CandidTree: Visualizing Structural Uncertainty in Similar Hierarchies

Abstract

Most visualization systems fail to convey uncertainty within data. To provide a way to show uncertainty in similar hierarchies, we interpreted the differences between two tree structures as uncertainty. We developed a new interactive visualization system called CandidTree that merges two trees into one and visualizes two types of structural uncertainty: location and sub-tree structure uncertainty. Since CandidTree can visualize the differences between two tree structures, we conducted a series of user studies with tree-comparison tasks. First a usability study was conducted to identify major usability issues and evaluate how our system works. Another qualitative user study was conducted to see if biologists, who regularly work with hierarchically organized names, are able to use CandidTree, and to assess the "uncertainty" metric we used. A controlled experiment with software engineers was conducted to compare CandidTree with WinDiff, a traditional files and folders comparison tool. The results showed that users performed better with CandidTree. Furthermore, CandidTree received better satisfaction ratings and all users preferred CandidTree to WinDiff.

Keywords: Uncertainty visualization, Structural uncertainty, Tree comparison, Graphical user interfaces, Evaluation

Introduction

Most current visualization systems suggest certainty. This means that when we show visualizations to users, they believe that what is currently displayed is ground truth. However, there are many cases where this is not true. For example, there often exist several biological taxonomies and phylogenetic trees for any group of organisms because not all biologists or analysis techniques agree on one taxonomy or on one phylogenetic tree. Current tree visualizations such as Hyperbolic Tree [11] and TaxonTree [13] typically show one taxonomy at a time without any certainty information, which often may not be easily computed. Hence, there is no way to see which parts of the tree are certain or uncertain. To address this problem, we interpreted the differences between two tree structures as uncertainty and developed a new interactive visualization system, called CandidTree [14] (see Figure 1), to visualize the differences.

One of the most common approaches to comparing two tree structures is to use paired side-by-side views, using coupled interaction to allow users to compare and navigate the two trees. This approach helps users identify where the differences are (usually by highlighting), but does not explicitly show the degree to which any two parts differ.

CandidTree merges two tree structures into one and computes two types of structural uncertainty for each node: 1) location of a node relative to its parent and 2) the sub-tree structure of a node. CandidTree represents these uncertainties with color and with transparency, respectively, so that users can easily identify which parts are most certain or uncertain. It also enables users to interactively explore the merged tree to investigate those parts in more depth. Furthermore, when users select a node, CandidTree shows paths to the root in each tree so that they can see how its absolute location differs in the two trees.

While CandidTree was originally developed to show structural uncertainty, it can also be applied to visualize the differences between two tree structures. For example, when we have two (backup) directory structures (for two different time points) containing backups for the same folder, CandidTree can help users find added, deleted, or moved files, in addition to modified folders. Furthermore, it enables users to identify which folder has been changed the most or least.

After reviewing related work, we explain how these two types of the structural uncertainty are defined. We describe how CandidTree visualizes them using a set of two classifications of scientific names. We also report three user studies we conducted to identify major usability issues and evaluate how our system works. We then conclude with future work.

Related Work

To provide a complete and accurate visual representation of data, it is important to show uncertainty within the data. Uncertainty has been very broadly defined to include concepts such as error, inaccuracy/imprecision, minimum-maximum ranges, data quality, and missing data [7][12][25][27][30]. For more than a decade, much research has described approaches to handling these various aspects of uncertainty [2][3][24][29][30][32].

The geographic visualization, geographic information science, and scientific visualization communities have given particular attention to uncertainty visualization and many techniques have been developed [4][18][19][20][25][27][31]. The main techniques used to visualize uncertainty include adding glyphs [16][35], adding geometry, modifying geometry [9], modifying attributes, animation [6][16], and sonification [17]. While these techniques have been applied to a variety of applications such as fluid flow, surface interpolants, and volumetric rendering, only a few of them have been evaluated. Furthermore, there has been very little research on visualizing uncertainty in tree structures. To our knowledge, only Griethe and Schumann proposed visual representations to represent uncertainty in parent-child relationship in structures [8]. For example, for node-link diagrams they used blurred or dotted links to indicate less certain relationships. However, they did not describe how to represent the degree of uncertainty. Moreover, while these authors brought up example applications for uncertainty visualization on structure information, their solutions were neither thoroughly investigated nor properly tested. In fact, it was beyond the scope of their paper to find effective metaphors in more challenging situations [8]. Since there is no formal definition of structural uncertainty, we propose two types of uncertainty for tree structures: location and sub-tree structure uncertainty, which will be explained in the following section.

An error can be defined as a difference between a computed, estimated, or measured value and the true or correct value. There are many cases where we do not know correct values but can estimate those using different techniques or algorithms. It is common to use the differences between two results as an error (or uncertainty). For example, Pang and Freeman visualized differences between 3D surfaces generated by various rendering algorithms [26]. In fact, side-by-side comparison is one of the most commonly applied existing uncertainty visualization methods [27]. Therefore, theoretically we can use these kinds of visualization tools to show uncertainty in tree structures.

In the biology domain, there often are several biological taxonomies and phylogenetic trees for a group of organisms because not all biologists or analysis methods agree on one taxonomy and one phylogenetic tree. In the case of phylogenies, the true tree (reflecting evolutionary history) cannot be known. Instead, there are trees that are thought to be more or less likely to be correct. Often analysis methods produce multiple trees with similar chances of being correct. To assess the quality of taxonomies and phylogenetic trees, it is important to understand which parts of two trees agree or disagree [23]. Areas of disagreement thus may be interpreted as uncertainty in the correctness of the tree at that point. One of the commonly used approaches to comparing two trees is to show paired views side-by-side, with coupled interaction. In fact, many submissions to the InfoVis 2003 contest, Visualization and Pair Wise Comparison of Trees (http://www.cs.umd.edu/hcil/iv03contest), used side-by-side views. For example, TreeJuxtaposer automatically matches nodes in two trees based on the shared ancestors, and highlights where the differences are [22]. InfoZoom transforms a tree into a tabular representation, in which each leaf is represented as a column and the path from the root is stored in the attributes (rows). It displays both trees (in a tabular form) side-by-side and marks the cells of differences [28]. Some visualizations provide a merged tree by combining two trees into one. For example, TaxoNote shows the merged tree on the left, the first tree at center, and the second tree on right [21]. It uses multiple tables to provide taxonomic names that are common or different. Zoomology also provides a single overview of the merged tree with the indication of the differences, and uses matched twin detail windows to show similarities and differences via a zoomable interface [10]. While these tools show where the differences are, they do not show the magnitude of the differences, and none are widely used by the scientific community.

Structural Uncertainty

As for the cases where we do not know the correct tree structures, we interpret the differences between two tree structures as uncertainty. For each node, we measure two types of uncertainty; location and sub-tree structure uncertainty.

Location Uncertainty

Within a tree structure, the location of a node can be represented in two different ways: 1) its absolute path from the root and 2) its relative path from its parent. The main drawback of the first representation is that a small difference close to the root would affect its whole sub-tree. Therefore, we decided to use the relative path to compute the uncertainty in node location. The location uncertainty is not scalar but categorical in value and three possible categories are as follows:

- A node is in both trees at the same location (i.e., under same parent) – most certain
- A node is in both trees but at different locations (i.e., under different parents)
- A node is included in only one of the trees – most uncertain

Sub-tree Structure Uncertainty

Whether or not a node is in the same location in two trees, its sub-tree structures can be different. We compute the sub-tree structure uncertainty by measuring how many links overlap in two sub-trees. So, the sub-tree structure uncertainty function for a node *v* can be defined as: Eq. (1).

$$\text{sub - tree structure uncertainty}(v) = 1 - \frac{n(L_1(v) \cap L_2(v))}{n(L_1(v) \cup L_2(v))} \tag{1}$$

where $L_i(v)$ is the set of links in the sub-tree of v in the ith tree.

CandidTree

We developed a visualization system, named CandidTree, to show the structural uncertainty described in the previous section. CandidTree automatically merges two tree structures into one and computes structural uncertainty based on the differences between them. As shown in Figure 1, it consists of two views: 1) a tree browser to show the merged tree and 2) a paths view to show paths to the root in each tree from the currently selected node.

