Visualising File-Systems Using ENCCON Model

Quang V. Nguyen and Mao L. Huang

Faculty of Information Technology
University of Technology, Sydney, Australia

quvnguye@it.uts.edu.au, maolin@it.uts.edu.au

Abstract

[1]This paper describes a new approach for visualising the file structures. Our technique uses an *enclosure* + *connection* (ENCCON) approach that provides an overall view of the entire file/directory hierarchy that gives a better understanding of the folder-folder and folder-file relationships, and therefore makes easier for the navigation. We firstly use the rectangular regions to present the folders. This allows the user to immediately percept the location, size (number of files contained) and other properties of any particular folder. To enrich the visual attributes of the file structure, we also use a node-link diagram to present the "belonging" relationships. This "belonging" relationship includes folder-folder and folder-file relationships. We use a semantic zooming technique to enlarge the display of files and folders in a particular focused area. The animation is also accommodated in order to preserve the mental map [Bartram 1997] during the navigation.

Keywords: file systems, information visualisation, semantic zooming, animation.

1 Introduction

There are many types of information that is hierarchically structured, such as computer file systems, organisation chats, catalogues, web-site maps, etc. The sizes of these structures increase significantly each year, and the traditional visualization techniques have been insufficient to handle such size of the information.

Although several techniques for visualising and manipulating computer file systems have been proposed and implemented (Bruggemann-Klein and Wood 1988, Lamping and Rao 1995, Johnson, and Shneiderman 1991, etc), there are few adequate tools that can be used to handle large directory structures on hard-disk drives. Traditional methods for representing file/directory hierarchies can be roughly organised into four categories: *listings*, *outlines*, *enclosure approach* and *connection approach*.

- **Listings:** they are ordinary methods of displaying the file systems in the early state. Nowadays, this approach is still used for browsing and manipulating file systems in most of UNIX and DOS command line based file systems. Listings are simple, text-oriented, and capable of providing detailed content information. The technique, however, present poorly the structure of information that takes large amount of time and effort for browsing and understanding the structures (see Figure 1). Furthermore, it do not have interaction feather.

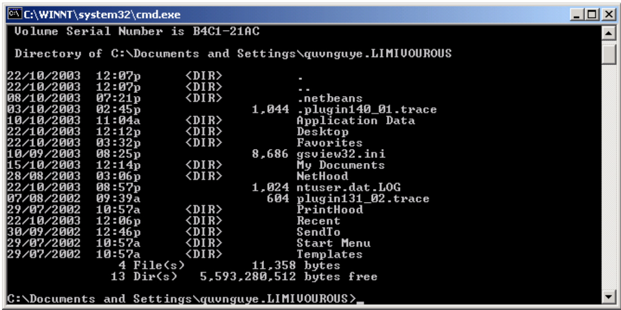

Figure 1: An example of a listing method (MS DOS Command Window)

- **Outlines:** outline methods can explicitly provide both structural and content of information. They are commonly used in modern window based file/directory systems, such as window explorer (see Figure 2) and UNIX 'du'. This approach, however, is not often adequate for presenting file systems with deep-hierarchies (say more then 12 levels) or with more than a few hundred nodes. This is because the structural indentation is limited and we can only view a few lines at a time.

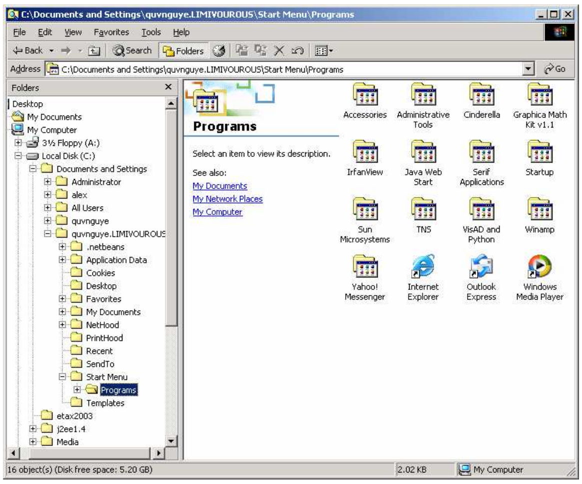

Figure 2: An example of using outline method (Window Explorer of MS Window 2000)

