

TreeVersity: Visualizing hierarchal data for value with topology changes

John Alexis GUERRA GÓMEZ, Audra BUCK-COLEMAN, Catherine PLAISANT,
and Ben SHNEIDERMAN

University of Maryland

Abstract

As the world becomes more complex and plagued by wicked problems, there is an increased need for more multifaceted visual frameworks to untangle and clarify global issues. Budgets, census outcomes, stock market results, and other statistical data are best understood and explored through comparison, however to date there has been no effective visual solution framework for assessing a variety of hierarchal data sets. TreeVersity, a new interactive framework, is changing this.

TreeVersity facilitates visual comparisons for hierarchical data, revealing trends, significant changes and other data revelations through multiple tree visualizations. The framework enables comparison within small, 10-node data sets to those with as many as 1,000,000 data points or more; it can reveal both overall structural changes such as the addition and removal of nodes and the differences in the node attributes including absolute and relative changes.

TreeVersity is a user-centric program that delivers control of information to its users, allowing them to select, prioritize, recast and narrate the story they want data sets to tell. It offers customization of color ranges as well as the expansion or deletion of visual nodes based upon a user's desired data-significance range. Users can quickly discover relevant information through framework interaction that includes real-time highlighting of hyperlinked relevant nodes and branches. TreeVersity offers both macro and micro views of comparison data through the dynamic progressive exposure of information, as dictated by the user.

The project has progressed significantly, and TreeVersity collaborators will continue to develop the framework. In this paper, we will present TreeVersity through data visualizations as well as outline the visual system's in-progress expansion possibilities.

Keywords: *information visualization, tree comparisons*

Introduction

In this paper we introduce TreeVersity, an interactive tool initiated and implemented by John Alexis Guerra Gómez, that visualizes and enables comparisons of hierarchical multivariate data. What distinguishes TreeVersity from other visualization tools is its ability to compare hierarchical datasets with topological and node value differences. It allows users to explore, to integrate and to visualize small to enormous amounts of in a single visually coherent space.

As the world continues on its path of increasing complexity, the impact upon designers and information visualization is direct: it increases the challenge to reflect these compounded trends aptly. With wicked problems and seemingly infinite and contradicting amounts of information, stakeholders need clear, effective, and efficient tools to better understand the world today and make decisions for its future. This means changing the way information visualization both renders and addresses these problems. Instead of starting with the traditional steps of first formulating a question to a well-defined problem, then researching and collecting data, and concluding with a visual solution (Shapiro, 2010: 18), designers should move the first step to the last, starting with information and visualizations to seek the most relevant questions.

As the design practice shifts from being artifact-focused to service-focused, it is important to realign design thinking not just to problem solving but visualizing the complex and wicked problems, which do not have simple—if any—solutions. Budgets, census outcomes, stock market results, and other statistical data are best understood and explored through comparison, however to date there has been no effective visual solution framework for assessing a variety of hierarchal data sets. “In a world that is perceived and understood as complex the design ability of render and describe emerges as an answer to the need of orientation and search.” (Scagnetti, 2007: 4) TreeVersity is addressing this through its interactive framework that visualizes hierarchal data using trees.

Defining Data Differences

Information can be visualized in a variety of ways depending upon how the data is structured, but not all data types are applicable to different visualizations. Within the computer science arena there are seven commonly agreed upon groupings of data (Shneiderman, 1996): one dimensional linear data such as the number of students in a class, two dimensional data such as a map that denotes place and distance, three dimensional data such as that of a physical space, multidimensional data that contains any number of values or information categories, such as the different ways we might explore the demographics of a large crowd attending a sporting event, temporal data such as timelines, networked data that has interconnected nodes of information such as social networking graphs, and finally, hierarchal data that have generational relationships that do not intersect across data sets, such as a visual display of a family tree. It is this last category, also referred to as “trees,” that TreeVersity seeks to visualize. (Figure 1.)

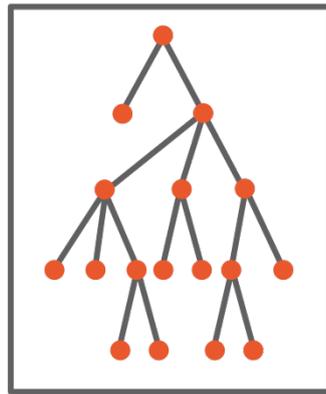


Figure 1: An example of a tree hierarchal display.

Source: Guerra Gómez, J.A. (2012)

Most data can be viewed through a tree perspective, as long as there is a parental hierarchy present and the data sets do not crisscross each other. Budgets, census reports, medical and sports data, and stock market closings are but a few of the different types of information that can be presented as a tree. Trees can be used to represent simple and complex amounts of information, however the more nodes or data points on the tree and the more branches or divisions of the tree, the more complicated and potentially muddled the visual.

Within a tree structure, there are five ways the data might change within multiple tree views. (Figure 2.) These differences are important when comparing tree structures. One way to understand the disparities is by using two examples: the US. Federal Budget organized by Agencies and Bureaus and the organizational chart of a company with a hierarchy of each employee her boss and dependents and their salaries.