To describe how CandidTree works in this section, we use a set of two classifications of scientific names of birds; one from the National Center for Biotechnology Information (NCBI, http://www.ncbi.nlm.nih.gov) and the other from the Animal Diversity Web (ADW, http://animaldiversity.ummz.umich.edu). The fact that these two authoritative sources disagree on these classifications illustrates the degree of uncertainty in biological classifications, which CandidTree helps expose.

Visualizing Structural Uncertainty

CandidTree shows the location uncertainty of a node relative to its parent by color (of the node label). The color black means that the node is included in both trees under the same node; 1 representation for the first category of location uncertainty described above. The colors red and blue indicate that the node is included in the first and second tree, respectively; 2 representations for the third category of location uncertainty. If the node is included in both trees but under different nodes, this means that the node moved from the first tree to the second one. To represent this move case, CandidTree shows the red node in the first tree with a strikethrough and the blue node in the second tree with an underline; 2 representations for the second category of location uncertainty. For example, “Megapodiidae” was under “Craciformes” in the first tree and then moved under “Galliformes” in the second tree (Figure 1).

CandidTree shows the sub-tree structure uncertainty of a node by transparency (of the node label). To make the node readable, even if uncertainty is very high, CandidTree uses 128 as a minimum alpha value (50% transparent). From the usability study, which will be described later, we learned that it was difficult to distinguish small differences in this manner (e.g., the difference between 1 and .9). To help users distinguish the 100% certain information from less certain data, CandidTree uses a solid link only when the sub-tree structure uncertainty is 0. For example, in Figure 1, among the children of “Megapodiidae,” the sub-tree structure uncertainty of “Leipoa” and “Macrocephalon” is 0 and that of the others is non-0.

We also decided to use four discrete alpha values; 1) 255 (0% transparent) when u (uncertainty) = 0, 2) 214 (≈ 16%) when, 0 < u < .5, 3) 171 (≈ 33%) when.5 <= u < 1, and 4) 128 (50%) when u = 1. To enable users to compare uncertainties in the same range, CandidTree provides exact values in a tooltip. The default set of alpha values means that the more certain the data, the more opaque (and readable) it is. However, from the usability study we also learned that users may want to focus on uncertain (different) information depending on data and tasks. CandidTree reverses the order of alpha values when users check the “Highlight Changes” check box (bottom right of Figure 1) to make more uncertain information more readable.

Children of each node are grouped by their representations for the location uncertainty; 1) nodes only in the first tree, 2) nodes moved from the first tree, 3) nodes under the same parent in both trees, 4) nodes moved to the second tree, and 5) nodes only in the second tree. This helps users capture only one tree from the merged tree even without un-highlighting the other tree. For example, if users want to focus on the first tree, they can ignore group 4) and 5). Within each group children are sorted from highest to lowest sub-tree structure uncertainty.

Changes in Paths to the Root

When users click on a moved node (either red with strikethrough or blue with underline) in the tree browser, CandidTree finds the matching node in the other tree and opens them together. The selected node is indicated by a light blue background with a rectangular border and the matching node with an oval border. In the example of Figure 1, when users click on “Megapodiidae” (in the first tree) under “Craciformes,” “Megapodiidae” (in the second tree) under “Galliformes” is automatically opened.

CandidTree’s tree browser and paths view are tightly coupled. So, when users select a node in the tree browser, paths to the root in both trees from the selected node are shown in the paths view. If the selected node is included in only one tree or its absolute location is the same in both trees, only one path is shown. When two paths are different, CandidTree vertically aligns the nodes with similar labels from two paths; the similarity of the labels is computed by Levenshtein distance [15]. This helps users see what changes are made between two trees. For example, in Figure 2, two levels – “Neognathae” and “Neoaves” – are added to the second tree between “Aves” and “Strigiformes”. This could also help users identify possible errors in node labels. For example, the parent of

“Musophagidae” is supposed to be “Musophagiformes” as in the first tree and “Musphagiformes” in the second tree is a typographical error (Figure 3).

When users click on a node in the paths view, CandidTree temporarily highlights the corresponding node in the tree browser with a thick purple rectangle surrounding the node so that users can recognize where the node is. In addition, if the corresponding node is off-screen, CandidTree pans the tree to view the node.

Emphasizing Nodes of Interest and Search

As users browse through a tree, especially for trees with large fan outs, the selected node and its matching node could be very far away from each other and the matching node could be off-screen. Even when there is no matching node, the screen could still be cluttered. We assume that users are mainly interested in the selected node and its children, siblings, and direct ancestors. So, when users select a node, CandidTree uses a fisheye technique [5] and deemphasizes all other nodes by making them smaller and less opaque (Figure 1). If it exists, the same rule is applied to the matching node of the selected node.

For the cases where users are only interested in the nodes with a specific certainty range, we provide a double-headed range bar (bottom center of Figure 1) in the control panel to allow users to un-highlight nodes that do not meet the certainty requirement. Users can focus on one tree by manipulating the check boxes in the Trees list (bottom left of Figure 1); the unchecked tree is un-highlighted. Since 128 (50% transparent) was used as an alpha value for the most uncertain nodes for readability, CandidTree uses 64 as an alpha value (75% transparent) for un-highlighted nodes.

CandidTree provides support for search; it performs simple substring matching with node labels. Typing a word and pressing the “Go” button displays the search results colored in orange and restricts the view to the nodes relevant to the search results (Figure 4).

Implementation Details

CandidTree is implemented in C# with Piccolo.NET, a shared source toolkit that supports scalable structured 2D graphics [1] (http://www.cs.umd.edu/hcil/piccolo). It uses a classical tree layout by Walker [33].

CandidTree reads data from files in an xml format. Each node is represented with a “node” element having two attributes; “id” and “name.” The id attribute, which should be unique, serves as an identifier to be used to match nodes in two trees. The name attribute is used as a label of the node. The current implementation can be extended to show other node attributes or handle other data formats.

CandidTree builds the merged tree in memory and computes uncertainty at startup. It loads the first tree and then merges the second tree with the first one. The location uncertainty is computed during this merge process. Once the merged tree is built, CandidTree computes the sub-tree structure uncertainty using the equation described in the previous section, after recursively counting the number of links in the sub-trees of each node.

As mentioned above, CandidTree uses the id of the node to match nodes in two trees. To visualize the differences between two (backup) directory structures, we wrote a simple program to build a data file that CandidTree can process by scanning a file system. Assigning file IDs was problematic. We could not use the file name as an ID because it is not unique. We also could not simply use the full path because it is changed when a file is moved from one location to the other. So, after we set the full path as an ID, we had to post-process the second tree to assign the right IDs to moved files. For each file in the second directory structure, we first checked if there was a file with the same name and same size. If it existed, we compared the contents of two files (one from the first directory structure and the other from the second one). If they were exactly the same, we replaced the ID of the second file with that of the first one.

Evaluation

We conducted three user studies with tree-comparison tasks to evaluate how our system works. The first usability study helped us improve the CandidTree design before the second and third study, which will be described later in this section.

Study 1: Usability Study with Computer Scientists

To identify major usability issues, we conducted a preliminary usability study with six participants: two researchers, three developers, and one research intern (all male computer scientists). We used two sets of backup directory structures, each with two trees representing the data at two different time points. Trees in these sets contained about 150 nodes and 500 nodes.

Participants were given a brief tutorial of the system for up to 15 minutes including the time to play with the system and ask questions. Next, they were asked to perform 10 tasks (1-5 for the small tree and 6-10 for the large tree) with the system, which were timed and scored for correctness. The tasks were meant to cover major tree-comparison tasks and to evaluate the usability of the main features that represent structural uncertainty. All participants were asked to perform the tasks as quickly and accurately as possible. Once they completed all of the tasks, participants were asked to fill out a satisfaction questionnaire. Each session lasted about 30 minutes and participants were given a $5 snack coupon for their participation. The list of tasks follows:

1) How many files were deleted from the "DynaVis" folder?

2) Among the sub-folders of "DynaVis," which folder has been changed the most? (We asked users to ignore added or deleted folders.)

3) Describe the changes made to the "DynaVis\DynaTestWin\obj" folder.

4) Which file was moved from the "DynaVis\DynaVis\obj\Debug" folder?

5) Where did the file move to?

Tasks 6-10 were equivalent to tasks 1-5 but applied to a large tree. Tasks 7 and 9 asked participants to count the number of folders/files.

