

Combining linking & focusing techniques for a multiple hierarchy visualisation

Martin Graham and Jessie Kennedy

*School of Computing, Napier University, 219 Colinton Road, Edinburgh, EH14 1DJ, UK
e-mail: {marting, jessie}@dcs.napier.ac.uk*

Abstract

Visualisations have previously combined linking and brushing operations successfully. However, none to our knowledge have combined linking and focussing techniques. We describe the motivation behind developing such a technique, and show how our visualisation takes advantage of inherent links in a set of multiple overlapping hierarchies to build a mechanism that can produce linked focus + context effects between these multiple hierarchies with a single user action. Previously, our visualisation had utilised linking only, using colour to highlight correlations between distinct but overlapping hierarchies.

1. Introduction

A considerable body of research in Information Visualisation (IV) has addressed various single and multiple focus + context techniques [1]. These techniques resulted from a need to show in detail specific information within a larger structure, but at the same time show this information's relationships with the whole of the structure. Integrated focus + context methods combined these two needs seamlessly in one visualisation.

Likewise, linking techniques [2] were developed in IV to enable users to easily ascertain correlations in a multiple view visualisation. Objects highlighted or manipulated in one view are similarly acted upon where they occur in other views.

Our research has been involved with developing visualisation tools for botanical taxonomy, to allow the comparison of multiple classifications of particular groups of plants [3]. In general terms the data structure is formed from a number of distinct trees, each composed of a differing arrangement of the same set (or sub-set or super-set) of nodes. Within the visualisation the trees are arranged vertically, and each tree is given a title with its

leaf nodes and internal nodes laid out below this descriptor. Groups of leaf nodes are represented as grids of small squares, with internal nodes displayed as indicative labelled brackets above the set of leaf and other internal nodes that it contains. The overall visual effect is one of groups and super-groups of leaf nodes, and therefore we term the display a 'set-based visualisation', as shown in Figure 1.

The visualisation is a compromise between the traditional node-link diagrams found in techniques such

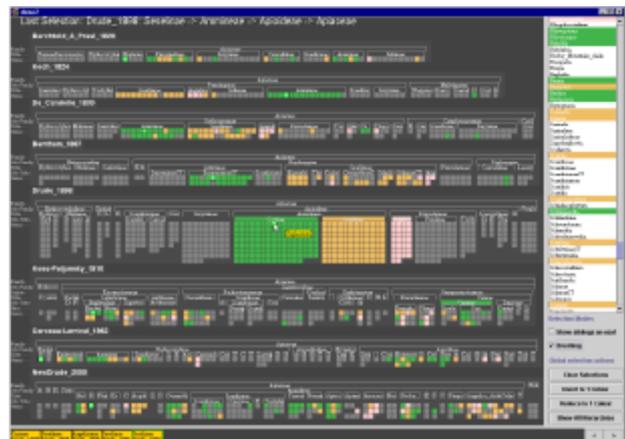


Figure 1. Linking without focus. Selecting a block in one hierarchy colours contained nodes where they occur in the other hierarchies.

as Robertson et al's Cone Trees [4] and the enclosure-based approach of visualisations such as Johnson and Shneiderman's TreeMaps [5]. The visualisation style combines the node-link display's ability to easily determine internal tree structures and the space-efficient layout of enclosure methods. Layouts similar in approach, but different in execution and confined to one overall hierarchy, can be found by Stasko et al [6] and Andrews & Heidegger [7].

The visualisation already utilises linking techniques to show matches between nodes that occur in more than one hierarchy. Selection of a leaf node or sub-tree (group of leaf nodes) in one hierarchy will mark out all representations of these nodes in a distinct colour, and the distribution of these nodes throughout the other hierarchies can then be seen. Interaction is simply a matter of using the mouse to select nodes within the visualisation or from an alphabetically ordered list of node names. Previous selections are stored in a history bar at the bottom of the display and can be re-selected from there.

Our visualisation had reached a limit of about 1,500 nodes (the data set displayed in Figure 1), beyond which individual node representations become too small to manipulate without difficulty. However, as focus + context techniques allow visualisations to give space to more interesting information, we decided to investigate their possible use in our visualisation, aiming to increase the size of the data sets it could comfortably handle. However, as the nodes were linked across several hierarchies, such a technique would need to accommodate multiple focal points linked across these hierarchies. To our knowledge, although visualisations have utilised various combinations of linking and brushing such as is found in the BEAD system [8], previous visualisations have not utilised combinations of linking and focusing.