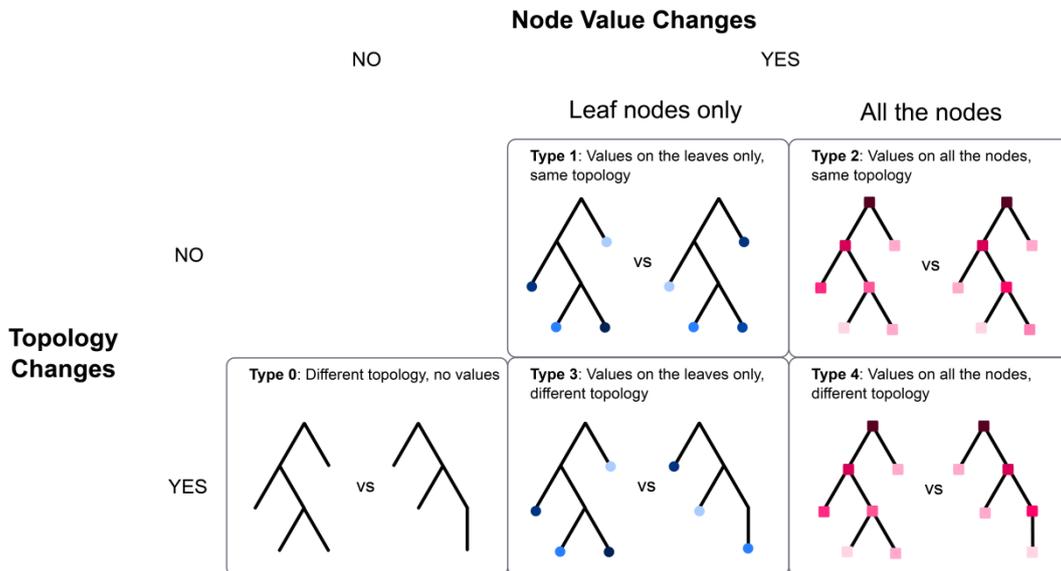


Figure 2: Types of tree comparison problems handled by TreeVersity

Source: Guerra Gómez, J.A. (2012)

Type 0: Changes in topology with no values on the nodes. An example of this would be when we compare the difference between what divisions and departments are allotted money between two years' budget. The monetary amounts (node values) are not included in this simple data comparison.

Type 1: Changes in the leaf node's value without changes in topology. An example of this would be when we compare the difference between two years of the budget, assuming that no departments were added or deleted or moved between the two years.

Type 2: Changes in the leaves and interior nodes' values, without changes in topology. An example of this would be comparing the changes in salaries of the employees of a company, group by the organizational chart, where no new employees were hired and no lay-offs were made.

Type 3: Changes in leaf node's values with changes in topology. This comparison of the Federal Budget has both changes in each department's budgets, as well as the addition and deletion of different departments. Here the tree branch structures vary as well as the value of the leaf nodes.

Type 4: Changes in leaves and interior nodes' values with changes in topology. An example of this would be again the comparison of the changes of the salaries of a company's employees group by the organizational chart, but considering also newly hired, moved and laid off employees.

Icicle, sunburst, Treemaps, hyperbolic and other single-tree visualizations have been extensively researched and published, as related by Shroeder *et al* in 2001; Graham and Kennedy in 2009; and Jürgensmann and Schultz in 2010. Multi-node, multi-tree tree comparisons become exponentially more complicated and time consuming as the data to visualize, compare and analyze increases. These visual solutions have been explored less and yet are central to visualizing today's complex issues and information needs.

Typically the research conducted on visualizing differences between tree structures has focused on Types 1 and 2, with some attempts at Type 3, but to the best of our knowledge, none of these solutions addresses the problem of comparing changes in node values (Types 2 and 4). TreeVersity, however, can visualize not only Types 1 and 3, but also Types 2 and 4. This makes the framework applicable to more complex sets of data sets.

Information Visualization Design Criteria

When setting out to design an information visualization framework, it can be difficult to know where to begin. Fry (2008) poses seven stages for the creation of information visualization: Acquire, Parse, Filter, Mine, Represent, Refine, and Interact. The second phase, Parse, calls upon designers to decide which information within the data is most relevant and thus will be the focus of the visualization. This model best fits problem-specific visualizations as prioritizing one set of information can exclude the visualization of another. To design a visualization tool that anticipates a variety of parsing needs both known and unknown is inherently more complex. This is the mission of TreeVersity.

Shneiderman (1996) describes the Visual Information Seeking Mantra, seven user-tasks that designers need to anticipate: Overview, Zoom, Filter, Details-on-demand, Relate, History, and Extract. Along with these seven tasks, the designer needs to create an aesthetic harmony of the overall visual elements. Typography, layout, animation, toggle views of information, and color choices all contribute to a framework's success. Further, when analyzing data with 1,000,000 or more points of information, the need to manage and prioritize displays while also realizing Occam's razor is fundamental. Considerations about scalability of information, the need to keep all of the data present in one screen view versus panning, partial views or multi-screen views, and animated displays of information all play into the design decisions. Finding balance between these factors is

essential. The tool's efficacy has a direct relationship on the user's ability to make summations and decisions about the data (Wright, 2009).