There were only 6 incorrect answers provided out of 60 questions across participants. Three participants answered all questions correctly and the other three each answered incorrectly one, two, and three questions, respectively. Task 2 (and 7 for the large tree) got the most wrong answers because participants had difficulty distinguishing between the colors black and green (green was the color of the second tree at the time of experiment). Two participants each gave incorrect answers to Task 4 and 9, respectively. Overall, average task times were fast. While Task 8 took longer than the others (46.5 seconds on average), it was primarily because it takes time to describe all the changes, not because participants had a hard time finding the changes.

Table 1 shows the average satisfaction ratings on a 7-point Likert scale, with 1=Disagree and 7=Agree. There was clear user frustration around the use of transparency to represent the sub-tree structure change. This is related to the readability issue participants raised, as mentioned earlier. The ratings are fairly consistent with the usability issues we identified. We here summarize the major iterations made to CandidTree based on the first user study (Figure 5):

1) Do not use the fourth color for the moved nodes; instead use strikethrough and underline.

2) Use solid links only when the uncertainty is 0; otherwise use dotted links.

3) To improve readability of the node labels, make the direct ancestors of the selected node opaque even though their sub-tree structure uncertainty is not 0.

4) Use four discrete alpha values.

5) Provide an option to highlight changed (uncertain) information. Furthermore, by default, CandidTree highlights (makes more opaque) changed information.

6) Provide a legend to show the color scheme.

Study 2: Qualitative User Study with Biologists

We conducted a qualitative study with biologists who regularly work with hierarchically organized names, with two main goals. First, we wanted to know if these users would be able to use CandidTree to correctly and quickly complete tree-comparison tasks, and we also wanted to know which tasks would pose more difficulty for users. Secondly, we wondered if CandidTree would support advanced information understanding and insight. Gaining insight from data does not lend itself easily to the metrics typically used in quantitative studies. Finally, we were interested in an assessment of the “uncertainty” metric we used.

Participants

We recruited 8 biologists (3 females and 5 males, 28 to 59 years old) from the Smithsonian Institution and University of Maryland. They included two graduate students, two post-doctoral fellows, and four curators or research faculty. All were unfamiliar with the testing datasets, though all had previously used data from the same source. Two were familiar with the tutorial data. Three mentioned regularly working with datasets of more than 150 terminal taxa (leaves); typical datasets include between 40 and 150 leaves (median 143). However, several of the biologists were associated with NSF projects dealing with trees of names of thousands of organisms and two participants mentioned that CandidTree might be useful for those projects. Each participant was given a $20 Amazon.com gift certificate for his/her participation.

Datasets

For both demonstration and testing we used classifications of scientific names from the National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov/) that were downloaded on different dates: December 14, 2005 and September 19, 2006. Tutorial data were from the Lepidoptera branch (moths and butterflies; tree 1=4,103 nodes, tree 2=6,262 nodes), while the test data were from the Aves branch (birds; tree 1=6,912 nodes, tree 2=8,140 nodes).

Procedure

Each participant filled out a background survey. They received a demonstration of CandidTree features and were told to freely explore and ask questions for a total tutorial time of up to 20 minutes. The search feature was not described or tested, nor were default settings for transparency or filtering changed or described. As described above, CandidTree was changed to show uncertain (more changed) information more opaque by default after the first usability study. Therefore, nodes that were more opaque were those with the highest uncertainty scores. Participants who asked were told how uncertainty was calculated. Participants were then asked to perform eleven tasks, described below. Participants then completed the same preference survey as in the usability study (Table 1). We videotaped the computer screen throughout the tutorial and testing. Each session lasted 45 to 60 minutes.

Tasks

Biologically meaningful tasks (Table 2) were chosen based on 30-minute interviews with three biologists (one of whom was subsequently a participant). They were presented in order of increasing complexity. The first nine had single, correct answers. Task 10 was judged by the number of insights given by the participant, and Task 11 was an opinion.

Results

Overall ease of use improved slightly over the usability study (Table 1). Still, transparency as an indicator of the sub-tree structure change scored particularly low. “Color representing the node location change was clear” and “It was easy to read the labels of the nodes” also received relatively low scores. As suggested by the accuracy results below, this may be because our use of strikethrough in addition to color confused some users.

Of the 72 possible answers (8 participants x 9 tasks), only 8 were incorrect (Table 2). The oldest participant had the most difficulty, answering incorrectly for 3 out of the 5 most complex tasks. Task 4, which required understanding the coding of location uncertainty, was answered incorrectly most often. Task 7, which required an understanding of whether the location change was up or down a level, was answered incorrectly by two participants but they both admitted a lack of concern about the utility of biological ranks.

Coding of location uncertainty continued to be problematic despite the iteration and the addition of a legend. Several participants answered question 4 incorrectly at first and then corrected themselves after glancing at the legend (they were counted as correct answers). Those who missed it gave all red names as answers instead of just the names in red with strikethrough. One participant remarked that she expected the two red codes to be additive: “It implies a hierarchy but in fact they signify different rather than nested ideas.”

Average times to complete biologically relevant tasks were somewhat longer than in the usability study (Table 2). Some participants took 1 to 2 minutes for Task 1, either to orient to the testing protocol (giving answers verbally) or to the sort order of uncertainty. Otherwise, the simplest tasks each typically took less than 30 seconds. For task 10, most participants explored for 3 to 6 minutes before being satisfied they had given a good summary.

For task 10, participants reported 3 insights (the participant with the least domain expertise who spent only 110 seconds exploring) to 9 insights (participant 4, who systematically explored for 6 minutes). Many participants made the same insights; still, pooling across participants, 18 distinct insights were reported. Five of the 8 participants found a typographical error correction that required a correct interpretation of transparency, and 3 found a subtle location change that required understanding of location coding and the paths view. The example in Box 1 shows how Participant 4 built his insights in steps while exploring CandidTree. Participants also reported some insights that were unprompted during both the tutorial and test. For example, participants mentioned that the datasets were obviously large, had few changes, were different from a dataset they were familiar with, and had a sub-tree with lots of "problem children."

Box 1. Participant 4 used CandidTree to make the following insights. Insight 9 builds on 6 and 7 which build on 5 which builds on 4.

Furnariidae sub-tree

1. Immediately identifies that there has been a spelling change between tree 1 and 2 [requires opening the least certain sub-tree]
2. Counts 17 new genus-level nodes added
3. Finds in a sub-tree that there has been a new species added

Sylviidae sub-tree

4. Three nodes in tree 2 are not in tree 1
5. Two of these are entirely new subfamilies [judged by interpreting the label]
6. They contain both taxa that have been moved here from the first tree
7. . . . and also include some taxa entirely new to the second tree
8. Elsewhere, some new subspecies have been added that were not in the first tree
9. Says, "Basically, taxa which had not previously been in subfamilies [a particular rank in the hierarchy] were moved into Acrocephalinae"

For task 11, six of 8 participants felt that sub-tree Sylviidae was more uncertain than the other. However, though the certainty scores were very close, half of the participants thought Sylviidae was much more uncertain because the changes involved numerous rearrangements and addition of internal nodes rather than simple addition of nodes at the leaves. Two participants thought it was not reasonable to compare them because the kinds of uncertainty were so different.

Consistent with the preference survey, the most common complaint was that transparency differences were too subtle to be usable. Also, two participants thought the more transparent names should be those with less sub-tree certainty. However, generally high task performance shows that these issues did not pose significant problems. Two participants thought that the colors should be reversed (red should represent the second, more recent or important tree). Several had trouble managing opening and then closing sub-trees and two suggested it would be useful to have a way to open or close one level of children all at once across the whole tree.

Participants offered many ideas for additional features or applications. Three participants wanted to use CandidTree to compare more than two trees, particularly with particular scientific datasets and websites such as NCBI or Tree of Life (http://www.tolweb.org). Some expressed interest in linking nodes to further information such as the GenBank sequence or host plants.

Study 3: Controlled Experiment

We conducted a controlled experiment to see if CandidTree could outperform a classic tree comparison tool. Since our target participants for this study were software engineers we decided to compare CandidTree with WinDiff [34] (Figure 6), a classic files and folders comparison tool. We used a 2 × 2 (two Interfaces with two Sizes of the trees) repeated measure, within-subject design. Task completion time, error rate, and user preference were collected in this study. We used a PC running Windows XP (3.0GHz Pentium 4 with 2GB RAM) with a 21.4'' LCD monitor at 1600 × 1200 resolution.

Participants and Datasets

We recruited twelve participants (2 females and 10 males) and one male pilot tester internally from Microsoft. They were already familiar with WinDiff, though some of them had used it mainly for file comparison rather than directory structure comparison. Each participant was given a $10 lunch coupon for his/her participation.

