



Discovering temporal changes in hierarchical transportation data: Visual analytics & text reporting tools



John Alexis Guerra-Gómez^{a,1}, Michael L. Pack^{b,*}, Catherine Plaisant^{c,2}, Ben Shneiderman^{d,3}

^a University of Maryland, College Park, Center for Advanced Transportation Technology Laboratory, Human-Computer Interaction Lab, United States

^b University of Maryland, College Park, Center for Advanced Transportation Technology Laboratory, Human-Computer Interaction Lab, Department of Civil & Environmental Engineering, United States

^c University of Maryland, College Park, Human-Computer Interaction Lab, United States

^d University of Maryland, College Park, Human-Computer Interaction Lab, Department of Computer Science, United States

ARTICLE INFO

Article history:

Received 14 November 2014

Received in revised form 25 November 2014

Accepted 25 November 2014

Available online 26 December 2014

Keywords:

Information visualization

Visual analytics

Change visualization

Tree visualization

Difference visualization

ABSTRACT

Analyzing important changes to massive transportation datasets like national bottleneck statistics, passenger data for domestic flights, airline maintenance budgets, or even publication data from the Transportation Research Record can be extremely complex. These types of datasets are often grouped by attributes in a tree structure hierarchy. The parent-child relationships of these hierarchical datasets allow for unique analytical opportunities, including the ability to track changes in the dataset at different levels of granularity, over time or between versions. For example, analysts can use hierarchies to uncover changes in the patterns of passengers flying in the United States over the last ten years, breaking down the data by states, cities, airports, and number of passengers. Exploring changes in travel patterns over time can help carriers make better decisions regarding their operations and long-range planning.

This paper describes TreeVersity2, a web-based data comparison tool that provides users with information visualization techniques to find what has changed in a dataset over time. TreeVersity2 enables users to explore data that can be inherently hierarchical or not (by categorizing them by their attributes). An interactive textual reporting tool complements the visual exploration when the amount of data is very large. The results of two case studies conducted with transportation domain experts along with the results of an exit questionnaire are also described. TreeVersity2 preloaded with several demo datasets can be found at (<http://treeversity.cattlab.umd.edu>) along with several example videos.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Hierarchies help us organize and understand information. Examples of hierarchies include the evolutionary tree of species, the federal budget, and business organizational charts. Three examples are shown in Fig. 1. Many visualizations have been designed to represent, navigate, and analyze tree structures using node link representations (Plaisant et al., 1998),

* Corresponding author. Tel.: +1 301 405 0722.

E-mail addresses: john.guerra@gmail.com (J.A. Guerra-Gómez), packml@umd.edu (M.L. Pack), plaisant@cs.umd.edu (C. Plaisant), ben@cs.umd.edu (B. Shneiderman).

¹ Tel.: +1 740 591 0435.

² Tel.: +1 301 405 2768.

³ Tel.: +1 301 405 2680.

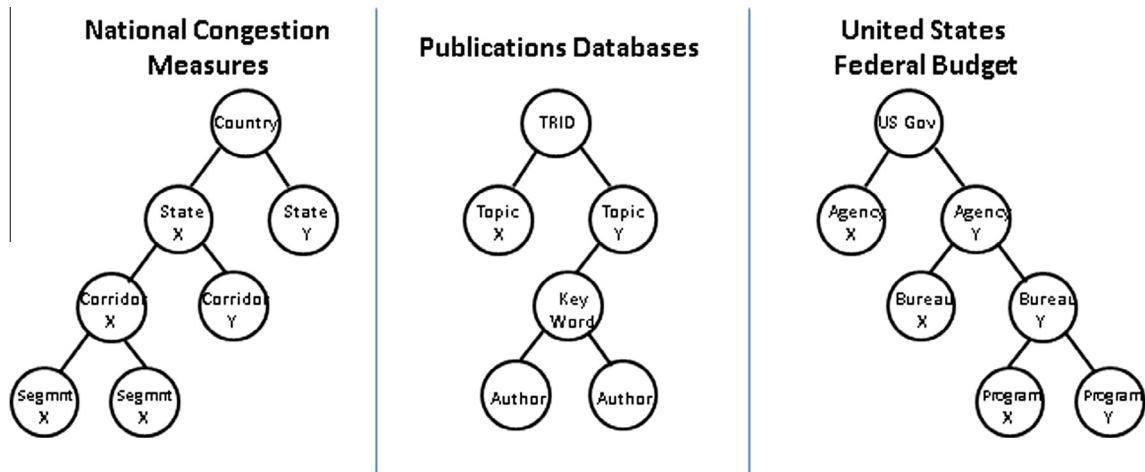


Fig. 1. Example of datasets viewed as hierarchies.

treemaps (Shneiderman, 1992), radial representations (Fisher et al., 2001) or icicle trees (Kruskal and Landwehr, 1983). Such techniques are now used in scientific or general-audience publications and websites to show transportation data. Our previous work (Guerra Gómez et al., 2013a) presented TreeVersity2 a visual comparison tool that introduced novel visualization techniques for finding changes in datasets over time looking at them as trees. This paper demonstrates how this novel visualization technique can be used by analysts to visually compare trees that change over time and thereby derive insights from transportation data.

Information visualization (InfoVis) techniques have been used to analyze a variety of transportation data (Beecham and Wood, 2014; Hughes, 2004; Pack et al., 2005; Pack, 2010; Wang et al., 2014) including airline passenger data (Hurter et al., 2014) and road congestion data (Cheng et al., 2013). The analysis of trends requires users to answer questions such as: how and where has congestion changed both nationally and locally? What new topics are emerging in our scientific publications? How has the department of transportation budget changed over the last ten years? How have airline maintenance budgets shifted from year to year, and have those changes affected safety? Those questions are challenging and require sifting through large amounts of data to evaluate multiple aspects of change, at multiple levels of aggregations. To address this challenge, a partnership between the Center of Advanced Transportation Technologies and the Human–Computer Interaction lead to the development of TreeVersity2 (Guerra Gómez et al., 2013b). TreeVersity2 uses visualization, an approach that complements alternative techniques used by others working with similar data, e.g. (Zografos and Madas, 2006) for airline passenger data or (Kamga and Yazıcı, 2014; Liu and Fei, 2010) for traffic congestion.

This paper first describes TreeVersity2 and illustrates how it allows transportation analysts to explore their data, then discusses the performance and implementation issues we encountered and how they were addressed – which will inform the development of future web-based visualization systems. Finally, two case studies conducted with transportation domain experts are described in detail, as well as the results of an exit questionnaire (based on a total of ten case studies).

2. Related work and definitions

While a significant amount of research has been conducted to visualize (Kruskal and Landwehr, 1983; Lamping, 1996; Robertson et al., 1991; Shneiderman, 1992) and explore (Card and Nation, 2002; Heer and Card, 2004; Parr et al., 2004; Plaisant et al., 1998) **single** trees, the problem of comparing **two** trees is significantly harder. We have identified and classified the following types of tree comparison (Fig. 2):

- **Type 0:** topological differences between two trees where the nodes only contain a label. Example: finding created, moved and removed topics between two versions of the TRB research publications hierarchical categorization system, without looking for number of papers.
- **Type 1:** positive and negative changes in leaf node values with aggregated values in the interior nodes (i.e. trees that can be visualized with a treemap (Shneiderman, 1992)) and no changes in topology. Example: comparing the change on the number of workers on the traffic agencies of the country, grouped by Agencies, Counties and Regions, given that no Agencies are created or removed.
- **Type 2:** positive and negative changes in leaves and interior node values with no changes in topology. Example: comparing the salaries in the Department of Transportation's organizational chart between two years, when no reorganization has occurred.