It can be argued that a general graph visualisation technique such as Osawa's [9] would combine our multiple hierarchies into a single visualisation artefact, so any linking between spatially distinct visualisations for a focus + context technique is unnecessary. However, graph visualisations tend to be harder to comprehend than tree visualisations, and this was confirmed by our user testing [10], as well as previously by Mukherjea et al [11].

Our structure is also similar to the MultiTree structure proposed by Furnas and Zacks [12], but differs in allowing multiple diverging paths between nodes in the overall structure. Our structure is also formed from what can be, individually, very different trees in structural terms, whereas MultiTrees were originally designed to re-use sub-sections of a prior reference tree. These differences effectively rule out the integrated tree-based visualisation that MultiTrees uses. Therefore we decided our visualisation should display each distinct hierarchy

separately.

Sometimes the idea of linking areas of focus does not make sense, as is the case with linked scatterplots [8; 13], where points or nodes are mapped and defined by a series of 2D co-ordinate spaces. Expanding areas around points in one scatterplot would require a scaling function along the co-ordinates. Linking such areas of focus to other scatterplots by data points could create conflicts. If for example two nodes in a scatterplot, one a focal point and the other not, shared the same position in another scatterplot, should the scale contract or expand at such a point? Solutions could be found, but it makes interpretation difficult. Highlighting or filtering is a more practical method for these structures. Trees and hierarchies however, can be visualised in relative layouts, where a node is defined by its relative spatial relationship to other nodes, rather than to an absolute external co-ordinate system. As long as a focus + context method respects and preserves these relative relationships, linking focal effects should not cause a problem in interpretation.

Given that we have a visualisation with linked, multiple hierarchies, we can attempt to extend the focus + context metaphor through linking areas of interest between the hierarchies, in order to focus on more relevant information within larger data sets than we have hitherto been able to accommodate.

2. DOI and display calculation for a single hierarchy

Focus + context techniques consist primarily of either spatial transformations on continuous co-ordinate spaces, such as those described by Keahey and Robertson [14], or relative layouts such as Bartram et al's Continuous Zoom [15], where an object's absolute co-ordinates depend on their neighbours positioning, rather than any global transformation function. Specific examples involving hierarchical structures are, for the former, Hyperbolic Trees [16], and for the latter, TreeMaps [5]. These relative layout techniques depend on functions called Degrees Of Interest (DOI) associated with objects in the visualised structure e.g. for TreeMaps the DOI is traditionally the relative size of files within a visualised directory structure.

Furnas [17] described a DOI as a function which

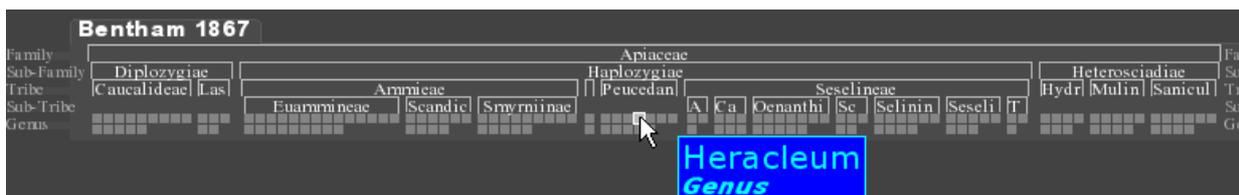


Figure 2. Initial display of a hierarchy. All nodes are of a uniform size.

“assigns to each point in the structure a number telling how interested the user is in seeing that point, given the current task”. Furnas’s approach was then to compare these values against a threshold value, which would decide whether an object was shown or not, resulting in an all-or-none decision. However, as DOI functions can produce a range of values, they can be used to produce a range of outcomes. In the case of a DOI-based focus + context technique, the outcomes will be a range of sizes for the visualised objects. In effect there are two steps; the first is deciding how to calculate DOI values, and the second being how to use those DOIs to calculate object sizes within the display.

Our DOI function is calculated as follows when applied to just a single hierarchy. All leaf nodes start with an average value as their DOI, indicating that the layout is initialised with no focal areas, as shown in Figure 2. Whenever a node representation is selected within a tree representation (or from an ordered list positioned to the side of the trees), the node and all its descendent nodes, if any, are given a maximum DOI value. All other nodes not affected by this selection have their DOI values reduced slightly to or above a non-negative minimum DOI value.