As with the usability study with computer scientists, we used two backup directory structures (DynaVis and Polyarchy in Figure 7), each with two trees representing the data at different time points. We replicated this twice (Set1 and Set2 in Figure 7) for a total of eight trees. Trees in these sets contained about 150 nodes (small trees; DynaVis in Figure 7) and 500 nodes (large trees Polyarchy in Figure 7). To control for the effect of order and learning, we also counterbalanced the order of presentation of the interfaces and the set of trees. Since we tweaked the trees so that we could generate tasks with similar difficulties, two tree structures from both sets to represent the same directory structure were slightly different. For example, two tree structures from Set1 and Set2 for the 1.0 folder under Polyarchy were not the same even though they refer to the same folder (Figure 7). We also prepared another set of backup directory structures for training. Trees in this set contained about 120 nodes. We kept constant the order of sizes (from small to large) and the order of tasks.

Procedure

Each participant used both interfaces. Participants received a tutorial of the first interface if it was CandidTree. We did not give training for WinDiff because the participants were already familiar with it. However, they were told to freely explore and ask questions for both interfaces. For the CandidTree interface, as with the qualitative study with biologists, the search feature was not described or tested. While we described default settings for transparency or filtering, they were not changed or tested. So, again nodes that were more opaque were those with the highest change (uncertainty) scores. For the WinDiff interface, participants were allowed to change options so that they can have their best configurations.

Participants were then asked to perform ten tasks, which were timed and scored for correctness. They were the same types of tasks that we used for the first usability study with computer scientists; tasks 1-5 were for the small trees and tasks 6-10 were for the large trees. Each task had a 3-minute time limit and participants were allowed to give up a task at any time. Since tasks 5 and 10 are dependent on tasks 4 and 9, respectively, when participants missed tasks 4 and 9, we gave participants the correct answer for those tasks so that they could still complete tasks 5 and 10.

As described before, CandidTree represents a file move using a red strikethrough (for the source file) and a blue underline (for the destination file). However, WinDiff shows file move as deletion and addition; the source file is removed and the destination file is added. This means that it is impossible to complete four tasks (4, 5, 9, and 10) by using only WinDiff. Therefore, for the WinDiff interface, we allowed participants to use additional tools such as Windows Explorer, command line prompt, and Windows desktop search to browse folders and find files.

Once participants completed all tasks for the first interface, they completed the subjective satisfaction questionnaire. While we used the same set of questions that we used for the previous two studies, we rephrased two questions (6 and 7; see Table 1) so that we could use them for WinDiff as well. After a short break, the same procedure was repeated with the second interface. Preferences, comments, and suggestions were collected during debriefing. Each session lasted about 60 minutes.

Results

We present the results from the study in three parts; Error, completion time, and subject preference. One participant answered incorrectly eight tasks with WinDiff. Since he answered incorrectly all five tasks for the small tree we could not include his data for analysis even though he answered correctly all ten tasks with CandidTree. The pilot data was also not included, because we slightly modified the description of the tasks after the pilot.

Participants made significantly fewer errors using CandidTree ($\chi^2(1)=31.22$, $p<.001$). While there were only 3 incorrect answers for CandidTree provided out of 110 questions across participants, there were 34 incorrect answers (including 2 timeouts and 5 give-ups) for WinDiff. For both interfaces, Task 2 got the most incorrect answers (2 for CandidTree and 9 for WinDiff). For CandidTree it was because some participants had difficulty distinguishing between the colors black and blue. For WinDiff, it was too difficult to measure the degree of structural changes in a subdirectory. We were surprised to see that 6 out of 11 participants (for small trees) gave wrong answers for the simplest task: count added/deleted files in a folder. It was also interesting to see that participants gave far fewer wrong answers for large trees (28 for small trees and 6 for large trees). We suspect that, in addition to a learning effect, this is because tasks 7 and 9 asked participants to count the number of folders/files instead of finding the specific one. For example, it seemed to be easy to identify whether the folder was changed but not easy to identify the most/least changed one among the changed folders.

For completion time analysis, we did not include the times with incorrect answers. We performed 2 (Interface) x 2 (Size) repeated-measures analysis of variance (RM-ANOVA) and found a significant main effect of Interface that favored CandidTree, $F(1, 40)=27.00$, $p<.0001$. Contrary to our expectation, results showed no significant differences in completion times between sizes. There was no interaction effect either. Figure 8 shows the median task completion time for each task for both interfaces.

When asked which interface they preferred overall, all participants chose CandidTree over WinDiff. Figure 9 shows the average Likert scale satisfaction ratings by question for both interfaces, where 1=Disagree and 7=Agree. We performed individual two-tail t-tests for each of the seven questions and corrected them using the Bonferroni technique. We found a significant effect of Interface favoring CandidTree for four questions (Q1, Q4, Q6, and Q7). We also found a marginally significant effect of Interface that favored CandidTree for Q3.

Discussion

In contrast to the second study (the qualitative study with biologists), overall user satisfaction for CandidTree dramatically improved over the usability study (see Table 1). Furthermore, CandidTree was rated fairly highly for two representations for structural changes (uncertainties).

1) 6.9 (out of 7) for the question 6: It was easy to identify the node location change

2) 6.6 for the question 7: It was easy to identify the sub-tree structure change

We were surprised to observe this because we did not make any changes in CandidTree after the second study and major characteristics of tasks between two studies are similar. However, differences between datasets may explain this result. The use of strikethrough was higher in the second study because the animal classification datasets had more moved nodes among directories. We suspect that strikethrough increased the difficulty in reading node labels.

Several participants, who often need to compare and merge two tree structures, liked the features to identify structural change of directories and to let users show where files were moved to. They also pointed out that WinDiff was difficult and cumbersome to use for that purpose because it is more for file comparison than directory comparison.

There is still room for improvement. Two participants gave us feedback concerning the readability of red nodes with strikethrough. One way to alleviate this issue would be to use underline instead of strikethrough for red nodes. This may make our encoding more coherent; moved nodes are represented with an underline. Several participants wanted to sort the children alphabetically so that they can find a particular node more easily. Also, some participants mentioned that the transparency of the text was too subtle to tell the difference between nodes without actually hovering over them to get the actual numbers.

Limitations and Future Work

We compared CandidTree with WinDiff because it is one of the most commonly used files and folders comparison tools and uses one merged view to show the differences between two file structures. However, it is a simple text-based tool and more useful for file comparison. Since the comparison was conducted as a controlled lab study, the results could have been affected by a number of variables such as application domains, datasets, and tasks. We believed that it is better to enable users to focus on one merged tree rather than going back and forth between two trees. It would be interesting to see if CandidTree could perform better than side-by-side tree comparison tools. Further comparisons with other tree comparison tools such as TreeJuxtaposer using other datasets including phylogenetic trees would shed more light on the benefits of CandidTree.

While the current implementation works only for two trees, we can handle more than two trees by providing the list of possible combinations of multiple trees and showing only one combination at a time. By ranking each combination based on the sub-tree structure uncertainty of the root node, we could enable users to easily identify most certain/uncertain (similar/dissimilar) tree combinations. As mentioned before, CandidTree loads each entire tree and builds the merged tree in memory, which is impractical for large trees. We can preprocess building of the merged tree and the computing structural uncertainty, and store them in a database. By accessing the data from a database when needed, CandidTree can be scaled up to support very large trees and with multiple attributes. We could also make the definition of structural uncertainty more sophisticated by incorporating node attributes of the trees being compared. For example, in phylogenetic trees, bootstrap values or decay indices are measures of how well each node in a tree is supported by data. These could be used to weight our measure of uncertainty so that differences in poorly supported sub-tree structures are downweighted relative to differences in nodes that have good support in both trees. This would take into account uncertainty within the trees as well as between the trees.

Conclusion

We proposed two types of uncertainty for tree structures – location and sub-tree structure uncertainty – based on the differences between them. To visualize those structural uncertainties, a new interactive visualization system called CandidTree was developed. Since CandidTree computes uncertainty by comparing two trees, we were able to apply it to visualize the differences between them. For example, CandidTree helps users find added, deleted, or moved files as well as modified folders within two (backup) directory structures containing backups for the same folder.