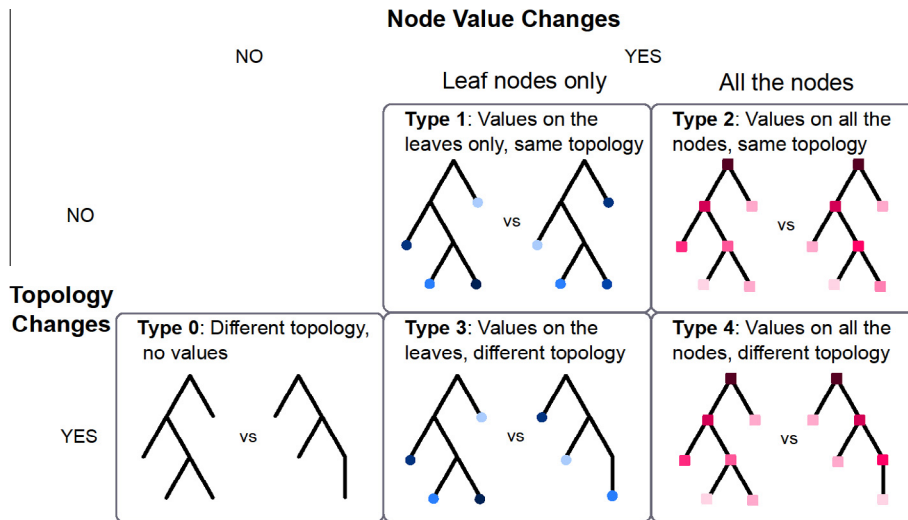


Fig. 2. Five types of tree comparison problems. Previous work has addressed Types 0 and 1, with only one attempt at Type 3 [46]. TreeVersity supports all five cases, with emphasis on Types 1–4, the ones that include node value changes.

- **Type 3:** positive and negative changes in leaf node values with aggregated values in the interior nodes and with changes in topology. Example: again, finding changes in the TRB research publications hierarchical categorization system, but looking at changes in the number of papers published by topic and at created, removed nodes.
- **Type 4:** positive and negative changes in leaves and interior node values, with changes in topology. Example: comparing the number of TRB's website visits between two months using the file hierarchy as a natural organization. Some pages might be created or removed, and each page in the hierarchy has an independent number of visits.

2.1. Dimensions of node changes

According to related work and our own research, analysts who want to perform these types of tree comparisons want to be able to find and understand the following dimensions of tree node changes:

- **Direction of change:** positive, negative or neutral (no change).
- **Absolute change:** the actual amount of change, e.g. the Department of Transportation budget will be decreased by 5.99 billion dollars between 2012 and 2013.
- **Percentage change:** the absolute change with respect to the original value, such as the cut in the Department of Transportation represents a 1.32% decrease with respect to its budget in 2012.
- **Relative change:** how does a node change compare to the changes of other nodes in the tree, e.g. the cut in the Department of Transportation (\$-5.99 Billion, -1.32%) will be considerably smaller than the expected decrease in the Department of Labor's budget (\$-52.66 Billion, -29.84%).
- **Topology change:** nodes that are created, removed, or moved. e.g. The Office of Thrift Supervision (\$64 million dollars) was removed from the Department of Treasury in 2012.

2.2. Types of hierarchies

TreeVersity2 allows the exploration of changes over time in datasets using hierarchies. These hierarchies can be either **fixed** when there is an inherent parent-to-child relationship (e.g. the Agency → Bureau classification in the US Federal Budget, where Bureau → Agency does not make sense), **dynamic** when the hierarchy is constructed by grouping rows by their attributes as defined in the original treemap paper (Shneiderman, 1992) (e.g. Census population group by gender, race and age range), or **mixed** where some levels of the hierarchy are fixed and some dynamic (e.g. grouping the US Federal Budget by Discretionary/Mandatory Accounts that are dynamic and then by Agency/Bureau that are fixed). On the other hand, each one of these hierarchies can be **aggregated** if the parent node values for are calculated as a function of their child values (e.g. adding up the values) or **non-aggregating** if the parent node values are independent from their child values.

2.3. Related work

Researchers have proposed a significant number of solutions for comparing trees by topology (Type 0) (Amenta and Klingner, 2002; Auber et al., 2003; Bremm et al., 2011; Card and Nation, 2002; Furnas and Zacks, 1994; Graham and

Kennedy, 2001, 2005, 2007, 2008; Graham et al., 2000, 2002; Hong et al., 2003; Lee et al., 2007; Mohammadi-Aragh and Jankun-Kelly, 2005; Morse et al., 2003; Munzner et al., 2003; Plaisant et al., 1998; Spenke, 2001; Visualizations for taxonomic and phylogenetic trees, n.d.) or for visualizing changes in node values with aggregated values in the interior nodes (Type 1). Only one project (Tu and Shen, 2007) has attempted combining both types of differences at the same time (Type 3). For a more comprehensive description of the related work please refer to (Guerra Gómez et al., 2013a).

The Multiple Skyline Graphs by Caemmerer is a visualization artifact designed to show changes in datasets. It uses the concepts of variable width bar charts that are similar to the ideas used in the StemView, however Skyline Graphs were not designed for tree structures and therefore do not support them. The Skylines, were featured in an on-line article in the SAP Design Guild (“SAP Design Guild – UI Design Blinks (2012),” 2013) and do not seem to have been academically published anywhere else. On the other hand (Brodbeck and Girardin, 2003) work on visualizing survey results use a area filling hierarchical visualization base with overlying line graphs. This is a similar technique to the one used for the StemView, but was not designed for showing change, and uses a different representation.

Related work on the reporting tool described in this paper has focused mainly on annotating charts to offer more information about them. (Kong and Agrawala, 2012) developed a system that analyzes a bar chart, computes and displays graphical overlays such as reference structures, highlights, redundant encodings, summary statistics, and annotations; these graphical overlays were designed to enrich the information provided by static graphs, and do not allow for interactions. In contrast, the reporting tool described here was designed to guide users through the most significant changes in a visualization using interactive textual representations.

Contextifier by (Hullman et al., 2013) is another example of an annotation system for visualizations. Contextifier was designed to generate annotated stock line charts using a news article of a company (owner of the stock). The tool explores the data for points of interest, which could be either visually salient points on the graph, or relevant information in the news article, and using this information produces a customized line chart with textual annotations. Apart from having a different problem domain, Contextifier differs from the reporting tool in that it was not designed to guide users through a list of the most relevant points in the chart, instead it highlights in a static way a selection of points of interest using overlays.

Our early work on TreeVersity1 (Guerra Gomez et al., 2011; Guerra Gómez et al., 2012, 2013a) introduced a novel visualization glyph called the Bullet that represented four characteristics of tree nodes. TreeVersity1 allowed users to compared two versions of a tree (e.g. the change in the number of passengers flying in the United States (US) between 2001 and 200). But we found that users wanted to expand their explorations to multiple versions of a tree (e.g. the tendencies of passengers

