

Visualizing MeSH Dataset using Radial Tree Layout

Nihar Sheth

nisheth@indiana.edu

Qin Cai

qcai@cs.indiana.edu

Indiana University, Bloomington

April 29, 2003.

Abstract

Many display layout techniques were developed for the visualization of hierarchical data set like Cone tree, Treemap, Hyperbolic tree etc. Here, we present a new focus + context (fisheye) technique for visualizing and manipulating large hierarchies. We use the well known radial tree layout method, in which focused node is placed in center of display, and all other nodes are rendered on appropriate circular level around that selected focused node. Our main goals are displaying large hierarchies, developing animation technique for transition to a new layout when a new focus node is selected, and displaying non hierarchical relationship in hierarchical data on demand. We apply this technique for the visualization of MeSH Data set.

Keywords

MeSH, Information Visualization, Radial Tree Layout, Interaction, Focus+context technique

Introduction

In last one decade, Information Visualization research community has explored many display layout techniques for the visualization of large hierarchical datasets. Most of them have used the power of focus + context technique, in which detailed views of particular parts of an information set are blended in some with an overall view of the dataset. Some of them applied interactive graphics and animation techniques for visualizing and making better sense of large information set [4]. We present a technique, called the RadialTree Browser, for visualizing and manipulating large hierarchies.

Radial Tree Browser was initially inspired by the work [1] and [2], as neither of them were able to tackle what we want to achieve. We want to visualize large hierarchies with few non-tree cross links. At the same time, we want to provide animation and interaction features for better usability of visualization system. The silent features of Radial Tree layout : 1) Components diminish in size as they move outwards, 2) there can be exponential growth in the number of components from center to edges. These properties – “fisheye” distortion and the ability to uniformly embed an exponentially growing structure are the aspects of the layout that attracted our attention.

The RadialTree Browser initially displays a tree with its root at the center, but the display can be smoothly transformed to bring other nodes into focus. In all cases, the amount of space available to a node falls of as continuous function of its levels in the tree from the node in the center. Thus, the context always includes several generations of parents, siblings, and children, making it easier for the user to explore the hierarchy without getting lost.