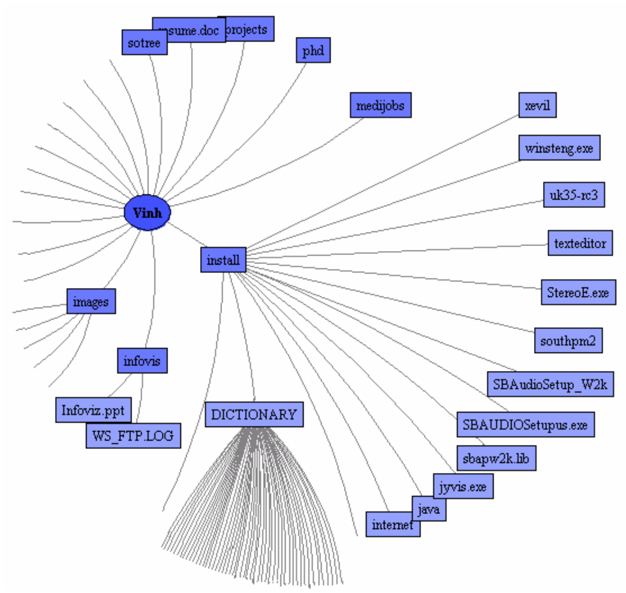

Figure 3: An example of connection method (a file system visualisation using Hyperbolic Browser)

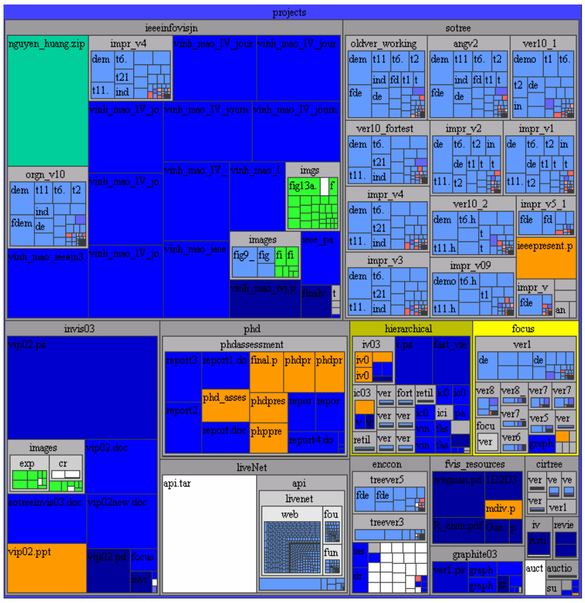

Figure 4: An example of enclosure method (a file system visualisation using Squarified Treemaps)

- **Connection:** this approach uses a node-link diagram to visualise the hierarchies of file systems. Specifically, it draws a set of visible graphical edges in the diagram to link a set of visible graphical nodes from the parents to their children. The advantage of this approach is that the human can directly see the relationships among the structural information. These techniques, however, are not efficient in term of utilising display space. Figure 3 shows a visualisation of the hierarchy using connection approach. The conventional connection techniques for visualising file/directory systems are: *Classical Hierarchical Tree Layout* (Reingold and Tilford 1981, Bruggemann-Klein and Wood 1988), *Hyperbolic Browser* (Lamping and Rao 1995), *RINGS* (Teoh and Ma 2002), *SpaceTree* (Plaisant et al. 2002), *3D Information Landscape* (Andrews et al. 1996), *Cone Tree* (Robertson et al. 1993), etc.

- **Enclosure:** This approach uses the area partitioning concept to represent the hierarchical structures that each sub-hierarchy is represented by a geometric region and located inside their parents' geometric region. Enclosure techniques are usually more efficient in term of utilising the display space in comparison with the connection methods. However, the lack of direct edges linking between nodes might prevent viewers from understanding the relational information. The typical enclosure techniques for visualising file systems are *Tree-Maps* (Johnson, and Shneiderman 1991, Wijk et al. 1999, Bruls et al. 2000), *Information Pyramids* (Andrews 1997), *Information Slices* (Andrews and Heidegger 1998), etc. Figure 4 shows an example of visualising a file system using enclosure approach.