Related work

Single-tree visualizations are incredibly diverse as evidenced by Jürgensmann and Schultz (2010), and multi-tree comparisons also offer divergent visual solutions of data with similar change constraints. These solutions visualize differences through typography, color, animation or interaction emphasis or a combination within. With these areas of emphasis are also varying types of data comparisons: topological changes and node value changes. The following is a brief summary of the most notable visual tree comparison solutions¹. Many of these forthcoming examples fall under the “form follows function” philosophy where attention to aesthetics is notably deficient.

Typographic emphasis

TreeJuxtaposer by Munzer *et al* (2003) designed a highly functional tree comparison that is lacking in aesthetic consideration. TreeJuxtaposer, an interactive tool, visualizes topological changes through columns of text and coordinating colors for related data points. The utility is much more engaging than the aesthetic, which incorporates a single-weight and point size of a Chicago-like font and masses of edge-drawn connecting gray lines. Admittedly, creating hierarchy within a million-point tree comparison is more complicated in a platform with an automatic layout feature, and the use of the section colors to identify related data points is useful, but the screen quickly becomes cluttered due to the absence of a typographic and visual hierarchy. The overlay of type on the color sections further dilutes the TreeJuxtaposer's perceptual effectiveness.

Furnas *et al.* (1994) and Lee *et al.*'s CandidTree (2007), focus on certainties and uncertainties within node link relationships. Here columns of text are color-coded to represent the different data relationships. Connecting the text between the columns are solid and dotted lines to represent either certain or uncertain relationships between the data. The result is visually appealing with its ample spacing between data generations and the framework can compare thousands of trees, however there is no quantitative visual relationship between the data. Further, the combination of visuals plus text is more effective than using either alone (Faraday and Sutcliffe, 1997), and the above-mentioned examples could benefit from such inclusion.

Color emphasis

Graham *et al.*'s (2003) Icicle-like representation is another strong visual solution that uses color to show the value and relationships between node values. This visual tool uses rows of colored squares and divides the screen space either horizontally or vertically to visualize different hierarchies within trees. The split screen weakens overall visual comparisons, but Graham addresses that through an aggregation view in later work. The Icicle's use of color aids in preattentive processing (Ware, 2004) and additional label and textual information can be revealed through interaction.

The treemap idea (Shneiderman, 1992) and its first implementation (Johnson & Shneiderman, 1991) is similar to the icicle representation, however it adjusts both the horizontal and vertical space to define value, as opposed to the Icicle's one direction. One rectangle shape is proportionally subdivided into smaller square or rectangle

¹ Algorithmically oriented comparisons are a category of tree comparisons, however since they are mathematical rather than visual, they will not be addressed in this paper.

shapes. These smaller shapes are located within the larger shape and color-coded to show visual relationship. The visual result is informative and functional. TreeMap had a feature that displayed changing values on the hierarchy but it was never developed for comparison.

SmartMoney's Map of the Market (Wattenberg, 1999) uses the TreeMap principles to visualize the changes in stock market values from one day to the next. A large rectangular space is divided into different sections (Health Care, Financial, Energy, etc.). These sections are then further proportionally divided into smaller rectangles representing the different companies categorized within the sections. Each stock's rectilinear space, which represents its market capitalization, is then filled with a shade of green or red to identify either the increase or decrease in value since the day prior. This is a visually effective tool for a specific tree problem—Type 1 as mentioned above—but it cannot render other data differences. Additionally comparing node values isn't as rich as comparing tree topologies.

Finally, Contrast TreeMaps (Tu and Shen, 2007) represents the only project that, to the best of our knowledge, combines some topological differences and changes in node values. Based on the traditional TreeMap, Contrast Treemap also divides each sub rectilinear shape into two triangles to represent value changes and structural differences. The color palette is especially problematic as Contrast TreeMap uses mostly dark, analogous colors to denote changes. Scattered across the Contrast TreeMap are thinly stroked circles ringing specific rectilinear shapes. The circles do not have enough visual weight for easy visual detection, lessening the significance of the data points they intend to highlight. The visual result is a mass of uninviting darkly colored, ill-defined triangular and rectilinear shapes.

Animation or interaction emphasis

Ghoniem and Fekete's Animated TreeMaps (2001) represents changes in two trees by animating these changes. The rectilinear spaces of one TreeMap shrink or expand relative to the node value changes. The animation is visually engaging but not highly functional as it requires the user to remember the previous values as the compared values are displayed. With large data displays, this can be especially challenging and confusing. Further, the multiple vividly colored boxes strain the eye as the electric hues simultaneously compete for visual attention.

Plaisant *et al.*'s SpaceTree (1998) tree browser offers dynamic rescaling of tree branches, making optimal use of screen space. Space Tree uses lines to vertically or horizontally connect generations of information, confined by rectangles. Gray triangles attached to the rectangles signify that more data has been collapsed and out of view. With a click of the user's mouse, the triangle turns and the targeted data is revealed through animation. Other branches are rescaled to allow viewing of the desired data. Although its aesthetic is lackluster with its gray tones, standard box shapes and default typography, SpaceTree is highly functional tool. Parr *et al.*'s DoubleTree (2004) expands upon the SpaceTree browser to show two animated tree views within one overall screen. There are no significant changes to the program's aesthetics, and thus it is subject to the same visual design critique as the original SpaceTree.

