

Motivation and Audience:

The motivation behind this project is the Rebirth Analysis Tool which I am currently working on for ECSE Design. The idea behind the Rebirth Analysis Tool is to develop a web application which allows engineers to upload information about a product they are designing or plan to use to our site and view information about the sustainability of the product. The goal is to assist engineers in making sustainable designs.

My project for this class is one form of output for the tool. It is a connectivity map (CMap) which takes the parts and connections of a product then displays them in a graph. There are several types of connections which we currently have defined and plan on expanding the list over time. Each connection is assigned a value between 0 and 1 known as its Jointed Strength Index (JSI). The idea behind the graph is provide the user with a graphical layout of their product and easily identify nested parts, poor connections, and gain a better overall sense of the product. The audience for this project is anyone who would use the Rebirth Analysis Tool, such as engineers, students, or designers.

Visualization Design Evolution:

The idea for this project began at the beginning of this semester. My team for ECSE Design picked up the Sustainable Design project where the previous semester's team left off. We have several visualizations representing the output, one such was meant to be the connectivity map. The previous team actually developed a Bill of Materials graph (seen in Figure 1) which we decided to keep but it did not meet the criteria that were expected. The first, very poor, draft I created was just to get used to a new tool. I chose to recreate the bill of materials in a hierarchy using Google Chart Tools, seen in Figure 2. This design does not come close to the current RGraph from Jit that we use. It lacks interactivity and would quickly become unreadable for larger products. This design was quickly scrapped.

The second design I attempted was another version of the RGraph used for the BoM graph called a Weighted RGraph (Figure 3). This was a good step towards learning Javascript since I had never touched code like it before. It met the basic criteria required but lacked much the crucial information. The edges and nodes were all one color and the nodes were also a uniform shape. The only information it provided were the names of each node and what nodes they connected to. User interactivity was also lacking since users were required to click a node's name to re-center the graph around it in order to traverse the graph. For the third design, I implemented more interactivity. This allowed users to pan around by clicking and dragging the background as well as zoom in and out by scrolling on the mouse.

The fourth design incorporated colors to distinguish edge types, seen in Figure 4. I chose to represent each different connection type with a unique color. The downside to this is there are already 13 different connection types and we plan to expand which means it will become increasingly difficult to add new colors for each different connection type.

The fifth and final design for this semester is shown in Figure 5. After unanimous feedback to switch, I implemented a Force Directed Graph using Jit. The largest difference between a Forced Directed Graph and an RGraph is that Force Directed cannot be centered. Instead, the user can click and drag each node to place them wherever he or she chooses. The other enhancement was to color each edge based on the JSI. I chose a color gradient which ranges from red to green. A 0 JSI will paint the edge red while 1 will paint the edge green and I change the color for every step of 0.1 JSI. When a user clicks a node's name, the node and all of its connections become enlarged and change color depending on the JSI. Edges that are not highlighted will remain thin and blue. I chose to change the background color to white in order to match the color scheme of the site which made choosing black for the text and obvious choice.

The color of each node is still up for debate but I chose a dark grey to provide a strong contrast to the background and still allow for the text to be readable.

