

Network Visualization by Semantic Substrates

Ph.D. Research Proposal and Reading Lists

Aleks Aris
May 11, 2006

Advisory Committee: Ben Shneiderman, Chair
Wayne McIntosh
Doug Oard
Amitabh Varshney

Network Visualization by Semantic Substrates

Abstract

Network visualization and graph drawing research has concentrated on improving the visual organization of nodes and links according to graph drawing aesthetics. Although these approaches improve layouts in terms of minimization of link crossings, the longest link length, and other such criteria, these do not always lead to comprehensible layouts.

This proposal focuses on placing nodes according to their attributes, which is an interpretable node placement rule that conveys information about the nodes. The author believes that this will increase the comprehensibility of a network. Another room for improvement arises when there are too many links to draw. To improve comprehensibility, the author proposes new approaches based on (1) user controlled visibility of links and (2) novel *linkviz methods*, which avoid drawing links.

The proposal applies the interpretable node placement rule in a practical and conceivable form: *semantic substrates*. A semantic substrate is a template that defines the placement of nodes. This template consists of rectangular regions in which nodes are placed according to their attribute values. The semantic substrate idea is applied to an example network dataset of precedent patterns, where 2780 nodes represent court cases and 16,000+ links represent citations. This dataset has been explored by NVSS 1.0, a Java program that was implemented by the author. The proposed work consists of pushing this idea from the prototypical example to a more general form, where other datasets can be visualized this way. The plan includes a substrate editor for users to create their own substrates in order to leverage their experiences and understanding, enabling them to express the representation in their mind. Furthermore, features such as overview and interactive filtering are considered for scalability. Evaluation is proposed in the form of case studies to confirm benefits and obtain feedback, especially about semantic substrates, linkviz methods, and the interactive features.

The contributions of this research will be: (1) design principles for semantic substrates, linkviz methods, and user interaction, (2) software and algorithms that implement the ideas, and (3) a summary of case studies illustrating the experience of the use in application domains. The outcome will also include insights, suggestions, and ideas for future work that potentially could benefit designers, developers, and researchers.

Table of Contents

Abstract	2
1. Introduction.....	4
2. Previous Work	8
2.1 Graph drawing aesthetics	8
2.2 Node placement strategies	9
2.3 Inspirations for semantic substrates	11
3. Current State	12
3.1 Categorization of 100 network visualizations.....	12
3.2 Exploration of precedent data using NVSS 1.0	14
3.3 Experience and insights regarding user tasks.....	25
3.3.1 Experience from court precedent users.....	26
3.3.2 Other Domains.....	27
4. Planned Work	28
4.1 Substrate generation.....	28
4.2 LinkViz Methods: An alternative approach to drawing links	29
4.3 Scalability	33
4.3.1 Node and link scalability	33
4.3.2 Scalability of the number of regions.....	34
4.3.3 Scalability in terms of node attribute values.....	34
4.4 Other issues.....	35
5. Evaluation.....	35
6. Research Outcomes and Expected Contributions.....	36
7. Detailed Plan of Work	39
8. Bibliography	41
9. Reading Lists	45
9.1 Information Visualization	45
9.2 Network Visualization	45
9.3 Measurement/Evaluation Methods, Examples, and Criteria.....	46

1. Introduction

Networks are found in different forms across various application areas, such as court cases that cite another, biological food webs, social networks, and legal precedent data (court cases that refer to one another). Network visualizations enable users to see the pattern of nodes and links, detect interesting cases, analyze, interpret and arrive at conclusions.

The terms *network* and *graph* are used interchangeably in the literature. *Network visualization*, also known as *graph visualization*, is a visual presentation of data and the relations between them. Network visualization can be applied only to data that is in the form of objects and relations between these objects. The visual representation of an object is usually a shape such as a circle, a square, or a rectangle, which is generically referred to as *a node*. The relation between two nodes is depicted via a line, a curve, or a similar representation, which is generically referred to as *a link* (also known as *an edge*).

Figure 1 shows a typical network visualization, where nodes are represented as squares and links are represented as gray lines. Nodes are companies that had a role in Apple's popular product, iPod, in 2001. Red is used for accessory makers, while blue is used for technology providers, and green is used for competitors. Links show communication between companies.

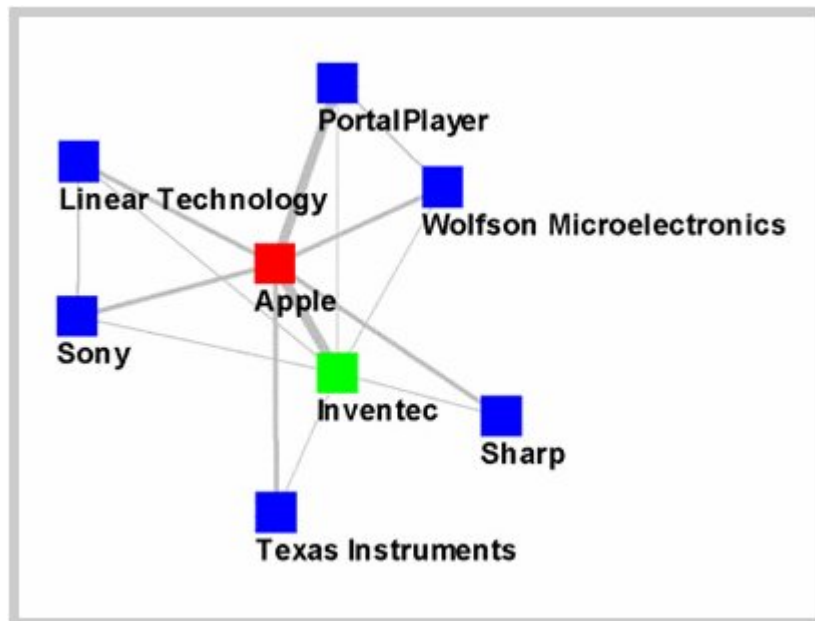


Figure 1 A network visualization that shows companies and communications between them.
Source: Matt Woolman, 2005,
<http://www.visualcomplexity.com/vc/project_details.cfm/?index_number=152&id=164&DomainName=>.

Network visualization is superior to textual representation for some tasks because it utilizes the capabilities of human's visual perceptual system. A well-designed network visualization enables fast and accurate interpretation and overview. While network visualization has the potential for fast and better understanding, presentation becomes a challenge as the number of nodes and links increases. Labels are no longer possible to show and the display begins to get too cluttered to comprehend (see Figure 2).

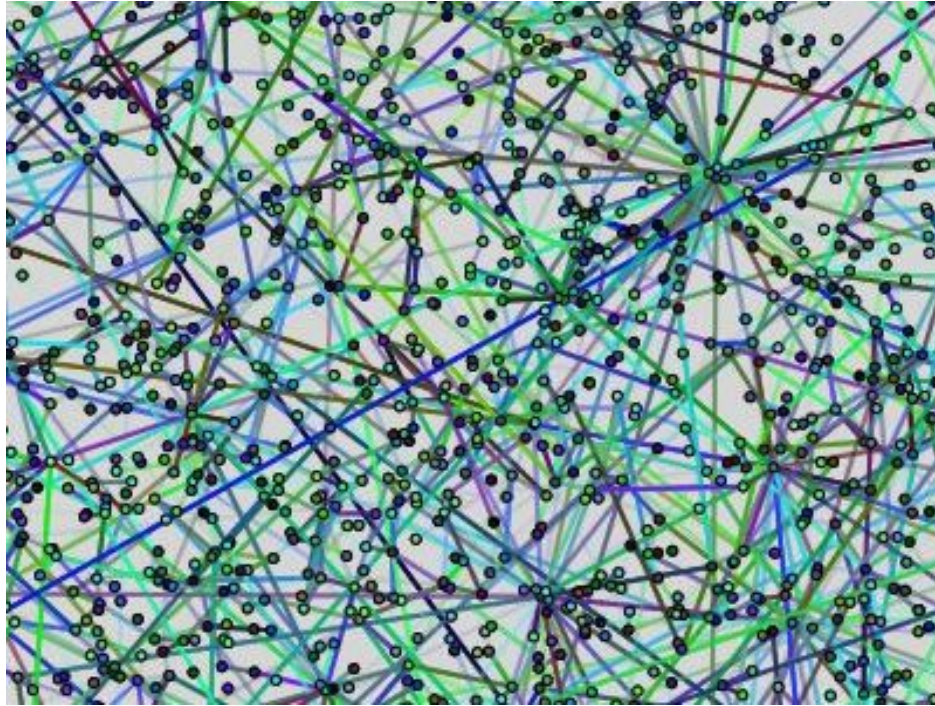


Figure 2 Graphs become harder to understand as the number of nodes and links increase.
Source: Matt Woolman, 2005,
<http://www.visualcomplexity.com/vc/project_details.cfm/?index_number=135&id=146&DomainName=>.