Heer and Card's Degree-of-Interest Tree (2004) is an interactive work that reveals information relationships as determined by the user. The initial visual resembles the SpaceTree with its horizontal distribution of textual information across a series of columns. By selecting different words, the user initiates animated expansions that reveal the related data. Thin gray lines are drawn connecting textual information from one vertical column to the next. The information is displayed in black and gray tones. Key

connecting words are highlighted in coral or light blue. Gray triangles indicate where additional data has been collapsed to better manage the amount of data visible, especially when comparing hundreds of thousands of nodes. The work is functional and makes effective use of space and interactivity but doesn't allow any type of comparison.

Lastly, Graham and Kennedy's iterative work, now entitled TaxVis (2011), is the most comprehensive tree comparison platform to date. It offers multiform visualizations of the same data including TreeMap-like view, Directed Acrylic Graph (DAG) view, Icicle representations and text-based lists. All visualizations employ a color palette of muted brown and yellow gradients. A light cyan is used for highlighting. This variety offers the user to fully comprehend the information at hand and also threatens to overcomplicate the tool. The muted browns and yellows are offset by a light cyan that is used for highlighting and a bold and roman fonts are incorporated for further hierarchy.

Visualizing differences within data sets, especially those with thousands or more data points, is complicated. As Graham and Kennedy (2008) acknowledge, providing all necessary information within one view is difficult if not impossible. Comprehensible tree searches necessitate multiple views and/or multiform views of information. The related works were successful in some areas of comparison, but to the best of our knowledge, none of these solutions compares changes in node values. TreeVersity, however, can address all of the topological and node value changes as described in Figure 2 including node value variables.

Defining TreeVersity

The ability to create and design a successful visualizations requires a plurality of skills and knowledge sets: concepts of usability, user interface design, statistical methods, keen understanding of visual principles, and proficient coding and framework-building know-how. TreeVersity has been a collaborative effort led by PhD computer science candidate (Guerra Gómez), two computer science faculty advisors (Plaisant and Shneiderman) and a design faculty member (Buck-Coleman). The members' areas of specialty have kept simultaneous focus on the functionality and aesthetics.

Through TreeVersity we seek to develop a single comparison framework that can visualize all five tree changes as described in Figure 2 while also being aesthetically pleasing. To cover a broader spectrum of problems, TreeVersity is designed to support the following types of differences: created and removed nodes; absolute and relative differences of the node attributes values; polarity of the differences; changes only on the leaf nodes of the tree, or changes in all the nodes of the tree (including interior nodes); and amount of change compared with all the other nodes on the tree (or compared with the siblings). To do this, the framework can visualize positive and negative changes in direction, absolute change, percentage change, and relative node changes. Labels further clarify the on-screen information and applied colors aid in indicating data value.

This diverse range of visualizations enables users to find and examine pertinent data comparisons. For example, accountants might want to review changes between annual budgets, a transportation manager might want to view the number of traffic accidents happening on different roads under different weather conditions, and a biologist might want to create a taxonomy of species that includes extinct and newly existing creatures.

TreeVersity incorporates three different modes of visual comparison: side-by-side views, an aggregate overall view and animation that shows changes morphing over time and visual relationships within the different data sets, as shown in Figure 3. Further, it allows macro-to-micro views of data, comparing nodes of information across multiple sets of information and with variability in that information. Edward Tufte identified the strength of

this: “Micro/macro designs enforce both local and global comparisons, and at the same time, avoid the disruption of context switching. All told, exactly what is needed for reasoning about information.” (1990: 50) Information becomes knowledge that aids in the decision-making process, helps decision makers control the information to seek answers to the questions that best serve them.