2 ENCCON Model

In this paper, we propose an <u>enc</u>losure + <u>con</u>nection approach (ENCCON) to enrich the visual attributes of file systems. Our technique layouts the directory hierarchy using area partitioning while it also uses *node-link diagrams* to show the folder-file (and folder-folder) relations of file/directory systems. Detail of the *enclosure + connection* approach can be found at SO Tree (Nguyen and Huang 2003). The current prototype of ENCCON is programmed in Java Applet in order to make it more available to users. The system includes a small program which scans the local file systems and creates HTML files to run using the web browsers.

2.1 Layout

Our technique provides an overall view of the entire file/directory hierarchy that gives a better understanding of the folder-folder and folder-file relationships. This makes easier for the navigation. Figure 5 shows a ENCCON display of the file directory rooted at '*Vinh*'. This graph uses the rectangular regions to present the folders. This allows the user to immediately percept the location, size (number of files contained) and other properties associated with of any particular folder. To enrich the visual aspects of the graph, we also use a node-link diagram to present the parent-child relationships. This parent-child relationship includes folder-folder and folder-file relationships. This hierarchy has the maximum depth of 6, and the subfolder 'Medijobs' contains much more files compared to its siblings, etc. In other word, the user understands easily the structural relationship of the file systems.

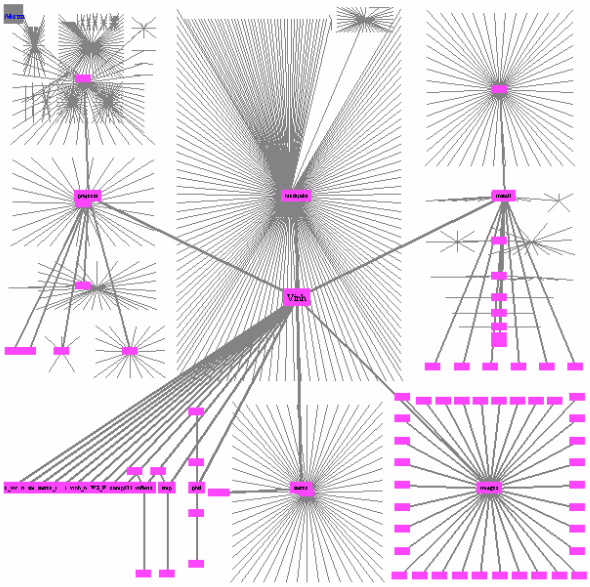

Figure 5: An example of a file system visualisation using ENCCON

2.2 Navigation

We use semantic zooming for the navigation of the file hierarchy. This technique enlarges the display of a focused sub-hierarchy with more details the (see Figures 6a and 6c). Detail of the navigation is described below.

When being selected by a left mouse-click (zoom in), the selected node moves forward to the position of the root, i.e. centre of the rectangular display area. The display region of the selected node now expands to the entire display area. In other words, we only visualise the sub-tree of selected node. This viewing technique requires the recalculation of positions of all vertexes in a sub-tree at a time in corresponding to the left-click (see Figure 6c). The system displays back the father's hierarchy (zoom out) when the user clicks on the right button.

The animation is also accommodated with the navigation in order to preserve the mental map (Bartram 1997) during the navigation. The animation is achieved by smoothly moving the nodes to the new location, and fading in and fading out the nodes. In detail, the animation process can be described below:

- Animation for zooming in (corresponding to the left mouse-click on a node): the focused sub-hierarchy expands smoothly to occupy the entirely display area. The size of nodes in this sub-tree also increases to reach their new size. Simultaneously, the colour of nodes, which is visible at the previous state, is fading out to the background colour (see figure 6b).
- Animation for zooming out (corresponding to right mouse-click): the nodes of the sub-tree (from the previous state) move smoothly to their new location, and their size decreases to the new size. Simultaneously, the parent's hierarchy is smoothly fading in from the background colour to the node colour (see figure 6d).

2.3 Display property

File systems usually include hundreds or even millions files and directories. Unfortunately, it is inadequate to display hundreds or more fully labelled nodes on the normal computer screen at once. Thus, we only display the labels for nodes which are within three levels from the focused directory. This property can be adjusted through an interactive menu to increase or decrease the number of nodes whose labels are displayed on the screen.