The minimum, average and maximum DOI values are assigned as non-negative integers e.g. 1 for minimum, 6 for average and 12 for maximum. Hence in this case after five selections without participation, a node’s DOI would drop from the average to the minimum value, and a node chosen in the first selection, would now have a DOI dropping back and approaching the average value. Alteration of these values can have various effects e.g. setting the average and maximum values to be equal would result in a situation where unselected nodes had their DOI values reduced but selected nodes’ DOIs remained at the average value.

The policy of slightly reducing unselected nodes’ DOI values reflects the notion that a new selection would slightly reduce the interest of the user in all other unaffected nodes. Over a number of selections this gives nodes and sub-trees of little activity smaller and smaller DOI values in absolute terms as well as relative terms, and provides a mechanism for bringing DOI values back down from the maximum threshold. It also differentiates our DOI function from Furnas’ original description [17], as we don’t have his API (a priori importance) measure but rather an on-the-fly historical importance factor

incorporated into our DOI.

At this point, visualising nodes directly by a linear extrapolation of their individual DOI values would give a display of larger, selected nodes distributed throughout the tree. However, one of the aims of our visualisation is to encourage browsing of the structure through the selection of previously unselected but related nodes. This intention was based on previous experience with our visualisation, which had shown that users tended to make an initial selection from a list of ordered nodes, and would thereafter browse selected or related nodes using the tree representation. Therefore all nodes in each distinct group of leaves within a tree (*leaf group*) are drawn as the same size, indicating that the unselected but related nodes are also of interest. To accomplish this, we decided that the size of node representations should be dependent on the average DOI value for the nodes in a leaf group.

One problem with this approach is that selection of one node, and therefore increasing the DOI of one node, in a large leaf group will not lead to a discernible overall increase in the average DOI value of that group. The average DOI value of a leaf group is controlled by the ratio of selected to unselected nodes in that group, regardless of whether 1 or 100 nodes were selected in total. To give a fairer average value, each unselected node in a leaf group has their DOI value altered by the following function:

$$\text{NewDOI} = (\text{MaxDOI} - \text{OldDOI}) * (\frac{\text{nodes in last selection}}{\text{leaf group nodes}}) / (\frac{\text{nodes in last selection}}{\text{nodes in last selection}})$$

Simply put, a node has its DOI value increased to a value somewhere between its current DOI and the maximum DOI. The proportional increase in value is directly related to the percentage of nodes picked by the last selection action that are siblings of the node (in the same leaf group). So, if only one node was selected in the last action; and was not a sibling of another node in this tree; that other node would not have its DOI value for that tree increased at all, as the proportion of selected nodes in its leaf group was 0%. However, if the two nodes were siblings, then the DOI value for the unselected sibling would increase to maximum, as all (one) of the selected nodes were siblings in that tree. All other nodes in the same leaf group would similarly have their DOI values for that tree increased to maximum.



Figure 3. Selection of a single node increases the DOI and size of all nodes in that group.



Figure 4. Multiple selections. The first of these selections, towards the left of the figure, are beginning to shrink in size.

This method ensures that DOI values are a function of proportional measure, rather than absolutes. Having 100 selected nodes rather than just 1 selected node will not necessarily result in a higher average DOI value for a leaf group. Rather, it is the proportion of selected nodes from the whole selected set that affect the group's average DOI. Figure 3 shows the result of a single selection whilst Figure 4 shows the results after five selections. Figure 4 also demonstrates the reduction in DOI values of previous selections. The figure shows a number of selections that have been made going from left to right in order of selection. The selected groups towards the left-hand side of the hierarchy have slightly smaller node representations than the selected groups towards the right-hand side, emphasising the fact that they are not the most recent selections. The unselected groups have shrunk almost to vanishing point.

2.1. Display Calculation

After each selection, DOI values are recalculated for all leaf nodes. Following this, the distribution of free screen space is similar to Bartram et al's Continuous Zoom [15] technique. Free screen space in this context is understood to be the remaining space after tree details and internal nodes have been displayed. The horizontal space for the tree is allocated proportionally to leaf groups, the metric being the group's share of the tree's total DOI value.

