HVS: A Framework for Visualising Hierarchies

Keith Andrews[*] Werner Putz[†]

IICM, Graz University of Technology

Abstract

Numerous techniques have been developed for visualising hierarchically structured information. This demo presents a new framework for the visualisation of hierarchies called the Hierarchical Visualisation System (HVS). HVS is a general framework implemented in Java. It provides a synchronised, multiple view environment for visualising, exploring, and managing large hierarchies.

CR Categories: H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces;

Keywords: information visualisation, hierarchies, trees, framework, Java, coordinated views, empirical test environment

1 Introduction

Information is often organised into hierarchies of one kind or another. Various techniques have been developed to improve access to hierarchically structured information, from the humble tree view (such as the Macintosh File Finder or Windows Explorer) to realistic 3d botanical renderings of trees.

2 The Hierarchical Visualisation System (HVS)

The Hierarchical Visualisation System (HVS) is a general framework implemented in Java, which provides a synchronised, multiple view environment for visualising, exploring, and managing large hierarchies. Figure 1 shows an application of HVS, with five synchronised views (browsers) of a hierarchy. A more typical application might utilise two or three synchronised views.

The multiple view environment in HVS is based on the model-view-controller (MVC) design pattern [5]. HVS currently reads hierarchies either from the file system or from TreeML files. The controller manages synchronisation of the various views and provides centralised sorting, searching, and filtering functionality to all of its browsers.

When reading from the file system, HVS attempts to collect or extract as much metadata about objects in the hierarchy as possible. The metadata are extracted in a background thread and are cached in an XML format between HVS sessions. The various metadata fields are then available to all of the browsers within HVS.

Since HVS provides much of the necessary infrastucture, developing a new hierarchy visualisation is much less work than starting from scratch. In addition to being a tool to explore and manage hierarchies, HVS will also serve as a platform for the empirical evaluation and comparison of different hierarchy browsers.

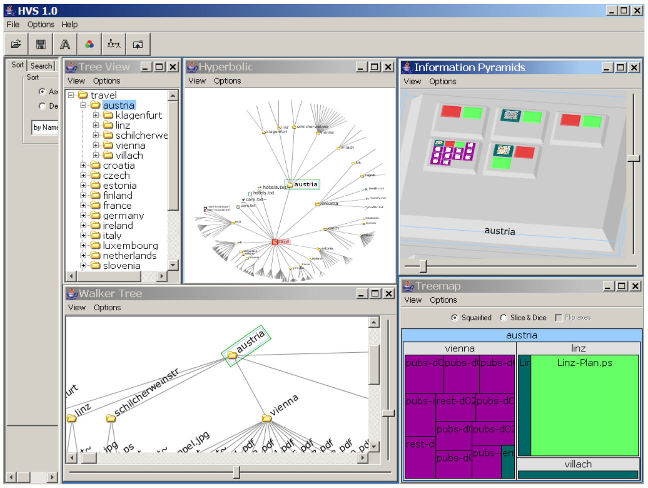

Figure 1: The Hierarchical Visualisation System (HVS) provides multiple, synchronised views of a hierarchy. Top left is a classic Windows Explorer style tree view. Top middle is an implementation of a hyperbolic browser. Top right is the information pyramids browser. Bottom left is a classic Walker tree layout. Bottom right is a treemap browser. The hierarchy shown here is a tree of notes and documents concerning various travel destinations.

3 Hierarchy Visualisations

Nine different hierarchical visualisation techniques are currently implemented within HVS:

- *Tree View*. The classical Windows-Explorer style tree browser based on the Java JTree class, where directories can be expanded and collapsed.
- *Walker Tree Browser*. The root of the tree is drawn at the top, its children are placed in a row beneath it, and so forth [3].
- *Information Pyramids*. Directories and subdirectories are placed recursively on plateaus [1].
- *Treemaps*. Recursive subdivision of rectangles according to the size of their children. Both squarified and slice and dice treemaps are implemented [7, 12]
- *Hyperbolic Browser*. The tree is laid out in hyperbolic space and then mapped to the unit disc for display [10].
- *Magic Eye View*. A classic Walker layout is projected onto the surface of a hemisphere for display [9], as can be seen in Figure 2.
- *InfoLens*. A pear-shaped magnifying lens is fixed in the centre right of the display and a Walker tree layout can be dragged over it to magnify a certain area of interest.
- *Sunburst*. A pace-filling radial layout, where subtrees are expanded like opening a fan.

[*] e-mail:kandrews@iicm.edu

[†] e-mail:Werner.Putz@gmx.net

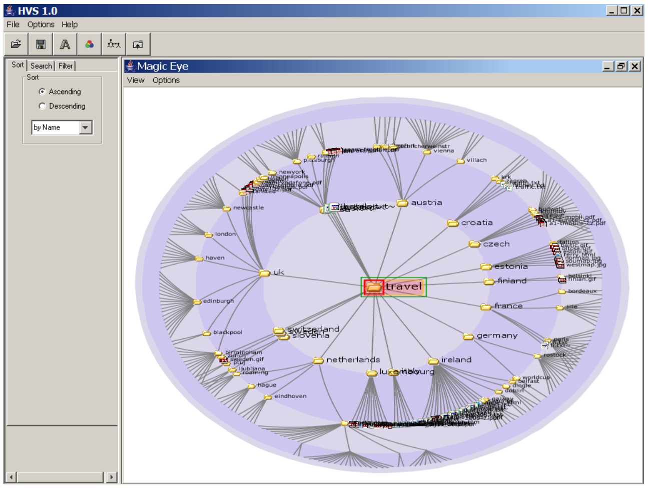

Figure 2: The HVS magic eye view.