Much research exists on network visualization, part of which is focused on improving the presentation. Given a network, the number of different arrangements to draw it is virtually unlimited. One trail of research concentrates on graph drawing aesthetics, criteria for drawing networks for optimal perception and, thus, understanding. Another trail of research –implicitly or explicitly– uses these aesthetics and provides algorithms that strive to conform to these aesthetics when drawing networks. Some of these algorithms are concerned with efficiency, as well, since it is computationally expensive to achieve some of these aesthetics (Brandenburg 1988). For example, determining the number of link crossings is NP-complete (Gary 1983).

Given a network, as the number of nodes and links grow, it becomes increasingly difficult to display it on a computer screen. With the large number of possibilities for existing links, arranging nodes such that users' perception of them is optimal (fast and as clear as possible rather than confusing) becomes a challenging problem that many algorithm designers are trying to solve. Likewise, users have an increasingly harder time to understand and interact with networks, especially as the complexity increases.

There are possibly many reasons that a network is hard to understand. Low conformity to graph drawing aesthetics comprises one set of reasons. Another reason that the author claims is the seemingly arbitrary placement of the nodes on the screen. Users tend to associate the place of a node with the object the node represents as they are looking at a node. Since nodes represent objects, a node consistently placed in the same location is likely to become familiar over time, where users start to use its location to interpret it. Furthermore, when nodes with similar characteristics are placed close to each other, they can be conceptualized as a group when needed, which helps users abstract several nodes into one object (the group), which reduces complexity at the expense of detail. Reducing complexity in this way is plausible especially when the detail that is removed is not needed at the time of the abstraction. For example, cities on a map of the United

States are tightly associated with their places. There is a direct association between the place of a node on the display and the place of the city in the world. Place becomes a cue for identifying a city. Cities that are in the same state are usually close to each other when compared to cities in other states. In this example, one possible grouping is in terms of states, where the abstraction moves the level of detail from cities to states.

Arbitrary placement of nodes in a network may confuse users especially if users assume that there is a simple logical explanation for the location of nodes. The spatial location of a node can be leveraged to convey information about the node to the user. Using attribute values of nodes to determine where to place the nodes provides not only a simple logical explanation for the node location on the display but also information about the node. Using placement of nodes in this way, the comprehensibility of networks can be enhanced leading to possible increases in productivity and user satisfaction.

Other strategies –rather than the placement method– such as overview, aggregation, and filtering could be used to deal with the complexity of the network. This way, the comprehensibility of the network can be improved by using placement to convey information and by using other strategies than placement to deal with the complexity of the network.

The approach this research conveys information about nodes is to use attribute values of nodes to determine a spatial layout. We started exploring this possibility with researchers from the Government & Politics department. A sample network shows the precedent patterns, where a node represents a court case and a directed link represents a citation from one court case to another. Nodes have attributes such as date, name, court type, and court name. Court type (Supreme or Circuit) is used to group the nodes into regions and dates to further arrange the nodes inside the regions. Region and size along with the method to arrange the nodes inside each region define a *substrate*. Such a substrate is said to be *semantic* as nodes are laid out according to their attributes within regions. Since the application uses semantic substrates, it is called NVSS, short for Network Visualization by Semantic Substrates.

User tasks are highly important in a network visualization domain as it is in any human-computer interaction field. Understanding users and their needs is essential to solving improve their experience and improve the effectiveness of tools. While exploring one set of users, who are our Government Politics collaborators, from our point of view, the user needs and tasks can be summarized in terms of the following four major categories:

- *Understanding the network and its elements (nodes and links).* Regions and good node placement methods within regions are directed to fulfill this need.
- *Being able to manipulate the network layout and the visual properties.* The author believes that the best strategy to determine regions, placement strategies, and visual properties is to allow users to define and manipulate them. For the definition of those, a substrate editor is envisioned, where users will be able to define regions, their placement, size, the node placement methods used within each of them, size encoding of nodes and color encoding of nodes and links by type.
- *Being able to understand the relationships between nodes and node groups.* In a typical network visualization using node-link diagrams, the relationships between nodes are depicted via lines or curves connecting the nodes that are related to one another. The author believes that with the introduction of meaningful placement methods alternative methods besides drawing links emerges as a promising direction to explore (which was not promising before, that is, without using meaningful placement methods). In addition, the same factor (meaningful placement methods) imposes restrictions on links that

drawing of them as lines or curves is likely to diminish their readability when they are represented as lines or curves. An initial taxonomy of link tasks is provided in section 4, which may be extended over time.

- *Scalability.* Users are likely to need to explore larger networks. Challenges arise as the number of nodes and links grow. The plan includes addressing those challenges and providing solutions that cope with some of them. Although generally overview, zoom, and filtering are known as general solutions, those take a different form in the context of networks, and more so with the introduction of semantic substrates.

Section 2 discusses previous work while section 3 describes the current state of this research. Then, section 4 provides the plan for the future work. Next, section 6 points out the research outcomes and expected contributions. Finally, section 7 provides more details on the plan.

2. Previous Work

Section 2.1 reviews the history of the graph drawing aesthetics literature. Section 2.2 provides the node placement strategies used in the network visualization literature, while section 2.3 indicates inspirations for the semantic substrate idea.

2.1 Graph drawing aesthetics

Several researchers recognized the importance of improved graph presentation as the number of nodes and links increases. In fact, even with small graphs, a bad layout can make a graph difficult to comprehend. While it is difficult to comprehend the graph in Figure 3a, it becomes considerably easier to perceive the structure when its layout is improved as in Figure 3b.

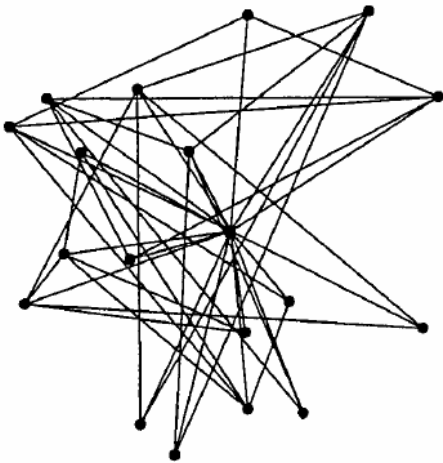


Figure 3a A graph with a bad layout
Source: (Davidson & Harel, 1996)

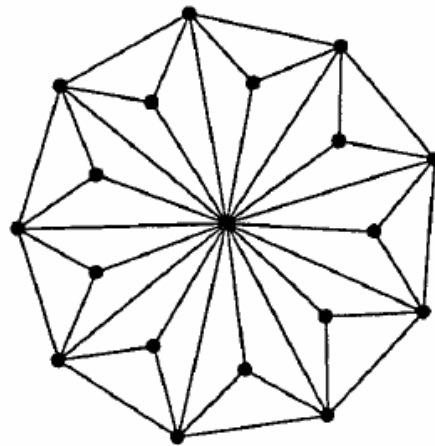


Fig 3b Improved layout of the graph in Fig 3a
Source: (Davidson & Harel, 1996)

Researchers used the term *graph drawing aesthetics* to pinpoint the criteria that make a graph easier to perceive. The seminal publication of Sugiyama et al. defined graph drawing aesthetics to improve the readability of a graph drawing (Sugiyama 1981). They provided the following graph drawing aesthetics:

- *Minimization of the number of link crossings.* Drawing a graph to minimize the number of edges that cross each other so that it is easy to follow the links.
- *Proximity of nodes that have a connection.* Placing the nodes that have a connection close to each other so that the lengths of links are minimized.
- *Straightness of lines.* Drawing links as straight as possible, avoiding bends or minimizing the number of bends.
- *Balanced drawing of links.* Leaving as much even spacing as possible between the links that are incident to a node.
- *Hierarchical layout of nodes.* Introducing levels, imaginary horizontal or vertical lines, and placing the nodes on these levels.