This gives each leaf group an area decided by the amount of free vertical screen space with horizontal space decided by the groups' own relative DOI total within the tree. The individual leaves in a group are arranged in a grid pattern throughout the group's allocated space. As previously stated, the leaves in a group are drawn as the same size to aid browsing behaviour, irrespective of whether they have been directly selected or not. Individual DOI ratings are calculated, so leaf nodes could be drawn in proportion to these if needed. However, this would reduce the size of unselected sibling nodes, making mouse selection of such nodes, and hence browsing of the structure, more difficult; effectively reducing the main part of the visualisation interface to an output channel only.

Our approach also gives the leaf groups a crisper, more cohesive appearance than would be the case by assigning

individual sizes proportional to the varying DOI values of the leaves in a tree. Furthermore, it simplifies calculation of a leaf group's internal layout and helps emphasise the set-based nature of the visualisation.

One problem we encountered with the layout method was that leaf groups of only one or two nodes in a large hierarchy would sometimes be squeezed to less than one pixel in width after the space allocation stage. This occurs because the algorithm distributes area to the groups without consideration of proportion. It is hard to draw meaningful representations in a strip 1 pixel wide by 60 pixels deep. We remedy this by calculating a vertical 'efficiency' for each group, namely the amount of vertical space they would use out of their allocation if square node representations were drawn in their allotted area. Groups with a low 'vertical efficiency' are given 'extra' horizontal pixels to work with, in which larger width, and therefore heights, of square leaf node representations can be drawn.

These extra pixels are available through a pixel 'salami fraud' in the space allocation algorithm. Groups are given a real number of pixels in width, which in practicality is rounded down to an integer for display purposes. The missing fractions add up and form a cache of breathing space that can then be used to give small-sized groups a better chance of displaying themselves.

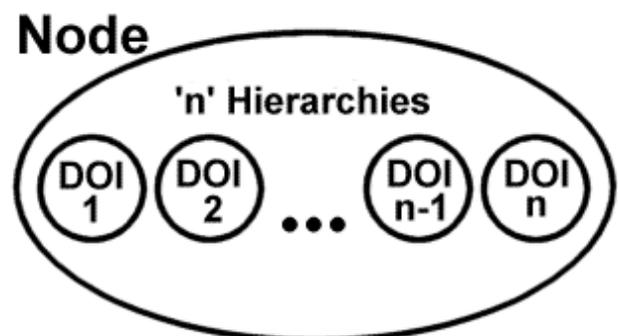


Figure 5. A node has multiple DOI values, one per hierarchy.

3. DOI Calculation Across Multiple Hierarchies.

Given that in one tree our selection choice is simply between selecting one node in a group, or the entire group, we must extend the DOI calculation algorithm to multiple trees for its full use to be understood.

As in our underlying data structure, each node can appear within multiple hierarchies. Therefore each node is assigned multiple semi-independent DOI values, one per representation as shown in Figure 5, which quantify the current level of interest they hold for the user given the history of previous selections and their context within each hierarchy. The problem now is how to link these separate DOI values for the same node.

If a node is selected, the answer would seemingly be to simply give each of its DOI values the same rating, namely the maximum DOI value we gave to nodes when

selecting within a single hierarchy. As previously stated, visualising nodes directly by their individual DOI values would give a display of larger, selected nodes distributed throughout the multiple trees. It would also seem to indicate that one global DOI for each node would suffice, calling into question the need for multiple DOIs for one node.

This cannot occur though, because the previous DOI function alters an unselected node's DOI according to the proportion of selected nodes that are siblings. As nodes have different siblings in different trees, the proportions, and hence derived DOI values, will be different for the same unselected node in different trees. For example, consider unselected nodes in a leaf group in a tree that had 5 out of 25 selected nodes as siblings. They would see their DOI values for that tree increase by $(5/25 =) 20\%$ of the difference between their current and maximum values. In another tree, some of these same unselected nodes may have 10 out of the 25 selected