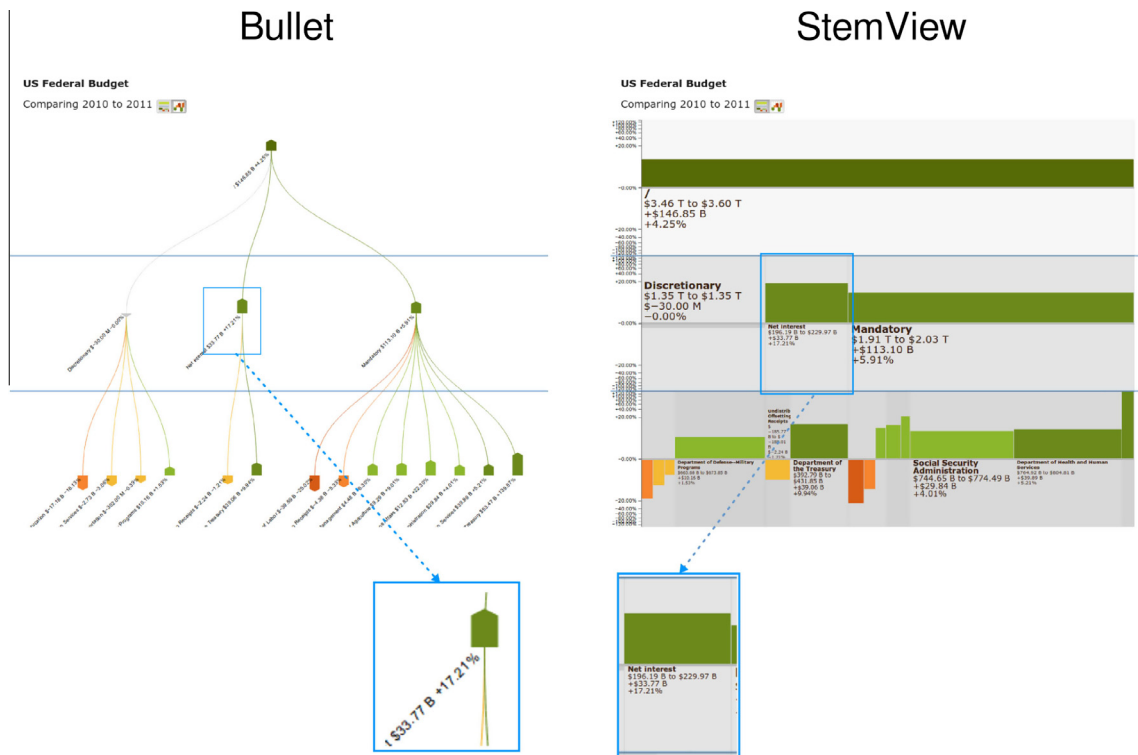


Fig. 3. Bullet vs StemView: US Federal Budget grouped by BEA Category and then by Agency, filtering by the biggest agencies. The Bullet behaves better on filtered results that reduce the number of Bullets on the screen and allow the placement of labels. However, the Bullet fails to show the actual size of the agencies.

flying in the US for the last ten years). To address this, we developed TreeVersity2 (Guerra Gómez et al., 2013b) that allows users to explore changes over time in datasets. TreeVersity2 also introduced the StemView, an area-based visualization technique that represents five characteristics of change in all the nodes of a tree, not only the leaves. Fig. 3 presents a side-by-side comparison of the Bullet vs the StemView.

TreeVersity2's interface was initially described in (Guerra Gómez et al., 2013b) intended for the information visualization community. It described two case studies conducted with the Food and Drug Administration and the National Cancer Institute. This paper focuses on transportations case studies, and also provides performance and implementation information, as well as more information on the user interactions, and reports the results from a user survey.

3. TreeVersity2

3.1. Overall interface description

TreeVersity2 is an interactive web based data visualization tool that allows the exploration of time changing datasets using hierarchies. Users can see an overview of the entire time range and select two time points to compare. Controls allow analysts to specify the hierarchy they want to see (i.e. its structure and depth), and map data attributes to the visual attributes of the visualization. Fig. 4 shows an example from the case study in which US Department of Transportation compared changes in the number of passengers flying (explained in more detail in Section 3.2). We see on the left an overview of the 13 years of data with 3 levels of the hierarchy (year, month and state). Year 2000 and 2001 are selected as end points for the detailed comparison, which is shown in the center of the screen, using the StemView. The StemView (described in detail in Section 3.2) is a novel tree visualization that enables users to examine the detailed differences by highlighting actual and relative changes, positive and negative changes, created and removed nodes, and starting or ending values of the nodes, all while keeping the context of the tree and showing inner node changes. Finally on the right on the screen we see four control panels (with the legend panel currently opened in Fig. 4, and the reporting tool labeled “What changed?” opened in Fig. 5). Other panels give control of the hierarchy being shown, and provide filters to focus on parts of the tree.

The left side overview of the entire time range can use traditional timelines to compare actual values (as shown in see Fig. 5, using a budget analysis example) or TimeBlocks for comparing differential values (Fig. 4). The TimeBlocks use color boxes to represent differential change between sequential years. For example, in Fig. 4, increases in the number of passengers taking off an airport are represented by green boxes while decreases are shown in red. Each horizontal line in the TimeBlocks represents a corresponding attribute (or node in the tree), so for the “Month” overview there are twelve TimeBlocks with one for each month of the year (January at the top, December at the bottom).

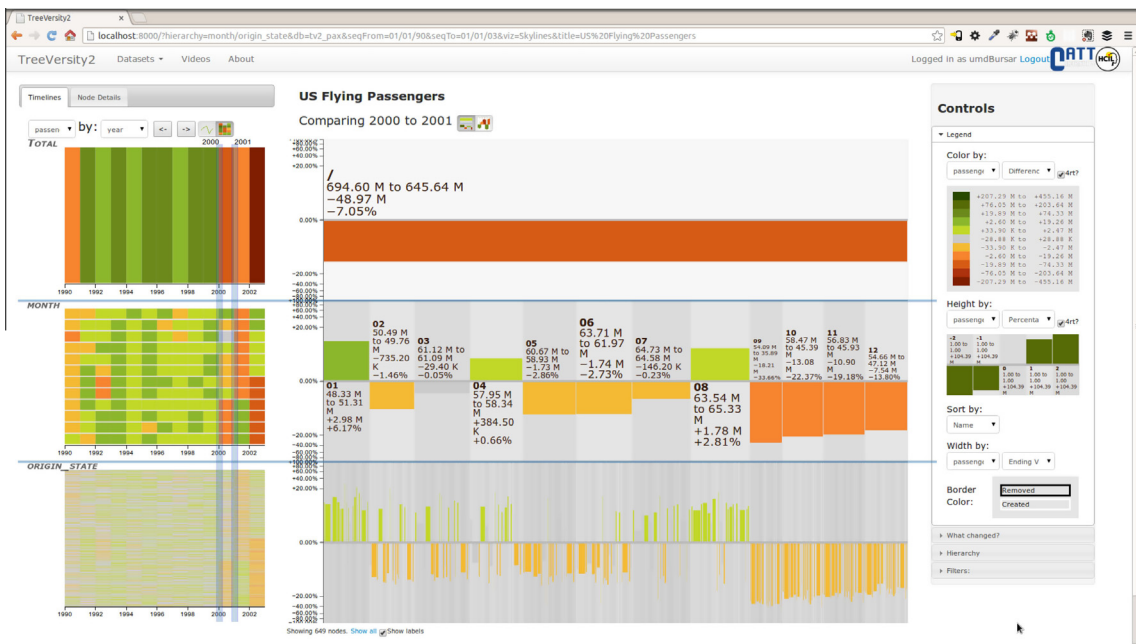


Fig. 4. Change in the number of passengers among US States divided by month and by state. Increases are shown in shades of green and decreases in shades of red to orange. On the left side the time blocks shows change over the 13 year period. In the center the StemView shows more details between 2 selected points, and on the right are the control panels (legend etc.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

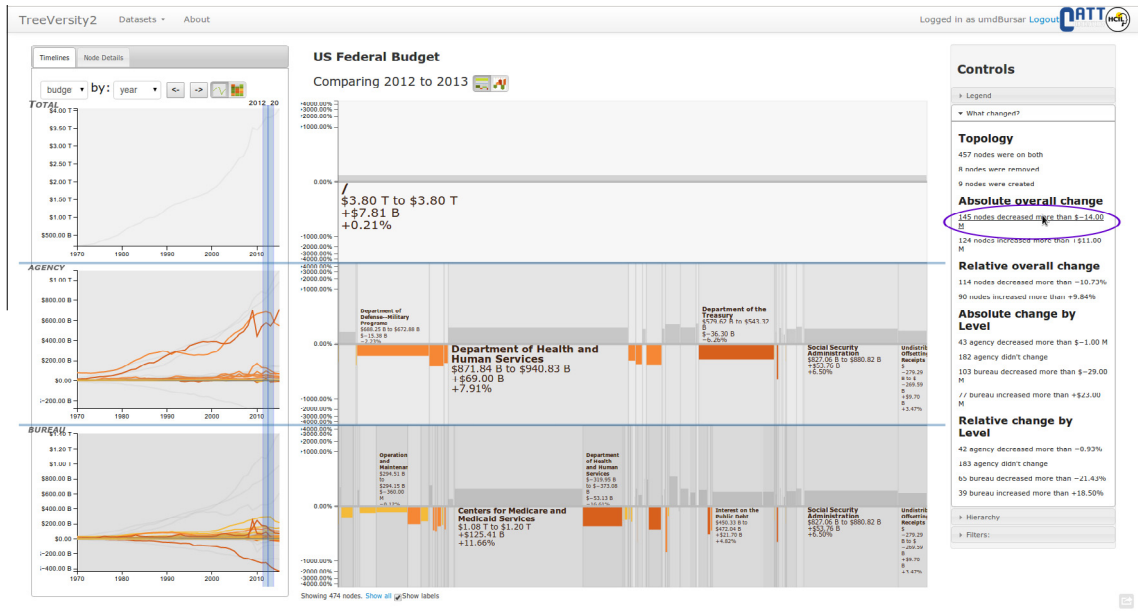


Fig. 5. The reporting tool highlighting all the agencies and bureaus in the US Federal Budget that decrease the most (i.e. more than \$14 million dollars). Users can filter the display down to only those accounts by clicking on the corresponding line of text in the reporting tool. To provide context parents and siblings of those nodes remain visible, shown in gray.