They also defined the terms *regularity* and *traceability* as follows (Sugiyama 1987):

- *Regularity*. Placing the nodes according to a principle, such as grouping them together according to some criterion, and applying this principle consistently throughout the graph.
- *Traceability*. Drawing the links such that it is easy to follow paths (connected links).

Later, Sindre et al. provided a list of graph drawing aesthetics, which additionally included the following (Sindre 1993):

- *Minimization of area*. Saving as much space as possible while drawing the graph.
- *Centralization of high-degree nodes*. Placing the nodes that have a high degree (i.e. are connected to a high number of other nodes) in the center of the drawing.
- *Uniform density of nodes*. Placing the nodes such that their distribution leads to the same number of nodes per unit area.
- *Maximization of convexity*. Drawing the links such that they form convex polygons.

Their list includes a few more aesthetics, which are either specific to one type of graph drawing, such as *verticality in hierarchical structures* (it is specific to graph drawings that have a hierarchical structure), or it is an elaboration of an already listed aesthetic, such as *the minimization of the longest edge* (a further elaboration of proximity of nodes that have a connection).

In 1996, Purchase et al. (Purchase 1996) investigated five aesthetics to confirm their empirical validity: symmetry, orthogonality, maximization of the minimum angle, minimization of the number of edge crossings, and straightness of lines. While the latter two are defined above, the former three are defined as follows:

- *Symmetry*. Drawing the parts of a graph that have the same structure in the same way and placing them in balanced directions (left – right, top – bottom, or multiple opposite directions) within the graph whenever possible.
- *Orthogonality*. Placing nodes on the intersections of a grid and allowing links only to be drawn on the edges of the grid.
- *Maximization of the minimum angle*. Placing the nodes such that any two links that are incident to the same node in the graph have as large an angle as possible.

In a follow up study, Purchase found that the most important among these aesthetics was the *minimization of the number of link crossings*, while the other two were less important (Purchase 1997) but did not take *good continuity* into account, which was defined later by Ware et al. (Ware 2002):

- *Good continuity*. Keeping multi-link paths as straight as possible.

Their study revealed that good continuity was even more important than the number of link crossings, especially when users need to identify shortest paths within a graph. This finding suggests that allowing a few additional link crossings to draw paths straighter could increase the understanding of a graph. Moreover, minimizing the number of links that cross a path was found to be more promising for better perception than minimizing the number of link crossings in the entire graph.

2.2 Node placement strategies

There is a huge literature on network visualization (Eades 1984; Di Battista 1999; Herman 2000) and entire conferences devoted to the topic, such as the 13-year old International Symposium on Graph Drawing (<http://www.gd2005.org/>). Force-directed strategies have dominated the literature

because they produce elegant spreading of nodes and reasonable visibility of links. The electrical-like forces are repulsion among nodes, opposed by the attraction of the links. Eades (Eades 1984) proposed the idea but the most common reference is to the refined algorithm by Fruchterman and Reingold (Fruchterman 1991), with further refinements by many others (Gansner 1998). Variations are sometimes called spring-embedding to describe the connections between every pair of nodes (Kamada 1989; Kamps 1995) or simulated annealing, which alludes to the process of heating and cooling metals (Davidson 1996; Harel 2000). Multi-scale algorithms (Harel 2000; Hadany 2001) are scaling up of force-directed methods which work on a coarse version of a large network and then refine the layout locally to achieve remarkably rapid layout for large networks (10^6 nodes in a few seconds).

A second common layout strategy is based on a geographical map in which the node locations are fixed, as in cities on a world map. These strategies generate familiar and comprehensible layouts (Becker 1995; Misue 1995).

A third common strategy uses a circular layout for nodes that produces an elegant presentation with crisscrossing lines through the center of the circle (Huffaker 1999; Breitkreutz 2003). Multiple concentric strategies 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 (Becker 1995; Ghoniem 2004). Such representations avoid some of the problems of node-link diagrams (especially with large graphs), such as node occlusion, link crossings, and links 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 (Lee 2005).

Meaningful groups of nodes can be formed by hand (Nardi 2002) or algorithmically (Heer 2005) 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. (Schaffer 1996) 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 2-dimensional plane (Eades 1996). While such an approach promises an increase in comprehension, problems of occlusion and finding the best view-angle may pose challenges with larger graphs. These and other clustering approaches (Best 2002; Borner 2003) have some commonality with semantic substrates, but, by contrast, in the semantic substrate approach groups are formed 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, sometimes called historiographs (Garfield 2004). 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 (De Nooy 2005). Similar looking layered layouts have long been in use (Sugiyama 1981; Brandes 2003), but these layers are based only on links. Kosak et al. (Kosak 1994) 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 considered the importance of stability of the network layout and suggested methods to preserve user's mental map when additions or changes to a network (such as adding a node, adding an edge, or expanding a cluster) are made (Misue 1995).

Two recent systems have elements of semantic substrates. Jambalaya (Storey 2001) 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 (Wattenberg 2006) places nodes on a two dimensional grid by their node attributes and nicely aggregates nodes by their attributes to present a useful overview.

2.3 Inspirations for semantic substrates

The notion of user-defined semantic substrates proved beneficial in a network visualization tool for author name resolution in bibliographic database (Bilgic 2005). 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 are the user defined spatial layouts for photos with shared attributes (Kang 2005).

3. Current State

The following sections provide the work completed for this research. Section 3.1 provides the categorization of 100 network visualizations in terms of node placement methods they used. Section 3.2 explains the exploration of precedent data using NVSS 1.0, the network visualization tool developed to apply and explore the semantic substrate idea.

3.1 Categorization of 100 network visualizations

To gain an insight about the placement methods utilized in common practice, 100 network visualizations were analyzed and categorized in terms of the method they used for node placement. These 100 network visualizations were selected from Visual Complexity (Woolman 2005), a web site that lists more than 200 graph visualizations. The first 100 network visualizations were selected. The method for node placement was not specified for 25 of the visualizations. Some of the projects were submitted without an author and without a link. Some of the links were not available and for some visualizations no related publication in the literature seemed to exist. The rest of the visualizations were categorized. Table 1 summarizes the results. The second column indicates the abbreviation for the category, the third column shows how many visualizations are placed in this category, the fourth column shows the percentages (calculated as 3rd column / 75), while the last column lists how many of the visualizations (the number in parentheses; 1 if there is no parentheses) that fell into this category were also categorized in another category.

Table 1 Categorization of 75 of the 100 network visualizations on the Visual Complexity web page

Category	Abbreviation	Frequency	Percentage	Also categorized as
Force-directed	fd	25	33%	kx
Geographical map	gm	20	27%	hx(2), sx
Circular	cx	12	16%	sb, tx, dx, sx
Hand-made	hx	12	16%	gm(2)
Spatial calculated	sx	6	8%	cx(2), gm
Clustering	kx	4	5%	fd
Time-oriented	tx	2	3%	cx
Substrate based	sb	1	1%	cx
Random	rx	1	1%	

Force-directed algorithms emerged as the most frequently used method (33%). One of those also used a clustering method. The second most frequently used method was a geographical map. While most of the visualizations in this category used a map showing the whole world, a continent, or a few countries, a few of them used a city or a room as a map. Visualizations that did not show a map on the background but still transformed geographical properties of nodes, such as latitude and longitude, to screen coordinates were included in this category.

Visualizations in the circular category used concentric circles or a single circle sometimes with a node or a collection of nodes in the center. Some of these visualizations specify the order of the nodes. Among the visualizations that use concentric circles, one node was designated as a root node and it was placed in the middle. The connected nodes to this node were placed at the closest enclosing circle, and this principle was repeated for the rest of the nodes. Among the

visualizations that used only one circle, one sorted the nodes alphabetically around the circle, while another used an external factor such as an input file specifying the order of the creation of the objects, that nodes represent (as in “Social Circles”), in time. Others are placed in a second category according to the method they use (see Table 1).

The hand-made category includes visualizations that were either drawn by hand, or had their nodes placed according to a pre-generated input, such as an input file (visualizations in the geographical map category were excluded from this category unless they were literally hand-drawn on paper).

