Comparative Evaluation of Hierarchy Browsers with the Hierarchical Visualisation Testing Environment (HVTE)

Janka Kasanicka

Comparative Evaluation of Hierarchy Browsers

with the Hierarchical Visualisation Testing Environment (HVTE)

Master's Thesis

at

Graz University of Technology

submitted by

Janka Kasanicka

Institute for Information Systems and Computer Media (IICM),
Graz University of Technology
A-8010 Graz, Austria

September 2006

Advisor: Ao.Univ.-Prof. Dr. Keith Andrews

Vergleichende Auswertung von Visualisierungen von Hierarchien

mit dem Hierarchical Visualisation Testing Environment (HVTE)

Diplomarbeit

an der

Technischen Universität Graz

vorgelegt von

Janka Kasanicka

Institut für Informationssysteme und Computer Medien (IICM),
Technische Universität Graz
A-8010 Graz

September 2006

Diese Arbeit ist in englischer Sprache verfasst.

Betreuer: Ao.Univ.-Prof. Dr. Keith Andrews

Abstract

The field of information visualisation has seen tremendous growth over the last decade, with many new methods and approaches presented in the literature. However, evaluations of these new techniques are still rather scarce. This thesis presents a comparative evaluation of four visualisations of hierarchies: TreeView, Information Pyramids, TreeMap, and Hyperbolic Browser. The visualisations are implemented in Java as part of the Hierarchical Visualisation System (HVS), developed at Graz University of Technology.

A semi-automated testing environment, The Hierarchical Visualisation Testing Environment (HVTE), was built upon HVS to simplify the running of a series of usability tests and to automate the collection of performance data. HVTE automatically provides participants with tasks in the sequence corresponding to their randomly assigned test case. Task completion times as well as typed answers are collected in a database.

A counterbalanced, repeated measures study was designed to compare the four visualisations, with 32 test users each performing 8 tasks. The results showed almost no significant differences in task completion times between the four visualisations. The participants significantly preferred the TreeView in subjective ratings. All participants were familiar with this kind of visualisation and had years of experience working with it. The TreeMap visualisation performed quite well, but was rather unpopular in the subjective ratings. The Pyramids and Hyperbolic visualisations were rated in the middle.

Kurzfassung

Das Feld der Informationsvisualisierung ist in den letzten Jahrzehnten gewachsen. Viele neue Methoden und Ansätze wurden präsentiert. Evaluierungen der neuen Techniken sind noch immer eher selten. Diese Arbeit stellt eine vergleichende Evaluierung von vier Visualisierungen von Hierachien vor: TreeView, Pyramids, TreeMap und Hyperbolic. Die verwendeten Visualisierungen sind implementiert in Java als Teil von Hierarchical Visualisation System (HVS), das auf der Technischen Universität Graz entwickelt wurde.

Das Hierarchical Visualisation Testing Environment (HVTE) wurde speziell entwickelt, um die Evaluierung der Visualisierungen in HVS zu erleichtern. Mit den Teilnehmern wird die entsprechende, zufällig zugewiesene, Testsequenz so einfach wie möglich durchgespielt. In einer Datenbank werden die Zeit pro Aufgabe sowie die Antworten gespeichert.

Die Studie wurde als repeated measures durchgeführt, in der jeder der 32 Teilnehmer 8 Aufgaben erfüllt hat. Die Ergebnisse zeigen kaum signifikante Unterschiede bei den Zeiten pro Aufgabe zwischen den Visualisierungen. Die Teilnehmer haben bei den subjektiven Bewertungen die TreeView Visualisierung signifikant bevorzugt. Alle Teilnehmer sind mit dieser Visualisierung vetraut und haben jahrelange Arbeitserfahrungen. Die TreeMap Visualisierung hat bei den Zeiten pro Aufgabe gut abgeschnitten, wurde aber in den subjektiven Bewertungen nicht gut bewertet. Die Pyramids und Hyperbolic Visualisierungen wurden von den Teilnehmern ähnlich bewertet.

I hereby certify that the work presented in this thesis is my own and that work performed by others is appropriately cited.

Ich versichere hiermit, diese Arbeit selbständig verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient zu haben.

Acknowledgements

I would like to thank my advisor, Keith Andrews, for guiding me through the highs and lows of such a large project as a master thesis.

My special thanks go to all the participants, without wanting to name them all. This thesis would not have been possible without all the friends, colleagues, and friends' friends.

Werner Putz and Alexander Nussbaumer provided me with valuable hints about the HVS - thank you.

An extra special thanks to my fiancé and his family for their help, support and understanding not only during the work on this thesis. I am very grateful to have found my Austrian family in them. Thank you.

Last, but definitely not least, my thanks to my family, especially to my brother and my mother. They have taught me to always go on, even in the hardest of times. I dedicate this work to my mother.

Janka Kasanicka
Graz, Austria, September 2006

Credits

- This thesis was written using Keith Andrews' wonderful skeleton thesis [Andrews, 2006c].
- The following figures are subject to the ACM copyright notice:[1]

  Figure 2.4

  Figure 2.13

  Figure 2.14

  Figure 2.17

  Figure 2.21

  Figure 2.20

  Figure 2.19

  Figure 3.6
- The following figures are subject to the University of Maryland copyright notice:[2]

  Figure 2.11

  Figure 3.3

  Figure 3.5

  Figure 3.12

  Figure 3.14
- The following figures are used with permission of InXight Software Inc.:

  Figure 2.2

  Figure 2.12

  Figure 3.8
- The following figures are used with permission of Keith Andrews:

  Figure 2.5

  Figure 2.7

  Figure 2.8

  Figure 2.15

  Figure 2.18

  Figure 3.2

  Figure 3.7

Figure 3.10

Figure 3.20

Figure 3.22

- Figure 3.9 is used with permission of Tamara Munzer and ©1997 IEEE.
- Figure 3.13 is used subject to the Copyright notice of The Regents of University of California [3]
- Figure 3.24 is used with permission of Jack van Wijk.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The aim of this thesis was to plan and conduct a comparative evaluation of selected information visualisations in the Hierarchical Visualisation System (HVS). The thesis describes the theoretical background in the embedding chapters (Chapters 2 to 6), before giving the details of the actual work of the thesis - the study and its results (Chapters 7 to 9).

The field of information visualisation is broad. A short summary along with an overview of the different approaches is presented in Chapter 2. Selected realisations of information visualisations, grouped into broad categories, are also described in this chapter as well. Four visualisations of hierarchies were evaluated in this thesis. Therefore, the methods and approaches together with several implementations of hierarchical visualisation are summarised in Chapter 3. Chapter 4 presents the Hierarchical Visualisation System (HVS) developed at Graz University of Technology, Austria. The evaluated visualisations are the implementations included in HVS. The necessity for usability evaluation, its methods and techniques, especially in the field of information visualisation, are discussed in Chapter 5. Chapter 6 summarises related work in the field of information visualisation evaluation. Studies evaluating visualisations of hierarchies and their results are presented.

Chapters 7 to 9 give details on the actual work of this thesis: the planing and the realisation of the study, the development of the testing environment, and the analysis and presentation of results. The Hierarchical Visualisation Testing Environment (HVTE) was developed to ease data collection and test sequencing for this evaluation. Chapter 7 describes HVTE in detail. Chapter 8 reviews the whole process of this study: beginning with the planning and methods, to task design, through user acquisition, to testing itself. Chapter 9 gives a detailed analysis of the collected data and the results of the study. A discussion of the results as well as subjective user comments is included.

Outlook and proposals for future work in evaluating the HVS are given in Chapter 10. Appendix A includes the questionnaires the users filled out. Appendix C shows the step-by-step analysis of the data in SPSS.

Chapter 2

Information Visualisation

One problem of our information society is the huge amount of data. Approximately 5 exabytes (10^{18} bytes) [School of Information Management and Systems, 2003] of unique data are produced each year and somehow this data should be processed and understood. One solution is information visualisation. Through the visual channel, humans are capable of receiving large amounts information. Thus information visualisation uses the human perceptual system to help explore, present, and communicate information. Human vision is well adapted to receive information; it is the sense with the highest bandwidth, fast, parallel and pre-attentive. People think visually and are good at scanning, recognising, and remembering images.

The main goal of information visualisation is to convert abstract data and information into a graphical presentation of this data. Some well-known visual effects support the understanding of information. Graphical elements facilitate comparisons via length, shape, orientation, or texture. Animations can show changes through time and colours can help make distinctions.

Information visualisation should present huge amounts of information compactly, from different points of view and at different levels of detail and support visual comparisons (adapted from [Hearst, 2005]). According to Card et al. [1999], information visualisation is

> "The use of computer-supported, interactive visual representations of data to amplify cognition."

Although they have similar goals and means of reaching these, information visualisation is not the same as scientific visualisation. Scientific visualisation represents physical or geometric phenomena and is a tool to enhance scientists' ability to explore large scientific data sets. Thus, scientific visualisation visualises aspects of the "natural world" which have a physical representation.

In information visualisation, on the other hand, information is abstract and does not have a direct physical representation in the world. Information visualisation takes items without a direct physical correspondence and maps them to a 2-D or 3-D physical space. This process is shown in Figure 2.1.

Although there are different approaches to dealing with the representation of data, all of the approaches try to fulfil the information visualisation mantra formulated by Shneiderman [1996]:

> "Overview first, zoom and filter, details on demand."

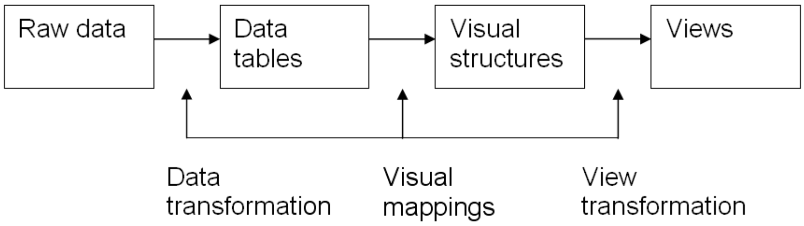


Figure 2.1: The process of information visualisation. [Image adapted from [Stasko, 2006]]

The information visualisation techniques developed so far can be divided into three broad categories depending on the representation method:

- **Focus plus context** is an approach supporting the user's orientation by providing details as well as the surrounding context. 3D perspective offers a natural focus plus context representation by focusing on details in the foreground, but still providing an overview. The fisheye view uses a magnifying glass-like distortion in order to present the details of the inspected area, while the remaining context is presented with less detail. The focus plus context approach is space efficient and details are connected to context. The disadvantages of this approach are distortion, longer learning time, and an unclear zoom factor.

- **Overview plus detail**: a detailed view and an overview are presented in separate maps or windows. The advantages of the overview plus detail approach are scalability, the possibility of multiple overviews and easier implementation. Among the disadvantages are the disconnected detail and overview, space management issues, and the fact that data is replicated in the overview.

- **Space filling techniques** attempt to use as many screen pixels or even subpixels as possible to present information. The whole screen is used to represent the data. Possible patterns occurring in the data can still be seen [Jerding and Stasko, 1998].

There are different types of information by means of which information visualisation can be divided into different categories presented below (adapted from [Andrews, 2006b]).

- *Linear information* such as tables, program source code, alphabetical lists, chronologically ordered items, etc.

- *Hierarchical information* such as tree structures, family histories, files/folders systems on computers.

- *Networks* or *general graph structures*, such as hypermedia node-link graphs, semantic networks, webs, etc.

- *Multidimensional information* is often metadata or “information about information”. File attributes such as type, size, author etc. are examples of multidimensional information.

- *Vector spaces* are known from information retrieval. Examples are text document corpora, word frequencies, similarity measures between documents, document clustering, etc.
- *Query spaces* come from document search. Means for formulating search terms and how the results match them are provided.

The individual information categories and their visualisations are covered in the following sections. Section 2.1 discusses Linear Structures, Section 2.2 Visualising Networks and Graphs, Section 2.3 Multidimensional Data, Section 2.4 Vector Spaces and Section 2.5 Query Spaces. Visualisation of hierarchies is addressed separately in Chapter 3 Visualising Hierarchies.

2.1 Linear Structures

Linear structures represent one-dimensional data. The problem for users can be the feeling of being lost or disoriented due to the lack of context. A visualisation of linear structures presents the context, enables direct access, and encourages the users to explore the data. Through visual overview, users can see whether (or not) information is available and recognise the organisation of data. An overview-plus-detail approach is realised in the SeeSoft system. The focus-plus-context approach is used in the Perspective Wall visualisation.

2.1.1 Perspective Wall/TimeWall

Perspective Wall [Mackinlay et al., 1991] was developed at Xerox PARC. It is now called the TimeWall and is being further developed by InXight Software [Inxight, 2006c]. The Pespective Wall visualises linear information such as time lines. The idea behind the Perspective Wall is to fold a 2D layout onto a 3D wall, but to fully utilise screen space. The central region presents the details while the side panels, one on each side, present the context. Through the perspective, a natural focus-plus-context representation is achieved. Information is presented from left to right, corresponding to the natural reading order (Latin alphabet). Detailed information is presented in the central panel. The left panel shows preceding information. The right panel shows succeeding information. The Perspective Wall uses space efficiently and offers smooth transitions between views. Users interact by moving in a linear direction. When a document is chosen, the wall moves smoothly and transports the chosen document on the central panel. The x-axis represents linear categories such as time. The y-axis can represent other document attributes. Figure 2.2 shows the TimeWall visualisation.

2.1.2 SeeSoft

SeeSoft [Eick et al., 1992] visualises the evolution of complex software systems. With SeeSoft, different metrics of software systems can be made visible. Such metrics include modifications of code or occurrence of bugs. SeeSoft tries to utilise as many pixels on the screen as possible to represent information.

The line representation is the standard view of SeeSoft. In this view, every line of code is represented as a coloured line on the screen. Files are groupings of lines represented as rectangles, where the size of the rectangle represents the size of the file. The line colour can be used to show several aspects, such as date (creation or change), person who wrote or changed the line, and so on. Figure 2.3 shows this standard view.

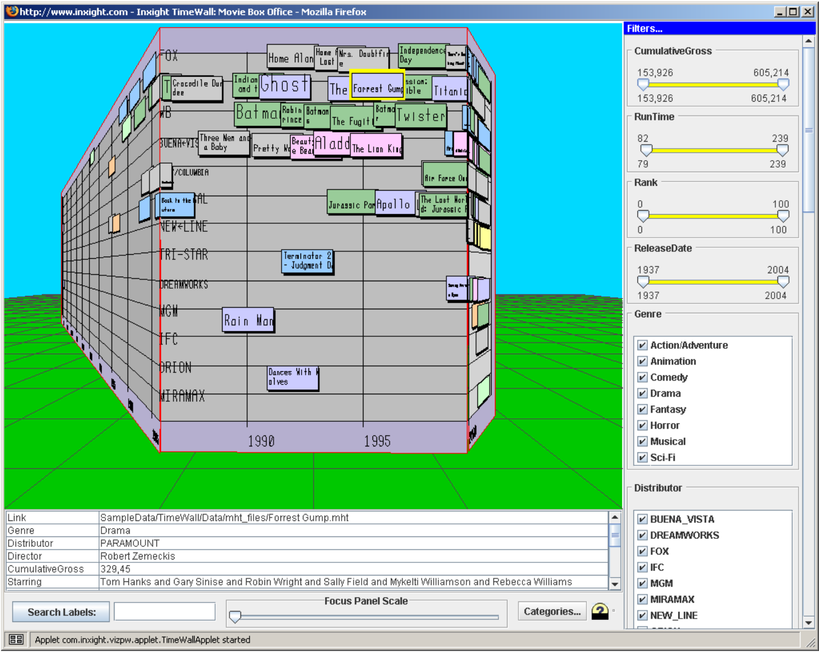


Figure 2.2: TimeWall showing movies released between 1997 and 2004. The front panel represents the years 1985-2000. The colours represent different genres. Details of the selected movie are shown. [Image extracted from [Inxight, 2006c]. Used with kind permission of InXight Software,Inc.]

Other representations include the pixel representation (each line of code taking one or more pixels), the file summary representation (files summarized in a coloured rectangle) and hierarchical representation (similar to a tree map, where the sizes of the blocks and rectangles correspond to their code sizes).

2.1.3 Lifestreams

Lifestreams [Freeman and Fertig, 1995] visualise documents as a time-ordered stream. Lifestreams attempt to minimise the time users need to manage documents and other electronic data. The objective is to increase the user's ability to find and utilise information. The tail of the stream represents old documents; newer documents are nearer to the front, with the newest documents at the head of the stream. Documents can be local files, papers sent by others, incoming or outgoing emails, pictures or movies. A Lifestream can go beyond the present by representing to-do lists, reminders, and calendar items. There are possibilities to organise the stream, search and filter, create calendar items and compress large numbers of documents into overviews. Figure 2.4 shows a Lifestream with a selected item in detail.

2.2 Visualising Networks and Graphs

Graphs consist of vertices (or nodes) connected by edges (or links). The edges of graphs can be directed or undirected and can connect nodes in a way where cycles occur. Trees are a sublass of general graphs: they may not have cycles, their edges are typically directed, and one node is designated as the root. Trees and hierarchies are covered in Chapter 3 of this

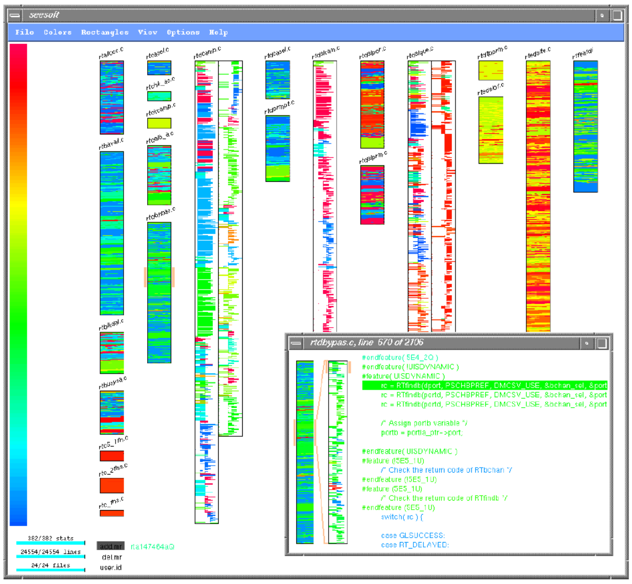


Figure 2.3: SeeSoft standard view showing the code age from newest (red) to oldest (blue). The browser window on the right shows the pixel, line, and text representations of code. [Image extracted from [Ball and Eick, 1996].]

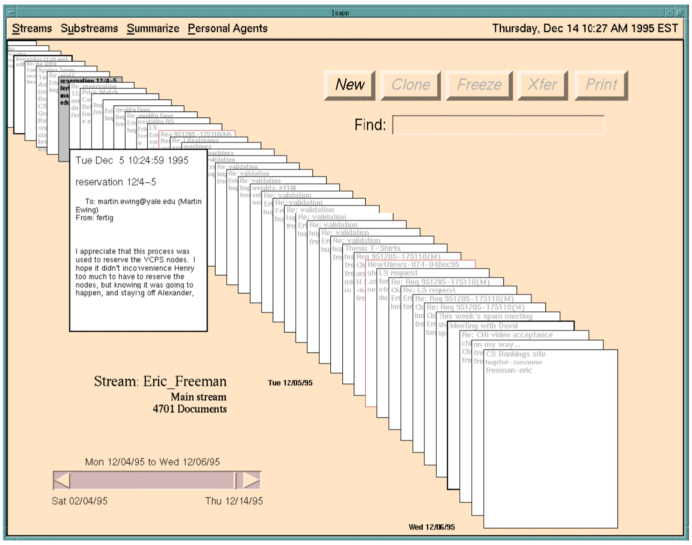


Figure 2.4: Lifestreams showing documents from 2nd to 6th of December, 1995. The selected document is highlighted and shown in detail. [Image extracted from [Fertig et al., 1996]. Copyright ©Association of Computing Machinery, Inc.]

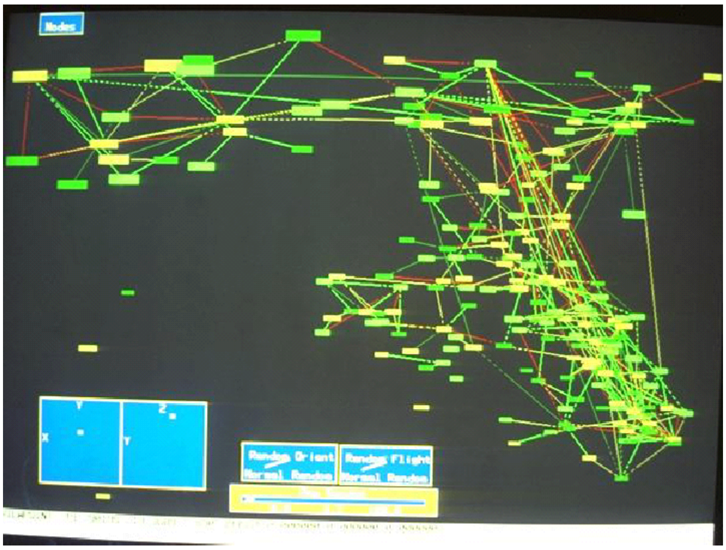


Figure 2.5: The SemNet visualisation showing a complex graph representing a knowledge base. [Image provided by Kim Fairchild and used with kind permission of Keith Andrews, Graz University of Technology.]

thesis.

Graphs are suitable for modeling a variety of information, such as the World Wide Web, a telephone system, distribution networks, and semantic maps. Graphs facilitate the understanding of relations and flows. Networks can have cycles and are often very large, containing many edges.

2.2.1 SemNet

SemNet [Fairchild et al., 1988] is a 3D interface for interacting with semantic networks or knowledge bases. The nodes and the connections between them, represented as edges, are placed in 3D space. The idea behind SemNet is to let the user examine local detail, while still maintaining a global representation of the rest of the knowledge base. SemNet uses "semantic navigation" techniques such as relative/absolute movement and teleportation. It is an interface useful for both users and developers of knowledge bases. Figure 2.5 shows a screenshot of a SemNet interface.

2.2.2 HyperSpace (Narcissus)

Narcissus [Hendley et al., 1995], later renamed HyperSpace [Wood et al., 1995], creates 3D visualisations of hyper structures. For expample, software structures can be visualised by displaying the components' relationships. Hyperspace is an extension for visualising hypertext documents such as web pages. The visualisations in both systems are 3D wire frame graphs, where the pages are represented by nodes and links between them by edges. The wire graph can be seen in figure 2.6. Drawback of the visualisation is that the structure changes its appearance considerably after adding new nodes or links. This inconsistency of the structure makes the creation of a cognitive model more difficult.

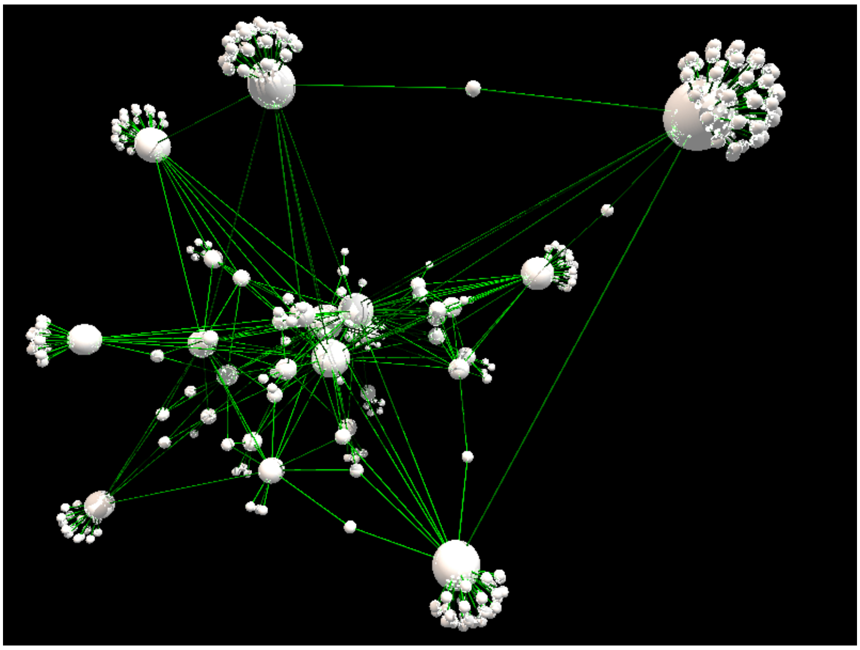


Figure 2.6: The Narcissus visualisation showing a group of web pages that have been explored. All incoming and outgoing links (edges) from the explored pages (nodes) are drawn. [Image extracted from [Wood et al., 1995].]

2.2.3 Harmony Local Map 3D

The Harmony Local Map 3D [Wolf, 1996] is an extension of the Harmony Information Landscape and displays both hierarchical structure and associative hyperlinks. However, the layout is different in Harmony Local Map 3D than in Harmony Information Landscape. The hierarchical structure is presented in the vertical plane mimicking a landscape. Incoming links are placed beneath the plane. Outgoing links are presented above the plane where the selected document is placed. Figure 2.7 demonstrates this. Again, the type of hyperlink to be presented can be chosen by the user (references, inline images, annotations...). It is possible to show where the linked documents come from, by displaying connection lines to the corresponding locations in the hierarchy.

2.2.4 Harmony Local Map

The Harmony Local Map [Schlipflinger, 1998] visualises links from and to a document, yielding a dynamically generated graph. The selected document is positioned in the centre of the screen. All incoming links to the selected document are positioned on the left from the document. All outgoing links are correspondingly located to the right from the selected document. The shape of the map resembles an hour glass lying on its side, see Figure 2.8. Documents are represented by nodes and the links by edges. The extent of the link neighbourhood around the focus document (number of incoming and outgoing hops) can be specified by the user. Additionally, the user can choose the type of link (references, images or annotations) to be represented.

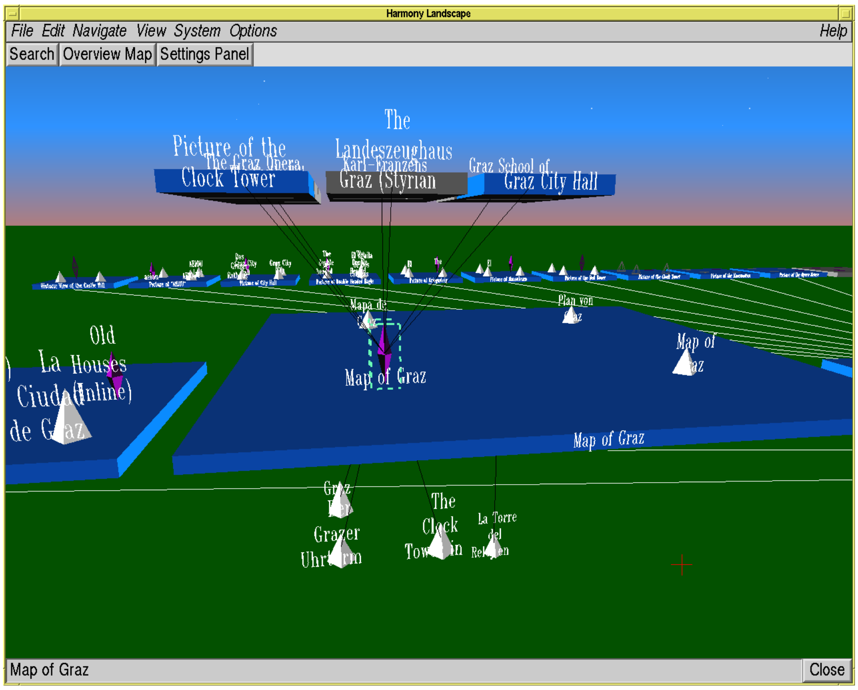


Figure 2.7: Harmony Local Map 3D showing the document "Map of Graz" on the plane and its outgoing and incoming links above and beneath the plane, respectively. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

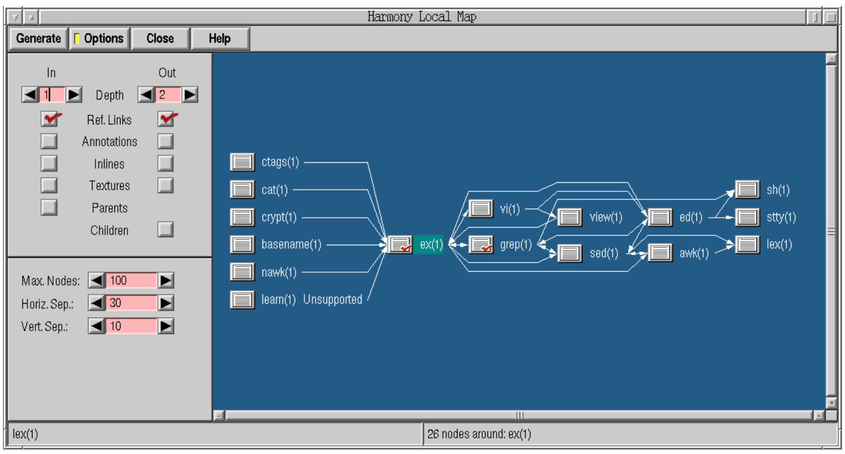


Figure 2.8: Harmony Local Map displaying incoming and outgoing links of the selected highlighted document. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

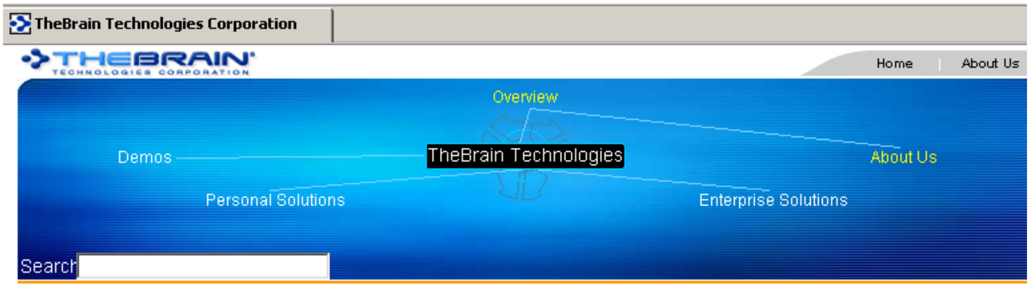


Figure 2.9: The Brain. A simple interactive node-link diagram showing the menu structure of the thebrain.com web site.

2.2.5 The Brain

The Brain [TheBrain.Com, 2002] is a semantic zooming interface for visualisation of web sites. Actually, all general graphs can be visualised. The Brain uses the metaphor of the human brain, where different thoughts trigger other thoughts. The visualisation shows a graph with nodes connected by edges as in figure 2.9. Selecting a node displays that node and updates the graph.

2.3 Visualising Multidimensional Data

Multidimensional information is information with two or more dimensions or attributes. Metadata about types of cars is multidimensional - to describe a car, brand, production year, colour, engine type or fuel consumption are typical. Another example is file attributes such as type, size, author etc.

2.3.1 Parallel Coordinates

Parallel Coordinates [Inselberg and Dimsdale, 1990] visualise relationships in multivariate data. The different attributes are represented as parallel vertical axes and the data items themselves as polylines between the axes. For each dimension in the data, there is one vertical axis. The values on each axis are normalised to display the current scale. Each item is thus represented by a polyline between the axes, see Figure 2.10. Similar items are represented by similar polylines. A quick overview of trends is possible, but occlusion occurs when too many items are displayed. Users can interact with the interface by using sliders, selecting colours, and by reordering the axes manually.

2.3.2 Starfield Displays

Dynamic Queries [Ahlberg et al., 1992] serve to find useful sets in multivariate data. Users explore the database by moving sliders representing some attributes. Both the objects and actions taken are represented visually. Dynamic Queries have many advantages: no command language, easy exploration and discovery of the data, the possibility to see outliers, trends and dependencies and reversibility. The results are presented in a scatterplot and the results of new queries are animated. Starfield displays [Ahlberg and Shneiderman, 1994] are 2D scatterplots used to structure result sets. Every point in the display represents a record in a database. The representation of the points can be from simple coloured

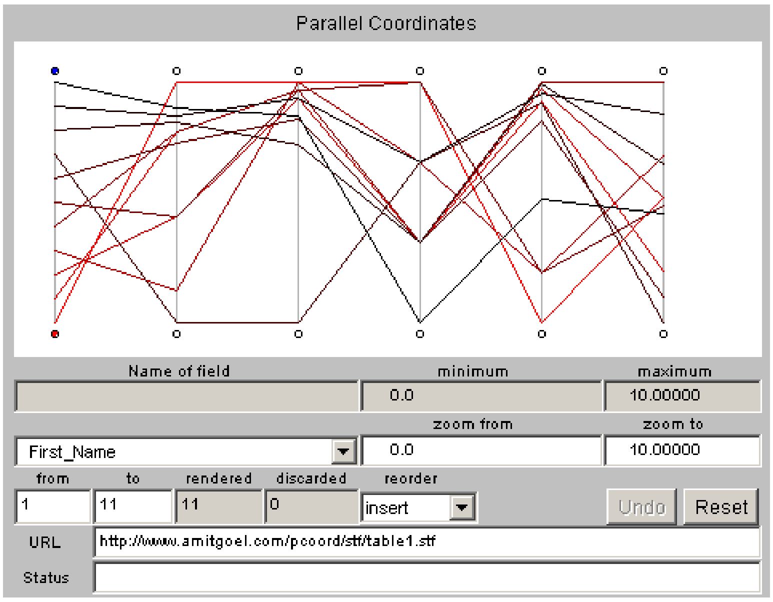


Figure 2.10: Parallel Coordinates displaying six dimensions (vertical axes) and eleven objects (polylines between the axes). [Image shows a screenshot of the applet by Amit Goel [Goel, 1999].]

points or shapes to complex figures and labels. Users can choose the representation they prefer and assign attributes. Users can interact with the display by selecting subsets of points, scaling, and scrolling. In order to prevent clutter, zooming is available. Figure 2.11 shows the Film Finder, which combines dynamic queries with a starfield display.

2.3.3 Table Lens

Table Lens was developed by Rao and Card [1994] and is now commercially available from Inxight [2006b]. The purpose of Table Lens is to represent relations in multivariate data. Table Lens uses the focus + context method within one screen. Data is shown in a spreadsheet-like layout where values in cells are encoded as coloured bars of different lengths. Usually, the whole data set can be seen on the screen without scrolling. Data can be analysed for patterns and correlations visually in the overview. Columns can be rearranged and sorted by the user. In order to explore a portion of the data in more detail, the user simply focuses on this part. The whole context remains visible; only the portion of interest is enlarged. Possibilities to search the data are provided. Another feature, spotlighting, highlights the cells having given values in chosen columns. Users can also change the colours used to represent the data and highlights. Figure 2.12 shows top 100 movies in a Table Lens.

2.3.4 Attribute Explorer

The Attribute Explorer [Tweedie et al., 1994] shows attributes by assigning them to histograms. Each item is present in one of the histograms along each dimension. Users can interact directly with the data: for continuous data by moving the sliders and defining limits. Discrete attributes can be manipulated by pressing the buttons representing them. If the user selects an object in one histogram, the object is highlighted in all other his-

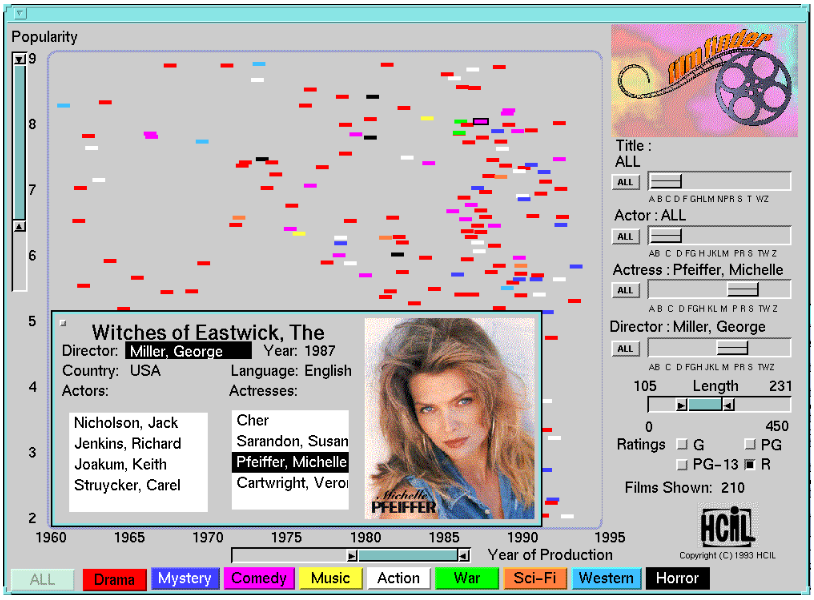


Figure 2.11: The Film Finder combines dynamic queries (sliders on the right) with a starfield display (central part of the screen). Details on the selected film are shown. [Image extracted from [HCIL, 2002]. Copyright ©University of Maryland]

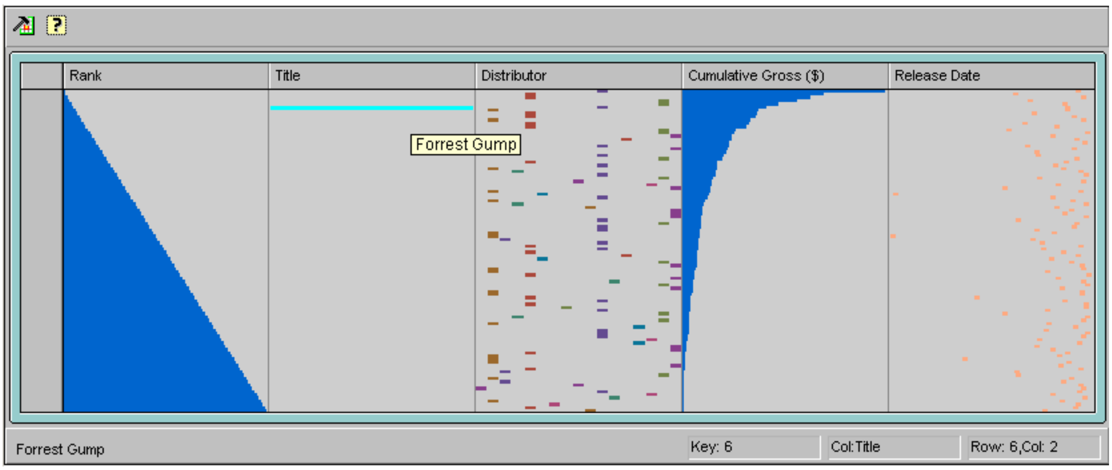


Figure 2.12: The Table Lens displaying the top 100 movies by gross income. The values of the selected lines (here title) are shown in a tooltip and in the status bar at the bottom. [Image extracted from [Inxight, 2006b]. Used with kind permission of InXight Software, Inc.]

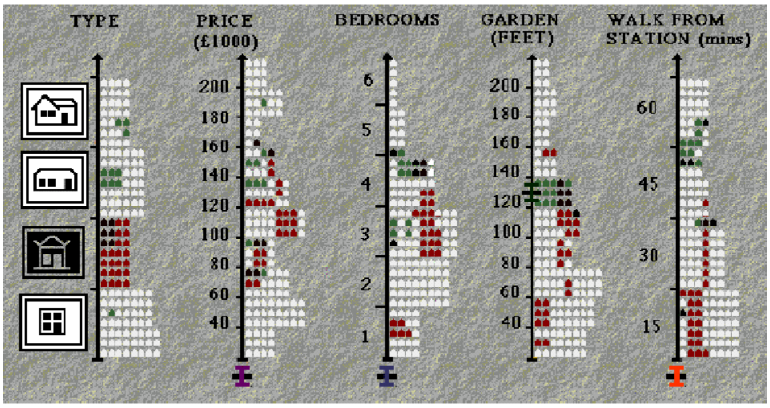


Figure 2.13: The Attribute Explorer showing house information. On each scale, each small rectangle represents an individual house. Houses of the selected type are highlighted in the other histograms. [Image extracted from [Tweedie et al., 1994]. Copyright ©Association of Computing Machinery, Inc.]

tograms. This is shown in Figure 2.13. For an easier handling of the data when examining more than three attributes, the values of attributes in the histogram can be locked. In this case, only the items of interest remain visible.

2.3.5 Envision

Envision [Nowell et al., 1996] offers full-text search possibilities as well as full-content retrieval in a multimedia archive of computer science literature. Search results are displayed as icons arranged in a matrix as well as in a list showing the results with details. In the Graphic View, users can control the visualisation by selecting which attributes to map to the the x- and y- axes, to icon size, shape and colour. The Item Summary can display information such as relevance, type, author, and publication year. Figure 2.14 displays the results of a query.

2.3.6 Search Result Explorer

The result sets from searches are displayed in a scatterplot (starfield display) in the Search Result Explorer [Sabol, 2001]. In the scatterplot, two of the attributes correspond to the x and y axes. Documents are plotted in the scatterplot according to their attributes. By coding other attributes to icon colour and icon size, four different metadata attributes (dimensions) can be plotted in the Search Result Explorer. The colour of icons can be used to represent the age of documents, for example. After selecting a document, its metadata is shown in a table in a separate window. If several documents with similar attributes would be displayed close to one other, a group icon is used instead. To keep orientation when zooming, an overview window is displayed. Figure 2.15 shows the results of a search.

2.3.7 Dust & Magnet

Dust & Magnet [Yi et al., 2005a] uses the metaphor of a magnet attracting particles. Data attributes are represented by magnets, whereas data items are represented by dust particles. Depending on the values of the attributes, the dust particles are more or less attracted to the corresponding magnets. The application shows three windows: the main

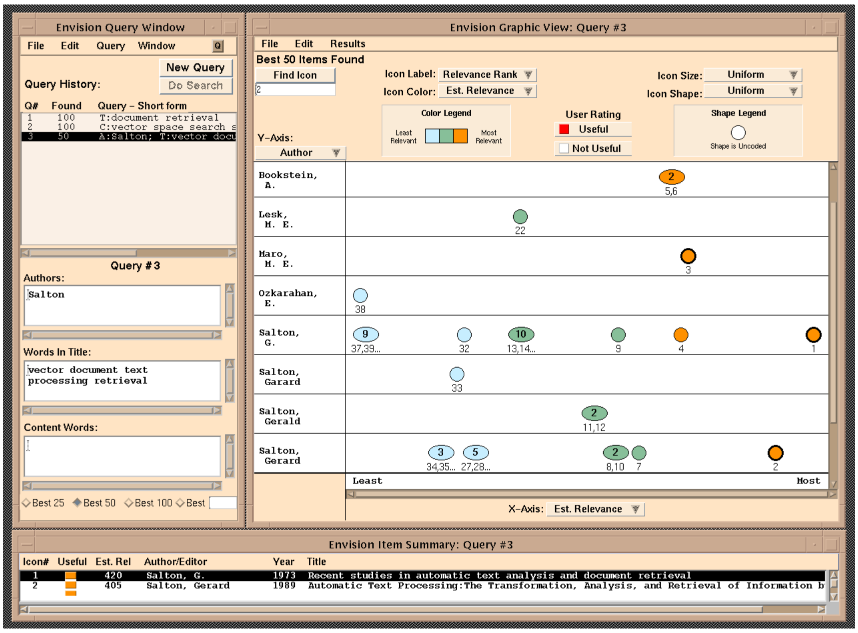


Figure 2.14: Envision displaying the results of a query. The most relevant publications are listed in the item summary. [Image extracted from [Nowell et al., 1996]. Copyright ©Association of Computing Machinery, Inc.]

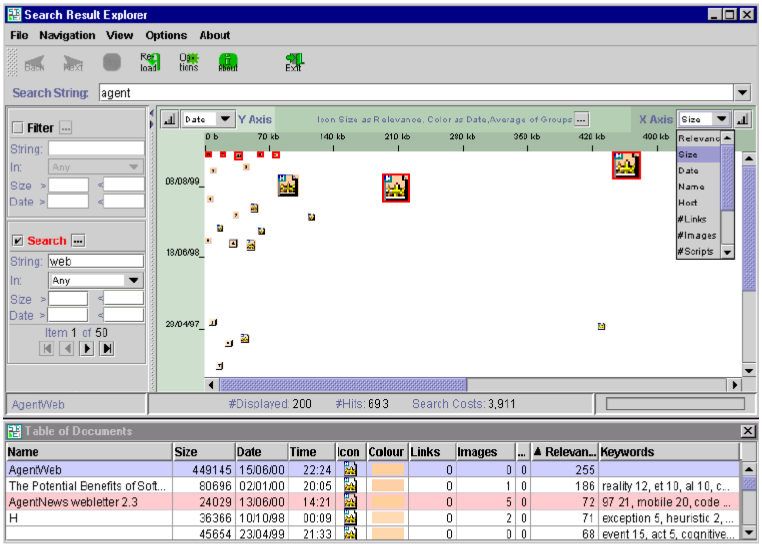


Figure 2.15: Search Result Explorer displaying the results on the search string "agent". Document size and date are plotted on the x and y axes, respectively. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

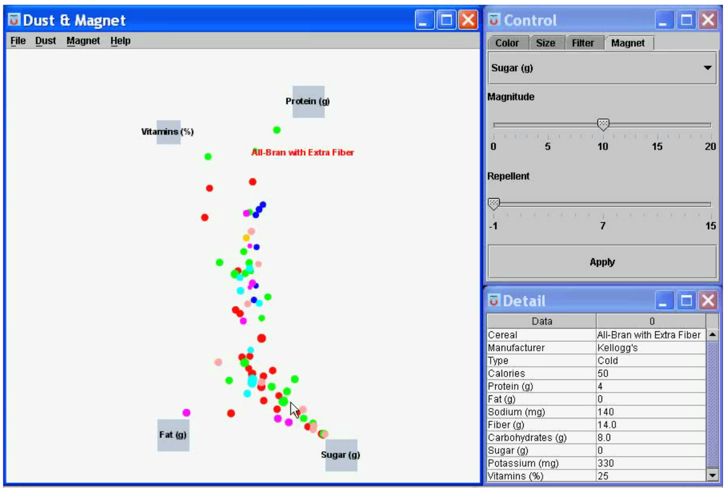


Figure 2.16: Dust & Magnet full view. Control and Detail windows are on the right. Colour and size coding are turned on. The Dust particles are attracted to four attributes Magnets. Details of the selected item are shown in the Detail window. [Image extracted from [Yi et al., 2005b].]

Dust & Magnet window with magnets and dust particles, the control window for user adjustments, and the detail window showing information on selected items. In the main view, the attributes are represented by rectangular magnets, and the data items are represented by circles. The closer a dust particle is to a magnet, the higher the value of that attribute. The magnitude of an magnet (meaning how strong it attracts dust) can be adjusted by the user in the control window. Users can move the magnets around in the main window and thus interactively explore the data. Following features for user interaction are available: zooming and panning, filtering, and colour and size coding. Figure 2.16 shows all three Dust & Magnet views.

2.4 Visualising Content-Based Vector Spaces

Access to large collections of documents has become easier in the recent years through internet and digital libraries. Information visualisation can help users collections of documents even without having to read every document.

In order to compare the similarity of documents, vector space analysis is often used. Typically, lists of unique words which appear in documents are made, then commonly occurring words like “a”, “an”, “the” are erased and different word forms made the same (write, wrote, written). The words are then listed according to appearance frequency, these lists alphabetised and a vector is created. These lists form a so-called term vector, which represents the words contained in each document. The similarity between any two documents is often caclulated as a value between 0 and 1 by calculating the scalar product of the two term vectors. Inter-document similarities can then be used as the basis, upon which to create visualisations of the document space.

2.4.1 BEAD

BEAD [Chalmers and Chitson, 1992] is a system for visualising documents in three dimensions using a force-directed technique (simulated annealing). The layout criterion is

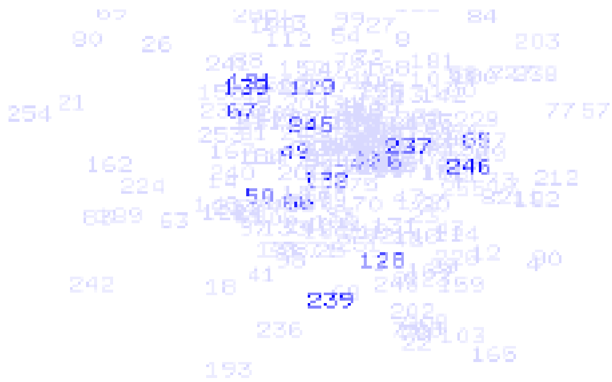


Figure 2.17: BEAD highlighting several documents matching a search. [Image extracted from [Chalmers and Chitson, 1992]. Copyright ©Association of Computing Machinery, Inc.]

the similarity between documents based on word co-occurrence. Thus, similar documents are attracted towards each other and are grouped closer together. Documents which have less in common are laid out farther away from each other. There are different interaction possibilities for the users with the system. If a search is performed the matching results are highlighted. The highlighting gives additional information about how close the documents match the query - the darker the colour the better the match. Zooming to a chosen document in order to examine its neighbours is possible as well. Figure 2.17 shows a screenshot of BEAD.

2.4.2 SPIRE

SPIRE [Hetzler et al., 1998] analyses a collection of text documents and produces two visualisations of the collection. Multidimensional scaling (MDS) from high-dimensional vector space to 2D display coordinates is used. Galaxies display documents based on word similarities in a galaxy of stars. Similar documents are placed near to one another, while documents having less in common are displayed further away from each other. In Themescape, themes within the document spaces are displayed in two-and-half dimensional map. The mountains in Themescape indicate dominant themes and their shape reflects how the thematic information is distributed across documents. The search can be refined using different tools, such as word search, time analyser, and document characterisation.

2.4.3 VisIslands

VisIslands [Andrews et al., 2001] displays search result sets using dynamic thematic clustering based on word frequencies in documents. Documents are first clustered using hierarchical agglomerative clustering. Cluster centroids are positioned using force-directed placement. Finally, the documents in each cluster are placed around the centroid and their positions are fine-tuned. The contributions of all documents to the cluster are summed up and colour-coded as the height of the cluster. The resulting image looks like a geographical map of a group of islands, see Figure 2.18. Documents about similar themes are represented as islands close together, whereas islands farther away from each other represent documents having less in common. The clusters resulting from pre-clustering are listed

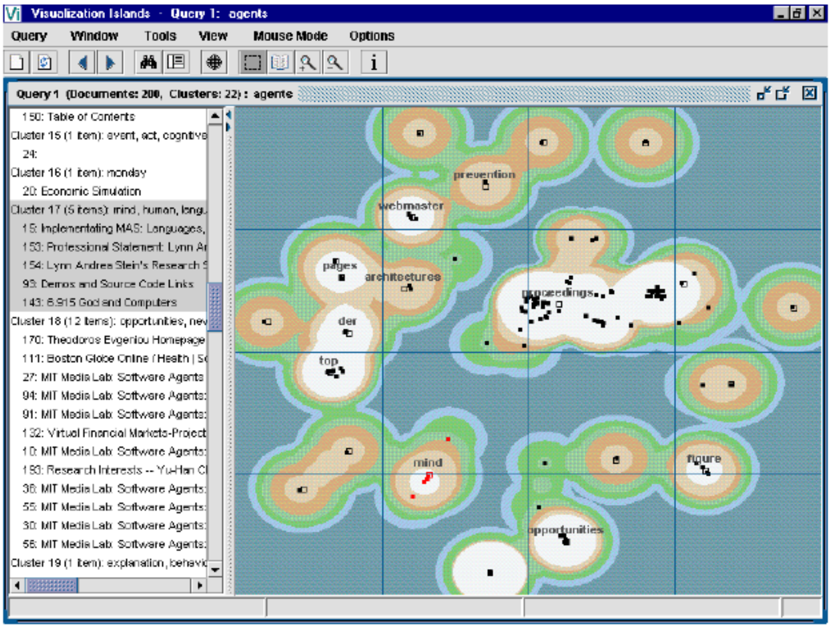


Figure 2.18: VisIslands dispalying 200 documents in 22 clusters. The selected cluster is highlighted in the list as well as in the graphical representation. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

and can be selected to display metadata information on individual documents.

2.5 Visualising Query Spaces

Query spaces visualise the affinity of documents to query terms. In query spaces, users can formulate their search item(s) and see how the results match the query. Boolean searches are usually supported. Different means of representing which documents match the query and to which extent have been developed. The goal of visualisations of query spaces is to help users with the decision which documents match their request and are worth reading.

2.5.1 InfoCrystal

InfoCrystal [Spoerri, 1993a,b] is both an information visualisation technique and a visual query language. InfoCrystal builds upon Venn diagrams where relations are shown as intersections of geometric shapes. In InfoCrystal, the intersections are transformed into icons depending on the relationship. These small icons are called rank icons. Their shape depends on the number of criteria the represented element fulfils. For instance, a circle icon means one criterion is fulfilled, and a triangle shows that three criteria are fulfilled. Depending on the number of criteria, the rank icons are placed inside geometric shapes. Those shapes are called criterion icons. There are n criterion icons, which are the corners of a polygon. For three criteria, the shape is apparently that of a triangle, where each corner represents one criterion. Figure 2.19 displays the result on a query on four search terms. When used as a query language, users simply select/deselect the inner rank icons depending on the query. The rank icons represent the Boolean relations of the outer criterion icons.

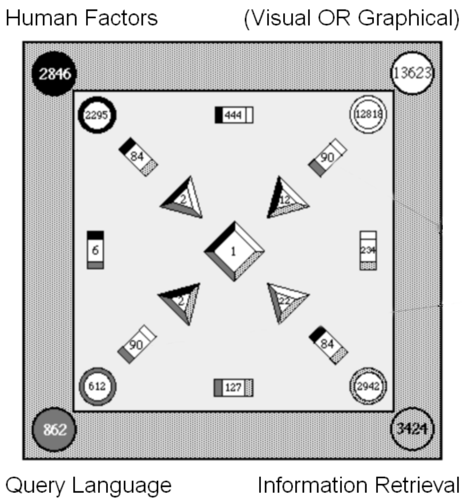


Figure 2.19: InfoCrystal showing the result of a search on four search terms, building the four corners of the large square - the criterion icon. Only one document contains all terms, represented by the square rank icon in the middle and a "1" printed inside. [Image extracted from [Spoerri, 2001]. Copyright ©Association of Computing Machinery, Inc.]

2.5.2 LyberWorld

LyberWorld [Hemmje et al., 1994] is a tool for visualising full text databases. Two visualisations can be used: NavigationCone and RelevanceSphere. NavigationCone is used during search and RelevanceSphere is used to display the found documents. The search terms are placed on the surface of the sphere surface and the documents within the sphere are attracted to them. Figure 2.20 shows a RelevanceSphere and the corresponding documents.

2.5.3 TileBars

The goal of TileBars [Hearst, 1995] is to support the decision making which documents from a collection to view in detail. Query terms are presented in the document structure. Unlike rankings of search engines where the ranking is arbitrary, TileBars represent different attributes and thus the reason for the ranking at a glance. Each search result in the result list is accompanied by a TileBar rectangle showing metadata about the document. The rectangle length corresponds to the document length. The rows correspond to the search terms, one row per search term. The columns represent topical sections inside the document. The shade of the colour of an individual tile gives information about the distribution of the search term in the document, where darker shades mean higher frequency. Clicking on a tile opens the corresponding section of the document. Figure 2.21 shows a search result.

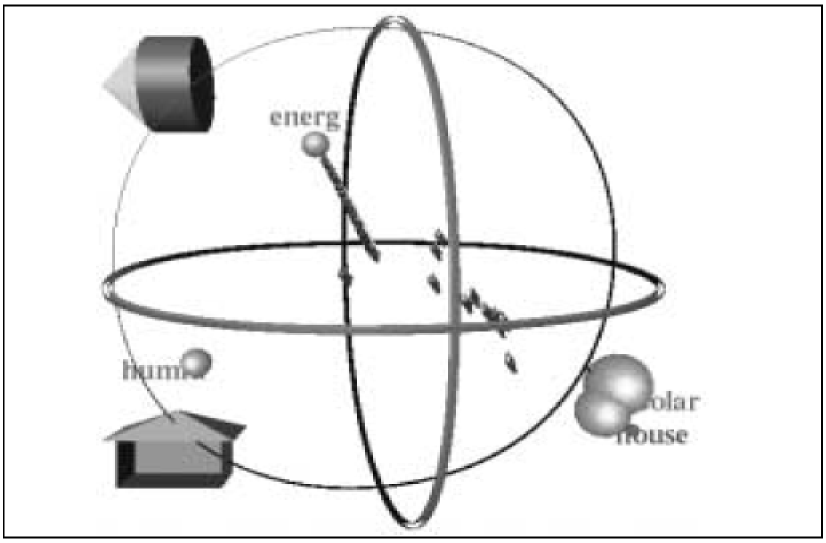


Figure 2.20: The LyberWorld RelevanceSphere dispalying the resulting documents. The search key words are represented by icons on the surface of the sphere. [Image extracted from [Hemmje et al., 1994]. Copyright ©Association of Computing Machinery.]