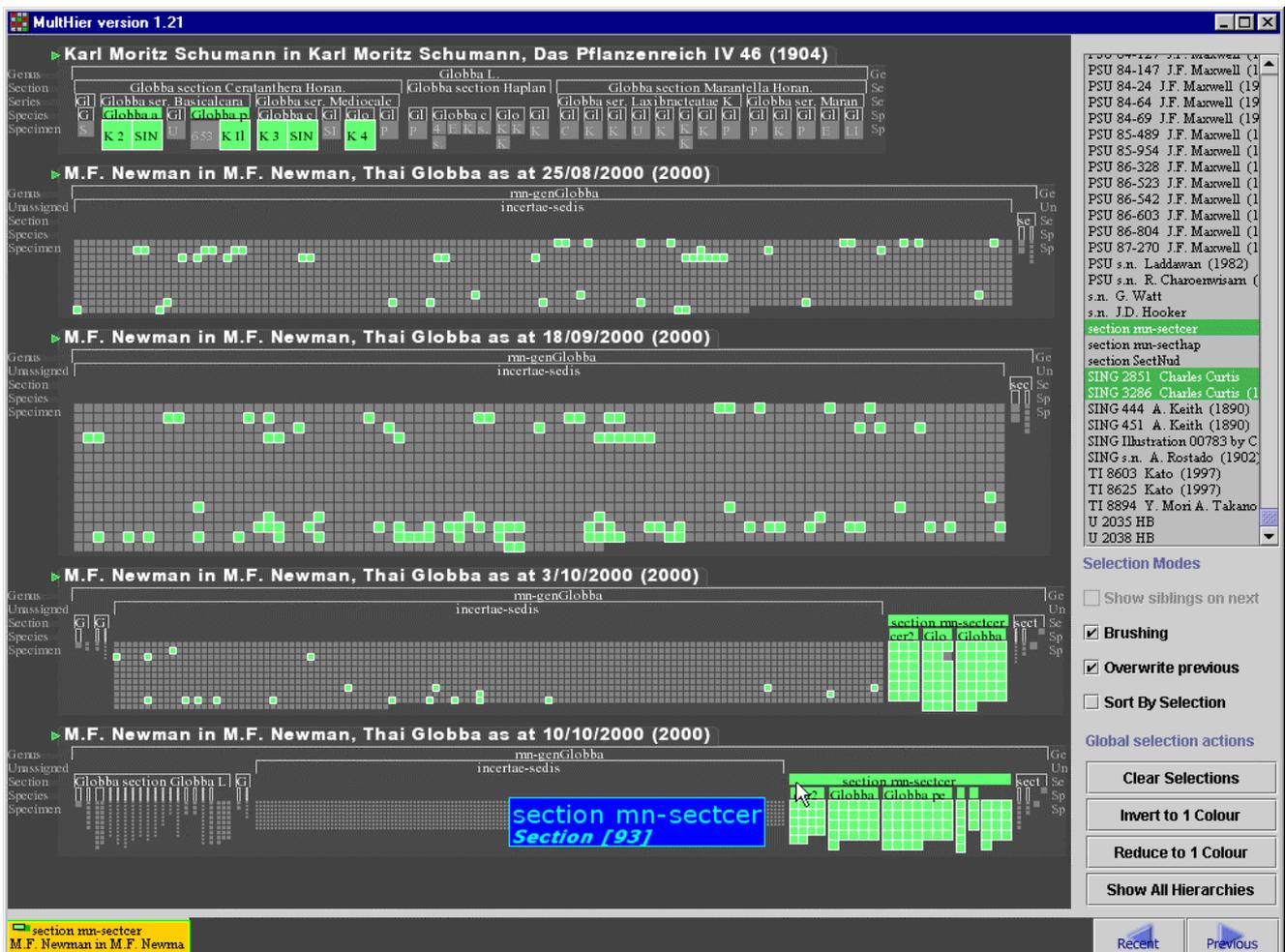


Figure 6. Selection of a group of nodes. The proportion of these nodes contained in other groups is the main factor in deciding their DOIs and sizes.

nodes as siblings, leading to a (10/25 =) 40% increase for the nodes DOI value in that tree and so on. This leads to differing DOI values for the same node across different trees, hence the need for one node to accommodate multiple DOI values, and their description as 'semi-independent'.

Selections can now be made in sequence to compare and contrast different leaf groups distributions. Each selection will produce it's own focal effects across the hierarchies, and these will have a cumulative effect on the nodes' DOI values.

3.1. Display Calculation for Multiple Trees

Displaying the multiple hierarchies is essentially the same process as for single trees, except of course the screen space needs to be vertically sub-divided first amongst the trees. Screen space is allocated in proportion vertically according to the relative total sums of leaf node DOIs for each tree. Trees can be marked as hidden from view, and subsequently such trees are given no space allocation. If the multiple trees' other display elements overwhelm the display area by themselves, then leaf node sizes are set to a minimum, and the display can be scrolled vertically to view all trees.