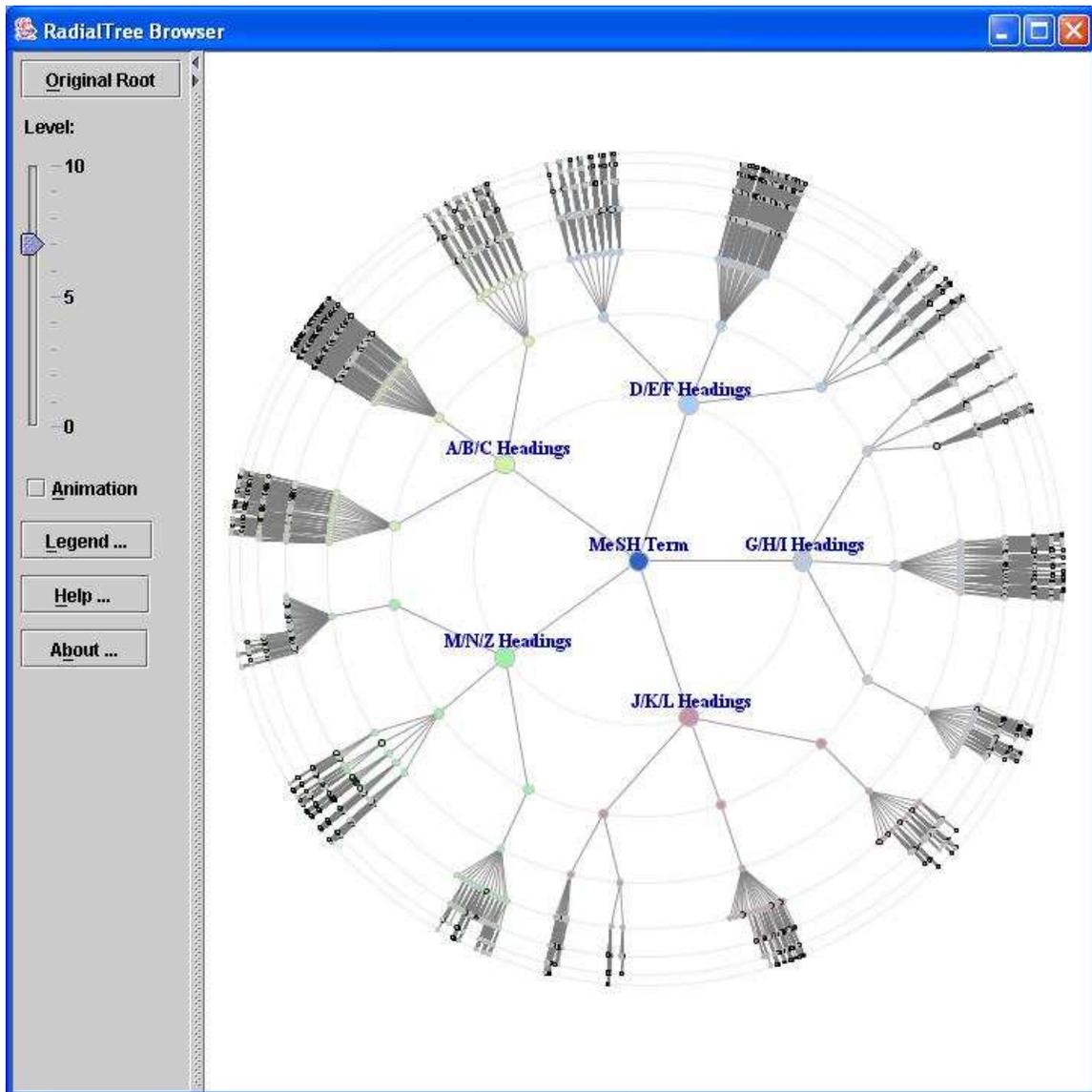


Figure 1: RadialTree Browser

The RadialTree browser supports effective interaction with much larger hierarchies than conventional hierarchy viewers. During the development, we used file and directory structure for visualization. In a 800 pixel by 600 pixel window, we were easily able to visualize 40000 nodes thru RadialTree Browser. We implemented RadialTree Browser using Java Swing API and thus, it is totally platform independent.

Problems and Related Work

Many hierarchies, such as controlled vocabularies or file directory structures, are too large to display in their entirety on a computer screen. The conventional display approach maps all the hierarchy into a region that is larger than the display and then uses scrolling to move around the region. This approach has the problem that the user can't see the relationship of the visible portion of the tree to the entire tree structure. It would be really

useful to be able to see the entire hierarchy while focusing on any particular part so that the relationship of parts to the whole can be seen and thus, if needed, focus can be moved to other parts in a smooth and continuous way. Many times, in large hierarchies, there exist few nodes in hierarchy which have non-hierarchical relationship with nodes from other sub-hierarchies, at the same time, playing role in their hierarchy also. Such links between different sub-hierarchies are difficult to display in hierarchical display techniques.

Many focus+context display techniques have been introduced in the last fifteen years to address the needs of many types of information structures [5, 6]. Many of these focus+context techniques, including the document lens [7] and the perspective wall [8] could be applied to browsing trees laid out using conventional 2D layout techniques. The problem is that there is no satisfactory conventional 2D layout of a large tree, because of its exponential growth.

The Cone Tree [9] modifies the above approach by embedding the tree in a three dimensional space. This embedding of the tree has joints that can be rotated to bring different parts of the tree into the focus. This technique requires currently expensive 3D animation support. Plus, trees with more than approximately 1000 nodes are difficult to manipulate.

Another tree browsing technique is Treemaps [10] which allocates the entire space of a display area to the nodes of the tree by dividing the space of a node among itself and its descendants according to properties of the node. This technique utilizes space efficiently and can be used to look for values and patterns amongst a large collections of values with agglomerate hierarchically, however it tends to obscure the hierarchical structure of the values and provides no way of focusing on one part of hierarchy without losing the context.