The spatial calculated category includes visualizations that calculate the coordinates of nodes according to the spatial locations of *related entities* in the visualization. A related entity is an object that is in some way related to a collection of nodes in the network. The related entities could be part of the network, in which case they are a subset of the nodes, or not a part of the network, in which case they are different types of objects either surrounding the network or dispersed within the network. These related entities could represent an attribute of the nodes. For example, in “Making Visible the Invisible”, the 36th visualization, sorted Dewey classification numbers (related entities) appear in two horizontal strips above and below the nodes (sandwiching the nodes) and each node is centered among the Dewey classification numbers it belongs to. Sometimes, the strength of the relation between a node and the related entity is used to affect the final placement of the node. For example, in “Kryptasthesie,” the 62nd visualization, documents are placed closer to the search word that they contain more information about, where search words are the related entities and documents are the nodes. A special case is when the related entities are determined via user interaction. This causes the layout to change every time users select a different set of related entities. In “Non-geographic Mapping”, the 92nd visualization, there is only one related entity, which is selected by the user’s clicking on a node. In this visualization, the nodes represent cities. When a city is clicked, every other city is placed according to its relation to the selected city. Specifically, duration of flight determines the distance to the selected city and the geographical direction determines the angle.

Visualizations that divide the screen space according to a template depending on the node attributes fall into the substrate-based category. There was only one visualization, called “Interactive Activation” that fell into this category. It grouped the nodes, which represent a feature in a connectionist network, according to their type. Nodes representing marital status (single, married, divorced) stay together on a continuous portion of one of the concentric circles. In this way, they are spatially separated from the other nodes. This principle of grouping spatially is also used to place the other types of nodes that represent age group, education level, the gang they belong to, etc.

Time is used to layout the nodes in two time-oriented visualizations. In “Time Graphs,” the 87th visualization, which uses thumbnail images for nodes to represent photographs, the horizontal axis represents time increasing from left to right for an entire year using one day for the smallest unit of time, while the vertical position represents the time of day the photo was taken. The second visualization used time to divide the concentric circular layout into twelve sections, each representing a month, sections being similar to slices of a typical pie chart.

Finally, there were two visualizations that used a random distribution to place the nodes.

3.2 Exploration of precedent data using NVSS 1.0

We have been collaborating with researchers in the Government & Politics department, who are studying one type of court case, regulatory taking cases, over time and across federal courts. A general task is to find the potential role of cases or courts in terms of influencing future cases. To visualize and explore the data, a prototype tool for network visualization that uses the substrate idea, NVSS (Network Visualization by Semantic Substrates), was created.

The entire dataset of regulatory taking decisions contains approximately 2,700 court cases and 15,700 links representing citations across court cases. For every court case, the name, date, citation, court type, court name, and its text are available in the dataset. To create the dataset, a semi-automatic procedure is used. First, a program connects to the Westlaw database issuing about 30 searches and the result of these searches are downloaded automatically along with the metadata found in the result pages. Next, the downloaded data is processed and converted into a MySQL database. For each case, the database also contains information on which search(es) this case appeared in, what other cases it cites in this database and what other cases in this database cite this case. In addition, the database also keeps information on parallel citations, all citations that can be used to refer to a specific case. This database is used by the Cite-It project by our collaborators in the Government Politics Department. This Java based tool, which enables users to mark cases to indicate whether they fall into the regulatory takings case category. The database also keeps track of these markings. Some statistics on court types and courts are as follows:

- There are three major court types (Supreme, Circuit, and District). In total, there are 11 types, one of which is “Other”, which includes the cases that do not fall into the other 10 categories.
- There are 204 courts, all of which are federal courts.
- The date of cases ranges from 1978 to 2005.

To apply the idea of semantic substrates, a subset of this dataset of court cases is used. The subset is generated by including only Supreme Court and Circuit Court cases that are cited at least 45 times, thereby including the most prominent court cases. The subset has a total of 49 court cases (nodes) and 368 citations (links) between 1978 and 2002.

In NVSS, two rectangular regions were created. The region above is larger in size and contains 36 Supreme Court cases ranging from 1978 to 2002, while the region below is smaller in size and contains 13 circuit court cases ranging from 1980 to 1995 (Figure 4). Supreme Court cases are represented with red (dark) nodes and Circuit Court cases are represented with white (light) nodes. They are arranged from left to right in terms of invisible vertical slots, where each slot represents a year. The size of the nodes is proportional to how many times they cite other cases in the original database. The tendency of nodes getting larger towards the right can be explained by the increasing opportunity to cite regulatory takings decisions in the history.

Within a vertical slot, a vertical jittering function is used to spread the items out to reduce link crossing and tunneling under nodes. The jittering function shifts decisions within a vertical slot representing a year either up or down. The shift amount depends on the year and the vertical gap size between decisions within a year. The shift is applied periodically as down 50%, up 75%, down 0%, and up 25% of the gap size for every first, second, third, and fourth vertical slot representing a year.

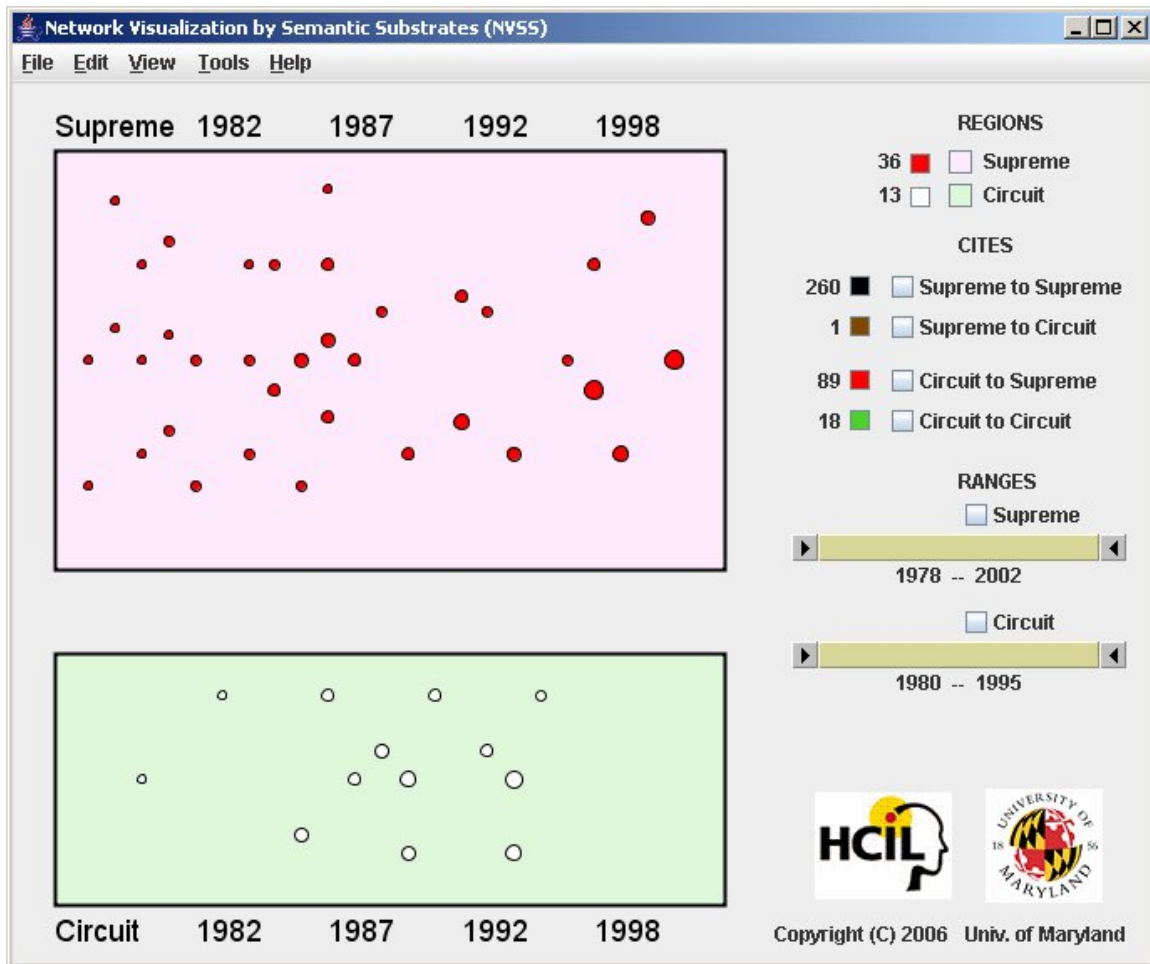


Figure 4 Supreme Court region holds 36 cases from 1978-2002. Circuit Court region holds 13 cases from 1980-1995.