We conducted three user studies to identify major usability issues and to evaluate how well our system works. The first preliminary usability study with six computer scientists helped us fix major usability problems. Our qualitative study with biologists showed that while we have improved the uncertainty representation so that task performance and insight-building is high even with large trees, ways to improve satisfaction are needed. Also, while most users concur with relative uncertainty scores, there is not universal agreement on how to weigh the kinds of uncertainty. We also conducted a controlled experiment with computer scientists to compare CandidTree with WinDiff, a traditional files and folders comparison tool to see whether users could perform better with CandidTree. Results showed that users performed better in terms of task completion time and error with CandidTree. Furthermore, CandidTree received better satisfaction ratings and all users preferred CandidTree to WinDiff.

Acknowledgement

We would like to thank the participants of our three user studies for their participation and comments. Charlie Mitter and Ashleigh Smythe helped define biologist tasks as did Nathan Edwards who also provided NCBI data. Danyel Fisher reviewed our paper and gave thoughtful comments.

References

1. Bederson, B.B., Grosjean, J., and Meyer, J. Toolkit Design for Interactive Structured Graphics, *IEEE Trans. on Software Engineering* 2004; **30**(8): 535-546.

2. Cleveland, W.S. *The Elements of Graphing Data*; 1985.

3. Eaton, C., Plaisant, C., and Drizd, T. *Visualizing Missing Data: Graph Interpretation User Study*, Proc. of Interact 2005, Lecture Notes in Computer Science, Vol. **3585**. Springer-Verlag, Berlin Heidelberg New York; 851-872.

4. Fegeas, R.G., Cascio, J.L., and Lazar, R.A. An Overview of FIPS 173, The Spatial Data Transfer Standard, *Cartography and Geographic Information Systems* 1992; **19**(5): 278-293.

5. Furnas, G.W. *Generalized Fisheye Views*, Proc. of CHI 1986, ACM Press; 16-23.

6. Gershon, N.D. *Visualization of Fuzzy Data using Generalized Animation*, IEEE Symposium on Visualization 1992, IEEE Computer Society Press: Chicago; 268-273.

7. Gershon, N.D. Visualization of an Imperfect World, IEEE Computer Graphics and Applications 1998, **18**(4), 43-45.

8. Griethe, H. and Schumann, H. *The Visualization of Uncertain Data: Methods and Problems*, Proc. of SimVis 2006, 143-156.

9. Grigoryan, G. and Rheingans, P. Point-based probabilistic surfaces to show surface uncertainty, IEEE Transactions on Visualization and Computer Graphics 2004, **10**(5), 564-573.

10. Hong, J.Y., D'Andries, J., Richman, M., and Westfall, M. *Zoomology: Comparing Two Large Hierarchical Trees*, Posters Compendium of Information Visualization 2003, 120-121.

11. Hyperbolic Tree for the Green Tree of Life, http://ucjeps.berkeley.edu/TreeofLife/hyperbolic.php (accessed 4 April 2007).

12. Johnson, C.R. and Sanderson, A.R. A Next Step: Visualizing Errors and Uncertainty, IEEE Computer Graphics and Applications 2003, **23**(5), 6-10.

13. Lee, B., Parr, C.S., Campbell, D., and Bederson, B.B. *How Users Interact with Biodiversity Information using TaxonTree*, Proc. of Advanced Visual Interfaces 2004, ACM Press; 320-327.

14. Lee, B., Robertson, G., Czerwinski, M., and Parr, C.S. *CandidTree: Visualizing Structural Uncertainty in Similar Hierarchies*, Proc. of Interact 2007, LNCS 4663, 250-263.

15. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals, *Doklady Akademii Nauk SSSR* 1965; **163**(4): 845-848 (Russian). English translation in *Soviet Physics Doklady* 1966; **10**(8): 707-710.

16. Lodha, S.K., Pang, A., Sheehan, R.E., and Wittenbrink, C.M. *UFLOW: Visualizing Uncertainty in Fluid Flow*, IEEE Symposium on Visualization 1996, IEEE Computer Society Press: Chicago; 249-255.

17. Lodha, S.K., Wilson, C.M., and Sheehan, R.E. *LISTEN: Sounding Uncertainty Visualization*, IEEE Symposium on Visualization 1996, IEEE Computer Society Press: Chicago; 189-196.

18. MacEachren, A.M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., and Hetzler, E. Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know, *Cartography and Geographic Information Science* 2005; **32**: 139-160.

19. Moellering, H. Continuing Research Needs Resulting from the SDTS Development Effort, *Cartography and Geographic Information Systems* 1994; **21**(3): 180-189.

20. Morrison, J. The Proposed Standard for Digital Cartographic Data, *American Cartographer* 1988; **15**(1): 9-140.

21. Morse, D.R., Ytow, N., Roberts, D.M., and Sato, A. *Comparison of Multiple Taxonomic Hierarchies Using TaxoNote*, Posters Compendium of Information Visualization 2003, 126-127.

22. Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L., and Zhou Y. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility, Proc. of SIGGRAPH 2003, published as *ACM Transactions on Graphics* 2003; **22**(3): 453-462.

23. Nye T.M.W., Lio, P., and Gilks, W.R. A Novel Algorithm and Web-based Tool for Comparing Two Alternative Phylogenetic Trees, *Bioinformatics* 2006; **22**(1): 117-119.

24. Olston, C. and Mackinlay, J.D. *Visualizing Data with Bounded Uncertainty*, IEEE Symposium on Information Visualization 2002, IEEE Computer Society Press: Chicago; 37-40.

25. Pang, A. *Visualizing Uncertainty in Geo-Spatial Data*, Proc. of the Workshop on the Intersections between Geospatial Information and Information Technology.

26. Pang, A. and Freeman, *A. Methods for Comparing 3D surface Attributes*, Proc. of SPIE- VDA 1996, 58-64.

27. Pang, A.T., Wittenbrink, C.M., and Lodha, S.K. Approaches to Uncertainty Visualization, *The Visual Computer* 1997; **13**(8): 370-390.

28. Spenke, M. and Beilken, C. *Visualization of Trees as Highly Compressed Tables with InfoZoom*, Posters Compendium of Information Visualization 2003, 122-123.

29. Sulo, R., Eick, S., and Grossman, R. *DaVis: A Tool for Visualizing Data Quality*, Posters Compendium of Information Visualization 2005.

30. Taylor, B.N. and Kuyatt C.E. Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, NIST Technical Note 1297, 1994.

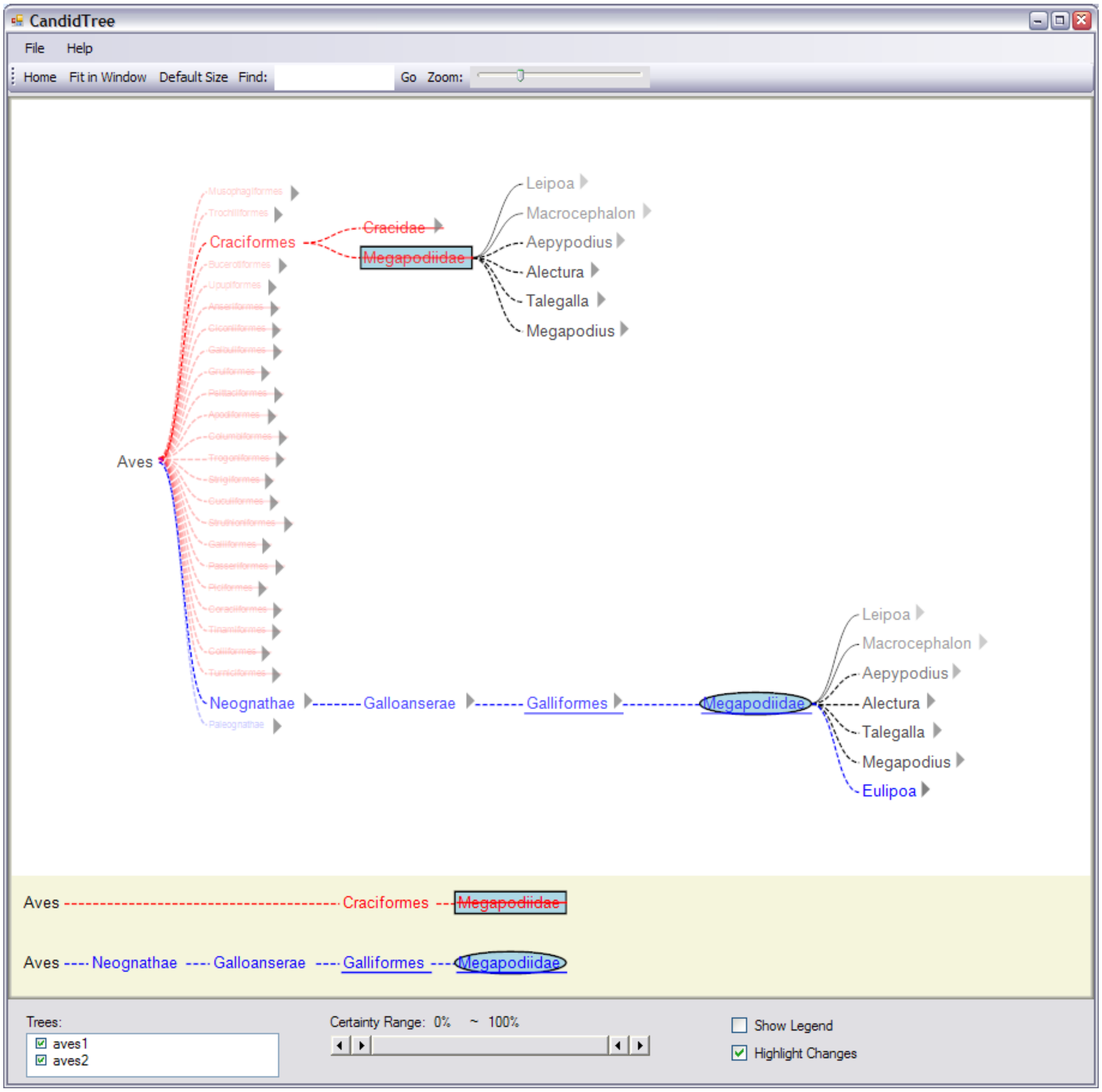
31. Thomson, J., Hetzler, B., MacEachren, A., Gahegan, M., and Pavel, M. *A Typology for Visualizing Uncertainty*, Proc. SPIE-VDA 2005, 146-157.

