* Slide 1

Welcome everyone, today I will be giving a presentation on my data visualization master’s project proposal.

* Slide 2

First I will give an outline of what I will be talking about today.

First I will describe the Problem slash motivation. Next I will discuss the dataset the audience and the goals in more detail. Then I will talk about some related works. And then give a basic overview of the design. And then finally a conclusion.

* Slide 3

First lets talk about the problem/motivation. (CLICK)

Over the recent years, research involving the data visualization of both networks and trees has increased significantly. Such research involves the user interaction capabilities, optimal layout, and, minimized edge crossings. (CLICK)

However, minimal research has gone into the unique dataset combination of both trees and a network. In this situation there would be a set of trees in which the edges represent the hierarchical information of the tree. While the network edges provide information about the relationships between the trees or within a single tree. Even though the edges of the network and trees are representative of completely different information, the nodes themselves are the same and therefore causes both network edges and tree edges to be related. Consequentially, viewing only one of either tree or network data loses either the hierarchical component of the trees or the dependencies defined by the network. (CLICK) Therefore, our problem is to show the intra and inter dependencies of a network while also preserving the hierarchical component of the trees.

* Slide 4

Now lets go into more detail about the dataset. (CLICK)

Sandia first created the tree dataset which is a group of functional decomposition trees. A functional decomposition tree is a tree that breaks down an overall function of some device or system into smaller parts or subfunctions. The definition of the relationship between a parent and child was created. Additionally the node types and definitions of level of hierarchy was defined. (CLICK)

Next, the Sandia team created the network dataset of both intra and inter dependencies. (CLICK) AN intra edge is an edge between two nodes in the same tree that doesn’t follow the hierarchical edges already defined. (CLICK) While an inter edge is an edge between 2 nodes from different trees. (CLICK)These dependencies have edge types which define the relationship between 2 nodes. The definition of 2 nodes having generally an inter or intra dependency was formed as well. (CLICK) As a result there were 55 trees with approximately 50 nodes per tree giving us approximately a max of 2750 edges to work. Making this combined network and tree dataset a considerably medium dataset.

* Slide 5

Now that we described the dataset lets talk more about the goals of the users. Now keep in mind that the users would be the Sandia developers that created the dataset. (CLICK)

Therefore, Sandia’s goal is to perform what I consider step 0 of analysis of dataset: verifying and validating the dataset. There are two main components that need to be verified: the logic of the hierarchical functional decomposition definition of the trees, and the logic of the inter/intra dependencies definitions of the trees that make up a network. (CLICK) How this can be achieved is by identifying anomalies and patterns within the dataset.(CLICK)

Some basic anomalies or patterns worth identifying include: largest number of edges of a verb type, nodes with most connections via inter, intra and hierarchical connections, and the size, width, height of the trees. In this example we can tell Sub function 1 is an anomaly as it has many intra connections.

By identifying these types of anomalies developers will be able to determine that how the dataset was defined might have resulted in some unintentional and potentially undesirable anomalies or patterns. And therefore prompting the Sandia developers to rethink and edit the definitions of hierarchical or network connections in the dataset.(CLICK)

Therefore our newly defined problem is to (CLICK) Design a data visualization application that not only visualizes both the hierarchical and network components of the unique dataset but also allows developers to verify and validate the dataset by detecting network and tree related anomalies or patterns.

* Slide 6

Now with the goal and problem in mind lets discuss what research is already out there. First is this paper called Tree Matrix that takes a combination of treemap , matrix, and arc diagram to visualize compound graphs. Now compound graphs are networks with a hierarchical clustering defined on the nodes. Sounds similar however that hierarchical clustering that forms a tree, the leaves of the tree are the vertices of the network. (CLICK) In other words the intermediate nodes of this tree diagram are not consider in this visualization for the network connections up on top. Only the leaves are considered, and we want to consider those intermediate nodes in our application.

* Slide 7

Another area of research that is similar is for call graphs. IN this paper, Visualizing Dynamic Call Graphs, they created various node-link and radial diagrams to visualize time varying call graphs. Here you can see the node-link diagram on a timeline. The call graphs discussed and visualized in this paper are typically used for representing relationships between the subroutines of a computer program. And while the functional decomposition tree component of the data follows that structure the network component does not necessarily follow the edge representation of procedure f calls procedure g. Hence the edge type verbs in the network data.

* Slide 8

Additionally the Visualizing Hierarchical Social Networks paper, does research in networks in which certain nodes and group of nodes can be classified through a relation of precedence. In which the author uses cohesive blocking and k-core decomposition algorithms and then node/link tree map combination vis to visualize the hierarchical network. However for the paper’s context it is a single network with a single defined hierarchy while our data set has various trees of multiple hierarchies.

* Slide 9

Now I will be going into the design overall layout. The design can be broken up into 6 main components, and we will go into more detail about each one in a minute. But the 6 components are an overview vis, Tree/Icicle Detail Vis, Mini Hierarchy Vis, Min Intra Edges Vis, Mini Inter Edges Vis , and features slash toolbar.

* Slide 10

First is the overview vis. The main visualization used will be a directed chord dependency diagram. (CLICK) In which each tree has an arc and within each arc, summarizes the number of edges and where they got to. (CLICK) The color of the arcs and ribbons will be based on the tree type, in this case there are 4 tree type. (CLICK) The point of this vis is to summarize the network data to provide a basic overview of the all the trees at once. (CLICK) From this vis the user will able to detect outliers and patterns such as tree with largest number of outgoing or incoming edges or all tree of tree type x have a large amount of intra connections, which would be an arrow going back to itself. (CLICK)

This vis will have user interaction capabilities such as tooltip for more detail as well as zooming and panning. The user will also be able to double click a tree indicated by the arc to then see the next view, for example tree #2. As a result of selection a minimized form of the chord diagram will be put in the top corner to be used as context of the network data.