As same as the other area division approaches where a child's area is always smaller than its father, we also apply this rule to our viewing technique. The size of nodes and the width of edges we choose are proportional to their levels in the hierarchy. This means that, the closer to the root of focused hierarchy, the larger of nodes and the wider of edges. This rule, some how, improves the clarity of the presentation of tree hierarchies.

When the mouse pointer is over a node, the system will highlight the display of this sub-hierarchy. The selected node is also highlighted by increasing its size and brighter its colour.

3 Examples

Figures 7a and 7b are examples of applying our technique two file systems. Figure 7a displays the hierarchy of my 'projects' folder (approximately 2750 files and directories). Figure 7b displays the hierarchy of 'J2SE 1.4's Documentations' folder (approximately 9550 files and directories).

4 Conclusion and future works

We have presented our ECCON approach for visualising file systems. This technique uses both enclosure and connection in its visualisation. The layout algorithm can draw the entire tree structure of the file systems using enclosure manner. The system applies the semantic zooming technique for navigating through the file/directory hierarchy. Animation is also accommodated in order to preserve the mental map and reduce the human cognitive loads during the navigation. This technique still has a few limitations and needs further refinement in order to improve the optimisation of display space, and overcome the problem with long-name files and directory. In spite of these limitations, we believe that it is a valuable tool for visualisation tool for the file systems.

Next step, we will investigate new layout algorithms to improve the efficiency of using display space. We will investigate new focus+context (Herman et al. 2000) viewing techniques that can keep the global view of the entire file/directory structures in a small region while a detailed view of focused directory can be displayed in the large region during the navigation. We also try to overcome the overlapping labelling problem where the labels of files or directories are too long.

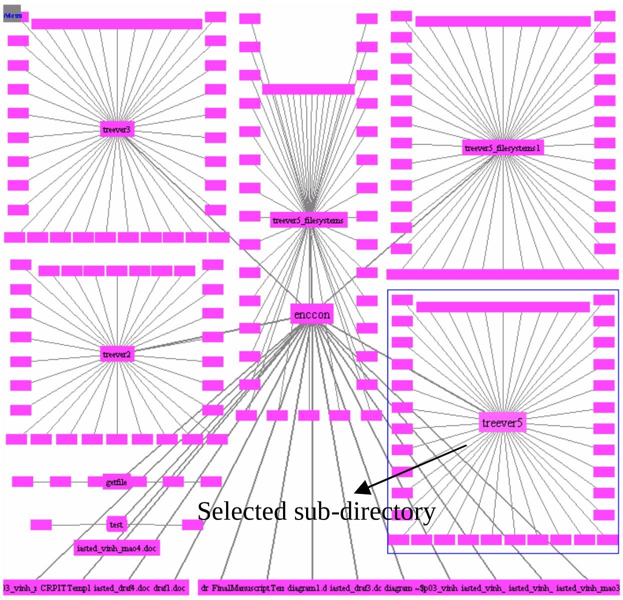

Figure 6a: An example of displaying the entire directory before the navigation.

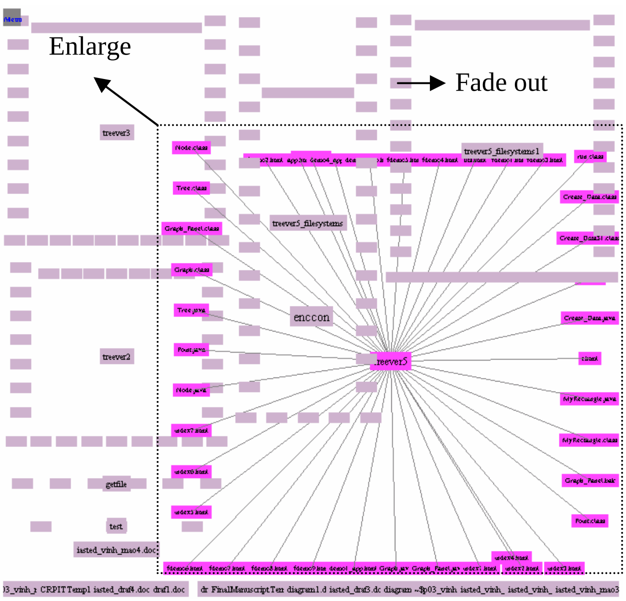

Figure 6b: An example of animated *zooming in* after the mouse-click on the node 'treever5'.