TreeVersity2 allows users to review the detailed changes between any two time points. The range and granularity varies according to the data characteristics and can range from seconds (e.g. comparing number of tweets in periods of five seconds) to tens of years (e.g. comparing the number of publications in a research field in the last twenty years by decades).

In the legend panel users can assign the available variables to the color, height, width, and sorting order of the boxes. Since the visualizations on the StemView represent change, users can select one of five modifiers for each variable: (1) actual difference, (2) relative difference, (3) starting value, (4) ending value, or (5) maximum of the starting and ending values. Different combinations of these parameters allow richer explorations.

The reporting tool (right side of Fig. 4) helps users locate major differences by exploring a textual list of outliers calculated for each pair of compared time points. For instance, Fig. 5 illustrates how analysts at the Office of the Management of the Budget (OMB) could identify all of the accounts decreasing more than \$14 million dollars in the US Federal Budget between 2012 and 2013, and see those changes in the center StemView while keeping the context of the hierarchy (more detail in Section 3.3). Lastly, users can apply specific range filters on each one of the characteristics of change. For example, users can explore all of the changes in all accounts in the US Federal Budget that have a budget higher than \$10 million dollars or all accounts increasing or decreasing more than \$1 million dollars. The nodes that do not match the criteria are removed and the remaining nodes are given all the available space. Smooth animations and transitions allow users to remain oriented while the visualization is updated.

3.2. The StemView visualization

The StemView is the central visualization that represents five characteristics of change in tree nodes: direction of change, absolute change, relative change, starting or ending values, and created and removed nodes (Fig. 6 top). It uses an area filling representation based on icicle trees (Kruskal and Landwehr, 1983) where the levels of the hierarchy are distributed vertically in equally sized rows. Fig. 6 (bottom) shows an example StemView constructed for the US Federal Budget between 2008 and 2009, aggregating the budget accounts by their pertinence to the budget (On or Off budget) and by their Budget Enforcement Act Category (BEA, that determines if they are Discretionary, Mandatory or Net Interest). The vertical space available is distributed equally among the levels, and inside each level, the horizontal space is distributed among the nodes represented as boxes according to their respective ending budget. Fig. 6(a) shows this first step, which is basically an icicle tree showing the budgets of each node for 2009. The StemView builds on top of the icicle to also show the actual and relative changes of each node. For this purpose, it splits each level vertically using a horizontal line that will represent zero change, as shown in Fig. 6(b). From that zero line, a sub-box is drawn with the same width of the node's containing box but with a height relative to the relative change of the node (e.g. +17.94% for the overall budget). The sub-boxes go upward from the horizontal line for increasing nodes and downward for decreasing nodes (Fig. 6 c and d shows the final step where the sub-boxes are colored using the actual amount of change of each node (e.g. +\$535.12 billion dollars) using two color scales. These scales

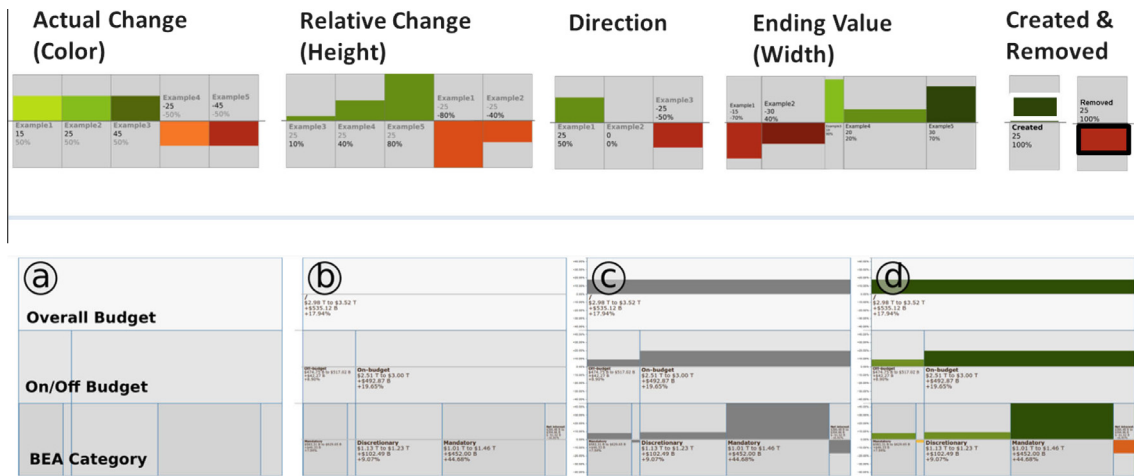


Fig. 6. The StemView: on the top the five characteristics of change in tree nodes in one visualization. On the bottom steps for the StemView construction: (a) first an icicle tree for the ending values is used as the base of the visualization, (b) then inside each level a horizon line is drawn representing no change. (c) Sub-boxes with height corresponding to the relative change are drawn inside each node. (d) Finally the nodes are colored using the actual amount of change.

are typically set to green for increasing values and yellow-to-red for decreasing values. Finally, the StemView uses white borders around the sub-box to represent created nodes and black borders for the deleted ones. Each one of the characteristics of the StemView, height, width, color of the boxes, and the order in which they are distributed among their parents, can be assigned to different variables of the dataset and their modifiers (starting value, ending value, actual difference or relative difference).

3.3. Finding major differences: the reporting tool

While TreeVersy2 was designed for trained analysts the complexity of the data and tasks is such that analysts can have a difficult time getting started. To address this issue, we developed an interactive, text-based, reporting tool for TreeVersy2 which aids users in finding significant changes in their datasets. Every time users change the end points for detail analysis, the reporting tool generates a new list of the major changes in the tree, grouped by different *reporting metrics*. The current version of TreeVersy2 includes five reporting metrics:

- *Topological changes*, which finds the number of nodes that were created (e.g. new bureaus), the number removed, and the numbers that appear in both compared time points.
- *Absolute changes overall*, which finds outliers on the actual changes (i.e. changes in dollars for the US Federal Budget dataset) on the tree node values compared with all the other nodes in the tree.
- *Relative changes overall*, same as the previous one but with the relative change (i.e. the nodes that have seen the largest percentage increase, or decrease).
- *Absolute change by level*, similar to the *absolute changes overall* but compared against the nodes in a certain level only. For example, this metric will reveal a Bureau that increased more than other Bureaus in the Budget, but that would have been dwarfed by the even bigger opposite changes at the agency level.
- *Relative changes by level*, same as before but for the relative change.

The reporting tool was designed to be extensible, so adding new metrics simply requires adding a metric algorithm module in the source code. While our current implementation remains generic, we anticipate that advanced metrics will be application domain dependent.

Every time users change the compared time points in the dataset, the reporting tool is recalculated. The results are presented in a textual list of reporting items that provide a human readable description of each group of interesting nodes, such as *145 nodes decreased more than \$-14 M*. When users hover over one reporting item, the corresponding nodes are highlighted on the StemView and timelines as shown on Fig. 5. This allows users to identify nodes while keeping the context of the tree. If users want to explore further, they can click on the reporting item, which will filter categories of change with smooth animation (e.g. change by topology or change by level). Each item in the list describes a group of nodes with a node count and why they are interesting (e.g. *145 nodes decreased more than \$-14 M*). Users can hover over an item in the report to highlight the corresponding nodes in the StemView and time visualizations as shown in Fig. 5. Users who want to explore further can click over the report item to filter out all the non-matching nodes, leaving only the nodes referenced by the report item.