* Slide 11

After a user selects the tree a detail view will appear in which the goal is to show the hierarchical component of the tree and provide a detail view of the tree. (CLICK) IN the center we use a combined node link and icicle diagram in which the color of each icicle block is based on the node type attribute. (CLICK) As one of the main contributions to create this vis, an algorithm will have to be created. (CLICK) Firstly, it will have to minimize the leaf block size to leave room for a node. As icicle leaves can be extremely thin and hard to see otherwise, there would not be room for a node in the leaf rectangle of a regular icicle diagram. (CLICK) Additionally, the algorithm will center the node in the middle of the rectangle of the icicle tree because a normal tree would center the parent between the children. As most obvious at the 1st and second level, the parent node is closer to the left node rather be in the middle of the right and left node. (CLICK)And finally, it will scale the diagram to fit within the bounds.

Now one of the main reasons I wanted to show this combined form of a node link diagram and icicle is because the icicle will provide a hierarchical context while being able to explore inter and intra dependencies of the network, which we will see in a minute. And the nodes will provide a clear target for arrow drawings.

* Slide 12

When a user filters to see only intra dependency edges, in the tree/icicle diagram the hierarchical edges disappear and the intra edges are drawn. But the icicle rectangles are left in the background for hierarchical context. (CLICK) Besides the obvious purpose of showing the intra edges of the tree. This vis also preserves the hierarchy of the tree through the combined tree/icicle diagram. (CLICK)Additionally from this view a user can detect outliers and patterns. Such as nodes with most incoming edges, such as this selected node. (CLICK) And Patterns of node type connections such as a red node type tends to connect to another red node type.

* Slide 13

But to further explore the patterns or anomalies of these intra dependency edges we have a mini vis. (CLICK) This mini vis main goal is to provides quick summaries of node type connections and edge types of the intra dependencies. (CLICK)This vis will be a scrollable window that shows the total intra edges first, and then groups the edges together first by node type to node type and then by the edges type. For this example, there are total 13 intra edges. (CLICK) We also notice there are 3 edges for arrow to go from a red node to another red node. Then lets say 2 of those edges are of edge type 1 (CLICK) to produce this first diagram, and the other edged is of edge type 2 to produce the next diagram. This pattern follows for the other edges. This vis gives the user not only then an indication of what type of nodes the edges connect together but also lists the edge types of the tree. (CLICK)

Several instances of anomalies and patterns can be identified from this mini vis such as dealing with most edges types, patterns of node types connections, and edge type and node type correlation patterns. (CLICK)

There will be user interactions again such as tooltip provide more detail about the tree.

* Slide 14 Tree/icicle inter edges

When a user filters to only see inter dependency edges, similar to the intra edges, in the tree/icicle diagram (CLICK) the hierarchical edges disappear and the inter edges are drawn. But the icicle rectangles are left in the background for hierarchical context. (CLICK)The edges connect a arrow from the outside ring and a node in the tree.

Each arrow represents a tree in the dataset, mimicking the same position of the trees in the directed chord diagram. These can be used to navigate to another tree and also explore inter dependencies. (CLICK)

Besides the obvious purpose of showing the inter edges of the tree. This vis also preserves the hierarchy of the tree through the combined tree/icicle diagram. (CLICK) Additionally from this view a user can detect outliers and patterns. Such as arrows with most inter edges, nodes with most incoming edges, and node type patterns to certain tress.

* Slide 15 Hierarchy

When a user hovers over or selects one the arrow in the ring, a small icicle of that trees hierarchy will appear in the hierarchy block. (CLICK)For example hovering over the top left corner shows the icicle for that tree. This vis uses the same algorithm used for the center tree/icicle however there are no nodes or hierarchical edges due to size constraints. (CLICK)This will allow a quick and easy comparison of two trees, hierarchy, size, width, height and node types. With enough comparisons among trees then a user will be able to detect either anomalies or patterns such as hierarchy patterns among the trees, the tree with the most nodes, and the widest tree. (CLICK)

This vis will also be linked to the tree/icicle image as it highlights which nodes the inter dependencies are connecting to.

* Slide 16 Mini Inter Vis

To further explore the patterns or anomalies of these inter dependency edges we have another mini vis.

The vis is very similar to the intra except for one part. (CLICK) The is another level of grouping by the outside tree and edge is connecting to. In this example we are using the left diagonal arrow which is Tree Name # 2. Noticed there is a total of 4 edges touching the arrow. The edges go from the 2 red nodes to a green node, Then this outgoing edge goes from a red node from Tree Name#2 to the green node of the current tree to produce this diagram of 3 edges from red to green of edge type 3. An then finally we have an edge from purple to blue. (CLICK)

The anomalies and patterns that can be detected are similar to those described for intra dependencies but additionally can detect which trees the current tree is most connected to. (CLICK)

Similarly to the intra edges there will be a tooltip showing the name of the nodes but also the name of the starting and end trees to verify direction.

* Slide 17

In addition to the visualizations there will be a variety of user interaction features. Some include showing either hierarchical intra or inter edges. Filter or selecting edge type, node name, node type, or tree name. Also will have multiple selection from either lasso or box approach.