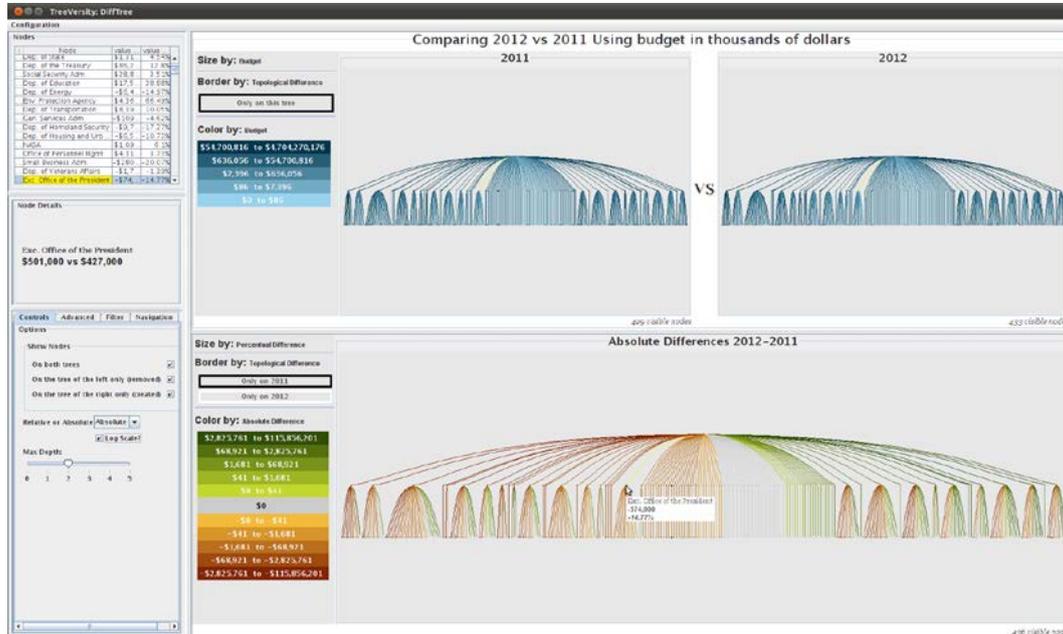


Figure 3: An overall view of TreeVersity. On the top are the two original trees being compared (budgets for 2011 and 2012). At the bottom the DiffTree shows the amount of change for each node.

Source: Guerra Gómez, J.A. (2012)

Layout

As with any multi-view interface, maximizing screen space is crucial. TreeVersity divides this space into four sections. (Figure 3) On the left is a vertical column that lists the data being compared and filtering controls. The two subsets at the center top show aggregated overall views of the different data sets. The remaining screen area is allotted to the DiffTree, a consensus tree-like feature that shows the differences between the compared data. When first viewed, the DiffTree contains all of the node differences from both original trees. Users can then view desired nodes through the filtering options, described below.

The different layout components allow users to make side-by-side comparison of the original data while simultaneously visualizing the data differences. Nodes are dually encoded through size and color choices to represent data amount variations. When viewing thousand-node-or-more data sets, these views can quickly become cluttered. To alleviate this, the last three mentioned visualizations all offer panning and zooming options for navigation to keep the data manageable. Further, TreeVersity distributes the visualized nodes to fit the screen. This provides users with optimal views with available screen space.

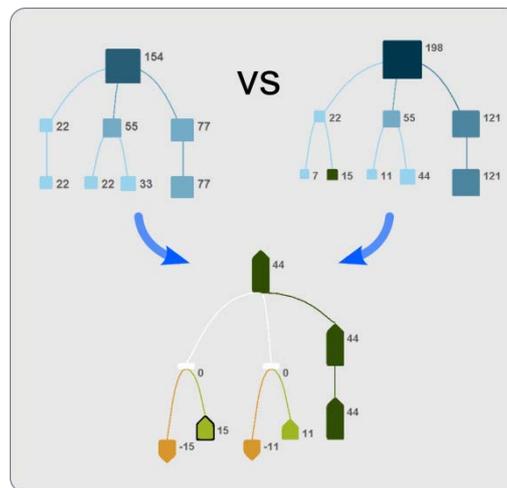


Figure 4: The DiffTree. The yellow and green arrow-like shapes at the bottom are Bullets, which represent node differentiations between the two original trees, shown in blue.

Source: Guerra Gómez, J.A. (2012)

Symbols

TreeVeristy's DiffTree employs a customized glyph, the Bullet (Figure 4), to represent the cardinality of change, the amount of change, and the creation or deletion of nodes. The shape's direction represents a positive or negative change of the compared data: left for negative and right for positive in the horizontal layout and down for negative and up for positive in the vertical layout. The Bullet's proportional size ranges represent the differences in the node values. The colors in the DiffTree are altered from those in the original trees because the DiffTree nodes usually represent different value ranges.

Labels and colors

TreeVeristy offers multiple options to control textual information and colors of the visualizations. Users can reveal or collapse the amount of information that is represented about the data, such as name of the node, its value, relative values, and other descriptions. Users can also choose to display the node's values and other descriptive information as an adjacent label or hide it completely. TreeVeristy maximizes spatial considerations when displaying the nodes and their corresponding labels.

Users also choose which of the pre-established color gradients will be applied to which information. Color is used to encode both the cardinality and amount of change in the nodes. The lighter colors are automatically applied to smaller data values and the darker colors are applied to larger values. This gives consistent visual weight to the more substantial data amounts and allows for easy scanning to find significant nodes. In the DiffTree, white rectangles represent nodes where the amount of change is zero. White borders ring the Bullet to denote newly created nodes and black borders signify removed nodes.



Figure 5: This view shows a filtered view where the user has filtered the data.
 Source: Guerra Gómez, J.A. (2012)

Interactivity

After looking at the overview, a user might choose to focus on a subtree comparison. This is achieved by double clicking on the root node of the sub-tree of interest. (Figure 5) After navigating into a sub-tree, all the views will be updated to display only the nodes on it, prioritizing the on-screen information. The nodes are sorted according to the amount of change (absolute or relative). The DiffTree is sorted first, and then the same order is applied to the original trees. A navigation panel records currently navigated nodes and allows users to return to a previously navigated node. TreeVersity also includes yellow link brushing to help users visualize related data within the three tree views. (Figure 5) As the user mouses over the nodes in the DiffTree, for example, the corresponding nodes and links are temporarily highlighted in the two original trees above.