In Figure 4, the right side contains the control panel of the application.

The top part of the control panel shows the number of nodes in each regions. The middle part, with the heading “CITES” has controls to manipulate the visibility of links. Links are divided into types according to the regions their adjacent nodes fall into. Links become visible when their corresponding checkbox is checked. The color and number on the left indicate the color that will be used to draw the links and the number of links, respectively.

Clicking the “Supreme to Circuit” checkbox reveals the single brown link from a Supreme Court to a Circuit Court case. Clicking “Circuit to Circuit” checkbox further reveals the yellow links that are from Circuit Court to Circuit Court cases (Figure 5). This way, users can limit the links on the display to a manageable amount.

In this dataset, this division of the links depend on the types of nodes (i.e. the attribute for “court type”: Supreme or Circuit) contained in regions, which turns out to be the most useful method to divide because the users tend to define the links this way, as well. Although there are only 4 types of links in this example, in general, this number depends on the number of regions in the substrate.

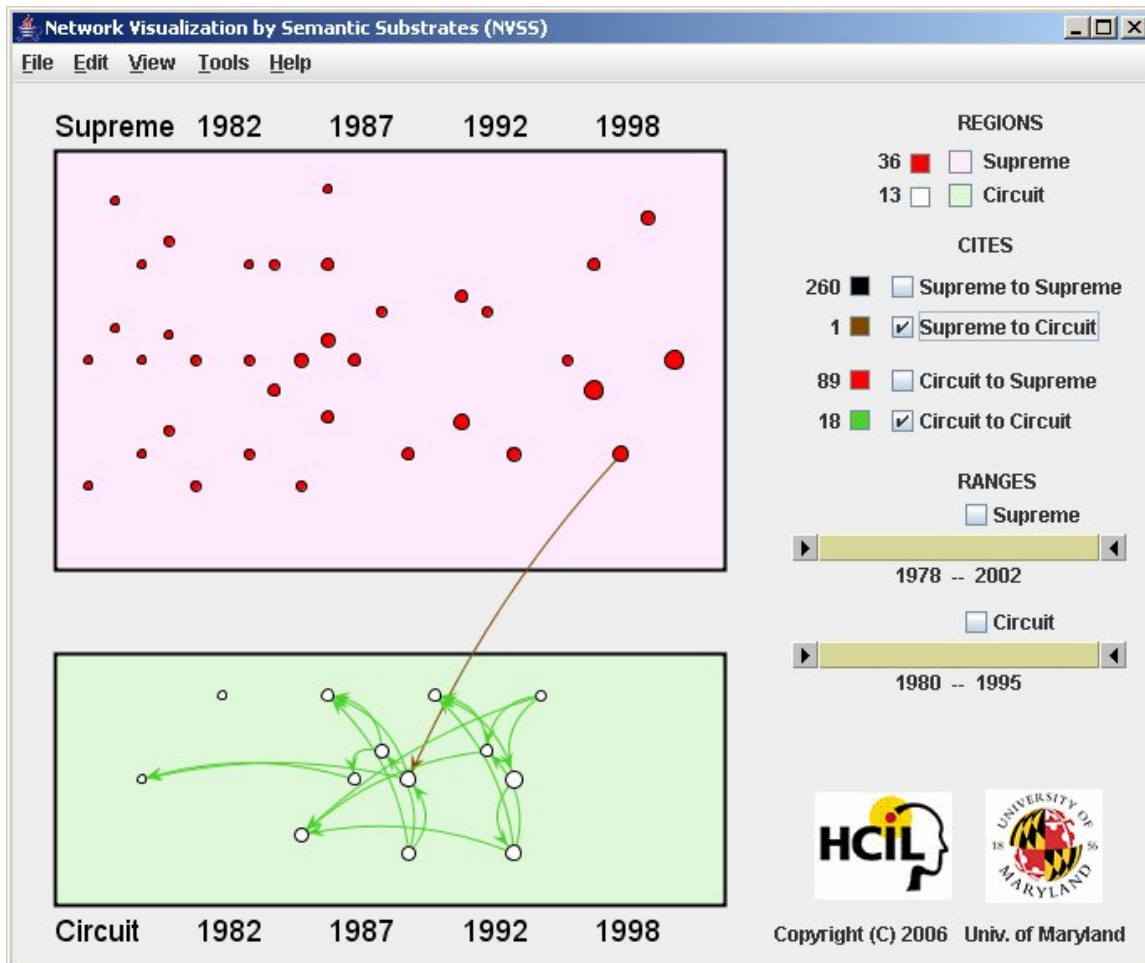


Figure 5 Clicking the “Supreme to Circuit” checkbox reveals the single brown Supreme to Circuit citation, while clicking “Circuit to Circuit” checkbox reveals the 18 green citations within the region where all Circuit Court cases are located.

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 . Listing all checkboxes may not be feasible if the number of regions is large. In that case, another method could be used, such as the following:

- A way to select the checkboxes to be displayed could be provided.
- After determining a number of checkboxes to appear on the control panel, users could dynamically change the meaning of a checkbox by changing what the end points should be. (A further enhancement could allow users to modify the number of visible checkboxes, that is, add new checkboxes or remove existing ones.)

There are controls to restrict links under the “RANGES” heading in the control panel. First, links are restricted to outgoing links from a region by selecting a checkbox corresponding to a region. For example, checking “Supreme Courts” will restrict links to outgoing links from the Supreme Court region. Next, links are further restricted by specifying a time period via the corresponding

double-box dynamic query slider. Restricting the range to 1986-1986 will lead to the links in Figure 6 to be displayed.

