Network Visualization by Semantic Substrates

Ben Shneiderman, *Senior Member, IEEE*, and Aleks Aris

Abstract—Networks have remained a challenge for information visualization designers because of the complex issues of node and link layout coupled with the rich set of tasks that users present. This paper offers a strategy based on two principles: (1) layouts are based on *user-defined semantic substrates*, which are non-overlapping regions in which node placement is based on node attributes, (2) users interactively adjust sliders to control link visibility to limit clutter and thus ensure comprehensibility of source and destination. Scalability is further facilitated by user control of which nodes are visible. We illustrate our semantic substrates approach as implemented in NVSS 1.0 with legal precedent data for up to 1122 court cases in three regions with 7645 legal citations.

Index Terms— Network visualization, semantic substrate, information visualization, graphical user interfaces.

1 Introduction

Existing network visualizations often seem impressive because of the colorful display of nodes richly connected with links. These visually engaging images enable users to estimate the network size while revealing important clusters. However, in most examples, the overlapped nodes prevent users from estimating cluster size and the crossed links make it impossible to follow connections, count node in-degree, or carry out other tasks.

Successful visualizations enable users to gain meaningful high-level information from an overview as well as to ascertain the details of each node and link. This paper begins by considering users' tasks with basic networks containing simple nodes and links, and then builds up to more complex challenges, such as networks with node labels, directed links, node attributes, and link attributes. We use the terminology of networks, nodes and links, but these are often called graphs, vertices, and edges.

Our contribution is a system in which (1) layouts are based on *user-defined semantic substrates*, which are non-overlapping regions in which node placement is based on node attributes, (2) users interactively control link visibility to limit clutter and thus ensure comprehensibility of source and destination.

User-defined semantic substrates allow automatic node placement by their attributes, so that the location conveys information. For example, if a node, representing a scientific article, is in a region labeled 'Journals,' users immediately learn more. If the node is to the left in the region, the node represents an earlier paper, while nodes on the right are later papers.

After studying hundreds of existing network visualizations, we believed that the most successful ones showed small networks with 10-50 nodes and 20-100 links. *In these effective examples users can count the number of nodes and links, and follow each link from source to destination.* In these examples, links rarely cross and links are drawn to avoid the confusion of tunneling under nodes. When links cross, they do so at close to a 90-degree angle to facilitate visual tracking. Another requirement for effectiveness is that users can determine the degree of every node, that is, the number of links to other nodes. These modest, yet comprehensible, visualizations are the starting point for this discussion, as our goal is to increase the number of nodes and links that are visible while still preserving the comprehensibility that supports effective task completion. Users interactively control the visualizations to display only a small number of nodes and links, possibly from a very large network. Since many networks have millions of nodes and links, users must be able to rapidly select and display the key nodes and links by dynamic query sliders that act as filters.

- *Ben Shneiderman is a Professor with the Computer Science Department and the Human-Computer Interaction Laboratory at the University of Maryland, College Park, E-Mail: ben@cs.umd.edu.*
- *Aleks Aris is a PhD Candidate with the Computer Science Department and the Human-Computer Interaction Laboratory at the University of Maryland, College Park, E-Mail: aris@cs.umd.edu.*

Manuscript received 31 March 2006; accepted 1 August 2006; posted online 6 November 2006.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2 Previous Work

There is a huge literature on network visualization [11] and entire conferences devoted to the topic, such as the 13-year old International Symposium on Graph Drawing (http://www.gd2005.org/). Zooming [2] and fisheye (or other distortions) approaches have been used to give users some control, but effective layouts are still needed to minimize link crossings and tunneling under nodes. In addition, dynamic query filters may still be needed to reduce node and link density. NicheWorks included helpful interactive features, such as highlighting nodes, links, and hiding them, for analysis purposes of graphs ranging from 10,000 to 4,000,000 nodes. Using a subset of a telephone network call graph, Wills illustrates how an analyst could narrow the search to find patterns suggesting fraud [38].

The literature on network layout has been dominated by force-directed strategies because they produce elegant spreading of nodes and reasonable visibility of links. Nodes are laid out as if there were electrical forces between them, where links determine the attraction between connected nodes. Eades [12] proposed the idea but the most common reference is to the refined Fruchterman-Reingold (FR) algorithm [14], with further refinements by many others [15]. Variations are sometimes called spring-embedding to describe the connections between connected pair of nodes ([23], [24]) or simulated annealing, which alludes to the process of heating and cooling metals ([9], [20]). Multi-scale algorithms ([18], [19]) are scalable versions of force-directed methods that work on a coarse representation of a large network, which refine the layout locally to achieve remarkably rapid layout for large networks (10^6 nodes in a few seconds).

A second common layout strategy, which generates familiar and comprehensible layouts, uses geographical maps, in which the node locations are fixed, as in cities on a world map ([1]).

A third common strategy uses a circular layout for nodes that produces an elegant presentation with crisscrossing lines through the center of the circle ([22], [7]). Multiple concentric circles are sometimes used. A further variation is the radial or egocentric layout, which places an individual at the center of a social network with closeness along radial lines to other nodes indicating strength of relationship.

A different strategy is to use matrix-based representations instead of node-link diagrams ([17], [1]). Such representations avoid some of the problems of node-link diagrams (especially with large graphs), such as node occlusion, edge crossings, and edges tunneling under nodes by having fixed places for nodes and links on the screen. On the other hand, spatial characteristics may become harder to perceive, such as finding nodes on a path and identifying clusters. Network exploration by tabular lists of nodes and links can facilitate many tasks, especially when reading of textual labels and attributes is helpful [28].

Meaningful groups of nodes can be formed by hand [32] or algorithmically [21] based on linking strength. This spatial approach is easily understood by users and is appealing since it may reveal surprising groupings. Nested or hierarchical clusters enable users to navigate large graphs, focus on regions of interest, and choose the level of detail by zooming. Schaffer et al. [34] report that the use of fisheye enhances the productivity of users in such systems compared to local zoom without an overview. An alternative approach to zooming is to show all levels of the hierarchy at the same time, each level on a 2D plane in 3D space [13]. While such an approach promises an increase in comprehension, problems of occlusion and finding the best view-angle, common in 3D visualizations, may pose challenges. These and other clustering approaches ([3], [5]) are related to semantic substrates, but, by contrast, we seek to form groups based on node attributes. Algorithmic layout approaches for nodes based on multi-dimensional scaling, self-organizing maps and Sammon maps have some value, but these methods do not have the clarity that user-defined regions have.