Users can filter the nodes viewed by value ranges, topological change, and maximum depth. The value ranges option allows users to view only the nodes that fall within a specified range, using absolute or relative amounts of change. Topological change allows users to see only the nodes that were created, or removed, or that are present on both trees. Finally, the filter by maximum depth hides all the nodes that are deeper than a specified maximum depth. After the filtering operation, any nodes that do not fit the new selection criteria will be hidden from all the views (including the table), and then by an animation, the empty space will be reclaimed for the remaining nodes, making optimal use of screen viewing space. TreeVersity also includes a scatter plot option for filtering data. Only the nodes a user selects from the scatter plot will be displayed in the two original trees and the DiffTree. This gives users a visual-to-visual option for editing representation, as opposed to the mathematical or other filtering criteria. All of TreeVersity's navigation and filtering options are available for the newly composed subtree.

Applications

To demonstrate TreeVersity's functionalities, researchers compared the 2011 and 2012 U.S. Federal Budgets². This tree is composed of five levels: Agencies, Bureaus, Accounts, Sub-functions and the Type of Account (if it is Discretionary or Mandatory). The values are displayed in thousands of dollars. These are large data sets that need filtering

² The U.S. Federal Budget is available at www.whitehouse.gov. The negative budget values were not included in the analysis to avoid user confusion about the concept of negative budget values. The above-mentioned budget node numbers were calculated with this exclusion.

to comprehend the information at hand. For example, the 2011 budget is a tree of 4,103 nodes, and 2012 has 4,024 nodes. (Figure 3) The shifts in colors from reds to greens in the DiffTree indicate potentially significant changes occurred throughout the various areas between the two years.

From there we wanted to find the largest changes overall. (Figure 6) For this we selected the tabular view and filtered by the amount of relative change. We configured the to show the relative change of each node between the two years. Because of this, the color and ordering views of the nodes follows the perceptual value. After the ranking, the Sub-function "Pollution control and abatement" and its Account "State and Tribal Assistance Grants" rose to the top with an unbeatable increase of 473,500% (their budget went from \$1-million dollars to \$4.73-billion). The node value increases from \$1,000,000 in 2011 to \$4,736,000,000 in 2012, a relative increase of 473%. This significant variation could easily be overlooked when browsing this very large tree. This node's value difference is seen also related in its significant size and color differences as compared to the other nodes.

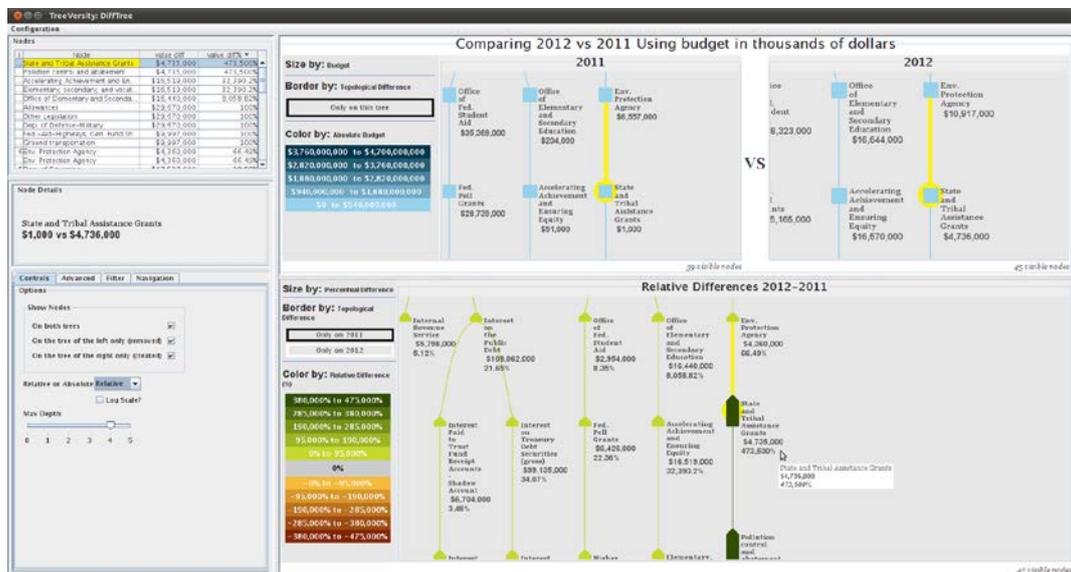


Figure 6: By ordering the table by relative change users can find the nodes with largest relative increase or decrease. Clicking on a row centers the display on the corresponding node. Here the largest relative increase is for "Pollution control and abatement".

Source: Guerra Gómez, J.A. (2012)

User Study

Earlier in the development of TreeVersity, we conducted a user study with 8 participants to evaluate if users could understand the framework's visual encodings and interface organization. The participants had no training on TreeVersity and were unaware of its existence prior to the study. We used two years of a fictitious budget for comparison. The dataset had 46 nodes distributed over four levels with 1 node removed from the first year (2011) and 3 nodes created in the second year (2012). We included multiple spikes in data that could be easily spotted by the users (if they could interpret the encodings accurately).