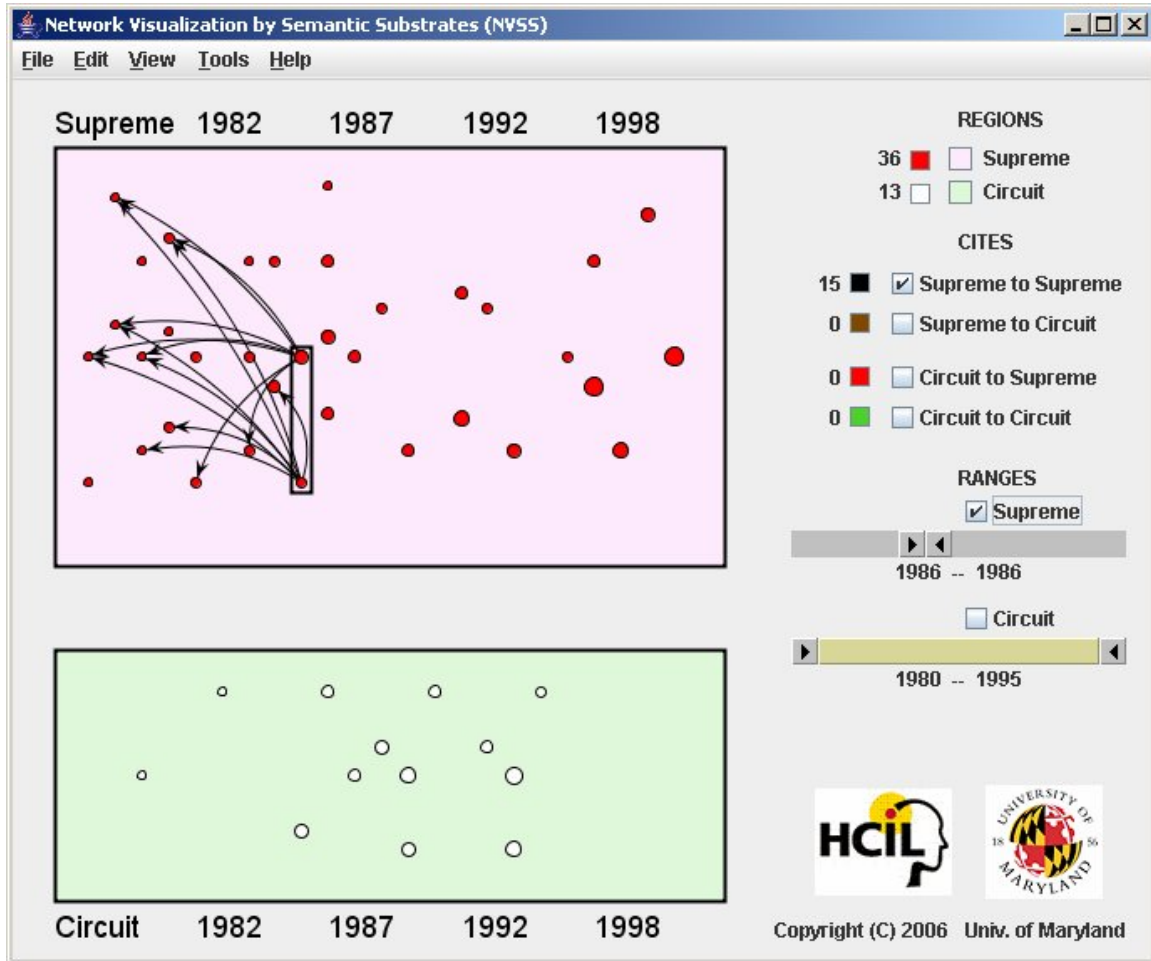


Figure 6 Limiting the selected cases to the two in 1986 generates a comprehensible display of 15 citations, with five cases being cited twice and five cases being cited once.

The range selection also works across regions. By selecting the 1991 to 1993 Circuit Court cases using the Circuit Courts slider, users can see the two citations to Circuit Court cases and the 18 to Supreme Court cases (Figure 7).

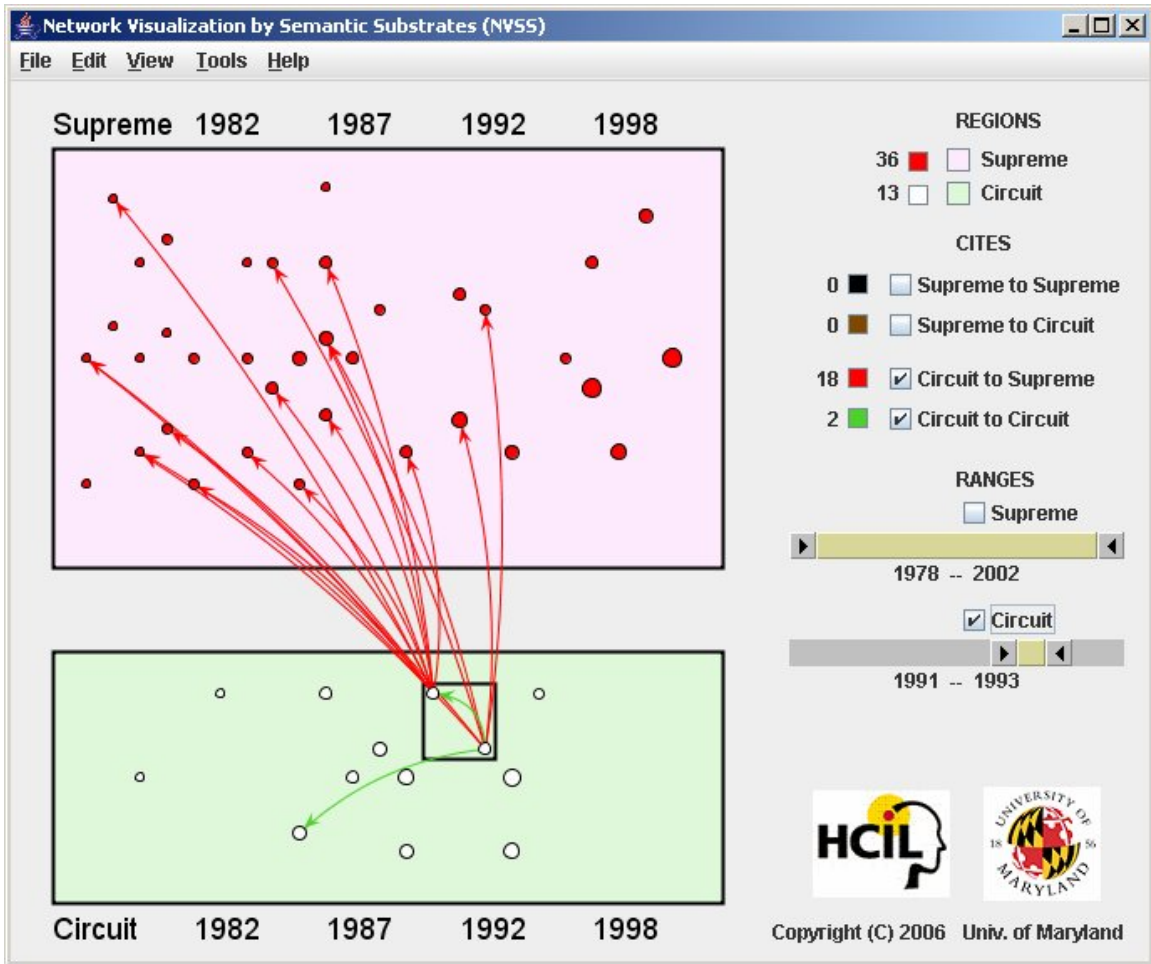


Figure 7 Limiting the selected Circuit Court cases to the two in 1991-1993 generates a comprehensible display of the 18 red Supreme Court citations and the 2 green Circuit Court citations.

While citations in Figure 7 are still comprehensible, there is room for improvement to avoid the overlapping of links. Links tend to overlap more when the angles between links are small. This is better illustrated in Figure 8. One way is to improve link routing. The current strategy NVSS 1.0 uses is the quad curve implementation in JUNG. There might be other alternatives to improve this situation, such as the use of different substrates or node placement methods that minimize if not eliminate the number of small angles between links.