The Hyperbolic tree is one more good technique which replaces the conventional approach of laying a tree out on a Euclidean plane by doing layout on the hyperbolic plane and maps this plane onto a circular display region. This mapping displays portions of the plane near the origin using more space than other portions of the plane. Translating the hierarchy on the hyperbolic plane provides a mechanism for controlling which portions of the structure receives the most space without compromising the illusion of the viewing the entire hyperbolic plane. This layout technique doesn't utilize display space very efficiently and trees with more than approximately 5000 nodes are difficult to display. It also doesn't visualize non-hierarchical cross-links.

[1] and [2] used RadialTree layout for visualizing hierarchies, in which Nodes are arranged on concentric rings around the focus node. Each node lies on the ring corresponding to its distance from the focus node. In [1], they were able to successfully visualize large hierarchies, but there was no animation or interaction feature available. While in [2], they provided very efficient animation and interaction features, but was not scalable. We are using Radial Tree layout for our work, but our clear goal of RadialTree browser is to visualize very large hierarchies with non-hierarchical cross-links, at the

same time, providing animation and proper interaction features for better usability of visualization.

RadialTree Browser

Layout

In Radial Tree layout, the focus node is placed at the center of the display and all other nodes are laid out around it. These nodes are arranged on concentric rings around the focus node. Each node lies on the ring corresponding to its level in hierarchy from the focus node. Immediate children of the focus node lie on the smallest inner ring, their children lie on the second smallest ring, and so on. We displayed these rings explicitly to make the levels apparent. The angular position of a node on its ring is determined by the sector of the ring allocated to it. Each node is allocated a sector within the sector assigned to its parent, with size proportional to the angular width of that node's subtree.

Space Allocation

Simplest approach for space allocation can be as per described in [1], in which all the nodes are the same size, and the angular width of a node's subtree is simply the number of leaf nodes among its descendants. In our approach, node size decrease as number of levels increases. For space allocation of a node, two parameters need to be calculated: angular width and angular angle. For better space utilization, we decided to different space allocation rules to nodes depending upon the level of node. Node in center is given full 360 Angle as angular width. We mainly have different rules of node of level one and node of other levels.

For nodes of level one,

Angular width

= (angular width of center node / no. of nodes at level two) * no of children, if non-leaf

= angular width of center node / no. of nodes at level two , if leaf node.

Here, angular width of center node always be 2π

Angular angle is just mid point of angular width of the node.

For nodes of level other then level one,

Angular width

= (angular width of parent node) / no. of siblings

Angular angle is calculated using index of given node and angular angle and width of its parent.

Angular angle = Parent's angle – parent's angular width / 2 + node's angular width * node's index.

From above rules, it is clear that nodes in other then level one are given equal width as with their siblings.

Animation

To explore the hierarchy, a user selects a visible node to become the new focus node. The new layout is found by performing node position calculation from the new focused node. The edges between nodes are reinterpreted as a new set of parent-child relationships. The new layout is determined by assigning each node to the appropriate ring and allocation angular widths of rings discussed above.

While it is sufficient to show the hierarchy from the perspective of the new focus, simply switching to this new view can cause a highly disorienting rearrangement. To reduce this disorientation, we use animation to perform a smooth transition from old view to new view. We employed simplest approach of animation, in which each node is moved along the straight-line path from its old position to its new position. To provide good visual constancy, we used slow-in, slow-out animation technique rather than straight linear animation. As per this technique, the animation begins slowly, smoothly accelerates, and then decelerates at the end. Most of the movement occurs in the middle third of the time intervals. This provides good visual cues to help the user anticipate the movement of nodes into their new positions.