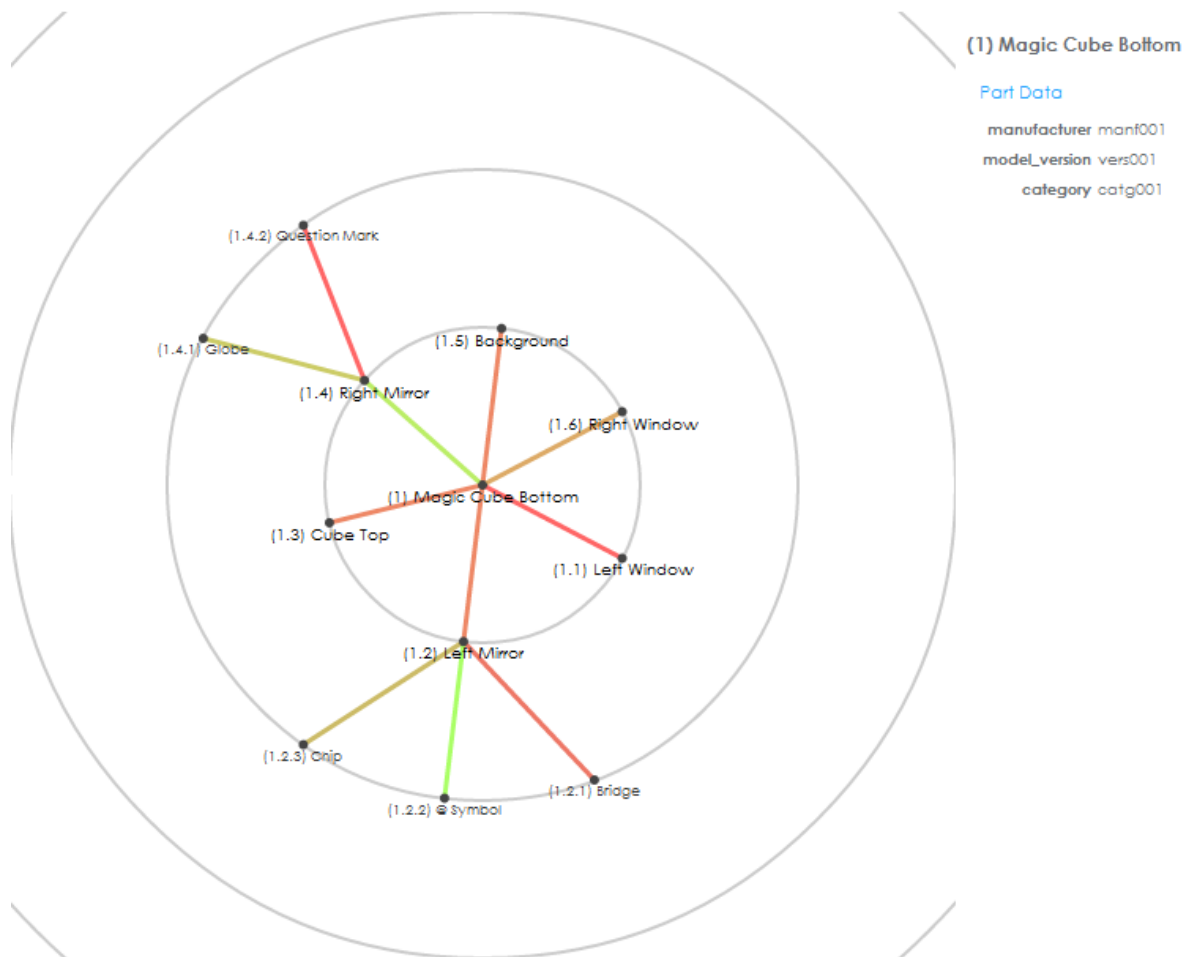


Figure 1 - Bill of Materials graph

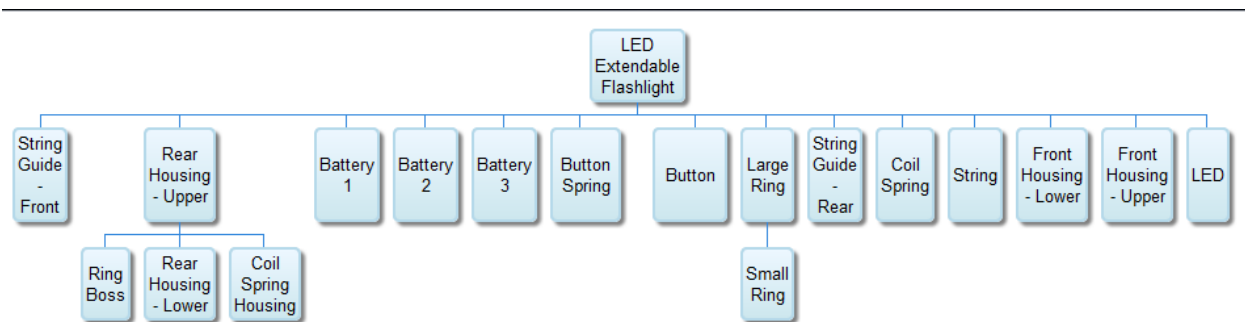