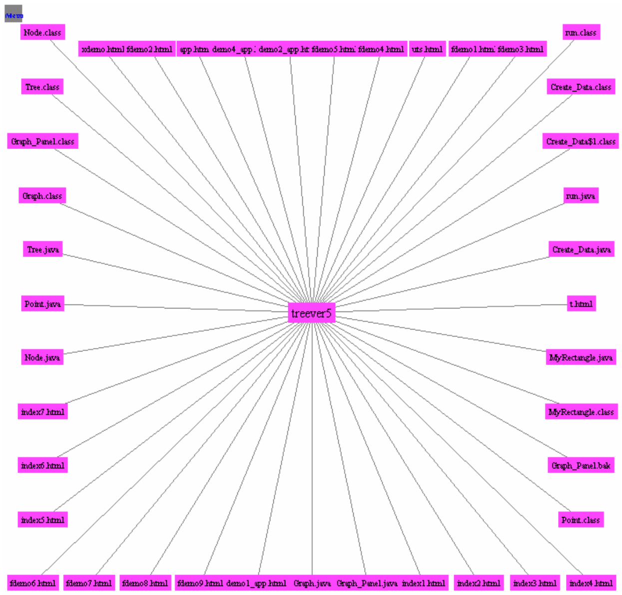

Figure 6c: An example of semantic *zooming in* after the animatinon.

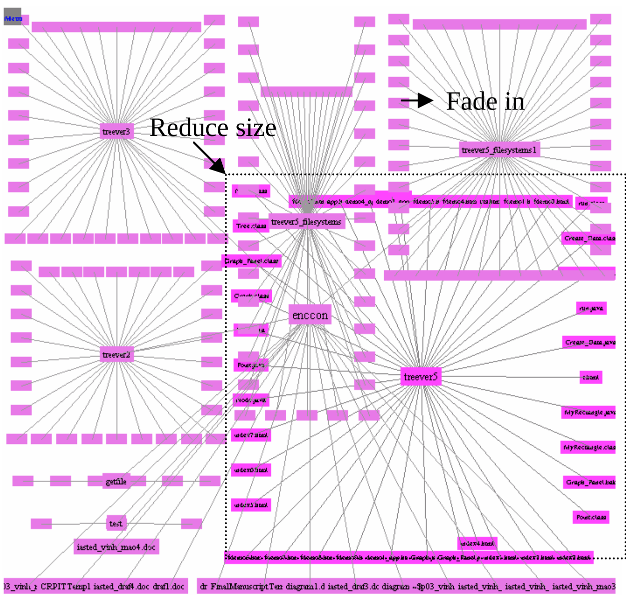

Figure 6d: An example of animated *zooming out* after the right mouse-click to move back to the display of parent's hierarchy.

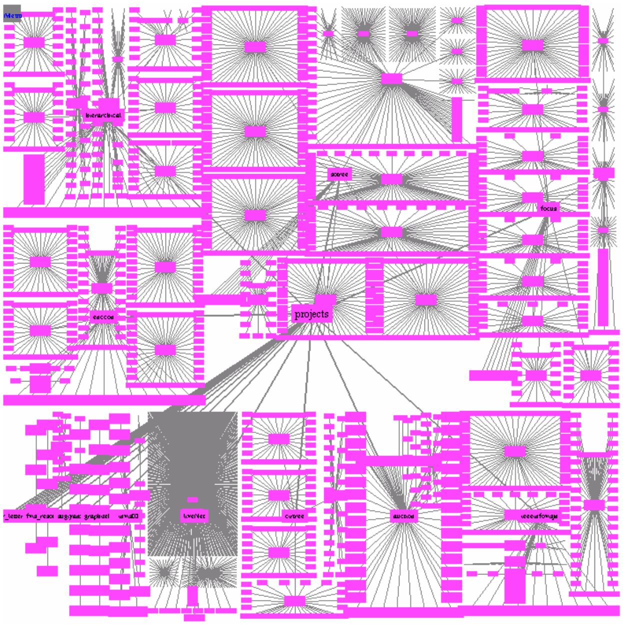

Figure 7a: An example of the visualisation of a file system (approximately 2750 files and directories)