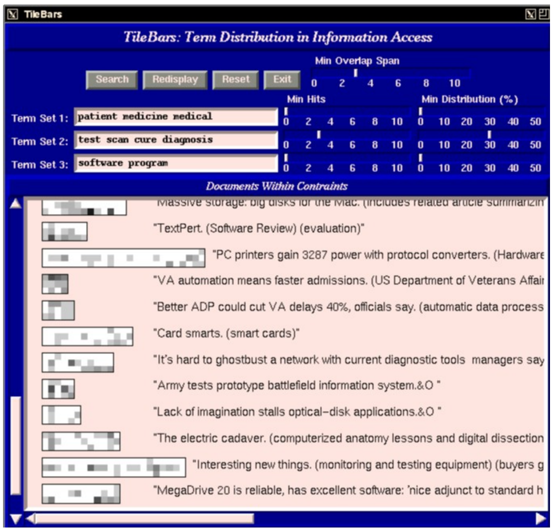


Figure 2.21: TileBars displaying the results of the search on three search term sets, shown in Term Set 1, 2, and 3. [Image extracted from [Hearst, 1995]. Copyright ©Association of Computing Machinery, Inc.]

Chapter 3

Visualising Hierarchies

Hierarchies can be thought of as imposing an ordering in which items are parents or ancestors of other items. Hierarchy examples are family histories and ancestries, file and directory systems on computers, organisation charts, and object-oriented software classes. A tree is a hierarchy in which one node (the root node) is the parent of all others. A tree is a directed, acyclic graph. There are two main representation schemes for trees: node-link representations and space-filling representations. However, other approaches and metaphors are also used.

3.1 Outliners

Tree Views are the most common type of outliners for hierarchy representations. File browsers are usually Tree Views. Probably the most widely spread tree view browser is the Windows Explorer. The structure of the hierarchy is represented simply by a list of directories and files. The different levels are separated by indentation. The elements belonging to their parent element are listed beneath the parent element with a small indentation. In Windows Explorer, they are additionally connected by a thin line. In file browsers, directories and different file types can be given icons for better identification. For large hierarchies, the scrolling problem arises. To explore, users have to open and close directories and subdirectories. Obtaining a general overview and building a mental model of the structure is difficult. Figure 3.1 shows a Windows Explorer file browser.

File Magnitude Viewer [Smith and Clark, 2001] provides iconic directory size information. This Java application first browses the file system. It then produces a pie chart icon for each directory. The size of the pie icon corresponds to the directory size. These icons are placed next to the corresponding directories in an adjusted JTree viewer. Thus, the size of each directory is displayed directly in the tree representation of the hierarchy, see Figure 3.2.

3.1.1 WebTOC

WebTOC [Nation et al., 1997] displays the structure of a hierarchy, primarily web sites, as a table of contents. The table of contents can be either generated by following links or by analysing the directory structure. Using a standard web browser, the WebTOC Viewer displays the information gathered by the WebTOC Parser in the left half of the browser window, showing the site itself in the right half of the screen. The structure is displayed in a TreeView-like manner together with a bar showing different attributes for each branch.

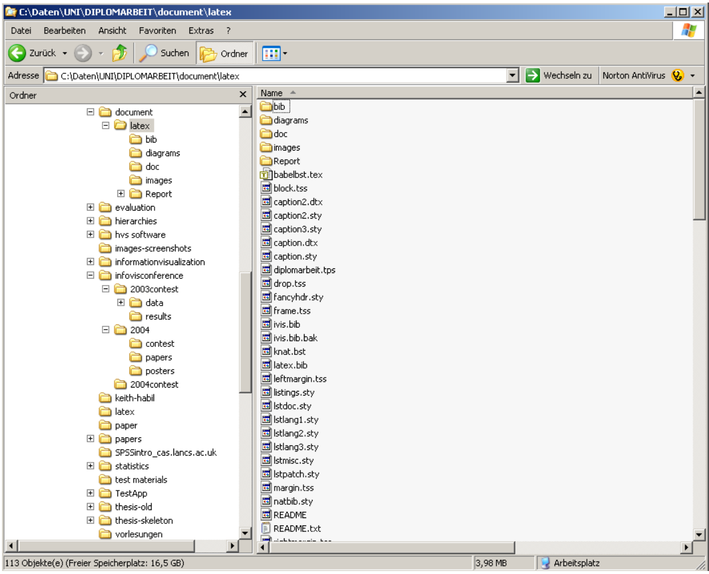


Figure 3.1: The Windows Explorer displaying files and directories in a file system. The Windows Explorer is a classic example of a collapsible tree view.

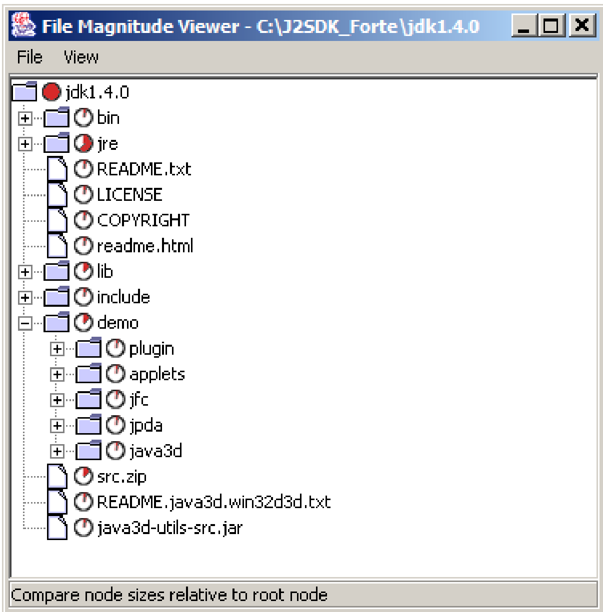


Figure 3.2: File Magnitude Viewer displaying the hierarchy of a file system along with size information of directories. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

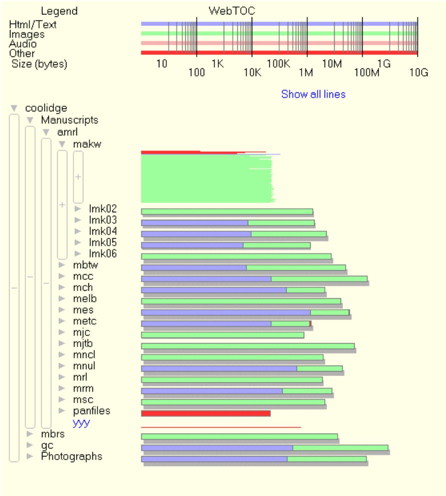


Figure 3.3: WebTOC Viewer showing a hierarchy. The logarithmic size scale and type legend are included. Shadows indicate the number of individual files. [Image extracted from [HCIL, 2002]. Copyright ©University of Maryland]

The length of the bar represents the size of the element. A single bar represents a file, whereas directories are represented by a group of bars to show the overall size. A logarithmic scale is used to show the size in bytes. File types are colour-coded accordingly to their amount inside the bar. For web sites, the file types text, images, audio, and other are used. The number of individual files is represented by the size of the shadow under the bar. Users can interactively change the representation used, for example, to define the bar to show the number of items or show the whole structure as lines only (without titles). Figure 3.3 shows the WebTOC Viewer of a hierarchy.

3.2 Node-Link Representations

3.2.1 Classic Walker Layout

The Classic Walker Layout [Walker, 1990] draws trees in a simple node-link manner, but tries to efficiently utilise the available screen space. The tree width is minimised by a careful positioning of nodes. Smallest subtrees are positioned first and then combined to build larger subtrees. Minimal distances between sibling nodes and subtrees are predefined. The nodes of one level are always positioned on the same horizontal level. By doing so, the individual levels of the hierarchy can be easily depicted. Parent nodes are always centered above their child nodes. Figure 3.4 shows the Walker Layout as included in HVS.

3.2.2 Dendrograms

Dendrograms are a representation of the results of hierarchical clustering. The data is only present in the leaves of the tree. Leaves which fulfil certain similarity issues are clustered together in pairs. Then, similar clusters are paired into new clusters and so on. The dendrograms are usually drawn as binary trees, where the leaves are at the bottom. The clusters belonging together are connected by edges. The Hierarchical Clustering Explorer [Seo and Shneiderman, 2002] was developed for interactive exploration of multi-

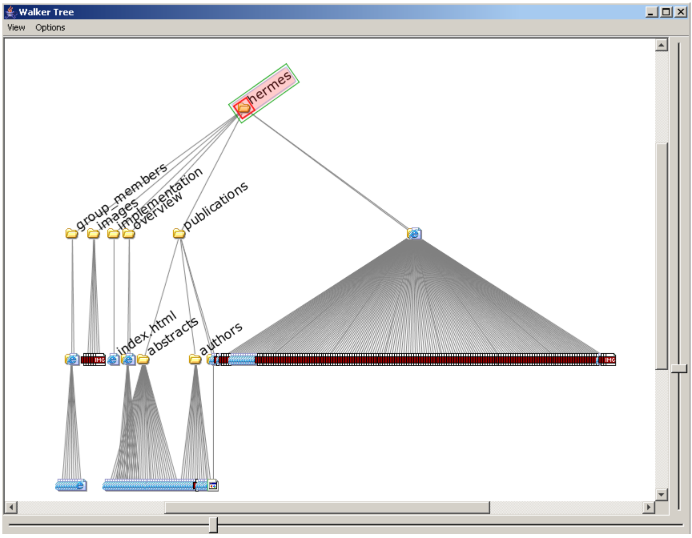


Figure 3.4: Walker Tree Layout in HVS showing a small hierarchy. The four levels can be easily distinguished.

dimensional data, especially in the genome research area. Besides the dendrograms, HCE provides coloured tiles arranged in a mosaic for the researchers to explore. The colours code the expression level of the examined genes. Figure 3.5 shows a dendrogram provided by the HCE.

3.2.3 Cone Trees

Cone Trees [Robertson et al., 1991] display hierarchies in 3D space. The root of the hierarchy is the apex of a cone; all children are placed evenly along the base of the cone. The body of the cone is transparent. This 3D visualisation uses space efficiently. Animations are used to help users keep track of changes. As with all 3D visualisations, occlusion of parts can occur. However, all occluded items can be revealed by rotating the cone. If a node is selected, it is rotated (if necessary) and brought to the front with the path down to the selected node highlighted. The hierarchy can be laid out either vertically from top to bottom or horizontally from left to right. Shadows on the ground plane provide additional visual feedback. The drawbacks of this visualisation are the need for animation and the difficulty of handling large hierarchies (over 1000 nodes). A ConeTree can be seen in Figure 3.6.

3.2.4 File System Navigator

The File System Navigator [Tesler and Strasnick, 1992] uses a landscape metaphor to display hierarchical data; in this case a Unix file system. Directories and files are represented by boxes located in a landscape. The nodes are connected by lines lying on the plane. The height of the plateau-like boxes representing directories corresponds to their size. The smaller boxes representing files are positioned on a grid on their directory's plateau. The height of the file boxes corresponds again to their size. Colour is used to code the age of the files. Icons on the top of the box show the file type. The selected file is highlighted by a "spotlight" directed on it. A separate overview window shows the structure of the entire hierarchy. The File System Navigator is presented in Figure 3.7. File System Navigator is a real file management system where the user can perform operations on the files and direc-

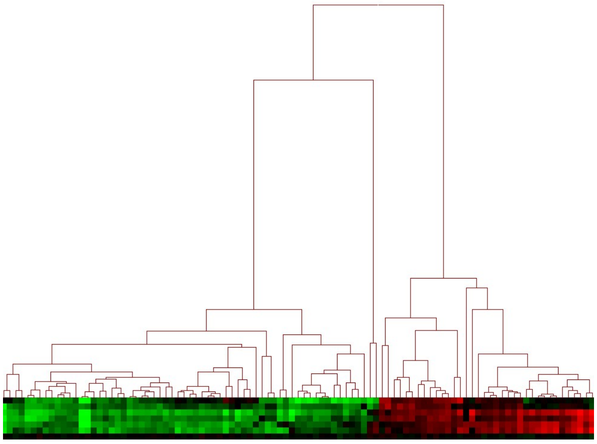


Figure 3.5: A dendrogram in the Hierarchical Clustering Explorer. The colour mosaic representing the expression level of the genes is shown at the bottom. [Image extracted from [HCIL, 2002]. Copyright ©University of Maryland.]

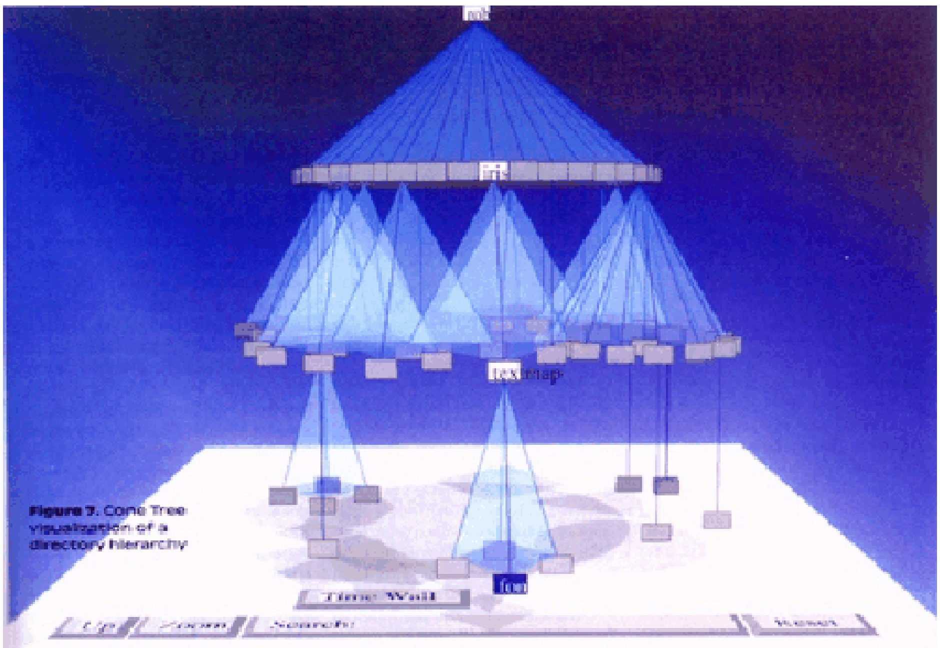


Figure 3.6: A file hierarchy represented as a Cone Tree. Shadows on the ground plane provide an additional visual cue. [Copyright ©Association for Computer Machinery, Inc.]

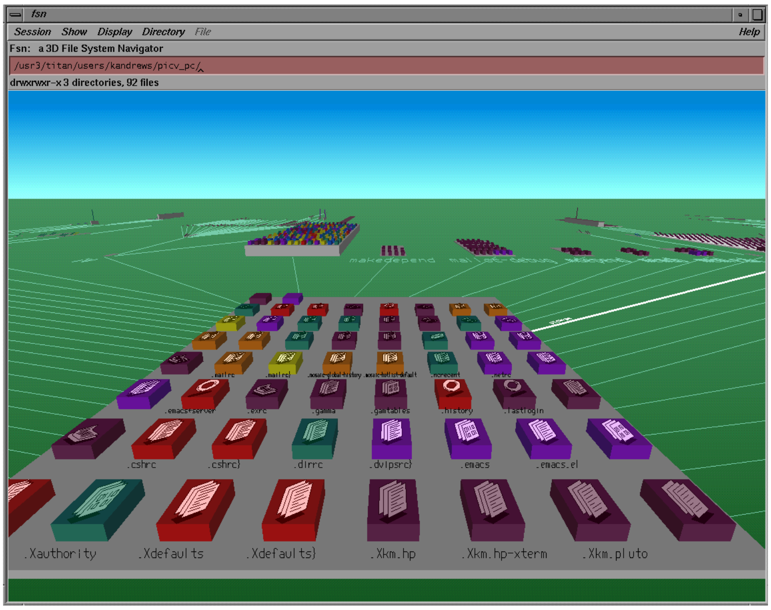


Figure 3.7: The File System Navigator showing a directory in a hierarchy. File types are colour coded. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

tories. Copying and moving files as well as opening or editing them (with the appropriate program) is possible.

3.2.5 Hyperbolic Browser

The Hyperbolic Browser [Lamping et al., 1995] (now StarTree by InXight) is a graph-like visualisation of hierarchies where nodes are connected by edges. The approach is to lay out the hierarchy on the hyperbolic plane and then to map it to a Euclidean presentation plane. With this approach, the entire hierarchy is always displayed. However, parts of the hierarchy close to the periphery are strongly distorted and almost invisible. A distortion area in the middle of the presentation plane acts like a lens, displaying the elements underneath enlarged. The overall structure of the hierarchy is well visible. When a node is selected, the view is animated to bring this node into the distortion area. Thus, the selected node and its neighbours can be examined. However, there are drawbacks to this visualisation. Often, details can only be recognised after zooming in, but when zoomed in too deep, the rest of the hierarchy is distorted and the user loses any sense of orientation. Figere 3.8 displays the Ebay hierarchy in a StarTree.

3.2.6 3D Hyperbolic Browser Layout

In the 3D hyperbolic browser [Munzner, 1997], the hierarchy is laid out in 3D hyperbolic space and then mapped to the unit sphere, see Figure 3.9. The root node is located at the centre of the sphere. The bottom level nodes are positioned on the surface of the sphere. Thus, the overall structure of the hierarchy is visible and can be examined by rotating the sphere. For general graphs, a spanning tree is generated first. By clicking on a node of interest, an animated repositioning of this node to the centre of the sphere is executed. The new position of the node includes having the node's ancestors on the left and the

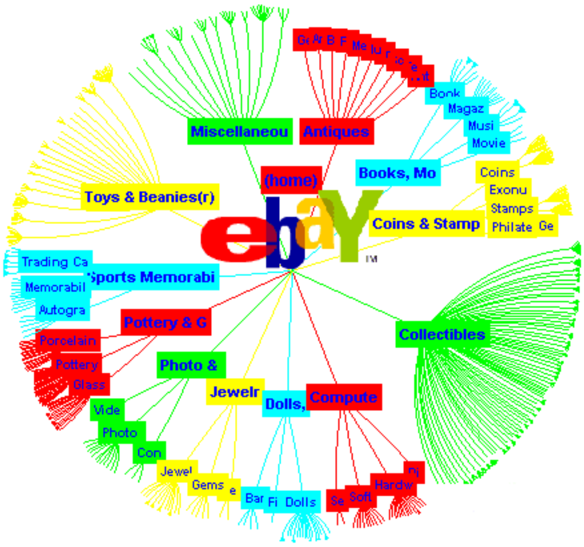


Figure 3.8: The StarTree implementation of the hyperbolic browser showing the Ebay hierarchy. Additionally, colour coding is used for the first level categories. [Image extracted from [Inxight, 2006a]. Used with kind permission of InXight Software, Inc.]

descendants of the right. This positioning helps the user maintain the mental model of the structure and reduces occlusion of the nodes. The 3D hyperbolic browser can handle large hierarchies of over 100 000 edges.

3.2.7 Harmony Information Landscape

The Harmony Information Landscape [Andrews et al., 1996] is a 3D visualisation tool built for the Hyper-G web server contents, similar to the landscape of FSN. Documents are displayed as objects in 3D space, as can be seen in Figure 3.10. The height and colour of the objects represent size and type of the documents, respectively. Besides the documents themselves, hyperlinks to and from the documents are shown as lines. Other link types such as inline images or annotations can be displayed as well. Users can create their own collections of documents and provide the collections with arbitrary objects for recognition (Eiffel Tower could as example represent Paris). Documents returned by a search can be centered in the landscape to show their position relative to other documents. Users can “fly” around in order to explore the structure.

3.2.8 MagicEye

The MagicEye View [Burger, 1999] is a focus-plus-context method for visualisation of large hierarchies. The hierarchy is first laid out in 2D and then projected onto the surface of a hemisphere. In order to help distinguish the individual levels in the hierarchy, a visual clue of alternating dark and light stripes is used, see Figure 3.11. The focused parts of the hierarchy are enlarged, whereas the rest of the hierarchy is scaled down. Thus, the whole structure of the hierarchy remains visible while the area of interest can be examined in more detail. The changes in focus are animated smoothly and thus the users do not loose their sense of orientation.

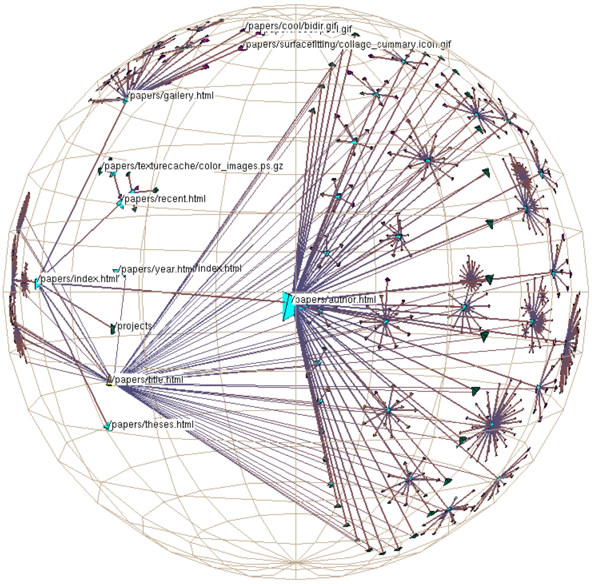


Figure 3.9: The H3 3D hyperbolic browser. Part of the structure of the Stanford graphics group web site is shown. [Image extracted from [Munzner, 2002]. Used with kind permission of Tamara Munzer. Copyright ©1997 IEEE]

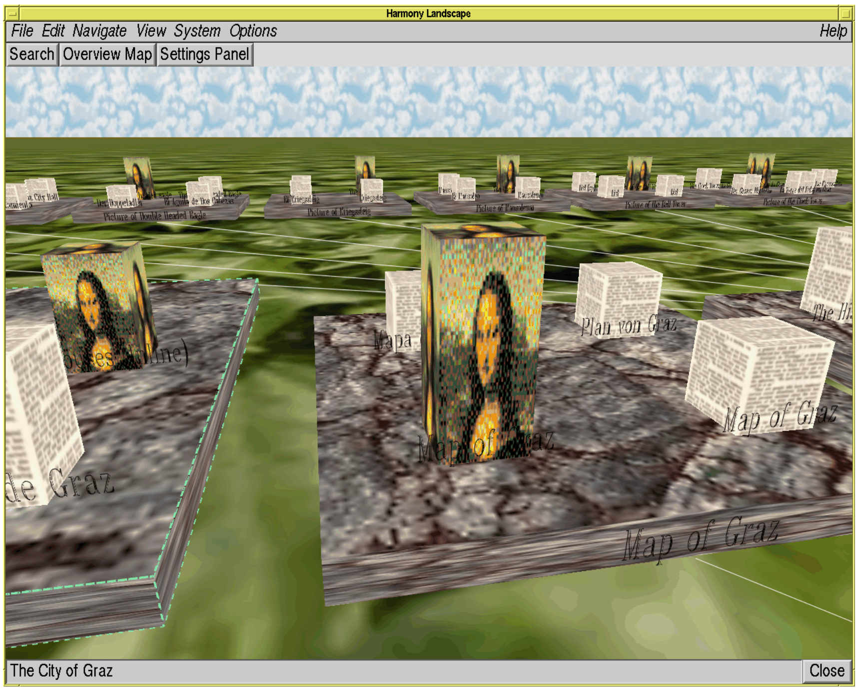


Figure 3.10: The Harmony Information Landscape displaying a part of a hierarchy. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

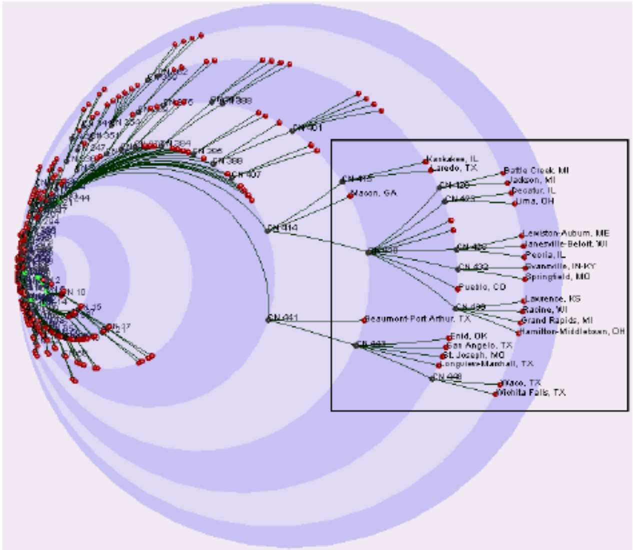


Figure 3.11: MagicEye View showing a hierarchy. The framed part of the hierarchy is enlarged. [Image extracted from [Kreuseler et al., 2000]. Copyright ©IEEE.]

3.2.9 SpaceTree

SpaceTree [Plaisant et al., 2002] is a visualisation technique for hierarchies which builds upon the classic node-link tree layout. The tree branches are laid out to best use the available screen space. Only the nodes which can be recognized easily are fully presented. Branches of the hierarchy which do not fit on the screen are represented by small triangles. The height of the triangle corresponds to the depth of the branch it represents. Its width corresponds to the width of the branch it represents. The colour intensity of the triangle corresponds to the density (number of nodes) of the branch - the darker, the denser. Users can navigate through the hierarchy by clicking on nodes or using the arrow keys to step through the nodes. Users can also choose the tree orientation: top-to-bottom, left-to-right or bottom-to-top. Searching and filtering is also integrated. The locations of the search results are highlighted in the hierarchy. Figure 3.12 shows a sample SpaceTree.

3.2.10 WALRUS

Walrus [Hughes et al., 2004] is another 3D hyperbolic visualisation system for directed graphs and hierarchies. The layout of the graph is calculated based on the user supplied spanning tree. 3D hyperbolic geometry is used yielding a fisheye distortion. The hierarchies are displayed inside a sphere which is the Euclidean projection of the hyperbolic 3D space. The magnification near the centre of the sphere is greatest, decreasing towards the boundary. The overall structure of the hierarchy as well as details in the centre is shown at the same time. The user can examine the parts of interest by dragging them near the centre of the sphere. Figure 3.13 shows the Walrus visualisation.

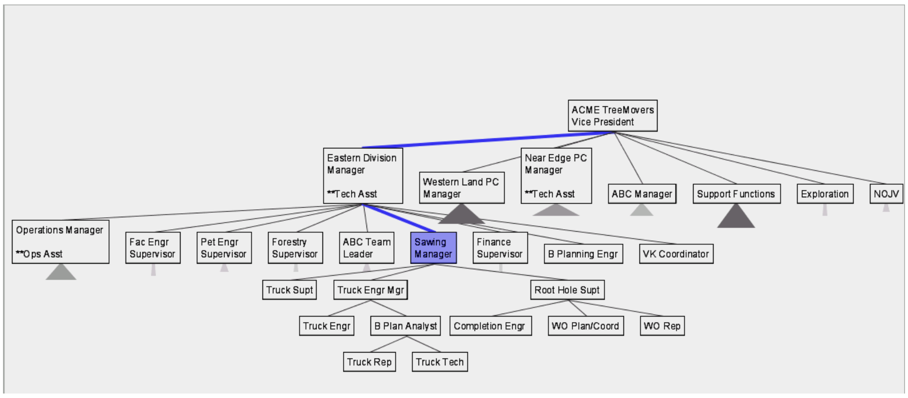


Figure 3.12: SpaceTree visualisation showing a sample tree. The view tries to show all child nodes of the selected node (highlighted). Subtrees which would not fit on the screen are reduced to triangles. [Image extracted from [Grosjean et al., 2002]. Used with kind permission of Catherine Plaisant. Copyright ©University of Maryland.]

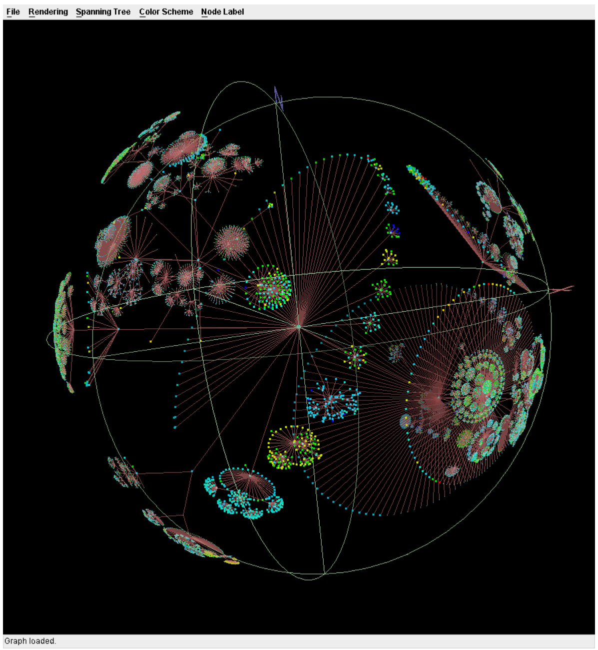


Figure 3.13: The WALRUS 3D hyperbolic visualisation displaying a directory tree. [Image extracted from [Hyun, 2005]. Used with permission. Copyright ©2005 The Regents of the University of California. All Rights Reserved]

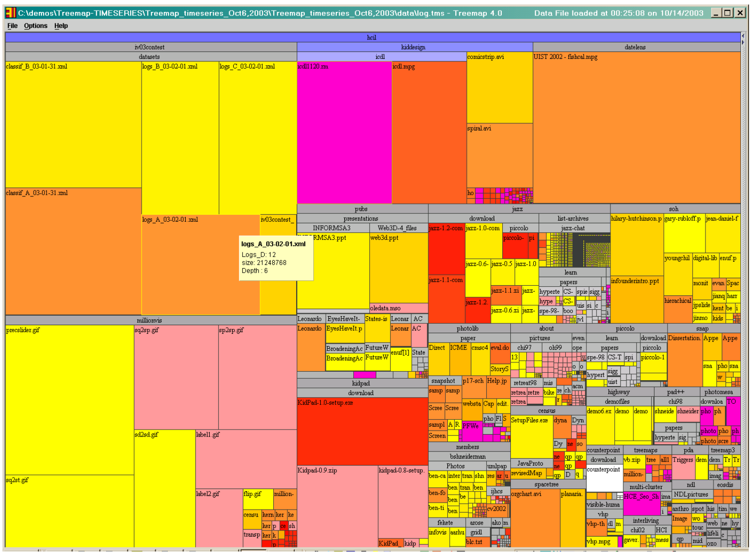


Figure 3.14: The TreeMap showing a file hierarchy. Colour coding of file types is turned on. [Image extracted from [HCIL, 2002]. Copyright ©University of Maryland]

3.3 Space-Filling Representations

3.3.1 TreeMaps

TreeMaps [Johnson and Shneiderman, 1991] represent hierarchical structures using nested rectangles, see Figure 3.14. TreeMaps are constructed by recursive subdivision of the rectangles. The outermost rectangle represents the highest level in the hierarchy. Sub-nodes are thus represented by rectangles included in other rectangles. The size of each rectangle reflects the size of the node it represents. In a file system, rectangles can represent both directories and files. Colour coding of different file types can be used as well. TreeMaps make the full use of screen space. However, this advantage can become a disadvantage for large hierarchies. Additional zooming is needed for the case where there are many small rectangles. Another disadvantage are rectangles which become narrow strips.

3.3.2 Market Map

SmartMoney.com provides a Map of the Market [Wattenberg, 1999] on the web reflecting stock prices on the NY Stock Exchange. The visualisation is a squarified tree map, which avoids very narrow rectangles, as can be seen in Figure 3.15. Colour coding is used to represent the changes in percent. The users can choose either a red/green or a blue/yellow colour coding. Bright red (blue) means great losses, whereas bright green (yellow) means gains. The Market Map is a three level hierarchy. The map is divided into large rectangles representing coarse categories such as Energy or Technology. Smaller rectangles represent the individual companies, according to size. A small tool tip appears when the mouse hovers over the map showing the name and exact percentage change. By clicking on the individual rectangles, their details are shown.

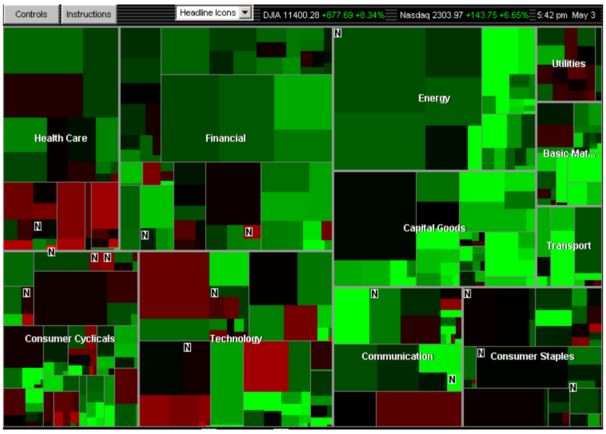


Figure 3.15: The MarketMap showing market changes over 26 weeks. Red/green colouring is chosen, with red indicating falling values and green rising values. [Image extracted from [SmartMoney, 2002].]

3.3.3 SequoiaView

SequoiaView [SequoiaView, 2005] is a disk browsing tool showing disk usage. SequoiaView implements the treemap visualisation with an additional feature called cushion treemap [van Wijk and van de Wetering, 1999]. Cushion treemaps use shading for each rectangle in order to better reveal the structure of the hierarchy. Users can change the settings of the shading in order to highlight either the overall structure or deeper levels of the hierarchy. A colour coding for different file types is provided and can be modified by the users as well. Sorting on name, size, date (creation or modification) or a combination of these is available. Users can choose between the slice-and-dice representation and the squarified cushions representation. Figure 3.16 shows a screenshot of a file system represented in SequoiaView.

3.3.4 Icicle Plots

Icicle Plots [Kruskal and Landwehr, 1983] were developed for representing hierarchical clustering. In an icicle plot, the topmost horizontal line holds all objects to be clustered. According to a given similarity function, the objects are clustered together. At the end of the clustering, so-called singleton clusters (holding only two objects) remain. The clusters are built vertically, yielding vertical “icicles” of different lengths, hence the name. For representing trees, the icicle plots are drawn from top to bottom, where each level corresponds to one level of the hierarchy. The topmost bar represents the root directory. The next level of the hierarchy is represented by individual bars for the nodes. The width of the bas corresponds to the size of the nodes. As the directories of the displayed hierarchy might be of different depths, the resulting drawing is similar to icicles of different lengths. In their experiment, [Barlow and Neville, 2001] compared different representations of decision trees, see Figure 3.17 and Chapter 6.

3.3.5 XDU

XDU [Dykstra, 1991] is a visualisation system for the X window system to show disk usage in a Unix file system. The look of XDU is similar to a TreeMap, but it is laid out in a

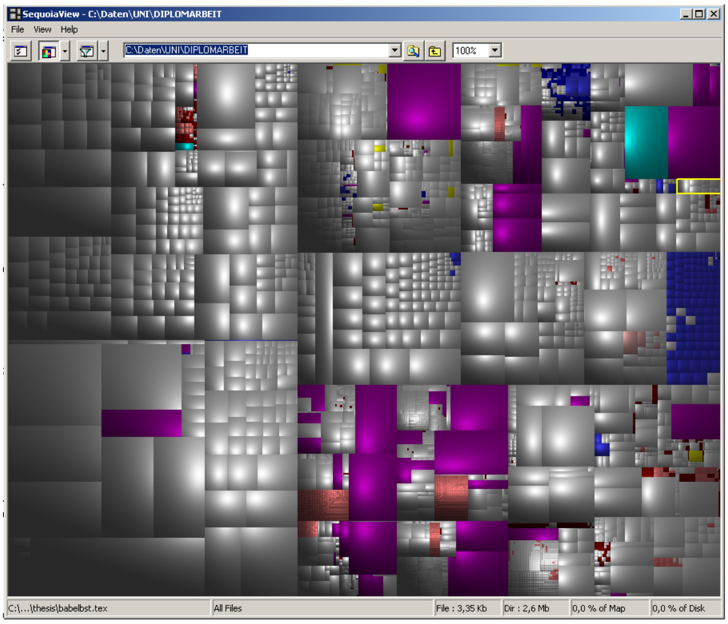


Figure 3.16: SequoiaView showing a part of a file system as squarified cushions. File colour coding is turned on. [Demo available at [SequoiaView, 2005]]

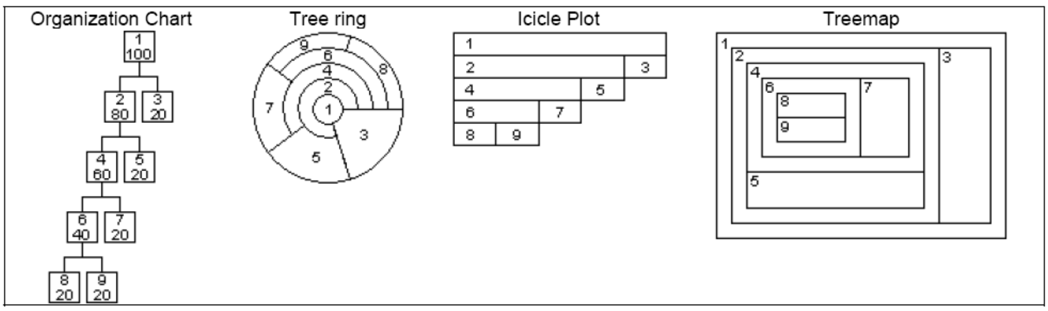


Figure 3.17: Icicle Plot and other representations of a decision tree. [Image extracted from [Barlow and Neville, 2001]]

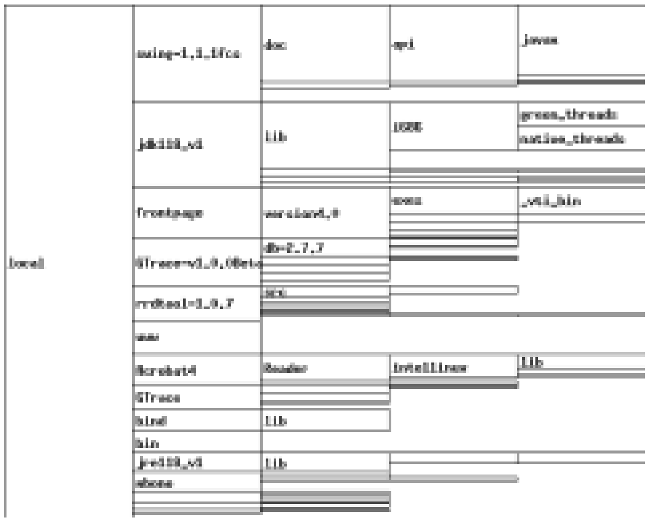


Figure 3.18: A file system represented in XDU. [Image extracted from [Dykstra, 1991]]

different manner. Directories and files are represented by rectangles, showing the structure of the hierarchy from left to right. The currently selected directory is shown on the left within the window. The child nodes of this directory are positioned to the right. The height of the child nodes corresponds to their size. Each level of the hierarchy occupies one column. Figure 3.18 shows a part of a file system in XDU.

3.3.6 Information Pyramids

Information Pyramids ([Andrews et al., 1997; Andrews, 2002]) use a pyramid metaphor to display directories and their contents. The entire structure is visible at a glance. The root directory is represented by a large pedestal and its contents are placed on top of it. Subdirectories are represented by smaller pedestals, whereas files are represented by coloured blocks. The height of the "pyramid" grows with the depth of the hierarchy. The colours of the individual blocks correspond to the file types and can be assigned by the user. The file or directory names appear on the front of each blocks. Thumbnails of the file contents are shown on top of the file blocks (e.g. the first page of the document or the thumbnail of the image), where possible. Figure 3.19 shows a hierarchy represented by Information Pyramids.

3.3.7 Information Slices

The idea behind Information Slices [Andrews and Heidegger, 1998] is to lay out a hierarchy on a semi-acrular disc in 2D space. The interface shows two discs on the screen. For very large hierarchies, a chain of cascading discs is used, see Figure 3.20. Usually, the left disc presents the whole hierarchy and the right one is used to display details of the selected directory. Each disc can represent several levels of the hierarchy, the number of levels can be configured by the user. When a directory is selected, its contents are displayed in the right hand side disc. In deep hierarchies, selecting a directory in the right disc makes the left disc shrink and move to the left upper corner of the left half of the screen. After that, the right disc slides to the left and the selected directory is presented in the right disc.

Figure 3.19: Information Pyramids displaying a hierarchy in HVS. File colour coding is turned on.

The size of portions allocated to an element at each level corresponds to the element's size. Users can choose the ordering of the files to be by size, name, or date.

3.3.8 SunBurst

SunBurst [Stasko and Zhang, 2000] is a radial space-filling visualisation technique for hierarchies. The levels of the hierarchy are represented by concentric circles with the root at the centre. Each directory and file is given an arc corresponding to its size. Elements included in a directory are drawn within the directory's arc, see Figure 3.21. The distinction of small elements deep in the hierarchy is difficult, leading to three ways of fanning out the contents of a selected directory. In the Angular Detail method, after selecting an item, the whole hierarchy shrinks and moves to a corner of the screen. Thus, space is provided for the selected item to be extended out of the hierarchy and represented by a larger arc, allowing sub-items to be examined. In the Details Outside method, the selection of an item makes the hierarchy shrink, but it stays in the centre. The selected item is then expanded and represented by a 360° ring around the hierarchy. Its sub-items are represented by arcs corresponding to their size. The Details Inside method makes the selected item appear in the centre of the view, pushing the hierarchy outwards. The selected item and its children are drawn from the centre outwards. In all three views, the selected item is highlighted for better orientation and the transitions are animated. Users can colour code the files according to their type or modification date. A tooltip displays an item's name and path.

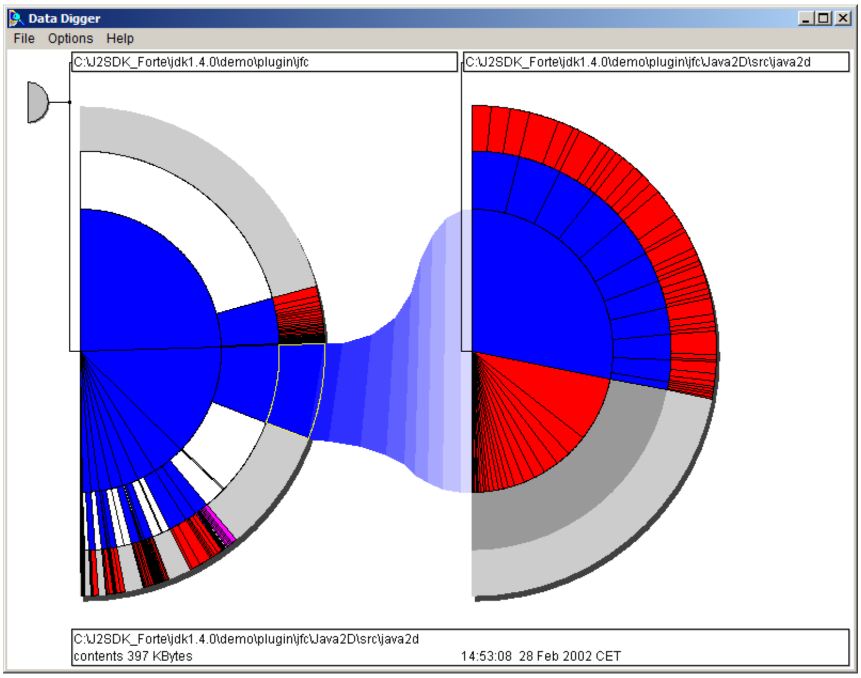


Figure 3.20: Information Slices displaying a deep hierarchy. The uppermost levels of the hierarchy have been reduced to a small disc in the upper left part of the screen. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

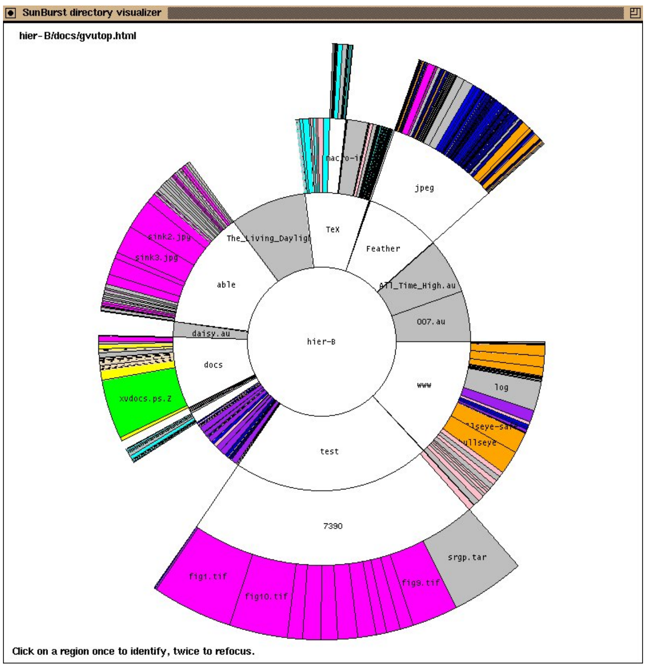


Figure 3.21: SunBurst dispalying a file system. Colour coding of file types is turned on. [Image extracted from [Stasko et al., 2000].]

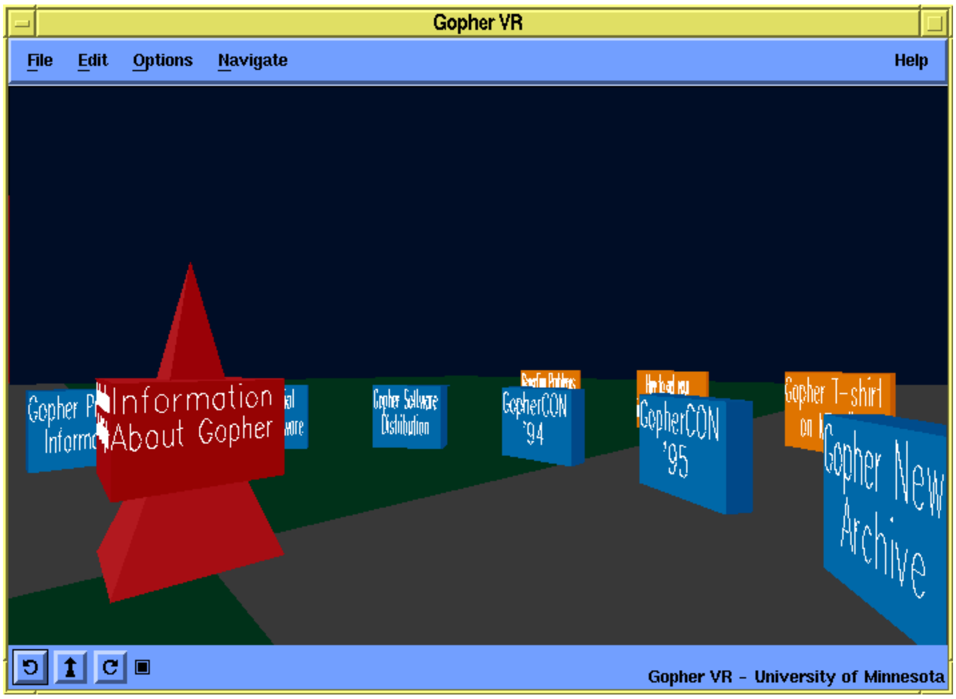


Figure 3.22: One level of a hierarchy represented by GopherVR. The red cone-shaped element is the special object which leads back to the previous level. [Image used with kind permission of Keith Andrews, Graz University of Technology.]

3.4 Other Approaches

3.4.1 GopherVR

GopherVR [McCahill and Erickson, 1995] is hierarchically structured information system. It visualises the contents of a Gopher server. It shows the elements at one level of the hierarchy in a 3D space, represented by different 3D objects. The object type and colour are used to distinguish between different document types. The element's name is written on the object, see Figure 3.22. The objects of one hierarchy level are arranged in a circle around a special middle object. This special object is the entrance to the parent level of the current level. Search results are arranged in spirals, where the most relevant results are located near the centre of the spiral. Users can open the documents; for that purpose the corresponding helper applications are used. For navigation, users can walk through or fly over the scene. The major drawback of this system is that only one level of the hierarchy is displayed at a time, yielding a "focus without context" representation. Therefore, no overview of the overall structure is possible.

3.4.2 CHEOPS

CHEOPS [Beaudoin et al., 1996] uses overlapping triangles to compactly visualise hierarchies. Each node is represented by a triangle. Those triangles are overlapped in order to save space. The selection of a node highlights and brings to the fore the parents and children of this node. For better orientation, the colours of the triangles are used as visual cues. Different colours are used for the selected node, child nodes, and unused nodes. In order to navigate through the hierarchy, users can click on the triangles. Navigation Buttons

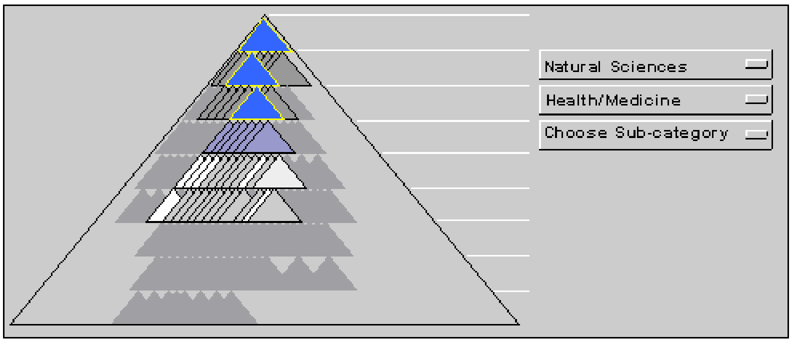


Figure 3.23: Cheops visualisation displaying a large hierarchy. Choice boxes for the first three levels of the hierarchy are shown on the right. [Image extracted from [Beaudoin et al., 2002]. Copyright ©IEEE.]

allow the user to step through the hierarchy in four directions - left, right, up, or down. An additional button selects the root node. Choice Boxes can also be used for top-down navigation. One choice box per level allows the user to choose the nodes in this level by name, as can be seen in Figure 3.23. This selection also displays the selected node and the corresponding branch.

3.4.3 Botanical

Botanical visualisation [Kleiberg et al., 2001] is a 3D visualisation technique. It displays a hierarchy as a real tree, see Figure 3.24. The hierarchy is first converted into its botanical model, then the results are adjusted (contract long branches, show leaf nodes as fruits, etc.). Directories are represented by branches and on the lowest level by spheres. Files are represented by cones ("fruit") on the surface of the spheres. The size and colour of the individual elements are cues for the user. The size of the cones depends on the size of the file. Similarly, the size of the spheres depends on the size of all contained files. The thickness of a branch reflects the overall size of the directory.

3.4.4 BeamTrees

BeamTrees [van Ham and van Wijk, 2002] are created using a variation of the treemap algorithm. In this case, overlap rather than containment (as in treemap) is used to depict structure, see Figure 3.25. Beginning with a treemap representation, the rectangles are scaled down to reveal the higher level rectangles. This will result in leaf nodes overlapping their parent nodes, thus revealing the hierarchy structure. The size of the rectangles, though reduced, still corresponds to the size of the elements they represent. In the 2D version of BeamTrees, additional visual cues such as shading and cast shadows help depicting the structure. The 3D version represents the nodes using cylinders rather than rectangles. Leaf nodes are represented as coloured rings stacked on the cylinders. In order to better examine the structure, users can rotate the tree.

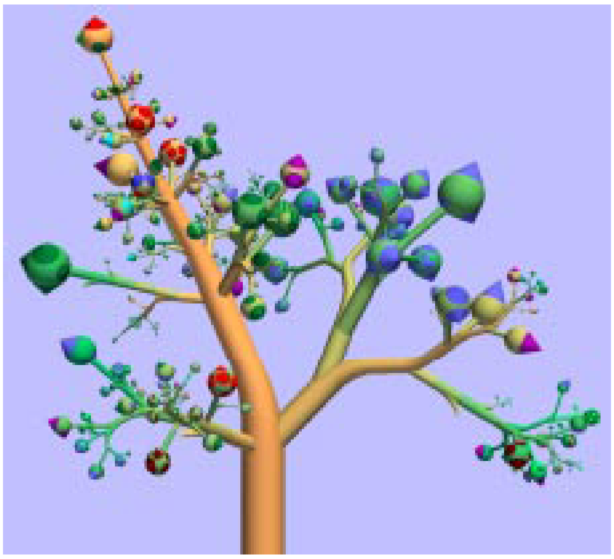


Figure 3.24: The Botanical visualisation displaying a file system hierarchy. File types are colour coded. [Image extracted from [Kleiberg et al., 2001]. Used with kind permission of Jack van Wijk.]

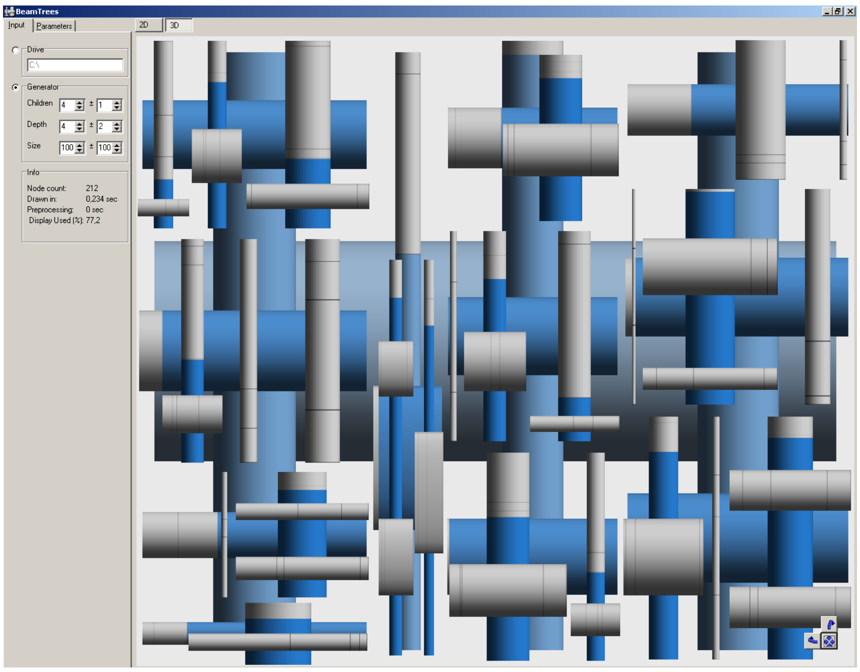


Figure 3.25: The BeamTrees visualisation. [Screenshot shows an example hierarchy as provided in the BeamTree demo [van Ham and van Wijk, 2002].]

Chapter 4

The Hierarchical Visualisation System (HVS)

The Hierarchical Visualisation System (HVS) is a toolkit for the visualisation of hierarchies, developed by Putz [Putz, 2005]. HVS has advantages for both users and developers. The HVS framework can be easily extended by new visualisations. As the management of hierarchical data is handled in the framework, developers of new visualisations can concentrate on the development of the visualisation itself. Currently, the following visualisations are included in the framework: Information Pyramids, Hyperbolic, Walker Tree, Tree View, Tree Map, Magic Eye, Cone Trees, Sunburst and Info Lens. These visualisations and their realisation in HVS are described in Section 4.3.

Users can choose from different visualisations, whereby these are synchronised. Data can be explored in different visualisations at one time in one application. Several data manipulation facilities are provided as well as searching and filtering.