4. Case studies

4.1. Change in passengers flying in the US between 1990 and 2003

This case study provided the original motivation to build TreeVersity2. We worked with a data analyst of the Bureau of Transportation Statistics (BTS) of the US Department of Transportation (DOT). The dataset used in this case study contained 65,535 records representing the number of passengers reported to have taken off from a US airport, month by month, between 1990 and 2003. The data also included the airport's state and city, which were used to create the hierarchy. The analyst wanted to explore the dataset to see if it was possible to easily identify interesting patterns, but he was not searching for anything specific. We knew beforehand that significant patterns might emerge from the attacks of 9/11, but it was unclear what other insights would be identified while exploring changes over time in the data.

The analyst was excited to discover some unexpected results, for example not all the states reported decreases in September 2001 (see Fig. 7) Maryland reported an increase (possibly because the Washington airports had strong decreases). TreeVersity's zooming and filtering functions were extensively used for finding airports with an unexpected increasing number of passengers after 9/11. We expanded this exploration, as shown in Fig. 8 by adding a city level to the hierarchy and filtering to show only the cities presenting increases. This revealed that major airports such as Oakland, Pittsburgh, Fort Lauderdale and Baltimore increased their number of passengers overall along with Minot International Airport (MOT) in North Dakota. MOT stood out because it reported an increase of more than 442% compared to the previous year, going from 673 to 3654 enplanements.

The Bureau of Transportation Statistics Analyst along with the Director of BTS were surprised by these trends, and noted that they would not have noticed these patterns as quickly (or at all) without the use of the TreeVersity2 visualization tool. While TreeVersity did not offer any causality for the increases in certain airport—especially MOT—the insights gained drove BTS analysts to revisit this data and start an internal discussion about why this might have occurred.

4.2. Transportation bottleneck data

Using internal bottleneck statistics created from probe-based speed data, researchers within the Center for Advanced Transportation Technologies (CATT Lab) used TreeVersity2 to analyze changes in traffic congestion levels over time for various states, counties, roads, etc. The CATT Lab had already used visualization tools to explore congestion levels in single geographies and predefined date ranges, but they lacked the means to quickly analyze change over time at multiple, hierarchical geographies. Their bottleneck data contained 96,205 rows representing traffic bottlenecks for six US States between 2010 and 2011. The data also included geographical information (state and county), as well as details of the roads where the

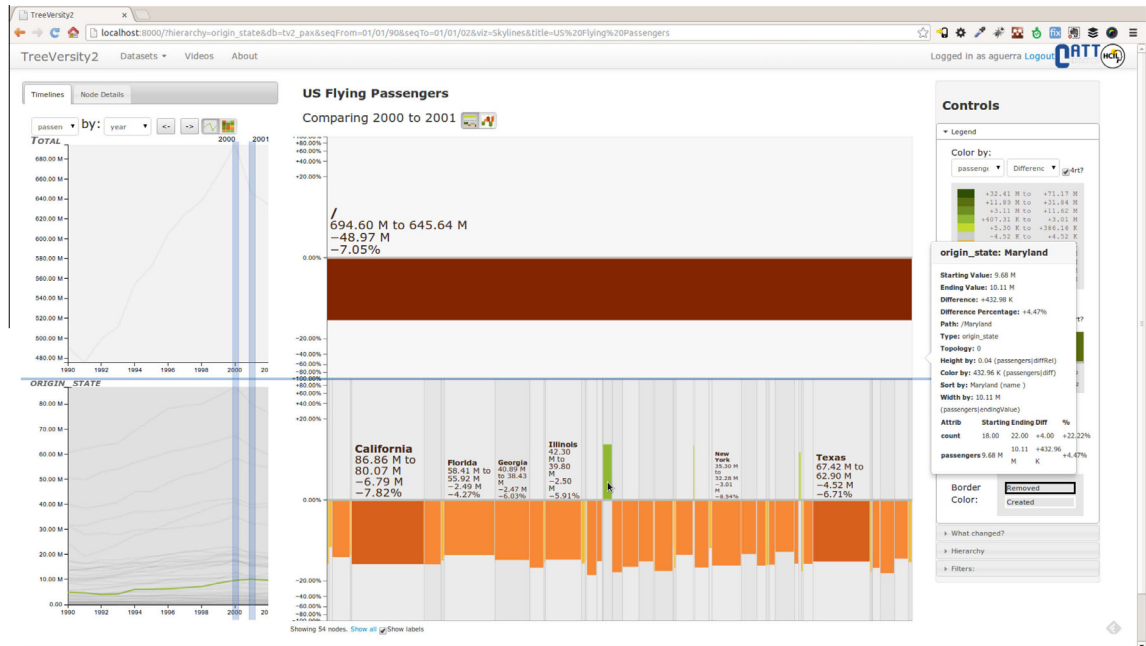


Fig. 7. Change in the number of passengers among US Airports between 2001 and 2000 divided by states. Nodes are sorted by their actual difference. The StemView shows that despite the widespread decrease in the number of passengers; some states such as Maryland (under the cursor) actually reported increases.

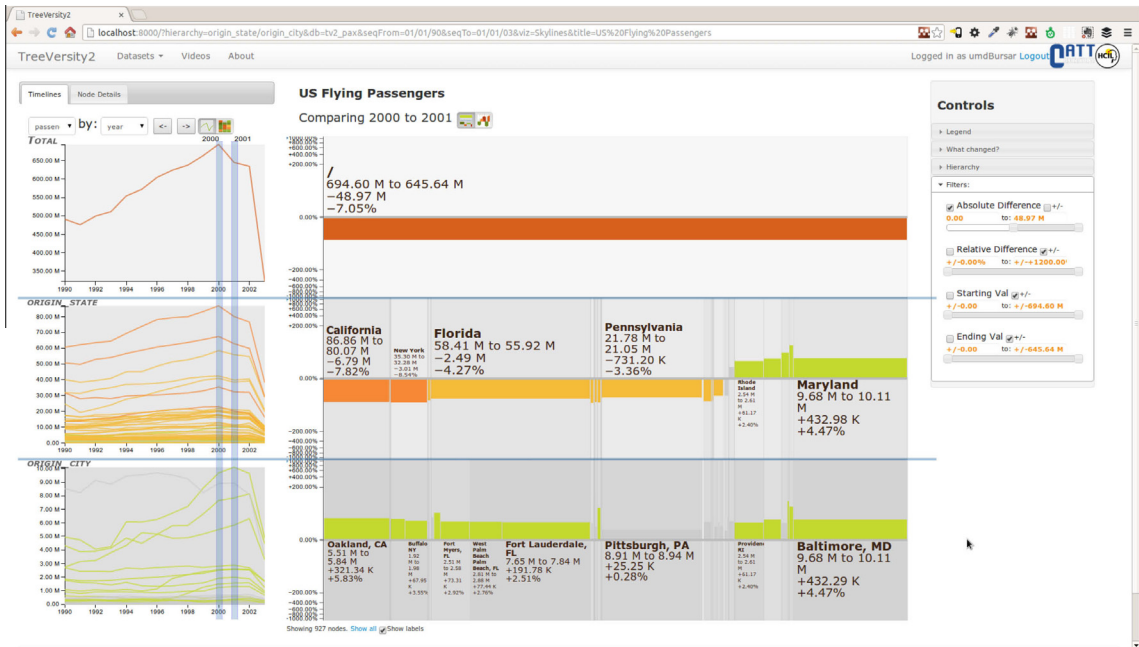


Fig. 8. To further understand where the increases had occurred, a third level was added to the hierarchy (US Cities), then the hierarchy was filtered to show only the cities that have an increase (their parent states remain visible in the hierarchy for context). This view shows that apart from Baltimore, other cities like Fort Lauderdale, Oakland and Pittsburg also reported increases in 2001.

bottlenecks occurred (road name, class, and location). The numeric variables included the average duration and length of the bottlenecks, as well as the number of occurrences and a measure of its impact or severity (Fig. 9).