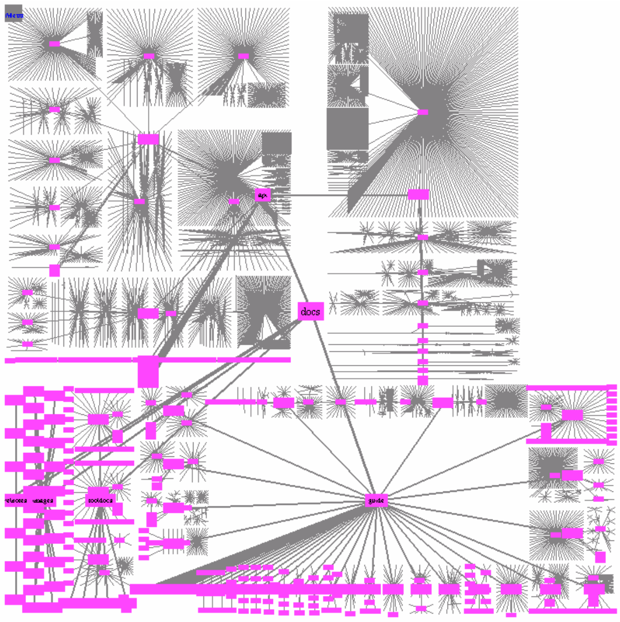

Figure 7b: An example of the visualisation of the file system of the entire Java Documentations directory (approximately 9550 files and directories)

5 References

Andrews, K., Heidegger, H. (1998): Information slices: visualising and exploring large hierarchies using cascading, semicircular discs. *Proc. IEEE Information Visualization Symposium*, Carolina, USA, 9-12.

Andrews, K., Pichler, M., Wolf, P. (1996): Towards rich information landscapes for visualising structured web spaces. *Proc. 2nd IEEE Symposium on Information Visualisation*, California, USA, 62-63.

Andrews, K., Wolte, J., Pichler, M. (1997): Information pyramids: a new approach to visualising large hierarchies. *Proc. of IEEE Visualization'97*, Phoenix, Arizona, 49-52.

Bartram, L. (1997): Perceptual and interpretative properties of motion for information visualization, *Technical Report CMPT-TR-1997-15*, School of Computing Science, Simon Fraser University.

Bruggemann-Klein, A., Wood, D. (1988): Drawing trees nicely with TEX. Electronic Publishing, **2**(2): 101-115.

Bruls, M., Huizing, K., van Wijk, J.J. (2000): Squarified treemaps. *Proc. VisSym '00*, Amsterdam, the Netherlands, 33-42.

Herman, I., Melançon, G., Marshall, M.S. (2000): Graph visualization in information visualization: a survey, *IEEE Transactions on Visualization and Computer Graphics*, **6**, 24-44.

Johnson, B., Shneiderman, B. (1991): Tree-maps: a space-filling approach to the visualization of hierarchical information structures. *Proc. the 1991 IEEE Visualization*, Piscataway, NJ, 284-291.

Lamping, J., Rao, R. (1995): The hyperbolic browser: a focus+context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, **7**, 33-55.

Nguyen, Q.V., Huang, M.L. (2003): Space-optimized tree: a connection + enclosure approach for the visualization of large hierarchies, *Information Visualization Journal*, Palgrave, **2**(1), 3-15.

Plaisant, C., Grosjean, J., Bederson, B.B. (2002): SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. *Proc. IEEE Symposium on Information Visualization*, Boston, USA, 57-64.

Reingold, E.M., Tilford, J.S. (1981): Tidier drawing of trees. IEEE Transactions on Software Engineering, **7**(2): 223-228.

Robertson, G.G., Card, S.K., Mackinlay, J.D. (1993): Information visualization using 3D interactive animation. *Communication of the ACM*, **36**, 57-71.

Teoh, S.T., Ma, K.L. (2002): Rings: a technique for visualizing large hierarchies. *Proc. Graph Drawing (GD 2002)*, California, USA, 268-275.

van Wijk, J.J., van de Wetering, H. (1999): Cushion treemaps: visualization of hierarchical information. *Proc. IEEE InfoVis '99*, San Francisco, California, 73-78.