4.1 Hierarchical Data Model

The Hierarchical Visualisation System was developed for the visualisation of hierarchies. In HVS, a hierarchy is defined as a directed acyclic graph. The information about the structure is stored for each node as a parent-child relationship. Information about the content of the leaf nodes (e.g. files) is stored as an attributes set.

Currently, HVS reads hierarchical data from two sources - either from a file system or from an XML file. New modules for reading other sources can be easily developed and plugged into the HVS framework. When the underlying hierarchy is the local file system, HVS transforms the hierarchical structure into an XML file containing all the attributes of the nodes and leaves as well as thumbnails of the files. This XML file can be saved by the user for later use. If the hierarchy changes, HVS only reads the changes rather than the whole hierarchy once again. The TreeML format developed for the InfoVis03 contest [Fekete and Plaisant, 2003b] can be opened in HVS as well.

Users can decide whether the data modifications they perform are done directly in the source data or not. The possible data modifications include insertion of new nodes, removal, and renaming of nodes. Colour coding and icons are used to represent directories and different types of files.

4.2 Synchronisation of Visualisations

HVS has a synrchonisation mechanism, which can keep various visualisations of the same hierarchy in synch with one another.
The following user actions are synchronised between the visualisations:

- node selection (several nodes can be selected).
- node expansion.
- navigation, such as scrolling, maximising and focusing (only one node).
- tree operations, such as renaming or removing nodes.
- search results.
- filtering.
- settings (colours, fonts).

Synchronisation for a particular visualisation can be switched on and off by the user (Synchronised/Independent View).

The option Overview shows the whole hierarchy, whereas Detail View only shows the selected nodes and their path from the root. The option Hide Documents shows only the structure of the hierarchy, hiding the documents, whereas the Show Documents option shows the complete hierarchy.

A search and filter facility is provided in HVS also on document attributes. After a search is performed, the resulting nodes are highlighted in the visualisation (see Figure 4.1). An optional search result list showing all results and their attributes can then be opened. Filtering is possible on name or type. Users can choose node sorting either in descending or ascending order by name, size, or type (see Figure 4.2).

The browsers allow typical operations on the hierarchy. Nodes can be removed, renamed, and new nodes can be inserted. Users can decide whether these changes apply to the underlying hierarchy, thus changing the "real" data, as well.

A properties panel is available in each visualisation, as shown in Figure 4.3. It lists the names and attributes of all selected nodes. Users can choose which attributes are listed in the properties panel. The list is ordered by default alphabetically by node name. However, the ordering can be changed for any attributes.

In settings, users can choose fonts or colour-coding for different file types (see Figure 4.4). Depending on the visualisation, directories and files can be represented by icons or coloured dots. The colour-coding can be saved and used as a default for all visualisations.

4.3 Currently Available Visualisations in HVS

Twelve visualisations are currently available in HVS. The next sections present these visualisations.

4.3.1 Information Pyramids

In Information Pyramids pedestals are used to represent directories and blocks to represent files. The pedestals and blocks sit atop of each other, thus depicting the structure.

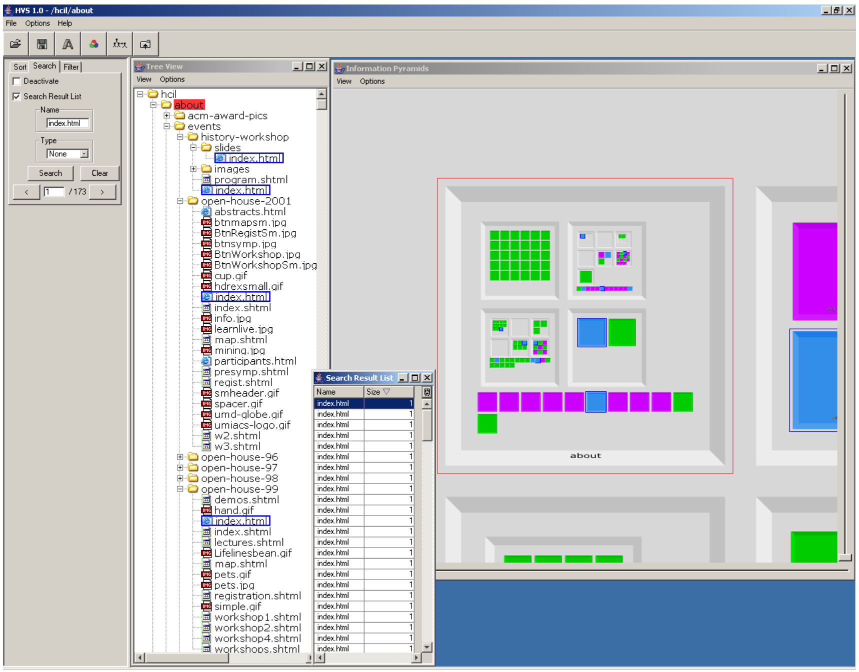


Figure 4.1: The search facility in HVS. A search for “index.html” highlights all occurrences of files with this name in all open visualisations.

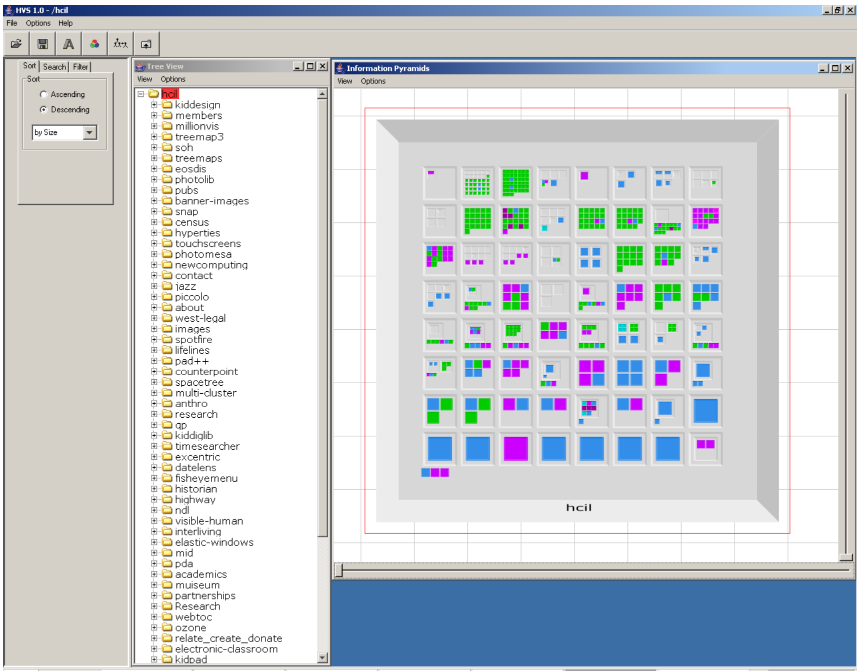


Figure 4.2: The sort option in HVS. Sorting in descending order by size was chosen. The data is sorted in all open visualisations.

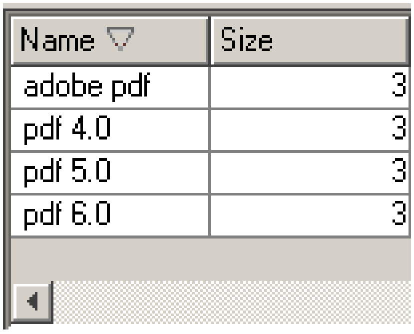


Figure 4.3: The properties panel in HVS. The panel lists all available properties of the selected objects.

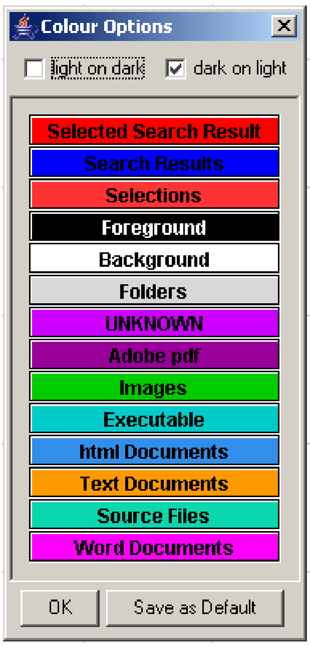


Figure 4.4: The colour option panel in HVS. The chosen colours for the different file types and directories is shown (if applicable) in all open visualisations.

Figure 4.5: Information Pyramids in HVS showing a small hierarchy. Colour coding of file types is turned on.

The height of the pyramid grows with the depth of the hierarchy. The root directory is logically the largest pedestal and the pyramids of its subdirectories are positioned on top of it, as can be seen in Figure 4.5. The contents of the directories at all levels are always laid out in the following manner. First the subdirectories ordered according to the current sort order in HVS beginning in the left corner, then the files, again ordered. The names of the directories and files appear on the front side of the pedestals and blocks. If used, thumbnails of files are displayed on top of the blocks representing them. For navigation, users can double-click on the directory or file and the view is animated smoothly to zoom in. When zoomed in, only a part of the structure is visible on the screen. In order to explore more of the structure at a selected zoom level, the users can pan the display with the mouse. Zooming can also be done by using the scroll wheel of the mouse. The viewing angle can be selected by the user as well. A bird's-eye perspective is best for viewing the overall structure. The view from the front shows the depth of the directories and the pyremid-like structure.

4.3.2 Hyperbolic Browser

In order to use the available screen space more effectively, the Hyperbolic browser first lays out the hierarchy in the hyperbolic plane and then projects the layout onto a circle in the Euclidean plane. Thus, the whole hierarchy fits in this circle. The root directory is placed in the centre of the circle. Its contents (subdirectories and files) are arranged around the root directory clockwise in alphabetical order. Directories are listed first, followed by files. The contents of the subdirectories (other levels than the root) are displayed in an arc. The elements belonging together are connected by lines for better orientation. In order to navigate, users simply double-click on the directory of file of interest. This part of the hierarchy is enlarged, whereas the opposite part of the hierarchy shrinks. Only the directories and files having an icon or dot can be double-clicked. The root directory is highlighted for better orientation. By double-clicking the root directory, the whole hierarchy repositions itself in

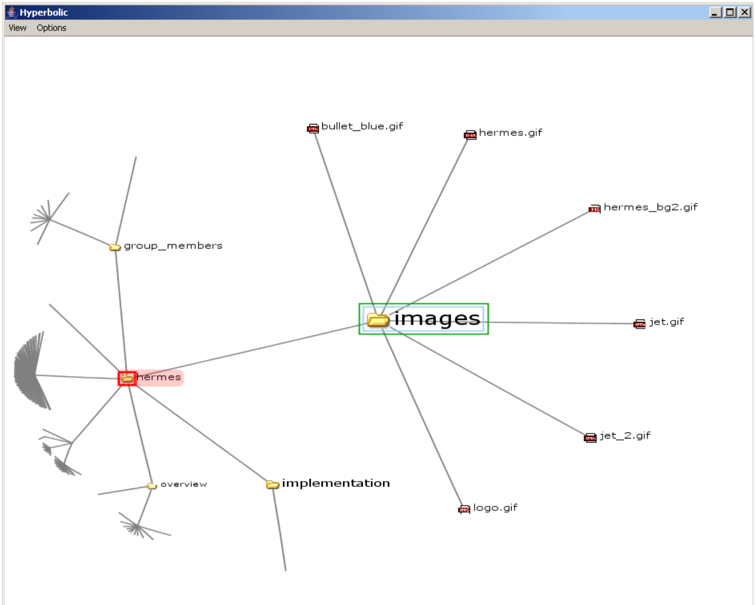


Figure 4.6: A hierarchy in Hyperbolic browser in HVS. Directory "images" is enlarged and the root directory "hermes" is highlighted.

the centre of the screen. The zoom factor can be changed with the mouse wheel. In order to navigate more quickly and freely, users can pan the entire hierarchy while holding down the right mouse button. Figure 4.6 shows a hierarchy in Hyperbolic browser.

4.3.3 Walker Tree

In the Walker Tree visualisation, the hierarchy is presented as a node and edge tree usually drawn by hand. The root directory is on top, its children underneath, connected by lines. Individual levels of the hierarchy are always placed at the same height. Directories and files are represented by common icons or coloured dots, depending on user settings. The angle of the node names can be selected by the user in the settings. For small hierarchies, the whole tree fits on the screen. For larger hierarchies, the screen becomes crowded and zooming is necessary. When a node is double-clicked, it is positioned in the centre of the screen. Two scroll bars, vertical and horizontal, can be used to move the tree vertically and horizontally, respectively. The horizontal slider changes the horizontal zoom, i.e. makes the tree wider or narrower. The vertical slider changes the vertical zoom, making the tree higher or shorter. The sliders zoom at the centre of the screen. The mouse wheel changes the overall zoom, whereby the horizontal and vertical zoom are synchronised. Zooming with the mouse wheel is centered where the mouse pointer is located. The whole tree can be panned in all directions by hand while holding down the right mouse button. Figure 4.7 shows a hierarchy in the Walker Tree layout in HVS.

4.3.4 Tree View

The Tree View visualisation presents the hierarchy as an alphabetical list of directories and files in the manner of a traditional file system browser such as the Mac File Finder or Windows Explorer. The individual levels of the hierarchy are separated by indentation and all elements of one level are connected by a line. At each level, the directories are

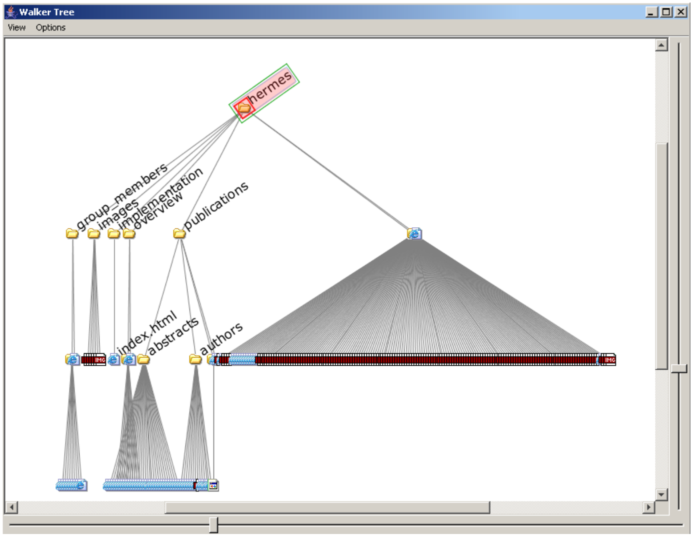


Figure 4.7: A four-level hierarchy in Walker Tree visualisation in HVS.

listed alphabetically, followed by the files, sorted alphabetically as well. The directories and files can be represented by common icons or coloured dots, depending on user settings. In order to navigate, double-clicking a directory opens it and shows its contents. Figure 4.8 shows a hierarchy with several open directories. Opening a directory can also be achieved by single-clicking the plus left of the directory name. Closing a directory and hiding its contents is done by double-clicking the directory or single-clicking the minus on the left. For large hierarchies or when several directories are open, the list will not fit on the screen. In order to move up and down the list, the mouse wheel or the scroll bar can be used.

4.3.5 Tree Map

The Tree Map visualisation implemented in HVS uses two approaches: the slice & dice approach and the squarified approach. These can be changed by a single click by the user. The hierarchy is represented by nested rectangles. The largest rectangle is thus the rectangle representing the root directory. Directories are shown as a frame with their names written in the upper part of the frame. Files are filled rectangles with their names written inside the rectangles. The colours for representing directories and individual file types can be chosen by the user. The contents of a directory are shown as frames and rectangles inside this directory (frame). In the slice & dice approach, the layout of the hierarchy is achieved by alternating vertical and horizontal subdivision. Here, the directories are vertical rectangles of the full height of their parents. The directories are ordered from left to right. Their contents are positioned horizontally. Here, the ordering is from top to bottom. The vertical frames and rectangles tend to be very narrow, making it impossible to read the names. This problem hardly occurs in the squarified approach. Here, the ordering of the contents is according to size. The largest directories and files are located in the left upper corner; the smallest ones in the right lower corner. Navigation is the same for both approaches. Single-clicking a file or directory simply highlights it, making it easier to see the borders and depicting the contents. Double-clicking a directory opens the directory. The directory now fills the screen. All upper directories are not visible at this moment. If a file is double-clicked, the directory it belongs to is opened, thus filling the screen. A double-click on the currently open directory moves one level up in the hierarchy. Figure 4.9

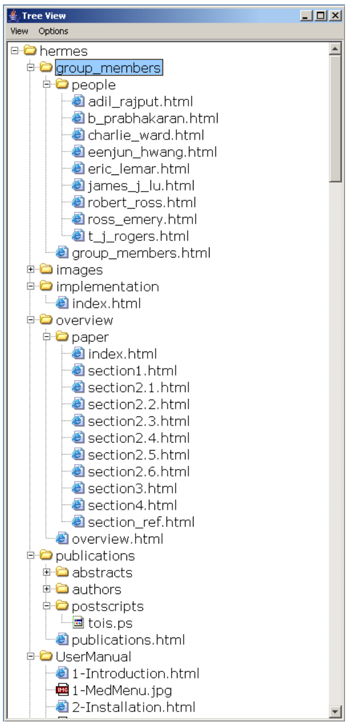


Figure 4.8: Tree View visualisation in HVS displaying a hierarchy with several open directories. Alphabetical sorting order and icons for representing elements are chosen.

shows the entire hierarchy in the squarified approach.

4.3.6 Magic Eye

The focus + context Magic Eye visualisation lays out the hierarchy as a Walker tree and projects this layout onto the surface of a hemisphere. For better orientation, the levels of the hierarchy are placed on alternating dark and light stripes from the pole to the equator of the hemisphere. The focus area is around the pole of the hemisphere. As the whole hierarchy is laid out on the hemisphere, the overview (context) is always visible. The root directory is positioned on the pole, its contents around it in alphabetical order, as can be seen in Figure 4.10. Directories and files can be represented either by icons or coloured dots, depending on user settings. In order to navigate, users simply double-click on the desired directory or file, which is moved to the pole, where magnification is largest. The other parts of the hierarchy shrink, but remain visible. Manual panning while holding down the right mouse button is possible as well. Zooming with the mouse wheel enlarges or shrinks the whole hemisphere.

4.3.7 Cone Trees

In Cone Trees the hierarchy is represented by a number of cones in 3D. The root directory is the apex of the first cone. Its children are placed uniformly around the base of the cone. The subdirectories are again apexes of cones touching the base of the parent cone (at all levels). Due to the 3D layout, the whole hierarchy can be represented on the screen. The nodes (directories and files) are represented by coloured dots. Users can colour-code the

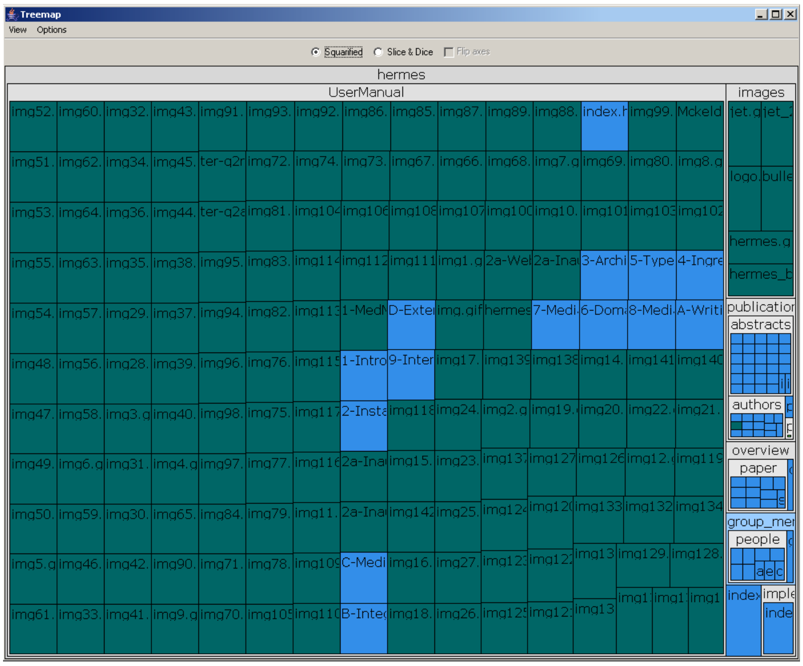


Figure 4.9: A hierarchy in the Tree Map visualisation in HVS. Squarified approach and file type colour coding are turned on.

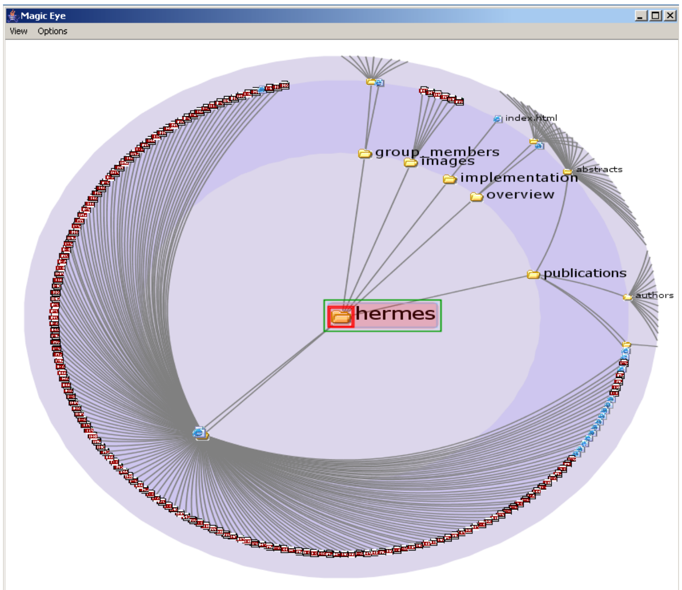


Figure 4.10: The Magic Eye visualisation in HVS showing a hierarchy with the root at the pole. The root is highlighted and icons are used.

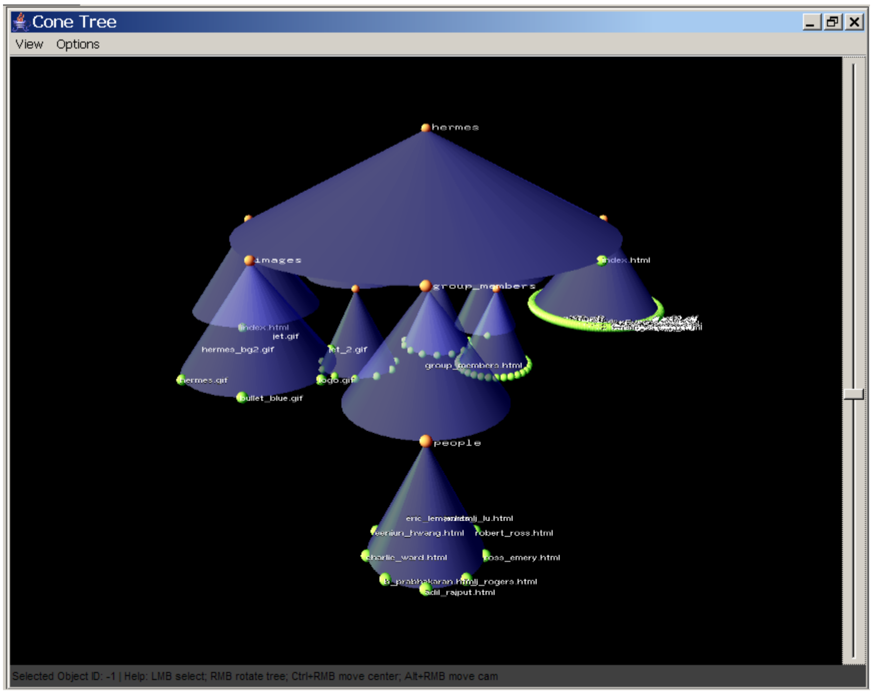


Figure 4.11: A hierarchy shown in ConeTree in HVS. Colour coding of file types is turned on.

directories and file types, as in Figure 4.11. By clicking on a node, the tree is rotated in order to show the selected node. Users can as well rotate the tree freely to explore while holding down the right mouse button. Users can further explore the hierarchy by moving around a "camera", thus changing their point of view.

4.3.8 Sunburst

In Sunburst (Figure 4.12), the hierarchy is represented by concentric circles and arcs. The root directory is the middle circle. Its contents are represented by arcs around it. The size of the arcs corresponds to the size of the directories and files. The contents of subdirectories are again arcs, where these arcs together have the size of the parent arc. Since hierarchies usually have directories of different depths, the representation resembles a hand drawn sun with sunbursts, hence the name. As the arcs tend to become very narrow with hierarchy depth, navigation is necessary. By double-clicking an arc, this arc and its contents is shown. The whole hierarchy shrinks to make room for the extension of the desired arc. Then this arc is enlarged outside of the hierarchy. The hierarchy returns to the initial representation (with no arcs enlarged) after double-clicking the root directory. Users can assign colours to directories and file types. The level of displayed details can be adjusted by a slider.

4.3.9 InfoLens

The InfoLens visualisation [Nussbaumer, 2005] is a visualisation which combines some aspects of the Hyperbolic and Magic Eye techniques. The hierarchy is laid out as a Walker tree, and then projected onto a disc. In contrast to the Magic Eye visualisation, the tree is drawn from left to right (the root is on the left, its children drawn to the right). The

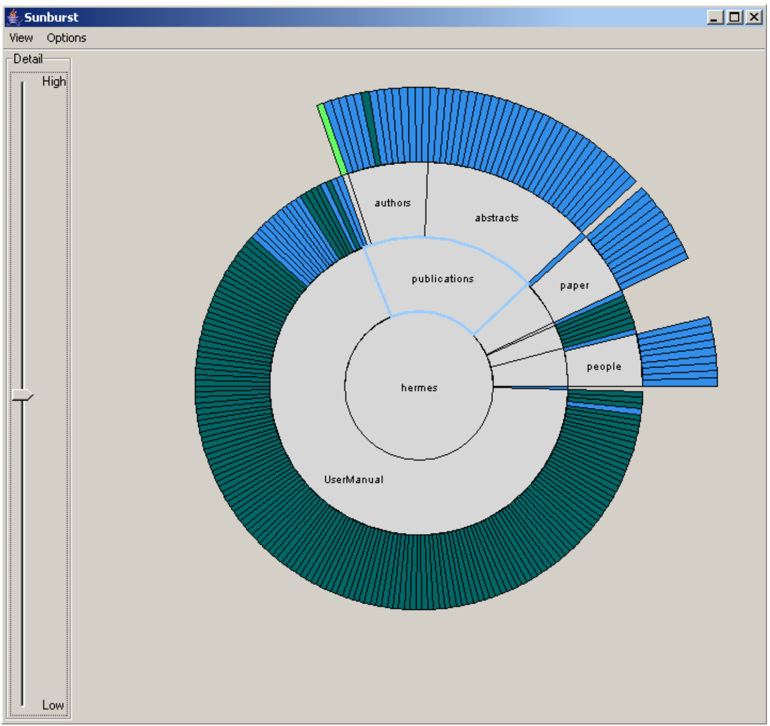


Figure 4.12: The Sunburst visualisation in HVS showing a hierarchy at a middle level of detail. File type colour coding is turned on.

magnification area is a pear-shaped lens, see Figure 4.13. The parts of the hierarchy in this area are enlarged. A visual cue is used to show the lens' magnification. The central part having the largest magnification power has the darkest shade of colour. The shades become lighter with decreasing magnification towards the border of the lens. The directories and files can either be represented by icons or coloured dots, depending on user settings. By double-clicking a directory or file, the hierarchy is rotated in order to bring the desired part onto the lens. Thus, the desired node and its surroundings are enlarged. Users can also drag the whole hierarchy and move the parts of interest onto the lens. The mouse wheel changes the zoom, here the area and magnification power of the lens.

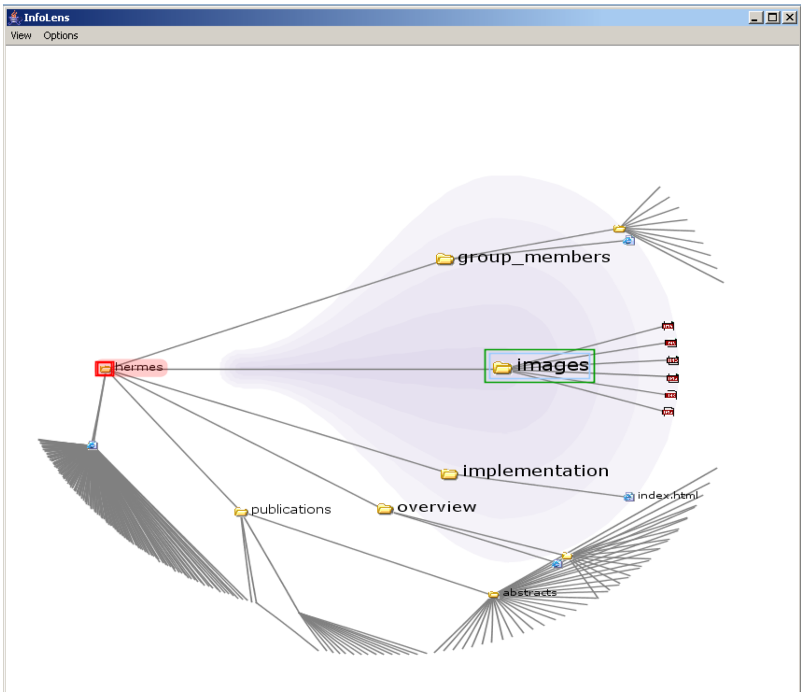


Figure 4.13: The InfoLens visualisation in HVS showing a hierarchy. Directory "images" is enlarged and icons are used.

Chapter 5

Usability Evaluation

Usability is defined by the ISO 9241-11 Standard [ISO, 1998] as

> "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

Evaluation is the determination or assessment of the worth of a product. Evaluation can be conducted in order to:

1. check integration of a product into real working conditions,
2. compare different methods or
3. ensure that standards are met

(adapted from [Horton, 2006])

Usability evaluation measures the usability of a product. Usability evaluation helps answer questions such as: Does the product do what it should? Does the product meet user expectations? Does the product work the way intended by designers/programmers? Thus, new products or programs should be evaluated, ideally alongside with the development and implementation. This also applies to information visualisation, as its primary goal is to help users deal with data. Especially in information visualisation, new techniques and approaches should be evaluated in order to find their strengths and weaknesses. It is not sufficient if the new methods are just "cool" [Czerwinski and Larson, 1997]. The goals of usability evaluation are to find out about:

- the functionality of the tested system
- users' liking of the interface
- potential problems with the system

(adapted from [MacKenzie, 2006])

Evaluation can help find out which methods and approaches are really helpful. Continuous evaluation can monitor improvements of products. Independent of the methods and techniques, all usability evaluation has to be planned and documented carefully. Only with detailed documentation, the experiments are reproducible and comparable with other experiments.

The DECIDE framework [Preece et al., 2002] was developed to help plan and organize an evaluation study. The acronym DECIDE refers to:

D - Determine the goals (what is the purpose of the evaluation)
E - Explore the questions to be answered (questions to guide the evaluation)
C - Choose the paradigm and techniques for evaluation
I - Identify the practical issues (schedule, users, equipment, etc.)
D - Decide on ethical issues (user privacy, consent forms, politeness)
E - Evaluate, interpret and present the data (data analysis)

The different methods for usability evaluation can be divided into two coarse categories: involving experts and involving users. Usability inspection done by experts is covered in Section 5.1. Usability testing involving users is covered in Section 5.2. Query methods which also involve users are covered in Section 5.4.

5.1 Usability Inspection Methods

Usability inspection examines the user interface by analysing and judging it according to certain heuristics (usability recommendations). Here, the system is evaluated by experts, not by users. Compliance with general "good design" and usability guidelines is checked.

The heuristics are published by usability experts and bureaus, such as the Nielsen Norman Group [Nielsen et al., 2006]. As usability inspection is conducted by usability experts rather than real users, it is less time-consuming and cost-consuming.

The following taxonomy of usability inspection methods is adapted from [Andrews, 2006a] and [UsabilityHome, 2005]:

- **Heuristic Evaluation**: Trained evaluators examine the interface following a list of usability recommendations (called heuristics). The aim is to find possible problems with the interface design. First, the evaluators may have an initial meeting. Then, every evaluator goes through the interface alone, usually twice. When all evaluators have finished their analyses, they may meet to discuss and write the problems they found. The advantages of heuristic evaluation are the low price, intuitiveness, applicability in all stages of development and the fact that many problems can be found.

- **Cognitive Walkthrough**: Evaluators imagine a goal for novice users and a usage scenario for the interface. Then the evaluators "walk" the imagined way through the interface to achieve the goal. They keep in mind three questions and by going through the tasks try to answer them: Is the correct way obvious? Is the correct way easy to detect? Are the reactions along the correct way unambiguous? Possible task-oriented problems can be detected easily. This method is applicable in early stages of development. However, its drawbacks are time and training requirements.

- **Pluralistic Walkthrough**: Users, developers and usability experts work with the paper prototype of the product. All participants have to go through the interface in order to fulfil a set of tasks. They write down in detail the paths they would take. Then, the results are presented and discussed. First, the users present their solution, then the developers and usability experts. Afterwards, all are shown the correct solution as intended in the design.

- **Action Analysis**: Estimates the task completion time of a skilled user. The complete sequence of steps necessary for a task must be found. These steps are very small, such as moving a mouse to a given button or typing something on a keyboard. The times to accomplish the steps are estimated. Then, the partial times are summed up to obtain an estimate for overall task completion time. Action analysis is very precise and detailed. However, it requires skilled experts and time.

- **Feature Inspection**: Studies the features of a product. The evaluators get a task they must accomplish by using the product. Such a task could be to enter a table with a table processor. The evaluators analyse the features needed to finish the task for their usability. The features such as entering the data, formatting the table or performing simple operations are checked for their understanding and ease of use.

5.2 Usability Testing Methods

In usability testing, the usability of a product is measured in tests by using real users rather than experts. The users try to use the product as intended by the developers. Usability testing with real users can reveal problems of the interface the developers did not see. Apparently, developers can not "forget" their experience and act completely like first-time users. Therefore, usability testing with real users is a very important step in development.

The advantages of usability testing are controlled conditions and easily collected data (automated time recordings, video taping of user behaviour). User feedback is often also collected in questionnaires which can be analysed statistically. The major drawback is the fact that the tests usually do not take place under the real life working conditions of the users. These methods are also quite time consuming. However, useful feedback on user experience, including problems, can be gathered.

- **Thinking Aloud**: While performing tasks, users speak their thoughts aloud. This method yields much information with few users (three to five). Different problems and their causes can be found. The drawback of this method is that users are not used to commenting their steps while working and this often slows them down.

- **Co-Discovery**: Two test users work together with the interface. The users try to accomplish the given tasks. Their communication and comments give valuable information about the interface. This method is more natural than thinking aloud, as the user naturally talk to each other. However, this method requires twice as many users than other methods.

- **Coaching Method**: Users are given tasks to fulfil, an expert is present and answers any questions about the system the user would ask. This method is used to improve the documentation or training materials of a system. It may also help redesign the system so that users would have fewer questions. The drawback of this method is the need for a system expert able to answer unexpected questions from the users "on the fly".

- **Question-Asking Protocol**: Users work with the tested system to accomplish given tasks. They are encouraged to think aloud as they work. Additionally, they are asked direct questions about the system and their actions. Answering concrete questions is easier and more natural for the users. However, this method is prone to constructed and incorrect answers.

- **Formal Experiments**: Users work with the system or interface under controlled conditions. Usually, task completion times are collected along with subjective user opinions. Formal experiments are described in detail in Section 5.3.

5.3 Formal Experiments

Formal experiments are controlled experiments with selected test users under controlled conditions. Users perform different test tasks while being observed. Several kinds of data can be collected, such as task completion time, or the number of correct answers. In formal experiments, either one interface can be tested, or several interfaces can be compared. The data collected in formal experiments is analysed statistically. In order to obtain statistically meaningful results, many users are necessary (16-20).

Typical steps in a formal experiment are as follows (adapted from [MacKenzie, 2006]):

- Prepare equipment (product and testing equipment)
- Design experiment (test procedure and design, pilot study)
- User study (data recording and interviews)
- Analyse data (hypothesis testing, statistical significance)
- Publish results

Formal experiments can take place either in a usability laboratory or the users are observed in their real-life working environment. Both methods have their advantages and drawbacks.

Testing in *usability laboratories* ensures a controlled environment with no external interferences. Usability laboratories are equipped with all necessary devices. However, the laboratories are not the natural working environment for the users and this might intimidate them. Users might act artificially in an artificial laboratory environment.

In order to get more natural feedback from the users, direct observation in *users' workplace* can be conducted. Here, users work in their familiar setting with the product. No simulation of a workplace is needed. However, capturing the data can be a challenge. Depending on the data to be collected, users can be audio taped or video taped, or their interaction with the system logged automatically. The equipment has to be arranged on-site and adapted to the environment. This method is rather time-consuming for the observer.

The collected data such as time, key presses or mouse clicks is analysed afterwards. Task completion times and error rates are usually calculated. Feedback questionnaires are used to collect users' subjective opinions and preferences. User comments can reveal wishes or suggestions for improvement.

When comparing two or more interfaces, two possible experimental designs can be used: between-groups or within-groups.

- In a between-groups experiment, the users are randomly divided into two groups. Each group tests only one interface with identical tasks. As every user works with only one interface, no learning effect occurs. However, twice as many users are needed. Additionally, user skills may vary heavily between the groups and thus bias the results.
- In a within-groups experiment (also called a repeated measures experiment), only one group of users is necessary. The users are randomly assigned to one of the two scenarios. Either interface A is tested before interface B, or interface B is tested before interface A. Each participant thus uses both interfaces, performing equivalent tasks on them. This design reduces the influence of the individual user skills. Learning effects still occur, but are counterbalanced across the two interfaces.

In a repeated measures design, since each user tests each interface, fewer users are needed in total. Additionally, the variation in individual user skills is reduced. In order to reduce the learning effect, particular care must be taken to counterbalance the testing order. Latin squares are used to find the order. Latin squares are n×n tables where each case only occurs once in each row and column. So in order to test three different test conditions, three users are needed. Each user tests one combination of conditions, corresponding to one row in the Latin square.

Simple Latin squares are obtained by shifting each row by one to the next row. Table 5.1 shows a simple 3×3 Latin square. The drawback of this simple method is that condition A always follows condition B (except for the last row). A learning effect from condition A to condition B is thus possible.

Balanced Latin squares reduce this problem by having each condition followed by the other conditions an equal number of times. Balanced Latin squares exist for even numbers only. Table 5.2 shows a balanced 4×4 Latin square.

A	B	C
C	A	B
B	C	A

Table 5.1: A 3×3 Latin square.

A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

Table 5.2: A 4×4 balanced Latin square.

5.4 Query Methods

Query methods are used to collect users' subjective opinions after they have used a system. Query methods often follow other usability tests such as formal experiments. The methods include interviews and questionnaires.

- **Interviews**: Users are encouraged by interviewers to give their opinions. Usually, the interviewer has a set of questions to which the users answer orally and the interviewer writes these answers down or records them on audio or video. Interviews can be either structured or unstructured. While structured interviews follow a script and may resemble a questionnaire, unstructured interviews are not scripted. Semi-structured interviews are often used; using a script, but allowing to investigate special topics in more detail. Unstructured interviews are usually used in the early stages of development. Interviews are more complicated to analyse and compare with each other than questionnaires.
- **Questionnaires**: Written forms with different questions to be answered by the users. Open or closed questions are used. Closed questions can be analysed more

easily, but closed questions, open questions allow room for user comments. Finding appropriate questions is a crucial task. With questionnaires, a large number of participants can be contacted, either by post or on the internet. The fixed structure of questionnaires makes them easier to analyse, but they are less flexible than interviews and do not allow on-demand investigation.

Chapter 6

Comparative Studies of Hierarchy Visualisations

The area of information visualisation has been booming in the last decades, yielding new methods and techniques to represent data visually. Evaluation studies, though very important, are still rather rare. This chapter summarises some of the published comparative studies. As the aim of this thesis was a study on hierarchical visualisations, this overview concentrates on hierarchies only. The range of the studies is very wide: from small studies performed after the development of a new technique to large comparative studies of commercially available products.

In her paper, Plaisant [2004] discusses the evaluation of information visualisation and presents some proposals for improvement. Current evaluations can be divided into four categories:

- Controlled experiments which compare design features.
- Usability evaluation of an existing application.
- Controlled experiments which compare different applications.
- Case studies of applications in real-life settings.

Controlled experiments are currently most widespread. Evaluation reports are very important for finding and understanding strengths and weaknesses of visualisations. Still, these methods should be revised in order to help spread information visualisation. Convincing evidence that information visualisation is useful is needed. Furthermore, the researchers should make their studies comparable and reproducible. Task taxonomies and benchmark repositories of tasks and data sets, such as the Information Visualization Benchmark Repository [Fekete and Plaisant, 2006], are the first steps. Another important step are toolkits and development tools helping developers include information visualisation features into other applications. This might help to bring the techniques to the public.

Comparing the published studies is still very difficult. Not only are different evaluation methods being used, but the reports are not consistent in presenting results. Chen and Yu [2000] compare studies based on meta-analysis of three aspects: users, tasks, and tools. The emphasis of this paper is on visualisations of trees and networks. Only studies on information retrieval tasks were analysed. Out of 35 studies, only six fulfilled all selection criteria. The studies must:

- Report on experimental design.
- Include at least one visualisation technique.
- Include at least one dependent variable on accuracy or efficiency.
- Report the results in sufficient detail.

Due to the diversity of studies, meta-analysis was difficult to perform. The combined size effect of visualisations both on accuracy and efficiency was positive, but not significant. Therefore, more studies are needed in order to prove that presence of visualisation is more efficient than its absence. Individual differences in cognitive abilities should be studied and reported more precisely in future studies. The authors propose six aspects to be followed in future studies:

- Use standartised testing information.
- Clearly describe the visualisation and its features.
- Use standartised task taxonomies.
- Focus on the relationship between task and feature.
- Use standartised tests of cognitive ability.
- Report statistical results in sufficient detail.

The authors also propose further development and use of task-feature taxonomies in future studies. In consequence, better comparison of studies will be possible.

Lamping et al. [1995] presented the Hyperbolic browser and a small study comparing the Hyperbolic with the tree view browser. Only four participants took part in a within-subject design. The tasks were to find and double-click particular nodes in four WWW hierarchies which were identified by their URL. There were no significant differences between the browsers in task completion times. Subjectively, all four participants preferred the hyperbolic browser both overall and in terms of providing a sense of the overall tree structure and finding nodes by names.

Czerwinski and Larson [1997] compared the hyperbolic browser with a tree view browser. Seventeen participants, all of them with computer and internet experience, participated in the study. The users were given eight tasks, four for each browser. The order of browsers was counterbalanced. The hierarchy used in this study was a part of the Microsoft Encarta encyclopaedia, having an $8 \times 8 \times 8$ structure. The users were asked to think aloud while working. Users' task completion times as well as their paths through the hierarchy were recorded. There was no statistical significance in the task completion times. In the subjective ratings, the tree browser came out ahead on most ratings. Significant differences in favour of the tree browser were found for the following statements: ease of use, familiarity, and ability to show current location. Significant differences in favour of the hyperbolic browser were found in terms of using new technology and feeling unique.

In a comparative study, Wiss and Carr [1999] evaluated three interfaces: Information Landscape, Cam Tree, and Information Cube. The environments were in-house VRML models with no custom navigation possibilities. Instead, all interfaces used the same navigation offered by CosmoPlayer2.0. Participants in the study were 25 students with at least one year of computer experience. The tasks were two search tasks, two count tasks and two compare tasks. The six hierarchies used in the study were representing a file system. None of the participants used a hierarchy twice. The study usually lasted between 60 and 90 minutes and included a 20-minute introduction in the visualisations and example tasks. Task completion times and task correctness were recorded. A task timed out after five minutes, the users then went on with the next task. The participants filled out background and feedback questionnaires. There were significant differences for task completion times (Information Cube slowest, then Cam Tree, and Information Landscape fastest). Significant differences in accuracy were found between the interfaces. Subjective feedback also rated the Information Landscape highest and the Information Cube the lowest.

Stasko et al. [2000] present two experiments comparing the TreeMap and the SunBurst approach. The visualisations were implemented for Sun workstations running Unix. Besides the window with the visualisation, a control panel window and a colour legend window were available. Participants were given a tutorial on one tool and performed eight training tasks (similar to the tasks in the study) on an appropriate training hierarchy. After that, the participants performed 16 tasks of the study for this tool and filled out a questionnaire. Next, they were given a tutorial in the other tool and performed example tasks. After this, the study with the other tool was conducted, followed by a questionnaire. The tasks on the two tools were performed on two similar hierarchies. The hierarchies consisted of approximately 500 files and directories in the first experiment. In the second experiment, the number of files and directories in the hierarchies was increased to 3000. A maximum time limit of 60 seconds was set per task. Besides task completion time, the correctness of answers was recorded. The tasks belonged to the following categories:

- Find the largest and second largest files.
- Find the largest directory (concerning size).
- Find a file with given path.
- Find a file without path, only by name.
- Find the deepest subdirectory.
- Find a directory having files of a given type.
- Find the largest file of a given type.
- Compare two files and find the larger one.
- Find two duplicated directories having the same files.
- Find the larger of two directories by size.
- Find the directory containing more files of two directories.

32 students participated in the first experiment. The participants were more successful using the SunBurst tool compared to the TreeMap tool. The results are, however, not statistically significant. Users did not prefer one tool over the other. 28 students participated in the second experiment, using the larger hierarchy. Concerning the correctness of answers and task completion times, no significant differences between the two tools could be

found. The subjective evaluation yielded a significant preference of the SunBurst tool. [1]

Risden et al. [2000] compared three different interfaces for visualising web browser content and relations. The interfaces were a 2D collapsible tree browser, a 2D snap.com categories view and a novel 3D viewer called XML3D. At the time of this study, snap.com was a web-based hierarchy of categories, but is now a standard searching engine. XML3D combines a 3D hyperbolic graph with 2D lists of data and links to related content. All three interfaces provided a search facility, not just category browsing. The target user group were web page administrators. 15 skilled programmers participated in the study. The hierarchy used in this study was a 12000 node snap.com hierarchy, which ported to the XML3D viewer and the collapsible tree browser. The live version of snap.com was used as the second 2D viewer. Tasks were separated into four groups, depending on the number of parents of a category. The four groups were: one parent, existing category; multiple parents, existing category; one parent, new category and multiple parents, new category. The participants were asked to time themselves. When starting a task, they started the timer in a log tool, and stopped this timer at the end of task. A short introduction and example tasks were given to the users before the study. When using XML3D, participants were faster answering tasks concerning existing categories (both single and multiple parents) than if using the other visualisations. No difference could be found in the new category tasks between the 3D and 2D tools. None of the results were statistically significant. The subjective rating slightly preferred the 2D tools over the 3D tool, but these resutls were also not statistically significant.

Barlow and Neville [2001] compared four different tree visualisations: Organization Chart, Tree Ring, Icicle Plot, and TreeMap. All four visualisations were studied in the first experiment; TreeMap was not studied in the second experiment. The first experiment was conducted with fifteen participants in a repeated-measures design. There were five tasks:

- Is the tree binary or n-ary?
- Is the tree balanced or unbalanced?
- Find the deepest common ancestor of two given nodes.
- Count the number of levels in the tree.
- Find the three largest leaves. (This task was not used for the organization chart.)

The users performed the tasks for all visualisations in eight trees created for the study. Users were first introduced to the visualisations and performed training tasks. After that, they performed the actual tasks of the experiment. The users rated the visualisations subjectively. The analysis of task completion times shows significant differences between the browsers. The correctness of answers was lower for the TreeMap compared to the other visualisations. However, the differences between Icicle Plot, Organization Chart and Tree Ring were not significant. The subjective ratings preferred Icicle Plot significantly over TreeMap, Icicle Plot over Tree Ring, and Organization Chart over TreeMap. A further fifteen users participated in the second experiment. Only three visualisations were used in the second experiment: Icicle Plot, Organization Chart, and Tree Ring. A repeated-measures design was used. Two tasks were used: node description task and node memory task. As in experiment 1, users were given an introduction and example tasks before

[1] Statistical significance is not explicitly given in the paper. Based on the number of votes, a one-way Chi Square test was performed by hand.

working on the actual tasks. Nine trees were created; each participant worked with all nine trees. Concerning the correctness of answers, no significant differences between the visualisations were found. As for the task completion times, the visualisations performed differently for the tasks. For the node description task, Icicle Plot and Organization Chart were both significantly better than Tree Ring. For the node memory task the differences were not significant. Icicle Plot and Organization Chart were significantly preferred for the node description task.

SpaceTree, Hyperbolic, and Windows Explorer visualisations were used in the study by Plaisant et al. [2002]. Eighteen computer science students participated. A 3×7 repeated measures design was used. The seven tasks belonged to one of three categories: node searches, search for the nodes visited earlier, and topology tasks. Users first explored the browsers alone, then supervisors showed them all features. After performing the tasks with all visualisations, users filled out a background and feedback questionnaire. For the find node tasks, Windows Explorer was significantly faster than Hyperbolic for the first task and SpaceTree significantly faster than Explorer for the third task. For the search for nodes visited earlier, SpaceTree performed significantly faster than Hyperbolic and Explorer significantly faster than Hyperbolic and SpaceTree. For the topology task - finding all ancestors of a node - SpaceTree was significantly faster than Explorer. In tasks concerning the local topology, Hyperbolic performed significantly better than SpaceTree, but not significantly better than Explorer. Subjectively, users rated Explorer significantly less "cooler" than the other visualisations. The differences between SpaceTree and Hyperbolic were not significant. As for using the visualisations again, there were no significant differences between the visualisations.

Four information visualisation systems were compared by Werner [2002]: Star Tree, The Brain, HypViewer, and Internet Explorer. 27 students participated in the study. After up to 30 minutes of exploring the visualisations, participants had to find six given pages within the Porsche website. Task completion times and correctness of answers were recorded. Concerning the task completion times as well as correctness of answers, Internet Explorer performed the best. The performance of HypViewer was followed by The Brain. The worst performance in both task completion time and correctness was shown by Star Tree. The poster does not state any significance of results and further results.

Kobsa [2004] conducted a comparative evaluation of six externally implemented tree browsers: TreeMap, SequoiaView, BeamTrees, TreeViewer, StarTree, and Windows Explorer as a baseline. A between-groups design with 48 participants was used. The participants were students with a minimum of one year computer experience. The hierarchy used in this study was a part of the eBay hierarchy with five levels containing 5799 nodes. The participants were randomly divided into groups, each testing one condition. A 30 minutes introduction to the visualisation was given to groups of two to four students. Additionally, they worked on training tasks. During the experiment, participants had to answer 15 tasks. Nine of the tasks were structure-related; six were attribute-related tasks. Participants timed themselves under supervision and were advised to abort tasks after five minutes. Task completion times and correctness of answers were recorded. Users gave feedback on ease of use, effectiveness and whether they would use the system again. The findings of the analysis of correctness of answers are as follows: TreeMap had the most correct answers, TreeViewer and BeamTrees the least. Concerning the correctness of answers, BeamTrees were significantly worse than TreeMap, SequoiaView, StarTree and Explorer. TreeViewer performed significantly worse than TreeMap, StarTree, Explorer and SequoiaView. As for the task completion times, Explorer and TreeMap were the fastest and BeamTrees the slowest. Significant differences could be found between the browsers: BeamTrees worse than TreeMap, SequoiaView, StarTree, TreeViewer and Explorer. TreeViewer performed worse than TreeMap, Explorer and SequoiaView. SequoiaView and StarTree performed both worse than TreeMap and Explorer. User satisfaction yields following significant differences between the browsers. Concerning the ease of use, BeamTrees were rated worse than TreeMap, SequoiaView, StarTree, TreeViewer and Explorer. Explorer was rated better than SequoiaView and TreeViewer. In terms of effectiveness, BeamTrees were rated worse than TreeMap, SequoiaView, StarTree and Explorer; TreeViewer was rated worse than TreeMap and Explorer. As for the system being used again, BeamTrees were rated worse than TreeMap and Explorer. Explorer was rated better than StarTree, SequoiaView and TreeViewer. TreeMap war rated better than TreeViewer. All the stated differences are statistically significant.

Table 6.1 summarises the findings of the presented studies. Differences in task completion times are given in column "Completion Time", differencies in correctness of answers are shown in column "Accuracy". Not all papers report accuracy, this fact is noted in the table by "N/A". Often, objective measurement data do not reveal statistically significant differences. On the contrary, significant differences are usually found in subjective ratings. Objective measurements do not always support the subjective preferences. Task completion times are often similar, but users tend to prefer one visualisation over other(s) subjectively.

Some studies are performed with too few users, usually small studies performed after the development of a new visualisation. The findings of such studies can hardly be compared to experiments with tens of users. In addition, the method of the study is important. Results of an evaluation of one visualisation can not be compared to results of comparative studies. Some researchers use external implementations of visualisations, whereas others make their own in-house implementations. Thinking aloud might reveal subjective opinions while working with the tool, but at the cost of slowing the users down. Besides, commenting their own steps is rather unnatural for the users. Self-timing of tasks might stress users too much and bias the results.

The inconsistency in reporting the results is the major obstacle in comparing different studies. Not all papers give details of the analysis or exact data. Statistical significance of results is the most important outcome of studies. Therefore, researchers should take care in reporting them. Careful documentation of methods and results makes studies reproducible and comparable.

Study	Completion Time	Accuracy	Subjective Ratings
Lamping1995	no significant differences	N/A	too few votes
Czerwinski1997	no significant differences	N/A	significant differences
Wiss1999	significant differences	significant differences	significant differences
Stasko2000	no significant differences	no significant differences	significant differences
Risden2000	no significant differences	no significant differences	no significant differences
Barlow2001	significant differences	significant differences	significant differences
Plaisant2002	significant differences	N/A	significant differences
Werner2002	no significant differences	no significant differences	N/A
Kobsa2004	significant differences	significant differences	significant differences

Table 6.1: Results of the presented studies.

Chapter 7

The Hierarchical Visualisation Testing Environment (HVTE)

In the study performed for this thesis, different users had to perform tasks in four hierarchy browsers. The aim of this study was to obtain objective and subjective measurement data to allow a comparison of the four browsers. The best way to record the data is to embed the visualisation application within a test environment. The following requirements were posed on the test environment:

- Launch or open the corresponding view of the visualisation.
- Load the data required for performing the task.
- Log the time stamp of opening the browser and confirming the answer.
- Log the answers themselves.
- Keep the GUI as simple as possible to prevent users from being distracted from the task.
- Use most of the available screen space for displaying the hierarchy browser, rather than the test environment itself.

7.1 HVTE Design

7.1.1 Test Cases

In this case study, four different sets of tasks were used. There were four visualisations and four task sets, yielding 16 unique combinations of test cases (see Section 8.2.2). Each scenario was performed by one user. Users should be able to select their set of tasks, whereby the tests already performed may not be selected. To provide an easy way for the user to select their task set, a simple list of test cases is embedded in the test environment. The whole testing environment should be as simple and intuitive as possible. Furthermore, if a test is suspended or interrupted - for instance by a software crash or an unwanted reboot - it should be possible to continue the test. Therefore, the test sequence within the set of questions has to be reproducible.