After integrating the data into TreeVersity2, the application was then demonstrated to multiple operations engineers and planner within the North Carolina Department of Transportation. In less than 2-min, the analysts were able to identify a previously unknown “major long-term shift in congestion” that significantly affected a small rural county in the state. Congestion

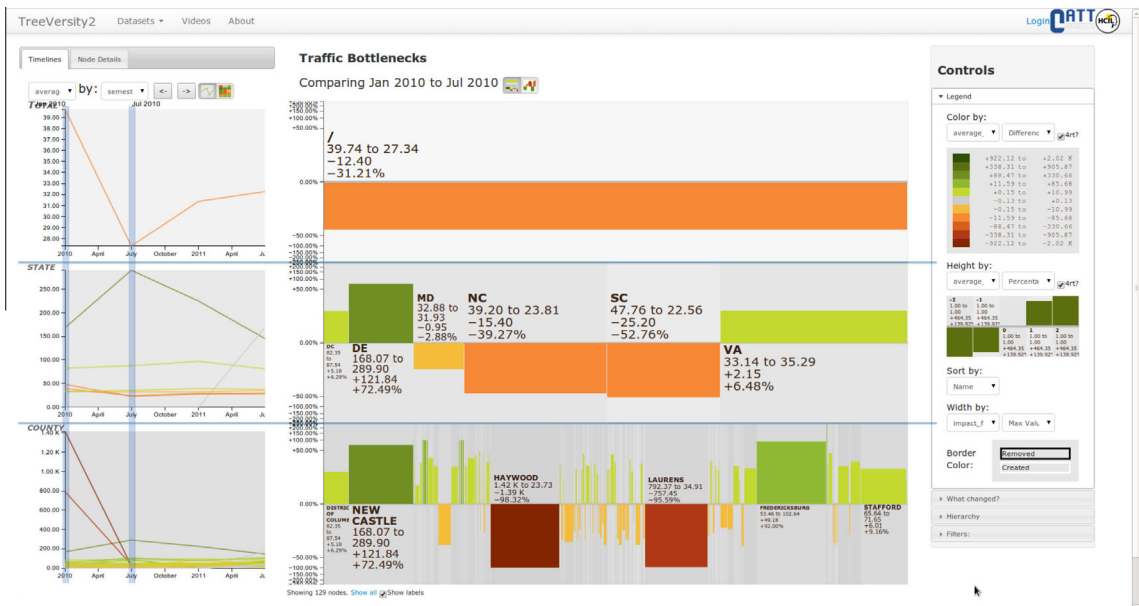


Fig. 9. Average traffic bottlenecks grouped by state and county for six states of the US (Washington DC, Delaware, Maryland, North Carolina, South Carolina and Virginia). The StemView shows in dark red that there are two counties with significant decreases in their traffic bottlenecks Haywood (NC) and Laurens (SC). On the other hand New Castle (DE) showed a significant increase in its traffic bottlenecks with an increase of 72.49% shown on dark green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in this region was unheard of, so the analysts immediately started asking follow up questions or making other statements including:

- What is that significant spike in “red” in that one county? Is that a data error? That cannot be right.
- Why did congestion in the neighboring county go down?
- What happens if you look at the following month or the prior month?
- Can you filter the results geographically?
- Can you filter the congestion based on severity of the bottlenecks?

These questions and others drove the analysts to dig deeper into the issues of severity and cause of the congestion. It quickly became clear that a major rockslide in the neighboring county had closed down an interstate for that particular month. The resulting congestion occurred because of detours that were being taken by travelers through the adjoining county. These unapproved detour routes were believed to be the cause of the spike in congestion in the neighboring county.

The analysts were extremely impressed with the speed at which they were able to identify outliers in the data, filter, zoom, and collaboratively come to a deeper understanding of what was going on in their own state with data that they had had access to for years, but were unable to interpret. They noted that the tool was responsive and drove them to ask additional questions that they would not have previously thought to ask. Their only significant complaint was in our labeling

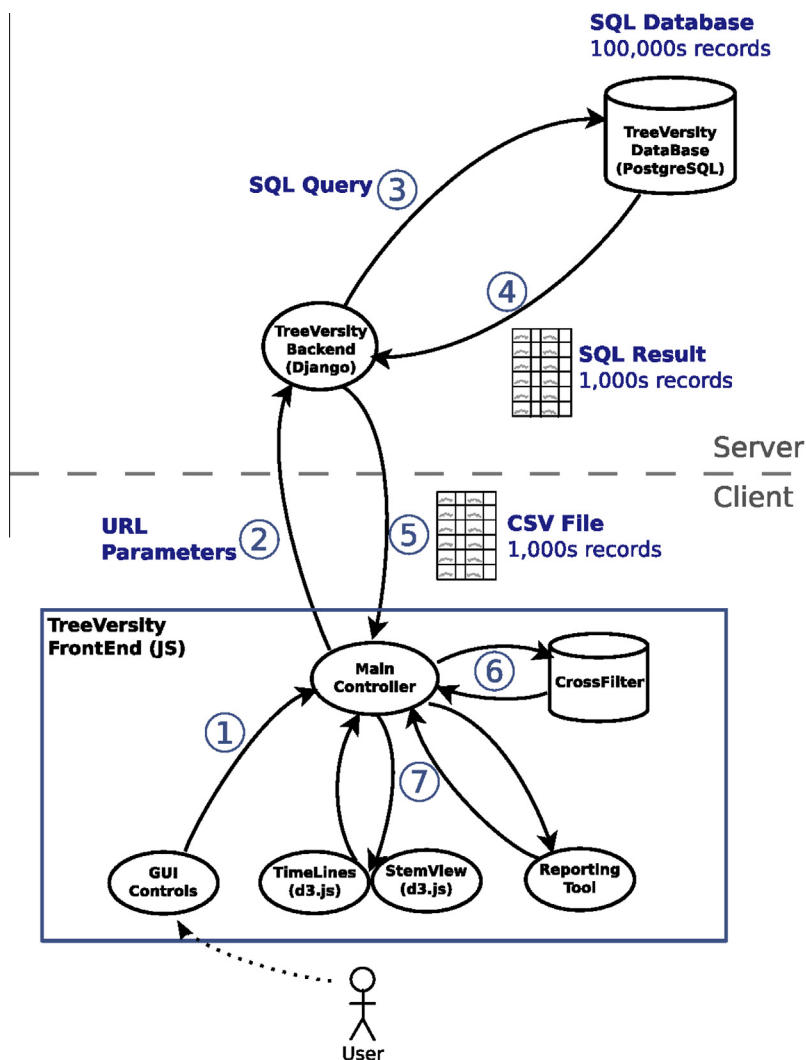


Fig. 10. TreeVersity2 architecture. The tool divides the processing between the server and client. When the user starts an exploration, (1) TreeVersity2 queries the back-end (2) with the parameters of the exploration encoded in the URL. The back-end then generates a dynamic SQL query (3) and sends it to the PostgreSQL database. The database returns the result in SQL format (4) and the back-end creates a CSV file that is passed back to the front-end (5). The front-end loads the CSV file in a CrossFilter (6) and then passes it (7) to the visualizations and the reporting tool. Once the data is loaded into the CrossFilter all of the following interactions which do not change the hierarchy, are processed on the front-end without having to query again the back-end.

of units. TreeVersity was designed to be data agnostic, and at the time publication, did not have the ability to import units of measure from datasets and was unable to label them properly. This required initial guesswork and confusion by the analysts.

This case study helped validate the exploratory nature of TreeVersity2. It demonstrated how analysts could generate new hypotheses (such as possible causes for a widespread congestion events), and answer questions by visually interacting with the data using TreeVersity2.

5. Implementation

TreeVersity2 was designed to allow fluent interactions with datasets in the order of hundreds of thousands of records which generate trees with thousands of nodes. This was achieved by carefully distributing the workload between the client and server applications (as shown in Fig. 10).