Non-tree cross link representation

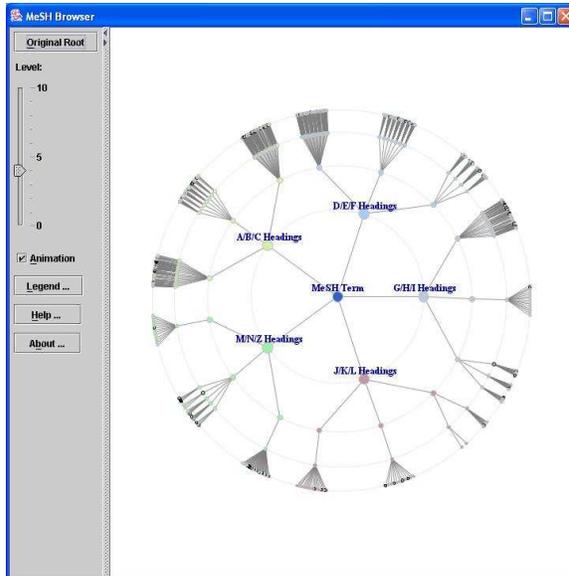
As main goal of the visualization system was to visualize large hierarchies, we decided to display non-tree cross link on user demand only. Non-tree cross links are displayed differently than usual parent-child links in hierarchies. We used dotted lines for non-tree cross links. User can either double click on node or select appropriate pop-up by right clicking on node to display non-tree cross links. In the same way, user can also hide the cross links any time he/she wants. The main aim behind on-demand cross link display was not to make actual hierarchical visualization crowded with cross-links, at the same time, providing facility to visualize cross-links embedded in the hierarchy.

Visual Clues using Colors

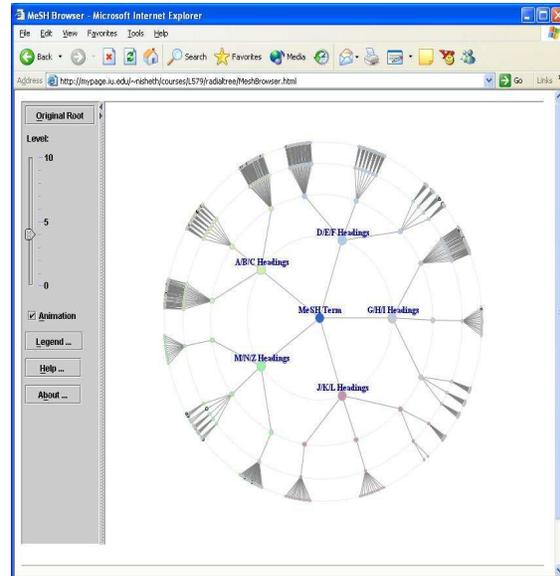
We successfully used different colors for providing users various visual clues in large hierarchical dataset. We gave one constant color to original root node to make it easily identifiable in large data set. Similarly, we gave one other constant color to path to current focus node to original root node. These two visual clues become very handy when a user is exploring information in low level of hierarchy. We also provided the feature to give any color to any node in visualization. This can be done by specifying appropriate color information with node information in input data file.

Application: MeSH Browser

We applied our work for the development of visualization tool for MeSH dataset. We implemented this tool using Java Swing, and thus it was totally platform independent. It can run in both mode – Application and Applet. So it can be easily deployed on the web.



MeSH Browser Application



MeSH Browser Applet

Data Analysis

Medical Subject Headings (MeSH) [12] is the National Library of Medicine's controlled vocabulary thesaurus. It consists of set of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity. The MeSH thesaurus is used by NLM for indexing articles from 4,600 of the world's leading biomedical journals for the MEDLINE [11] database. It is also used for other NLM-produced databases that include cataloging of books, documents and audiovisual acquired by the Library. Each bibliographic reference is associated with a set of MeSH terms that describe the content of the item. Similarly, search queries use MeSH vocabulary to find items on a desired topic. More detailed information can be found at [12].

MeSH dataset is publicly available for download from MeSH Home page [12]. US NLM is the only creator, maintainer, and provider of MeSH Data set. The dataset version used here is 2003 MeSH. The dataset is divided into two parts: 1) Total detailed MeSH information in single file. XML and ASCII file formats are supported for this. 2) MeSH Terms Tree structure in ASCII format [appendix 1]. This file contains only tree structure information. In this file, MeSH main headings with their tree numbers are placed in hierarchical arrangement by sorting them by tree number. We used this file for the purpose of development of visualization tool.