The test scenarios were a combination of four hierarchies with four different task sets. In total, each user had to perform 32 different tasks. In order to counterbalance the task

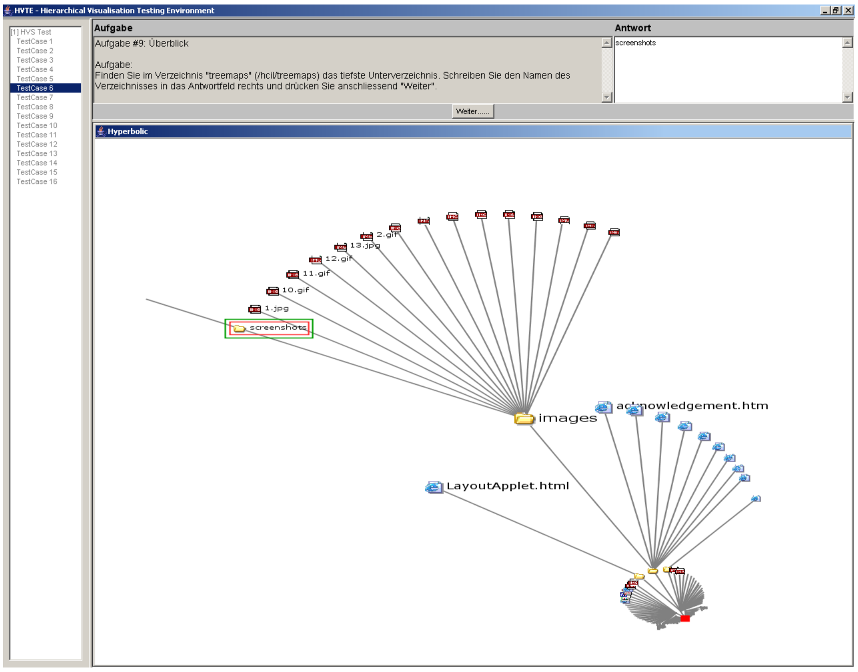


Figure 7.1: Screenshot of a test in HVTE. User entered the correct answer in the answer field.

order, a Latin square was constructed. As the order and combination of test case to question and visualisation was fixed, these assignments were stored in a database.

7.1.2 Tasks Completion

A task always contains the task definition and the corresponding visualisation. An example task would be “find the deepest subdirectory” or “count the elements in directory X”. Thus, a wide variety of answers was possible. When counting elements, a number should be the answer. When finding a specific element, a simple “OK” would be the correct answer. The typing of “found” or “OK” in the search tasks was chosen to reduce the bias. All other tasks required a typed answer and not having to type for search tasks could bias the task completion times. Additionally, users should not be able to skip a question. Writing an answer into the text field was required before continuing to the next task. To prevent users from being distracted from the task itself, a unique and simple means of answering is necessary. Therefore, only a text field and a button were available to complete a task by entering an answer.

7.1.3 Launching Applications

Initially, the testing environment (TE) was intended to be able to launch any application. There would be no limit for the type of test and tasks. However, the application launched by the TE should be command line configurable. The TE could only launch the application by a command line instruction with the corresponding arguments. The testing environment and the launched program are independent applications. This means that no interaction between these two applications is possible. For instance, if a user finishes a task, there is no means of clicking the button in the visualisation and letting the TE know that the task has been finished. Setting the focus on a specific window like the TE window when presenting the task definition and for typing the answer or on the application window where the task should be performed was also not possible. Furthermore, there was the risk of confusing the user by switching between many different windows. Therefore, the initial idea of a universal testing environment had to be discarded.

To overcome the limitations and to enhance the possibilities for the configuration and the handling of HVS, the testing environment and HVS were tightly coupled. The visualisations tested in this study were included in HVS. By embedding HVS in the testing environment, complete control of any visualisation was possible and the visualisation itself may be directly shown in the testing environment window. Thus, Hierarchical Visualisation Testing Environment (HVTE) was born.

7.2 HVTE Database

Each test case involves much information. On the one hand, the complete configuration and task set must be defined. On the other hand, the answers together with the timestamps for the task completion times must be stored. An efficient way to manage such data is a database. For Java, a database which may be completely included in an application exists. Thus, no installation of a database server was required. This database, called Hyper-Sonic or HSQL, has only one disadvantage: There is no practicable means of data management available. To overcome this problem, the same database may be installed on a MySQL database. With an included tool, the data and tables may be transferred between these two types of databases. Hence, the MySQL database together with any of the available management tools is used for configuration and data entry of the database. Then, the database is transferred to HSQL and ready for use within the application. Finally, the data entered during the tasks (answers and times) may be transferred back to MySQL, where the analysis is easier.

The database layout was designed to be very simple. All necessary data should be stored within the database and nowhere else. Tables for holding data for users, tests, tasks, hierarchies, and visualisations are necessary. Information about users, such as which user has already performed a test, the corresponding tasks, and timestamps need to be stored. In the relational database used here, the design in Figure 7.2 was developed. With this schema, easy access to all required data is possible. Table *testsequence* links all necessary tables together. Table 7.1 describes the database tables.

In order to clarify the relation between the different tables of the database, the two most important queries are used: The first query is to retrieve the list of available test cases within one test. HVTE provides the possibility of running different tests within the same testing environment. In this study, only the HVS test was used. The second query retrieves the corresponding tasks and visualisations for the particular test case.

Retrieving the list of test cases: As mentioned before, all tables are connected together through the table *testsequence*. Before the first user performs the test, the necessary test data have to be set up. This means that for all test cases, the complete list of tasks with the corresponding visualsations and hierarchies has to be defined. These definitions are stored in the table *testsequence*. Retrieving the list of test cases for one test requires the evaluation of two table connections. The first connection is required to find all *testsequence* entries which are defined for the desired test. The second connection evaluates all test cases available for the determined list of *testsequence* entries. As there would be a vast amount of results, only unique results (each test case only once) are returned.

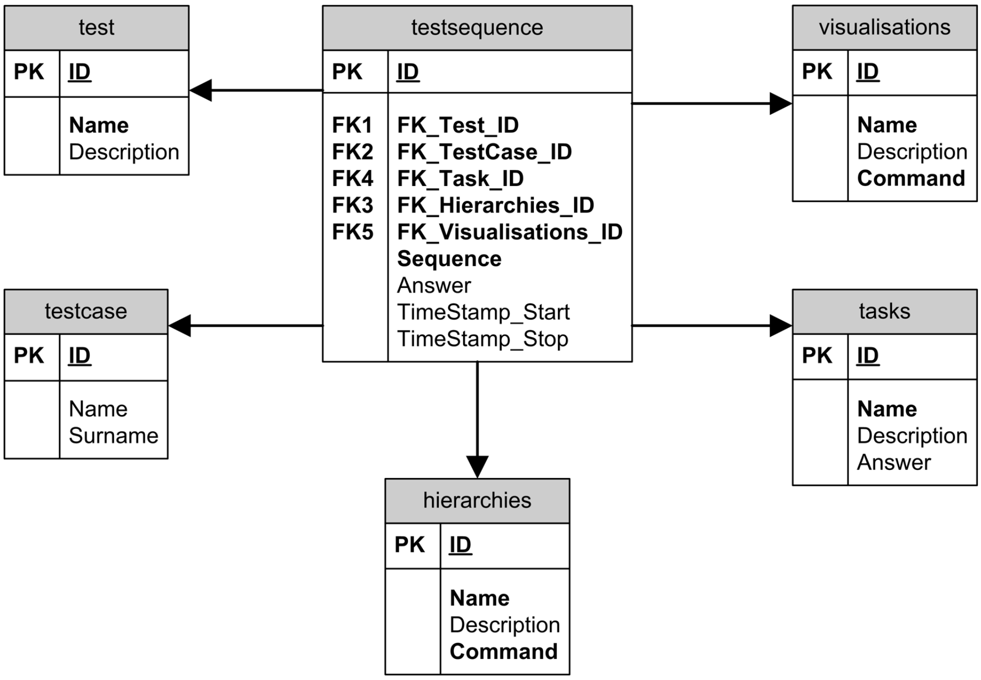


Figure 7.2: The database schema used in HVTE to store test cases and test results.

Determining all required data for the next valid task: From the list of all available test cases determined above, one test case can be selected for which a test will be started. At the beginning as well as during the test, for each task the user has to perform, the next task, visualisation and hierarchy defined by the test sequence (according to the Latin square) have to be fetched. First, according to the active test case, the proper entry in *testsequence* has to be selected. The valid entry is the one with the lowest sequence number, where none of the start timestamp, end timestamp, and answer are set. Next, the necessary data from the task, visualisation and hierarchies tables are retrieved. The connections to these tables are defined by the foreign keys in the *testsequence* table. Now, the test case can be performed by the assigned user. The starting and ending time stamps and the answer are stored correspondingly.

All tasks were stored in the database and their order had to be generated before the test, according to the Latin square. When the users started with the task, the corresponding task description was read from the database and displayed in the task field. When the user clicked on the "Continue…" button, the required visualisation was opened and the time was written to the corresponding row to the task table in the database. The users had to type the answer in the answer field (top right) and acknowledge it by clicking on the "Continue…" button. This triggered the program to write the answer and the timestamp of the click to the database. This ensured a very precise measurement of task completion time. Upon changing the visualisation a pop-up reminder was displayed, informing the user that another visualisation was going to be used in the following tasks.

Each task may have its own visualisation and hierarchy. The hierarchy is initialized with an XML file. Therefore, it is sufficient to store the available hierarchies as a file reference (filename) in the database. Opening a new hierarchy requires several seconds. During this time, HVS completely blocks any operation. Therefore, it is strongly advised to use as few changes as possible in the hierarchy. The different visualisations are referenced

Table	Description
Test	Holds the ID, name and description of tests (different tests are possible). Here, only one test was conducted.
TestCase	Holds the ID of the test case.
Tasks	Holds the ID, name and description of the tasks. Correct task answers can also be stored.
Visualisation	Holds the ID, name, description, and launching command. The fully qualified Java names are used.
Hierarchies	Holds the ID, name, description, and launching command. Only one hierarchy was used in this study.
TestSequence	Besides the foreign keys, the table stores the task time stamps and the typed answer.

Table 7.1: Tables in the HVTE database.

by the fully qualified Java class name and package name. These are also stored in the database. When HVS should open the visualisation, the Java class loader uses the specified name to look for the plugin and its required files and classes.

7.3 HVTE Implementation

HVS is a plain Java application. Java provides some packages to assist with the creation of user interfaces. In HVS, the package "Advanced Widget Toolkit" (AWT) is used. Therefore, AWT was chosen for HVTE as well. All AWT objects used for the GUI are organised in a tree-like structure. In this structure, the application window (MainFrame) is the top-most object. In this window, arbitrary elements may be added and positioned. These elements are, for example, menus, images, or a "JPanel". HVS uses a separate window for each visualisation so that these windows may be resized and moved at will. Furthermore, HVS requires its own so-called "MainFrame". In the MainFrame, all prerequisites are initialised and the basic GUI is set up. The perfect place to position HVTE is between the MainFrame and the different visualisations. HVTE thus replaces the built-in GUI of HVS.

HVTE has its own "main" class which initialises the GUI and the database. The GUI structure of HVTE was kept simple. The list containing the 16 test cases (corresponding to the 16 rows in the Latin square) is on the left-hand side of the screen. The top part of the GUI is used for the task field and the answer field with the corresponding button. The remaining part of the GUI is used for displaying the visualisations (see Figure 7.4).

The visualisations are always opened maximised to fill the available screen space. The hierarchy is always displayed in the initial state (showing the root directory open) after opening the visualisation. Figures 7.8, 7.9, 7.10 and 7.11 show the hierarchy in the different visualisations in their initial state as opened after task acknowledgement. The hierarchy is only displayed when the user has to work with it (between acknowledgement of the task description and confirmation of the answer). It is possible to continue a test for instance after system malfunction or any other reason. After restart of HVTE, the test can be continued where it was interrupted.

7.3.1 Test Flow Logic

The class BrowserLogic controls the behaviour of HVTE. Each request sent by the user is processed here. The rest of the application handles the GUI and the visualisations. The first implementations of HVTE were closely coupled with a HTML-browser. The pages contained HTML forms for the user to interact with HVTE. Therefore, each request was wrapped within a URI (Uniform Resource Identifier). This concept was further used in the final version of HVTE as it proved to be working well. An extra benefit is that the application may be used as a server side application.

The class BrowserLogic extracts all necessary data from the URI. The parameters used are testid, testcaseid, taskid, sequence, action, and answer, as shown in Table 7.2. Figure 7.3 shows the workflow through a test.

Parameter	**Description**
Test_ID	Defines which test is currently active.
TestCase_ID	Defines the test case currently active.
Task_ID	Defines which task is currently active.
Sequence	The sequence corresponds to the question number according to the test case.
Action	The action states how the request should be processed.
Answer	The answer holds the plain text answer typed in by the user.

Table 7.2: Parameters used by the class BrowserLogic.

The six parameters listed in Table 7.2 have the following meaning:

- The parameter **action** defines how to process the request. *Starttest* starts the test after clicking on the test case number. *Dotest* performs the test once it has been started. This difference is necessary for the case where a test is interrupted. After restart, the test is continued where it was interrupted. *Saveanswers* saves the typed answer in the database. *Endtest* ends the test and displays a message to the user in the task field.

- The parameter **answer** holds the answer which was typed by the user in plain text.

- The parameter **sequence** holds the task numbers for that particular test case, according to the Latin square. Every user has to perform 32 tasks in the order given by their randomly assigned test case.

- The parameter **Task_ID** holds the ID of the task currently being processed by the user.

- The parameter **TestCase_ID** defines the ID of the currently active test case. Only one test case can be active at a time. The test for a given case can only be run once, thus preventing a test from being overwritten.

- The parameter **Test_ID** allows for different tests. In this study, only one test was performed. With this parameter, different tests could be run in the testing environment. For instance, every second user could perform a different test without changing the test environment.

The class BrowserLogic generates a ResponseObject. The ResponseObject holds all necessary data for HVTE to display the task or open the next visualisation. The main parameters within the ResponseObject are a text to prompt to the user (the welcome message or the formatted task description), the visualisation to open (plugin name and descriptive name) and the hierarchy to use with the visualisation (again plugin name and descriptive name). This data is passed back to the desktop class which is responsible for the main parts of the GUI of HVTE. The desktop class is a nested "panel" within the HVS MainFrame. In HVTE, the HVS MainFrame is only used to initialise all dependencies and core objects.

The desktop class opens, hides, and closes the visualisation windows as defined by the data within the ResponseObject. In case of a change of hierarchy, the new hierarchy is loaded and instantiated. However, changing hierarchies requires long loading times.

7.3.2 Tests in Different Languages

HVTE is designed for use in different countries. Therefore, a means of language configuration is required. One language-specific element is the task description. These are stored in the database individually per test. Changing the task language is not a major effort. The more complex part is to provide status messages, menu items and buttons in different languages. In a simple configuration file, a key-to-value list is provided. The application is capable of replacing specific parts of the values to customise them. In the configuration file "globals.dat", the values of the key words and messages can be changed to other languages. The buttons and user messages can thus be easily customised.

7.3.3 Tweaking HVS within HVTE

Some visualisations within HVS use quite a large amount of memory, but do not release all of the memory. In this case, the garbage collection is not allowed to free the memory. This leads to extreme memory consumption, slowing down the visualisation rendering severely, and sometimes causing memory overflow and system instability. During a test, 32 visualisation windows are opened. Each visualisation has to be removed and deleted by HVTE after closing the visualisation window. However, it was not possible to remove all remnants from the memory. Thus, the Java virtual machine needed to be started with extra memory. For safety reasons, an initial 256MB of memory were assigned with a peak maximum of 512MB, thus maintaining stable execution of HVTE.

Opening and closing visualisation windows worked correctly, but the current view of the visualisation was not reset after closing the previous one. Thus, the following visualisation was rendered identically to the previous one and the hierarchy was not shown in its initial state. This behaviour was caused by the synchronisation feature of HVS. Therefore, this feature had to be deactivated globally and in each visualisation.

7.4 Using HVTE to Run a Test

The test flow and interaction with the database are shown in Figure 7.3 showing the workflow in the BrowserLogic class. Here, the steps through a typical test will be illustrated.

After staring HVTE, only the user list is visible. The task and answer field as well as the area where the visualisations open are empty (see Figure 7.4).

The participants drew their test case number from a box to ensure random assignment of a test case. After clicking at the test case number, a prompt asking whether the test should be started appears (see Figure 7.5).

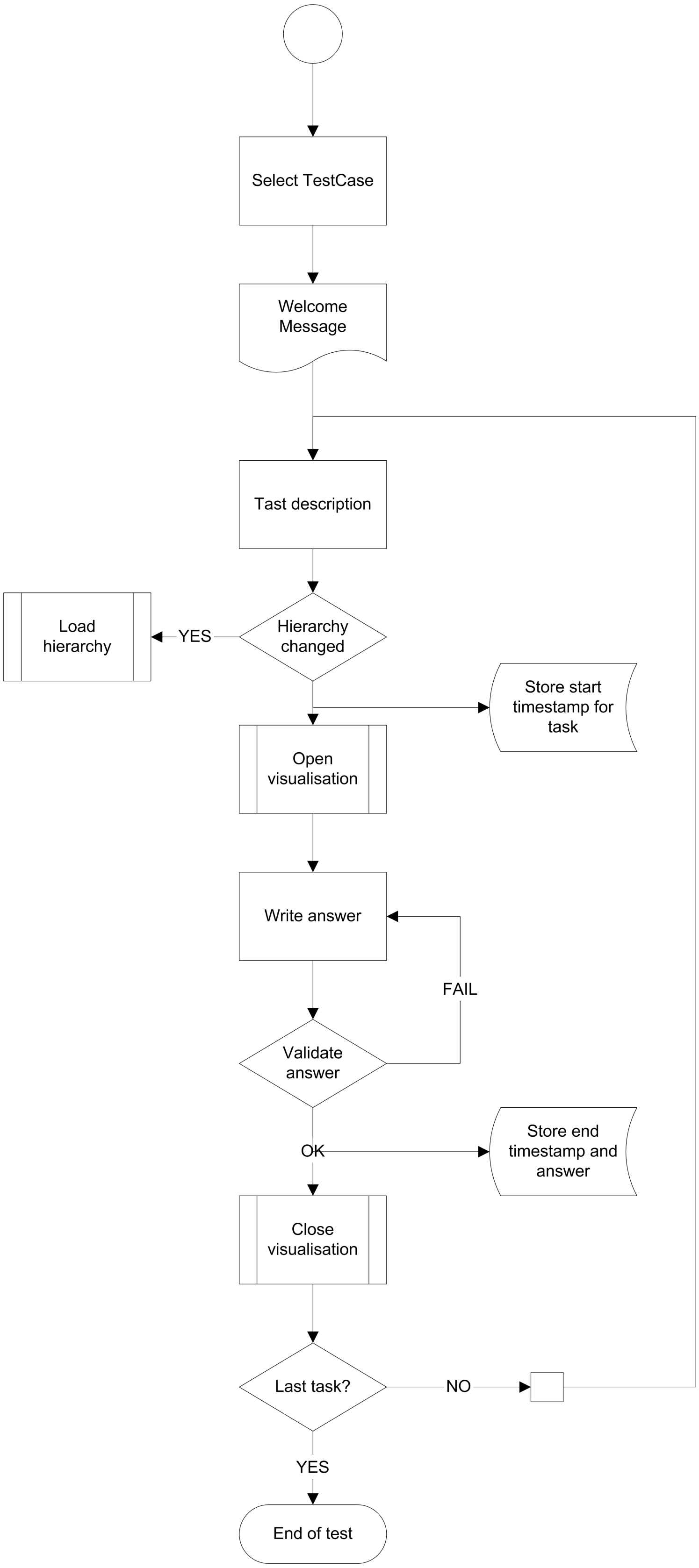


Figure 7.3: The flow logic used to run through a test in HVTE.

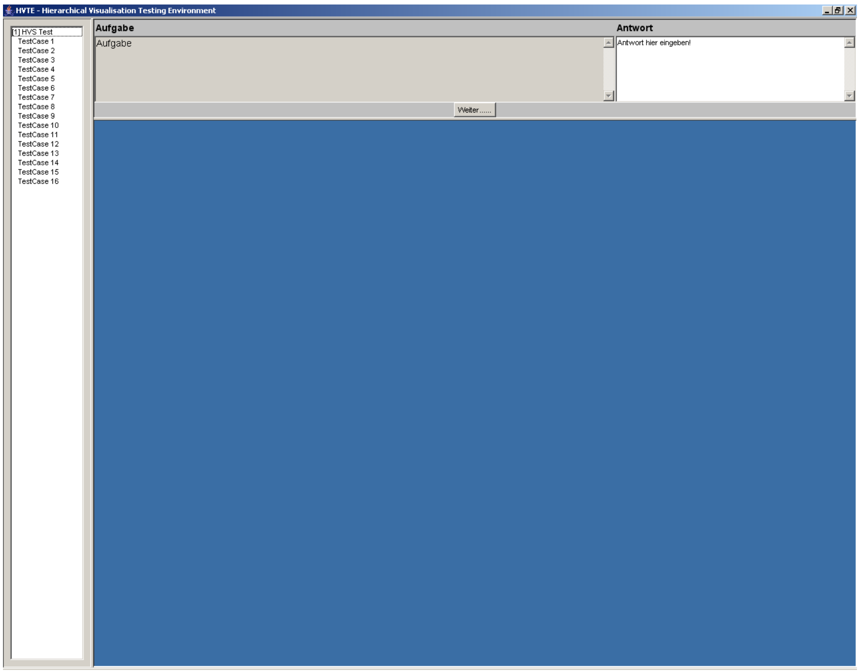


Figure 7.4: The HVTE start screen. The list of test cases is shown on the left.

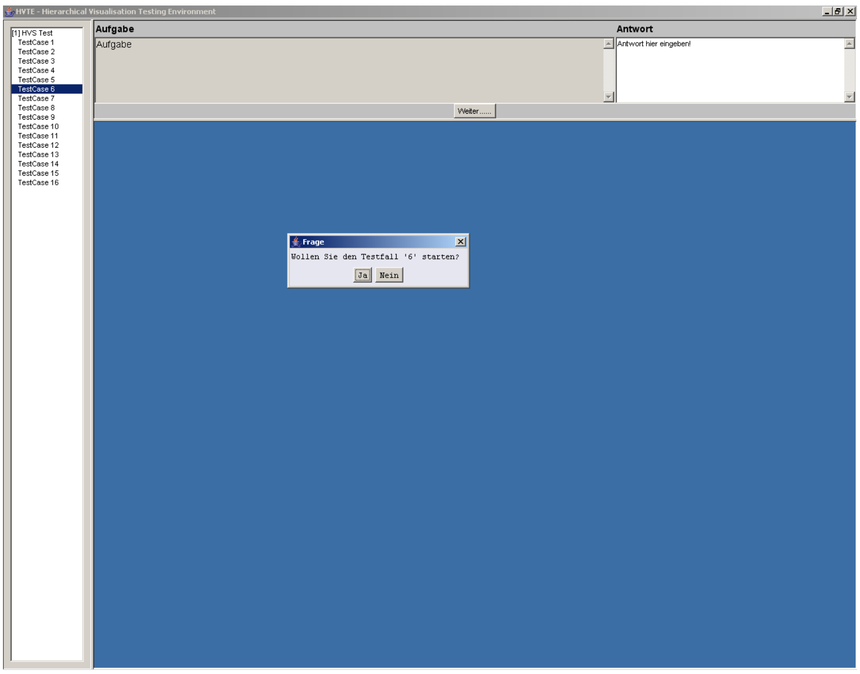


Figure 7.5: The HVTE pop up to confirm starting a test for a particular test case.

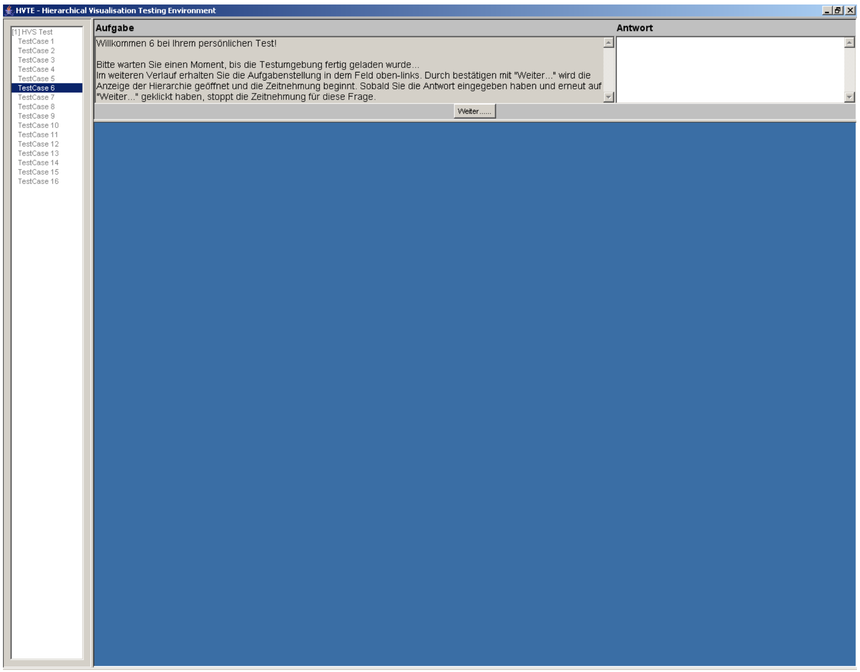


Figure 7.6: The HVTE welcome message and explanation.

A welcome message explaining the test appears in the task field (see Figure 7.6). It says: “Welcome (user number) to your personal test! Please wait a moment until the test environment has loaded. You will see the tasks in the top-left field. After clicking on “Continue. . . ”, the hierarchy will be opened and you may begin. When you have finished, type in the answer and click on “Continue. . . ”.”

After clicking on the “Continue. . . ” button, the welcome message disappears and the first task is shown. Before every block of tasks for a visualisation, a pop-up informing the user which visualisation would come next was shown (see Figure 7.7). The pop-up is closed by clicking on “OK”. After this, users could read the task and continue the test with the new visualisation.

The users had as much time as they needed to read the task. Only after clicking on the “Continue. . . ” button, thus acknowledging the task, did the visualisation appear and the timestamp was written (see Figure 7.8). Figures 7.9, 7.10 and 7.11 show the visualisations TreeView, TreeMap and Hyperbolic, respectively, in their initial states after the acknoweledgement of the task by the user.

Skipping of tasks was not possible. Users had to type in an answer before going on to the next task. If the users clicked on “Continue. . . ” without typing an answer, a pop-up reminding them to type an answer was shown (see Figure 7.12).

If, for any reasons, the HVTE terminated during a test, this test could be continued after restart. After starting HVTE again, a popup was shown (see Figure 7.13). It says: “The test case (case number) has not been finished yet. Continue test?” After clicking on “YES”, the test was resumed with the task the user was working on before termination. After finishing the test, a short thanking message is shown in the task field (see Figure 7.14).

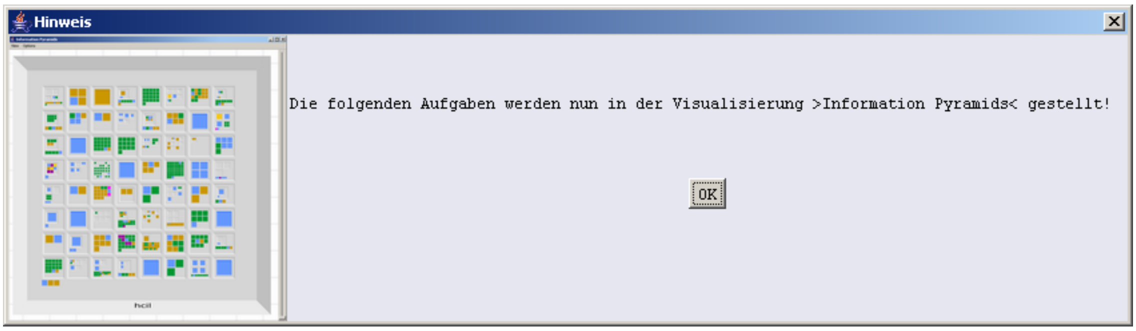


Figure 7.7: The HVTE pop-up informing about a new visualisation to be used for the next set of tasks.

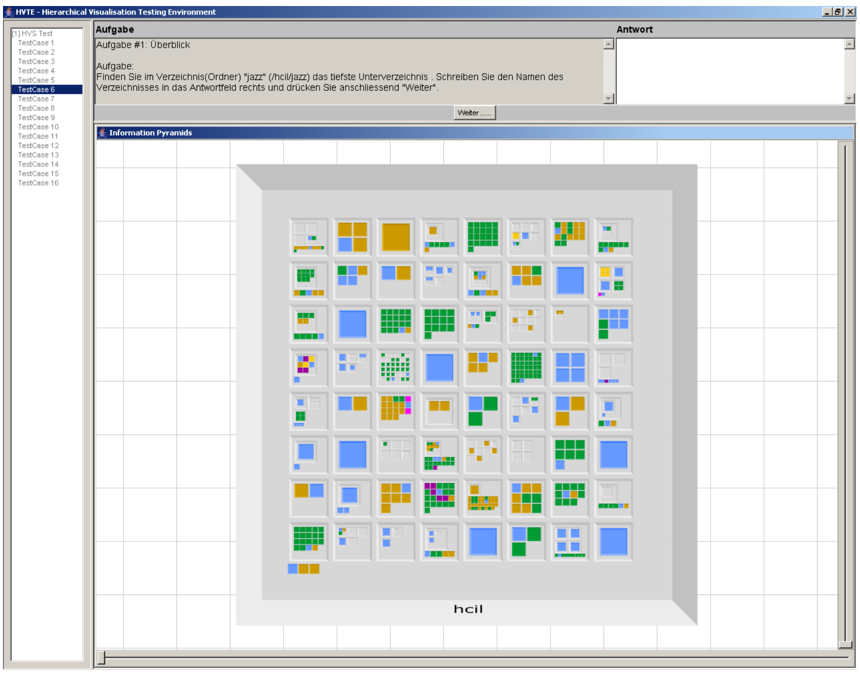


Figure 7.8: The first task for test case 6 in HVTE, showing Pyramids in its initial state.

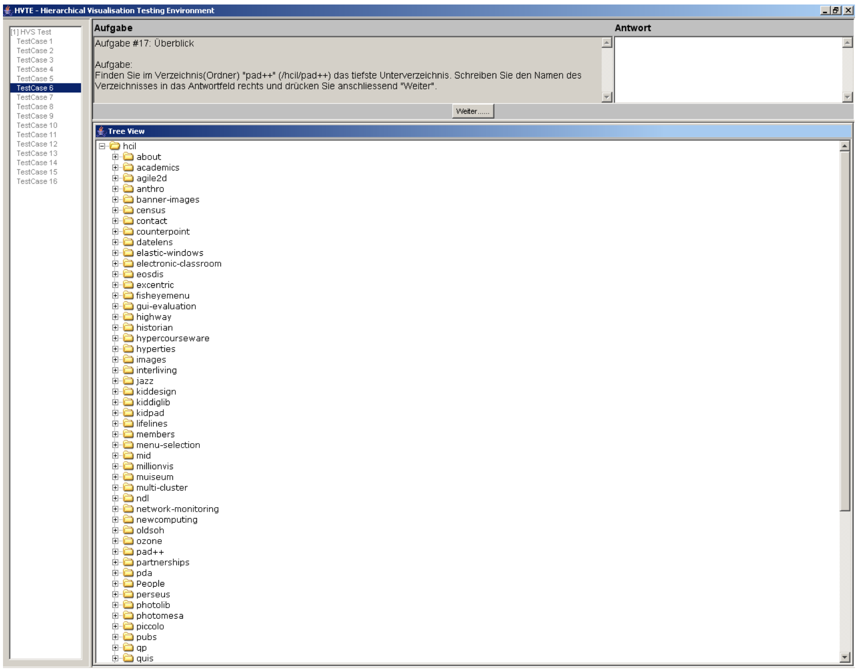


Figure 7.9: The TreeView visualisation in HVTE in its initial state.

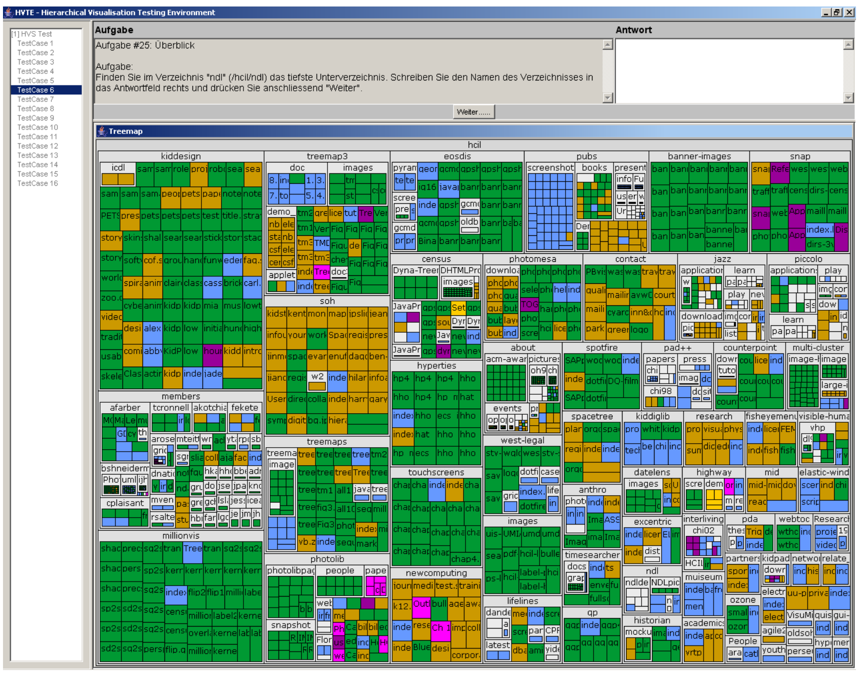


Figure 7.10: The TreeMap visualisation in HVTE in its initial state.

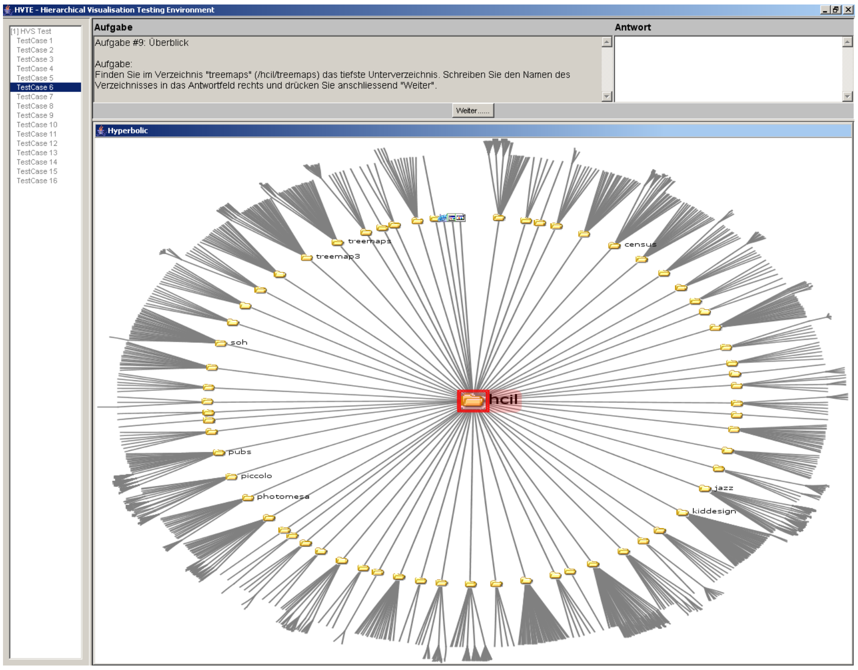


Figure 7.11: The Hyperbolic visualisation in HVTE in its initial state.

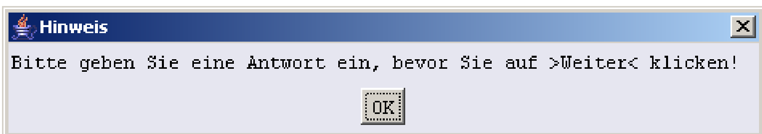


Figure 7.12: The HVTE prompt reminding the user to enter an answer.

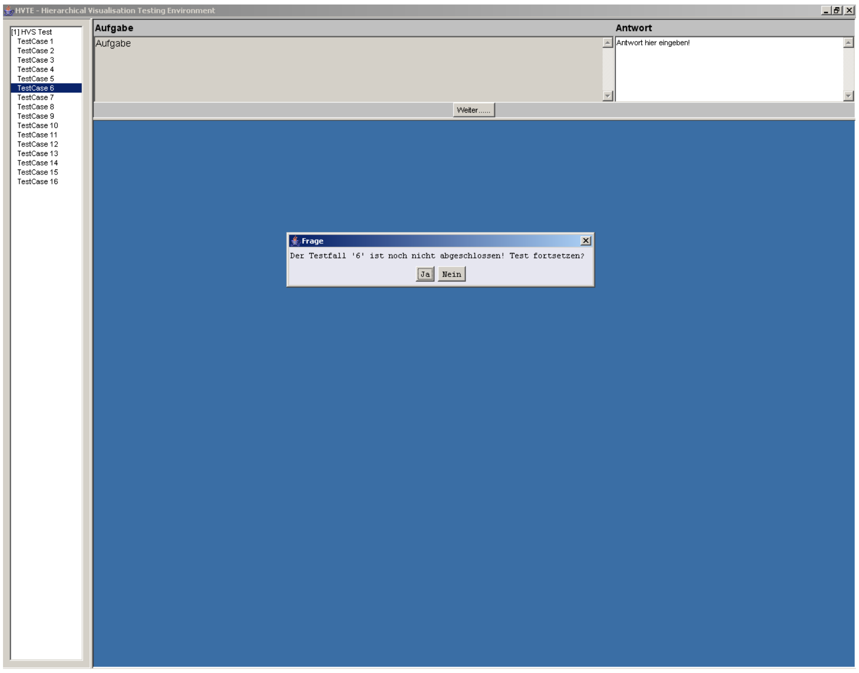


Figure 7.13: The HVTE pop-up asking if an interrupted test should be resumed.

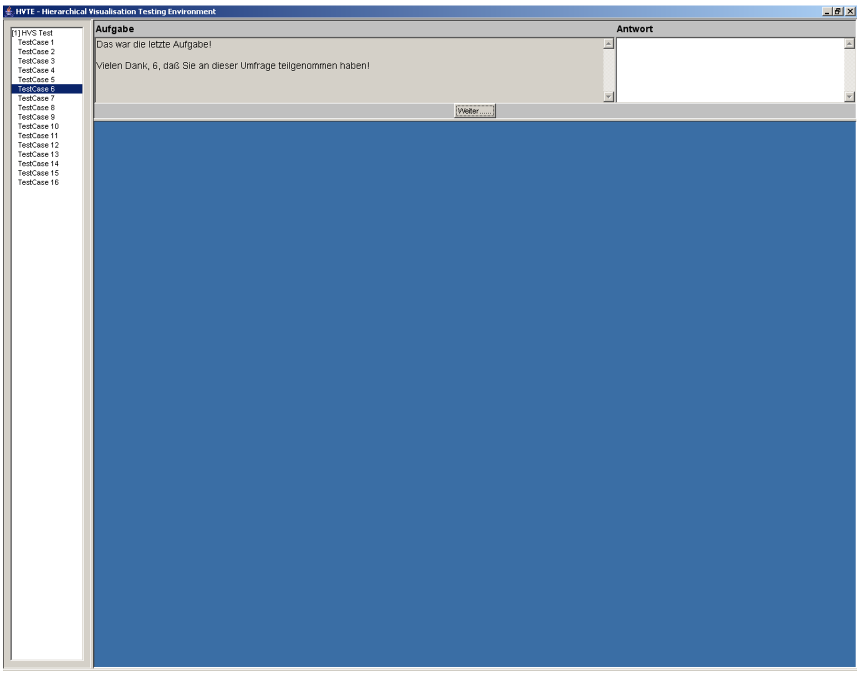


Figure 7.14: The HVTE thanking message, displayed at the end of the final task.

Chapter 8

Evaluating HVS

This thesis describes the first formal study of HVS. Some approaches realised in HVS also exist in other packages and have been evaluated in different studies (see Chapter 6). However, their implementation in HVS has not been studied in formal experiments before.

The tests were designed to compare individual visualisations and thus only one visualisation was available at a time for the users. Searching and filtering was also disabled for the tests. Only mouse navigation could be used to fulfil the tasks. Thus, the comparison of the visualisations was “fair”, in the sense of having to navigate without shortcuts. Additionally, participants’ subjective comments were recorded for design feedback.

8.1 User Profile

The target users for this study are experienced PC users working with large amounts of information, data, and hierarchies. In order to cover a broader target group, not only specialists in one area were invited. Possible target groups for information visualisation are students, university assistants, professors, and other scientific analysts.

After two piplot tests, the initial idea of having hundreds of students participating in the study was discarded due to time and space considerations. After deciding on the test design and the Latin square, the number of participants was fixed to 32 (thus, each of the 16 cases in the Latin square was used twice). Furthermore, this number of participants could be recruited from friends. The users were given small refreshments, if they wished.

All of the participants in this study were unpaid volunteers. Most were fellow students, friends and colleagues. Eighteen of the 32 participants have a technical education, such as computer science, electronics or chemical engineering. The average computer experience of all users is almost 13 years. They all have experience with management of large amounts of information.

Tree participants did not take the study seriously and their data was discarded. The task completion rate of these participants was much lower (8 and 9 incorrect answers) than other participants and their average error rate (1.96 errors per participant) was higher than other participants. Due to the flexible test planning, new test users were found to replace the data. The three new participants used exactly the same test sequences (from the Latin square) as the replaced users. Thus, every test sequence was run exactly twice (see Section 8.2.2 for details).

8.2 Test Design

For objective comparison of alternative information visualisations, controlled experiments (formal experiments mentioned in Section 5.3) are predominantly used. The controlled experiment used for this study was repeated measures with every user testing every visualisation. In a repeated measures design, the number of participants can be kept reasonable (32). Besides, any variations in user skills are removed. As all users tested all visualisations, they could voice their opinion about the visualisations and compare them to each other. A Windows Explorer-like collapsible tree viewer was used as a baseline. All users were familiar with this visualisation and used it during the study.

The independent variable in this study is the visualisation used. The dependent variables are the time to fulfil tasks and the personal ratings. The data measured in this experiment is the time to fulfil the tasks. Preference ratings were collected in a feedback questionnaire at the end of each test.

Data was gathered in an automated testing environment, the Hierarchical Visualisations Testing Environment (HVTE), and stored in a database (see Chapter 7). The whole test process was also captured on video and thus analysis of user behaviour may be possible as well. The data were evaluated using SPSS 12.0 for Windows. Analysis and results are presented in Chapter 9.

8.2.1 Visualisations

The four visualisations used in this study were: TreeMap, Hyperbolic, and Pyramids browser, using the TreeView as baseline. Due to the widespread use of Windows Explorer, it can be assumed that all participants are used to a TreeView style visualisation. Thus, the other visualisations, unknown to the test participants, can be compared to the baseline of a TreeView. Figures 8.1, 8.2, 8.3, and 8.4 show the visualisations TreeMap, TreeView, Hyperbolic, and Pyramids included in HVTE, respectively.

8.2.2 Test Cases

In order to counterbalance the visualisations and tasks used for each user, the Latin Square in Table 8.1 was used. Users drew a piece of paper with a number between 1 and 16 from a box in order to randomly assign each user to a test case.

With four visualisations and four task sets, there are 16 combinations, or test cases, as shown in Table 8.2. The four task sets are A, B, C and D. The task sets were designed to be equivalent in difficulty and consisted of eight tasks each. The tasks and their taxonomy are described in more detail in Section 8.4.

According to the Latin square, a user with test case number 12 tested the Hyperbolic browser with the task set C first, the TreeMap browser using the task set A, followed by the Pyramids visualisation with task set D and finished the test with TreeView browser and task set B.

8.2.3 Test Hierarchy

The hierarchy used in this study, logs_A_03-02-01_HCIL.xml, is the reduced hierarchy from the InfoVis2003 contest [Fekete and Plaisant, 2003a]. The hierarchies "File System and Usage Logs" originally proposed for the contest were very large having some 70 000 leaf

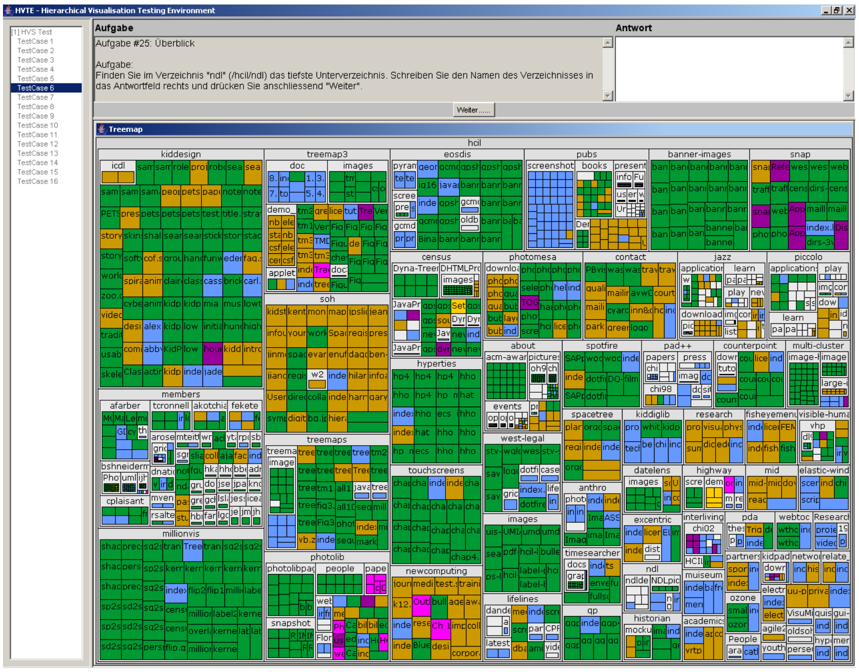


Figure 8.1: The Tree Map visualisation included in HVTE, as used in the study.

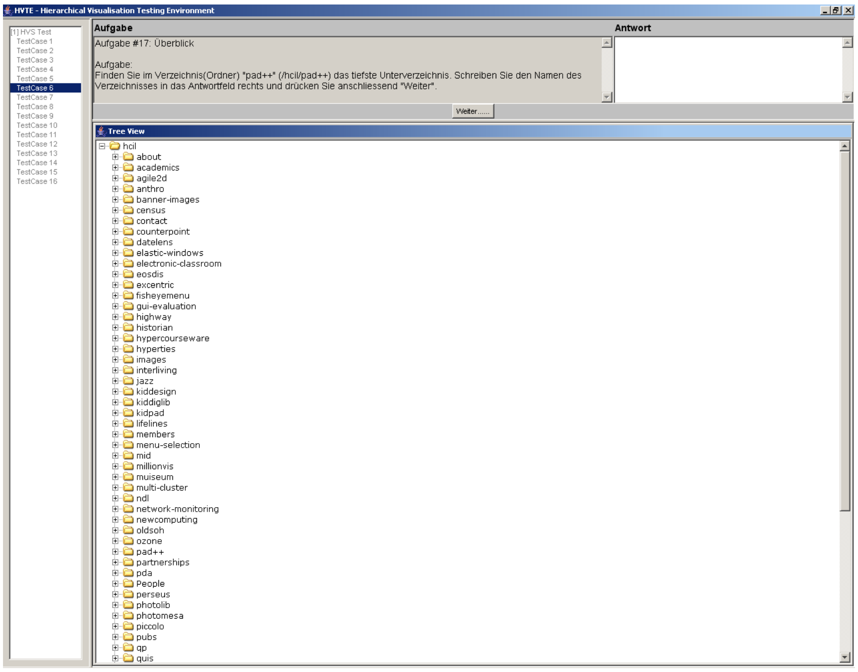


Figure 8.2: The Tree View visualisation included in HVTE, as used in the study.

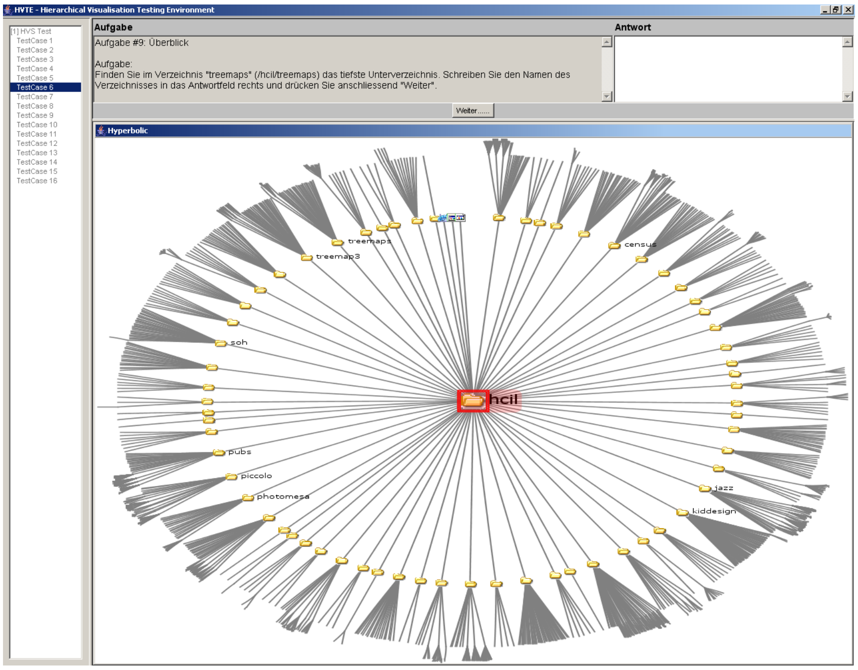


Figure 8.3: The Hyperbolic visualisation included in HVTE, as used in the study.

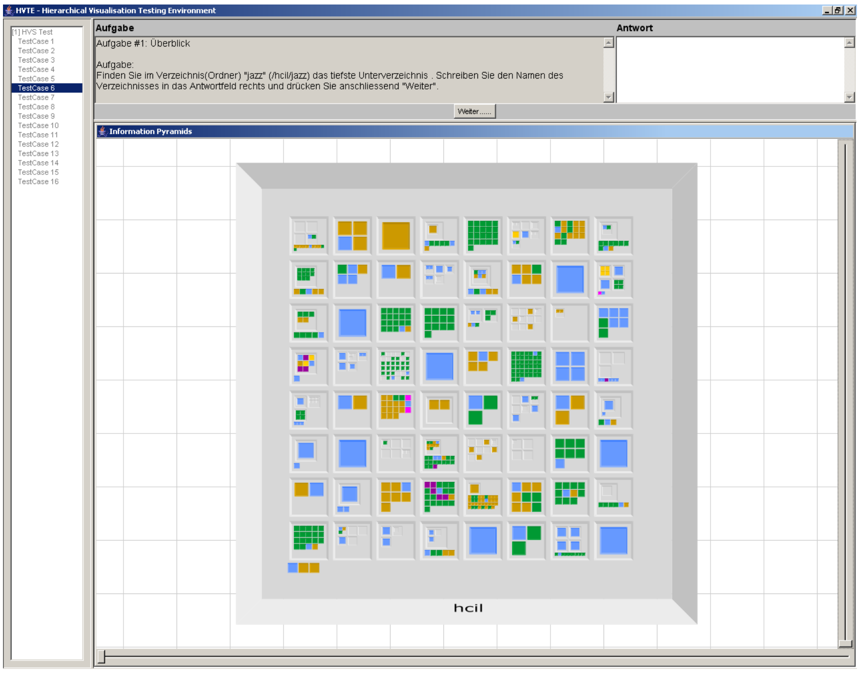


Figure 8.4: The Information Pyramids visualisation included in HVTE, as used in the study.

A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

Table 8.1: A 4×4 balanced Latin square.

Case	Configuration			
1	TreeView - A	Pyramids - B	TreeMap - C	Hyperbolic - D
2	TreeView - B	Pyramids - D	TreeMap - A	Hyperbolic - C
3	TreeView - D	Pyramids - C	TreeMap - B	Hyperbolic - A
4	TreeView - C	Pyramids - A	TreeMap - D	Hyperbolic - B
5	Pyramids - A	Hyperbolic - B	TreeView - C	TreeMap - D
6	Pyramids - B	Hyperbolic - D	TreeView - A	TreeMap - C
7	Pyramids - D	Hyperbolic - C	TreeView - B	TreeMap - A
8	Pyramids - C	Hyperbolic - A	TreeView - D	TreeMap - B
9	Hyperbolic - A	TreeMap - B	Pyramids - C	TreeView - D
10	Hyperbolic - B	TreeMap - D	Pyramids - A	TreeView - C
11	Hyperbolic - D	TreeMap - C	Pyramids - B	TreeView - A
12	Hyperbolic - C	TreeMap - A	Pyramids - D	TreeView - B
13	TreeMap - A	TreeView - B	Hyperbolic - C	Pyramids - D
14	TreeMap - B	TreeView - D	Hyperbolic - A	Pyramids - C
15	TreeMap - D	TreeView - C	Hyperbolic - B	Pyramids - A
16	TreeMap - C	TreeView - A	Hyperbolic - D	Pyramids - B

Table 8.2: Interleaving two 4×4 Latin squares produces the 16 test cases.

nodes. This hierarchy has been reduced to 3239 leaf nodes as proposed by the contest supervisors by using only the *HCIL* subtree. In order to obtain similar depths for the tasks, one directory containing one file was added to this hierarchy. In the directory “treemaps”, subdirectory “images”, a subdirectory called “screenshots”, including a file “screenshot1.jpg” was added. This was necessary for the tasks asking for the deepest subdirectory in a given directory. The hierarchy represents the file system of the University of Maryland Computer Science Department website (only the public files that are accessible via the web are included). Leaf nodes are files, subtrees are directories. Each node has a set of attributes. The XML file of this hierarchy is provided in the TreeML format created for the InfoVis2003 contest. The Hierarchical Visualisation System (HVS) can read the TreeML format, so no further changes were necessary.

8.3 Test Procedure

The initially estimated duration of every task of approximately 1-2 minutes from start to finish proved correct. Before starting the study, the users were asked to fill in a background questionnaire to gather their characteristics (such as experience with a PC or information maintenance). Users were asked to sign a non-disclosure and consent form. The forms and

Steps in the Test Procedure	Average Time
Consent and non-disclosure form	1 min.
Background questionnaire	5 min.
Introduction to the visualisations	15 min.
Exploration of the visualisations by the user	5–10 min.
Study - working on the tasks	45 min.
Feedback questionnaire	5–10 min.

Table 8.3: Procedure steps

questionnaires used can be found in Appendix A.

Users were given 15 minutes introduction to the visualisations, before exploring the visualisations on their own for approximately 10 minutes. The basic idea behind the visualisation and how it displays hierarchies was explained, with special emphasis on the representation and differences between files and directories. Only the interaction possibilities which could be used during the study were shown and explained to the users. The introduction was conducted in the HVS environment. During this introduction, users could ask questions about the system. No questions and help were granted during the tasks themselves. The study was not a thinking-aloud test, but the users were told to comment (or not) their actions and feelings freely. After the users had become familiar with the system, the test was started. The test itself was conducted in the Hierarchical Visualisation Testing Environment (see Chapter 7), especially designed and implemented for this study. One complete test took approximately 1.5 hours from start to finish. The complete procedure, from filling out the forms at the beginning to the feedback questionnaire at the end was captured on video.

8.3.1 Feedback Questionnaire

After finishing all tasks, the users were asked to fill out a subjective concluding questionnaire. This questionnaire should document any preferences for the different visualisations. Every visualisation was rated by the users on a seven point Likert scale regarding the following statements:

- overview
- operability
- intuitive
- usable
- understandable
- logical
- useful
- orientation
- navigation

A seven point scale (3-2-1-0-1-2-3) ranging from “very bad” to “very good” was used. For the complete questionnaires, see Appendix A. At the end of the questionnaire, users were asked to choose one visualisation which was “the best” for each of the nine factors above.

8.4 Task Taxonomy

Defining the appropriate test tasks and data set is both tedious and crucial. The tasks as well as the data set should be adapted to the visualisation used in the test. Using a task taxonomy defined in other, similar studies makes the present study comparable to them.

During the survey of related work (see Chapter 6), different task taxonomies were used in other studies. The task categories can broadly be divided into structure-related and attribute-related tasks. Structure-related tasks can be used to investigate the visualisation's ability to represent the structure of the hierarchy, such as the depth or the number of children of individual nodes. Attribute-related tasks concentrate on the attributes of the nodes in the hierarchy, such as file type, age, or size.

The tasks used in this study were chosen to be typical tasks a user would perform on data in a file system. The task taxonomy is taken from [Wiss and Carr, 1999]. Additionally, overview tasks were designed to answer general questions about the structure of the hierarchy. There are two main task categories: global or overview tasks, and local tasks. Local tasks themselves are divided into three subcategories: search, count, and compare. Thus, there were four categories of tasks: overview, search, count, and compare tasks.