On the server side, TreeVersity2 has two main components: (1) an application built on Python using Django and (2) a PostgreSQL database. The Django application has the main objective of answering requests from the client, parsing URL parameters, and generating dynamic SQL scripts to be issued to the Database. This application generates two types of scripts, one for aggregated trees and the other for non-aggregated trees.

The SQL generated for aggregating trees uses a GROUP BY clauses to aggregate values on the interior nodes of the tree using an aggregating function, such as average, or summation. For non-aggregated trees, the back-end stores the pre-calculated values of the interior nodes on the database and queries them directly (i.e. not calculating them on the fly).

TreeVersity2 front-end was built using web development libraries, namely:

- D³, used as the main visualization library.
- JQuery, used for DOM manipulation.
- Bootstrap and JQuery UI, for the UI layout and widgets.
- RequireJS for modularizing the pieces of code.
- LESS, for the use of variables inside CSS.
- CrossFilter, for data manipulation.

Adding meaningful labels to the StemView was as challenging as adding labels to any area based visualization. Different techniques were used to add labels and a dynamic algorithm that draws labels on the StemView boxes whenever the label can fit. The algorithm also checks to see if larger fonts can be used and still fit the StemView box. The resulting experience shows animated fonts that readjust, appear or disappear to fit the changing sizes of the StemView boxes.

6. Summary of user survey

As part of our evaluation methodology, an exit questionnaire was distributed to our domain expert users to gather their impressions. While we only report on two transportation-related case studies here, a larger number of case studies were conducted and we collected 10 responses representing a diverse set of expertise levels and application domain (Guerra Gómez et al., 2013b). The questionnaire included open ended questions about the way they used the tools and the findings they made with it, then 8 specific questions with possible answers along a 7-point Likert scale, ranging from 1 being “Not

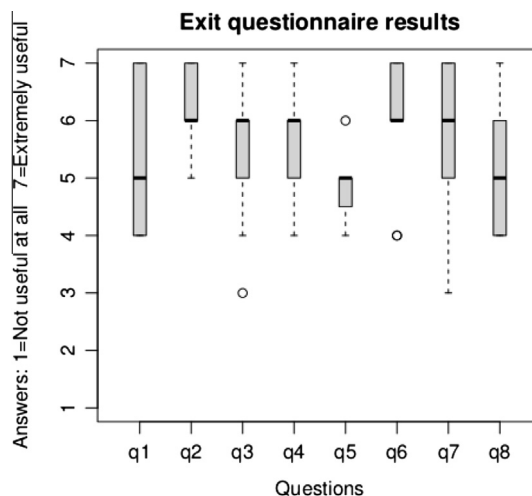


Fig. 11. Exit questionnaire results for ten domain experts that completed it.

useful at all” or “not comprehensible at all”, to 7 being “Extremely useful” or “Extremely easy to comprehend”. The eight questions were:

- q1: for this particular case study TreeVersity was:
- q2: in general the tool is likely to be:
- q3: did the reporting tool help direct your exploration?
- q4: did you find the StemView comprehensible?
- q5: did you find the Bullet comprehensible? (This was relevant only to early users)
- q6: would you like to continue working with TreeVersity?
- q7: Would you be willing to install and use TreeVersity on your own?
- q8: How does this compare to your original expectations before starting with the tool?

Users were allowed to skip questions, and only 4 questions were left unanswered. Fig. 11 shows the box-plots of the answers for the 10 answers collected. The answers showed mostly positive answers for most of the questions, with some outliers for the question about the reporting tool (q3) that may be due to the fact that not all the experts were exposed to it (it was implemented later in the process). A negative answer was also found on the question on the willingness to install TreeVersity on their own (q7), which is understandable for domain experts that are not necessarily computer experts. Answers also showed that experts found TreeVersity to be very useful (q2) and are willing to continue using it (q6). Moreover, experts found the StemView comprehensible (q4).

7. Conclusions

We described TreeVersity2, a novel web-based information visualization system to analyze change over time in a dataset. We demonstrated how the tool has been used successfully to analyze transportation data using two case studies. TreeVersity2 allows users to explore data that is inherently hierarchical (e.g. budget data) or not—by categorizing them by their attributes (e.g. publications). An interactive textual reporting tool complements the manual navigation when the amount of data is very large, helping users identify the largest differences using a set of standard metrics. The results of an exit questionnaire answered by ten domain experts who worked with TreeVersity2 suggest that users found TreeVersity2 potentially useful. Still we recognize that learning to use TreeVersity2 can be time consuming. Further work is needed to help first time or occasional users get started. Advanced features might be only accessible to expert users as professional analysts or regular users. The reporting tool was very well received and further research should continue to evaluate its benefits for first-time users as well as regular users.

7.1. TreeVersity availability

TreeVersity2 continues to be refined and strengthened at the CATT Lab, with the goal of making it available to the Lab's partners. To access a live demonstration please visit <http://treeversity.cattlab.umd.edu> (and use TreeVersity with preloaded sample datasets). To evaluate TreeVersity2 with your own data please contact the CATT Lab (<http://www.cattlab.umd.edu>).

Acknowledgements

We want to thank the Fulbright International Science and Technology Scholarship, the Center for Integrated Transportation Systems Management (a Tier 1 Transportation Center at the University of Maryland) and the Center for Advanced Transportation Technology Laboratory (CATT LAB) for partial support of this research. We also thank our case studies partners, Mike Bostock and Jason Davies for their work on and the Crossfilter library, and Audra Buck-Coleman for her thoughtful advice on graphical design.

References

- Amenta, N., Klingner, J., 2002. Case study: visualizing sets of evolutionary trees. In: *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*. IEEE Comput. Soc, pp. 71–74. doi: 10.1109/INFVIS.2002.1173150.
- Auber, D., Delest, M., Domenger, J.P., Ferraro, P., Strandh, R., 2003. EVAT: environment for visualisation and analysis of trees. *IEEE InfoVis Poster Compend.*, 124–125.
- Beecham, R., Wood, J., 2014. Characterising group-cycling journeys using interactive graphics. *Transp. Res. Part C Emerg. Technol.* 47, 194–206. <http://dx.doi.org/10.1016/j.trc.2014.03.007>.
- Bremm, S., von Landesberger, T., Hess, M., Schreck, T., Weil, P., Hamacher, K., 2011. Interactive visual comparison of multiple trees. In: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, pp. 31–40. doi: 10.1109/VAST.2011.6102439.
- Brodbeck, D., Girardin, L., 2003. Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree. In: *IEEE Symposium on Information Visualization 2003*, pp. 197–201. doi: 10.1109/INFVIS.2003.1249026.
- Card, S.K., Nation, D., 2002. Degree-of-interest trees: a component of an attention-reactive user interface. In: *Proc. of the Working Conference on Advanced Visual Interfaces*, pp. 231–245.
- Cheng, T., Tanaksaranond, G., Brunson, C., Haworth, J., 2013. Exploratory visualisation of congestion evolutions on urban transport networks. *Transp. Res. Part C Emerg. Technol.* 36, 296–306. <http://dx.doi.org/10.1016/j.trc.2013.09.001>.