Meaningful layouts by node attributes is an underlying principle of temporal placement strategies, called historiographs [16]. These typically show older nodes on the top and recent nodes below, with layers in between holding nodes in the same year. When used for citation networks, references from recent articles on the bottom point upwards to older articles. Bottom-to-top or left-to-right temporal sequences are also possible [10]. Similar looking layered layouts have long been in use ([6], [36]), but these layers are based only on links. Kosak et al. [27] group nodes according to their type and show two ways of organizing the nodes within each group: rule-based and using genetic algorithms. Other researchers have identified mental maps as useful guides to layout and warn about surprising changes to node placement [30].

The notion of user-defined semantic substrates proved beneficial in a network visualization tool for author name resolution in bibliographic database [4]. Author name nodes were laid out in five distinct regions so users could quickly spot shared and non-shared co-authors for suspected duplicate names. Another inspiration for semantic substrates is the user defined spatial layout for photos with shared attributes [26].

Three recent systems have elements of semantic substrates. Jambalaya [35] integrates SHriMP views into the Protégé framework. A graph metaphor is used to show links between concepts, which may include sub-concepts (subclasses). Users can manually place the nodes or automatically order them by some structural property of nodes, such as number of children, however, not by node attributes. PivotGraph [37] places nodes on a two dimensional grid by their node attributes and nicely aggregates nodes by their attributes to present a useful overview. Pretorius, et al. [33] represents multi-dimensional transitional systems as networks and uses the projection of multi-valued node attributes to the 2D plane to position nodes. The projection is parametrized and user adjustable, which users could experiment with to arrive at a good projection that fits their needs.

3 Network Visualization Tasks

To unravel the complex requirements for network visualization, we propose a collection of challenges:

C1) Basic networks with nodes and links. Nodes are unlabeled points and links are undirected.

C2) Node labels (e.g. article title, book author, or animal name)

C3) Link labels (e.g. strength of connection, type of link (active/inactive, car/train/boat/plane)).

C4) Directed networks (links go from source to destination, such as from citing to cited article in citation networks or from predator to prey in food webs).

C5) Node attributes that allow meaningful grouping (spatial layout), coloring (continuous or discrete), or sizing (continuous or discrete) of nodes:

a) categorical (e.g. journals/conferences/books/websites or mammals/reptiles/birds/fish/insects)
b) ordinal (e.g. winter/spring/summer/fall or small/ medium/large)

numerical (integer or real) (e.g. age or weight)

C6) Link attributes that allow coloring (continuous or discrete) or thickness coding (thin or thick):

categorical (e.g. car/train/boat/plane)

ordinal (e.g. weak/normal/strong)

numerical (integer or real), (e.g. probability, length, time to traverse, strength)

Solving these challenges gets more difficult as the number of nodes and links grow. In the past, network drawing programs have been evaluated by algorithm execution speed and aesthetic criteria such as symmetry, balance, number of link crossings, maximum link lengths, etc. More sophisticated programs have tried to minimize overlapping of nodes, links, and labels. The common assumption has been that all nodes, links, and labels must be drawn for an output on paper or a static screen display. Since modern interfaces increasingly include interaction, the opportunities for improvement have dramatically expanded and designers are now paying more attention to supporting specific user tasks with interactive controls [38].

The unlimited number of tasks users might need to carry out on a network seems to make the design process difficult, but a priority ranking can guide the designer's way forward. A starting list for high priority tasks on basic networks includes:

T1) count number of nodes and links
T2) for every node, count degree
T3) for every node, find the nodes that are distance 1, 2, 3 …away
T4) for every node, find betweenness centrality
T5) for every node, find structural prestige
T6) find diameter of the network
T7) identify strongly connected or compact clusters
T8) for a given pair of nodes, find shortest path between them

When moving up to C2 and C3, where labels are allowed, additional tasks might be:

T9) for every node/link, read the label
T10) find all nodes/links with a given label/attribute

When moving up to C4, where directed links are allowed, additional tasks include variations on T1-10 that are based on directed links. For example, shortest paths would be along links from the start node to the end node.

When moving up to C5 and C6 where attributes are allowed, additional tasks include variations on T1-10 that are based on the categorical attribute values of nodes and links. For example, users might want to count the numbers of nodes of each categorical attribute value. In citation networks, users might want to know how many journals, conferences, books, or web sites are included in the network. In addition, there are new tasks, such as:

T11) find links between nodes with different attribute values (e.g. journal articles that cite conference articles or mammals that eat fish)

T12) find the proportion of links from a node that go to each category for every node (e.g. for a given article, what fraction of the citations go to each category of articles or for a given animal what fraction of its diet comes from eating each category of animal)

T13) for a pair of nodes, find paths with the lowest cost
T14) find links with connection strength greater than 0.5

These basic tasks are just a start, since there are an unlimited number of tasks that could be defined.

4 Interaction

Designers are increasingly aware that drawing a static representation of a network is a useful goal, but interaction is necessary to fully support visual analytic exploration tasks and to cope with larger networks [33]. Instead of viewing all million nodes, users may get what they need by viewing only the nodes with high out-degree, betweenness centrality, etc. Users may also know facts such as node labels, which they can specify to view only nodes having the given node labels, or their neighbors. Users may select nodes for display based on attribute values or ranges, e.g. show only articles written during 2002-2004 or mammals larger than 50 pounds. Users may select links for display based on their attribute values or ranges, e.g. show only co-author links if there are more than 5 jointly written papers.

These queries are increasingly supported by software packages that provide dynamic query sliders so users can make rapid, incremental, and reversible queries. If users are confronted with too many nodes and links, they can filter out less relevant nodes to see a meaningful subset. If moving the slider eliminates all nodes, users could merely move the slider back to see only a few.

For citation networks, there is a modest history of the PathFinder algorithms, which show only major papers that have numerous citations [8], but these strategies would be further improved if user control were provided.

5 Semantic Substrates For Node layout

This paper proposes to use semantic substrates to lay out nodes in non-overlapping screen regions based on node attribute values. Existing node layout strategies are usually based on force-directed, geographic, circular, and temporal strategies. Our goal is to promote more explicit organization by allowing users to specify regions for node placement based on node attributes.