Multiple trees may suffer from the space allocation problem described earlier that occurred between groups in the same tree. Smaller trees may have their space allocation squeezed by a number of larger trees to a point where meaningful node representations cannot be drawn, even if the tree has a high average DOI. Therefore each tree is given a minimum allocation of vertical display space to work in, and accounts for the perception that very small trees have an unfair allocation of space, as seen in Figure 6. Such a situation is preferable to one where the tree is hardly visible at all.

4. Example

An example interaction with the visualisation is described using the multiple hierarchy data set as shown in Figure 6. Each hierarchy displays a stage in the construction and refinement of a classification of plant specimens, termed a taxonomy.

To begin with, the display of approximately 5,500 node representations (leaf and category) draws all leaf nodes at the same scale. In such an interface, picking out with the mouse, never mind locating, one particular node becomes difficult. However, our prior testing showed that users usually have particular target nodes in mind when browsing these taxonomic structures, analogous to the manner in which users tend to have a page they want to head to when they start up a web browser. The ordered list on the right-hand side enables the user to quickly locate their node of interest and select it there. (It could of

course be argued that a lot of browser users head for a search engine site, but the ordered list serves the same function for our hierarchies).

Selection of a single leaf node from the list increases the DOI of the groups in which the leaf node is present. As some groups will now have relatively larger DOI totals, their contained leaf nodes are drawn proportionally larger and as such are able to be readily picked out by the mouse pointer, thus enabling a browsing behaviour if needed in the visualisation from here on.

Figure 6 displays the selection of a sub-tree in the bottom hierarchy. Here we can see that the proportion of selected nodes inside a leaf group affects the size of the leaf nodes in that group. The large node group in the third hierarchy down has large leaf node representations as it contains almost all of the nodes originally selected. The large groups in the second and fourth hierarchies have smaller leaf node representations, as they only contain a proportion of the selected node set, and the large group in the fifth hierarchy has no selected nodes and so is drawn with extremely small leaf nodes.

5. Discussion

Before the implementation of the multiple linked focus + context method, our visualisation had reached a practical limit of about 1500 nodes on-screen. Selected nodes were marked in colour, as they are now, but all nodes were drawn at a uniform scale. However, introduction of the linked focus + context method has so far enabled data sets with a total of up to 8000 nodes to be displayed in full.

One limitation inherent to the approach occurs when a selected leaf group in one tree is almost equally distributed throughout the leaf groups of another tree. Our method will produce almost identical average DOI values for each leaf group in this tree, so no group will get a proportionally larger share of space at the expense of another. The display will then show the tree peppered with coloured node representations, but all the leaf groups will be drawn with representations of roughly the same size, each acting as context for the others as much as it is a focal point itself, cancelling the effect of both. We term this effect 'focal spread'. If nothing else, such an occurrence could emphasise an almost deliberate orthogonality between the manner in which two trees classify or store the same nodes. Such an occurrence is shown in Figure 7, using another example data set. Here, the selected groups are almost uniformly distributed in the groups of the bottom hierarchy, under 'New Drude 2000'. This leads to no groups standing out in particular.

Another problem occurs with unbalanced tree structures such as those in Figure 6. Large groups of leaves prove difficult to display in relation to small groups and the opportunity to increase the size of such

groups is limited. However, such structures present difficulties for all tree visualisations, and we believe our

difference is that the zero minimum DOI choice removes context at a threshold, suppressing its output altogether,