32. Tukey, J.W. *Exploratory Data Analysis*, 1977.

33. Walker II, Q. A Node-Positioning Algorithm for General Trees, *Software-Practice and Experience* 1990; **20**(7): 685-705.

34. WinDiff, http://technet2.microsoft.com/WindowsServer/en/library/ac1ff104-dde3-43ea-98af-e5cf5e513d4f1033.mspx

35. Wittenbrink, C.M, Pang, A.T., and Lodha, S.K. Glyphs for Visualizing Uncertainty in Vector Fields, *IEEE Trans. on Visualization and Computer Graphics* 1996; **2**(3): 266-279.



Aves Craciformes ~~Megapodiidae~~

Aves Neognathae Galloanserae Galliformes Megapodiidae

Trees:
☑ aves1
☑ aves2

Certainty Range: 0% ~ 100%

☐ Show Legend
☑ Highlight Changes

Figure 1 CandidTree shows two types of structural uncertainty: location and sub-tree structure uncertainty, respectively, with color and transparency so that users can easily identify which parts are most certain or uncertain. CandidTree merges two trees into one and shows the merged tree in the tree browser at the top. The view of the paths at the bottom shows paths to the root in each tree from the currently selected node.

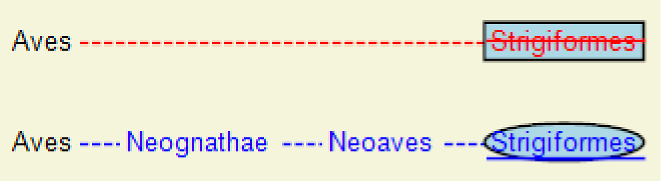


Figure 2 CandidTree shows paths to the root in each tree from the selected node and its matching node (if exists) in the paths view.

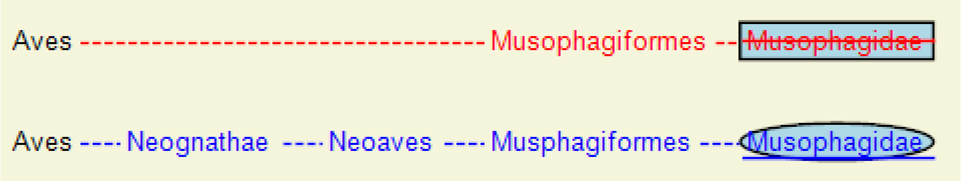


Figure 3 CandidTree aligns nodes with similar labels from two paths. “Musophagiformes” is aligned with “Musphagiformes” because it is the most similar one among three nodes (“Neognathae,” “Neoaves,” and “Musphagiformes”) from the second tree.

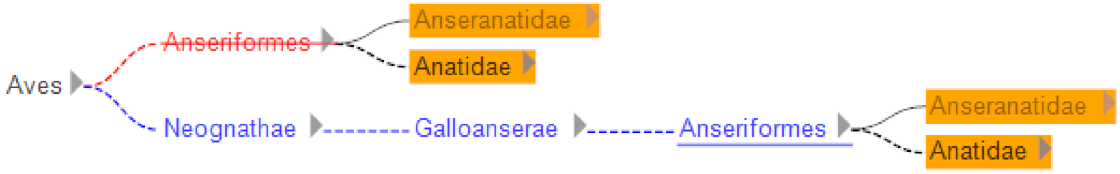
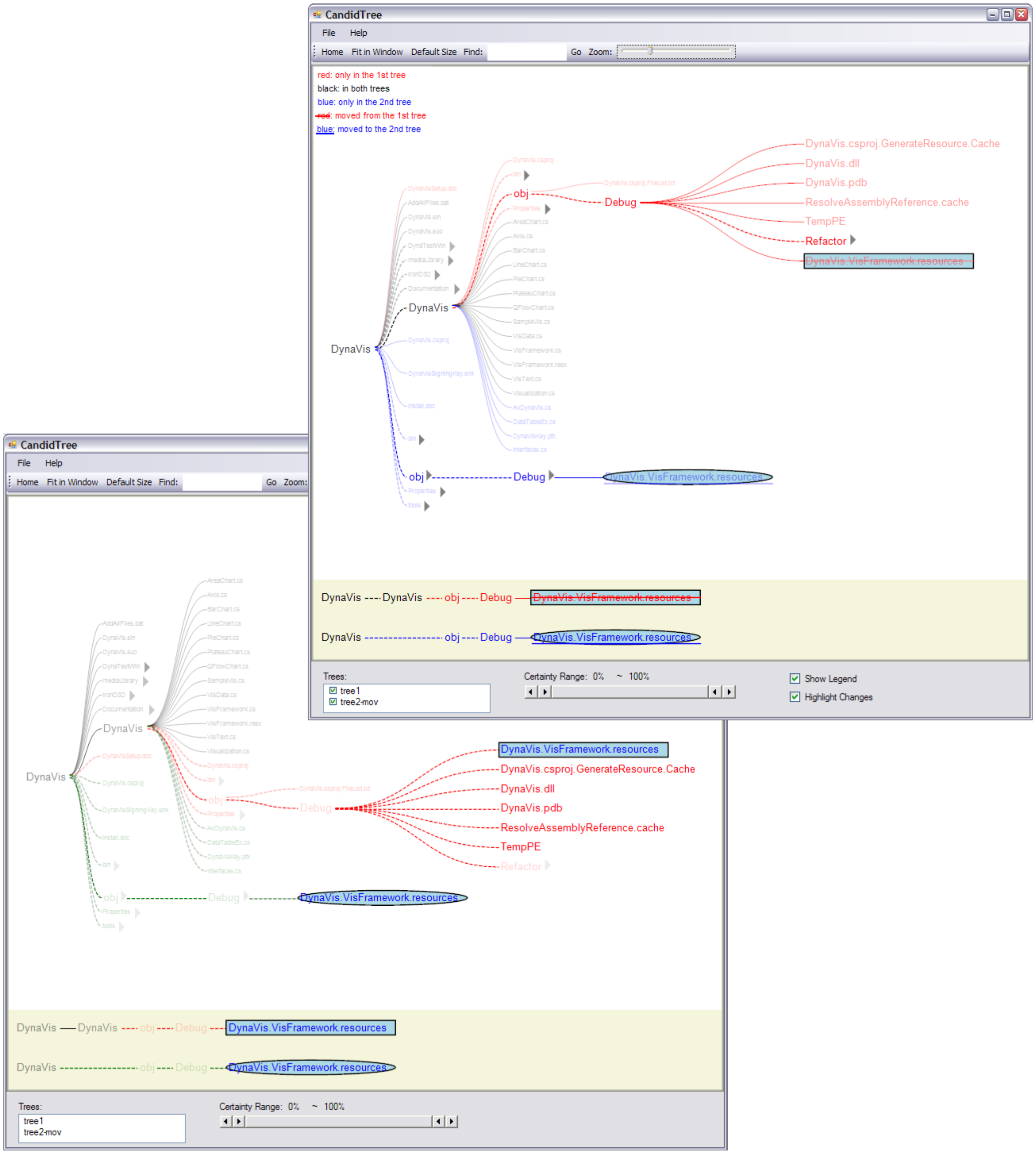


Figure 4 A search for “Anatidae” shows four search results (containing the keyword) and nodes relevant to them.