We visited the growing web resource at http://www.visualcomplexity.com, which has more than 300 examples of network layouts. Although this resource may not be the authoritative source on effective network visualization, our hope is that a sample from here might reflect common practice. We examined the first 100 in detail to determine the node layout strategy based on descriptions on the website and related papers. For those in which we were successful (about three quarters of them), more than a third used force-directed algorithms and just under a third used geographic placements. Circular layouts accounted for one sixth with a mix of single and multiple concentric circles. Placement on the circle varied among random, temporal, and geographic (based on longitude). A few layouts used spatial clustering, two were temporal, and only one had hand-made regions.

Within these examples, about 1/8 were basic networks (C1) without labels, the rest showed selected node labels or tried to place all the node labels when there were small networks (C2). Only 1/10 showed directed links (C4) although several more had implied directions, such as in food webs. Only 1/10 showed node attributes color or size coding (C5), and just a few had link attributes as shown by link color or thickness (C6).

These layout strategies are helpful for some of our tasks, but not for others. Most showed cluttered layouts that made it impossible to follow links or count nodes. If designers would consider support for specific tasks, it seems clear that they could improve these layouts or more likely provide a control panel for user interaction that would limit complexity and control visual features, such as color, size, labels, etc.

This paper and our implementation focus on supporting tasks related to categorical node attributes with undirected or directed links. We allow users to place nodes in a semantic substrate of non-overlapping regions. While color-coding of nodes may be helpful to view categorical node attributes, we believe spatial layouts in regions will be more effective. For example, a useful layout for scientific articles might be by four publication venues: journal, conference, book, and web. One advantage of semantic substrates is that proportionally-sized regions would immediately give users some idea of the relative cardinality of each category. For example, in a food-web layout with five regions by mammals, reptiles, birds, fish, and insect groups, users would be able to see that there are many more insects than mammals or reptiles.

A second advantage of semantic substrates is that users can quickly distinguish links that cross from one category (region) to another, for example, enabling users to see that reptiles eat insects and mammals, but insects do not eat reptiles or mammals. Node layout within a region is done by existing methods such as geographic, force-directed, or temporal algorithms, but new opportunities exist such as having nodes be closer to regions to which there are many links. Once node layout is done, user control of link display facilitates exploration. Users can elect to show only links within a region or only across selected regions. For example, it might be interesting to see only citations from journals to journals or only citations from journals to websites.

Of course semantic substrates are effective only if there is some categorical attribute or if a numerical attribute can be binned to form categories. A small number of categories, such as 2-5 is convenient for design. Another caution about semantic substrates is that they complicate node and link drawing by imposing an additional constraint on the layout. However, the added utility of user control of link visibility may prove more advantageous.

6 Implementation of NVSS 1.0

To explore the efficacy of network visualization by semantic substrates, we constructed a Java implementation called NVSS 1.0. This was implemented using the Java Universal Network/ Graph (JUNG) (http://jung.sourceforge.net/) Framework, an open source software library, widely used by network visualization researchers.

Our NVSS 1.0 design strategy is to allow users to specify screen regions with color backgrounds and region labels, in which nodes can be placed. Node placement algorithms within each region can be accomplished by force-directed, geographic, circular, temporal, treemap, or random layouts. Specifically, we considered that one or more attributes could be used to form regions that nodes fall into, and then one or more of the remaining attributes are used to determine node placement within each region. In the next section, one attribute is used to determine regions and one or two of the remaining attributes are used to place nodes within each region.

Users can control link visibility through checkboxes that allow separate control for within each region and across each region. For basic networks with undirected links, the number of checkboxes needed for 2 regions is 3, for 3 regions is 6, for 4 regions is 10, for k regions is $k*(k-1)/2 + k$ which equals $k*(k+1)/2$. For directed networks the number of checkboxes needed for 2 regions is 4, for 3 regions is 9, for 4 regions is 16, for k regions $k*(k-1) + k$ checkboxes which equals k^2. The checkbox approach has its limitations as the numbers of regions grow, but more advanced solutions based on visual models are possible.

Even within an NVSS 1.0 region, the number of links could create visibility problems. Therefore NVSS users can also control the number of nodes whose links are shown within a region. Many strategies are possible for node filtering based on attributes of the node, e.g. in-degree, out-degree, year, label, etc. NVSS 1.0 uses the JUNG strategy of quad curves for links, but better approaches are needed, especially for links that cross regions.

7 Legal Precedent Example

To demonstrate semantic substrates, we present an example of the NVSS 1.0 implementation with the data from our work with legal precedents. The database, collected by a team of researchers from the Department of Government and Politics at the University of Maryland (http://www.bsos.umd.edu/gvpt/CITE-IT/), contains 2780 federal judicial cases from the period 1978 to 2005 concerning the legal issue known as "regulatory takings." The U.S. Constitution requires the government to provide "just compensation" when it physically appropriates private property for a public use (building a highway, for example). A "regulatory taking" requiring the payment of "just compensation" may also occur when the value of private property is destroyed by government action that falls short of actual appropriation, such as when a zoning ordinance has the indirect effect of depriving the owner of any viable use of the property.

In this example, node placement is tied to the temporal attribute (year for the case) in which the oldest cases (nodes) are on the left and the newest on the right, organized into discrete vertical slots, as in historiographs (where they are usually horizontal). Within a year, a vertical jittering function spreads the cases out to reduce link crossing and tunneling under nodes. The jittering function, which moves nodes up every 2nd and 4th slot and down every 1st and 3rd slot within a 4-slot period, was arrived at experimentally and was found to decrease link overlaps.

The cases, ranging from 1978 to 2005, were carefully selected by political science researchers eager to study patterns of precedents. Their numerous questions involve issues such as changing patterns of reference over time. For example, they seek to understand whether Supreme Court cases rely more heavily on lower courts (Circuits and Districts) now than in the past. Another task is to study evolving patterns of reference to a key Supreme Court 1978 case by later court cases at each level. The problem is complicated by distinctions among the 13 Circuit Courts, and 90+ District Courts, but for the purposes of this first example, we will show only Supreme and Circuit Court cases. For comprehensibility, we selected the 36 Supreme Court and 13 Circuit Court cases that were cited at least 45 times by other cases in this 2780 case corpus, thereby indicating their importance. Within these 49 cases there are 368 citations from 1978-2002. This is a modest sized network, but is already difficult to draw in a way that preserves visibility. Fig. 1 shows the hopelessly cluttered display as a result of using the JUNG's layout that uses the FR algorithm. Larger node sizes indicate greater number of citations to previous cases in the text of the case, but other attributes can be used. This layout is additionally problematic because the interesting cases with many in and out links are tightly woven together in the center and temporal patterns are difficult to assess.