- Fisher, D., Dhamija, R., Hearst, M., 2001. Animated exploration of dynamic graphs with radial layout. In: Proc. of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01), pp. 43–50. doi: 10.1109/INFVIS.2001.963279.
- Furnas, G.W., Zacks, J., 1994. Multitrees: enriching and reusing hierarchical structure. In: CHI '94. ACM, New York, NY, USA, pp. 330–336. <http://dx.doi.org/10.1145/191666.191778>.
- Graham, M., Kennedy, J., 2001. Combining linking and focusing techniques for a multiple hierarchy visualisation. In: Proceedings Fifth International Conference on Information Visualisation. IEEE Comput. Soc, pp. 425–432. doi: 10.1109/IV.2001.942092.
- Graham, M., Kennedy, J., 2005. Extending taxonomic visualisation to incorporate synonymy and structural markers. *Inf. Vis.* 4, 206–223.
- Graham, M., Kennedy, J., 2007. Exploring multiple trees through DAG representations. *IEEE Trans. Vis. Comput. Graph.*, 1294–1301.
- Graham, M., Kennedy, J., 2008. Multiform Views of Multiple Trees. In: Proc. 2008 12th Int. Conf. Inf. Vis., pp. 252–257. doi: 10.1109/IV.2008.21.
- Graham, M., Kennedy, J.B., Hand, C., 2000. A comparison of set-based and graph-based visualisations of overlapping classification hierarchies. In: Proceedings of the Working Conference on Advanced Visual Interfaces – AVI '00. ACM Press, New York, New York, USA, pp. 41–50. <http://dx.doi.org/10.1145/345513.345243>.
- Graham, M., Watson, M.F., Kennedy, J.B., 2002. Novel visualisation techniques for working with multiple, overlapping classification hierarchies. *Taxon* 51, 351–358.
- Guerra Gomez, J.A., Buck-Coleman, A., Plaisant, C., Shneiderman, B., 2011. TreeVersity: comparing tree structures by topology and node's attributes differences. In: 2011 IEEE Conference on Visual Analytics Science and Technology (VAST). IEEE, pp. 275–276. <http://dx.doi.org/10.1109/VAST.2011.6102471>.
- Guerra Gómez, J.A., Buck-Coleman, A., Plaisant, C., Shneiderman, B., 2012. Interactive visualizations for comparing two trees with structure and node value changes. In: Proc. Conference of the Design Research Society – DRS2012, pp. 1–10.
- Guerra Gómez, J.A., Buck-Coleman, A., Pack, M.L., Plaisant, C., Shneiderman, B., 2013a. TreeVersity: interactive visualizations for comparing hierarchical data sets. In: Proceedings of the 2013 Transportation Research Board Annual Meeting.
- Guerra Gómez, J.A., Pack, M.L., Plaisant, C., Shneiderman, B., 2013b. Visualizing change over time using dynamic hierarchies: TreeVersity2 and the StemView. *IEEE Trans. Vis. Comput. Graph.* 19, 2566–2575. <http://dx.doi.org/10.1109/TVCG.2013.231>.
- Heer, J., Card, S.K., 2004. DOTrees revisited: scalable, space-constrained visualization of hierarchical data. In: Proceedings of the Working Conference on Advanced Visual Interfaces – AVI '04. ACM Press, New York, New York, USA, p. 421. <http://dx.doi.org/10.1145/989863.989941>.
- Hong, J.Y., D'Andries, J., Richman, M., Westfall, M., 2003. Zoomology: comparing two large hierarchical trees. In: Poster Compend. IEEE Inf. Vis.
- Hughes, R.G., 2004. Visualization in transportation: current practice and future directions. *Transp. Res. Rec.*
- Hullman, J., Diakopoulos, N., Adar, E., 2013. Contextifier: automatic generation of annotated stock visualizations. In: Proceedings of the 31th ACM Conference on Human Factors in Computing Systems (CHI), pp. 2707–2716.
- Hurter, C., Conversy, S., Gianazza, D., Telea, A.C., 2014. Interactive image-based information visualization for aircraft trajectory analysis. *Transp. Res. Part C Emerg. Technol.* 47, 207–227. <http://dx.doi.org/10.1016/j.trc.2014.03.005>.
- Kamga, C., Yazıcı, M.A., 2014. Temporal and weather related variation patterns of urban travel time: considerations and caveats for value of travel time, value of variability, and mode choice studies. *Transp. Res. Part C Emerg. Technol.* 45, 4–16. <http://dx.doi.org/10.1016/j.trc.2014.02.020>.
- Kong, N., Agrawala, M., 2012. Graphical overlays: using layered elements to aid chart reading. *IEEE Trans. Vis. Comput. Graph.* 18, 2631–2638. <http://dx.doi.org/10.1109/TVCG.2012.229>.
- Kruskal, J.B., Landwehr, J.M., 1983. Icicle plots: better displays for hierarchical clustering. *Am. Stat.* 37, 162–168.
- Lamping, J., 1996. The hyperbolic browser: A Focus+Context technique for visualizing large hierarchies. *J. Vis. Lang. Comput.* 7, 33–55. <http://dx.doi.org/10.1006/jvlc.1996.0003>.
- Lee, B., Robertson, G.G., Czerwinski, M., Parr, C.S., 2007. CandidTree: visualizing structural uncertainty in similar hierarchies. *Inf. Vis.* 6, 233–246.
- Liu, K., Fei, X., 2010. A fuzzy-logic-based system for freeway bottleneck severity diagnosis in a sensor network. *Transp. Res. Part C Emerg. Technol.* 18, 554–567. <http://dx.doi.org/10.1016/j.trc.2009.11.002>.
- Mohammadi-Aragh, M.J., Jankun-Kelly, T.J., 2005. MoireTrees: visualization and interaction for multi-hierarchical data. In: EUROVIS'05 Proceedings of the Seventh Joint Eurographics/IEEE VGTC Conference on Visualization. Eurographics Association, Leeds, United Kingdom, pp. 231–238. doi:10.2312/VisSym/EuroVis05/231-238.
- Morse, D.R., Ytow, N., Roberts, D., Sato, A., 2003. Comparison of multiple taxonomic hierarchies using TaxoNote. In: Compendium of Symposium on Information Visualization, pp. 126–127.
- Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., Zhou, Y., 2003. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. *ACM Trans. Graph.* 22, 453. <http://dx.doi.org/10.1145/882262.882291>.
- Pack, M.L., 2010. Visualization in transportation: challenges and opportunities for everyone. *IEEE Comput. Graph. Appl.* 30, 90–96. <http://dx.doi.org/10.1109/MCG.2010.79>.
- Pack, M., Weisberg, P., Bista, S., 2005. Four-dimensional interactive visualization system for transportation management and traveler information. *Transp. Res. Rec.* 1937, 152–158. <http://dx.doi.org/10.3141/1937-21>.
- Parr, C.S., Lee, B., Campbell, D., Bederson, B.B., 2004. Visualizations for taxonomic and phylogenetic trees. *Bioinformatics* 20, 2997. <http://dx.doi.org/10.1093/bioinformatics/bth345>.
- Plaisant, C., Grosjean, J., Bederson, B.B., 1998. SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. In: IEEE Symposium on Information Visualization, 2002. (INFOVIS, 2002), pp. 57–64. doi: 10.1109/INFVIS.2002.1173148.
- Robertson, G.G., Mackinlay, J.D., Card, S.K., 1991. Cone trees. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Reaching through Technology – CHI '91. ACM Press, New York, New York, USA, pp. 189–194. <http://dx.doi.org/10.1145/108844.108883>.
- SAP Design Guild – UI Design Blinks (2012) [WWW Document], 2013. SAP Des. Guild – UI Des. Blinks. URL <http://www.sapdesignguild.org/community/blinks/ui_blinks_gw_03.asp#skyline> (accessed 5.1.12).
- Shneiderman, B., 1992. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transp. Graph.* 11, 92–99. <http://dx.doi.org/10.1145/102377.115768>.
- Spenke, M., 2001. Visualization and interactive analysis of blood parameters with InfoZoom. *Artif. Intell. Med.* 22, 159–172, doi: 16/S0933-3657(00)00105-6.
- Tu, Y., Shen, H.-W., 2007. Visualizing changes of hierarchical data using treemaps. *IEEE Trans. Vis. Comput. Graph.* 13, 1286–1293. <http://dx.doi.org/10.1109/TVCG.2007.70619>.
- Visualizations for taxonomic and phylogenetic trees [WWW Document], n.d. URL <http://bioinformatics.oxfordjournals.org/content/20/17/2997.short> (accessed 6.20.11).
- Wang, Z., Ye, T., Lu, M., Yuan, X., Qu, H., Yuan, J., Wu, Q., 2014. Visual exploration of sparse traffic trajectory data. *IEEE Trans. Vis. Comput. Graph.* 20, 1813–1822. <http://dx.doi.org/10.1109/TVCG.2014.2346746>.
- Zografos, K.G., Madas, M.A., 2006. Development and demonstration of an integrated decision support system for airport performance analysis. *Transp. Res. Part C Emerg. Technol.* 14, 1–17. <http://dx.doi.org/10.1016/j.trc.2006.04.001>.