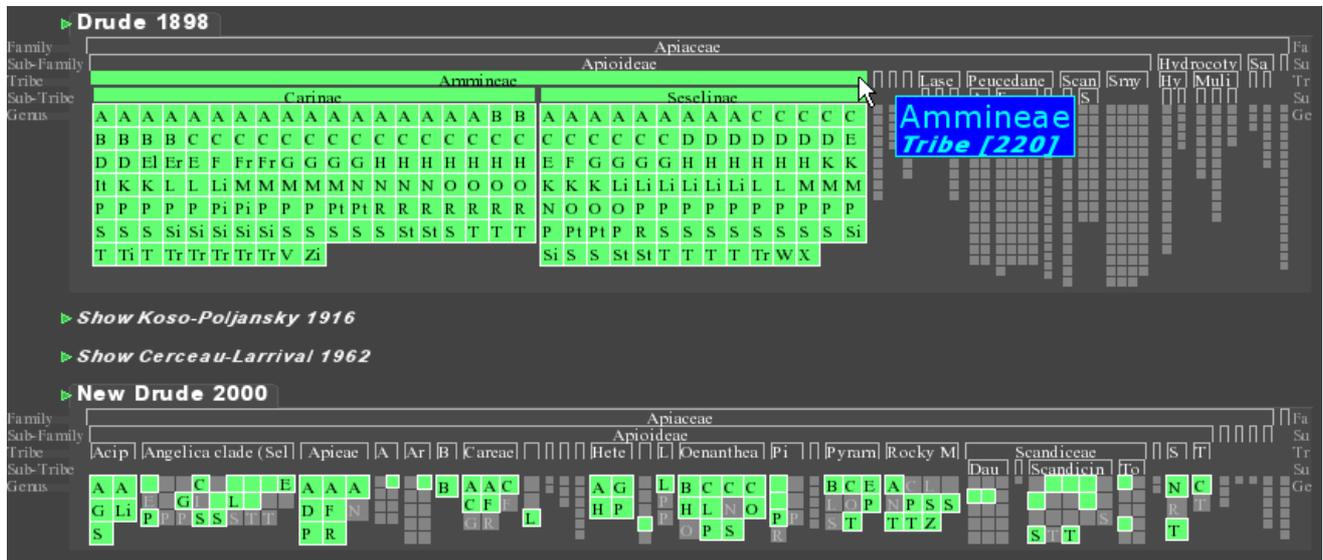


Figure 7. Focal spread in the bottom hierarchy. Many groups' DOIs are affected almost equally by the selection, and consequently no distinguishable focal areas appear in that hierarchy.

method to handle them as well as any, especially as we are dealing with multiple examples of them.

An interesting point occurs when the visualisation's display layout recalculates itself after a selection. Comparing Figures 2 and 3 as examples, it can be seen that there is a noticeable change in layout when nodes are selected. Misue et al [18] state that when layouts are altered it helps if the relative positioning ("above", "to the right of" etc) of objects remains the same. This eases the process of reconciling the new layout with the old layout. Our layout algorithm does that between hierarchies and between leaf groups in individual hierarchies, but not internally within individual leaf groups. Keeping the same shape for each leaf group would be inefficient in terms of screen space, and rather than use animation we use the colour of the selected nodes to act as perceptual anchors within the groups. For example, the highlighted node in Figure 2 can be easily correlated with the highlighted node in Figure 3, helping a user map between the changes in layout for groups of interest.

A further point is that if space is at a real premium, there is potential for the DOI mechanism to perform elision on individual leaf groups. If the minimum DOI value is allowed to be zero, after a number of selections a leaf group's total DOI would drop to zero if none of its constituent nodes had been selected, as each DOI is reduced slightly if its associated node is not involved in a selection action. The space allocation algorithm would therefore reserve no space for this group. The essential

similar to Furnas's [17] original display policy. A minimum DOI value above zero would mean leaf groups were always visible as context, and this is our current policy.

6. Conclusions

Our work combines a focus + context technique with a linking metaphor across an example data set of multiple hierarchies. An initial node or sub-tree is selected in one hierarchy and the resulting focal effects for these nodes are also calculated for the other hierarchies in which they occur. Individual node DOI values are used to produce an average size for each group of leaf nodes to encourage a browsing behaviour with the visualisation.

The technique enabled us to interact effectively with larger data sets than was previously possible in the same display area. There are limitations to the technique, such as focal spread, as multiple focus + context or multiple view visualisations are inherently more difficult to design than their singular counterparts.

The general idea of linking focus + context techniques can be applied to data which is represented through multiple views, which can be either multiple representations of the same structure, or multiple visualisations of data represented through different structures, though the latter requires relative layouts rather than absolute co-ordinate layouts.

7. Acknowledgements

We would like to thank the taxonomists at the Royal Botanical Garden Edinburgh for supplying the data sets used in the example. Martin Graham's PhD is funded by EPSRC.