Global or overview tasks should show the visualisation's ability to communicate the overall structure to the users. Local tasks should show how easy (or difficult) it is for the users to perform everyday tasks in file systems such as finding a file by name or comparing directories. As the hierarchy was unknown to the users in advance, complete paths (beginning with the root) were given in the tasks.

Examples for *overview* tasks include:

- Find the deepest subtree of the directory y.
- Find the directory containing most subdirectories (find the directory having the maximum branching factor).

Examples of *local* tasks include:

- *Search:* Find a directory or a file, given the path.
- *Count:* How many subdirectories/ files are in directory x.
- *Compare:* Which directory has more subdirectories/ files: x or y?

There were two tasks from each category, yielding eight tasks in each task set. Every user performed 32 tasks in total, 8 with each browser. The tasks in the four different task sets are similar in type and hierarchy depth in order to balance the level of difficulty. The task sets A, B, C and D are given in Tables 8.4, 8.5, 8.6, and 8.7, respectively. Note that the actual study was run in German and used the German version of the tasks, which can be found in Appendix B.

Number	Category	Task	Answer
A1	Overview	Find the deepest subdirectory inside the directory "pad++" (/hcil/pad++). Write the name of this directory into the answer field to the right and then press "Continue...".	hcil/pad++/papers/chi-97-kidpad/images
A2	Overview	Find the directory inside "ndl" (/hcil/ndl) with the most subdirectories. Write the name of this directory into the answer field to the right and then press "Continue...".	hcil/ndl/ndldemo
A3	Search	Find the directory "yidemo" (/hcil/lifelines/yidemo). When you have found the directory, write "OK" or "found" into the answer field to the right and then press "Continue...".	OK
A4	Search	Find the file /hcil/treemaps/-treemap2000/images/banner-logo-large.gif. When you have found the file, write "OK" or "found" into the answer field to the right and then press "Continue...".	OK
A5	Count	Count the number of subdirectories directly inside the directory "/hcil/pubs". Write the answer into the answer field to the right and then press "Continue...".	4
A6	Count	Count the number of files directly inside the directory "/hcil/qp". Write the answer into the answer field to the right and then press "Continue...".	7
A7	Compare	Which directory has more direct subdirectories: "/hcil/about" or "/hcil/eosdis"? Write the answer into the answer field to the right and then press "Continue...".	hcil/census
A8	Compare	Which directory has more files directly inside: "/hcil/spotfire" or "/hcil/spacetree"? Write the answer into the answer field to the right and then press "Continue...".	hcil/images

Table 8.4: Task set A. These are the English translations of the tasks.

Number	Category	Task	Answer
B1	Overview	Find the deepest subdirectory inside the directory “jazz” (/hcil/ jazz). Write the name of this directory into the answer field to the right and then press “Continue. . . ”.	hcil/jazz/learn/papers/-chi-97-kidpad/images
B2	Overview	Find the directory inside “about” (/hcil/-about) with the most subdirectories. Write the name of this directory into the answer field to the right and then press “Continue. . . ”.	hcil/about/events
B3	Search	Find the directory “oldbinary” (/hcil/-eosdis/oldbinary). When you have found the directory, write “OK” or “found” into the answer field to the right and then press “Continue. . . ”.	OK
B4	Search	Find the file /hcil/ndl/-ndl_secure/draft11/home9.html. When you have found the file, write “OK” or “found” into the answer field to the right and then press “Continue. . . ”.	OK
B5	Count	Count the number of subdirectories directly inside the directory “/hcil/life-lines”. Write the answer into the answer field to the right and then press “Continue. . . ”.	5
B6	Count	Count the number of files directly inside the directory “/hcil/interliving”. Write the answer into the answer field to the right and then press “Continue. . . ”.	3
B7	Compare	Which directory has more direct subdirectories: “/hcil/census” or “/h-cil/treemap3”? Write the answer into the answer field to the right and then press “Continue. . . ”.	hcil/census
B8	Compare	Which directory has more files directly inside: “/hcil/about” or “/hcil/images”? Write the answer into the answer field to the right and then press “Continue. . . ”.	hcil/images

Table 8.5: Task set B. These are the English translations of the tasks.

Number	Category	Task	Answer
C1	Overview	Find the deepest subdirectory inside the directory "ndl" (/hcil/ ndl). Write the name of this directory into the answer field to the right and then press "Continue...".	hcil/ndl/ndldemo/-anita/new/invit
C2	Overview	Find the directory inside "pubs" (/hcil/-pubs) with the most subdirectories. Write the name of this directory into the answer field to the right and then press "Continue...".	hcil/pubs/presentations
C3	Search	Find the directory "large-image" (/hcil/-multi-cluster/large-image). When you have found the directory, write "OK" or "found" into the answer field to the right and then press "Continue...".	OK
C4	Search	Find the file /hcil/jazz/applications/-cosmosgame/cosmosgame.jpg. When you have found the file, write "OK" or "found" into the answer field to the right and then press "Continue...".	OK
C5	Count	Count the number of subdirectories directly inside the directory "/h-cil/treemaps". Write the answer into the answer field to the right and then press "Continue...".	3
C6	Count	Count the number of files directly inside the directory "/hcil/piccolo". Write the answer into the answer field to the right and then press "Continue...".	5
C7	Compare	Which directory has more direct subdirectories: "/hcil/lifelines" or "/hcil/-pad++"? Write the answer into the answer field to the right and then press "Continue...".	hcil/pad++
C8	Compare	Which directory has more files directly inside: "/hcil/census" or "/hcil/counter-point"? Write the answer into the answer field to the right and then press "Continue...".	hcil/census

Table 8.6: Task set C. These are the English translations of the tasks.

Number	Category	Task	Answer
D1	Overview	Find the deepest subdirectory inside the directory "treemaps" (/hcil/treemaps). Write the name of this directory into the answer field to the right and then press "Continue...".	hcil/treemaps/-treemap2000/images/-screenshots
D2	Overview	Find the directory inside "pad++" (/hcil/pad++) with the most subdirectories. Write the name of this directory into the answer field to the right and then press "Continue...".	hcil/pad++/papers
D3	Search	Find the directory "ara" (/hcil/People/-ara). When you have found the directory, write "OK" or "found" into the answer field to the right and then press "Continue...".	OK
D4	Search	Find the file /hcil/timesearchrer/docs/-graphics/averages.gif. When you have found the file, write "OK" or "found" into the answer field to the right and then press "Continue...".	OK
D5	Count	Count the number of subdirectories directly inside the directory "/hcil/pad++". Write the answer into the answer field to the right and then press "Continue...".	7
D6	Count	Count the number of files directly inside the directory "/hcil/academics". Write the answer into the answer field to the right and then press "Continue...".	4
D7	Compare	Which directory has more direct subdirectories: "/hcil/highway" or "/hcil/photolib" ? Write the answer into the answer field to the right and then press "Continue...".	hcil/photolib
D8	Compare	Which directory has more files directly inside: "/hcil/members" or "/hcil/westlegal" ? Write the answer into the answer field to the right and then press "Continue...".	hcil/members

Table 8.7: Task set D. These are the English translations of the tasks.

Processor	AMD Athlon 1.6 GHz
RAM	512 MB
Monitor	Fujitsu Siemens 17”
Resolution	1280x1024 @ 32 bit colour
Keyboard	Logitech Internet Navigator
Mouse	Logitech Cordless Click Plus Optical
Operating System	Windows XP Proffesional

Table 8.8: The characteristics of the test PC.

8.5 Test Environment

For organisational reasons, the tests were conducted at the author’s home. With this solution, booking an equipped room at the university for a longer period of time was not necessary. The familiar home environment probably helped the participants feel more comfortable than being in an “official” usability laboratory. The capturing equipment (digital video camera and accessories) was provided by the university institute. All tests were performed on the same PC under the same conditions. The characteristics of the test PC can be seen in Table 8.8. The timetable could be adjusted to participant’s needs, thus also late evening tests or weekend tests were possible. This flexibility for testing times definitely facilitated the recruitment of participants, as many of them were employed and could not participate during normal working hours.

The test facility (see Figure 8.5) consisted of a desk with the monitor, keyboard, mouse and mirror. The camera was mounted on a tripod and recorded the monitor and the user’s reflection in the mirror. A printout with a short overview of the navigation possibilities of the four visualisations was available. A separate microphone was used to capture the sound. A Canon MV600 digital video camera was used for the video recording.

8.6 Pilot Tests

Before the real test, a pre-pilot test and further pilot tests were planned and conducted in order to find and remove any possible problems. There was one pre-pilot test which showed some problems caused by the Pyramids visualisation. Due to the steadily increasing memory usage, the whole system was getting slower and slower. A system restart helped finish the test. Since HVTE monitors whether a test has already been accomplished or not, the test could be finished with no further problems. The memory requirements of the Pyramids visualisation were met for the usability tests by assigning a larger memory space for HVS (256 MB).

Further pilot tests were conducted after this correction. The user participating in the first pilot test provided some valuable comments regarding the feedback questionnaire. The feedback questionnaire was redesigned to cover more aspects and provide space for user comments. Further pilot tests revealed a rather unclear task definition in some cases, especially regarding the number of subdirectories or files. Some users misinterpreted the question and started counting all the files or subdirectories down to the last level. The addition of the word “direct”, before files or directories in the task solved this problem. Additionally, an example of direct files or subdirectories of a directory was given during the introduction. Further pilot tests ran with no problems, so the actual study could be

Figure 8.5: The test environment used in the study.

conducted.

8.7 Actual Study

The actual study took several weeks to conduct, due to the difficulty of recruiting participants. All participants were volunteers from the circle of friends and colleagues. Some could only participate in the evenings or during weekend. If several tests were conducted on one day, two-hour intervals were reserved for each test to provide a time buffer.

After arrival, users were offered small refreshments and were assured, that it was not them who was going to be tested. They were told that all explanations were be given during the study itself. After they felt comfortable, the study could be started. The procedure for each test was as described in Section 8.3:

1. Start the video recording.
2. Greet the user and explain the aim of the study and its procedure.
3. Fill out the consent form and the background questionnaire.
4. Introduce the four visualisations and the navigation.
5. Exploration of the visualisations by the users, answering any questions.
6. When the users felt ready, the testing environment was launched.
7. Explain the testing environment (HVTE) and the interaction.

8. Users were told to enter whatever answer they thought was correct and to work at their own pace.
9. Drawing the case number for that particular user from the box.
10. Starting the test itself.
11. After the test, fill out the feedback questionnaire.
12. Thank the user and stop the video recording.

Chapter 9

Results and Discussion

The task completion times were recorded in the database and could be directly read into an Excel file. The ratings and preferences were filled by the users into a paper questionnaire and then typed into an Excel file for analysis. Due to problems with three participants, new test were run and their data replaced (see Section 8.1). The data presented here contains the replaced data. The statistical analysis of collected data was largely performed using SPSS 12.0 for Windows. All tests of statistical significance were performed at the level $p < 0.05$.

9.1 User Background Statistics

Of the 32 participants for which the data was analysed, 18 were male and 14 female. The median age was 29 years (ranging from 19 to 65 years). 18 participants had a technical education: 10 participants in computer science and/or electronics, five in chemical engineering, and three in civil engineering. 16 participants were university students, 12 participants had university degrees and two participants had a doctorate. The median of the PC experience of the participants was 12 years (ranging from 7 to 20 years). Mean PC working time per week was 29.72 hours (median = 27.5, ranging from 1 to 65 hours). 29 of the 32 participants used Windows and Explorer. Only three used Linux and command line regularly. Six participants perform information retrieval daily, seven weekly, and five monthly. Only five users have participated in a usability study before. Table 9.1 summarises user background statistics.

9.2 Task Completion Times (Efficiency)

The task completion times collected during the tests were recorded in ms and following the Latin square. For analysis, the datasets were reorganised to be grouped by visualisation and task. The corresponding tasks from the different sets were combined to form tasks one to eight yielding eight tables. This means that all first tasks (whether A1, B1, C1 or D1) for TreeView were combined into one table column for TreeView. Due to this rearrangement, the rows of the tables can not be assigned to specific test cases or users. Table 9.2 shows the resulting table for task one. All other tables for tasks two to eight were built correspondingly.

For better readability, following notations will be used: $\approx$ means no statistically significant differences, $<$ means statistically significant differences (at the level $p < 0.05$) between

Gender	18 men	14 women
Age	median=29	range 19-68
Seeing aid	glasses: 11	contact lenses: 4
Education	16 students	12 graduated
PC practice	median=12 years	range 7-20 years
PC usage	median=27.5 hours/week	range 1-65 hours/week
Operating system	Windows: 29	Linux: 3
File Management	Explorer: 29	command line: 3
Most data	home: 11	work: 14
Information retrieval	daily: 6	weekly: 7

Table 9.1: The user background statistics.

the browsers. The browser on the left of $<$ is thus significantly faster than the browser on the right. Additionally, average times in seconds are given in brackets. The notation:

Hyperbolic(87.6s) $\approx$ TreeMap(92.8s) $\approx$ TreeView(102.2s) $\approx$ Pyramids(110.8s)

means that Hyperbolic browser was fastest, followed by TreeMap, followed by TreeView, followed by Pyramids, but none was significantly faster than other. Similarly, the notation:

TreeMap(20.0s) $<$ Hyperbolic(33.3s)

means that TreeMap was significantly faster than Hyperbolic browser.

The following steps were conducted in the analysis (based on [Field and Hole, 2003]):

1. Descriptive statistic only looks at the data and summarizes them. It does not transform the data.

- *Frequency statistics* summarise the data in a table.
- *Histograms* show the distribution of data. General tendencies and deviation from a normal distribution can be seen.
- *Error bars* show the mean values together with the standard deviation. The whole range of values can be seen at a glance.
- *Check for normality*: the Kolmogorov-Smirnov test is used to test whether the distribution of the obtained data is normal. Actually, the difference to a normal distribution is calculated. If the difference to a normal distribution is significant, then the data is not normally distributed. If $Sig. < 0.05$, the tested distribution is significantly different from a normal distribution, thus not normal. If $Sig. > 0.05$, there are no significant differences between the tested and the normal distribution. Thus, the tested data is normally distributed. The inferential statistics which can be used depend upon whether the data is normally distributed or not.

2. Inferential statistics are used to test hypothesis and look for statistically significant differences in the data.

- If the distribution is normal, the *one-way repeated measures ANOVA* test is used. It is used for analysing data from three or more groups in a repeated measures design (every user testing each condition). The sphericity (variances of differences between groups are equal) using Mauchly's test must be checked. If $p < 0.05$, the assumption

	TreeView	Pyramids	TreeMap	Hyperbolic
	68,7	152,6	49,4	83,4
	42,2	201,5	122,3	118,2
	18,3	93,1	192,1	75,1
	89,9	100,5	48,2	269,2
	41,2	176,0	74,7	28,4
	63,8	38,8	19,7	398,6
	56,8	54,8	72,6	40,9
	109,6	134,9	67,7	39,4
	49,3	33,1	199,1	243,1
	36,3	45,7	98,3	133,9
	112,5	38,8	124,5	54,5
	85,3	56,4	89,0	41,9
	78,3	146,1	78,7	87,0
	36,3	85,9	33,3	37,3
	57,5	55,8	48,2	46,1
	213,0	62,6	55,6	108,4
	110,2	247,6	94,5	92,4
	46,4	189,1	45,8	26,0
	53,8	132,5	36,0	29,8
	84,5	161,0	141,3	19,5
	160,7	125,4	55,8	38,5
	89,6	94,8	55,4	37,3
	142,1	41,9	476,2	54,7
	169,9	50,0	51,6	44,4
	81,9	473,9	79,7	34,7
	195,3	77,7	54,1	45,7
	48,4	153,1	37,8	38,2
	633,0	98,3	46,8	72,2
	93,8	51,3	63,3	52,7
	33,3	53,4	51,3	221,0
	25,0	77,5	238,8	43,3
	144,0	41,3	68,1	148,2
Mean	102,2	110,8	92,8	87,6
Std. Dev.	108,9	86,8	86,4	85,1

Table 9.2: Table with reordered task completion times for task one. Times are given in seconds to one decimal place.

of sphericity is not met (there are some variances not equal). If $p > 0.05$, the assumption of sphericity is met, and ANOVA can be applied without further corrections. If the ANOVA results in the value of $Sig. < 0.05$, then the overall differences between the conditions are significant. If $Sig. > 0.05$, then the overall differences are not significant. However, the result only states that there are differences, not where they are. In order to find the differences, pairwise comparisons and a paired T-test are used.

A *paired T-test* is used to find differences between two categories where all users contributed to both categories. The result of the test is the value Sig. If $Sig. < 0.05$, the differences are significant; if $Sig. > 0.05$, the differences are not significant.

- If the distribution is not normal, the *Friedman test* is used, which is equivalent to a one-way repeated measures ANOVA, but for non-normally distributed data. If the significance value (Sig.) is less than 0.05, there are significant differences. Otherwise ($Sig. > 0.05$), the differences are not significant. However, this result only finds overall differences, not where they are. In order to find the differences, a Wilcoxon test is used.

  A *Wilcoxon test* corresponds to the T-test, but is used for not normally distributed data. If the Sig. value is less than 0.05, the differences are significant. Otherwise ($Sig. > 0.05$), the differences are not significant.

The step-by-step data analysis using SPSS 12.0 with screenshots is shown in Appendix C.

9.2.1 Task 1 (Overview Task): Deepest Subdirectory

The mean task completion times (see Table 9.2) from left (fast) to right (slow) for task 1 - finding the deepest subdirectory were as follows:

Hyperbolic(87.69s) $\approx$ TreeMap(92.8s) $\approx$ TreeView(102.2s) $\approx$ Pyramids(110.8s)

The histograms in Figure 9.1 show that the distributions were most probably not normal due to the left skew. This was proved by the Kolmogorov-Smirnov test of normality (for all visualisaitons, $p < 0.05$). The Pyramids visualisation was the slowest, followed by TreeView and TreeMap, with Hyperbolic being the fastest. A Friedman test was performed to look for significant differences between the visualisations. The value of Sig. is 0.055, meaning there were no overall significant differences between the visualisations. Furthermore, a pairwise Wilcoxon test showed no significant differences.

The graph-like layout of the Hyperbolic visualisation showing all levels of the hierarchy seemingly helped users find the deepest subdirectory faster. However, none of the visualisations were significantly faster or slower than the others.

9.2.2 Task 2 (Overview Task): Branching Factor

The ordering base don mean task completion times (see Table 9.3) from left (fast) to right (slow) for this task is:

TreeMap(52.9s) $\approx$ Pyramids(57.8s) $\approx$ Hyperbolic(58.5s) $\approx$ TreeView(61.1s)

From the histograms in Figure 9.3 it appears that the distributions could be normal. Only the data for Hyperbolic did not seem to be normally distributed. The Kolmogorov-Smirnov test confirmed this ($Hyp : D(32) = .228, p < .05$). The mean task completion times were shortest for TreeMap, followed by Pyramids, then Hyperbolic and the longest for

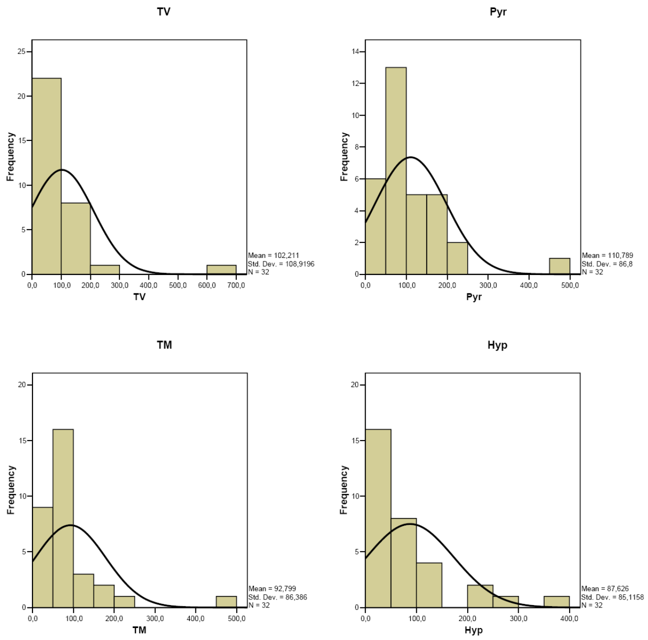


Figure 9.1: The task completion times histograms for task 1.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-,841[a]	-,598[b]	-1,159[b]	-1,515[b]	-1,515[b]	-,860[b]
Asymp. Sig. (2-tailed)	,400	,550	,246	,130	,130	,390

a. Based on negative ranks.
b. Based on positive ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.2: The results of the Wilcoxon test for task completion times for task 1.

	TreeView	Pyramids	TreeMap	Hyperbolic
	73,4	25,7	56,9	41,5
	42,4	38,6	74,0	134,2
	27,8	37,2	27,5	129,0
	89,2	65,9	21,3	76,5
	28,1	143,4	18,4	48,2
	60,1	24,5	19,4	37,8
	67,3	94,9	23,3	50,8
	136,7	61,9	53,0	50,0
	45,7	92,5	44,7	52,4
	91,3	58,6	40,3	78,3
	60,7	25,5	80,8	107,0
	47,3	41,5	84,7	38,6
	33,5	19,3	25,5	65,1
	17,8	72,8	37,6	46,5
	87,7	87,8	96,5	47,1
	105,8	63,8	17,9	40,4
	29,1	49,4	66,5	62,5
	42,5	106,3	46,8	60,5
	24,8	65,8	33,8	24,7
	127,0	108,1	57,6	16,8
	70,7	33,4	84,1	35,0
	26,5	35,5	47,7	33,6
	50,2	22,7	120,0	19,8
	102,3	37,2	98,0	49,6
	93,0	97,9	59,6	44,1
	75,0	62,8	71,1	41,8
	24,8	47,2	29,0	25,3
	76,7	40,3	55,7	50,7
	44,8	26,6	63,2	33,9
	70,5	40,5	32,0	125,2
	55,0	44,5	54,5	65,0
	26,5	79,0	51,9	141,4
Av	61,1	57,8	52,9	58,5
Std Dev	31,3	30,4	26,1	33,6

Table 9.3: Table with reordered task completion times for task two. Times are given in seconds to one decimal place.

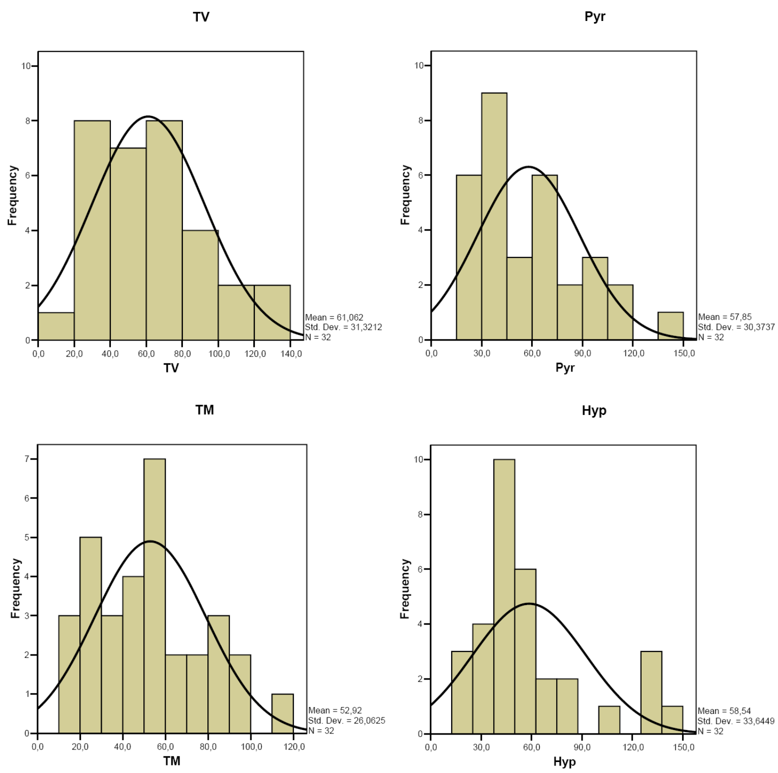


Figure 9.3: The task completion times histograms for task 2.

TreeView. In order to find any significant differences in task completion times between the visualisations, a Friedman test was performed. The Friedman test (rather than ANOVA) was performed due to the not normal distribution of the Hyperbolic data. Friedman test did not yield significant differences ($\chi^2(1.238), p > .05$). Additionally, a Wilcoxon test was performed which proved no significant differences.

For the task of finding the directory containing most direct subdirectories, the visualisations showing the hierarchy structure at a glance seem to be faster. Apparently, the need to open all directories first to see their content makes the TreeView perform slower. These differences in task completion times are not significant.

9.2.3 Task 3 (Search Task): Find Directory

The histograms in Figure 9.5 showed bipolar behaviour of the data, thus non-normal distributions were assumed. The Kolmogorov-Smirnov test verified this assumption (for all visualisations, $p < .05$). The mean task completion times (see Table 9.4) were as follows (the fastest leftmost):

TreeMap(36.0s) $\approx$ TreeView(36.5s) $\approx$ Hyperbolic(45.8s) $\approx$ Pyramids(47.3s)

The Friedman test showed no overall significant differences ($\chi^2(1.275), p > .05$). The Wilcoxon test did not yield significant differences between the pairs of visualisations.

Though not alphabetically ordered, the fastest visualisation for this task was the TreeMap. Apparently, the insight into deeper levels of the hierarchy is a major advantage. Pyramids performed rather poorly for this task. Much time was needed to navigate to find the correct directory. Due to the large hierarchy, the names of the directories and

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-,823[a]	-,813[a]	-,636[a]	-,561[a]	-,187[a]	-,524[b]
Asymp. Sig. (2-tailed)	,411	,416	,525	,575	,852	,601

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.4: The results of the Wilcoxon test for task completion times for task 2.

	TreeView	Pyramids	TreeMap	Hyperbolic
	49,9	64,0	11,6	30,7
	33,7	12,9	15,3	60,3
	16,7	52,1	36,8	191,0
	19,6	51,3	13,6	106,8
	116,7	345,6	24,8	8,5
	24,0	32,4	6,8	9,6
	28,3	17,7	17,2	26,3
	21,3	23,4	15,7	26,3
	79,7	8,8	37,8	124,7
	19,4	136,1	31,7	27,5
	35,2	13,2	46,7	61,6
	21,2	13,4	49,6	15,5
	9,5	8,5	20,9	165,6
	17,8	78,9	34,3	22,0
	44,9	14,7	23,1	25,5
	36,1	37,8	8,7	26,4
	13,3	11,9	25,6	77,5
	15,6	54,5	14,9	41,8
	26,2	60,2	29,5	16,7
	48,1	106,3	158,4	38,3
	123,2	11,0	26,7	53,2
	29,7	49,3	18,9	9,8
	30,9	13,7	44,5	15,5
	11,2	26,9	185,2	26,5
	20,3	53,7	20,8	16,2
	110,4	56,1	49,7	25,6
	8,6	19,6	21,2	22,5
	28,4	31,7	27,4	43,1
	38,8	18,4	40,0	22,5
	57,7	25,4	22,5	12,0
	13,7	22,2	32,1	37,1
	19,2	42,4	39,7	77,8
Av	36,5	47,3	36,0	45,8
Std Dev	30,4	61,8	37,7	44,5

Table 9.4: Table with reordered task completion times for task three. Times are given in seconds to one decimal place.

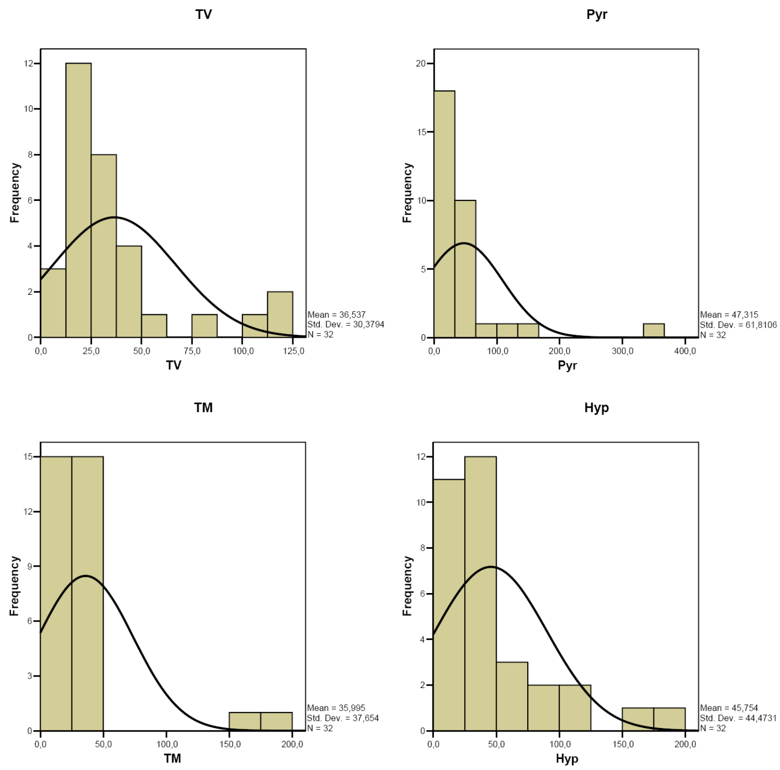


Figure 9.5: The task completion times histograms for task 3.

files could only be seen after zooming. A larger zoom factor necessitates much panning. However, the differences in completion times were not significant.

9.2.4 Task 4 (Search Task): Find File

All distributions had a left skew and were not normal (see Figure 9.7). This was verified by the Kolmogorov-Smirnov test. The ranking based on the mean task completion times(see Table 9.5) was as follows:

TreeMap(46.7s) ≈ TreeView(47.2s) ≈ Hyperbolic(47.9s) ≈ Pyramids(51.5s)

The Friedman test did not yield overall significant differences. The pairwise Wilcoxon test found no significant differences between the pairs of visualisations.

Similar to the task of navigating to a directory, the insight into deeper levels of the hierarchy in TreeMap made this task easier. As for Pyramids, the need to zoom and pan was the reason for the rather poor performance. None of the visualisations were significantly faster or slower than the others for this task.

9.2.5 Task 5 (Count Task): Count Subdirectories

The assumption from the histograms (Figure 9.9) of non-normal distribution was proved by the Kolmogorov-Smirnov test. The mean task completion times (see Table 9.6) were as follows:

Pyramids(23.4s) ≈ TreeView(26.5s) ≈ TreeMap(27.1s) ≈ Hyperbolic(32.0s)

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-1,122[a]	-,318[b]	-,598[a]	-,823[b]	-,299[a]	-1,103[a]
Asymp. Sig. (2-tailed)	,262	,751	,550	,411	,765	,270

a. Based on negative ranks.

b. Based on positive ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.6: The results of the Wilcoxon test for task completion times for task 3.

	TreeView	Pyramids	TreeMap	Hyperbolic
	103,7	61,0	21,6	42,1
	40,6	28,1	164,1	93,8
	75,8	42,8	33,4	181,7
	28,4	79,1	21,3	107,5
	20,5	154,3	15,0	12,2
	36,0	67,1	10,7	28,0
	28,8	19,5	21,9	39,0
	51,9	37,3	26,6	14,9
	104,8	29,5	66,6	47,6
	10,5	90,7	29,2	48,6
	39,4	11,5	35,7	36,5
	28,3	24,3	56,1	35,5
	21,0	12,2	28,2	86,3
	24,5	110,1	30,1	18,3
	25,6	18,4	44,3	73,2
	37,4	35,5	17,7	42,4
	25,3	15,1	31,2	49,4
	17,9	184,7	16,9	38,5
	41,2	56,0	27,6	19,8
	51,3	82,5	73,2	16,3
	243,3	27,5	49,8	28,6
	51,8	16,8	27,3	35,1
	46,1	10,3	83,2	16,4
	13,8	31,0	96,9	17,8
	40,3	88,0	31,1	18,6
	79,9	102,3	35,4	92,3
	19,2	20,6	16,5	47,8
	22,5	29,2	64,0	49,0
	39,7	33,9	28,7	33,4
	52,0	66,5	58,1	36,6
	65,6	24,8	87,2	66,1
	24,5	38,7	143,6	58,5
Av	47,2	51,5	46,7	47,9
Std Dev	42,8	42,1	36,1	34,9

Table 9.5: Table with reordered task completion times for task four. Times are given in seconds to one decimal place.

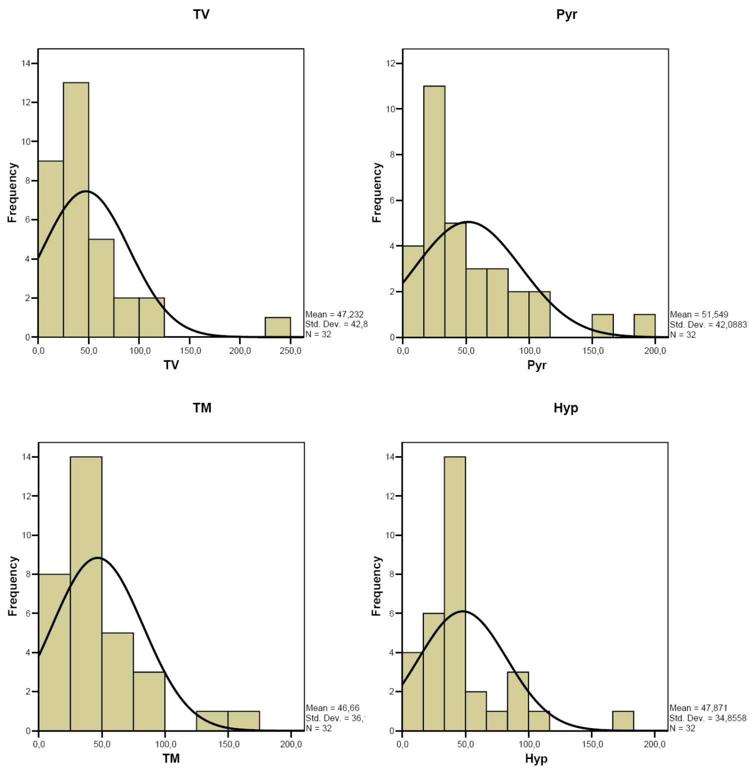


Figure 9.7: The task completion times histograms for task 4.

Test Statistics[d]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-,299[a]	-,224[b]	-,486[a]	-,224[b]	,000[c]	-,093[a]
Asymp. Sig. (2-tailed)	,765	,822	,627	,822	1,000	,926

a. Based on negative ranks.

b. Based on positive ranks.

c. The sum of negative ranks equals the sum of positive ranks.

d. Wilcoxon Signed Ranks Test

Figure 9.8: The results of the Wilcoxon test for task completion times for task 4.

	TreeView	Pyramids	TreeMap	Hyperbolic
	25,8	45,1	22,3	26,8
	20,7	13,8	58,4	84,3
	17,5	19,4	26,4	20,3
	58,4	55,7	10,1	89,7
	14,6	39,2	32,0	25,4
	31,7	26,5	5,4	7,4
	22,0	16,8	21,2	13,7
	81,4	13,0	11,9	45,1
	26,9	31,7	21,3	32,6
	15,7	14,0	20,3	32,6
	23,2	10,7	43,2	19,8
	17,5	10,9	42,1	16,7
	13,5	9,6	26,4	70,1
	17,0	19,2	18,1	15,4
	17,7	18,2	21,5	15,8
	37,4	19,5	10,0	43,1
	13,0	10,7	21,7	30,5
	20,7	19,3	13,0	18,0
	19,7	38,4	18,2	12,5
	23,4	37,5	72,7	12,1
	54,6	30,2	10,1	19,4
	13,6	18,3	14,0	9,1
	35,9	12,8	28,9	14,6
	10,4	19,4	35,5	36,2
	26,0	31,3	76,8	10,6
	46,3	24,8	59,8	32,3
	15,6	21,4	10,6	17,2
	25,5	20,3	23,8	42,4
	13,1	22,9	37,2	12,0
	21,3	28,9	17,6	24,4
	45,6	19,2	18,9	81,2
	22,2	31,4	16,4	92,8
Av	26,5	23,4	27,1	32,0
Std Dev	15,7	11,0	18,1	24,9

Table 9.6: Table with reordered task completion times for task five. Times are given in seconds to one decimal place.

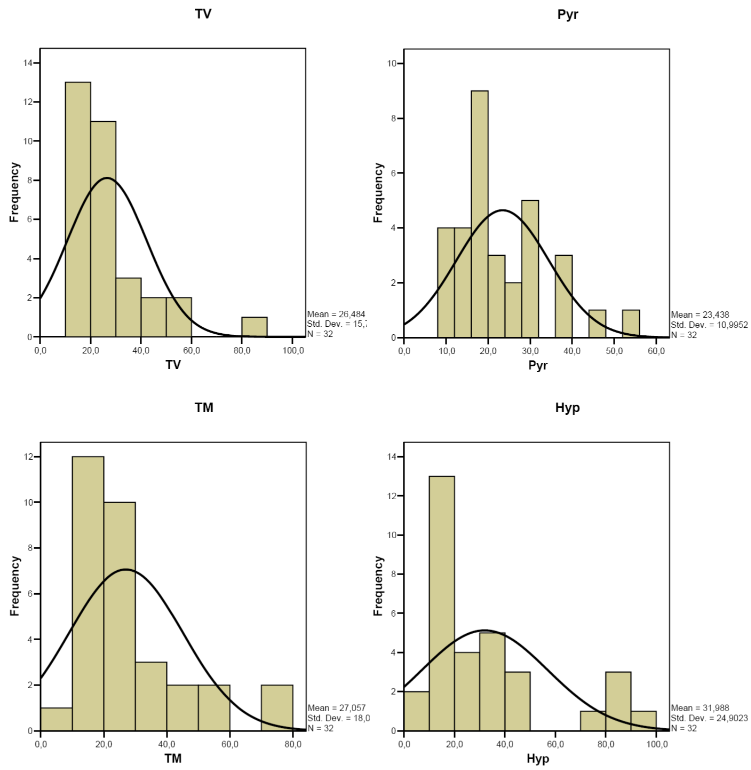


Figure 9.9: The task completion times histograms for task 5.

No overall significant differences could be found with the Friedman test. The Wilcoxon paired test did not yield significant differences between the visualisations.

For the task of counting subdirectories, Pyramids' ability to show the deeper levels of the hierarchy helped the users perform faster. In this case, there was no need for reading the names, and thus zooming and panning was not extensively used. Besides, the colour-coding made the distinction between files and directories easier. Counting directory icons in Hyperbolic seemed to be harder for the users. Again, the task completion times did not differ significantly.

9.2.6 Task 6 (Count Task): Count Files

The histograms (Figure 9.11) showed a slight left skew. The Kolmogorov-Smirnov test indicated non-normal distributions. For this task, the mean completion times (see Table 9.7) were:

TreeMap(20.0s) $\approx$ TreeView(26.6s) $\approx$ Pyramids(26.9s) $\approx$ Hyperbolic(33.3s)

TreeMap(20.0s) $<$ Hyperbolic(33.3s)

The Friedman test did not prove overall significant differences, but very close (Sig.=.051). The Wilcoxon pairwise test (see Figure 9.12) performed afterwards showed that the TreeMap visualisation ($M = 20.034$) is significantly faster than the Hyperbolic visualisation ($M = 33.251$). The differences between the other combinations of visualisations are not significant.

The good performance of TreeMap for the task of counting files was probably due to

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-,449[a]	-,299[b]	-,654[b]	-,617[b]	-1,066[b]	-,841[b]
Asymp. Sig. (2-tailed)	,654	,765	,513	,537	,286	,400

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.10: The results of the Wilcoxon test for task completion times for task 5.

	TreeView	Pyramids	TreeMap	Hyperbolic
	25,1	22,9	13,6	15,8
	20,2	15,5	19,4	58,9
	29,5	22,7	15,1	48,6
	63,6	91,5	11,4	30,0
	20,3	25,0	16,9	12,1
	29,0	16,7	3,6	11,9
	32,3	11,7	12,6	21,6
	60,9	8,7	12,8	36,3
	17,6	42,9	19,5	40,8
	20,0	18,4	18,5	53,8
	30,5	8,3	40,8	59,3
	31,0	6,8	40,0	12,5
	10,6	7,1	11,1	64,8
	7,9	28,6	8,9	9,4
	23,8	14,6	14,0	19,7
	27,0	9,7	10,4	49,6
	11,0	12,6	8,5	43,5
	59,2	139,7	7,9	17,0
	10,7	19,5	11,0	10,8
	15,8	35,5	44,8	45,0
	90,5	27,4	19,2	30,3
	12,1	15,6	12,1	45,5
	21,6	14,6	26,4	14,1
	10,9	31,6	58,3	36,0
	19,0	27,5	46,3	9,2
	44,7	22,3	39,3	55,3
	7,3	53,9	9,0	12,2
	33,4	18,5	27,1	31,7
	18,7	13,5	19,3	14,5
	14,8	14,9	10,6	88,1
	19,5	34,9	12,3	37,9
	13,2	28,8	20,5	27,8
Av	26,6	26,9	20,0	33,3
Std Dev	18,8	26,3	13,5	19,9

Table 9.7: Table with reordered task completion times for task six. Times are given in seconds to one decimal place.

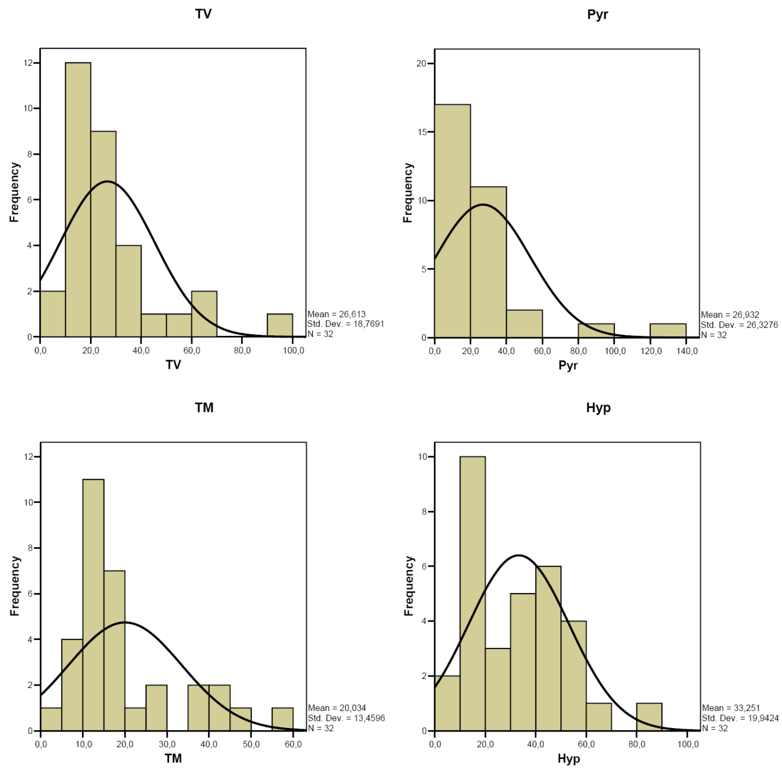


Figure 9.11: The task completion times histograms for task 6.

the nature of this method. Most of the screen space is used for representing files, only thin frames are used for directories. Though colour-coded, Pyramids performed rather poorly. Seemingly, counting file icons in Hyperbolic was harder for the users. TreeMap was significantly faster than Hyperbolic for the task of counting files.

9.2.7 Task 7 (Compare Task): Number of Subdirectories

All four distribution appear fairly normal from the histograms (Figure 9.13). The Kolmogorov-Smirnov test verified this (Figure 9.14). Due to the normal distribution, an ANOVA was performed. As Mauchly's test indicated that the assumption of sphericity was met ($\chi^2(5) = 1.058, Sig. = .958$), no further corrections were necessary. Ordering by mean task completion times (see Table 9.8) was as follows:

TreeView(44.4s) $\approx$ Hyperbolic(44.5s) $\approx$ TreeMap(45.6s) $\approx$ Pyramids(46.0s)

ANOVA showed no overall significant difference in task completion times. The paired samples T-test also indicated no significant differences between the visualisations, see Figure 9.15.

Users seem to be used to comparing directories in TreeView. The possibility of having two directories open at one time without having to zoom made TreeView faster. Only scrolling up and down and counting was needed to compare the directories, once open. The longer navigation and zooming in Pyramids were probably responsible for its long completion times. Nevertheless, these differences were not significant.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-,131[a]	-1,552[a]	-1,309[b]	-,729[a]	-1,627[b]	-3,029[b]
Asymp. Sig. (2-tailed)	,896	,121	,191	,466	,104	,002

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.12: The results of the Wilcoxon test for task completion times for task 6.

	TreeView	Pyramids	TreeMap	Hyperbolic
	18,5	67,4	26,6	66,7
	35,2	20,8	82,5	47,2
	41,8	56,5	79,2	42,3
	68,5	80,1	17,8	74,8
	75,7	56,5	55,2	16,3
	69,0	49,7	16,8	34,1
	66,9	23,7	34,0	28,2
	50,8	40,3	20,8	52,5
	32,5	58,6	72,7	54,0
	22,5	33,4	40,1	55,7
	60,5	15,9	42,8	30,5
	51,0	30,2	58,3	26,4
	21,6	19,2	47,7	55,5
	35,8	37,5	34,1	35,8
	32,8	29,3	45,7	26,6
	45,4	32,0	15,3	77,2
	22,8	24,6	34,7	73,1
	62,4	82,2	20,5	47,6
	55,5	54,4	32,6	17,1
	44,5	144,5	120,3	43,1
	46,2	64,2	44,2	37,7
	62,5	56,7	41,2	14,2
	71,5	14,4	82,9	12,4
	26,6	38,7	28,5	47,8
	40,0	54,5	85,3	27,6
	62,6	26,4	35,8	69,0
	19,1	44,7	19,7	22,3
	52,4	40,1	28,9	67,1
	29,9	59,9	50,9	54,9
	39,0	22,1	30,8	34,1
	19,1	52,9	35,0	30,8
	37,1	41,5	78,2	102,8
Av	44,4	46,0	45,6	44,5
Std Dev	17,4	25,5	25,1	21,5

Table 9.8: Table with reordered task completion times for task seven. Times are given in seconds to one decimal place.

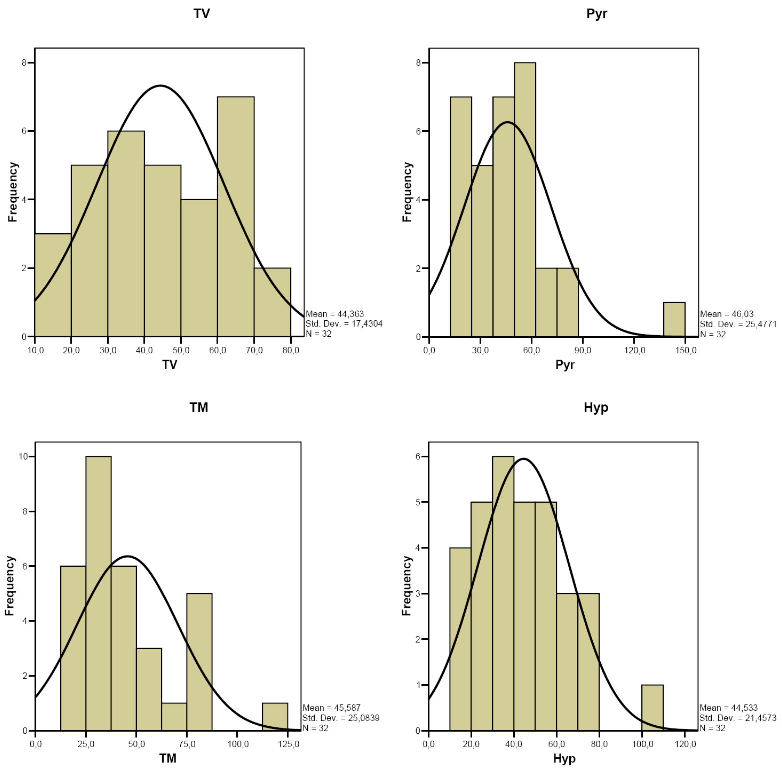


Figure 9.13: The task completion times histograms for task 7.

Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
TV	,104	32	,200*	,949	32	,135
Pyr	,137	32	,134	,849	32	,000
TM	,155	32	,050	,890	32	,004
Hyp	,096	32	,200*	,959	32	,254

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Figure 9.14: The results of the Kolmogorov-Smirnov test for task completion times for task 7.

Paired Samples Test

		df	Sig. (2-tailed)
Pair 1	TV - Pyr	31	,755
Pair 2	TV - TM	31	,825
Pair 3	TV - Hyp	31	,976
Pair 4	Pyr - TM	31	,935
Pair 5	Pyr - Hyp	31	,797
Pair 6	TM - Hyp	31	,865

Figure 9.15: The results of the paired samples T-test for task completion times for task 7.

9.2.8 Task 8 (Compare Task): Number of Files

The histograms (Figure 9.16) showed a light left skew of the distributions. The Kolmogorov-Smirnov (figure 9.17) test indicated that the distribution for Pyramids is normal ($Pyr : D(32) = .125, Sig = .200$). As the other distributions were not normal, a Friedman test was performed, finding no overall significant difference ($Sig. = .640$). A Wilcoxon test did not find significant differences between any pair of visualisations. For this task, the mean completion times (see Table 9.9) were:

TreeMap(46.5s) ≈ Pyramids(46.7s) ≈ TreeView(50.8s) ≈ Hyperbolic(53.1s)

The mean task completion times were similar for all visualisations. TreeMap's dedication of more space to files made it perform best. Pyramid's colour-coding of files also helped the users.

9.3 Task Success (Effectiveness)

Effectiveness is indicated by the number of successfully completed tasks. Failing to finish a task successfully was usually due to users' mistakes. The analysis of successful task completion was performed by using the adjusted Wald confidence intervals from [Sauro, 2006]. The adjusted Wald method yields better results for rather small samples ([Sauro and Lewis, 2005]). For the cases with 100% task completion, the Best Estimate is used (obtained by the Laplace method). Confidence intervals give the range within which the results for the whole population should lie. The range is obtained from the data for the sample.

As can be seen from Table 9.10, all the intervals overlap. Thus, the differences between the visualisations in terms of successful task completion are not significant. Clearly, all visualisations communicate the hierarchical data similarly well in terms of effectiveness for the tasks being used.

9.4 Ratings

Ratings for the visualisations were collected in the feedback questionnaire. The participants rated the visualisations on different aspects (such as overview or navigation). The ratings on the seven-point Likert scale were transformed for analysis. The worst rating

	TreeView	Pyramids	TreeMap	Hyperbolic
	29,2	67,3	30,2	50,7
	28,7	17,5	22,9	90,3
	56,9	31,7	59,1	58,7
	64,9	121,7	26,6	104,2
	51,7	55,0	38,3	12,7
	33,8	58,5	14,8	25,8
	68,5	27,8	29,7	18,2
	162,6	45,9	19,3	43,9
	40,3	51,4	32,2	248,0
	11,7	28,6	61,7	80,2
	121,6	17,2	52,5	26,3
	34,7	70,4	85,5	22,3
	21,1	25,3	18,8	156,5
	18,2	35,6	29,9	24,9
	45,4	29,2	35,2	24,1
	29,7	50,0	22,8	45,5
	27,0	33,2	48,2	73,9
	89,7	82,6	43,7	25,0
	49,7	33,2	24,3	28,3
	49,2	52,8	67,1	47,4
	63,2	24,3	34,0	59,2
	16,0	40,5	48,3	27,7
	77,2	17,9	73,3	18,1
	42,4	59,5	68,1	46,2
	42,0	58,7	93,9	24,5
	61,5	29,5	114,8	62,0
	39,6	85,9	31,6	31,5
	36,3	61,7	39,6	44,5
	14,0	44,9	47,2	39,1
	33,7	21,3	40,3	44,4
	133,2	47,2	46,3	40,6
	31,2	66,6	87,8	53,7
Av	50,8	46,7	46,5	53,1
Std Dev	34,6	23,3	24,4	46,2

Table 9.9: Table with reordered task completion times for task eight. Times are given in seconds to one decimal place.

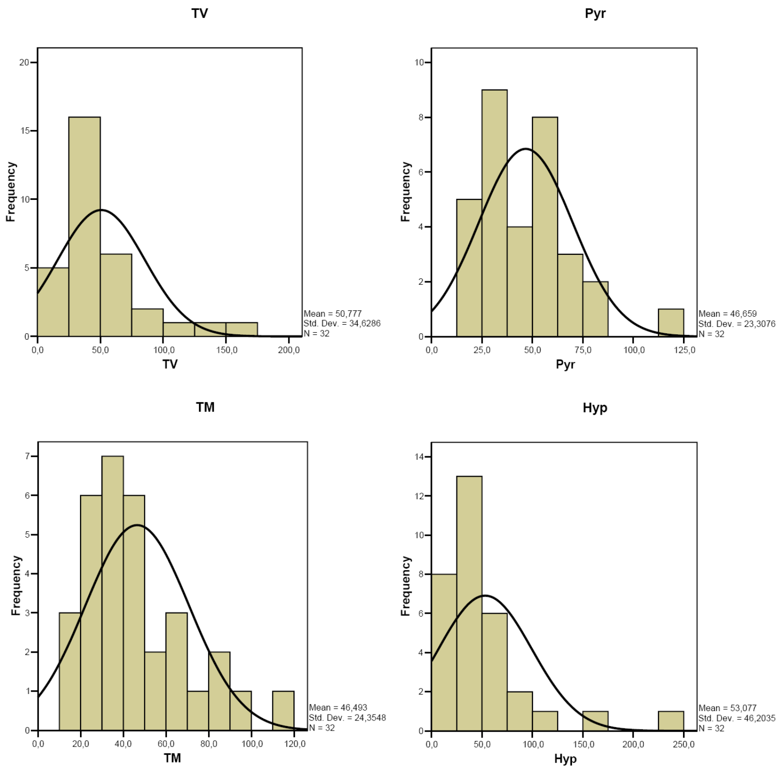


Figure 9.16: The task completion times histograms for task 8.

Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
TV	,177	32	,012	,826	32	,000
Pyr	,125	32	,200*	,916	32	,016
TM	,157	32	,043	,910	32	,011
Hyp	,236	32	,000	,680	32	,000

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Figure 9.17: The results of the Kolmogorov-Smirnov test for task completion times for task 8.

Task	Visualisation	Completed Tasks	Confidence Intervals
1: Deepest Subdirectory	TreeView	28	(0.7132 , 0.9564)
	Pyramids	28	(0.7132 , 0.9564)
	TreeMap	28	(0.7132 , 0.9564)
	Hyperbolic	26	(0.6432 , 0.9148)
2: Branching Factor	TreeView	32	(0.9070 , 1.0000)
	Pyramids	30	(0.7884 , 0.9928)
	TreeMap	28	(0.7132 , 0.9564)
	Hyperbolic	31	(0.8289 , >.9999)
3: Navigate to Directory	TreeView	32	(0.9706)
	Pyramids	32	(0.9706)
	TreeMap	32	(0.9706)
	Hyperbolic	31	(0.8289 , >.9999)
4: Navigate to File	TreeView	32	(0.9706)
	Pyramids	31	(0.8289 , >.9999)
	TreeMap	32	(0.9706)
	Hyperbolic	32	(0.9706)
5: Count Subdirectories	TreeView	30	(0.7884 , 0.9928)
	Pyramids	31	(0.8289 , >.9999)
	TreeMap	29	(0.7500 , 0.9754)
	Hyperbolic	30	(0.7884 , 0.9928)
6: Count Files	TreeView	30	(0.7884 , 0.9928)
	Pyramids	30	(0.7884 , 0.9928)
	TreeMap	24	(0.5767 , 0.8697)
	Hyperbolic	28	(0.7132 , 0.9564)
7: Compare Subdirectories	TreeView	32	(0.9706)
	Pyramids	32	(0.9706)
	TreeMap	31	(0.8289 , >.9999)
	Hyperbolic	31	(0.8289 , >.9999)
8: Compare Files	TreeView	29	(0.7500 , 0.9754)
	Pyramids	29	(0.7500 , 0.9754)
	TreeMap	30	(0.7884 , 0.9928)
	Hyperbolic	30	(0.7884 , 0.9928)

Table 9.10: Successful task completion table showing the number of completed tasks and the corresponding confidence intervals.

received zero points, the best rating six points. The analysis steps used were the same as for analysing task completion times (see Section 9.2).