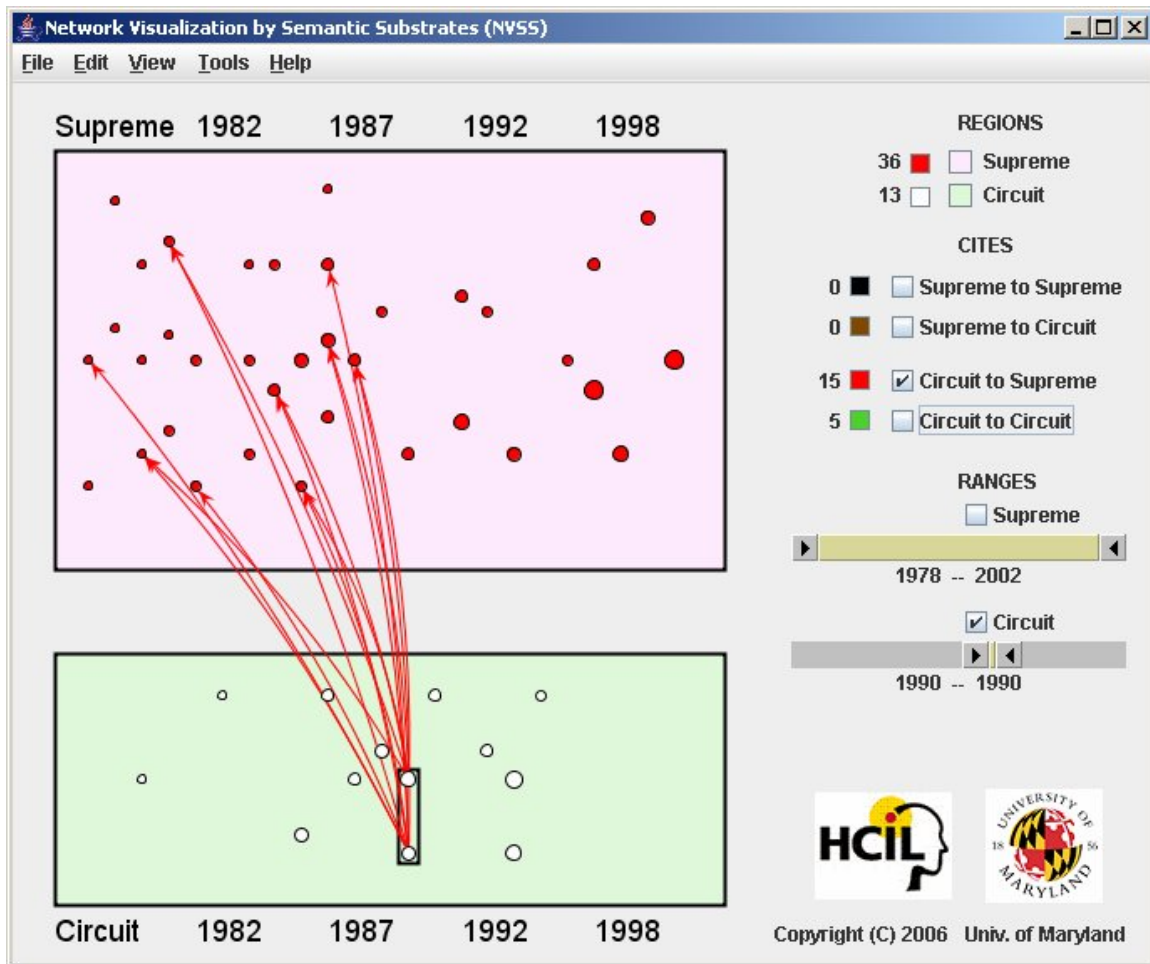


Figure 8 Limiting the selected Circuit Court cases to the two in 1990 generates overlapped links to Supreme Court cases, suggesting room for improvement in this aspect.

Being able to restrict links is important to increase the comprehensibility of the network users are looking at especially when the number of links is so large that overlaps and crossings are unavoidable. Checking all CITES checkboxes reveals all links as in Figure 9. Although the brown link from a Supreme Court case to a Circuit Court case and the green links within the Circuit Court region are mostly visible, there are too many black links in the Supreme Court region and red links from Circuit Court region to the Supreme Court region. Users get an understanding of the complexity of links even though not all the individual links are distinguishable. The numbers provided on the control panel further supports this understanding. Users can conclude that there are far more links within the Supreme Court than any other type of link, and the sense that Circuit to Supreme citations tend to increase over time, with the understanding that later cases still cite some of the earliest cases (that they are not overlooked or forgotten, and still being cited). A better understanding can be achieved by using the dynamic query sliders, restricting to a small time period such as 2 or 4 years and sweeping from left to right. When this type of interaction is applied to the Supreme court region, it also becomes apparent that later Supreme Court cases still are citing some of the earliest cases (see Figure 10, Figure 11, and Figure 12).

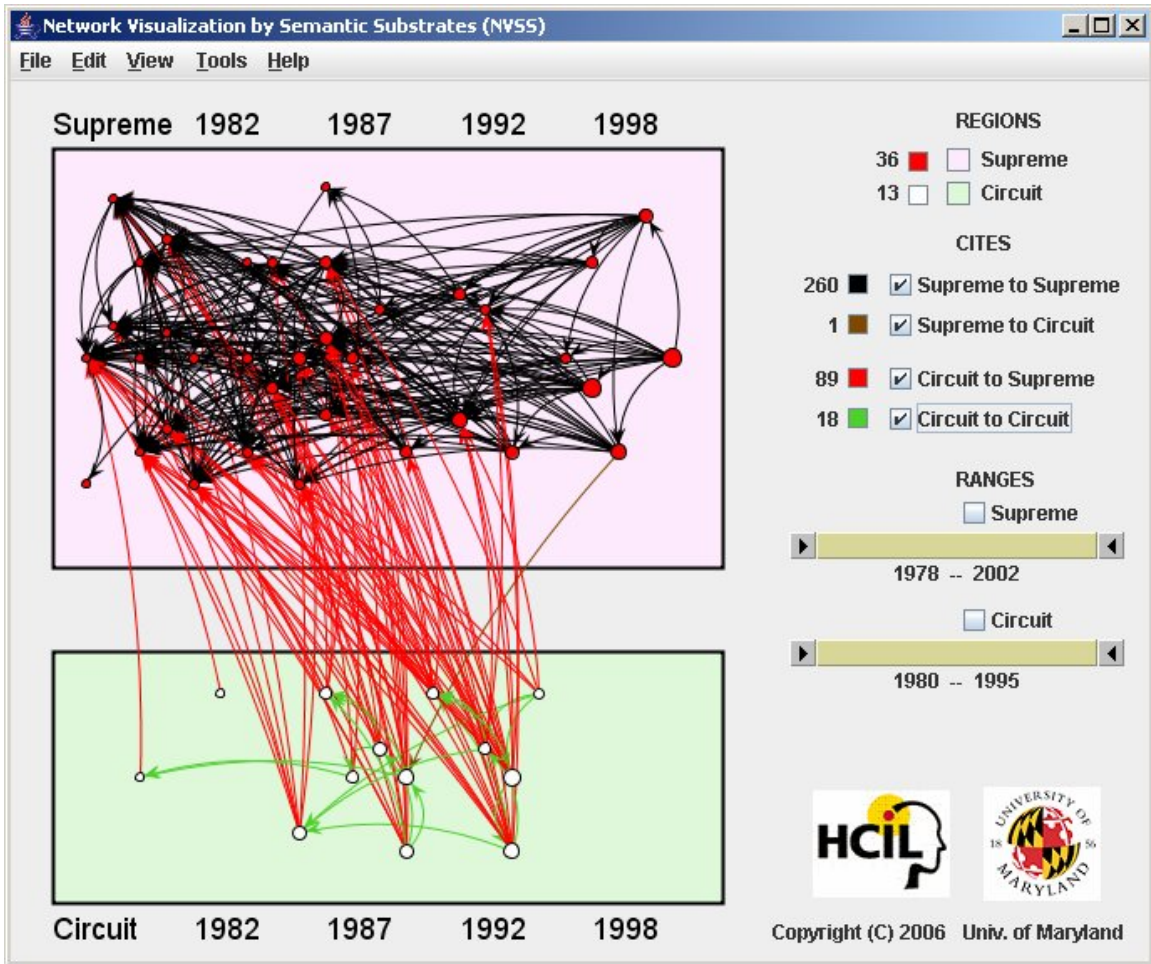


Figure 9 With all CITES boxes checked the 260 black Supreme to Supreme citations and the 89 red Circuit to Supreme citations are impossible to follow. However, the single brown Supreme to Circuit citation is apparent and the 18 green Circuit to Circuit citations are mostly visible.

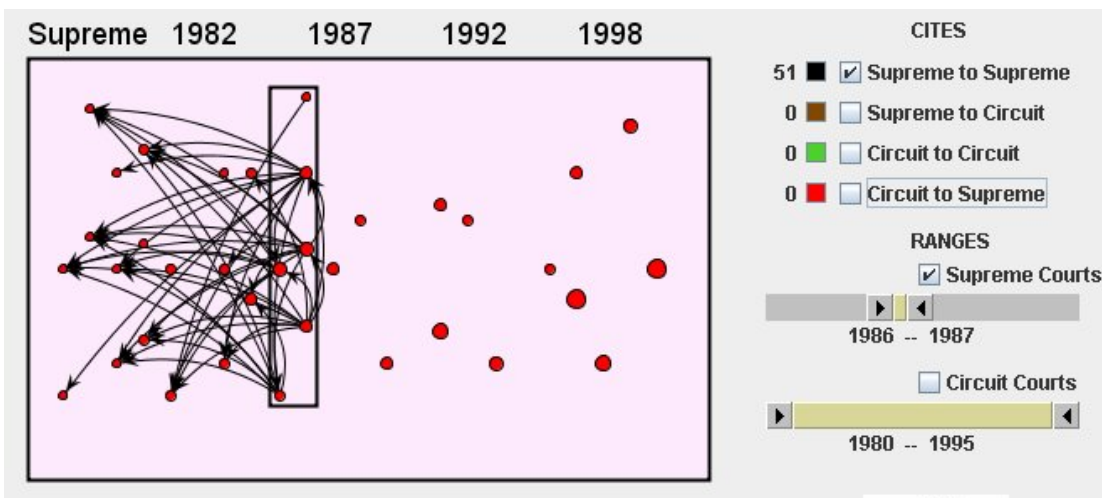


Figure 10 Six Supreme Court cases between 1986 and 1987 heavily cite early decisions.

