task solution times. There were merely two questions (namely Q5 and Q10) that most subjects were unable to answer correctly. Treemap also received comparatively

high user satisfaction ratings. In free comments, users expressed the view that the system is useful for obtaining a very general overview of a directory, but should be complemented by more powerful tools.

There were no significant differences between Treemap and Windows Explorer. While it is true that the Explorer data may be somewhat skewed due to the fact that all subjects had been using Windows for a long time, it is doubtful whether increased practice would enable Treemap users to clearly outperform Explorer users. Significant enhancements of Treemap seem still to be necessary before users will feel motivated to use this tree visualization system as a substitute of Windows Explorer in an MS Windows environment.

### 5.2 SequoiaView

SequoiaView showed to be an "average" system with respect to answer correctness, overall task solution times, and user satisfaction. Users had troubles answering Q5, Q7, Q10 and Q13 correctly. They also took a comparatively long time answering the structure-related questions, which is surprising at first sight. SequoiaView was originally designed to improve users' perception of tree structures by replacing the 2D rectangles of Treemap with shaded 2 1/2D "cushions". However, the resulting spatial appearance did seemingly not help very much in our domain, which contained relatively few directories with many files of mostly similar sizes (see Figure 2). To figure out the tree structure, users tended to move the mouse around (to prompt the system to delineate the current file and embedding directory), or would use the depth-limiting function that the system provides.

### 5.3 BeamTrees

BeamTrees arguably achieved the worst results in our experiment, both with respect to correctness (specifically for the attribute-related questions), task performance times, and user satisfaction. At first sight, this is surprising since BeamTrees offers a Treemap-like visualization in addition to its 3D node-link visualization. One would therefore intuitively expect that BeamTrees performs at least as well as Treemap.

The fact that this was not the case is mostly due to the lack of functionality beyond the mere visualization of the tree structure and the display of path names when brushing over file and directory representations. Tasks that required much more than this were not correctly answered (like e.g. Q5 and Q10 which required file dates that were not available from the system). In attribute-related questions, BeamTrees users had a significantly lower correctness rate than in structure-related questions (see Figure 7).

### 5.4 Star Tree

The Star Tree performance was "average", not only on a global level but also for nearly every individual question. The only major exceptions are Q5, Q6, Q7, Q9 and Q12, for which the correctness was very low. The nature of these tasks seems to indicate that it was difficult for users to scan numerous files and directories. Several users suggested a function that would count the number of nodes in a directory, seemingly because the fisheye perspective made a manual count difficult if the number of nodes became large. The addition of zoom and sorting functionality were also recommended.

Star Tree was rated as the most aesthetically pleasing system in the final questionnaire, but this result is not statistically significant.

## 5.5 Tree Viewer

Tree Viewer was clearly at the low end of the performance results. The data were particularly poor for attribute-related tasks. This is not surprising though. Like BeamTrees, it lacks basic functionality (e.g., search operations) and data displays (e.g., of file attributes) that is required or would at least have been very helpful to solve some tasks. Subjects therefore found Q5 and Q7-Q13 impossible or extremely difficult to answer.

Tree Viewer's Explorer-like listing of directories on the right-hand side was used very frequently, more than in any other system ( $p < 0.001$ ). This was even the case for questions like Q14 that one would expect to be solved in the visualization proper. The reason for relying on the directory nesting is most likely that continuing branches representing a directory cannot be easily followed at the branching points. To figure out what part of a branch constitutes a directory, subjects would click on the directory in the Explorer panel and then frequently remain in this panel to complete the task.

The rating for aesthetical appeal of Tree Viewer was slightly above average (0.25 on a scale from -3 to 3). However, the standard deviation of these ratings was by far the highest for all systems (1.98). User comments ranged from "I like the tree design" to "I hate this tree".

## 5.6 Windows Explorer

Windows Explorer showed a very good overall performance, both with regard to correctness, speed of task completion, and user satisfaction. Any comparison with other systems must be viewed with a grain of salt though since the subjects can be assumed to be highly skilled at least in its basic functionality. On the other hand, one should not overestimate the expected performance gains through more practice since all tested visualization systems are relatively simple and since the training was quite thorough.