After a one-minute introduction of TreeVersity, participants were asked to engage and describe their perceptions of the framework and the meaning of the information relayed in the visualizations comparisons. Without any training all participants grasped and correctly interpreted most of interface components of TreeVersity. All of the participants correctly described the side-by-side trees and the DiffTree. They also accurately described the

relationship between the three trees. Participants addressed their interpretations of the glyphs. For the side-by-side trees, everyone immediately associated color with the amount of money at each node, however some people overlooked the size property (both color and size encoded the same information). Several people had problems understanding the color legend, and we have made modifications to resolve those issues.

The participants were asked to rate three different aspects of TreeVersity's usefulness on a scale of 1 (not useful) through 7 (very useful). All of the responses were 5 or above, and six out of the eight participants answered the three questions with 6s and 7s.

Conclusion

TreeVersity is a flexible, in-depth tool framework that visualizes topological and node value changes between hierarchical data. It can visualize differences in created and removed nodes; absolute and relative differences of the node attributes values; polarity of the differences; changes only on the leaf nodes of the tree, or changes in all the nodes of the tree (including interior nodes); and amount of change compared with all the other nodes on the tree (or compared with the siblings). Further, the framework can visualize positive and negative changes in direction, absolute change, percentage change, and relative node changes. Labels further clarify the on-screen information and applied colors aid in indicating data value. This diverse range of visualizations means that users can find and examine the data comparisons at hand.

The tool allows users to control visualizing, editing and filtering targeted information. TreeVersity can generate content relationships not easily discovered/understood from other data representations, such as an Excel spreadsheet. To date, TreeVersity has only applied to dual tree comparisons. Future work will develop comparisons between multiple trees of large data sets. Further, the framework includes important design considerations that make it both visually appealing and highly functional.

Acknowledgements

We thank the Fulbright International Science and Technology Scholarship for supporting John's doctoral studies as well as the Center for Integrated Transportation Systems Management (a Tier 1 Transportation Center at the University of Maryland) for partial support of this research. We also thank the Center for Advanced Transportation Technology Laboratory (CATT LAB), Michael Pack, Michael VanDaniker and Tom Jacobs for their suggestions and feedback.

References

- Almendra, R. and Christiaans, H. (2009). Improving Design Processes through better Decision-Making: an experiment with a decision making support tool. IASDR 09, Seoul.
- Amenta, N. and Klingner, J. (2002). Case study: Visualizing sets of evolutionary trees. Information Visualization. INFOVIS 2002. IEEE Symposium, (pp. 71–74). Boston, MA. USA
- Arvola, M. (2010). Interaction designers' conceptions of design quality for interactive artifacts. Proceedings of Design Research Society (DRS) 2010, Montreal, Canada.
- Card, S., Mackinlay, J., and Shneiderman, B. (1999). Readings in Information Visualization: Using Vision to Think. San Francisco, CA: Morgan Kaufmann Publishing.
- Caviglia, G., Valsecchi, F., Ciuccarelli, P., and Ricci, D. (2010). A Didactical Framework to Experiment the Potential of Visual Languages in Engaging Social Complexity. In DRS2010 - Design & Complexity proceedings, eds. David Durling, Rabah Bousbaci, Lin-Lin Chen, Philippe Gauthier, Tiiu Poldma, Seymour Rowoth-Stokes and Erik Stolterman. Montreal, Canada.
- Fallman, D. (2008). The Interaction Design Research Triangle of Design Practice, Design Studies, and Design Exploration. *Design Issues*, 24(3), Summer 2008.