For better readability, following notations will be used: $\approx$ means no statistically significant differences, $<$ means statistically significant differences (at the level $p < 0.05$) between the browsers. The browser on the left of $<$ is thus rated significantly worse than the browser on the right. Additionally, average ratings are given in brackets. The notation Hyperbolic(3.38) $\approx$ Pyramids(4.13) means that Hyperbolic browser rated worse than Pyramids, but this difference is not significant. Similarly, the notation TreeMap(1.25) $<$ Hyperbolic(3.38) means that TreeMap was rated significantly worse than Hyperbolic browser. All pairs of browsers with significantly different ratings are given in next sections for a better overview.

Although there were almost no significant differences in terms of task completion times, the subjective ratings showed clear preferences of the users. The perceived performance differed greatly from the measured performance. This suggests that humans are creatures of habit and prefer things they know.

9.4.1 Question 1: Overview

The ratings for question 1: overview are shown in Table 9.11. The histograms (Figure 9.18) showed a not normal distribution, except probably Hyperbolic. The results from the Kolmogorov-Smirnov test indicate that none of the distributions is normal. TreeView was rated better than Pyramids and Hyperbolic. TreeMap was rated very badly. The Friedman test yielded an overall significant difference, and therefore a Wilcoxon test (Figure 9.19) was used to find the differences. TreeMap was rated significantly worse in terms of overview than all other visualisations. The difference between TreeView ($M = 4.78$) and Hyperbolic ($M = 3.38, T = 22, z = -3.531, p < .05$) is significant as well. Although TreeMap provides the whole structure of the hierarchy at glance, users did not feel comfortable, but rather overwhelmed. Statistically significant differences in ratings were found for the following pairs of browsers:

TreeMap(1.25) $<$ Hyperbolic(3.38)

TreeMap(1.25) $<$ Pyramids(4.13)

TreeMap(1.25) $<$ TreeView(4.78)

Hyperbolic(3.38) $<$ TreeView(4.48)

9.4.2 Question 2: Operability

The ratings for question 2: operability are shown in Table 9.12. The distributions appear non-normal from the histograms (Figure 9.20). The Kolmogorov-Smirnov test confirmed this assumption. The TreeView was rated best, followed by Pyramids, TreeMap and Hyperbolic. A Friedman test confirmed that there are overall significant differences. A pairwise Wilcoxon test (Figure 9.21) was performed to find these differences. TreeView was rated significantly better than all other visualisations in terms of operability. Additionally, the difference between Pyramids ($M = 4.16$) and Hyperbolic ($M = 2.97, T = 20, z = -2.228, p < .05$) is significant. In terms of operability, TreeView was rated significantly better due to the everyday practice with this kind of visualisation. Users felt more comfortable operating TreeView than the other visualisations. The perceived operability of Pyramids is significantly better than of Hyperbolic. This could be due to distortion in Hyperbolic. Statistically significant differences in ratings were found for the following pairs of browsers:

Hyperbolic(2.97) $<$ Pyramids(4.16)

User No	TreeView	Pyramids	TreeMap	Hyperbolic
1	4	1	6	1
2	5	0	1	5
3	1	4	3	1
4	6	0	4	2
5	6	0	5	5
6	4	0	1	4
7	5	1	5	3
8	2	5	5	2
9	3	1	4	2
10	6	1	4	4
11	6	0	6	3
12	4	1	2	2
13	6	0	4	3
14	6	0	4	6
15	0	2	6	6
16	5	2	1	4
17	6	2	5	4
18	6	0	6	4
19	5	4	3	5
20	6	1	6	1
21	5	2	4	2
22	4	5	5	3
23	6	0	5	4
24	3	0	3	3
25	3	1	5	3
26	6	0	6	3
27	6	0	0	4
28	5	1	5	3
29	6	0	6	6
30	5	1	4	3
31	6	2	3	2
32	6	3	5	5
Av	4,78	1,25	4,13	3,38
Std Dev	1,60	1,50	1,68	1,43

Table 9.11: User ratings for question one: overview. Zero points mean very bad, six points mean very good rating for the given aspect.

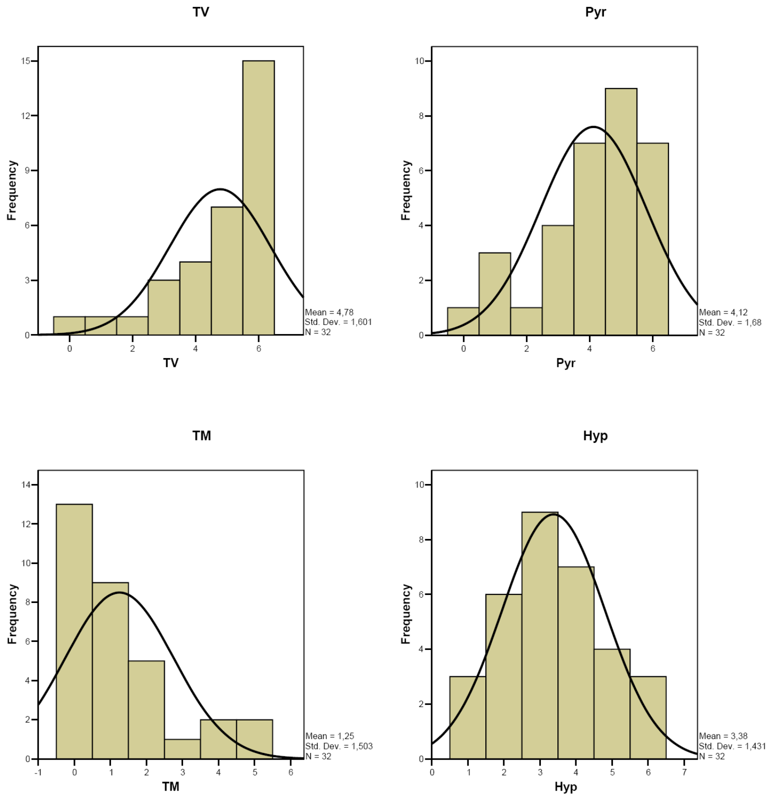


Figure 9.18: The histograms for ratings for question 1.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-1,719[a]	-4,507[a]	-3,531[a]	-4,464[a]	-1,566[a]	-3,736[b]
Asymp. Sig. (2-tailed)	,086	,000	,000	,000	,117	,000

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.19: The results of the Wilcoxon test for ratings for question 1.

User No	TreeView	Pyramids	TreeMap	Hyperbolic
1	5	5	6	6
2	6	0	1	5
3	1	5	5	1
4	6	0	5	2
5	6	2	4	2
6	6	5	2	4
7	5	2	4	3
8	5	4	4	5
9	5	3	4	1
10	6	4	6	0
11	6	4	6	4
12	4	4	4	1
13	6	3	5	3
14	5	5	2	4
15	4	4	4	4
16	3	1	3	4
17	5	4	5	0
18	6	0	1	2
19	5	6	3	6
20	6	3	6	1
21	4	4	5	3
22	5	5	4	2
23	6	5	5	4
24	5	1	5	1
25	5	3	5	2
26	3	6	6	5
27	4	1	1	4
28	5	5	6	1
29	6	6	2	6
30	5	5	5	1
31	6	3	5	4
32	6	2	4	4
Av	5,03	3,44	4,16	2,97
Std Dev	1,15	1,81	1,55	1,79

Table 9.12: User ratings for question two: operability. Zero points mean very bad, six points mean very good rating for the given aspect.

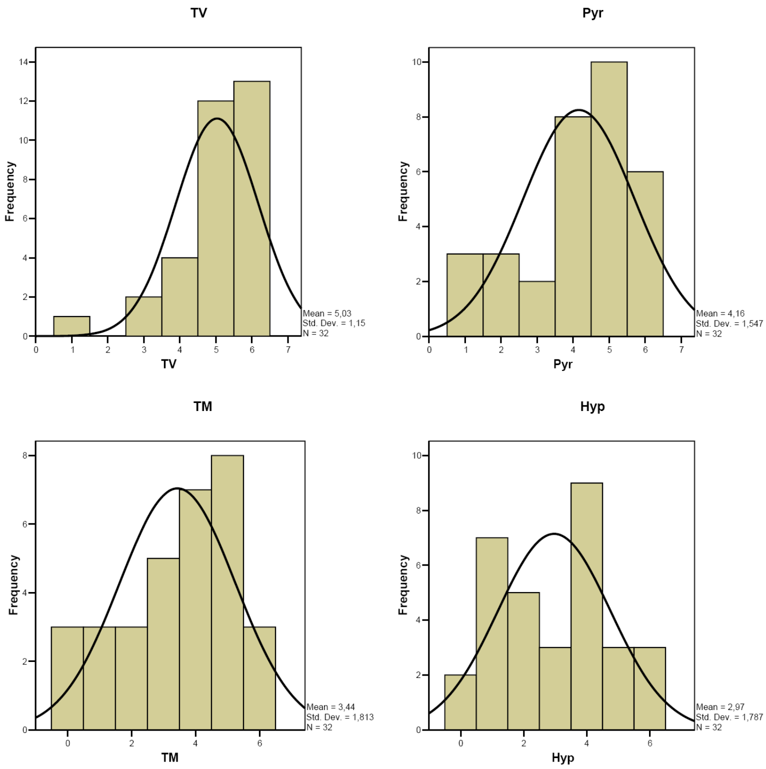


Figure 9.20: The histograms for ratings for question 2.

Hyperbolic(2.97) $<$ TreeView(5.03)

TreeMap(3.44) $<$ TreeView(5.03)

Pyramids(4.16) $<$ TreeView(5.03)

9.4.3 Question 3: Intuitive

The ratings for question 3: intuitive are shown in Table 9.13. The TreeMap distribution looked fairly normal in the histogram (Figure 9.22). The Kolmogorov-Smirnov test identified that the TreeMap distribution was normal ($TM : D(32) = .142, Sig = .098$), all the others were not. The TreeView visualisation was best rated, followed by Pyramids, Hyperbolic and TreeMap being rated worse. The Friedman test showed overall significant differences. The Wilcoxon test proved that TreeMap was rated significantly worse than all other visualisations (see Figure 9.23). An alphabetical ordering appeared to be much more intuitive for the users. Besides, the nested representation in TreeMap was not a common representation for the users. Statistically significant differences in ratings were found for the following pairs of browsers:

TreeMap(2.97) $<$ Hyperbolic(4.03)

TreeMap(2.97) $<$ Pyramids(4.28)

TreeMap(2.97) $<$ TreeView(4.38)

Test Statistics[b]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-2,424[a]	-3,149[a]	-4,071[a]	-1,863[a]	-2,228[a]	-1,128[a]
Asymp. Sig. (2-tailed)	,015	,002	,000	,062	,026	,259

a. Based on positive ranks.

b. Wilcoxon Signed Ranks Test

Figure 9.21: The results of the Wilcoxon test for ratings for question 2.

User No	TreeView	Pyramids	TreeMap	Hyperbolic
1	2	5	6	6
2	2	4	5	5
3	5	4	4	2
4	4	0	6	5
5	5	3	4	3
6	0	1	5	5
7	6	4	5	4
8	5	1	4	5
9	4	3	6	3
10	6	2	6	5
11	6	3	4	5
12	5	2	4	3
13	6	3	5	3
14	6	5	5	5
15	3	2	0	1
16	5	3	2	3
17	5	2	4	4
18	3	6	4	4
19	5	5	4	6
20	6	3	6	1
21	6	1	3	5
22	3	4	3	4
23	6	6	6	5
24	6	4	6	2
25	1	3	5	5
26	0	2	3	6
27	4	1	1	4
28	2	4	5	1
29	6	4	2	5
30	6	2	4	4
31	5	2	5	4
32	6	1	5	6
Av	4,38	2,97	4,28	4,03
Std Dev	1,86	1,53	1,51	1,47

Table 9.13: User ratings for question three: intuitive. Zero points mean very bad, six points mean very good rating for the given aspect.

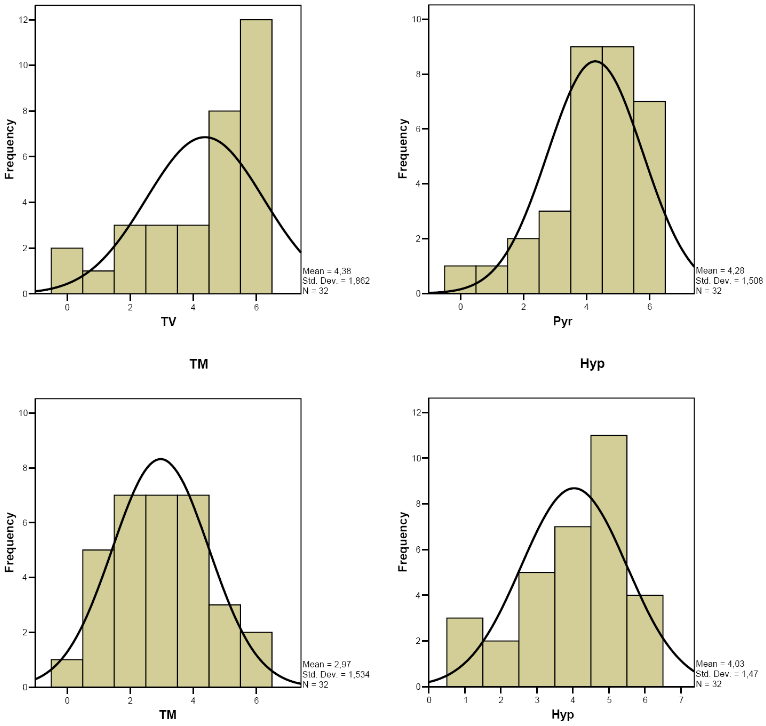


Figure 9.22: The histograms for ratings for question 3.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-,296[a]	-2,933[a]	-1,295[a]	-3,224[a]	-,539[a]	-2,402[b]
Asymp. Sig. (2-tailed)	,768	,003	,195	,001	,590	,016

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.23: The results of the Wilcoxon test for ratings for question 3.

9.4.4 Question 4: Usable

The ratings for question 4: usable are shown in Table 9.14. The histograms showed a rather normal distribution for Pyramids. The Kolmogorov-Smirnov test confirmed this assumption ($Pyr : D(32) = .145, Sig = .087$). There were overall significant differences (Friedman test). The Wilcoxon test (see Figure 9.25) confirms significant differences between all pairs of visualisations. In terms of usability, TreeView was rated significantly better than Pyramids, which is again better than Hyperbolic, being better than TreeMap. TreeMap was not perceived usable. As for perceived usability, prior experience with TreeView visualisations influenced the users greatly. Probably due to the non-alphabetical ordering, TreeMap was rated less usable. Statistically significant differences in ratings were found for all pairs of browsers:

TreeMap(1.72) < Hyperbolic(2.81)

TreeMap(1.72) < Pyramids(3.84)

TreeMap(1.72) < TreeView(5.09)

Hyperbolic(2.81) < Pyramids(3.84)

Hyperbolic(2.81) < TreeView(5.09)

Pyramids(3.84) < TreeView(5.09)

9.4.5 Question 5: Understandable

The ratings for question 5: understandable are shown in Table 9.15. As the histograms (Figure 9.26) showed, none of the distributions appear normal. This suspicion was confirmed by the Kolmogorov-Smirnov test. Therefore, a Friedman test was performed, finding that there were overall significant differences. The Wilcoxon test (Figure 9.27) showed that TreeView was rated significantly better as being understandable than all other visualisations. Furthermore, TreeMap ($M = 4.19$) was rated significantly worse than Pyramids ($M = 4.97, T = 15, z = -2.078, p < .05$). Again, the habit of working with TreeView resulted in it being rated significantly more understandable than the others. Probably due to the less-known nested representation, TreeMap was rated less understandable. Statistically significant differences in ratings were found for the following pairs of browsers:

TreeMap(4.19) < Pyramids(4.97)

TreeMap(4.19) < TreeView(5.75)

Hyperbolic(4.84) < TreeView(5.75)

Pyramids(4.97) < TreeView(5.75)

9.4.6 Question 6: Logical

The ratings for question 6: logical are shown in Table 9.16. The assumption made from the histograms (Figure 9.28) that none of the distributions were normal was confirmed by the Kolmogorov-Smirnov test. TreeMap was rated very poorly, TreeView best, Pyramids and Hyperbolic similar. The Friedman test resulted in overall significant differences. The pairwise Wilcoxon test (Figure 9.29) showed that TreeView was rated significantly better as being more logical than all other visualisations. On the contrary, TreeMap was rated significantly worse than all others for being logical. The difference between Pyramids and Hyperbolic was not significant. The representation by alphabetical listing and indentation in TreeView seemed more logical to the users. On the contrary, the nesting and

User No	**TreeView**	**Pyramids**	**TreeMap**	**Hyperbolic**
User No	TreeView	TreeMap	Pyramids	Hyperbolic
1	5	2	5	4
2	5	1	1	4
3	3	3	4	1
4	5	0	5	2
5	5	0	3	2
6	5	1	3	3
7	5	2	5	4
8	5	2	3	4
9	5	3	5	1
10	6	1	4	1
11	6	0	4	0
12	4	5	5	2
13	6	0	5	4
14	6	0	3	5
15	4	5	6	4
16	5	1	2	4
17	5	4	4	0
18	6	1	2	0
19	6	5	2	6
20	6	1	6	2
21	5	0	4	3
22	4	5	4	3
23	6	0	4	4
24	4	1	6	1
25	5	1	3	3
26	3	1	6	5
27	4	0	2	2
28	5	1	3	1
29	6	4	3	6
30	6	1	4	2
31	6	2	3	2
32	6	2	4	5
Av	5,09	1,72	3,84	2,81
Std Dev	0,89	1,67	1,32	1,71

Table 9.14: User ratings for question four: usable. Zero points mean very bad, six points mean very good rating for the given aspect.

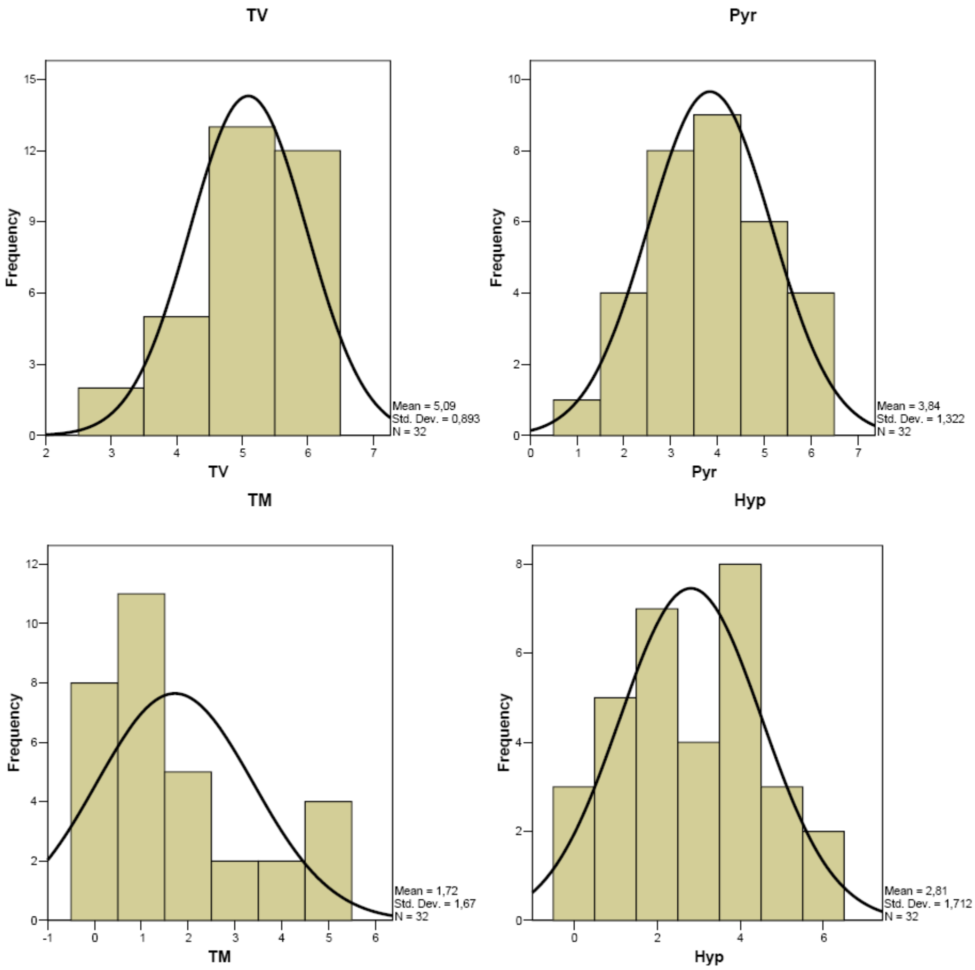


Figure 9.24: The histograms for ratings for question 4.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-3,215[a]	-4,704[a]	-4,452[a]	-4,143[a]	-2,319[a]	-2,529[b]
Asymp. Sig. (2-tailed)	,001	,000	,000	,000	,020	,011

a. Based on positive ranks.
b. Based on negative ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.25: The results of the Wilcoxon test for ratings for question 4.

User No	**TreeView**	**Pyramids**	**TreeMap**	**Hyperbolic**
User No	TreeView	TreeMap	Pyramids	Hyperbolic
1	6	6	6	6
2	6	0	4	4
3	5	5	5	4
4	6	5	6	5
5	6	1	5	5
6	6	0	6	6
7	6	5	5	5
8	5	2	4	5
9	5	3	5	3
10	6	6	6	6
11	6	6	6	6
12	4	4	5	4
13	6	3	5	4
14	6	5	4	5
15	6	5	6	6
16	6	5	3	5
17	6	5	6	5
18	6	6	5	5
19	6	5	4	6
20	6	6	6	3
21	5	4	4	4
22	5	6	2	2
23	6	6	6	6
24	6	5	5	5
25	6	6	6	5
26	5	6	4	6
27	6	0	5	3
28	6	3	5	5
29	6	3	5	5
30	6	5	6	5
31	6	4	4	5
32	6	3	5	6
Av	5,75	4,19	4,97	4,84
Std Dev	0,51	1,89	1,00	1,05

Table 9.15: User ratings for question five: understandable. Zero points mean very bad, six points mean very good rating for the given aspect.

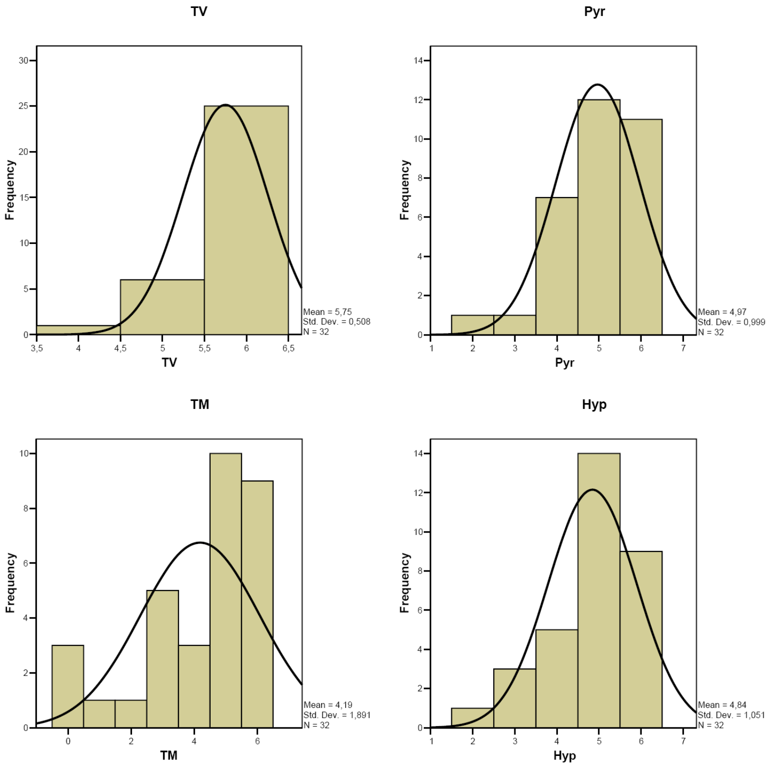


Figure 9.26: The histograms for ratings for question 5.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-3,684[a]	-3,872[a]	-4,017[a]	-2,078[a]	-,515[a]	-1,817[b]
Asymp. Sig. (2-tailed)	,000	,000	,000	,038	,607	,069

a. Based on positive ranks.
b. Based on negative ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.27: The results of the Wilcoxon test for ratings for question 5.

User No	**TreeView**	**Pyramids**	**TreeMap**	**Hyperbolic**
User No	TreeView	TreeMap	Pyramids	Hyperbolic
1	6	1	6	6
2	0	5	5	5
3	5	5	5	4
4	6	0	6	6
5	6	2	4	4
6	6	3	5	6
7	5	5	5	4
8	6	1	5	5
9	5	5	6	3
10	6	6	6	6
11	6	6	6	6
12	4	4	4	5
13	6	0	5	5
14	6	3	5	6
15	5	5	6	6
16	6	3	2	4
17	6	5	6	4
18	6	3	5	6
19	6	6	6	6
20	6	6	6	3
21	6	2	5	5
22	5	6	4	3
23	6	6	6	6
24	6	5	5	5
25	6	6	5	5
26	5	6	5	6
27	6	0	3	5
28	6	1	6	5
29	6	6	6	6
30	5	3	6	5
31	6	4	5	3
32	6	4	5	6
Av	5,53	3,84	5,16	5,00
Std Dev	1,14	2,05	0,95	1,05

Table 9.16: User ratings for question six: logical. Zero points mean very bad, six points mean very good rating for the given aspect.

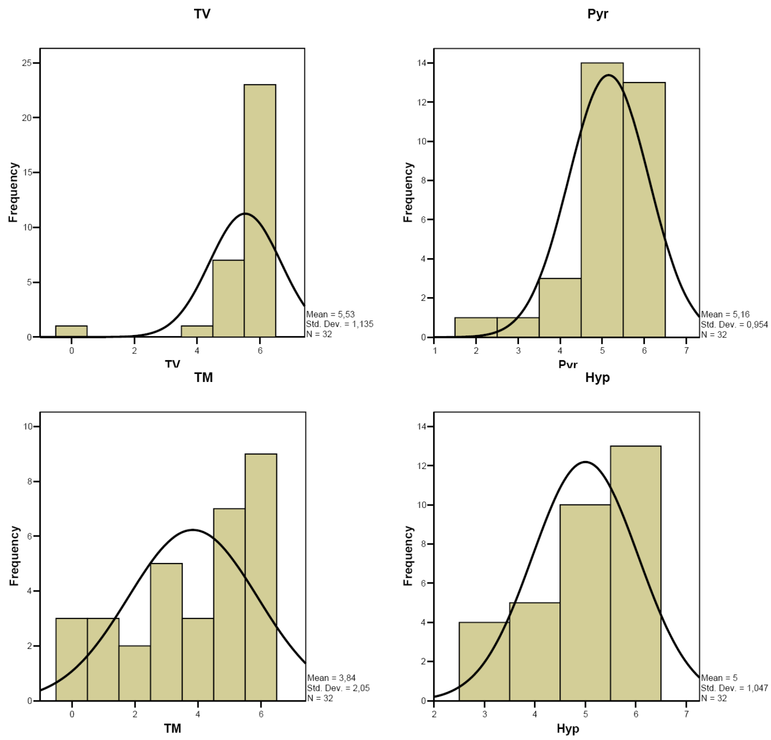


Figure 9.28: The histograms for ratings for question 6.

non-alphabetical ordering in TreeMap was perceived as illogical. Statistically significant differences in ratings were found for the following pairs of browsers:

TreeMap(3.84) < Hyperbolic(5.00)

TreeMap(3.84) < Pyramids(5.16)

TreeMap(3.84) < TreeView(5.53)

Hyperbolic(5.00) < TreeView(5.53)

Pyramids(5.16) < TreeView(5.53)

9.4.7 Question 7: Useful

The ratings for question 7: useful are shown in Table 9.17. The histograms (Figure 9.30) showed a non-normal distribution for the four visualisations. The Kolmogorov-Smirnov test yielded the same result. TreeMap was rated very badly, TreeView rather good, Pyramids and Hyperbolic similarly. Overall significant differences were found by the Friedman test. The Wilcoxon test (Figure 9.31) yielded following significant differences: TreeMap worse than all others, TreeView better than all others in terms of being useful. The difference between Hyperbolic and Pyramids is not significant. TreeView is considered most useful, TreeMap is considered least useful. The familiarity and alphabetical ordering of TreeView was perceived significantly more useful for everyday work. Statistically significant differences in ratings were found for the following pairs of browsers:

TreeMap(2.75) < Pyramids(3.88)

TreeMap(2.75) < Hyperbolic(3.91)

TreeMap(2.75) < TreeView(5.09)

User No	TreeView	Pyramids	TreeMap	Hyperbolic
1	5	3	3	4
2	0	1	2	4
3	5	3	3	3
4	5	0	4	5
5	6	3	5	3
6	5	1	3	4
7	5	2	4	3
8	6	1	3	5
9	6	3	5	1
10	6	5	6	6
11	6	6	5	5
12	4	4	2	2
13	6	0	5	3
14	6	0	3	6
15	4	4	5	6
16	5	3	1	4
17	6	3	5	1
18	6	4	4	4
19	5	6	1	6
20	6	6	6	3
21	4	3	5	4
22	3	4	5	5
23	6	0	4	2
24	5	1	5	1
25	4	1	4	5
26	5	1	1	5
27	6	2	4	3
28	4	4	4	2
29	6	5	5	6
30	5	1	5	3
31	6	5	2	5
32	6	3	5	6
Av	5,09	2,75	3,88	3,91
Std Dev	1,25	1,87	1,43	1,57

Table 9.17: User ratings for question seven: useful. Zero points mean very bad, six points mean very good rating for the given aspect.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-2,075[a]	-3,165[a]	-2,513[a]	-3,190[a]	-,661[a]	-2,474[b]
Asymp. Sig. (2-tailed)	,038	,002	,012	,001	,509	,013

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.29: The results of the Wilcoxon test for ratings for question 6.

Pyramids(3.88) < TreeView(5.09)

Hyperbolic(3.91) < TreeView(5.09)

9.4.8 Question 8: Orientation

The ratings for question 8: orientaion are given in Table 9.18. The distribution for Hyperbolic looked fairly normal in the histogram (Figure 9.32). This fact was confirmed by the Kolmogorov-Smirnov test ($Hyp : D(32) = .142, Sig = .097$). All other distributions were not normal. Therefore, a Friedman test was performed, finding overall significant differences. The Wilcoxon test found the following significant differences: TreeView was rated significantly better than all others and TreeMap was rated significantly worse than all other in terms of orientation. The difference between Hyperbolic and Pyramids was not significant. Although only the first level of the hierarchy is visible by default in TreeView and further exploration requires manually opening the directories, orientation was rated significantly higher. The ability to read the names and the alphabetical list upgraded the orientation in TreeView. Statistically significant differences in ratings were found for the following pairs of browsers:

TreeMap(1.28) < Hyperbolic(3.06)

TreeMap(1.28) < Pyramids(3.13)

TreeMap(1.28) < TreeView(5.13)

Hyperbolic(3.06) < TreeView(5.13)

Pyramids(3.13) < TreeView(5.13)

9.4.9 Question 9: Navigation

The histograms (Figure 9.34) showed non-normally distributed data. The Kolmogorov-Smirnov test proved that all of the distributions were non-normal. TreeView was rated very highly, Hyperbolic very poorly in terms of navigation, as can be seen in Table 9.19. Overall statistically significant differences were confirmed by the Friedman test. The Wilcoxon test showed that the perceived navigation in TreeView was significantly better than in all other visualisations. Familiarity and the fact that only simple navigation elements are needed were probably the reasons for the higher ratings for TreeView. Statistically significant differences in ratings were found for the following pairs of browsers:

Hyperbolic(2.75) < TreeView(5.16)

TreeMap(3.31) < TreeView(5.16)

Pyramids(3.63) < TreeView(5.16)

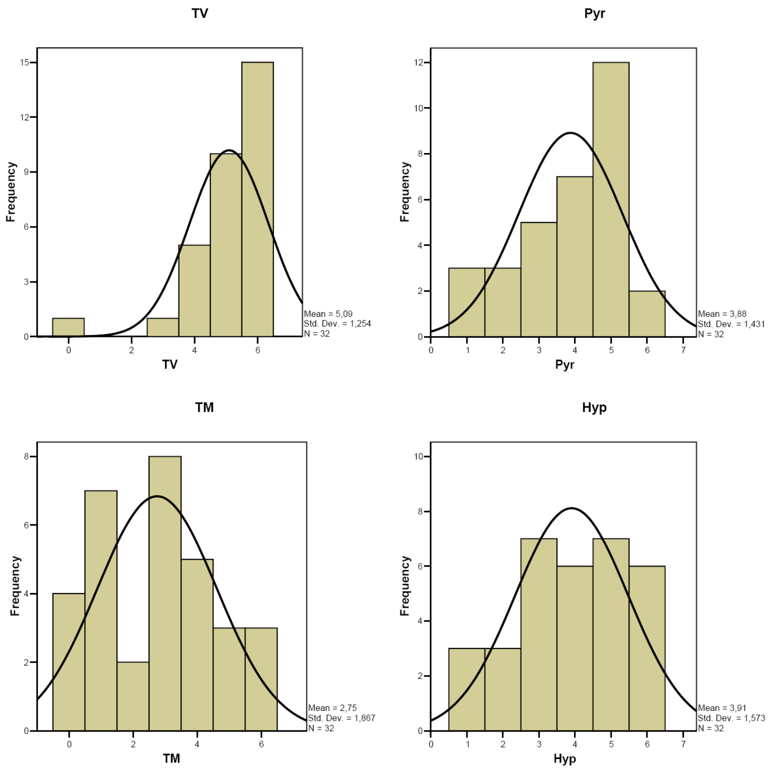


Figure 9.30: The histograms for ratings for question 7.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-3,388[a]	-4,272[a]	-2,890[a]	-2,586[a]	-,026[a]	-2,548[b]
Asymp. Sig. (2-tailed)	,001	,000	,004	,010	,980	,011

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.31: The results of the Wilcoxon test for ratings for question 7.

User No	TreeView	Pyramids	TreeMap	Hyperbolic
1	6	1	4	4
2	5	0	1	4
3	4	4	4	3
4	6	0	4	1
5	6	1	5	2
6	4	1	0	5
7	5	0	4	3
8	5	1	4	5
9	3	0	5	1
10	6	2	2	1
11	6	0	1	0
12	4	1	1	1
13	6	0	5	3
14	6	0	1	5
15	3	4	4	5
16	5	1	0	4
17	5	4	6	2
18	6	0	0	1
19	5	5	4	5
20	6	3	6	2
21	5	4	4	4
22	1	3	3	4
23	5	0	6	3
24	5	0	4	3
25	6	0	3	4
26	6	0	2	3
27	6	1	0	5
28	5	0	2	0
29	6	1	4	6
30	5	1	4	1
31	6	1	1	2
32	6	2	6	6
Av	5,13	1,28	3,13	3,06
Std Dev	1,16	1,53	1,95	1,74

Table 9.18: User ratings for question eight: orientation. Zero points mean very bad, six points mean very good rating for the given aspect.

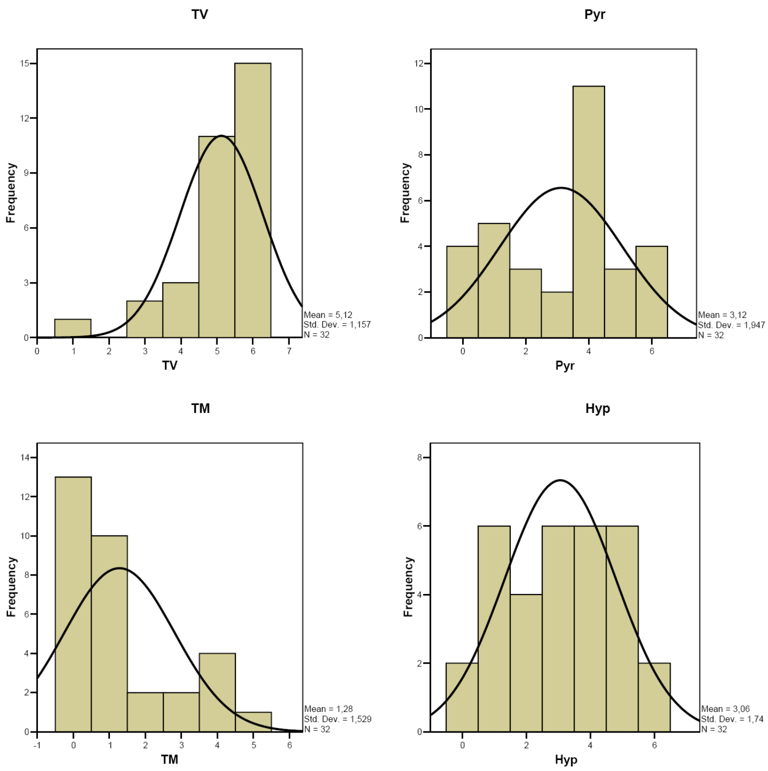


Figure 9.32: The histograms for ratings for question 8.

Test Statistics[c]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-3,737[a]	-4,690[a]	-3,913[a]	-3,851[a]	-,231[a]	-3,798[b]
Asymp. Sig. (2-tailed)	,000	,000	,000	,000	,817	,000

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.33: The results of the Wilcoxon test ratings for for question 8.

User No	TreeView	Pyramids	TreeMap	Hyperbolic
1	6	3	5	4
2	6	5	1	4
3	4	5	5	1
4	6	0	5	1
5	6	2	4	1
6	6	5	1	3
7	5	4	4	3
8	5	4	2	1
9	5	3	5	1
10	6	4	4	0
11	6	6	1	5
12	4	4	5	3
13	6	3	5	3
14	6	0	2	5
15	6	6	1	1
16	5	2	1	4
17	5	5	6	0
18	6	5	1	1
19	6	5	3	4
20	6	3	6	2
21	4	3	4	2
22	2	3	4	4
23	6	5	5	3
24	5	1	6	2
25	4	0	2	1
26	1	6	3	3
27	4	0	5	5
28	5	5	6	3
29	6	1	3	6
30	5	3	6	1
31	6	3	1	5
32	6	2	4	6
Av	5,16	3,31	3,63	2,75
Std Dev	1,22	1,86	1,81	1,74

Table 9.19: User ratings for question nine: navigation. Zero points mean very bad, six points mean very good rating for the given aspect.

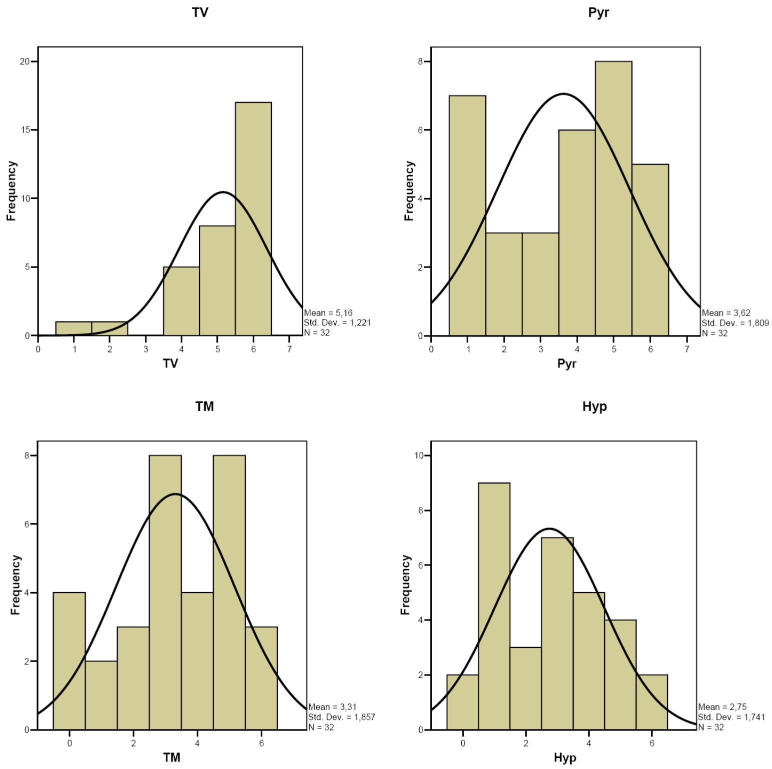


Figure 9.34: The histograms for ratings for question 9.

Test Statistics[b]

	Pyr - TV	TM - TV	Hyp - TV	TM - Pyr	Hyp - Pyr	Hyp - TM
Z	-3,082[a]	-3,716[a]	-4,299[a]	-,609[a]	-1,644[a]	-1,337[a]
Asymp. Sig. (2-tailed)	,002	,000	,000	,542	,100	,181

a. Based on positive ranks.

b. Wilcoxon Signed Ranks Test

Figure 9.35: The results of the Wilcoxon test for ratings for question 9.

9.5 Overall Preferences

The data for user preferences was gathered in the feedback questionnaire. After the participants rated the individual visualisations on different aspects, they chose one visualisation they thought was the best in each aspect. All votes were collected and combined in nine individual tables, one for each aspect.

A one-way Chi Square test for goodness of fit was used for analysis. The null hypothesis was that there would be no differences between the visualisations. Therefore, the same expected frequency for all visualisations was assumed. The expected frequency for Chi square should be ≥ 5. Here, there are 32 users voting for four categories (visualisations), yielding 8 expected votes for each category.

As there are problems when trying to calculate Chi Square with zero votes in one of the categories in SPSS, the analysis was conducted by hand for questions 6 (logical), 7 (useful) and 8 (orientation). Here, the Chi Square value was calculated and compared with the critical value for three degrees of freedom.

However, the Chi Square test only states that there are (or are not) significant differences between the visualisations. In order to find out where the differences are, a pairwise Chi Square analysis by hand was conducted. In this pairwise analysis, again the value of Chi square was calculated and compared with the critical value (for one degree of freedom).

The following notation will be used to show statistically significant differences (at level $p < o.o5$) between the browsers:

- $<$ meaning that the browsers left of $<$ are perceived as being significantly worse than the browsers right of $<$.
- $>$ meaning that the browsers left of $>$ are perceived as being significantly better than the browsers right of $>$.

9.5.1 Question 1: Best Overview

Visualisation	Frequency
TreeView	18
Pyramids	6
TreeMap	2
Hyperbolic	6

Table 9.20: Preferences for question 1. The table shows the number of votes for each browser.

Table 9.20 shows the user preferences in terms of overview for the individual visualisations. The users perceived the overview to be best in the TreeView. The Chi square test indicated that there are significant differences ($Sig. = 0.000$). In order to find those differences, pairwise Chi square tests were calculated by hand. The following formula was used:

$\chi^2 = \sum \frac{(O-E)^2}{E}$

O means the observed frequency, E means the expected frequency for the individual visualisations. The critical value for one degree of freedom is 3.8415. The expected frequency was calculated with the following formula:

$E = \frac{(O_1+O_2)}{2}$

where O_1 and O_2 are the observed frequencies for the two visualisations being compared.

For the pair TreeView - TreeMap, the Chi square test was calculated as follows:

$E = \frac{(O(TV)+O(TM))}{2} = \frac{(18+2)}{2} = 10$

Thus, the value of the expected frequency is 10 and can be used in the Chi square formula. Due to the fact that the expected frequency yields the average here, the value of χ^2 only has to be calculated once.

$\chi^2 = \sum \frac{(O-E)^2}{E} = \sum \frac{(18-10)^2}{10} = \sum \frac{(8)^2}{10} = 8$

$\chi^2 = \sum \frac{(O-E)^2}{E} = 8 + 8 = 16$

The obtained value for $\chi^2 = 16$ is greater than the critical value (3.8415). The difference between TreeView and TreeMap is thus significant.

All other combinations were calculated in the same manner. The differences between TreeView and Pyramids and TreeView and Hyperbolic were significant. The difference between TreeView and TreeMap was significant. The differences between TreeMap and Pyramids and TreeMap and Hyperbolic were not significant.

TreeView > Pyramids

TreeView > Hyperbolic

TreeView > TreeMap

In terms of overview, TreeView was significantly preferred over the other visualisations.

9.5.2 Question 2: Most Operable

Visualisation	Frequency
TreeView	23
Pyramids	5
TreeMap	2
Hyperbolic	2

Table 9.21: Preferences for question 2. The table shows the number of votes for each browser.

Table 9.21 shows that TreeView was absolutely preferred in terms of being most operable, TreeMap and Hyperbolic were equally disliked. The Chi square test ($Sig. = 0.000$) indicated significant differences. The differences were calculated by hand.

The differences between TreeView and Pyramids were significant. The differences between TreeView and TreeMap and between TreeView and Hyperbolic were significant. The differences between TreeMap and Pyramids and TreeMap and Hyperbolic were not significant.

TreeView > Pyramids

TreeView > Hyperbolic

TreeView > TreeMap

In terms of operability, TreeView was significantly preferred over to other visualisations.

9.5.3 Question 3: Most Intuitive

Visualisation	Frequency
TreeView	11
Pyramids	10
TreeMap	1
Hyperbolic	10

Table 9.22: Preferences for question 3. The table shows the number of votes for each browser.

TreeMap was considered by only one user to be most intuitive, other visualisations similarly popular (see Table 9.22). $Sig. = 0.041$ from the Chi square test proved significant differences. The differences were calculated by hand.

The differences between TreeView and Pyramids and TreeView and Hyperbolic were not significant. The difference between TreeView and TreeMap was significant. The differences between TreeMap and Pyramids and TreeMap and Hyperbolic were significant. In terms of intuitiveness, TreeMap was significantly disliked compared to the other visualisations.

TreeMap $<$ TreeView

TreeMap $<$ Pyramids

TreeMap $<$ Hyperbolic

9.5.4 Question 4: Most Usable

Visualisation	Frequency
TreeView	24
Pyramids	3
TreeMap	2
Hyperbolic	3

Table 9.23: Preferences for question 4. The table shows the number of votes for each browser.

TreeView was absolutely preferred for being most usable, other visualisations were less popular (see Table 9.23). The Chi square test found significant differences ($Sig. = 0.000$). The differences were calculated by hand.

The differences between TreeView and Pyramids and TreeView and Hyperbolic were significant. The differences between TreeView and TreeMap were significant. The differences between TreeMap and Pyramids and TreeMap and Hyperbolic were not significant. In terms of usability, TreeView was significantly preferred over the other visualisations.

TreeView $>$ Pyramids

TreeView $>$ Hyperbolic

TreeView $>$ TreeMap

9.5.5 Question 5: Most Understandable

Visualisation	Frequency
TreeView	20
Pyramids	5
TreeMap	1
Hyperbolic	6

Table 9.24: Preferences for question 5. The table shows the number of votes for each browser.

TreeView was considered most understandable by 20 of the 32 users. TreeMap only by one user. Pyramids and Hyperbolic were preferred similarly (see table 9.24). There were significant differences (Chi square: $Sig. = 0.000$). The differences were calculated by hand.

The differences between TreeView and Pyramids were significant. The differences between TreeView and TreeMap were significant. The differences between TreeView and Hyperbolic were significant. The differences between TreeMap and Pyramids and TreeMap and Hyperbolic were not significant. The differences between Pyramids and Hyperbolic were not significant. The differences between TreeMap Hyperbolic were not significant.

TreeView $>$ Pyramids

TreeView $>$ Hyperbolic

TreeView $>$ TreeMap

In terms of understandability, TreeView was significantly preferred over the other visualisations.

9.5.6 Question 6: Most Logical

Visualisation	Frequency
TreeView	15
Pyramids	6
TreeMap	0
Hyperbolic	11

Table 9.25: Preferences for question 6. The table shows the number of votes for each browser.

TreeView and Hyperbolic were considered logical, TreeMap not (see Table 9.25). Due to the zero votes for the TreeMap visualisation, the analysis could not be performed in SPSS. Instead, it was performed by hand using the following formula:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

O means the observed frequency, *E* means the expected frequency for the individual visualisations. The critical value for three degrees of freedom is 7.8147. The expected frequency is calculated with the following formula:

$$E = \frac{(O_1+O_2+O_3+O_4)}{4}$$

The value of the expected frequency is 8 (there were 32 votes for four visualisations). Now, the χ^2 values can be calculated for the individual visualisations.

TreeView: $\chi^2 = \frac{(O-E)^2}{E} = \frac{(15-8)^2}{8} = \frac{(7)^2}{8} = 6.125$

Pyramids: $\chi^2 = \frac{(O-E)^2}{E} = \frac{(6-8)^2}{8} = \frac{(2)^2}{8} = 0.5$

TreeMap: $\chi^2 = \frac{(O-E)^2}{E} = \frac{(0-8)^2}{8} = \frac{(8)^2}{8} = 8$

Hyperbolic: $\chi^2 = \frac{(O-E)^2}{E} = \frac{(11-8)^2}{8} = \frac{(3)^2}{8} = 1.125$

Then, those values were summed up to obtain the overall χ^2 value.

$\chi^2 = 6.125 + 0.5 + 8 + 1.125 = 15.75$

As the χ^2 value 15.75 is greater than the critical value of 7.8147, there are overall significant differences. The differences were also calculated by hand.

The differences between TreeView and Pyramids were significant. The differences between TreeView and TreeMap were significant. The differences between TreeView and Hyperbolic were not significant. The differences between TreeMap and Pyramids were significant. The differences between Pyramids and Hyperbolic were not significant. The differences between TreeMap and Hyperbolic were significant.

TreeView $>$ Pyramids

TreeMap $<$ TreeView

TreeMap $<$ Pyramids

TreeMap $<$ Hyperbolic

In terms of being logical, TreeMap was significantly disliked over the other visualisations.

9.5.7 Question 7: Most Useful

Visualisation	**Frequency**
TreeView	23
Pyramids	3
TreeMap	0
Hyperbolic	6

Table 9.26: Preferences for question 7. The table shows the number of votes for each browser.

TreeView was absolutely preferred in terms of being useful (see Table 9.26). TreeMap was not considered useful at all. The Chi Square test had to be performed by hand again, due to the zero votes for TreeMap.

$\chi^2(TV) = 28.125$

$\chi^2(Pyr) = 3.125$

$\chi^2(TM) = 8$

$\chi^2(Hyp) = 0.5$

$\sum \chi^2 = 31, 75 >> 7.8147$

There were significant differences. The differences were calculated by hand. The differences between TreeView and Pyramids were significant. The differences between TreeView and TreeMap were significant. The differences between TreeView and Hyperbolic were significant. The differences between TreeMap and Pyramids were not significant. The differences between Pyramids and Hyperbolic were not significant. The differences between TreeMap and Hyperbolic were significant.

TreeView > Pyramids

TreeView > Hyperbolic

TreeView > TreeMap

Hypebolic > TreeMap

In terms of usefulness, TreeView was significantly preferred over the other visualisations. Hyperbolic was significantly preferred to TreeMap.

9.5.8 Question 8: Best Orientation

Visualisation	Frequency
TreeView	19
Pyramids	6
TreeMap	0
Hyperbolic	7

Table 9.27: Preferences for question 8. The table shows the number of votes for each browser.

In terms of orientation, TreeView was preferred, TreeMap disliked and Pyramids and Hyperbolic similar (see Table 9.27). The Chi square value had to be calculated by hand again.

$\chi^2(TV) = 15.125$

$\chi^2(Pyr) = 0.5$

$\chi^2(TM) = 8$

$\chi^2(Hyp) = 0.125$

$\sum \chi^2 = 23,75 >> 7.8147$

There were significant differences. The differences were calculated by hand. The differences between TreeView and Pyramids were significant. The differences between TreeView and TreeMap were significant. The differences between TreeView and Hyperbolic were significant. The differences between TreeMap and Pyramids were significant. The differences between Pyramids and Hyperbolic were not significant. The differences between TreeMap Hyperbolic were significant.

TreeView > Pyramids

TreeView > Hyperbolic

TreeView > TreeMap

Pyramids > TreeMap

Hyperbolic > TreeMap

In terms of orientation, TreeView was significantly preferred over the other visualisations and TreeMap was significantly disliked over the other visualisaitions.

Visualisation	Frequency
TreeView	16
Pyramids	7
TreeMap	2
Hyperbolic	7

Table 9.28: Preferences for question 9. The table shows the number of votes for each browser.

Test Statistics[c]

	B1 - A1	C1 - A1	D1 - A1	C1 - B1	D1 - B1	D1 - C1
Z	-2,880[a]	-,262[a]	-,206[b]	-2,936[b]	-3,628[b]	-1,141[b]
Asymp. Sig. (2-tailed)	,004	,793	,837	,003	,000	,254

a. Based on negative ranks.

b. Based on positive ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.36: The Wilcoxon test results comparing the tasks A1, B1, C1 and D1.

9.5.9 Question 9: Best Navigation

TreeView's navigation was preferred, Pyramids and Hyperbolic were considered equal, TreeMap disliked (see Table 9.28). The Chi square test indicated significant differences ($Sig. = 0.005$). The differences were calculated by hand.

The differences between TreeView and Pyramids and between TreeView and Hyperbolic were not significant. The differences between TreeView and TreeMap were significant. The differences between TreeMap and Pyramids and TreeMap and Hyperbolic were not significant. In terms of navigation, TreeView was significantly preferred to TreeMap.

TreeView $>$ TreeMap

9.6 Task Set Comparison

Corresponding tasks in each task sets were designed to be similarly hard or easy for the users. To check this, an analysis comparing the task sets was performed. The task completion times for the individual tasks in the different task sets were compared. Due to the large variance of task completion times, there were differences between the tasks.

9.6.1 Tasks A1-D1: Deepest Subdirectory

Task completion times ordered by tasks are given in Table 9.29. The distributions were not normal (Kolmogorov-Smirnov), therefore a Friedman test was performed. There were overall significant differences ($Sig. = 0.011$). A pairwise Wilcoxon test was performed to find those differences. Task B1 took significantly longer to complete than the other tasks (A1, C1, and D1). Finding the deepest subdirectory in the directory "jazz" took longer because this directory has many subdirectories, and several have similar depth.

User No	Task A1	Task B1	Task C1	Task D1
1	201,5	133,9	46,4	54,1
2	81,9	247,6	199,1	83,4
3	189,1	118,2	195,3	102,1
4	38,2	36,0	38,8	18,3
5	75,1	124,5	132,5	48,4
6	48,2	85,3	19,5	98,3
7	89,0	633,0	269,2	161,0
8	68,7	33,1	94,5	34,7
9	141,3	89,9	72,2	56,4
10	49,3	473,9	49,4	92,4
11	29,8	192,1	153,1	112,5
12	54,5	37,8	93,1	53,8
13	46,8	84,5	41,9	100,5
14	45,7	45,7	42,2	45,8
15	77,7	26,0	36,3	122,3
16	110,2	152,6	79,7	243,1
17	78,3	51,3	74,7	38,5
18	38,8	37,3	89,6	51,3
19	85,9	221,0	63,8	55,4
20	160,7	176,0	63,3	87,0
21	40,9	48,2	41,9	25,0
22	67,7	213,0	44,4	41,3
23	41,2	146,1	55,8	52,7
24	51,6	109,6	148,2	62,6
25	53,4	37,3	36,3	19,7
26	43,3	476,2	55,8	56,8
27	55,6	144,0	39,4	50,0
28	68,1	169,9	108,4	134,9
29	94,8	398,6	33,3	33,3
30	93,8	125,4	78,7	28,4
31	46,1	238,8	54,8	142,1
32	54,7	72,6	77,5	57,5
Mean	75,7	161,9	82,2	73,9
Std. Dev.	43,0	146,4	56,9	48,3

Table 9.29: Task completion times for tasks A1 to D1. Times are given in seconds to one decimal place.

Test Statistics[c]

	B2 - A2	C2 - A2	D2 - A2	C2 - B2	D2 - B2	D2 - C2
Z	-2,244[a]	-,187[b]	-,692[a]	-2,001[b]	-2,169[b]	-,262[a]
Asymp. Sig. (2-tailed)	,025	,852	,489	,045	,030	,793

a. Based on negative ranks.

b. Based on positive ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.37: The Wilcoxon test results comparing the tasks A2, B2, C2 and D2.