Explorer users had considerable difficulties with Q13 since it required expanding and scanning all directories until a solution is found. None of our subjects was able to solve the problem. For similar reasons, they also had troubles with Q7. They also did not well in Q2, which required them to compare the depth of different subtrees.

## 6 Related work

[Kobsa 2001] compared three commercial information visualization systems for multidimensional data with respect to task completion times and error rate. [Mark et al. 2002; Mark et al. 2003] extended this comparison to two different forms of synchronous collaborative data visualization. [Wiss et al. 1998] analyzes the extent to which three 3-D browsers for hierarchies support the seven central tasks of information visualization systems postulated by [Shneiderman 1996]. [Barlow and Neville 2001] compared four different visualization paradigms (among them a Treemap-style approach) with respect to their ability to communicate the topology of the tree and support comparisons of node size. The tested prototypes were not full-fledged visualization systems as in our case, but merely demonstrated the "bare bones" of the four different visualization paradigms. The

Treemap-style visualization turned out to be the slowest for most tasks. It is difficult though to compare this result with our findings about Treemap 3.2 since the two visualizations are too different. For instance, Treemap 3.2 uses superpositioned hierarchical bars with directory labels that make the hierarchical structure more noticeable. This is completely lacking in the simplified reconstruction of [Barlow and Neville 2001].

[Lamping et al. 1995; Czerwinski and Larson 1997; Pirolli et al. 2000] evaluated various versions of Star View and standard 2-D browsers similar to Windows Explorer, but were unable to find performance differences. [Pirolli et al. 2000, 2001, 2003] found performance gains of Star View though when the information scent (i.e., clues to the direction and proximity of sought information) was high, and performance losses otherwise. Finally, *The Great CHI '97 Browse-Off* contest [Mullet et al. 1997] yielded Star View as the winner, ahead of Windows Explorer. [Pirolli et al. 2003] shows however that individual differences between subjects have a considerable higher effect on performance than differences between the two browsers, which may explain this outcome.

## 7 Summary and conclusion

This study compared several information visualization systems for tree hierarchies in a between-subject experiment. It unveiled statistically significant differences with respect to accuracy of task performance, task completion times and user satisfaction. It explained these differences by referring to missing functionality or characteristics of visualization paradigms that make certain types of tasks difficult or impossible. The study also showed a significant interaction between system used, and structure-related versus attribute-related task types.

While at least one system achieved the same performance as Windows Explorer, none of them showed benefits for users that went significantly beyond this baseline. Moreover, [Gonzales and Kobsa 2003a, b] found that the integration into users' current work environment is an extremely important factor for the adoption of a visualization system by administrative data analysts. Since none of the tested visualization systems is very much integrated into a file management system, their potential adoption must therefore be viewed with skepticism without significant enhancements.

Two more conclusions can be drawn from the results, which may be relevant for future experiments. First, this study showed the merits of distinguishing structure and attribute related tasks, for which some systems behave differently. So far, only local/global retrieval/comparison tasks have been distinguished in pure concept hierarchies [Pirolli et al. 2000, 2001, 2003]. Second, the extreme outliers in the Tree Viewer and BeamTrees data have mostly been caused by lack of functionality beyond the pure visualization. Once this problem is corrected, the effect size of differences in the visualization paradigms is going to be much smaller. This fact will have to be taken into account when planning the design, and specifically the sample size, of future experiments.