Figure 2 - Bill of Materials hierarchy

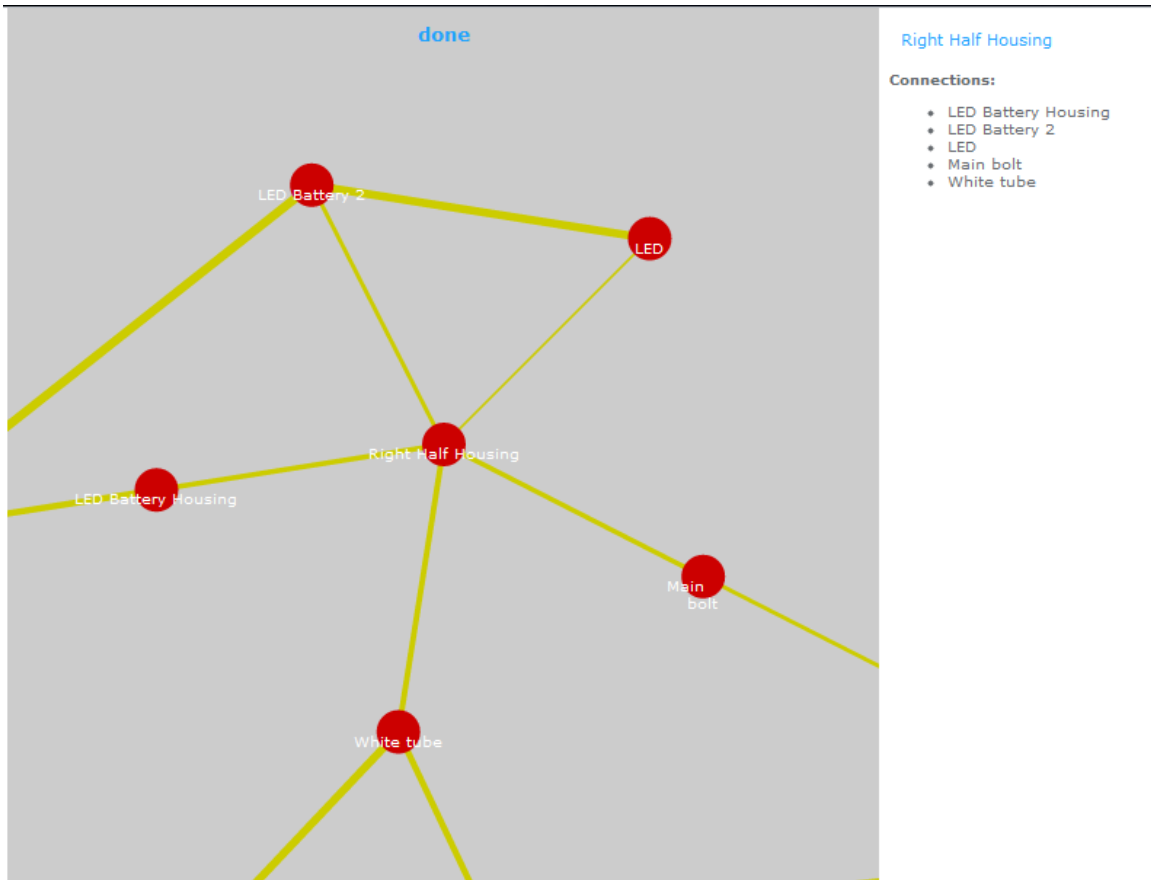


Figure 3 - First Connectivity Map

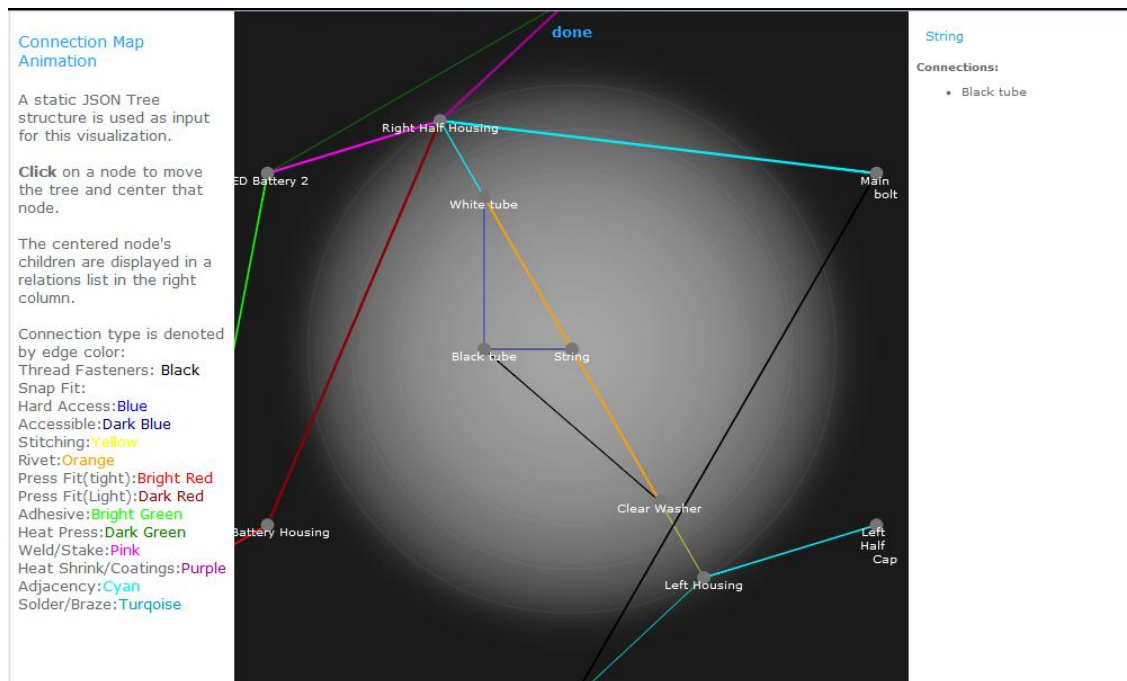