8. References

- [1] Y.K. Leung and M.D. Apperley, "A review and taxonomy of distortion-oriented presentation techniques," *ACM Transactions on Human-Computer Interaction*, vol. 1, no. 2, 1994, pp. 126-160.
- [2] M.Q.W. Baldonado, A. Woodruff and A. Kuchinsky, "Guidelines for Using Multiple Views in Information Visualizations," *Proc. AVI 2000*, ACM Press, Palermo, Italy, May 24-26, 2000, pp. 110-119.
- [3] M. Graham, J.B. Kennedy and D. Benyon, "Towards a methodology for developing visualisations," *International Journal of Human-Computer Studies*, vol. 53, no. 5, 2000, pp. 789-807.
- [4] G.G. Robertson, J.D. Mackinlay and S.K. Card, "Cone Trees: Animated 3D Visualizations of Hierarchical Information," *Proc. CHI '91: Human Factors in Computing Systems*, ACM Press, New Orleans, Louisiana, USA, April 27 - May 2, 1991, pp. 189-194.
- [5] B. Johnson and B. Shneiderman, "Treemaps: A Space-Filling approach to the visualization of hierarchical information structures," *Proc. IEEE Visualization '91*, IEEE Computer Society Press, San Diego, California, USA, Oct 22-25, 1991, pp. 284-291.
- [6] J. Stasko, M. Guzdial and K. McDonald, "Evaluating Space-Filling Visualizations for Hierarchical Structures," *Proc. IEEE InfoVis '99, Late Breaking Hot Topics*, IEEE Computer Society Press, San Francisco, California, USA, October 25-26, 1999, pp. 35-38.
- [7] K. Andrews and H. Heidegger, "Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs," *Proc. IEEE InfoVis '98, Late Breaking Hot Topics*, IEEE Computer Society Press, Research Triangle, North Carolina, USA, October 19-20, 1998, pp. 9-12.
- [8] D. Brodbeck et al., "Domesticating Bead: Adapting an Information Visualization System to a Financial Institution," *Proc. IEEE InfoVis '97*, IEEE Computer Society Press, Phoenix, Arizona, USA, October 20-21, 1997, pp. 73-80.
- [9] N. Osawa, "A Multiple-Focus Visualization Technique Using Heat Models and Force-Directed Layout," *Proc. IEEE InfoVis '99 Late Breaking Hot Topics*, IEEE Computer Society Press, San Francisco, California, USA, 25-26 October, 1999, pp. 27-30.
- [10] M. Graham, J.B. Kennedy and C. Hand, "A Comparison of Set-Based and Graph-Based Visualisations of Overlapping Classification Hierarchies," *Proc. AVI 2000*, ACM Press, Palermo, Italy, May 23-26, 2000, pp. 41-50.
- [11] S. Mukherjea, J.D. Foley and S. Hudson, "Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views," *Proc. ACM CHI '95*, ACM Press, Denver, Colorado, USA, May 7-11, 1995, pp. 331-337.
- [12] G.W. Furnas and J. Zacks, "Multitrees: Enriching and Reusing Hierarchical Structure," *Proc. ACM CHI '94*, ACM Press, Boston, Massachusetts, USA, April 24-28, 1994, pp. 330-336.
- [13] R.A. Becker and W.S. Cleveland, "Brushing Scatterplots," *Technometrics*, vol. 29, no. 2, 1987, pp. 127-142.
- [14] T.A. Keahey and E.L. Robertson, "Nonlinear Magnification Fields," *Proc. IEEE InfoVis '97*, IEEE Computer Society Press, Phoenix, Arizona, USA, October 20-21, 1997, pp. 38-45.
- [15] L. Bartram et al., "The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces," *Proc. ACM UIST '95*, ACM Press, Pittsburgh, USA, November 14-17, 1995, pp. 207-215.
- [16] J. Lamping and R. Rao, "Visualizing Large Trees Using the Hyperbolic Browser," *Proc. ACM CHI '96*, ACM Press, Vancouver, Canada, April 13-18, 1996, pp. 388-389.
- [17] G.W. Furnas, "Generalized Fisheye Views," *Proc. ACM CHI '86*, ACM Press, Boston, Massachusetts, USA, April 13-17, 1986, pp. 16-23.
- [18] K. Misue et al., "Layout Adjustment and the Mental Map," *Journal of Visual Languages and Computing*, vol. 6, no. 2, 1995, pp. 183-210.