## 8 References

- Barlow, T. and P. Neville (2001). A Comparison of 2-D Visualizations of Hierarchies. IEEE InfoVis, San Diego, CA, 131-138. file://Lit1/barlow-neville-InfoVis-2001.pdf



- Czerwinski, M. P. and K. Larson (1997). The New Web Browsers: They're Cool But Are They Useful? People and Computers XII: Proceedings of the HCI '97 Conference, Springer Verlag.
- Gonzales, V. and A. Kobsa (2003a). Benefits of Information Visualization Systems for Administrative Data Analysts. Submitted to IV03, London, England.
- Gonzales, V. and A. Kobsa (2003b). User Adoption of Information Visualization Systems. I-KNOW'03 Workshop on Knowledge and Information Visualisation 2003 (KIV2003), Graz, Austria.
- Johnson, B. and B. Shneiderman (1991). Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information. IEEE Visualization'91, San Diego, CA, 284-291.
- Kleiberg, E., H. van de Wetering and J. J. van Wijk (2001). Botanical Visualization of Huge Hierarchies. InfoVis 2001: IEEE Symposium on Information Visualization, San Diego, CA, 87-94.
- Kobsa, A. (2001). An Empirical Comparison of Three Commercial Information Visualization Systems. IEEE Symposium on Information Visualization, San Diego, CA, 123-130. <http://www.ics.uci.edu/~kobsa/papers/2001-INFOVIS-kobsa.pdf>
- Lamping, J. and R. Rao (1996). "The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies." *Journal of Visual Languages and Computing* 7(1): 33-55.
- Lamping, J., R. Rao and P. Pirolli (1991). A Focus+Context Technique Based in Hyperbolic Geometry for Visualizing Large Hierarchies. CHI'95 ACM Conference on Human Factors in Computing Systems, 189-194.
- Lamping, J., R. Rao and P. Pirolli (1995). A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. CHI'95, ACM Conference on Human Factors in Computing Systems, 401-408.
- Mark, G., K. Carpenter and A. Kobsa (2003). A Model of Synchronous Collaborative Information Visualization. submitted to IV03, London, England.
- Mark, G., A. Kobsa and V. Gonzales (2002). Do Four Eyes See Better than Two? Collaborative versus Individual Discovery in Data Visualization Systems. Proceedings of the Sixth International Conference on Information Visualisation, London, England, IEEE Press, Los Alamitos, CA., 249-255. <http://www.ics.uci.edu/~kobsa/2002-IV-kobsa.pdf>
- Mullet, K., C. Fry and D. Schiano (1997). On Your Marks, Get Set, Browse! (The Great CHI'97 Browse Off). Human Factors in Computing Systems, CHI '97 (Extended Abstracts), Atlanta, GA, ACM. <http://www.acm.org/sigchi/chi97/proceedings/panel/kem.htm>
- Pirolli, P., S. K. Card and M. M. Van Der Wege (2000). The Effect of Information Scent on Searching Information Visualizations of Large Tree Structures. Advanced Visual Interfaces, AVI 2000, Palermo, Italy, ACM, 261-272.
- Pirolli, P., S. K. Card and M. M. Van Der Wege (2001). Visual Information Foraging in a Focus + Context Visualization. CHI-2001, Seattle, WA.
- Pirolli, P., S. K. Card and M. M. Van Der Wege (2003). "The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser." *ACM Transactions on Computer-Human Interaction (TOCHI)* 10(1): 20-53. <http://doi.acm.org/10.1145/606658.606660>
- Shneiderman, B. (1992). "Tree Visualization with Tree-Maps: 2-D Space-Filling Approach." *ACM Transactions on Graphics* 11(1): 92-99. <http://doi.acm.org/10.1145/102377.115768>
- Shneiderman, B. (1996). The Eyes Have It: a Task by Data Type Taxonomy for Information Visualizations. IEEE Symposium on Visual Languages, Boulder, CO, 336-33.
- van Ham, F. and J. J. van Wijk (2002). Beamtrees : Compact Visualization of Large Hierarchies. InfoVis 2002, Boston, MA.
- van Wijk, J. J. and H. van de Wetering (1999). Cushion Treemaps: Visualization of Hierarchical Information. IEEE Symposium on Information Visualization (INFOVIS'99), San Francisco, CA, IEEE.
- Wiss, U., D. A. Carr and H. Jonsson (1998). Evaluating three-dimensional information visualization designs: a case study of three designs. IEEE Conference on Information Visualization, London, England, 137 -144.