Figure 4 - Colored CMap

Connection Map

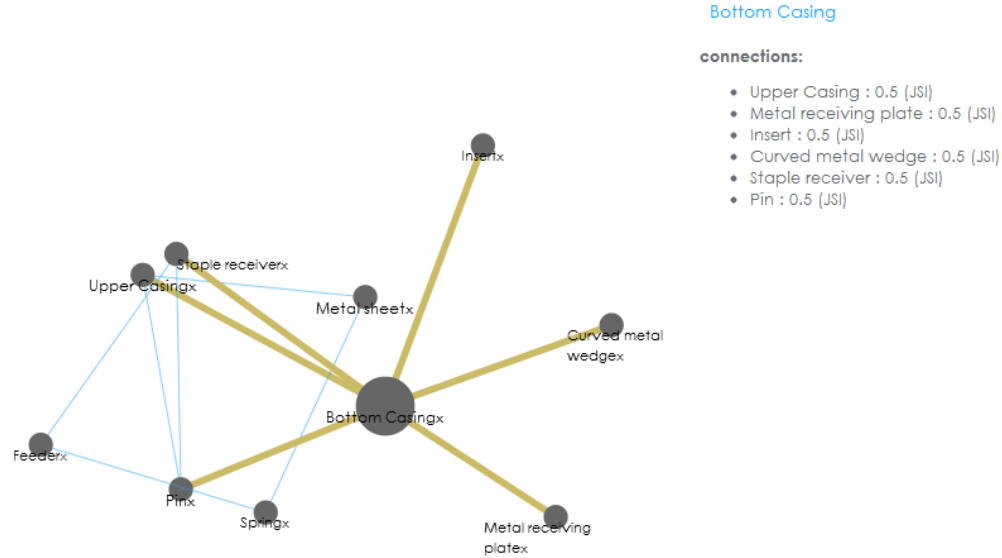


Figure 5 - Final Design

Feedback:

The strongest piece of feedback I received was to use the Force Directed Graph. Everyone that reviewed this graph agreed that the Force Directed Graph was best. Half of the people who provided feedback, those in this class, were not the intended audience but still provided excellent opinions while the other half, the members of my design team and project manager, gave much less feedback. Other pieces of advice received were to remove the focus area or make it resizable, choose a better color scheme, add tooltips to edges to provide more information, add the ability to filter components, select multiple points simultaneously, and fade nodes that are not directly connected to the part selected. I incorporated as many of these as possible. First and foremost, I switched to the Force Directed Graph which I feel allowed for the best changes. I also integrated it with the site so that the graph is now dynamically generated. The color scheme was changed so that it is more readable and will hopefully appease everyone. I looked into adding Tooltips to pop up while hovering the mouse over an edge but everywhere I checked stated that it was not supported for edges. Filtering components is still not implemented, although I did add an x next to each node's name which will remove it from the graph until the page is reloaded. Highlighting a node should resolve the issue about fading the unimportant nodes. I did not look into selecting multiple and I do not know if it would be beneficial to implement this.

Core features and Technical Implementation:

The data used for this are the datatables for products, parts, and connections in the database for the Rebirth Analysis Tool. The graph generates dynamically depending on the parts and connections specified by the user. The products currently on the site are mock products, used as tests and do not accurately represent a finished product.

The toolkit used to implement this visualization was the Javascript Infovis Toolkit (Jit) in alongside Ruby on Rails which was used to create the website. Jit is a powerful toolkit which provides templates for drawing several different types of visualizations and allows for a great deal of freedom. The information about drawing the graph is stored in a static .js file on our server and the ruby code is used to generate json input data which is then passed to the visualization dynamically.

The core features of the graph include:

1. Connection between all parts within a given product
 - a. Automated process
2. Interactivity
 - a. when node name is clicked highlight node and all connections
 - b. Zooming (Mouse Wheel) and Panning (Click & Drag) integrated
3. Information
 - a. Provides all necessary information
 - i. Connection Type
 - ii. Number of connections
 - iii. Parts connected
 - iv. IMPROVEMENT: Highlight or mark modules / assemblies
 1. Assemblies/modules will cluster on the graph but they are not directly marked.

Time Spent:

I spent the largest portion of time writing the ruby which dynamically generates the JSON data. Once the data was complete I changed the color schemes, adding different colors to edges depending on their corresponding JSI values, and polishing it overall so that it fit in with the site. The areas I spent most time in were Visualization Execution and Analysis & Validation. My motivation was clearly defined since I am currently working on the Rebirth Analysis Tool and this is a feature that has been sought after for some time with little effort put into it. Data Collection was also simple since I have access to a database full of mock objects to test with.

Approximate time spent this week:

0.0 hrs Motivation & Problem Definition

1.0 hrs Visualization Design

0.0 hrs Data Collection

5.0 hrs Visualization Execution

3.0 hrs Analysis & Validation

2.0 hrs Visualization Revision

1.0 hrs Presentation

4.0 hrs Total time spent

TOTAL time spent this semester:

1.0 hrs Motivation & Problem Definition

7.0 hrs Visualization Design

4.5 hrs Data Collection

16.5 hrs Visualization Execution

14.0 hrs Analysis & Validation

8.0 hrs Visualization Revision

2.0 hrs Presentation

53.0 hrs Total time spent

Bibliography:

1. Morris, Jeff, Mark Anderson, and Senior Capstone Students. "Sustainability Wiki."
[Http://mdl-vm2.eng.rpi.edu/wiki/index.php/Main_Page](http://mdl-vm2.eng.rpi.edu/wiki/index.php/Main_Page). Web. 10 Apr. 2012.
2. "JavaScript InfoVis Toolkit." *Demos* -. Web. 23 Apr. 2012. <<http://thejit.org/demos/>>.