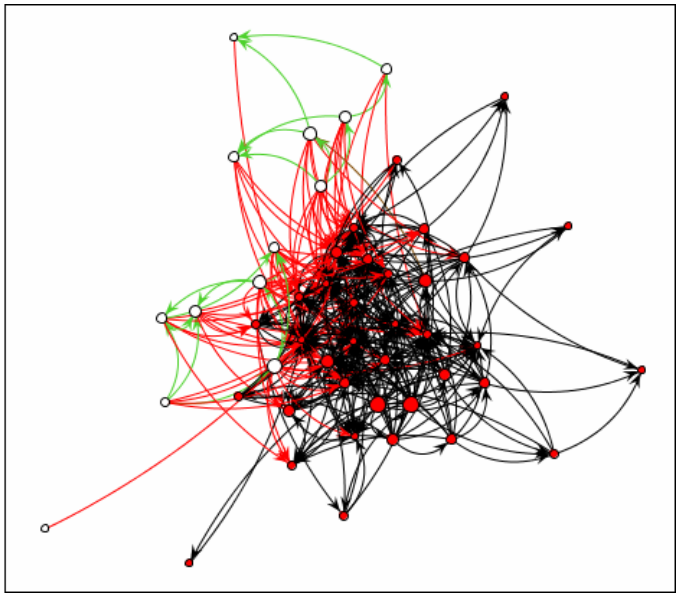

Fig. 1. Using JUNG's FR algorithm to place the 49 cases with all 368 citations makes it impossible to follow citations from source to destination or to see temporal patterns.

Using NVSS 1.0, we created regions for the Supreme and Circuit Court cases in temporal order with oldest on the left (Fig. 2). The controls for link visibility allow users to show the four flavors of citations: Supreme to Supreme (260 citations), Supreme to Circuit (1), Circuit to Circuit (18), and Circuit to Supreme (89). In this example, there is a highly asymmetric citing relationship, since Circuit Court cases are more likely to cite Supreme Court cases (89 times) than the other direction (only 1 time). To expose the number of citations across regions, NVSS includes numbers in the control panel, and includes a color key for the different kinds of citations.

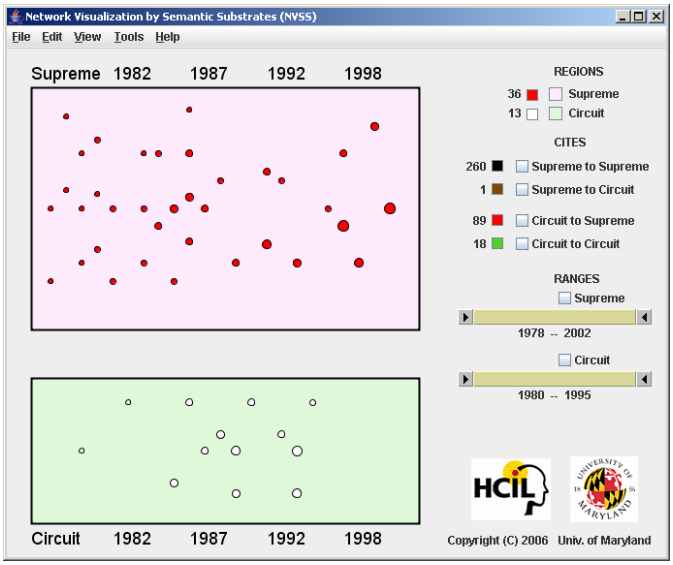

Fig. 2. Step 1 in simplification places nodes in regions without links. Supreme Court region has 36 cases from 1978-2002. Circuit Court region has 13 cases from 1980-1995.

In this example, the user controlled link visibility is best utilized to clearly display the single brown Supreme to Circuit citation and the 18 green Circuit to Circuit citations (Fig. 3).

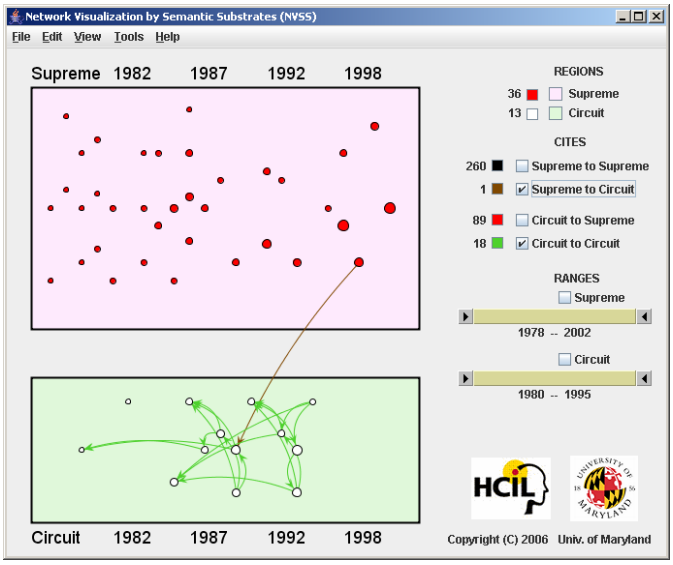

Fig. 3. Step 2 of applying interactive control with check boxes simplifies the display and shows just one brown Supreme to Circuit and 18 green Circuit to Circuit citations.

To cope with the clutter of the 260 Supreme to Supreme links, NVSS provides users with double-box dynamic query sliders to filter the year range for cases whose citations are displayed. Users can tightly limit the year range and then sweep through the full range of years for a satisfying animated overview (Fig. 4).