- Faraday, P. and Sutcliffe, A. (1997). Designing effective multimedia presentations. SIGCHI Conference on Human Factors in Computing Systems, (pp. 272-278). Atlanta, GA., USA
- Faraday, P and Sutcliffe, A. (1999). "Authoring animated Web pages using 'contact points,' " SIGCHI conference on Human factors in computing systems: the CHI is the limit. (pp. 458-465). Pittsburgh, PA. USA.
- Fisher, D., Dhamija, R., and Hearst, M. (2001) Animated exploration of dynamic graphs with radial layout. IEEE Symposium on Information Visualization. (pp. 43 –50). San Diego, CA., USA.
- Fry, B. (2008). Visualizing Data. Sebastopol, CA: O'Reilly Media, Inc.
- Furnas, G.W. and Zacks, J. (1994) Multitrees: enriching and reusing hierarchical structure. SIGCHI conference on Human factors in computing systems: Celebrating interdependence. (pp. 330–336). Boston, MA., USA. ACM ID: 191778.
- Graham, M. and Kennedy, J. (2001). Combining linking and focusing techniques for a multiple hierarchy visualisation. Fifth International Conference on Information Visualisation. (pp. 425-432). London, UK.
- Graham, M. and Kennedy, J. (2005). Extending taxonomic visualisation to incorporate synonymy and structural markers. *Information Visualization*, 4(3), pp.206-223.
- Graham, M. and Kennedy, J. (2007). Exploring multiple trees through DAG representations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1294-1301.
- Graham, M. and Kennedy, J. (2008). Multiform Views of Multiple Trees. 2008 IEEE Information Visualisation International Conference. London, UK. 252 - 257.
- Graham, M. and Kennedy, J. (2009). A survey of multiple tree visualisation. *Information Visualization*, 9(4), 1–18.
- Graham, M. (2011). TaxVis - Visualising Multiple Overlapping Linnean Taxonomies. Available at: http://www.soc.napier.ac.uk/~cs22/projectBlog2/?w=my_weblog [Accessed June 8, 2011].
- Ghoniem, M. and Fekete, _J. D. (2001). Animating treemaps. 18th HCIL Symposium-Workshop on Treemap Implementations and Applications. College Park, MD., USA.
- Heer, J. and Card, S. (2004) DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. *Advanced Visual Interfaces*, 421–424. Available at <http://vis.stanford.edu/papers/doitrees-revisited>.
- Herman, I., Melançon, G., and Marshall, M.S. (2000). Graph Visualisation and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 24-43.
- Johnson, B. and Shneiderman, B. (1991) Tree-maps: A space-filling approach to the visualization of hierarchical information structures. *IEEE Conference on Visualization*, (pp. 284–291). San Diego, CA., USA.
- Jürgensmann, S., and Schultz, H. (2010). A Visual Bibliography of Tree Visualization. <http://treevis.shows.it>. Accessed Aug. 10, 2011.
- Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, G. (Editors). (2010). Mastering the Information Age: Solving Problems with Visual Analytics. Goslar, Germany: Eurographics Association.
- Kruskal, J.B. and Landwehr, J.M. (1983) Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2), pp. 162–168.
- Lau A. and Vande Moere A. (2007), Towards a Model of Information Aesthetic Visualization. *IEEE International Conference on Information Visualisation (IV'07)*. (pp 87-92).Zurich , Switzerland.
- Lee, B., Robertson, G. G., Czerwinski, M., and Parr, C. S. (2007). CandidTree: visualizing structural uncertainty in similar hierarchies. *Information Visualization*, 6(3), pp. 233–246.
- Mohammadi-Aragh, M.J. and Jankun-Kelly, T.J. (2005). MoireTrees: visualization and interaction for multi-hierarchical data. *IEEE VGTC Symposium on Visualization*. Leeds, UK.
- Morse, D. R., Ytow, N., Roberts, D. M., and Sato, A. (2003). Comparison of multiple taxonomic hierarchies using TaxoNote. *Compendium of Symposium on Information Visualization*, 126–127.
- Munzner, T. *et al.* (2003). TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. In *ACM SIGGRAPH 2003 Papers*. (pp. 453-462). San Diego, California: ACM. Available at: <http://portal.acm.org/citation.cfm?id=1201775.882291>. Accessed Oct. 22, 2011.

- Parr, C. S., Lee, B., Campbell, D. and Bederson, B. B. (2004). Visualizations for taxonomic and phylogenetic trees. *Bioinformatics*, 20(17), 2997.
- Plaisant, C., Grosjean, J., and Bederson, B.B. (1998). SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. *IEEE Symposium on Information Visualization*,. (pp. 57–64). Research Triangle Park, NC, USA.
- Sarkar, P. and Chakrabarti, A. (2011). Assessing Design Creativity: Measures of Novelty, Usefulness and Design Creativity. *Design Studies Journal* , 32(4), 348–383.
- Scagnetti, G., Ricci, D., Baule, G. and Ciuccarelli, P. (2007) Reshaping communication design tools: Complex systems structural features for design tools. *International Association of Societies of Design Research*. (pp. 1-20). Hong Kong. PRC.
- Schroeder, M., Gilbert, D., Van Helden, J., and Noy, P. (2001). Approaches to visualisation in bioinformatics: from dendrograms to Space Explorer. *Information Sciences*, 139, 19-57.
- Shapiro, Mattias. (2010). Once Upon a Stacked Time Series. In J. Steele & N. Iliinsky (Eds.), *Beautiful Visualizations*. (pp. 15–36). Beijing: O'Reilly.
- Shneiderman, B. (1992) Tree visualization with tree-maps: A 2-dimensional space filling approach. *ACM Transactions on Graphics*, 11, (1): 92-99.
- Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages*. (pp. 336–343). Boulder, CO., USA.
- Thomas, J. and Cook, K. (2004). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. Richland, WA: Pacific Northwest National Labs.
- Tufte, E.R. (1990). *Envisioning Information*. Connecticut: Graphics Press.
- Ying Tu, and Han-Wei Shen. 2007. Visualizing Changes of Hierarchical Data Using Treemaps. *Visualization and Computer Graphics*, *IEEE Transactions* 13 (6): 1286–1293.
- Ware, C. (2004). *Information Visualization: Perception for Design Second Edition*. San Francisco: Morgan Kaufmann Publishers.
- Wattenberg, M. (1999). Visualizing the stock market. In *CHI'99 extended abstracts on Human factors in computing systems*. (pp. 188–189). Pittsburgh, PA., USA.
- Wright, P. (2009) What if ...? Designing tools to help the public make difficult decisions. *Information Design Journal*, 17(3). 202-210.