Test Statistics[c]

	B3 - A3	C3 - A3	D3 - A3	C3 - B3	D3 - B3	D3 - C3
Z	-1,870[a]	-2,132[a]	-,879[a]	-,748[b]	-1,496[b]	-,056[a]
Asymp. Sig. (2-tailed)	,061	,033	,379	,454	,135	,955

a. Based on positive ranks.

b. Based on negative ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.38: The Wilcoxon test results comparing the tasks A3, B3, C3 and D3.

9.6.2 Tasks A2-D2: Most Subdirectories

Task completion times ordered by tasks are given in Table 9.30. The distribution for task B2 was normal, the other distributions were not normal (Kolmogorov-Smirnov). A Friedman test indicated overall significant differences ($Sig. = 0.048$). From the Wilcoxon test can be deducted that task B2 took significantly longer to complete than the other tasks. This could have been caused by the fact that the number of subdirectories was very similar (6 and 7). Users recounted the subdirectories again to find the right one.

9.6.3 Tasks A3-D3: Find Directory

Task completion times ordered by tasks are given in Table 9.31. The distributions were not normal (Kolmogorov-Smirnov), therefore a Friedman test was performed. There were no overall significant differences ($Sig. = 0.397$). However, a pairwise Wilcoxon test was performed to look for pairwise differences. Task A3 took significantly longer to accomplish than task C3. This difference is due to user variation. In task A3, the directory to be found was one of five directories, in task C3, one of three.

9.6.4 Tasks A4-D4: Find File

Task completion times ordered by tasks are given in Table 9.32. The distribution for task C4 was normal, the others were not normal (Kolmogorov-Smirnov). A Friedman test was performed, showing no overall significant differences ($Sig. = 0.199$). A pairwise Wilcoxon test was also performed. Task A4 took significantly longer to accomplish than tasks B4 and C4. In task A4, the file was one of 16 files in the directory. In tasks B4 and C4, the file was the only file in the given directory. Having to read the names of the files to find the given one could have slowed the users down. However, in task D4, the file was one of 19 files, but

User No	Task A2	Task B2	Task C2	Task D2
1	38,6	78,3	42,5	71,1
2	93,0	49,4	44,7	41,5
3	106,3	134,2	75,0	111,7
4	25,3	33,8	25,5	27,8
5	129,0	80,8	65,8	24,8
6	21,3	47,3	16,8	40,3
7	84,7	76,7	76,5	108,1
8	73,4	92,5	66,5	44,1
9	57,6	89,2	50,7	41,5
10	45,7	97,9	56,9	62,5
11	24,7	27,5	47,2	60,7
12	107,0	29,0	37,2	24,8
13	55,7	127,0	38,6	65,9
14	58,6	41,8	42,4	46,8
15	62,8	60,5	91,3	74,0
16	29,1	25,7	59,6	52,4
17	33,5	26,6	18,4	35,0
18	24,5	46,5	26,5	32,0
19	72,8	125,2	60,1	47,7
20	70,7	143,4	63,2	65,1
21	50,8	96,5	22,7	55,0
22	53,0	105,8	49,6	79,0
23	28,1	19,3	84,1	33,9
24	98,0	136,7	141,4	63,8
25	40,5	33,6	17,8	19,4
26	65,0	120,0	87,8	67,3
27	17,9	26,5	50,0	37,2
28	51,9	102,3	40,4	61,9
29	35,5	37,8	70,5	37,6
30	44,8	33,4	25,5	48,2
31	47,1	54,5	94,9	50,2
32	19,8	23,3	44,5	87,7
Mean	55,2	69,5	54,2	53,7
Std. Dev.	28,7	39,9	27,2	22,4

Table 9.30: Task completion times for tasks A2 to D2. Times are given in seconds to one decimal place.

User No	Task A3	Task B3	Task C3	Task D3
1	12,9	27,5	15,6	49,7
2	20,3	11,9	37,8	30,7
3	54,5	60,3	110,4	58,2
4	22,5	29,5	13,2	16,7
5	191,0	46,7	60,2	8,6
6	13,6	21,2	38,3	31,7
7	49,6	28,4	106,8	106,3
8	49,9	8,8	25,6	16,2
9	158,4	19,6	43,1	13,4
10	79,7	53,7	11,6	77,5
11	16,7	36,8	19,6	35,2
12	61,6	21,2	52,1	26,2
13	27,4	48,1	15,5	51,3
14	136,1	25,6	33,7	14,9
15	56,1	41,8	19,4	15,3
16	13,3	64,0	20,8	124,7
17	9,5	18,4	24,8	53,2
18	32,4	22,0	29,7	22,5
19	78,9	12,0	24,0	18,9
20	123,2	345,6	40,0	165,6
21	26,3	23,1	13,7	13,7
22	15,7	36,1	26,5	42,4
23	116,7	8,5	26,7	22,5
24	185,2	21,3	77,8	37,8
25	25,4	9,8	17,8	6,8
26	37,1	44,5	14,7	28,3
27	8,7	19,2	26,3	26,9
28	39,7	11,2	26,4	23,4
29	49,3	9,6	57,7	34,3
30	38,8	11,0	20,9	8,5
31	25,5	32,1	17,7	30,9
32	15,5	17,2	22,2	44,9
Mean	14,2	22,3	18,9	47,3
Std. Dev.	1,8	7,3	4,7	3,4

Table 9.31: Task completion times for tasks A3 to D3. Times are given in seconds to one decimal place.

User No	Task A4	Task B4	Task C4	Task D4
1	28,1	48,6	17,9	35,4
2	40,3	15,1	66,6	42,1
3	184,7	93,8	79,9	42,4
4	47,8	27,6	11,5	75,8
5	181,7	35,7	56,0	19,2
6	21,3	28,3	16,3	29,2
7	56,1	22,5	107,5	82,5
8	103,7	29,5	31,2	18,6
9	73,2	28,4	49,0	24,3
10	104,8	88,0	21,6	49,4
11	19,8	33,4	20,6	39,4
12	36,5	16,5	42,8	41,2
13	64,0	51,3	35,5	79,1
14	90,7	92,3	40,6	16,9
15	102,3	38,5	10,5	164,1
16	25,3	61,0	31,1	47,6
17	21,0	33,9	15,0	28,6
18	67,1	18,3	51,8	58,1
19	110,1	36,6	36,0	27,3
20	243,3	154,3	28,7	86,3
21	39,0	44,3	10,3	65,6
22	26,6	37,4	17,8	38,7
23	20,5	12,2	49,8	33,4
24	96,9	51,9	58,5	35,5
25	66,5	35,1	24,5	10,7
26	66,1	83,2	18,4	28,8
27	17,7	24,5	14,9	31,0
28	143,6	13,8	42,4	37,3
29	16,8	28,0	52,0	30,1
30	39,7	27,5	28,2	12,2
31	73,2	87,2	19,5	46,1
32	16,4	21,9	24,8	25,6
Mean	22,3	35,3	21,4	30,5
Std. Dev.	55,5	31,2	22,1	29,6

Table 9.32: Task completion times for tasks A4 to D4. Times are given in seconds to one decimal place.

Test Statistics[c]

	B4 - A4	C4 - A4	D4 - A4	C4 - B4	D4 - B4	D4 - C4
Z	-2,674[a]	-3,310[a]	-1,683[a]	-1,103[a]	-,262[a]	-,692[b]
Asymp. Sig. (2-tailed)	,007	,001	,092	,270	,793	,489

a. Based on positive ranks.
b. Based on negative ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.39: The Wilcoxon test results comparing the tasks A4, B4, C4 and D4.

Test Statistics[c]

	B5 - A5	C5 - A5	D5 - A5	C5 - B5	D5 - B5	D5 - C5
Z	-1,066[a]	-1,384[a]	-1,197[a]	-,187[b]	-,393[b]	-,673[b]
Asymp. Sig. (2-tailed)	,286	,166	,231	,852	,695	,501

a. Based on negative ranks.
b. Based on positive ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.40: The Wilcoxon test results comparing the tasks A5, B5, C5 and D5.

task D4 was not significantly slower than other tasks. This difference is thus probably due to user variation.

9.6.5 Tasks A5-D5: Count Subdirectories

Task completion times ordered by tasks are given in Table 9.33. The distributions were all not normal (Kolmogorov-Smirnov). A Friedman test indicated no overall significant differences ($Sig. = 0.480$). Nevertheless, a pairwise Wilcoxon test was performed. There were no differences between the tasks, meaning that no task took significantly longer to accomplish than any other task.

9.6.6 Tasks A6-D6: Count Files

Task completion times ordered by tasks are given in Table 9.34. The distribution for D6 was normal, the other distributions were not normal (Kolmogorov-Smirnov). The Friedman test showed no overall significant differences ($Sig. = 0.473$). A pairwise Wilcoxon test was performed and showed no significant differences in completion times between the tasks.

9.6.7 Tasks A7-D7: Compare Subdirectories

Task completion times ordered by tasks are given in Table 9.35. The distributions for B7 and C7 were normal and for A7 and D7 were not normal (Kolmogorov-Smirnov). The Friedman test showed no overall significant differences ($Sig. = 0.098$). A pairwise Wilcoxon test was performed. Task C7 took significantly longer to accomplish than task A7. In task C7, the two directories to be compared had five and seven subdirectories. In task A7, the number of subdirectories was four and five. Again, this difference is most probably due to user variance.

User No	Task A5	Task B5	Task C5	Task D5
1	13,8	32,6	20,7	59,8
2	26,0	10,7	21,3	26,8
3	19,3	84,3	46,3	34,1
4	17,2	18,2	10,7	17,5
5	20,3	43,2	38,4	15,6
6	10,1	17,5	12,1	20,3
7	42,1	25,5	89,7	37,5
8	25,8	31,7	21,7	10,6
9	72,7	58,4	42,4	10,9
10	26,9	31,3	22,3	30,5
11	12,5	26,4	21,4	23,2
12	19,8	10,6	19,4	19,7
13	23,8	23,4	16,7	55,7
14	14,0	32,3	20,7	13,0
15	24,8	18,0	15,7	58,4
16	13,0	45,1	76,8	32,6
17	13,5	22,9	32,0	19,4
18	26,5	15,4	13,6	17,6
19	19,2	24,4	31,7	14,0
20	54,6	39,2	37,2	70,1
21	13,7	21,5	12,8	45,6
22	11,9	37,4	36,2	31,4
23	14,6	9,6	10,1	12,0
24	35,5	81,4	92,8	19,5
25	28,9	9,1	17,0	5,4
26	81,2	28,9	18,2	22,0
27	10,0	22,2	45,1	19,4
28	16,4	10,4	43,1	13,0
29	18,3	7,4	21,3	18,1
30	13,1	30,2	26,4	25,4
31	15,8	18,9	16,8	35,9
32	14,6	21,2	19,2	17,7
Mean	24,0	28,4	30,3	26,6
St. Dev.	16,9	18,4	21,2	16,0

Table 9.33: Task completion times for tasks A5 to D5. Times are given in seconds to one decimal place.

User No	Task A6	Task B6	Task C6	Task D6
1	15,5	53,8	59,2	39,3
2	19,0	12,6	19,5	15,8
3	139,7	58,9	44,7	20,3
4	12,2	11,0	8,3	29,5
5	48,6	40,8	19,5	7,3
6	11,4	31,0	45,0	18,5
7	40,0	33,4	30,0	35,5
8	25,1	42,9	8,5	9,2
9	44,8	63,6	31,7	6,8
10	17,6	27,5	13,6	43,5
11	10,8	15,1	53,9	30,5
12	59,3	9,0	22,7	10,7
13	27,1	15,8	12,5	91,5
14	18,4	55,3	20,2	7,9
15	22,3	17,0	20,0	19,4
16	11,0	22,9	46,3	40,8
17	10,6	13,5	16,9	30,3
18	16,7	9,4	12,1	10,6
19	28,6	88,1	29,0	12,1
20	90,5	25,0	19,3	64,8
21	21,6	14,0	14,6	19,5
22	12,8	27,0	36,0	28,8
23	20,3	7,1	19,2	14,5
24	58,3	60,9	27,8	9,7
25	14,9	45,5	7,9	3,6
26	37,9	26,4	14,6	32,3
27	10,4	13,2	36,3	31,6
28	20,5	10,9	49,6	8,7
29	15,6	11,9	14,8	8,9
30	18,7	27,4	11,1	12,1
31	19,7	12,3	11,7	21,6
32	14,1	12,6	34,9	23,8
Mean				
Std. Dev.	26,9	20,4	14,5	18,4

Table 9.34: Task completion times for tasks A6 to D6. Times are given in seconds to one decimal place.

User No	Task A7	Task B7	Task C7	Task D7
1	20,8	55,7	62,4	35,8
2	40,0	24,6	72,7	66,7
3	82,2	47,2	62,6	42,3
4	22,3	32,6	15,9	41,8
5	42,3	42,8	54,4	19,1
6	17,8	51,0	43,1	40,1
7	58,3	52,4	74,8	144,5
8	18,5	58,6	34,7	27,6
9	120,3	68,5	67,1	30,2
10	32,5	54,5	26,6	73,1
11	17,1	79,2	44,7	60,5
12	30,5	19,7	56,5	55,5
13	28,9	44,5	26,4	80,1
14	33,4	69,0	35,2	20,5
15	26,4	47,6	22,5	82,5
16	22,8	67,4	85,3	54,0
17	21,6	59,9	55,2	37,7
18	49,7	35,8	62,5	30,8
19	37,5	34,1	69,0	41,2
20	46,2	56,5	50,9	55,5
21	28,2	45,7	14,4	19,1
22	20,8	45,4	47,8	41,5
23	75,7	19,2	44,2	54,9
24	28,5	50,8	102,8	32,0
25	22,1	14,2	35,8	16,8
26	30,8	82,9	29,3	66,9
27	15,3	37,1	52,5	38,7
28	78,2	26,6	77,2	40,3
29	56,7	34,1	39,0	34,1
30	29,9	64,2	47,7	16,3
31	26,6	35,0	23,7	71,5
32	12,4	34,0	52,9	32,8
Mean	37,3	46,6	49,7	47,0
St. Dev.	23,7	17,2	20,8	25,7

Table 9.35: Task completion times for tasks A7 to D7. Times are given in seconds to one decimal place.

Test Statistics[c]

	B6 - A6	C6 - A6	D6 - A6	C6 - B6	D6 - B6	D6 - C6
Z	-,224[a]	-,935[b]	-,673[b]	-,477[b]	-,206[b]	-1,290[b]
Asymp. Sig. (2-tailed)	,822	,350	,501	,633	,837	,197

a. Based on negative ranks.
b. Based on positive ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.41: The Wilcoxon test results comparing the tasks A6, B6, C6 and D6.

Test Statistics[c]

	B7 - A7	C7 - A7	D7 - A7	C7 - B7	D7 - B7	D7 - C7
Z	-1,851[a]	-2,412[a]	-1,851[a]	-,561[a]	-,393[b]	-,898[b]
Asymp. Sig. (2-tailed)	,064	,016	,064	,575	,695	,369

a. Based on negative ranks.
b. Based on positive ranks.
c. Wilcoxon Signed Ranks Test

Figure 9.42: The Wilcoxon test results comparing the tasks A7, B7, C7 and D7.

9.6.8 Tasks A8-D8: Compare Files

Task completion times ordered by tasks are given in Table 9.36. The distributions were not normal (Kolmogorov-Smirnov). The Friedman test indicated overall significant differences ($Sig. = 0.010$). A pairwise Wilcoxon test was performed to find those differences. Task D8 took significantly longer to accomplish than the other tasks. The number of files in the directories for all tasks was very similar, ranging from nine to thirteen. In this case, the tasks were similar and the difference can not be given by the task being more difficult.

9.7 User Comments

Users were given the possibility to write down their comments about the individual browsers. Some suggestions for improvement were given as well:

- In TreeView, develop an automatic positioning after opening/closing directories.
- In TreeMap, provide a faster method for returning to the uppermost level.
- In TreeMap, provide a possibility to switch to alphabetical ordering.
- In all visualisations, show the path to the current position and selected item.
- In Pyramids, the name of the selected (single-click) directory or file should be displayed.
- In Pyramids, panning should also be available by scroll bars.
- In Hyperbolic, provide the possibility to return to the uppermost directory by clicking it.

User No	Task A8	Task B8	Task C8	Task D8
1	17,5	80,2	89,7	114,8
2	42,0	33,2	32,2	50,7
3	82,6	90,3	61,5	48,0
4	31,5	24,3	17,2	56,9
5	58,7	52,5	33,2	39,6
6	26,6	34,7	47,4	61,7
7	85,5	36,3	104,2	52,8
8	29,2	51,4	48,2	24,5
9	67,1	64,9	44,5	70,4
10	40,3	58,7	30,2	73,9
11	28,3	59,1	85,9	121,6
12	26,3	31,6	31,7	49,7
13	39,6	49,2	22,3	121,7
14	28,6	62,0	28,7	43,7
15	29,5	25,0	11,7	22,9
16	27,0	67,3	93,9	248,0
17	21,1	44,9	38,3	59,2
18	58,5	24,9	16,0	40,3
19	35,6	44,4	33,8	48,3
20	63,2	55,0	47,2	156,5
21	18,2	35,2	17,9	133,2
22	19,3	29,7	46,2	66,6
23	51,7	25,3	34,0	39,1
24	68,1	162,6	53,7	50,0
25	21,3	27,7	18,2	14,8
26	40,6	73,3	29,2	68,5
27	22,8	31,2	43,9	59,5
28	87,8	42,4	45,5	45,9
29	40,5	25,8	33,7	29,9
30	14,0	24,3	18,8	12,7
31	24,1	46,3	27,8	77,2
32	18,1	29,7	47,2	45,4
Mean	39,5	48,2	41,7	67,1
St. Dev.	21,2	27,4	23,3	47,5

Table 9.36: Task completion times for tasks A8 to D8. Times are given in seconds to one decimal place.

Test Statistics[c]

	B8 - A8	C8 - A8	D8 - A8	C8 - B8	D8 - B8	D8 - C8
Z	-1,945[a]	-,019[b]	-2,580[a]	-1,496[b]	-2,618[a]	-3,478[a]
Asymp. Sig. (2-tailed)	,052	,985	,010	,135	,009	,001

a. Based on negative ranks.

b. Based on positive ranks.

c. Wilcoxon Signed Ranks Test

Figure 9.43: The Wilcoxon test results comparing the tasks A8, B8, C8 and D8.

- In Hyperbolic, display the names of the clicked directories or files.
- In Hyperbolic, adjust the zoom speed.

9.8 Discussion

Due to large variations in task completion times among the users, almost no statistically significant differences between the visualisations were found. Only for one task, counting the number of files in a directory, was a statistically significant difference found. For this task, TreeMap was significantly faster than Hyperbolic browser, probably due to the fact that TreeMap uses most of the screen space to show the contents of a directory and colour codes the files. Apparently, the icon file representation in Hyperbolic was more difficult for counting.

TreeMap performed fastest, though mostly with very slight (and not statistically significant) differences, for the following tasks:

- Task 2: Find the directory with most subdirectories (no significant differences).
- Task 3: Navigate to directory (no significant differences).
- Task 4: Navigate to file (no significant differences).
- Task 6: Count files in a directory (TreeMap significantly faster than Hyperbolic).
- Task 8: Compare number of files in two directories (no significant differences).

The explicit distinction in the representation of files and directories in TreeMap (directories are frames, files are coloured rectangles) and rather simple navigation (only using the left mouse button) made TreeMap perform faster than TreeView. Even the unusual ordering by size rather than alphabetically did not affect the performance of users. This drawback was probably counterbalanced by the fact that the entire hierarchy was visible and most of directory and file names were legible without navigation.

No statistically significant differences between the browsers were found regarding successful task completion (effectiveness). All browsers depict the hierarchical structure equally well to users for the used types of tasks. Although different representations are used in the individual visualisations, users learn to discern these differences. Users were similarly successful in completing tasks with all four representations.

The subjective ratings varied greatly from the objective measurements. TreeMap was rated significantly worse than the other visualisations for all aspects. TreeView received the highest ratings far all aspects. TreeView was rated statistically significantly better than the other visualisations. Pyramids and Hyperbolic were rated equally for most of the aspects. In terms of overview, Hyperbolic was rated statistically significantly worse than TreeView. Though having similarities (node-link representation and alphabetical ordering), the Hyperbolic layout probably overwhelmed the users. Also in terms of operability, Hyperbolic was rated statistically significantly worse than Pyramids and TreeView. Users probably need more time to learn to operate Hyperbolic more efficiently. The combination of automatic zooming, free panning, and free zooming seems to require more practice.

In terms of overall preference, TreeView was rated statistically significantly as having the best overview, operability, being most usable, understandable, and useful, and offering the best orientation. These results clearly show a preference for familiarity over novelty. Users probably assumed they performed fastest with TreeView and rated it best. TreeMap was rated significantly worst as being least intuitive and logical, and offering the worst orientation. However, as can be seen from the objective measurements, users in fact did understand TreeMap's representation and could perform the tasks efficiently.

The task sets were also compared with each other. Although carefully planned, some tasks took significantly longer to perform than others. For the task of finding the deepest subdirectory, task B1 took significantly longer. This was due to the fact that the given directory had several subdirectories and some of them had the same depth. Most differences between the visualisations are due to user variation. No statistically significant differences between the task sets for the tasks of counting the subdirectories and files were found.

Although none of the users knew the other tested visualisations (TreeMap, Pyramids, and Hyperbolic), they were able to learn them quickly. Users with diverse backgrounds could to become acquainted with novel visualisations. No statistically significant differences (except in one case) between the visualisations were found. On the one hand, this could be due to large variation in task completion times among the users. On the other hand, however, this result proves that the studied visualisations are similar in performance for the tasks used in this study. All visualisations represent the data equally good and the users performed similarly well working with all of them.

Chapter 10

Outlook

The area of information visualisation has brought new methods for many different applications. With the PCs becoming more powerful, 3D techniques have emerged. This thesis presented some of the techniques for different tasks. However, very few techniques have managed to leave the laboratory and become commercialised. Especially in the visualisation of hierarchies, the predominant spread of tree view-like browsers made it more difficult for new methods to establish themselves.

The evaluation of new techniques in different contexts is very important. Only through evaluation can the strengths and weaknesses of the techniques be found. User experiments are an appropriate method for testing.

In the study planned and conducted for this thesis, four hierarchy visualisations were tested. The Hierarchical Visualisation System (HVS) is a framework combining different visualisations. The TreeView, Pyramid, TreeMap and Hyperbolic browsers were tested. The hierarchical Visualisation Testing Environment (HVTE) was developed to ease the running of the test and data collection. In future, other HVS visualisations can be tested with HVTE.

Although almost no significant differences in task completion times were found, users strongly preferred the TreeView visualisation. However, a majority would welcome the new methods as an additional option. For everyday use, close to all would prefer the familiar TreeView. The suggestions for improvement by the users are very valuable.

In future, more usability tests are needed. However, the findings of existing studies must not be forgotten. The user comments should be used to improve the existing visualisations and open new ways for further ones.

As new visualisations are included in HVS, they should be evaluated as well. HVTE provides a good testing environment to test these HVS visualisations straightforwardly.

In future, not only the individual visualisations should be evaluated. HVS provides a series of features which may further help to improve the usability and performance of visualisations. For instance, the synchronized view of different visualisations may provide better access to the desired data. There is still much further work to do for which this thesis should provide essential and helpful information.

Appendix A

Materials Used in the Study

These are English translations of the actual materials used in the study. The German originals as well as the original tasks can be found in Appendix B.

A.1 Non-Disclosure and Consent Form

Non-Disclosure and Consent Form

Thank you for participating in our study. Please be aware that confidential information may be disclosed to you and that you must not reveal information that you learn during the course of your participation. In addition, your session will be videotaped, to allow others who are not present to observe your session and benefit from your feedback.

Please read the statements below and sign where indicated. Thank you.

I agree that I will disclose no information about the study.

I understand that video and audio recordings will be made of my session. I grant permission to use these recordings for teaching purposes.
Date:
Name:
Signature:

A.2 Background Questionnaire

Date Time Test No User No

Background Questionnaire

Thank you for participating in our test. Please answer the following questions:

1. General Information

Sex: [ ] male [ ] female

Age:

Occupation:

2. Sight Impairment

1. Do you use a sight aid when working on the computer?

[ ] none [ ] glasses [ ] contact lenses [ ] other

2. Do you have any form of colour blindness?

[ ] no [ ] yes,

3. Education

1. Educational Level Attained:

[ ] vocational training [ ] secondary school [ ] university degree [ ] doctorate

2. If you are studying or have studied, please describe your main area of study:

4. Use of Computers

1. How long have you been using a personal computer?

years

2. How many hours per week do you use a computer?

hours

3. Which kind of computer do you normally use?

[ ] Microsoft Windows [ ] Apple Macintosh [ ] Unix [ ] Linux [ ] Other

5. Data Acquaintance

1. Which data management program do you use most?

[ ] Windows Explorer [ ] Windows Commander [ ] command line (such as MS Dos) [ ] Other

2. Where do you have the most data to manage?

[ ] at home [ ] at work [ ] university [ ] other

3. Do you perform information retrieval (searching for information in large amounts of data)?

[ ] daily [ ] weekly [ ] monthly [ ] seldom [ ] never

6. Experience with Usability Tests

1. Have you participated in a usability study before?

[ ] yes [ ] no

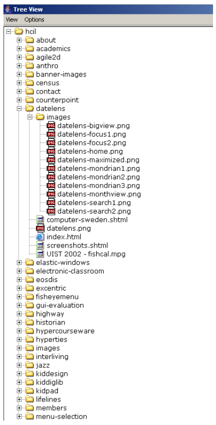
Date: Topic:

Thinking Aloud Test: yes [ ] no [ ]

A.3 Feedback Questionnaire

Date Time Test No User No

Feedback Questionnaire (Visualisations)



Please, rate the following aspects

The **TreeView** visualisation is

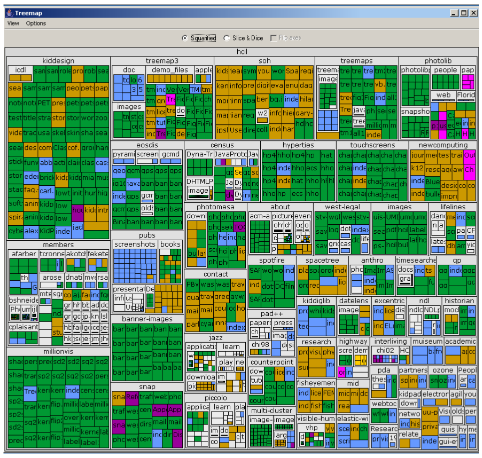
overview is bad	3	2	1	0	1	2	3	overview is good
operability is bad	3	2	1	0	1	2	3	operability is good
less intuitive	3	2	1	0	1	2	3	very intuitive
usability is bad	3	2	1	0	1	2	3	usability is good
less understandable	3	2	1	0	1	2	3	very understandable
less logical	3	2	1	0	1	2	3	very logical
less useful	3	2	1	0	1	2	3	very useful
orientation is bad	3	2	1	0	1	2	3	orientation is good
navigation is bad	3	2	1	0	1	2	3	navigation is good

Positive comments:

Negative comments:

Would you use this visualisation? Why (not)?

Feedback Questionnaire (Visualisations)



Please, rate the following aspects

The **TreeMap** visualisation is

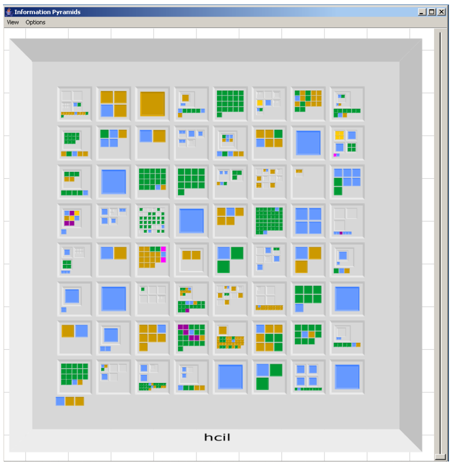
overview is bad	3	2	1	0	1	2	3	overview is good
operability is bad	3	2	1	0	1	2	3	operability is good
less intuitive	3	2	1	0	1	2	3	very intuitive
usability is bad	3	2	1	0	1	2	3	usability is good
less understandable	3	2	1	0	1	2	3	very understandable
less logical	3	2	1	0	1	2	3	very logical
less useful	3	2	1	0	1	2	3	very useful
orientation is bad	3	2	1	0	1	2	3	orientation is good
navigation is bad	3	2	1	0	1	2	3	navigation is good

Positive comments:

Negative comments:

Would you use this visualisation? Why (not)?

Feedback Questionnaire (Visualisations)



Please, rate the following aspects

The **Pyramids** visualisation is

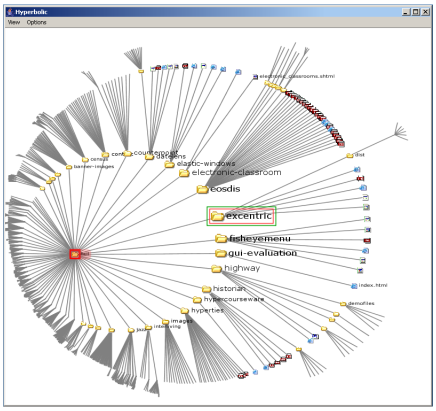
overview is bad	3	2	1	0	1	2	3	overview is good
operability is bad	3	2	1	0	1	2	3	operability is good
less intuitive	3	2	1	0	1	2	3	very intuitive
usability is bad	3	2	1	0	1	2	3	usability is good
less understandable	3	2	1	0	1	2	3	very understandable
less logical	3	2	1	0	1	2	3	very logical
less useful	3	2	1	0	1	2	3	very useful
orientation is bad	3	2	1	0	1	2	3	orientation is good
navigation is bad	3	2	1	0	1	2	3	navigation is good

Positive comments:

Negative comments:

Would you use this visualisation? Why (not)?

Feedback Questionnaire (Visualisations)



Please, rate the following aspects

The **Hyperbolic** visualisation is

overview is bad	3	2	1	0	1	2	3	overview is good
operability is bad	3	2	1	0	1	2	3	operability is good
less intuitive	3	2	1	0	1	2	3	very intuitive
usability is bad	3	2	1	0	1	2	3	usability is good
less understandable	3	2	1	0	1	2	3	very understandable
less logical	3	2	1	0	1	2	3	very logical
less useful	3	2	1	0	1	2	3	very useful
orientation is bad	3	2	1	0	1	2	3	orientation is good
navigation is bad	3	2	1	0	1	2	3	navigation is good

Positive comments:

Negative comments:

Would you use this visualisation? Why (not)?

Appendix B

Original Materials Used in the Study (in German)

These are the actual materials used in the study: the German originals as well as the original tasks.

B.1 Vertraulichkeits- und Einverständniserklärung

Vertraulichkeits- und Einverständniserklärung

Danke, daß Sie an unserer Studie teilnehmen. Bitte beachten Sie, daß Ihnen unter Umständen vertrauliche Informationen zuteilt werden und daß Sie diese nicht weitergeben dürfen. Ihre Sitzung wird auf Video aufgenommen, um bei Unklarheiten in der Analyse darauf zurückgreifen zu können und zur Unterstützung der Lehre.

Bitte lesen Sie die untenstehende Einverständniserklärung und unterschrieben Sie an der dafür vorgesehenen Stelle. Vielen Dank.

Ich erkläre, keine Informationen aus der Studie weiterzugeben.

Ich weiß, daß Bild- und Tonaufnahmen von meiner Sitzung gemacht werden. Ich gebe die Erlaubnis, diese Aufnahmen für Lehrzwecke zu verwenden.
Datum:
Name:
Unterschrift:

B.2 Hintergrundbefragung

Datum Zeit Test Nr. User Nr.

Hintergrundbefragung

Danke, daß Sie sich als Freiwilliger für unseren Test zur Verfügung stellen. Bitte beantworten Sie die folgenden Fragen:

1. Angaben zur Person

Geschlecht: [ ] mänlich [ ] weiblich

Alter:

Beruf:

2. Sehvermögen

1. Verwenden Sie eine Sehhilfe bei der Arbeit am Computer?

[ ] keine [ ] Brille [ ] Kontaktlinsen [ ] Sonstige

2. Sind Sie farbenblind?

[ ] Nein [ ] Ja, und zwar

3. Ausbildung

1. Abgeschlossene Ausbildung:

[ ] Lehre [ ] Matura [ ] Studium [ ] Doktorat

2. Wenn Sie studieren oder studiert haben, beschreiben Sie bitte Ihr Hauptstudiengebiet:

4. Umgang mit Computern

1. Wie lange benutzen Sie bereits Personal Computer?

Jahre

2. Wieviele Stunden pro Woche verwenden Sie einen Computer?

Stunden

3. Welche Art von Computer verwenden Sie am meisten?

[ ] Microsoft Windows [ ] Apple Macintosh [ ] Unix [ ] Linux [ ] Sonstige

5. Umgang mit Daten

1. Welches Datei-Verwaltungsprogramm verwenden Sie am meisten?

[ ] Windows Explorer [ ] Windows Commander [ ] Kommandozeile, z.B. MS Dos [ ] Andere

2. Wo haben Sie die meisten Daten zu verwalten?

[ ] Zuhause [ ] Büro/Arbeit [ ] Uni [ ] Sonstiges

3. Führen Sie Information Retrieval durch (Suchen von Informationen in sehr großen Datenmengen)?

[ ] täglich [ ] wöchentlich [ ] monatlich [ ] seltener [ ] nie

6. Erfahrung mit Usability Studien

1. Haben Sie schon vorher an einer Usability Studie teilgenommen?

[ ] Ja [ ] Nein

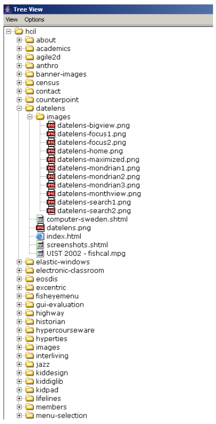
Datum: Thema:

Thinking Aloud Test: Ja [ ] Nein [ ]

B.3 Feedback Formular

Datum Zeit Test Nr. User Nr.

Feedback Formular (Visualisierungen)



Bewerten Sie bitte folgende Aspekte.

Die Visualisierung **TreeView** ist:

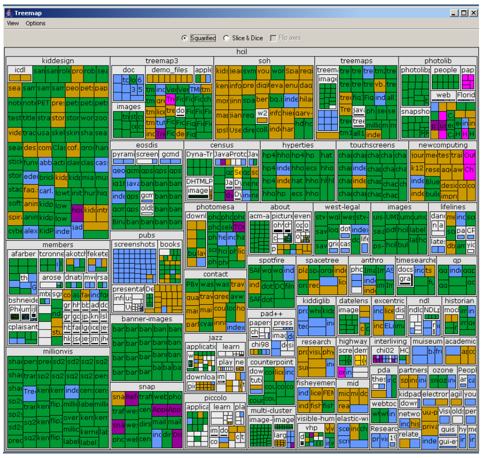
unübersichtlich	3	2	1	0	1	2	3	übersichtlich
schlecht bedienbar	3	2	1	0	1	2	3	gut bedienbar
wenig intuitiv	3	2	1	0	1	2	3	sehr intuitiv
schlecht verwendbar	3	2	1	0	1	2	3	gut verwendbar
schwer verständlich	3	2	1	0	1	2	3	leicht verständlich
unlogisch	3	2	1	0	1	2	3	logisch
nicht zielführend	3	2	1	0	1	2	3	zielführend
Orientierung ist schlecht	3	2	1	0	1	2	3	Orientierung ist gut
Navigation ist schlecht	3	2	1	0	1	2	3	Navigation ist gut

Positiv aufgefallen:

Negativ aufgefallen:

Würden Sie diese Visualisierung verwenden? Warum (nicht)?

Feedback Formular (Visualisierungen)



Bewerten Sie bitte folgende Aspekte.

Die Visualisierung **TreeMap** ist:

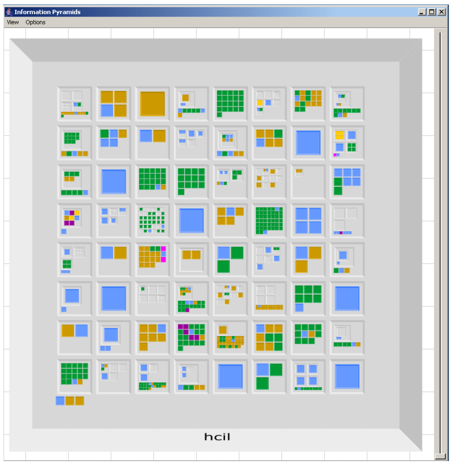
unübersichtlich	3	2	1	0	1	2	3	übersichtlich
schlecht bedienbar	3	2	1	0	1	2	3	gut bedienbar
wenig intuitiv	3	2	1	0	1	2	3	sehr intuitiv
schlecht verwendbar	3	2	1	0	1	2	3	gut verwendbar
schwer verständlich	3	2	1	0	1	2	3	leicht verständlich
unlogisch	3	2	1	0	1	2	3	logisch
nicht zielführend	3	2	1	0	1	2	3	zielführend
Orientierung ist schlecht	3	2	1	0	1	2	3	Orientierung ist gut
Navigation ist schlecht	3	2	1	0	1	2	3	Navigation ist gut

Positiv aufgefallen:

Negativ aufgefallen:

Würden Sie diese Visualisierung verwenden? Warum (nicht)?

Feedback Formular (Visualisierungen)



Bewerten Sie bitte folgende Aspekte.

Die Visualisierung **Pyramids** ist:

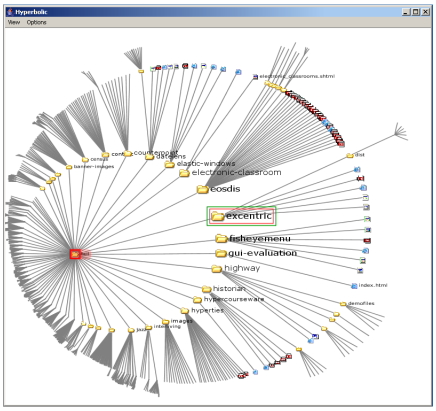
unübersichtlich	3	2	1	0	1	2	3	übersichtlich
schlecht bedienbar	3	2	1	0	1	2	3	gut bedienbar
wenig intuitiv	3	2	1	0	1	2	3	sehr intuitiv
schlecht verwendbar	3	2	1	0	1	2	3	gut verwendbar
schwer verständlich	3	2	1	0	1	2	3	leicht verständlich
unlogisch	3	2	1	0	1	2	3	logisch
nicht zielführend	3	2	1	0	1	2	3	zielführend
Orientierung ist schlecht	3	2	1	0	1	2	3	Orientierung ist gut
Navigation ist schlecht	3	2	1	0	1	2	3	Navigation ist gut

Positiv aufgefallen:

Negativ aufgefallen:

Würden Sie diese Visualisierung verwenden? Warum (nicht)?

Feedback Formular (Visualisierungen)



Bewerten Sie bitte folgende Aspekte.

Die Visualisierung **Hyperbolic** ist:

unübersichtlich	3	2	1	0	1	2	3	übersichtlich
schlecht bedienbar	3	2	1	0	1	2	3	gut bedienbar
wenig intuitiv	3	2	1	0	1	2	3	sehr intuitiv
schlecht verwendbar	3	2	1	0	1	2	3	gut verwendbar
schwer verständlich	3	2	1	0	1	2	3	leicht verständlich
unlogisch	3	2	1	0	1	2	3	logisch
nicht zielführend	3	2	1	0	1	2	3	zielführend
Orientierung ist schlecht	3	2	1	0	1	2	3	Orientierung ist gut
Navigation ist schlecht	3	2	1	0	1	2	3	Navigation ist gut

Positiv aufgefallen:

Negativ aufgefallen:

Würden Sie diese Visualisierung verwenden? Warum (nicht)?

B.4 Task Sets in German

Task Category	Task
Überblick	Finden Sie im Verzeichnis (Ordner) “pad++”(/hcil/pad++) das tiefste Unterverzeichnis. Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Überblick	Finden Sie das Verzeichnis mit den meisten Unterverzeichnissen im Verzeichnis (Ordner) “ndl”(/hcil/ndl). Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend“Weiter”.
Suchen	Finden Sie das Verzeichnis (Ordner) “yidemo” (/hcil/lifelines/yidemo). Wenn Sie das Verzeichnis gefunden haben, schreiben Sie “OK” oder “gefunden” in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Suchen	Finden Sie die Datei /hcil/treemaps/treemap2000/images/ banner-logo-large.gif. Wenn Sie die Datei gefunden haben, schreiben Sie “OK” oder “gefunden” in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Zählen	Zählen Sie die Verzeichnisse (Ordner) nur im Verzeichnis “/hcil/pubs” (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter.”
Zählen	Zählen Sie die Dateien nur im Verzeichnis (Ordner) “/hcil/qp” (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Verzeichnisse: “/hcil/about” oder “/hcil/eosdis”? Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Dateien: “/hcil/spotfire” oder “/hcil/spacetree”? Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.

Table B.1: Task set A in German.

Task Category	Task
Überblick	Finden Sie im Verzeichnis (Ordner) “jazz” (/hcil/jazz) das tiefste Unterverzeichnis . Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Überblick	Finden Sie im Verzeichnis “about” (/hcil/about) das Verzeichnis (Ordner) mit den meisten Unterverzeichnissen . Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Suchen	Finden Sie das Verzeichnis (Ordner) “oldbinary”(/hcil/eosdis/oldbinary). Wenn Sie das Verzeichnis gefunden haben, schreiben Sie “OK” oder “gefunden” in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Suchen	Finden Sie die Datei /hcil/ndl/ndl_secure/draft11/home9.html. Wenn Sie die Datei gefunden haben, schreiben Sie “OK” oder “gefunden” in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Zählen	Zählen Sie die Verzeichnisse (Ordner) nur im Verzeichnis “/hcil/lifelines” (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Zählen	Zählen Sie die Dateien nur im Verzeichnis (ohne Unterverzeichnisse) “/hcil/interliving”. Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Unterverzeichnisse: “/hcil/census” oder “/hcil/treemap3”?. Schreiben sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Dateien: “/hcil/about” oder “/hcil/images”? Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend “Weiter”.

Table B.2: Task set B in German.

Task Category	Task
Überblick	Finden Sie im Verzeichnis "ndl" (/hcil/ndl) das tiefste Unterverzeichnis. Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Überblick	Finden Sie im Verzeichnis "pubs"(/hcil/pubs) das Verzeichnis (Ordner) mit den meisten Unterverzeichnissen . Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Suchen	Finden Sie das Verzeichnis (Ordner) "large-image" (/hcil/multi-cluster/large-image). Wenn Sie das Verzeichnis gefunden haben, schreiben Sie "OK" oder "gefunden" in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Suchen	Finden Sie die Datei /hcil/jazz/applications/cosmosgame/cosmosgame.jpg. Wenn Sie die Datei gefunden haben, schreiben Sie "OK" oder "gefunden" in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Zählen	Zählen Sie die Verzeichnisse nur im Verzeichnis "/hcil/treemaps" (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Zählen	Zählen Sie die Dateien nur im Verzeichnis "/hcil/piccolo" (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Unterverzeichnisse: "/hcil/lifelines" oder "/hcil/pad++"? Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Dateien: "/hcil/census" oder "/hcil/counterpoint"? Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".

Table B.3: Task set C in German.

Task Category	Task
Überblick	Finden Sie im Verzeichnis "treemaps" (/hcil/treemaps) das tiefste Unterverzeichnis. Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Überblick	Finden Sie im Verzeichnis "pad++" (/hcil/pad++) das Verzeichnis mit den meisten Unterverzeichnissen. Schreiben Sie den Namen des Verzeichnisses in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Suchen	Finden Sie das Verzeichnis (Ordner) "ara" (/hcil/People/ara). Wenn Sie das Verzeichnis gefunden haben, schreiben Sie "OK" oder "gefunden" in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Suchen	Finden Sie die Datei /hcil/timesearcher/docs/graphics/averages.gif. Wenn Sie die Datei gefunden haben, schreiben Sie "OK" oder "gefunden" in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Zählen	Zählen Sie die Verzeichnisse nur im Verzeichnis (Ordner) "/hcil/pad++" (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Zählen	Zählen Sie die Dateien im Verzeichnis (Ordner) "/hcil/academics" (ohne Unterverzeichnisse). Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Unterverzeichnisse: "/hcil/highway" oder "/hcil/photolib"?. Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".
Vergleichen	Welches Verzeichnis (Ordner) hat mehr Dateien: "/hcil/members" oder "/hcil/west-legal"? Schreiben Sie die Antwort in das Antwortfeld rechts und drücken Sie anschliessend "Weiter".

Table B.4: Task set D in German.

Appendix C

Analysis in SPSS

This chapter shows how the analysis was conducted using SPSS 12.0 for Windows. It provides a step-by-step tutorial to the functions used for analysing the data obtained in this study.

C.1 Repeated Measures

This study used a repeated measures experimental design to collect performance data and subjective user ratings. SPSS provides functions for analysis of repeated measures. Which function is used depends on the distribution of the obtained data. Based on the distribution (normal or not normal), different tests should be used. Parametric tests can be used for normally distributed data. Non-parametric tests should be used for non-normally distributed data.

Both the task completion times and the subjective user ratings were analysed as follows. The data were scores, with one independent variable (visualisation). In order to analyse repeated measures with four conditions (four different visualisations), ANOVA (for normally distributed data) or Friedman's test (for non-normally distributed data) are used. For the pairwise analysis of differences between the visualisations, a paired samples T-test (normal) or a Wilcoxon test (non-normal) are used.

Data analysed in this thesis was provided in Excel tables. Excel tables can be automatically transformed into SPSS readable tables. Data in an Excel table should be organised in the following manner: first row showing the four variables (visualisations in this case) and the columns showing the data, as can be seen in Figure C.1. When opening Excel tables, SPSS opens the "Opening Excel Data Source" dialogue, see Figure C.2. Read variables from first row of data must be checked. The worksheet to be transformed can be chosen from a drop-down list. After this, the data is transformed into SPSS and can be saved in the SPSS .sav format and analysed.

C.1.1 Histograms

The first step was to look at the data statistics and histograms. A table summarising the data and histograms showing the values are produced with the following steps. The frequencies option can be found under *Analyze - Descriptive Statistics - Frequencies* (see Figure C.3). In the Frequencies dialogue, the variables were transfered from the list on the left to the list on the right. Additionally, the "Display frequency tables" option was deselected (Figure C.4). In the submenu "Statistics…", only Mean and Std. deviation

TV	Pyr	TM	Hyp
68,7	152,6	49,4	83,4
42,2	201,5	122,3	118,2
18,3	93,1	192,1	75,1
89,9	100,5	48,2	269,2
41,2	176,0	74,7	28,4
63,8	38,8	19,7	398,6
56,8	54,8	72,6	40,9
109,6	134,9	67,7	39,4
49,3	33,1	199,1	243,1
36,3	45,7	98,3	133,9
112,5	38,8	124,5	54,5
85,3	56,4	89,0	41,9
78,3	146,1	78,7	87,0
36,3	85,9	33,3	37,3
57,5	55,8	48,2	46,1
213,0	62,6	55,6	108,4
110,2	247,6	94,5	92,4
46,4	189,1	45,8	26,0
53,8	132,5	36,0	29,8
84,5	161,0	141,3	19,5
160,7	125,4	55,8	38,5
89,6	94,8	55,4	37,3
142,1	41,9	476,2	54,7
169,9	50,0	51,6	44,4
81,9	473,9	79,7	34,7
195,3	77,7	54,1	45,7
48,4	153,1	37,8	38,2
633,0	98,3	46,8	72,2
93,8	51,3	63,3	52,7
33,3	53,4	51,3	221,0
25,0	77,5	238,8	43,3
144,0	41,3	68,1	148,2

Figure C.1: Data to be analysed in an Excel table. The first row shows the variables, the columns hold the actual data.

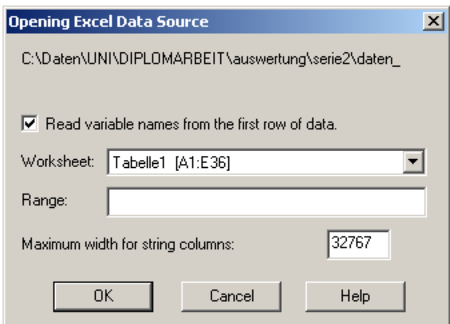


Figure C.2: Dialogue for transforming data in an Excel table into a SPSS table. Read variables form first row is checked.

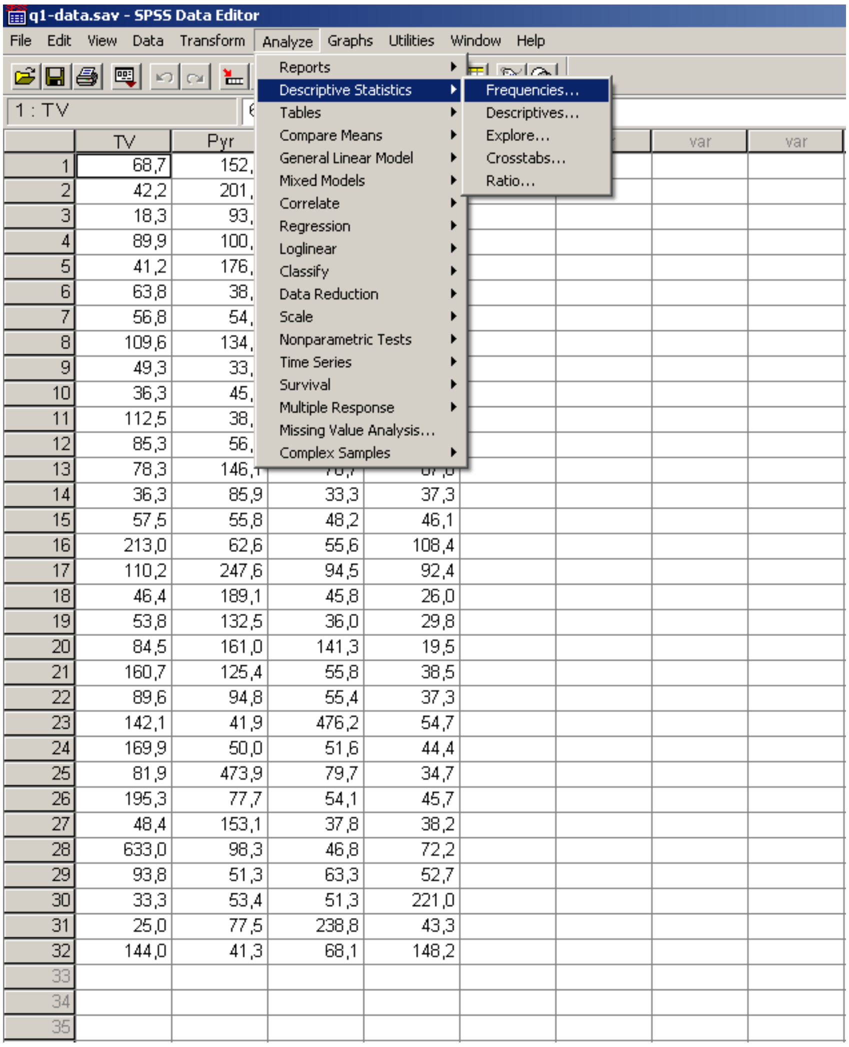


Figure C.3: How to reach the Frequencies option in SPSS 12.0.

were selected (Figure C.5). In the submenu "Charts...", Histograms were chosen. The option "With normal curve" was turned on as well (see Figure C.6). In Figure 9.1, all four resulting histograms have been combined into one figure.

C.1.2 Plot Error Bars

Error bars show the range within which the values lie. In SPSS, error bars can be defined under *Graphs - Error Bar*, as can be seen in Figure C.7. Here, simple bars for summaries of separate variables were chosen (see Figure C.8). After clicking on Define, the variables were transfered from the list on the left to the Error Bar list on the right (Figure C.9). Figure C.10 shows the resulting error bars.

C.1.3 Test for Normality

Whether the distribution is normal or not affects further tests on the data. Here, the Kolmogorov-Smirnov test for normality was performed. In SPSS, it can be found in the

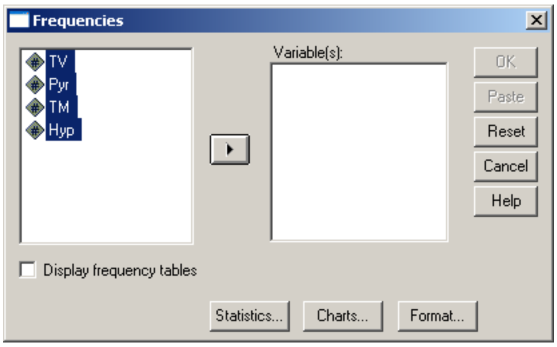


Figure C.4: Transferring the variables in the Frequencies dialogue.

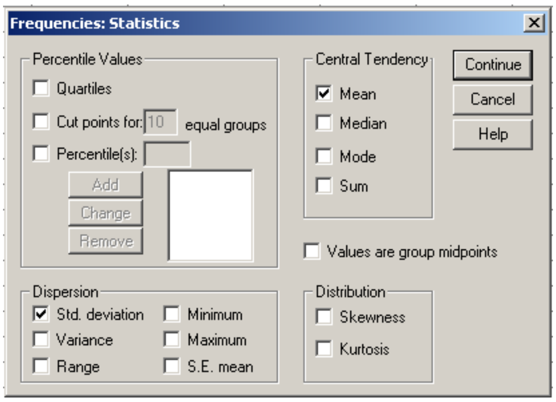


Figure C.5: Choosing statistics to be shown in the Frequencies: Statistics panel.

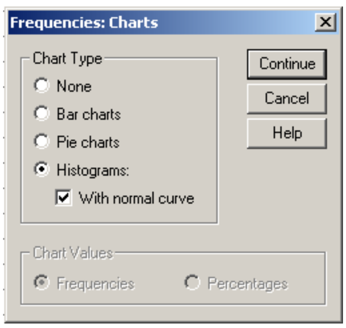


Figure C.6: Choosing the histograms in the Frequencies: Charts panel.

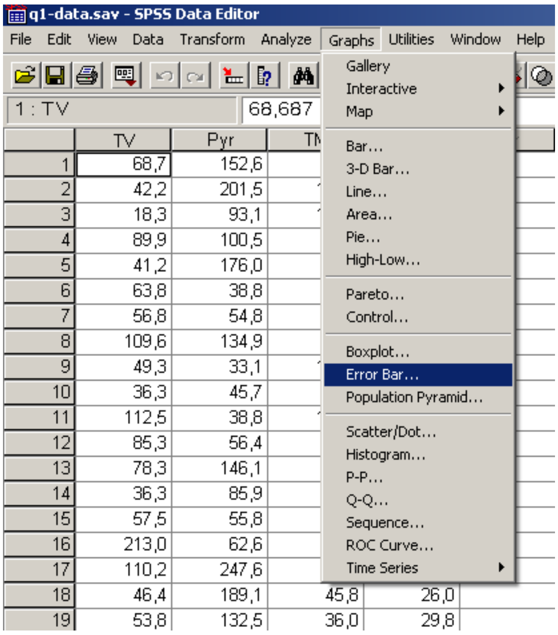


Figure C.7: How to reach the Error Bar dialogue.

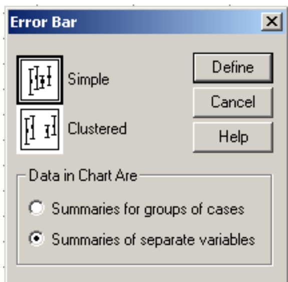


Figure C.8: Choosing the error bars in the Error Bar panel.