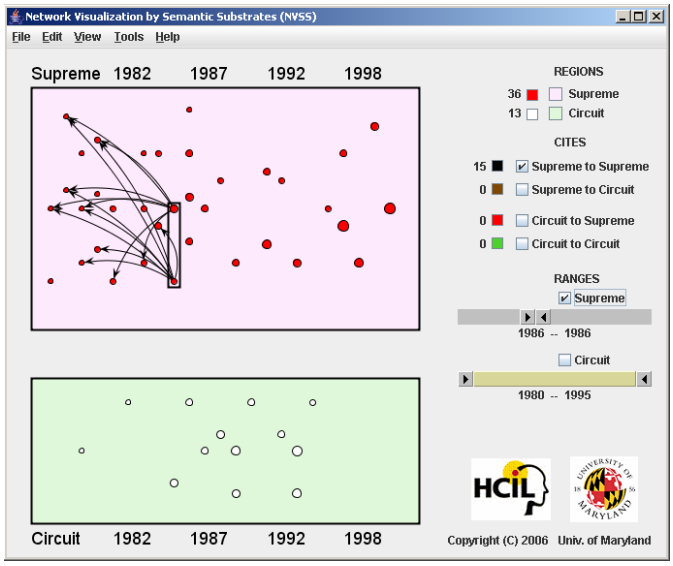

Fig. 4. Step 3 shows that even the clutter of Supreme Court cases is controlled by limiting to the 2 in 1986 with just 15 citations. Five cases are cited twice and 5 cases are cited once.

Sometimes, there are still quite a few citations and link visibility remains to be a problem.

The range selection works well across regions. By selecting the 1990 to 1991 Circuit Court cases using the Circuit Court slider, users can see the two citations to Circuit Court cases and the 18 to Supreme Court cases (Fig. 5).

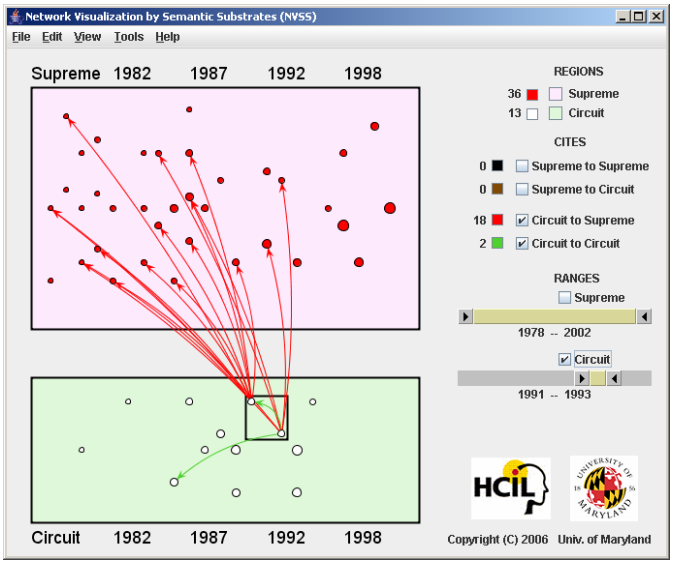

Fig. 5. Limiting the selected Circuit Court cases to the 2 in 1991-1993 generates a comprehensible display of the 18 red Supreme Court and the 2 green Circuit Court citations.

While citations in Fig. 5 are still comprehensible, sometimes the current link drawing strategy will need to be improved (Fig. 6). The close alignment of just the two Circuit Court cases makes the red citation links overlap, undermining visibility. Animated node movement or improved link routing are possible improvements.

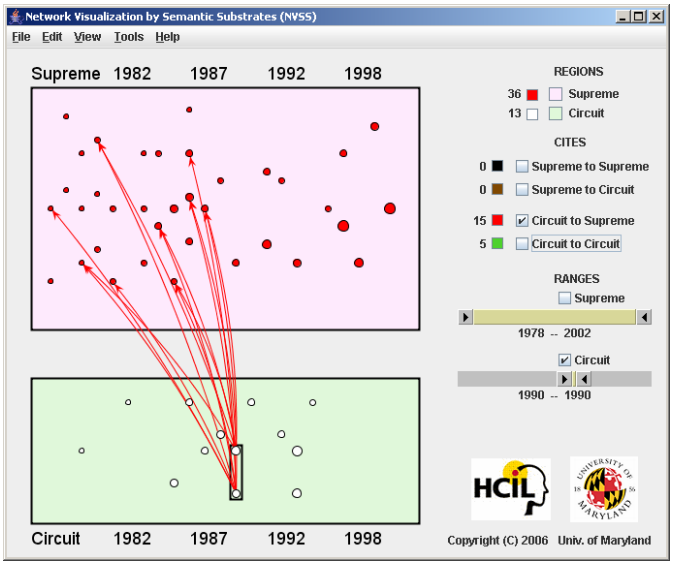

Fig. 6. Limiting the selected Circuit Court cases to the two in 1990 generates overlapped links to Supreme Court cases, suggesting the need for improved link routing strategies.

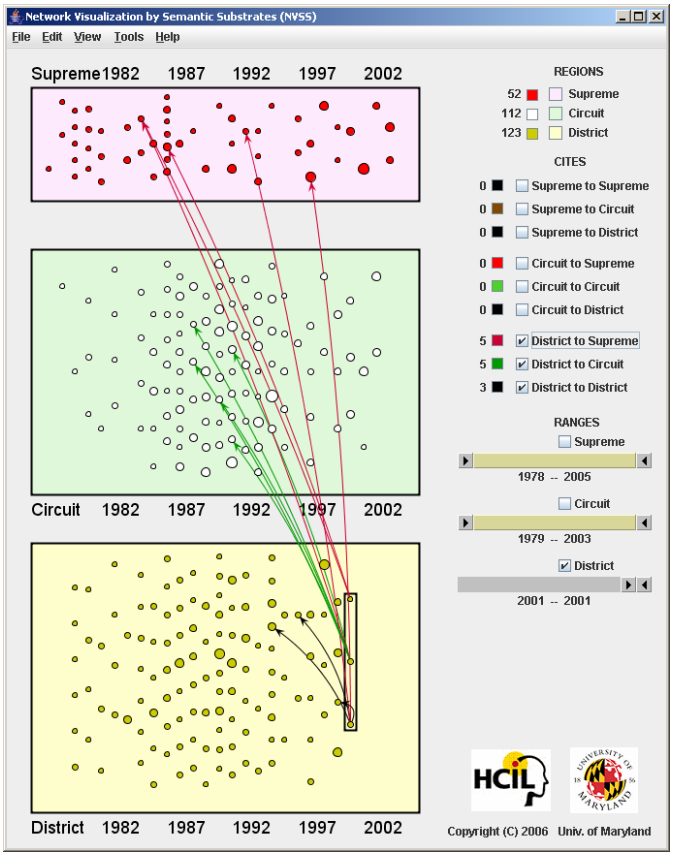

Fig. 7. Having District Court cases in a third region shows an anticipated referencing pattern, that is, District Court cases have a short reference half-life. This display shows 287 nodes and 2032 links.