User and Current System Analysis

MeSH is used by NLM for indexing and categorizing biomedical related articles. Thus, one of the main user group of MeSH includes the staff – indexers, catalogers at NLM. MeSH is also extensively used by biomedical community as vocabulary thesaurus in this

field. This user group has quite good subject expertise. Knowledge about how to do various mouse operations are only required for successful use of our tool. We can easily assume that this highly educated user group has little enough computer skills to explore and browser MeSH terms thru our tool.

The MeSH Browser [13] from NLM is developed as online vocabulary look up aid for use with MeSH. It is designed to help quickly locate descriptors of possible interest and to show the hierarchy in which descriptor of interest appear. It provides very simple text-based, web-based hierarchical tree structure to locate descriptors. The problem with this simple tree structure is that it is not possible to see the neighboring descriptors of selected descriptor, which can be often found to be useful to see them. Even grand parent node information is lost while exploring nodes at lower levels in hierarchy.

Data Preparation

As RadialTree browser tool takes data input thru XML file defined by XML schema [appendix 2], our main task in data preparation was to transform ASCII MeSH tree structure to required XML structure. We wrote one java program which read MeSH Tree structure and creates required XML file using Java XML API.

For better visualization, we decided to use color coding scheme. We decided to give different color to each category of MeSH. But as there are fifteen categories, it will information overload if we use fifteen different colors on the display. So we grouped the categories into five groups, each consisting of three categories. Then we used five different colors for each different group. Five groups are as below:

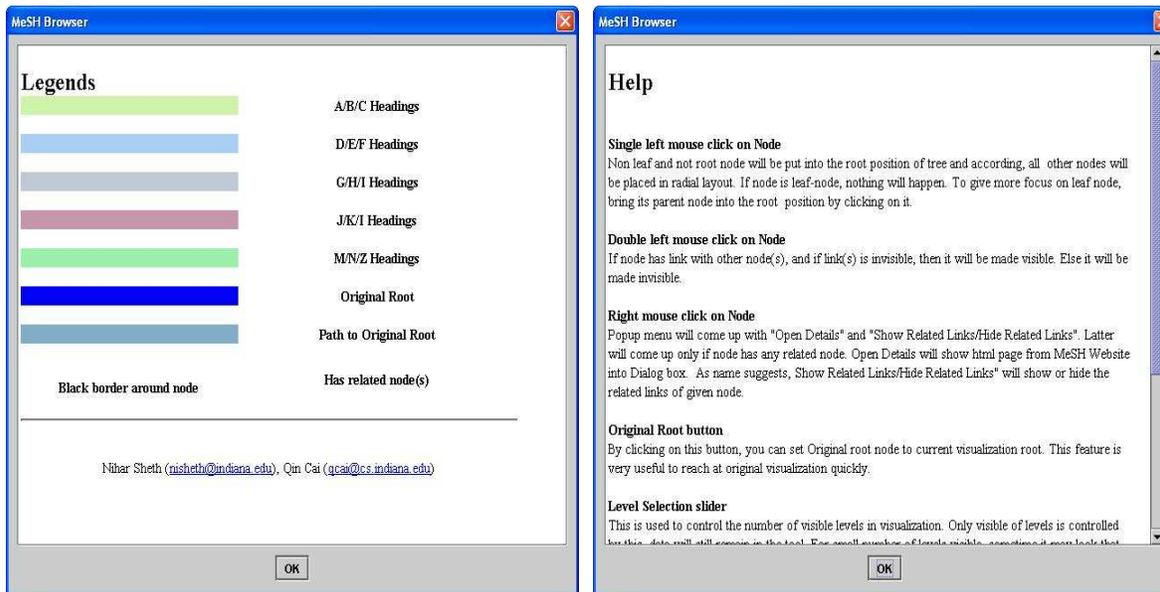
Group	Category	Color(R,G,B)
A/B/C Heading	A:Anatomy	207,244,170
	B:Organisms	
	C:Diseases	
D/E/F Headings	D:Chemicals and Drugs	170,207,244
	E:Analytical,Diagnostic and Therapeutic Techniquis and Equipments	
	F:Psychiatry and Psychology	
G/H/I Headings	G:Biological Sciences	193,200,213
	H:Physical Sciences	
	I:Anthropology, Education, Sociology and Social Phenomena	
J/K/L Headings	J:Technology and Food and Beverages	200,150,170
	K:Humanities	
	L:Information Science	
M/N/Z Headings	M:Persons	155,244,170
	N:Health Care	
	Z:Geographic Locations	