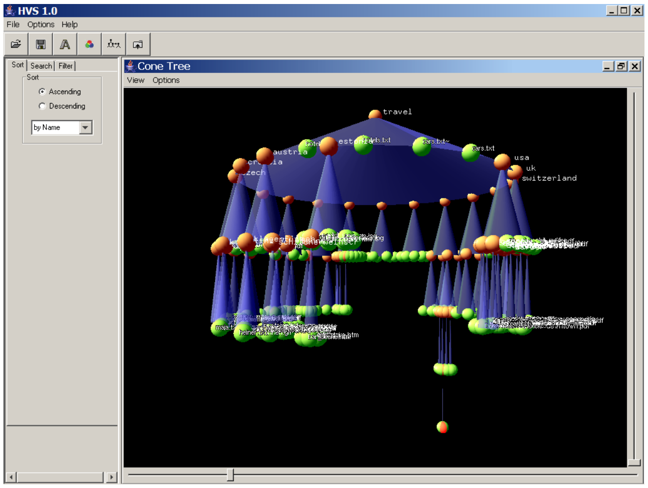

Figure 3: The HVS cone tree browser.

- *Cone Tree*. Children are recursively placed around the base of a cone emanating from their parent [11], as can be seen in Figure 3.

4 Related Work

The Information Visualization CyberInfrastructure (IVC) from Indiana University [2] is a fairly loose Java framework for information visualisation. IVC provides an XML-based interchange format, so that different visualisations can register and then interact with the same dataset. Treemaps, radial trees, and a hyperbolic browser have been implemented for hierarchies.

The InfoVis Toolkit (IVTK) [4] is an excellent general Java toolkit for information visualisation. It uses an efficient table-based internal data representation. IVTK is more general than HVS, supporting graph and tabular structures. Treemaps and radial trees are available for hierarchical information.

prefuse [6] is an extremely flexible Java framework for information visualisation based on a visualisation pipeline. It uses an underlying graph model for its internal data representation. Developers build an interactive visualisation at a high level of abstraction by chaining together components. Hierarchy browsers implemented include radial, treemap, hyperbolic, and degree-of-interest trees.

5 Concluding Remarks

This demo presented the Hierarchical Visualisation System (HVS), a framework for the visualisation of hierarchical structures. HVS is intended to serve as a reference collection for hierarchical visualisation techniques, as a testbed for new hierarchical visualisation techniques, and as an environment for running comparative studies of hierarchy browsers.

In the future, it is planned to implement further hierarchy browsers, including dendograms, information slices, a radial node-link layout, a botanic visualisation [8], and a recursive voronoi browser.

References

[1] Keith Andrews. Visual exploration of large hierarchies with information pyramids. In *Proc. Sixth International Conference on Information Visualisation (IV'02)*, pages 793–798, London, England, July 2002. IEEE Computer Society Press.

[2] Jason Baumgartner and Katy Brner. Interactive poster: Towards an xml toolkit for a software repository supporting information visualization education and research. In *Interactive Poster, IEEE Symposium on Information Visualization (InfoVis 2002)*, Boston, Massachusetts, USA, October 2002.

[3] Christoph Buchheim, Michael Jnger, and Sebastian Leipert. Improving walker's algorithm to run in linear time. In *Proc. Graph Drawing 2002*, Irvine, California, August 2002. Springer LNCS.

[4] Jean-Daniel Fekete. The infovis toolkit. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2004)*, pages 167–174, Austin, Texas, USA, October 2004.

[5] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[6] Jeffrey Heer, Stuart K. Card, and James A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. CHI 2005*, Portland, Oregon, USA, April 2005. ACM.

[7] Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proc. IEEE Visualization '91*, pages 284–291, San Diego, California, October 1991. IEEE Computer Society.

[8] Ernst Kleiberg, Huub van de Wetering, and Jack van Wijk. Botanical visualization of huge hierarchies. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2001)*, pages 87–94, San Diego, California, October 2001. IEEE Computer Society.

[9] Matthias Kreuseler, Norma Lopez, and Heidrun Schuhmann. A scalable framework for information visualization. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2000)*, pages 27–36, Salt Lake City, Utah, October 2000. IEEE Computer Society.

[10] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. CHI'95*, pages 401–408, Denver, Colorado, May 1995. ACM.

[11] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. CHI'91*, pages 189–194, New Orleans, Louisiana, May 1991. ACM.

[12] Ben Shneiderman and Martin Wattenberg. Ordered treemap layouts. In *Proc. IEEE Symposium on Information Visualization (InfoVis 2001)*, pages 73–78, San Diego, California, October 2001. IEEE Computer Society.