Having more than two regions reveals more information (Fig. 7). In this court case example, a natural choice for the third region is to include the District Court cases. In Fig. 7, the data is a subset that consists of Circuit Court cases that are cited more than 15 times, District Court cases that are cited more than twice and all Supreme Court cases. The size of each region is proportional to the number of nodes it contains (52, 112, and 123 nodes for Supreme, Circuit, and District regions, respectively as displayed on the top left corner.).

By limiting the District Court cases to the year 2001 and enabling all the links from the District Court region shows that this set of recent cases tend to cite Circuit Court cases that are between 1989 and 1992, whereas they cite Supreme Court cases that fall into a wider range of duration in history. Sweeping the District Court cases from left to right reveals a general tendency to cite only recent Circuit court cases (i.e. earlier Circuit Court cases are not cited). In contrast, both recent and old Supreme Court cases are cited. Sweeping the Circuit Court cases from left to right reveals a similar pattern supporting the hypothesis that "Supreme Court cases have a long-standing effect, while Circuit Court cases are influential for a shorter period of time in the regulatory takings cases domain." Our political science partners were pleased to see that the visual display added support to some of their conjectures such as this one about citation patterns for precedents. Furthermore, they were surprised to detect patterns that were not very clear before. For example, they discovered that depending on the court type, there is an approximate duration (in years) within which cases are more likely to be cited by future cases. If we call this number the "expected longevity" of a case, it is very unlikely for a case to be cited beyond its expected longevity. However, when it happens, it raises questions in mind as to what factors make the exception to the rule occur. One question that our collaborators had was whether these exceptional cases coincide with the most cited cases in the dataset, which indicates high importance.

The expected longevity of Supreme, Circuit, and District Court cases reveals itself when links are limited to one region and users limit originating links to 1-2 years and sweep the filtering box from left to right (past to future years). It is apparent that the expected longevity of a case depends on its court type and it is in increasing order from lower to higher level (District, Circuit, and Supreme) courts. In addition, the exceptional cases, the ones that are cited beyond their expected longevity, are discernable on the display and can be noted for further exploration by other methods.

In the precedent domain, another feature of interest is the jurisdiction, or circuit of a case (applies only to Circuit and District Court cases). To use this feature, NVSS can arrange the cases in horizontal bands according to their circuit, ranging from first to eleventh, DC, and federal circuit from top to bottom, forming a total of 13 horizontal bands (Fig. 8). This immediately reveals the expectation of our collaborators, which is "Circuit Court cases are more likely to cite within their circuit". Accordingly, links across bands are dominated by links within bands in Fig. 8. A similar hypothesis for the District Courts is also revealed by the visualization (that District Courts are likely to cite District Court cases that belong to the same circuit). Another outcome was that the 9^{th} and the Federal circuit were active and important, which was indicated by incoming citations.

Our collaborators were excited when they discovered unfamiliar or unexpected relationships and patterns in this setting. Sweeping among the years revealed to them that although both the Federal Circuit and the 9^{th} circuit were active, they differed in terms of incoming citations from other circuit courts. While the 9^{th} circuit was receiving many incoming citations from the other courts over the years, the Federal Circuit rarely did so. On the contrary, almost all incoming citations were within the Federal Circuit. Another outcome was the effect of the number of cases within a year and a circuit over the number of incoming citations. Visualizing and comparing the links over the years to such groups of cases suggests that the number of incoming links to the cases (their popularity) increase – perhaps unfairly – as the number of cases increases given a year and a circuit.

Interaction is smooth with more than 1,000 nodes and 7,500 links, which are displayed in Fig. 9. In this case, all Circuit Court and District Court cases that are cited at least once and all Supreme Court cases are included. When there is available screen space, users may want to utilize it to see nodes and links more clearly. Fig. 9 shows a still larger data set with 1,122 nodes and 7,645 links at a 1280x1024 resolution.

8 Conclusions and Future Work

Our organization of 6 network visualization challenges with associated tasks enables us to formulate an interface for NVSS 1.0. The interface allows users to specify regions, then to lay out nodes in those regions. This strategy will help users to cope with the complexity of large numbers of nodes and links. There are limitations in our implementation, but the utility of semantic substrates seems apparent, at least for some datasets and tasks. We believe that the partitioning of a large network into several smaller ones defined by non-overlapping regions facilitates completion of required tasks more rapidly and reliably. The enthusiastic comments of our political science partners support our conjecture. They were able to quickly identify patterns of interest, and are guiding the evolution of NVSS.