CandidTree
File Help
Home Fit in Window Default Size Find: Go Zoom:
DynaVis
DynaVis
obj
Debug
DynaVis.csproj.GenerateResource.Cache
DynaVis.dll
DynaVis.pdb
ResolveAssemblyReference.cache
TempPE
Refactor
DynaVis.VisFramework.resources
obj
Debug
DynaVis.VisFramework.resources
CandidTree
File Help
Home Fit in Window Default Size Find: Go Zoom:
DynaVis
DynaVis
obj
Debug
DynaVis.VisFramework.resources
DynaVis.csproj.GenerateResource.Cache
DynaVis.dll
DynaVis.pdb
ResolveAssemblyReference.cache
TempPE
Refactor
obj
Debug
DynaVis.VisFramework.resources

red: only in the 1st tree
black: in both trees
blue: only in the 2nd tree
~~red~~: moved from the 1st tree
blue: moved to the 2nd tree

DynaVis ---- DynaVis ---- obj ---- Debug —— ~~DynaVis.VisFramework.resources~~

DynaVis ---------------- obj ---- Debug —— DynaVis.VisFramework.resources

Trees:
☑ tree1
☑ tree2-mov

Certainty Range: 0% ~ 100%

☑ Show Legend
☑ Highlight Changes

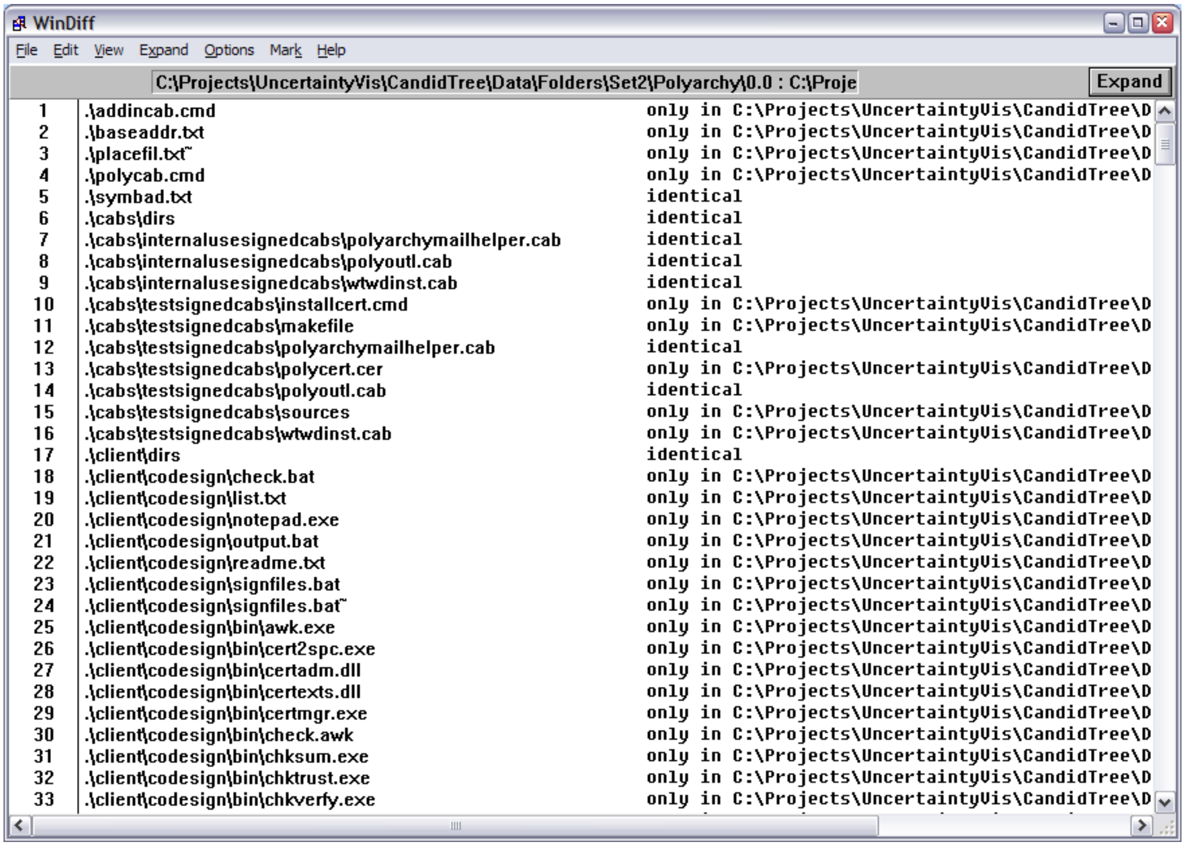
DynaVis —— DynaVis ---- obj ---- Debug ---- DynaVis.VisFramework.resources

DynaVis ---------------- obj ---- Debug ---- DynaVis.VisFramework.resources

Trees:
tree1
tree2-mov

Certainty Range: 0% ~ 100%

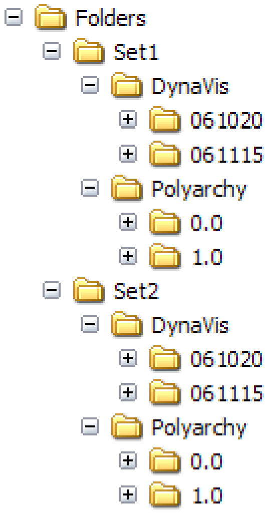
Figure 5 CandidTree improvements – before (bottom left) and after (top right) the usability study.



```
C:\Projects\UncertaintyVis\CandidTree\Data\Folders\Set2\Polyarchy\0.0 : C:\Proje

 1  .\addincab.cmd                                           only in C:\Projects\UncertaintyVis\CandidTree\D
 2  .\baseaddr.txt                                           only in C:\Projects\UncertaintyVis\CandidTree\D
 3  .\placefil.txt~                                          only in C:\Projects\UncertaintyVis\CandidTree\D
 4  .\polycab.cmd                                            only in C:\Projects\UncertaintyVis\CandidTree\D
 5  .\symbad.txt                                             identical
 6  .\cabs\dirs                                              identical
 7  .\cabs\internalusesignedcabs\polyarchymailhelper.cab     identical
 8  .\cabs\internalusesignedcabs\polyoutl.cab                identical
 9  .\cabs\internalusesignedcabs\wtwdinst.cab                identical
10  .\cabs\testsignedcabs\installcert.cmd                    only in C:\Projects\UncertaintyVis\CandidTree\D
11  .\cabs\testsignedcabs\makefile                           only in C:\Projects\UncertaintyVis\CandidTree\D
12  .\cabs\testsignedcabs\polyarchymailhelper.cab            identical
13  .\cabs\testsignedcabs\polycert.cer                       only in C:\Projects\UncertaintyVis\CandidTree\D
14  .\cabs\testsignedcabs\polyoutl.cab                       identical
15  .\cabs\testsignedcabs\sources                            only in C:\Projects\UncertaintyVis\CandidTree\D
16  .\cabs\testsignedcabs\wtwdinst.cab                       only in C:\Projects\UncertaintyVis\CandidTree\D
17  .\client\dirs                                            identical
18  .\client\codesign\check.bat                              only in C:\Projects\UncertaintyVis\CandidTree\D
19  .\client\codesign\list.txt                               only in C:\Projects\UncertaintyVis\CandidTree\D
20  .\client\codesign\notepad.exe                            only in C:\Projects\UncertaintyVis\CandidTree\D
21  .\client\codesign\output.bat                             only in C:\Projects\UncertaintyVis\CandidTree\D
22  .\client\codesign\readme.txt                             only in C:\Projects\UncertaintyVis\CandidTree\D
23  .\client\codesign\signfiles.bat                          only in C:\Projects\UncertaintyVis\CandidTree\D
24  .\client\codesign\signfiles.bat~                         only in C:\Projects\UncertaintyVis\CandidTree\D
25  .\client\codesign\bin\awk.exe                            only in C:\Projects\UncertaintyVis\CandidTree\D
26  .\client\codesign\bin\cert2spc.exe                       only in C:\Projects\UncertaintyVis\CandidTree\D
27  .\client\codesign\bin\certadm.dll                        only in C:\Projects\UncertaintyVis\CandidTree\D
28  .\client\codesign\bin\certexts.dll                       only in C:\Projects\UncertaintyVis\CandidTree\D
29  .\client\codesign\bin\certmgr.exe                        only in C:\Projects\UncertaintyVis\CandidTree\D
30  .\client\codesign\bin\check.awk                          only in C:\Projects\UncertaintyVis\CandidTree\D
31  .\client\codesign\bin\chksum.exe                         only in C:\Projects\UncertaintyVis\CandidTree\D
32  .\client\codesign\bin\chktrust.exe                       only in C:\Projects\UncertaintyVis\CandidTree\D
33  .\client\codesign\bin\chkverfy.exe                       only in C:\Projects\UncertaintyVis\CandidTree\D
```