We used MeSH Term heading as label of node and “Tree Number: Term Heading” as tool tip of the node. We coded this information into XML data file along with color coding information. We also created URL for each MeSH Term and incorporated it into each node description into XML data file.

System Features

We provided many interaction controls and on-demand detail feature in this tool. We can list down the features as following:

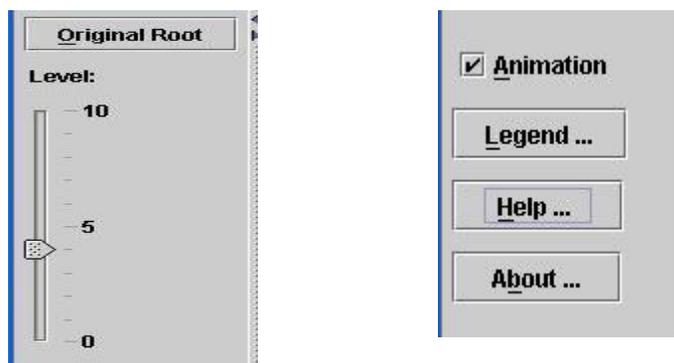
1. We provided enough legend information and help page for users can understand the tool themselves.



Legend

Help

2. We provided various control features to the user for controlling the working of the tool.



3. As shown in above figure, “Original Root” button is used to set Original root to current display root. This feature is very helpful to get back to main view when u r exploring nodes at very low levels in hierarchy.
4. “Level” Slider is used to control the number of levels to be displayed.

- Many times, users don't have patience to wait once they select new focus node. So we provided control to toggle "Animation" feature between on-off.
- User can select new focus node by single clicking on any of the node in visualization.
- User can ask for detailed information about the node by right-clicking on any node and then clicking on popup menu "Show details".



MeSH Browser

National Library of Medicine - Medical Subject Headings

2003 MeSH

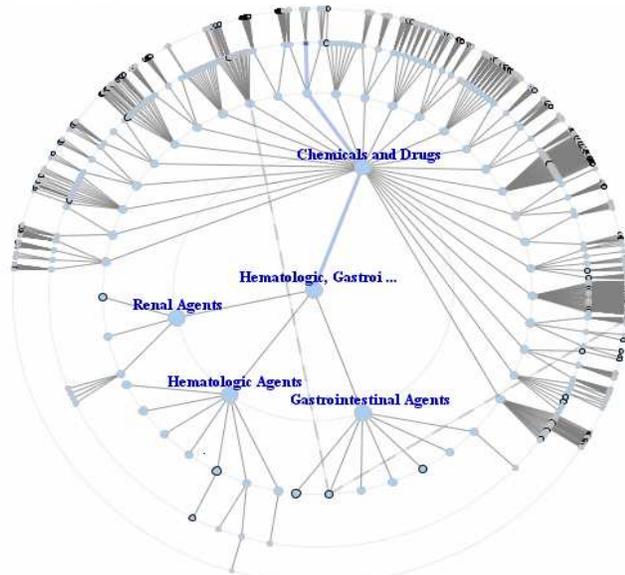
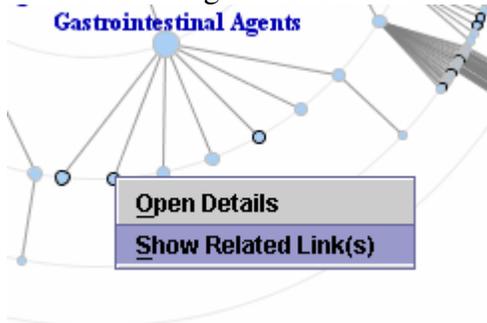
MeSH Descriptor Data