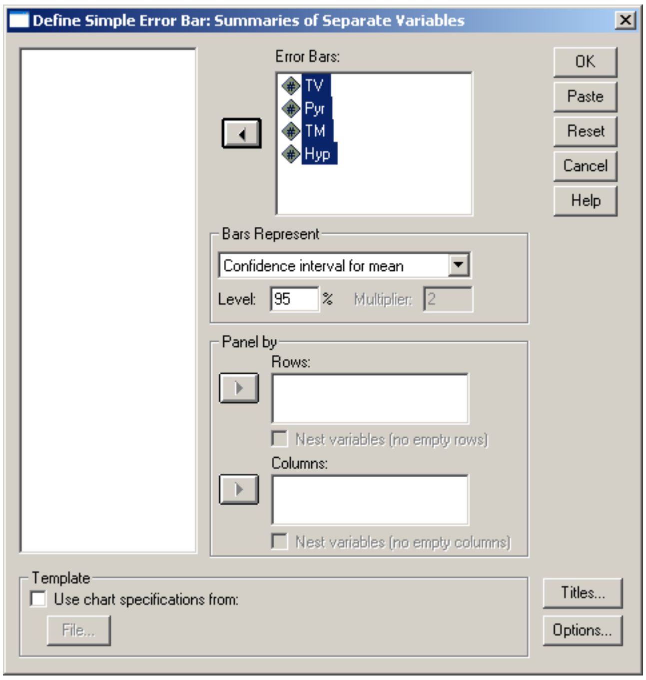


Figure C.9: Transferring the variables for the error bars in the Define Simple Error Bar panel.

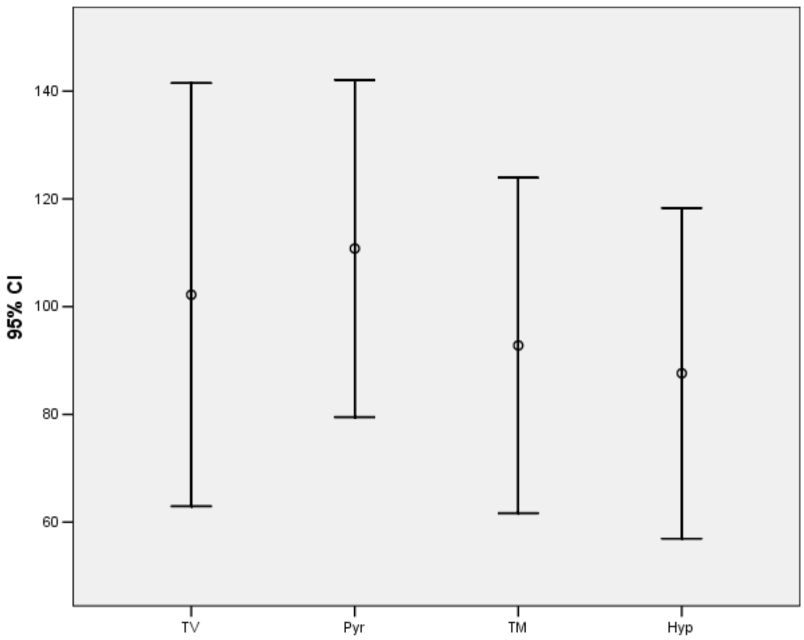


Figure C.10: Resulting error bars for the four selected variables.

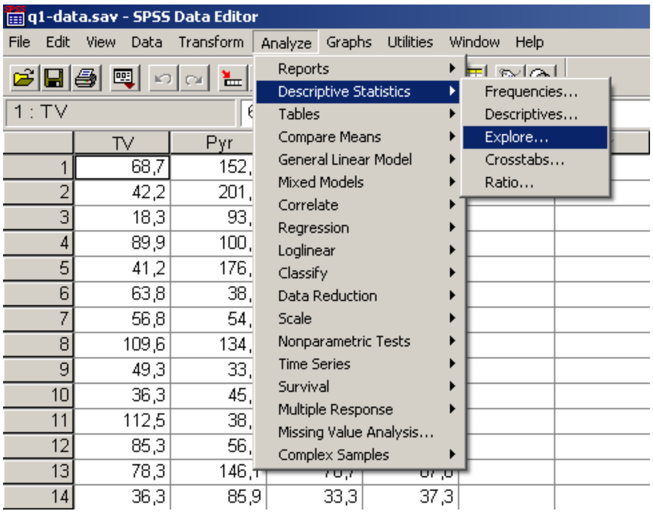


Figure C.11: How to reach the Explore dialogue in SPSS.

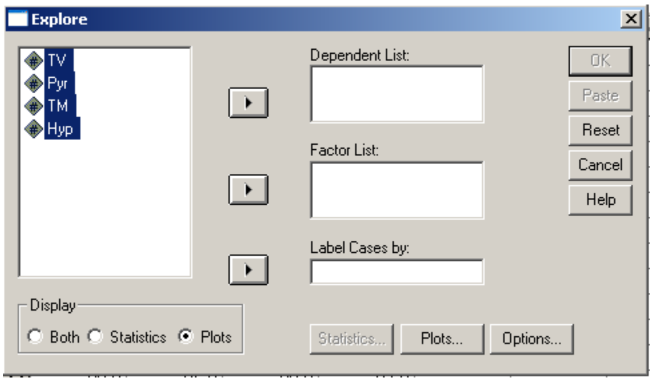


Figure C.12: Transferring the variables to the dependent List in the Explore dialogue.

Explore option: *Analyze - Descriptive Statistics - Explore...* (Figure C.11). Then, all variables were transfered into the Dependent List, see Figure C.12. In the "Plots..." option, the details for the plot were set (Figure C.13).

C.1.4 Friedman's Test

The Friedman test is used for non-normally distributed data. However, it only finds that there are overall significant differences. For details, another test is necessary. In SPSS, Friedman's test is under *Analyze - Nonparametric Tests - K Related Samples...* (Figure C.14). All variables were transferred and Friedman's test chosen (see Figure C.15).

C.1.5 Wilcoxon Test

The Wilcoxon test is used to find significant differences between pairs of variables. In SPSS, the Wilcoxon test is under *Analyze - Nonparametric Tests - 2 Related Samples...* (see

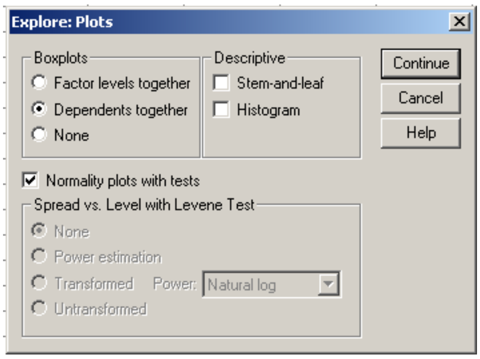


Figure C.13: Setting details for the plots in the Explore: Plots panel.

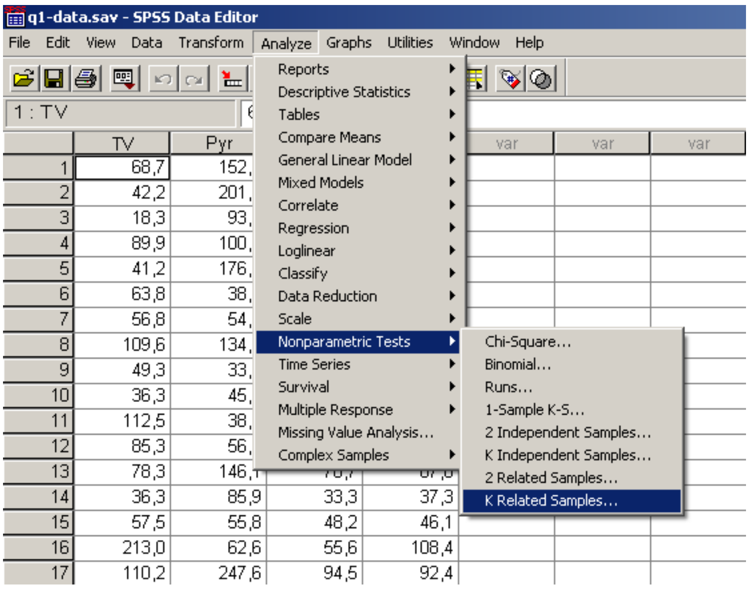


Figure C.14: How to reach Friedman's test.

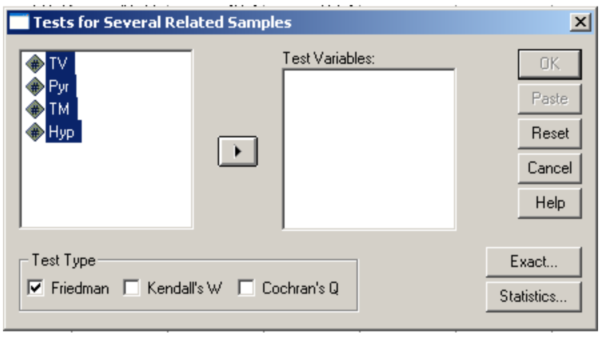


Figure C.15: Transferring the variables for Friedman's test.

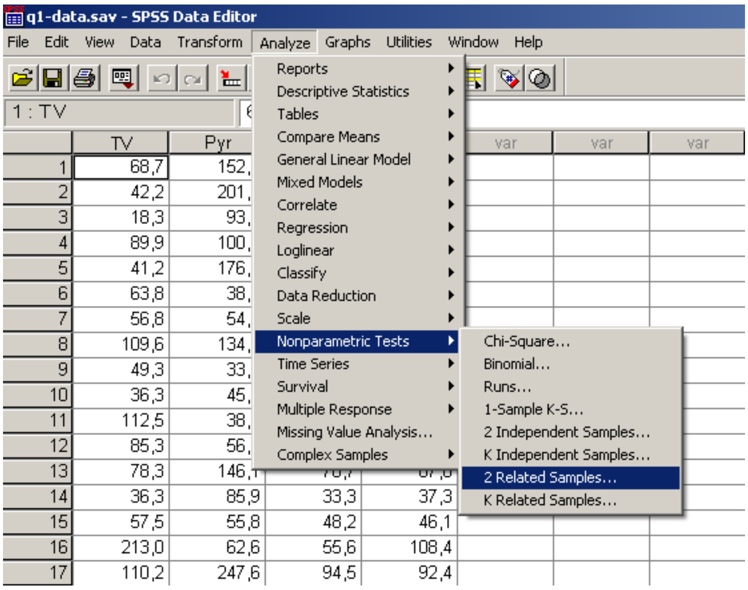


Figure C.16: How to reach the Wilcoxon test in SPSS.

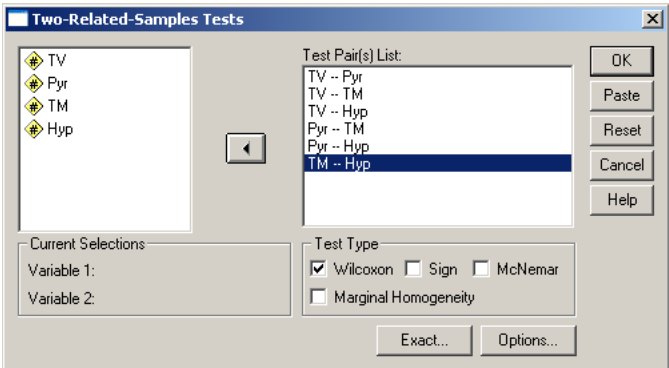


Figure C.17: Transferring the variable pairs for the Wilcoxon test.

Figure C.16). All pairs of variables were transferred to the right-hand list and the Wilcoxon test was chosen (see Figure C.17).

C.1.6 One-Way Repeated Measures ANOVA

One-way repeated measures ANOVA is used if the data is normally distributed. It is important to check the assumption of sphericity, using Mauchly's test. This test is done automatically when performing ANOVA. To perform ANOVA in SPSS, *Analyze - General Linear Model - Repeated Measures...* is selected (see Figure C.18). Then, Within Subject Factor Name (here browser) and Number of Levels (here 4) were added to the list (see Figure C.19). After clicking "Add", the variables are defined, see Figure C.20. After clicking "Define", the variables from the left list were transfered to the Within-Subjects Variables list (Figure C.21). In the "Contrasts..." option, "repeated" from the drop-down list was chosen (Figure C.22). In the "Options..." selection, "browser" was added from the left list in

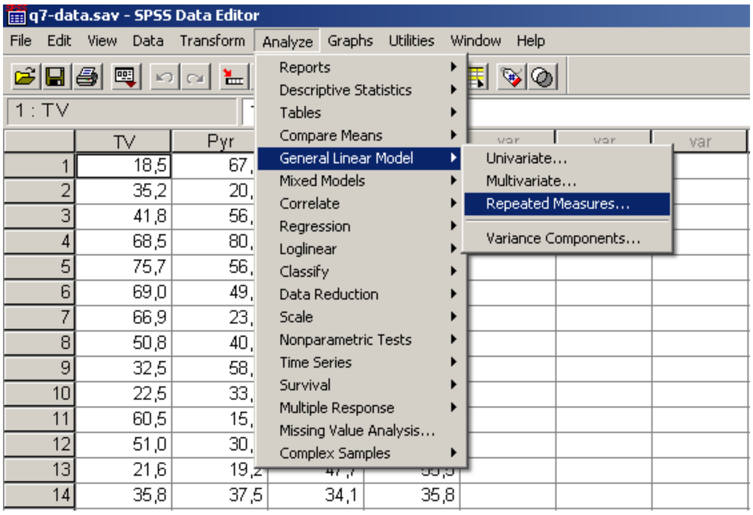


Figure C.18: How to find one-way repeated measures ANOVA in SPSS.

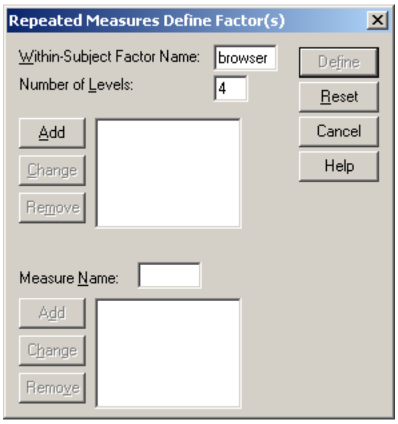


Figure C.19: Adding the variables for ANOVA in SPSS.

the "Display Means for" list and the functions were selected as can be seen in Figure C.23.

C.1.7 Paired T-Test

The paired t-test is analogous to the Wilcoxon test. It is used to find significant differences between all pairs of conditions. In SPSS, it can be found under *Analyze - Compare Means - Paired-Samples T Test...* (see Figure C.24). All pairs have to be added to the right-hand list, as shown Figure C.25.

C.2 Analysis of Preferences - Chi Square

The one-way Chi square test of goodness of fit was used to analyse user overall preferences data. The data were votes for one of the four different visualisations, where each user voted once. The Null Hypothesis was that all visualisations would be equally preferred, thus no

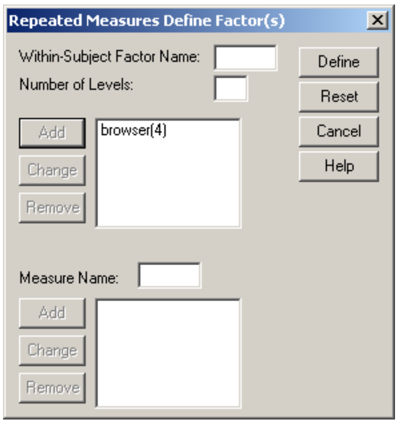


Figure C.20: Added variables for ANOVA in SPSS.

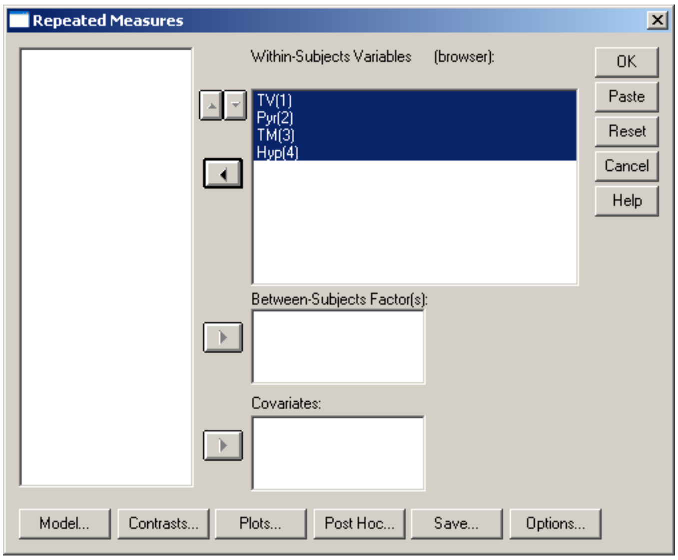


Figure C.21: Defining the variables for Repeated Measures ANOVA.

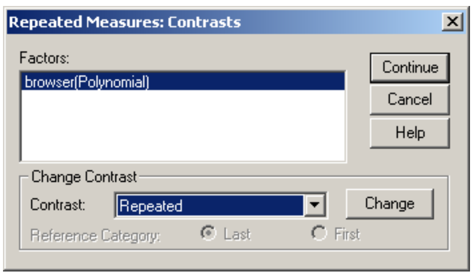


Figure C.22: Defining contrasts for Repeated Measures ANOVA.

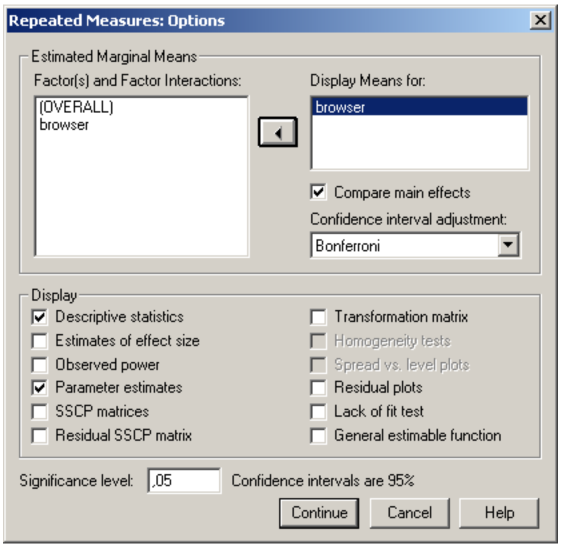


Figure C.23: Defining the options for Repeated Measures ANOVA.

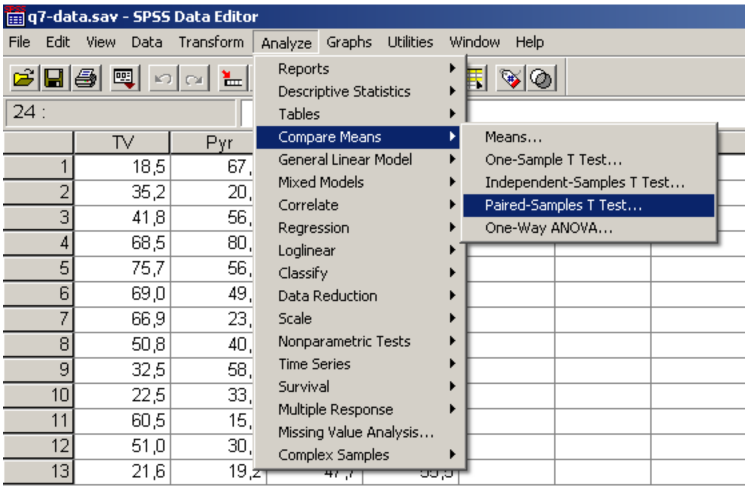


Figure C.24: How to reach the paired samples t-test in SPSS.

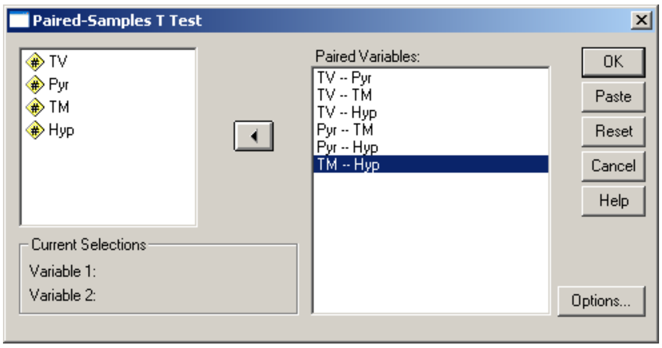


Figure C.25: Transferring each pair of variables for the paired-samples t-test.

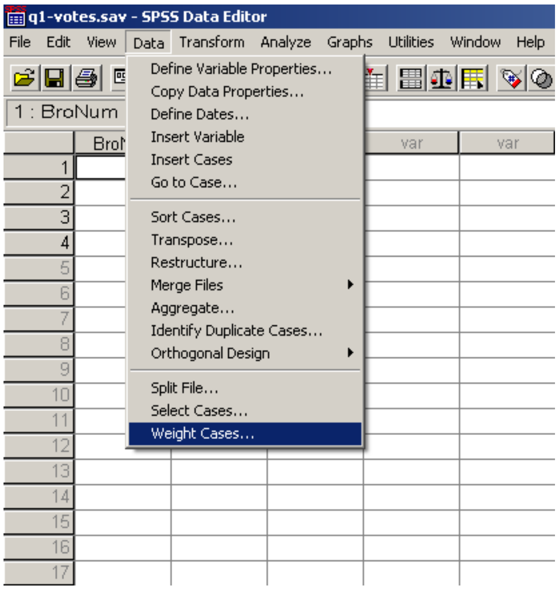


Figure C.26: How to find the Weight Cases dialogue in SPSS.

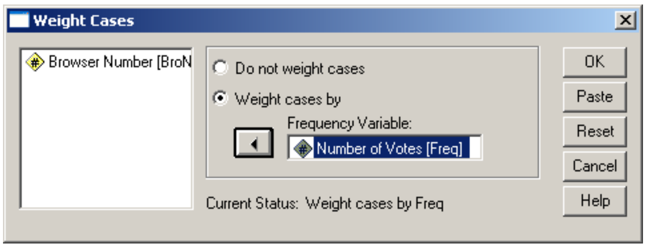


Figure C.27: Choosing the variable to weight cases by.

differences between them. This distribution would yield 8 votes per visualisation for all aspects. The expected frequency per cell should be ≥ 5 for a Chi square test to be valid. This requirement is thus fulfilled here.

In SPSS, the categories need to be coded numerically in order to run a Chi square test. Thus, the visualisations were assigned numbers from one to four along with names.

Before the Chi square test was run, the cases were weighted by frequency. In SPSS, this can be done under *Data - Weight Cases...*, see Figure C.26. Then, the frequency variable (number of votes here) was chosen as can be seen in Figure C.27. Next, a bar chart of the data was produced by going to *Graphs - Interactive - Bar...*(Figure C.28). In the tab "Assign Variables", Browser Name was dragged from the left list to the x-axis variable, as in Figure C.29. In the "Bar Chart Options" tab, Value was chosen as bar label (see Figure C.30). Then, in the "Options" tab, the variable Browser Name was ordered by Occurrence (from the drop-down list), see Figure C.31.

The one-way Chi Square test was run by choosing *Analyze - Nonparametric Tests - Chi Square...*, see Figure C.32. Browser Number was added from the left list to the Test Variable List; the option Get from data was checked for Expected Range and All categories equal for Expected Values, see Figure C.33.

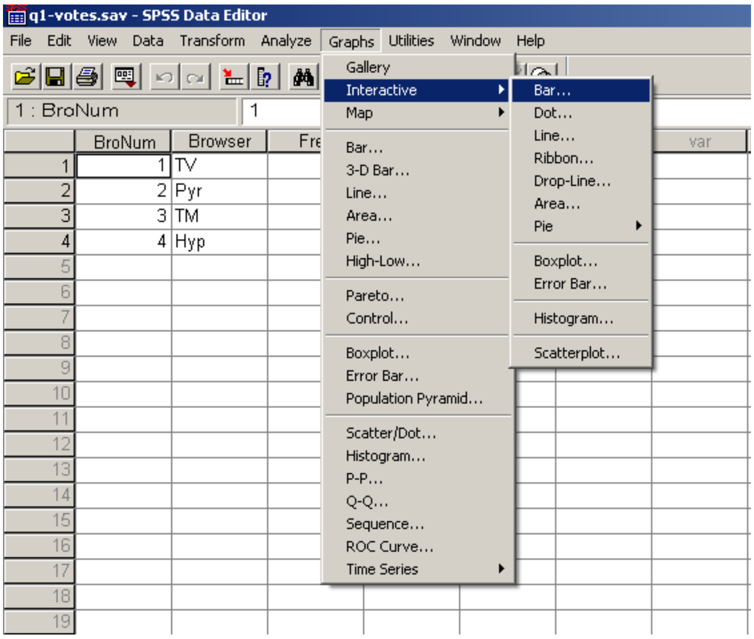


Figure C.28: How to find the Interactive Bar dialogue in SPSS.

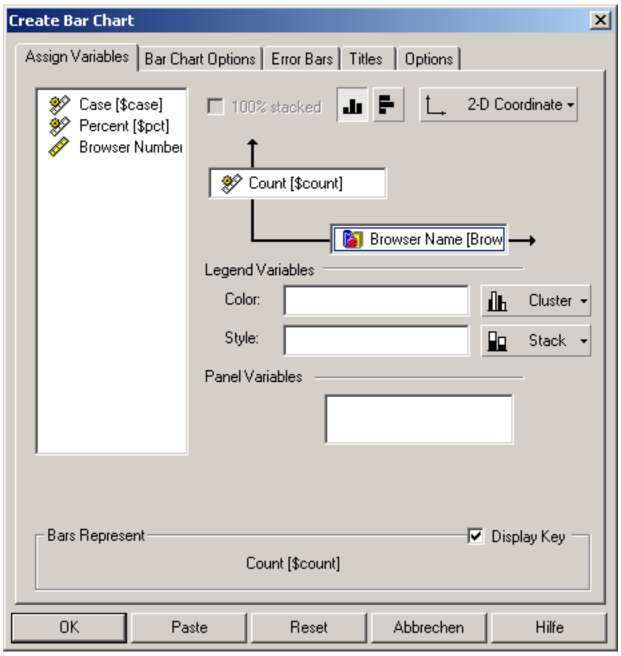


Figure C.29: Assigning variables for the interactive bar chart.

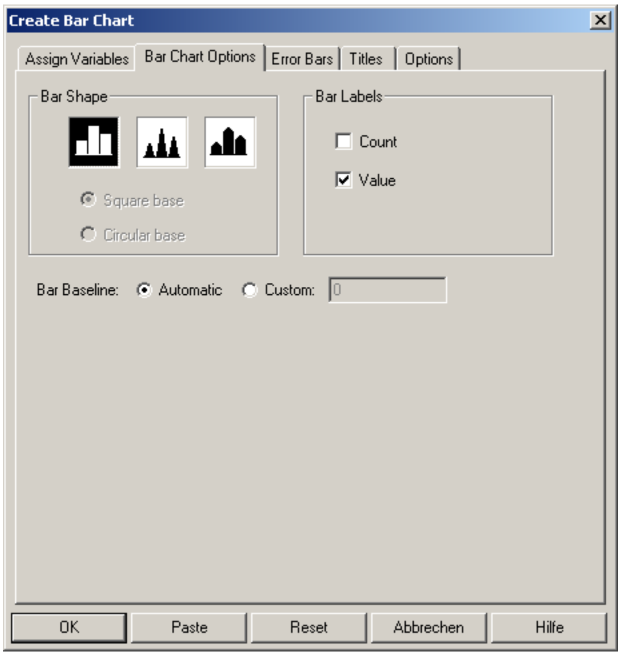


Figure C.30: Choosing the bar chart label.

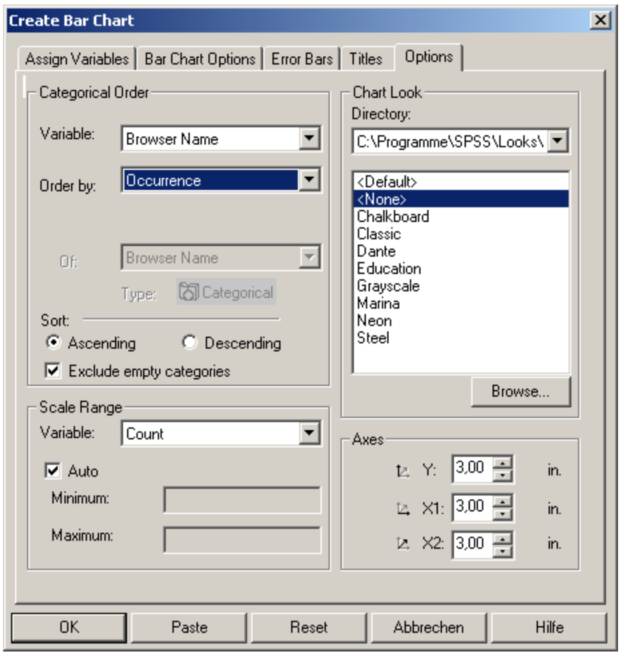


Figure C.31: Choosing the categorical order for the bar chart.

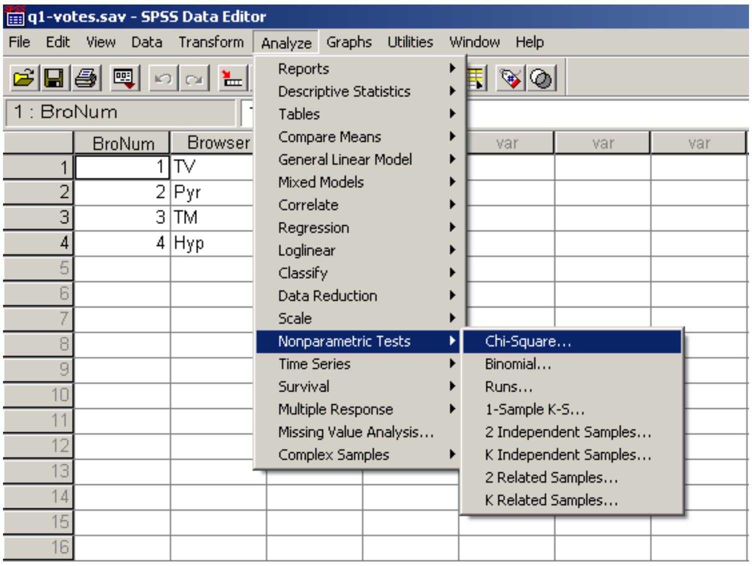


Figure C.32: How to find the Chi Square test in SPSS.

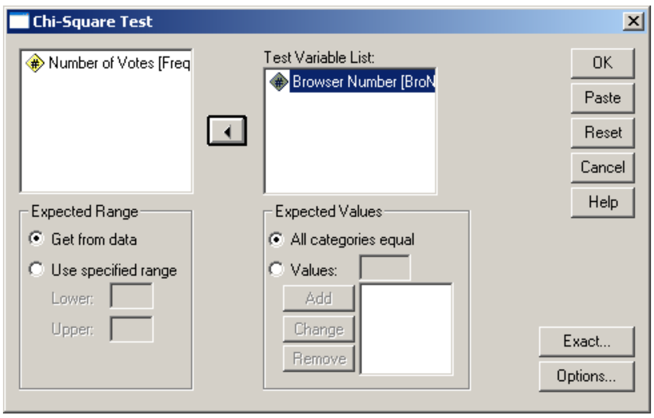


Figure C.33: Transferring the variable for the Chi square test to the Test Variable List.

Bibliography

Christopher Ahlberg and Ben Shneiderman [1994]. *Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays*. In *Proc. CHI'94*, pages 313–317. ACM, Boston, Massachusetts. doi:10.1145/259963.260390. `http://citeseer.ist.psu.edu/ahlberg94visual.html`. (Cited on page 11.)

Christopher Ahlberg, Christopher Williams, and Ben Shneiderman [1992]. *Dynamic Queries for Information Exploration: An Implementation and Evaluation*. In *Proc. CHI'92*, pages 619–626. ACM, Monterey, California. doi:10.1145/142750.143054. (Cited on page 11.)

Keith Andrews [2002]. *Visual Exploration of Large Hierarchies with Information Pyramids*. In *Proc. Sixth International Conference on Information Visualisation (IV'02)*, pages 793–798. IEEE Computer Society Press, London, England. doi:10.1109/IV.2002.1028871. (Cited on page 34.)

Keith Andrews [2006a]. *Human-Computer Interaction: Lecture Notes*. `http://courses.iicm.edu/hci/hci.pdf`. (Cited on page 54.)

Keith Andrews [2006b]. *Information Visualisation: Lecture Notes*. `http://courses.iicm.edu/ivis/ivis.pdf`. (Cited on page 4.)

Keith Andrews [2006c]. *Writing a Thesis: Guidelines for Writing a Master's Thesis in Computer Science*. Graz University of Technology, Austria. `http://ftp.iicm.edu/pub/keith/thesis/thesis.zip`. (Cited on page iii.)

Keith Andrews and Helmut Heidegger [1998]. *Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs*. In *Late Breaking Hot Topic Paper, IEEE Symposium on Information Visualization (InfoVis'98)*, pages 9–12. Research Triangle Park, North Carolina. `http://ftp.iicm.edu/pub/papers/ivis98.pdf`. (Cited on page 34.)

Keith Andrews, Michael Pichler, and Peter Wolf [1996]. *Towards Rich Information Landscapes for Visualising Structured Web Spaces*. In *Proc. 2nd IEEE Symposium on Information Visualization (InfoVis'96)*, pages 62–63. San Francisco, California. doi:10.1109/INFVIS.1996.559218. (Cited on page 27.)

Keith Andrews, Josef Wolte, and Michael Pichler [1997]. *Information Pyramids: A New Approach to Visualising Large Hierarchies*. In *IEEE Visualization'97, Late Breaking Hot Topics Proc.*, pages 49–52. Phoenix, Arizona. `http://ftp.iicm.edu/pub/papers/vis97.pdf`. (Cited on page 34.)

Keith Andrews, Christian Gütl, Josef Moser, Vedran Sabol, and Wilfried Lackner [2001]. *Search Result Visualisation with xFIND*. In *Proc. UIDIS 2001*, pages 50–58. IEEE Computer Society Press, Zurich, Switzerland. doi:10.1109/UIDIS.2001.929925. (Cited on page 17.)

Thomas A. Ball and Stephen G. Eick [1996]. *Software visualization in the Large*. IEEE Computer, 29(4), pages 33–43. doi:10.1109/2.488299. (Cited on page 7.)

Todd Barlow and Padraic Neville [2001]. *A Comparison of 2-D Visualizations of Hierarchies*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2001)*, pages 131–138. San Diego, California, USA. ISBN 1522404X. doi:10.1109/INFVIS.2001.963290. (Cited on pages 32, 33 and 62.)

Luc Beaudoin, Marc-Antoine Parent, and Loius C. Vroomen [1996]. *Cheops: A Compact Explorer for Complex Hierarchies*. In *Proc. Visualization'96*, pages 87–92. IEEE Computer Society, San Francisco, California. doi:10.1109/VISUAL.1996.567745. http://pages.infinit.net/lbeaudoi/cheops.html. (Cited on page 37.)

Luc Beaudoin, Marc-Antoine Parent, and Louis C. Vroomen [2002]. *Cheops: A Compact Explorer For Complex Hierarchies*. http://www.maparent.ca/paper.html. (Cited on page 38.)

Thomas Burger [1999]. *Magic Eye View: Eine neue Fokus + Kontext Technik zur Darstellung von Graphen*. In German, University of Rostock, Institute of Computer Graphics. http://wwwicg.informatik.uni-rostock.de/Diplomarbeiten/1999/Thomas_Buerger/. (Cited on page 27.)

Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman (Editors) [1999]. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann. ISBN 1558605339. (Cited on page 3.)

Matthew Chalmers and Paul Chitson [1992]. *Bead: Explorations in Information Visualization*. In *Proc. SIGIR '92*, pages 330–337. ACM, Copenhagen, Denmark. doi:10.1145/133160.133215. http://www.dcs.gla.ac.uk/~matthew/papers/sigir92.pdf. (Cited on pages 16 and 17.)

Chaomei Chen and Yue Yu [2000]. *Empirical Studies of Information Visualization: A Meta-Analysis*. International Journal of Human-Computer Studies, 53(5), pages 851–866. doi:10.1109/6.469330. (Cited on page 59.)

Mary Czerwinski and Kevin Larson [1997]. *The New Web Browsers: They're Cool but Are They Useful?* Presentation at HCI'97. http://www.research.microsoft.com/users/marycz/bhci97.ppt. (Cited on pages 53 and 60.)

Phil Dykstra [1991]. *XDU*. http://sd.wareonearth.com/~phil/xdu/. (Cited on pages 32 and 34.)

Stephen G. Eick, Joseph L. Steffen, and Eric E. Sumner Jr. [1992]. *SeeSoft – A Tool for Visualizing Line Oriented Software Statistics*. IEEE Transactions on Software Engineering, 18(11), pages 957–968. doi:10.1109/32.177365. (Cited on page 5.)

Kim Michael Fairchild, Steven E. Poltrock, and George W. Furnas [1988]. *SemNet: Three-Dimensional Representations of Large Knowledge Bases*. In Raymonde Guindon (Editor), *Cognitive Science and its Applications for Human-Computer Interaction*, pages 201–233. Lawrence Erlbaum, Hillsdale, New Jersey. ISBN 0898598842. (Cited on page 8.)

Jean-Daniel Fekete and Catherine Plaisant [2003a]. *Information Visualization Benchmarks Repository 2003*. http://www.cs.umd.edu/hcil/InfovisRepository/contest-2003/. (Cited on page 82.)

Jean-Daniel Fekete and Catherine Plaisant [2003b]. *InfoVis 2003 Contest*. http://www.cs.umd.edu/hcil/iv03contest/. (Cited on page 41.)

Jean-Daniel Fekete and Catherine Plaisant [2006]. *Information Visualization Benchmarks Repository*. http://www.cs.umd.edu/hcil/InfovisRepository/index.shtml. (Cited on page 59.)

Scott Fertig, Eric Freeman, and David Gelernter [1996]. *Lifestreams: An Alternative to the Desktop Metaphor*. In *CHI'96 Video Program*. ACM. http://acm.org/sigchi/chi96/proceedings/videos/Fertig/etf.htm. (Cited on page 7.)

Andy Field and Graham Hole [2003]. *How to Design and Report Experiments*. Sage. ISBN 0761973834. (Cited on page 96.)

Eric Freeman and Scott Fertig [1995]. *Lifestreams: Organizing Your Electronic Life*. In *AAAI Fall Symposium 1995*. AAAI. http://www.cs.yale.edu/homes/freeman/papers/AAAI/draft.ps. (Cited on page 6.)

Amit Goel [1999]. *Parallel Coordinates Visualization Applet*. Virginia Tech. http://csgrad.cs.vt.edu/~agoel/parallel_coordinates/. (Cited on page 12.)

Jesse Grosjean, Catherine Plaisant, and Ben Bederson [2002]. *SpaceTree: A Novel Node-Link Tree Browser*. http://www.cs.umd.edu/hcil/spacetree/. (Cited on page 30.)

HCIL [2002]. *University of Maryland Screeshots*. http://www.cs.umd.edu/hcil/pubs/screenshots/. (Cited on pages 13, 23, 25 and 31.)

Marti Hearst [2005]. *SIMS 247: Information Visualization and Presentation*. School of Information Management and Systems, University of California. http://www2.sims.berkeley.edu/courses/is247/f05/lectures/lecture1.ppt. (Cited on page 3.)

Marti A. Hearst [1995]. *TileBars: Visualization of Term Distribution Information in Full Text Information Access*. In *Proc. CHI'95*. ACM, Denver, Colorado. doi:10.1145/223904.223912. http://www.sims.berkeley.edu/~hearst/papers/tilebars-chi95/chi95.html. (Cited on pages 19 and 20.)

Matthias Hemmje, Clemens Kunkel, and Alexander Willet [1994]. *LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval*. In *Proc. SIGIR'94*. ACM, Dublin, Ireland. http://portal.acm.org/citation.cfm?id=188563. (Cited on pages 19 and 20.)

Bob J. Hendley, Nick S. Drew, Andrew M. Wood, and Russell Beale [1995]. *Narcissus: Visualising Information*. In *Proc. IEEE Symposium on Information Visualization (InfoVis'95)*, pages 90–96. IEEE Computer Society, Atlanta, Georgia. doi:10.1109/INFVIS.1995.528691. (Cited on page 8.)

Beth Hetzler, W. Michelle Harris, Susan Havre, and Paul Whitney [1998]. *Visualizing the Full Spectrum of Document Relationships*. In *Proc. Fifth International ISKO Conference*, pages 168–175. ERGON Verlag, Lille, France. http://www.pnl.gov/infoviz/isko.pdf. (Cited on page 17.)

Tom Horton [2006]. *CS305: HCI in Software Development*. Department of Computer Science, University of Virginia. http://www.cs.virginia.edu/~horton/cs305/. (Cited on page 53.)

Timothy Hughes, Young Hyun, and David A. Liberles [2004]. *Visualising Very Large Phylogenetic Trees in Three Dimensional Hyperbolic Space*. BMC Bioinformatics Journal, 5(48). doi:10.1186/1471-2105-5-48. http://www.caida.org/outreach/papers/2004/bioinformatics/. (Cited on page 29.)

Young Hyun [2005]. *Walrus*. http://www.caida.org/tools/visualization/walrus/. (Cited on page 30.)

Alfred Inselberg and Bernard Dimsdale [1990]. *Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry*. In *Proc. IEEE Visualization'90*, pages 361–378. IEEE Computer Society Press, Los Angeles, USA. ISBN 0818620838. doi:10.1109/VISUAL.1990.146402. (Cited on page 11.)

Inxight [2006a]. *Inxight VizServer StarTree*. http://www.inxight.com/products/sdks/st/. (Cited on page 27.)

Inxight [2006b]. *Inxight VizServer TableLens*. http://www.inxight.com/products/sdks/tl/. (Cited on pages 12 and 13.)

Inxight [2006c]. *Inxight VizServer TimeWall*. http://www.inxight.com/products/vizserver/tw/. (Cited on pages 5 and 6.)

ISO [1998]. *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*. (Cited on page 53.)

T. J. Jankun-Kelly [2005]. *CSE8990-03: Information Visualization*. Department of Computer Science and Engineering, Mississippi State University. (Not cited.)

Dean F. Jerding and John T. Stasko [1998]. *The Information Mural: A Technique for Displaying and Navigating Large Information Spaces*. IEEE Transactions on Visualization and Computer Graphics, 4(3), pages 257–271. doi:10.1109/2945.722299. http://www.cc.gatech.edu/gvu/softviz/infoviz/information_mural.html. (Cited on page 4.)

Brian Johnson and Ben Shneiderman [1991]. *Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures*. In *Proc. IEEE Visualization '91*, pages 284–291. IEEE Computer Society, San Diego, California. doi:10.1109/VISUAL.1991.175815. http://www.cs.umd.edu/hcil/treemaps/. (Cited on page 31.)

Ernst Kleiberg, Huub van de Wetering, and Jack van Wijk [2001]. *Botanical Visualization of Huge Hierarchies*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2001)*, pages 87–94. IEEE Computer Society, San Diego, California. doi:10.1109/INFVIS.2001.963285. http://www.win.tue.nl/~vanwijk/botatree.pdf. (Cited on pages 38 and 39.)

Alfred Kobsa [2004]. *User Experiments with Tree Visualization Systems*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2004)*, pages 9–16. Austin, Texas, USA. doi:10.1109/INFVIS.2004.70. (Cited on page 63.)

Matthias Kreuseler, Norma Lopez, and Heidrun Schuhmann [2000]. *A Scalable Framework for Information Visualization*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2000)*, pages 27–36. IEEE Computer Society, Salt Lake City, Utah. doi:10.1109/INFVIS.2000.885088. http://www.informatik.uni-rostock.de/~mkreusel/SInVis/infovis.html. (Cited on page 29.)

J. B. Kruskal and J. M. Landwehr [1983]. *Icicle Plots: Better Displays for Hierarchical Clustering*. American Statistician, 37(2), pages 162–168. (Cited on page 32.)

John Lamping, Ramana Rao, and Peter Pirolli [1995]. *A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies*. In *Proc. CHI'95*, pages 401–408. ACM, Denver, Colorado. doi:10.1145/223904.223956. http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/jl_bdy.htm. (Cited on pages 26 and 60.)

Leo [2004]. *Leo English-German Dictionary*. http://dict.leo.org/. (Not cited.)

Scott MacKenzie [2006]. *Human-Computer Interaction*. York University. http://www.cs.yorku.ca/course_archive/2005-06/W/4441/. (Cited on pages 53 and 56.)

Jock D. Mackinlay, George G. Robertson, and Stuart K. Card [1991]. *The Perspective Wall: Detail and Context Smoothly Integrated*. In *Proc. CHI'91*, pages 173–179. ACM, New Orleans, Louisiana. doi:10.1145/108844.108870. (Cited on page 5.)

Mark P. McCahill and Thomas Erickson [1995]. *Design for a 3D Spatial User Interface for Internet Gopher*. In *Proc. of ED-MEDIA 95*, pages 39–44. AACE, Graz, Austria. http://www.visi.com/~snowfall/GopherVR.html. (Cited on page 37.)

Tamara Munzer [2005]. *CPSC 533C: Information Visualisation*. Department of Computer Science, University of British Columbia. http://www.cs.ubc.ca/~tmm/courses/cpsc533c-05-fall/. (Not cited.)

Tamara Munzner [1997]. *H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space*. In *Proc. IEEE Symposium on Information Visualization (InfoVis'97)*, pages 2–10. IEEE Computer Society, Phoenix, Arizona. doi:10.1109/INFVIS.1997.636718. http://graphics.stanford.edu/papers/h3/. (Cited on page 26.)

Tamara Munzner [2002]. *Software*. http://www.cs.ubc.ca/~tmm/. (Cited on page 28.)

David A. Nation, Catherine Plaisant, Gary Marchionini, and Anita Komlodi [1997]. *Visualizing Websites Using a Hierarchical Table of Contents Browser: WebTOC*. In *Proc. 3rd Conference on Human Factors and the Web*. US WEST, Denver, Colorado. ftp://ftp.cs.umd.edu/pub/hcil/Demos/WebTOC/Paper/WebTOC.html. (Cited on page 21.)

Jakob Nielsen, Donald Norman, and Bruce Tognazzini [2006]. *Nielsen Norman Group*. http://www.nngroup.com/. (Cited on page 54.)

Chris North [2005]. *cs5764: Information Visualization*. Department of Computer Science, Virginia Tech. http://infovis.cs.vt.edu/cs5764/lectures/. (Not cited.)

Lucy Terry Nowell, Robert K. France, Deborah Hix, Lenwood S. Heath, and Edward A. Fox [1996]. *Visualizing Search Results: Some Alternatives to Query-Document Similarity*. In *Proc. SIGIR'96*, pages 67–75. ACM, Zurich, Switzerland. doi:10.1145/243199.243214. http://www.dlib.vt.edu/Papers/SIGIR96/SIGIR96.Env.html. (Cited on pages 14 and 15.)

Alexander Nussbaumer [2005]. *Hierarchy Browsers: Integrating Four Graph-Based Hierarchy Browsers into the Hierarchical Visualisation System (HVS)*. Master's Thesis, Graz University of Technology, Austria. ftp://ftp.iicm.edu/pub/theses/anuss.pdf. (Cited on page 50.)

Catherine Plaisant [2004]. *The Challenge of Information Visualization Evaluation*. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2004)*, pages 109–116. Gallipoli, Italy. doi:10.1145/989863.989880. (Cited on page 59.)

Catherine Plaisant, Jesse Grosjean, and Benjamin B. Bederson [2002]. *SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2002)*, pages 57–66. IEEE Computer Society, Boston, Massachusetts, USA. doi:10.1109/INFVIS.2002.1173148. (Cited on pages 29 and 63.)

Jennifer Preece, Yvonne Rogers, and Helen Sharp [2002]. *Interaction Design. Beyond Human-Computer Interaction*. Wiley. ISBN 0471492787. (Cited on page 53.)

Werner Putz [2005]. *The Hierarchical Visualization System: A General Framework for Visualizing Information Hierarchies Using the Example of Information Pyramids*. Master's Thesis, Graz University of Technology, Austria. ftp://ftp.iicm.edu/pub/theses/wputz.pdf. (Cited on page 41.)

Ramana Rao and Stuart K. Card [1994]. *The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information*. In *Proc. CHI'94*, pages 318–322. ACM, Boston, Massachusetts. doi:10.1145/191666.191776. (Cited on page 12.)

Kirsten Risden, Mary P. Czerwinski, Tamara Munzer, and Daniel B. Cook [2000]. *An Initial Examination of Ease of Use for 2D and 3D Information Visualizations of Web Content*. International Journal of Human-Computer Studies: Special Issue on Empirical Evaluations of Information Visualizations, 53(5), pages 695–714. doi:10.1006/ijhc.2000.0413. (Cited on page 62.)

George G. Robertson, Jock D. Mackinlay, and Stuart K. Card [1991]. *Cone Trees: Animated 3D Visualizations of Hierarchical Information*. In *Proc. CHI'91*, pages 189–194. ACM, New Orleans, Louisiana. doi:10.1145/108844.108883. (Cited on page 24.)

Vedran Sabol [2001]. *Visualisation Islands: Interactive Visualisation and Clustering of Search Result Sets*. Master's Thesis, Graz University of Technology, Austria. ftp://ftp.iicm.edu/pub/theses/vsabol.pdf. (Cited on page 14.)

Jeff Sauro [2006]. *Confidence Interval Around a Completion Rate*. http://www.measuringusability.com/wald.htm. (Cited on page 112.)

Jeff Sauro and James R. Lewis [2005]. *Estimating Completion Rates from Small Samples Using Binomial Confidence Intervals: Comparisons and Recommendations*. In *Proc. Human Factors and Ergonomics Society 49th annual meeting*. http://www.measuringusability.com/sauro-lewisHFES.pdf. (Cited on page 112.)

Jürgen Schlipflinger [1998]. *The Design and Implementation of the Harmony Session Manager*. Master's thesis, University of Technology Graz, IICM. http://www.iicm.edu/thesis/jschipf.pdf. (Cited on page 9.)

School of Information Management and Systems [2003]. *How Much Information? 2003*. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/printable_report.pdf. (Cited on page 3.)

Jinwook Seo and Ben Shneiderman [2002]. *Understanding Hierarchical Clustering Results by Interactive Exploration of Dendrograms: A Case Study with Genomic Microarray Data*. Technical report HCIL-2002-10, HCIL University of Maryland. http://hcil.cs.umd.edu/trs/2002-10/2002-10.pdf. (Cited on page 23.)

SequoiaView [2005]. *SequoiaView*. http://www.win.tue.nl/sequoiaview/. (Cited on pages 32 and 33.)

Ben Shneiderman [1996]. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. In *Proc. 1996 IEEE Symposium on Visual Languages (VL'96)*, pages 336–343. IEEE Computer Society, Boulder, Colorado. doi:10.1109/VL.1996.545307. http://citeseer.ist.psu.edu/shneiderman96eyes.html. (Cited on page 3.)

SmartMoney [2002]. *Map of the Market*. http://www.smartmoney.com/marketmap/. (Cited on page 32.)

Andy Smith and Dave Clark [2001]. *Building a Customized Tree View*. IBM developerWorks Tutorial. http://www.ibm.com/developerworks/training/. (Cited on page 21.)

Anselm Spoerri [1993a]. *InfoCrystal: A Visual Tool for Information Retrieval*. In *Proc. Visualization'93*, pages 150–157. IEEE Computer Society, San Jose, California. doi:10.1109/VISUAL.1993.398863. (Cited on page 18.)

Anselm Spoerri [1993b]. *InfoCrystal: A Visual Tool for Information Retrieval Management*. In *Proc. CIKM'93*, pages 11–20. ACM, Washington, D.C. (Cited on page 18.)

Anselm Spoerri [2001]. *InfoCrystal: A Visual Tool for Information Retrieval*. http://www.scils.rutgers.edu/~aspoerri/InfoCrystal/InfoCrystalSlides_files/frame.htm. (Cited on page 19.)

John Stasko [2006]. *CS7450 Class*. College of Computing, Georgia Tech. http://www-static.cc.gatech.edu/classes/AY2006/cs7450_spring/syllabus.html. (Cited on page 4.)

John T. Stasko and Eugene Zhang [2000]. *Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2000)*, pages 57–65. IEEE Computer Society, Salt Lake City, Utah. doi:10.1109/INFVIS.2000.885091. http://www.cc.gatech.edu/gvu/ii/sunburst/. (Cited on page 35.)

John T. Stasko, Richard Catrambone, Mark Guzdial, and Kevin McDonald [2000]. *An evaluation of space-filling information visualizations for depicting hierarchical structures*. International Journal of Human-Computer Studies, 53(5), pages 663–694. doi:10.1006/ijhc.2000.0420. http://www.cc.gatech.edu/gvu/ii/sunburst/. (Cited on pages 36 and 61.)

Joel D. Tesler and Steven L. Strasnick [1992]. *FSN: The 3D File System Navigator*. Silicon Graphics, Inc. ftp://ftp.sgi.com/sgi/fsn. (Cited on page 24.)

TheBrain.Com [2002]. *TheBrain*. http://www.thebrain.com. (Cited on page 11.)

Lisa Tweedie, Bob Spence, David Williams, and Ravinder Bhogal [1994]. *The Attribute Explorer*. In *CHI'94 Video Program*. ACM. doi:10.1145/259963.260433. (Cited on pages 12 and 14.)

UsabilityHome [2005]. *Usability Evaluation Methods*. Human Factors group, US WEST. http://www.usabilityhome.com/. (Cited on page 54.)

Frank van Ham and Jarke J. van Wijk [2002]. *Beamtrees: Compact Visualization of Large Hierarchies*. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2002)*, pages 93–100. IEEE Computer Society, Boston, Massachusetts. doi:10.1109/INFVIS.2002.1173153. http://www.win.tue.nl/~fvham/beamtrees/. (Cited on pages 38 and 39.)

Jarke J. van Wijk and Huub van de Wetering [1999]. *Cushion Treemaps: Visualization of Hierarchical Information*. In *Proc. IEEE Symposium on Information Visualization (InfoVis'99)*, pages 73–78. IEEE Computer Society, San Francisco, California. doi:10.1109/INFVIS.1999.801860. http://www.win.tue.nl/~vanwijk/ctm.pdf. (Cited on page 32.)

John Q. Walker, II [1990]. *A Node-Positioning Algorithm for General Trees*. Software: Practice and Experience, 20(7), pages 685–705. (Cited on page 23.)

Martin Wattenberg [1999]. *Visualizing the Stock Market*. In *CHI 99 Extended Abstracts*, pages 188–189. ACM, Pittsburgh, Pennsylvania. doi:10.1145/632716.632834. http://www.smartmoney.com/marketmap/. (Cited on page 31.)

Korin Werner [2002]. *A Comparison of Information Visualization Methods*. In *Student Poster, CHI 2002 Extended Abstracts*, pages 872–873. Minneapolis, Minnesota, USA. ISBN 1581134541. doi:10.1145/506443.506641. (Cited on page 63.)

Ulrika Wiss and David A. Carr [1999]. *An Empirical Study of Task Support in 3D Information Visualizations*. In *Proc. Third International Conference on Information Visualisation (IV'99)*, pages 392–399. IEEE Computer Society Press, London, England. doi:10.1109/IV.1999.781587. (Cited on pages 60 and 87.)

Peter Wolf [1996]. *Three-Dimensional Information Visualisation: The Harmony Information Landscape*. Master's Thesis, Graz University of Technology, Austria. http://ftp.iicm.edu/pub/theses/pwolf.pdf. (Cited on page 9.)

Andrew M. Wood, Nick S. Drew, Russell Beale, and Bob J. Hendley [1995]. *HyperSpace: Web Browsing with Visualisation*. In *Proc. WWW3*, pages 21–25. Darmstadt, Germany. http://www.igd.fhg.de/www/www95/proceedings/posters/35/index.html. (Cited on pages 8 and 9.)

Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko [2005a]. *Dust and Magnet: Multivariate Information Visualization Using a Magnet Metaphor*. Information Visualization, 4(4), pages 239–256. doi:10.1057/palgrave.ivs.9500099. (Cited on page 14.)

Ji Soon Yi, Rachel Melton, John Stasko, and Julie A. Jacko [2005b]. *Dust and Magnet: Overview of Project Video*. ftp://ftp.cc.gatech.edu/pub/people/stasko/movies/dnm.mov. (Cited on page 16.)