Figure 6 WinDiff comparing two directory structures.



```
Folders
  Set1
    DynaVis
      061020
      061115
    Polyarchy
      0.0
      1.0
  Set2
    DynaVis
      061020
      061115
    Polyarchy
      0.0
      1.0
```

Figure 7 Eight directory structures used in the user study.

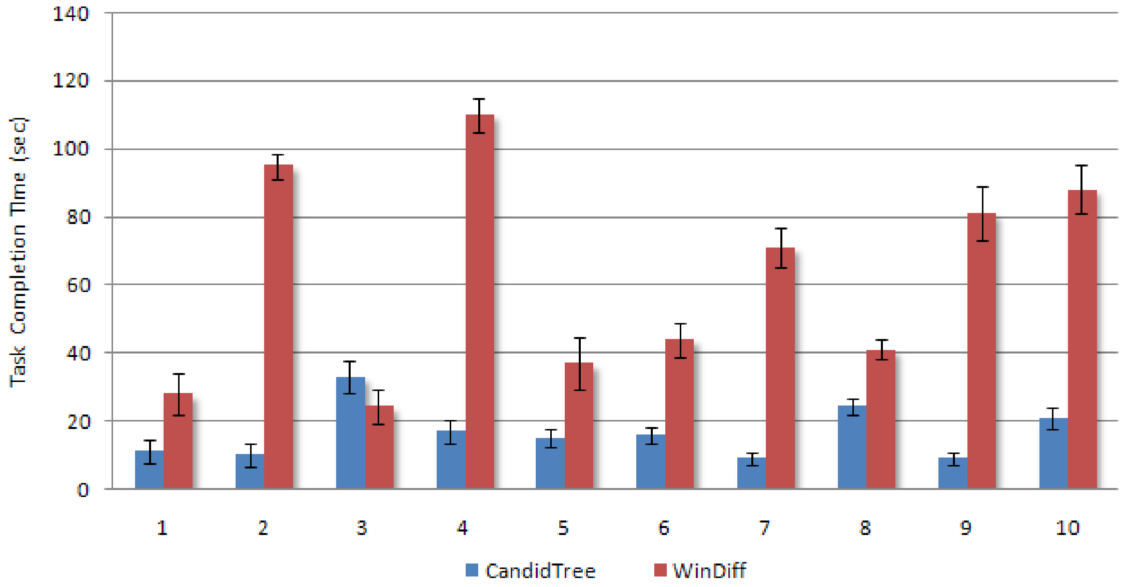
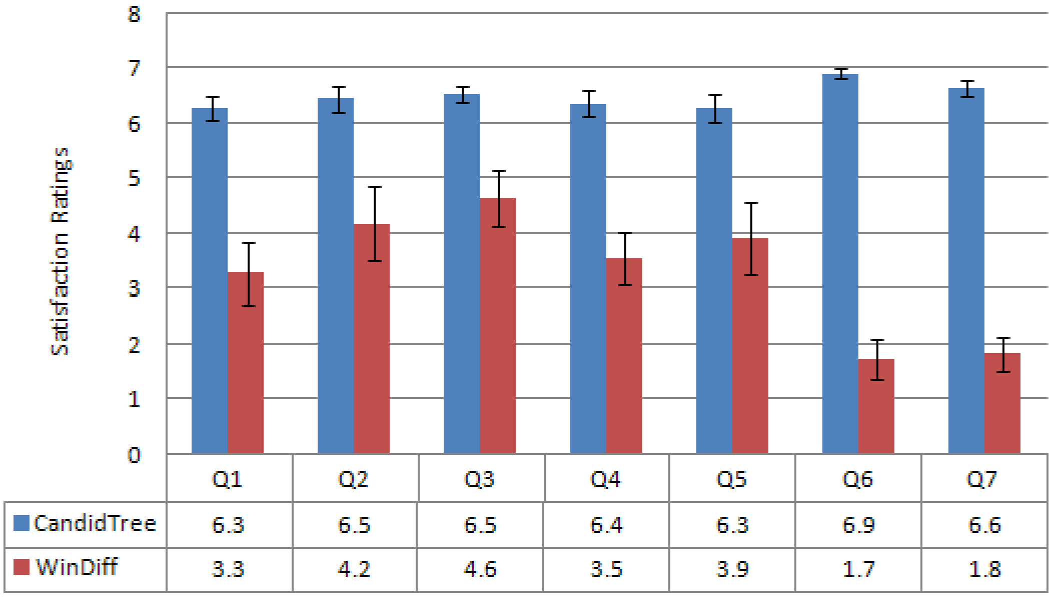


Figure 8 Median task completion times for Task 1 through 10. Error bars indicate Standard Error.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7
🟦 CandidTree	6.3	6.5	6.5	6.4	6.3	6.9	6.6
🟥 WinDiff	3.3	4.2	4.6	3.5	3.9	1.7	1.8

Figure 9 Average Likert scale ratings for Question 1 through 7 using the scale of 1=Disagree, 7=Agree. Error bars indicate Standard Error.

Table 1 Average Likert scale ratings for CandidTree, using the scale of 1=Disagree, 7=Agree. Study 1 refers to the Usability Study with Computer Scientists, Study 2 refers to the Qualitative User Study with Biologists described in the next section, and Study 3 refers to the Controlled Experiment with Computer Scientists described later. *We rephrased two questions for the controlled experiment.

	Study 1	Study 2	Study 3
Q1. Overall, the system was easy to use	5.0	5.7	6.3
Q2. I felt comfortable using this system	5.3	5.7	6.5
Q3. It was easy to learn to use this system	5.2	5.7	6.5
Q4. It was easy to navigate through this system	6.0	5.7	6.4
Q5. It was easy to read the labels of the nodes	4.7	4.6	6.3
Q6. Colors representing the node location change was clear (*It was easy to identify the node location change)	4.2	4.5	6.9
Q7. Transparency representing the sub-tree structure change was clear (*It was easy to identify the sub-tree structure change)	3.5	3.0	6.6

Table 2 Biologist user study tasks, with results from eight participants.

Task	# Missed	Mean duration (sec)
1. Which branch (child) of the Neognathae is the most uncertain, from tree 1 to tree 2?	0	73
2. In the children of Bucerotidae, are the changes additions, deletions, or changes in the placement of taxa?	1	28
3. How much growth (in number of new taxa) has Lampornis experienced from tree 1 to tree 2?	0	19
4. Of the children of Passeriformes, name all the taxa that are placed differently in the different trees.	3	31
5. For Estrildidae, describe the difference in the placement.	0	29
6. The genus Agelaius moved from one group to another. Did all of the other genera in its group move too?	1	51
7. Was the parent of Icterus promoted or demoted in rank?	2	104
8. What happened to the family Cracticidae?	1	57
9. Overall, how would you characterize the uncertainty or change in the tree: changes in deep relationships, mid-tree relationships, or among terminal taxa?	0	137
10. Summarize the changes to Furnariidae and to Sylviidae.	--	275
11. In your opinion, which of the two groups Furnariidae and Sylviidae is the most uncertain?	--	73