[Return to Entry Page](#)

Gastrointestinal Agents	
MeSH Heading	Gastrointestinal Agents
Tree Number	D19.316
Tree Number	D27.505.435.316
Scope Note	Drugs used for their effects on the gastrointestinal system, as to control gastric acidity, regulate gastrointestinal motility and water flow, and improve digestion.
Entry Term	Digestants
Entry Term	Gastric Agents
Entry Term	Gastric Drugs
Entry Term	Gastrointestinal Drugs

OK

- User can view non-tree cross links by right clicking on nodes having cross-links and then clicking on "Show Related Nodes" menu.



Future Extensions

We have identified following task as future extensions: 1) providing control of Node size and shape by specifying it in xml data file. 2) Search feature and visualize the resulting nodes so they can be easily differentiated from rest of the nodes.

Conclusions

Radial Tree layout provides an elegant solution to the problem of providing a focus+context display for large hierarchies with many non-tree cross links. Space allocation is done based on the level of node for efficient space utilization. We presented a method for animating the transitions from one view to the next in an appealing manner that reduces confusion. We used various color coding schemes to provide clue about the type of information they are looking. Interaction techniques were used for providing user to control the visualization. On-demand detail feature was also implemented.

We successfully applied these techniques to visualization of Mesh dataset. Informal user testing clearly suggests that animation, on-demand details, and other interactivity features improve the ability of user to explore the large dataset.

Acknowledgements

We like to thank Jason Baumgartner (jlbaumga@indiana.edu) for his constant guidance and help throughout this work. We also like to thank all students of Spring 2003 Information Visualization class for their comments, which helped us a lot during giving final touches to the RadialTree Browser.

References

1. G. Book and N. Keshary. Radial Tree graph drawing algorithm for representing large hierarchies. University of Connecticut, December 2001.
2. K Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated Exploration of Dynamic graphs with Radial layout. IEEE Symposium on Information Visualization, October 2001.
3. J. Lamping, R. Rao, and P. Pirolli. A Focus+context technique based on hyperbolic geometry for visualizing large hierarchies.
4. G.G. Robertson, S. K. Card, and J. D Mackinlay. Information Visualization using 3D interactive animation. Communications of the ACM, 36(4). 1993.
5. Y.K Leung and M.D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. ACM Transactions on Computer-Human Interaction. IEEE, 1993.
6. Manojit Sarkar and Marc H. Brown. Graphical fisheye views. Communication of the ACM, 37(12):73-84, December 1994.
7. George G. Robertson and J. D Mackinlay. The document lens. In proceeding of the ACM Symposium on User Interface Software and Technology. ACM Press, November 1993.
8. J. D Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In Proceeding of the ACM SIGCHI Conference of Human Factors in Computing Systems, pages 173-179, April 1991.

9. G. G. Robertson, J. D Mackinlay, and S. K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In Proceeding of the ACM SIGCHI Conference of Human Factors in Computing Systems, pages 189-194. April 1991.
10. B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information. In visualization 1991, pages 284-291. IEEE, 1991.
11. MEDLIN and PubMed Central: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>. Last accessed on May 3, 2003.
12. MeSH Home: <http://www.nlm.nih.gov/mesh/meshhome.html>. Last accessed on May 3, 2003.
13. MeSH Browser from NLM: <http://www.nlm.nih.gov/mesh/MBrowser.html>. Last accessed on May 3, 2003.

Appendices

1. MeSH Tree Structure File, Year 2003 version.
<http://mypage.iu.edu/~nisheth/courses/L579/radialtree/meshtree2003.txt>
2. XML Schema
<http://mypage.iu.edu/~nisheth/courses/L579/radialtree/tree.xsd>
3. XML Data file
<http://mypage.iu.edu/~nisheth/courses/L579/radialtree/NewMeSH.xml>