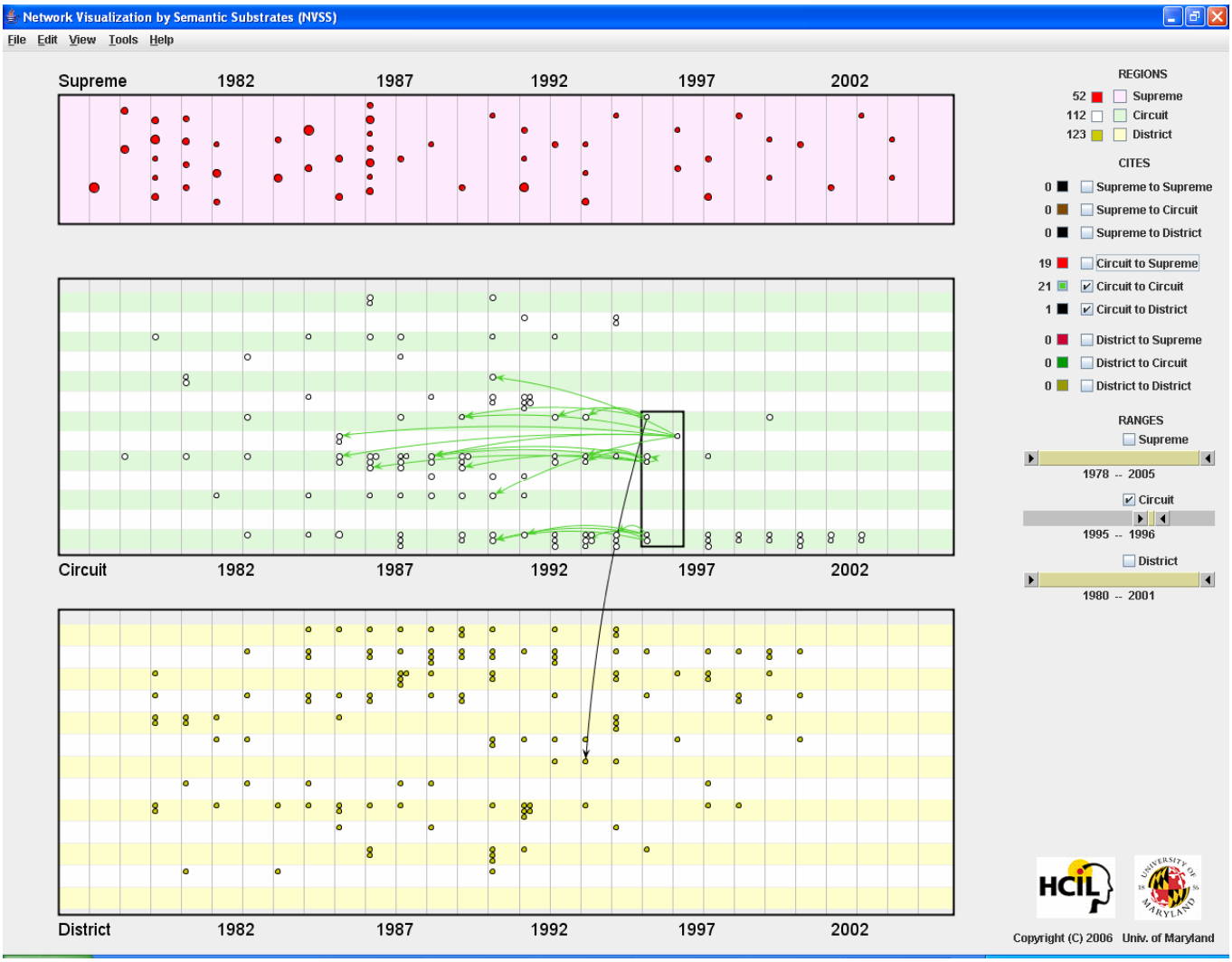

Fig. 8. The layout for Circuit Court cases is now organized by the 13 Circuits and the link pattern shows the strong likelihood that cases will reference precedents within the same Circuit.

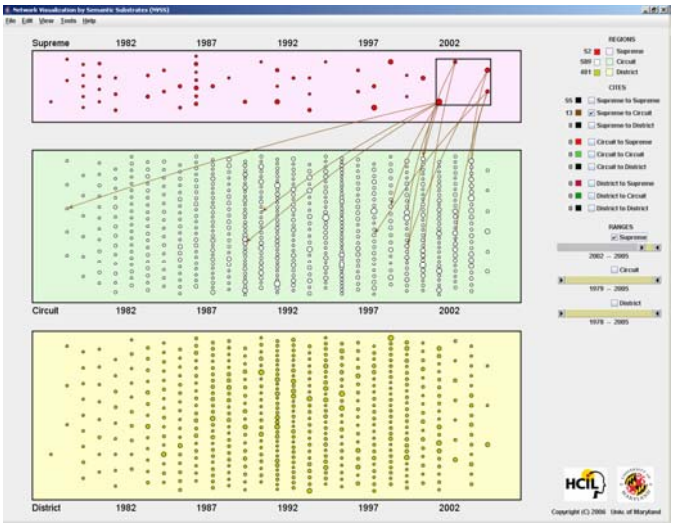

Fig. 9. Displaying 1,122 nodes and 7,645 links at a 1280x1024 resolution. The relatively small number of Supreme Court cases is apparent, as is the similar number of Circuit and District Court cases.

Distributions within years are also visible, enabling users to see the ebb and flow of activity.

As with many new ideas, there are numerous refinements that are needed. Designs for 3, 4, and 5 regions get more complex but we are finding strategies to deal with them.

In this example, our collaborators were certain about the important attributes, which we used as ingredients determining placement. In general, however, there may be many attributes and that users may have little awareness of which attributes are best to use to determine regions and placement for their task. Considering that users with such data exist, a user interface to help users explore combinations of attributes seems to be a promising future direction.

We have a plan for an iconic representation that would replace multiple check boxes, allowing easy selection of links within or between up to 5 regions.

The NVSS implementation is still developing and more features are needed in the user interface to simplify the specification of region size, location, color, labels, node layout strategy, etc. In addition, greater flexibility will certainly be needed for node, link, and label properties such as placement, size, color, font, and background. We plan to add dynamic properties to control node and link visibility, plus infotips, excentric labels, and window panes for textual lists.

Future work might also include elastic window strategies that enable users to enlarge one region while shrinking the others in a smooth animation [25]. For networks with millions of nodes, further work is needed on dynamic query sliders to limit node visibility

while preserving comprehensibility. A major new challenge is to improve link routing between regions to ensure comprehensibility. Overlapping regions to represent nodes with multiple attributes are possible, and may be helpful for certain situations.

While all these challenges remain before us, we have a strong sense of attractive new possibilities for network visualization based on semantic substrates. User-defined regions create some new problems, but they are proving to be beneficial in some application domains.

ACKNOWLEDGEMENTS

We appreciate the invaluable collaboration of Prof. Wayne McIntosh in the Department of Government & Politics at the University of Maryland and his students Ken Cousins and Stephen Simon. The U.S. National Science Foundation grant "Inter-Court Relations in the American Legal System: Using New Technologies to Examine Communication of Precedent II" provided partial support. We appreciate thoughtful and detailed comments from Jeffrey Heer, Hyunmo Kang, Bongshin Lee, Stephen North, Cynthia Parr, Adam Perer, and the anonymous reviewers.

REFERENCES

[1] R.A. Becker, S.G. Eick, and A.R. Wilks, Visualizing Network Data. *IEEE Trans. on Visualization and Computer Graphics*, 1(1), 16-28, March 1995.

[2] B. B. Bederson, J. Grosjean, and J. Meyer, Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering* 30 (8), 535-546, 2004.

[3] C. Best and H.-C. Hege, Visualizing and Identifying Conformational Ensembles in Molecular Dynamics Trajectories. *Computers in Science and Engineering* 4 (3), 68, 2002.

[4] M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman, D-Dupe: Entity Resolution in Networks, *IEEE Symposium on Visual Analytics Science and Technology*, 2006.

[5] K. Börner, C. Chen, and K. Boyack, Visualizing Knowledge Domains. *Annual Review of Info. Science and Technology 37*, 2003.

[6] U. Brandes and D. Wagner, Visone: Analysis and Visualization of Social Networks, In: M. Juenger and P. Mutzel, editors, *Special Issue on Graph Drawing Software*, Springer Series in Mathematics and Visualization, 321-340, Springer-Verlag, 2003.

[7] B.-J. Breitkreutz, C. Stark, and M. Tyers, Osprey: a network visualization system, *Genome Biol.* 4(3): R22, 2003. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=153462

[8] C. Chen, Bridging the Gap: The Use of Pathfinder Networks in Visual Navigation, *Journal of Visual Languages and Computing 9*, 3, 267-286, 1998.

[9] R. Davidson and D. Harel, Drawing Graphs Nicely Using Simulated Annealing, *ACM Trans. on Graphics* 15, 4, 301-331, 1996.

[10] W. De Nooy, A. Mrvar, V. Baragelj, and M. Granovetter, *Exploratory Social Network Analysis with Pajek*, Cambridge Univ. Press, U.K., 2005.

[11] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: algorithms for the visualization of graphs*. Prentice-Hall, 1999.

[12] P. Eades, A Heuristic for Graph Drawing, *Congressus Numerantium* **42**, 149-160, 1984.

[13] P. Eades and Q.-W. Feng, Multilevel Visualization of Clustered Graphs. *Proc. Graph Drawing, LNCS 1190*, 101-112, 1996.

[14] T.M.G. Fruchterman and E. Reingold, Graph Drawing by Force-Directed Placement, *Software-Practice and Experience* **21**:11, 1129-1164, 1991.

[15] E. Gansner and S. North, Improved Force-Directed Layouts, *Proceedings of Graph Drawing 98*, LNCS **1547**, 364-373, 1998.

[16] E. Garfield, Historiographic mapping of knowledge domains literature, *Journal of Information Science* 30 (2): 119-145, 2004.

[17] M. Ghoniem, J.-D. Fekete, and P. Castagliola, A Comparison of the Readability of Graphs using Node-Link and Matrix-Based Representations. *Proc. IEEE Symposium of Information Visualization 2004*, 17-24, 2004.

[18] R. Hadany and D. Harel, A Multi-Scale Method for Drawing Graphs Nicely, *Discrete Applied Mathematics* 113, 3-21, 2001.

[19] D. Harel and Y. Koren, A Fast Multi-Scale Method for Drawing Large Graphs. *Proceedings of Graph Drawing 2000*, LNCS 1984, 183–196, 2000.

[20] D. Harel and Y. Koren, Drawing Graphs with Non-Uniform Vertices. *Proc. Conf. on Advanced Visual Interfaces* (AVI'2002), Trento, Italy, ACM Press, 157-166, 2002. http://www.wisdom.weizmann.ac.il/~dharel/papers/non_uniform_avi_acm.pdf

[21] J. Heer and D. Boyd, Vizster: Visualizing Online Social Networks, *Proc. IEEE Symposium on Information Visualization*, IEEE Press, Piscataway, NJ, 33-40, 2005.

[22] B. Huffaker, E. Nemeth, and K. Claffy, Otter: A General Purpose Network Visualization Tool http://www.caida.org/tools/visualization/otter/paper/. 1999.

[23] T. Kamada and S. Kawai, An Algorithm for Drawing General Undirected Graphs, *Information Processing Letters* 31, 7-15, 1989.

[24] T. Kamps, J. Kleinz and J. Read, Constraint-Based Spring-Model Algorithm for Graph Layout, *Proceedings of Graph Drawing 95*, LNCS 1027, 349-360, 1995.

[25] E. Kandogan and B. Shneiderman, Elastic Windows: Design, Implementation, and Evaluation of Multi-Window Operations, *Software: Practice & Experience 28* (3), 225-248, 1998.

[26] H. Kang and B. Shneiderman, Personal Media Exploration: A Spatial Interface Supporting User-Defined Semantic Regions. *Journal of Visual Languages and Computing 17*(3), 254-283, 2006.

[27] C. Kosak, J. Marks, and S. Shieber. Automating the Layout of Network Diagrams with Specified Visual Organization. *IEEE Trans. on Systems, Man and Cybernetics, 24*(3), 440-454, 1994.

[28] B. Lee, M. Czerwinski, G. Robertson, and B.B. Bederson. Understanding Research Trends in Conferences using PaperLens. *Extended Abstracts of CHI 2005*, ACM Press, New York, 1969-1972, 2005.

[29] M. McGuffin and R. Balakrishnan, Interactive Visualization of Genealogical Graphs, *Proc. IEEE Symposium on Information Visualization*, IEEE Press, Piscataway, NJ, 17-24, 2005.

[30] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout Adjustment and the Mental Map. *Journal of Visual Languages and Computing*, 6:2, 183–210, 1995.

[31] T. Munzner. Interactive Visualization of Large Graphs and Networks. PhD thesis, Stanford University, 2000.

[32] B. Nardi, S. Whittaker, E. Isaacs, M. Creech, J. Johnson, and J. Hainsworth, ContactMap: Integrating Communication and Information Through Visualizing Personal Social Networks. *Communications of the ACM*, 45(4), 89-95, April 2002.

[33] A. J. Pretorius and J.J. van Wijk, Multidimensional Visualization of Transition Systems. *Proc. 9th Int'l Conf. Information Visualization*, 323-328, 2005.

[34] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Trans. on Computer-Human Interaction 3*(2), 162-188, 1996.

[35] M. A. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protege. In *Workshop on Interactive Tools for Knowledge Capture*, Victoria, B.C. Canada, October 2001. Available at http://sern.ucalgary.ca/ksi/K-CAP/K-CAP2001/

[36] K. Sugiyama, S. Tagawa, and M. Toda. Methods for Visual Understanding of Hierarchical Systems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(2):109–125, 1981.

[37] M. Wattenberg. Visual Exploration of Multivariate Graphs. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 811-819, 2006.

[38] G. Wills. NicheWorks – Interactive Visualization of Very Large Graphs. *Journal of Computational and Graphical Statistics 8*(2), 190-212, 1999.