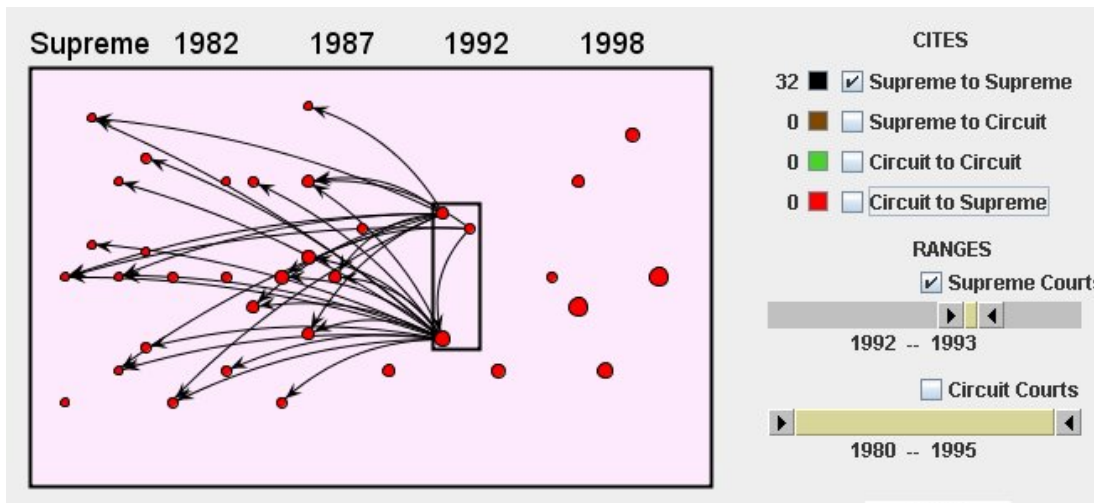


Figure 11 Three Supreme Court cases between 1992 and 1993 are still citing some of the early cases.

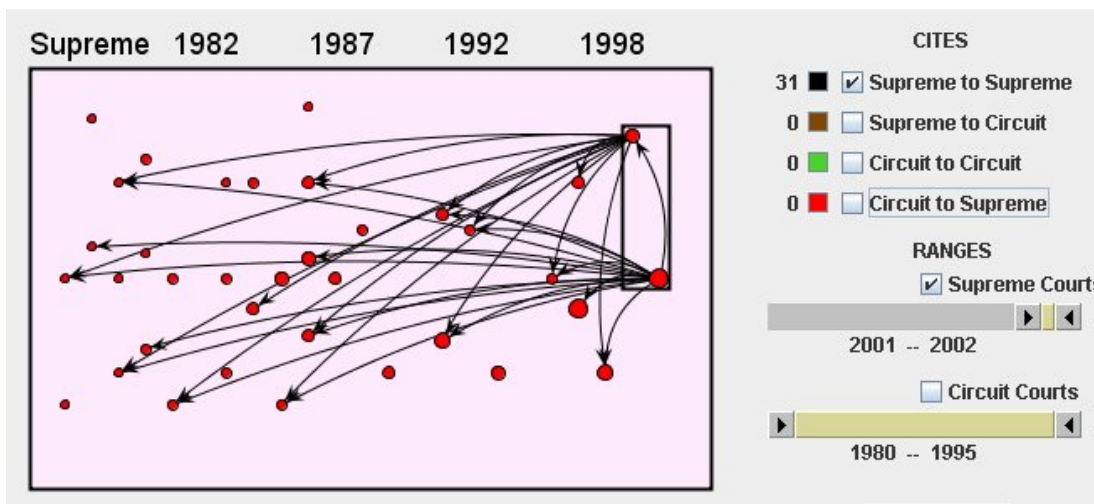


Figure 12 Two Supreme Court cases between 2001 and 2002 are still citing some of the early cases.

Even without any restriction on the links, using a substrate approach still compares to be better in terms of comprehension to without using substrates. Users instantly get a sense of:

- the groups by looking into regions (Supreme and Circuit)
- how many nodes each region contains (even without looking at the numbers in the control panel)
- the number of links within a region (Supreme contains a lot while Circuit contains few)
- the number of links across regions (there is only one link from Supreme to Circuit, but there are many more from Circuit to Supreme)

The arrangement of links in the presence of regions helps to distinguish the different types of links (in this example, type is defined as: from which region to which region they go to). This is better understood when looking at one alternative that uses the Fruchterman-Reingold algorithm implemented in JUNG rather than using substrates (Figure 13). Although one can still get a sense of the different types of links due to the different colors in Figure 13, the substrate approach provides a better sense due to the organization effect of the substrate on the links. For example, since the Circuit Court region is below Supreme Court region, links from Circuit to Supreme

must be always in the upwards direction crossing the empty space and this empty space cannot be occupied by any other type of link.

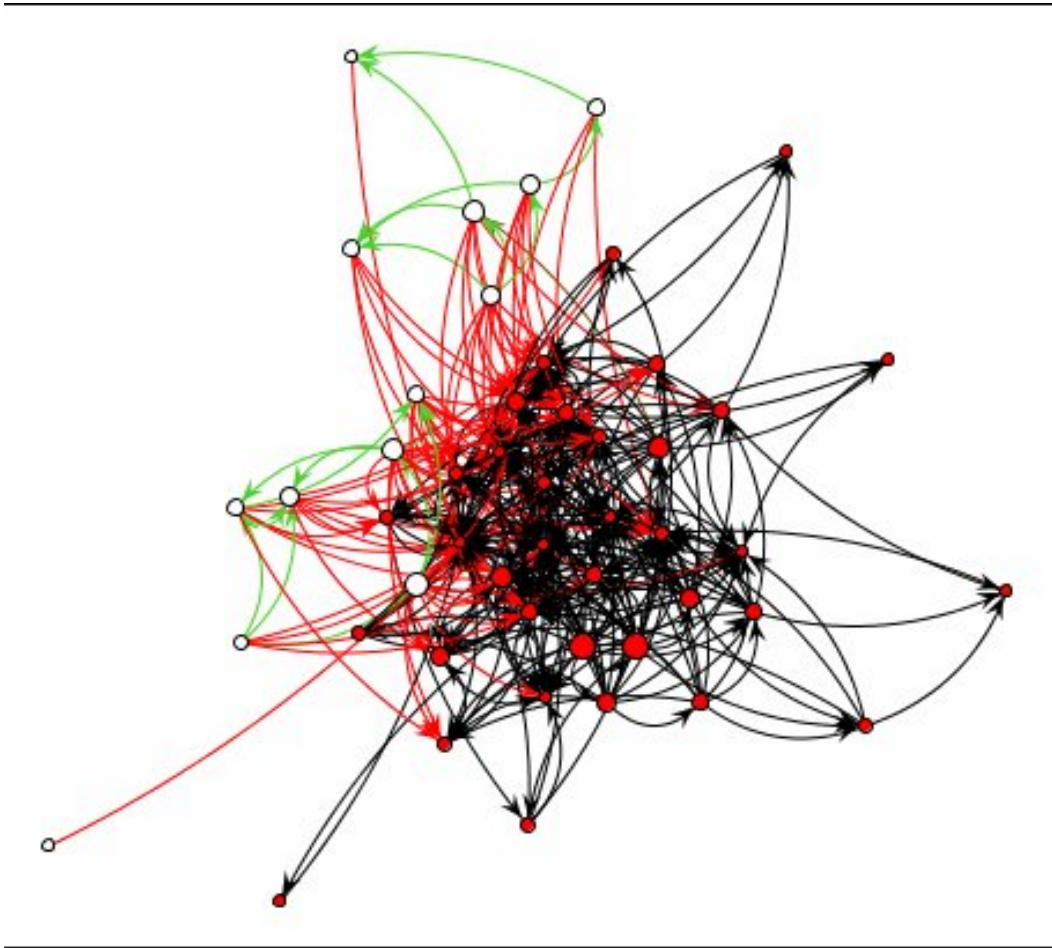


Figure 13 Using Jung's Fruchterman-Reingold layout to place the 49 cases with all 368 citations visible makes it impossible to follow citations from source to destination or to see temporal patterns.

Having more than two regions reveals more information (Figure 14). In this court case example, a natural choice for the third region is to include the District Court cases. In Figure 14, 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."

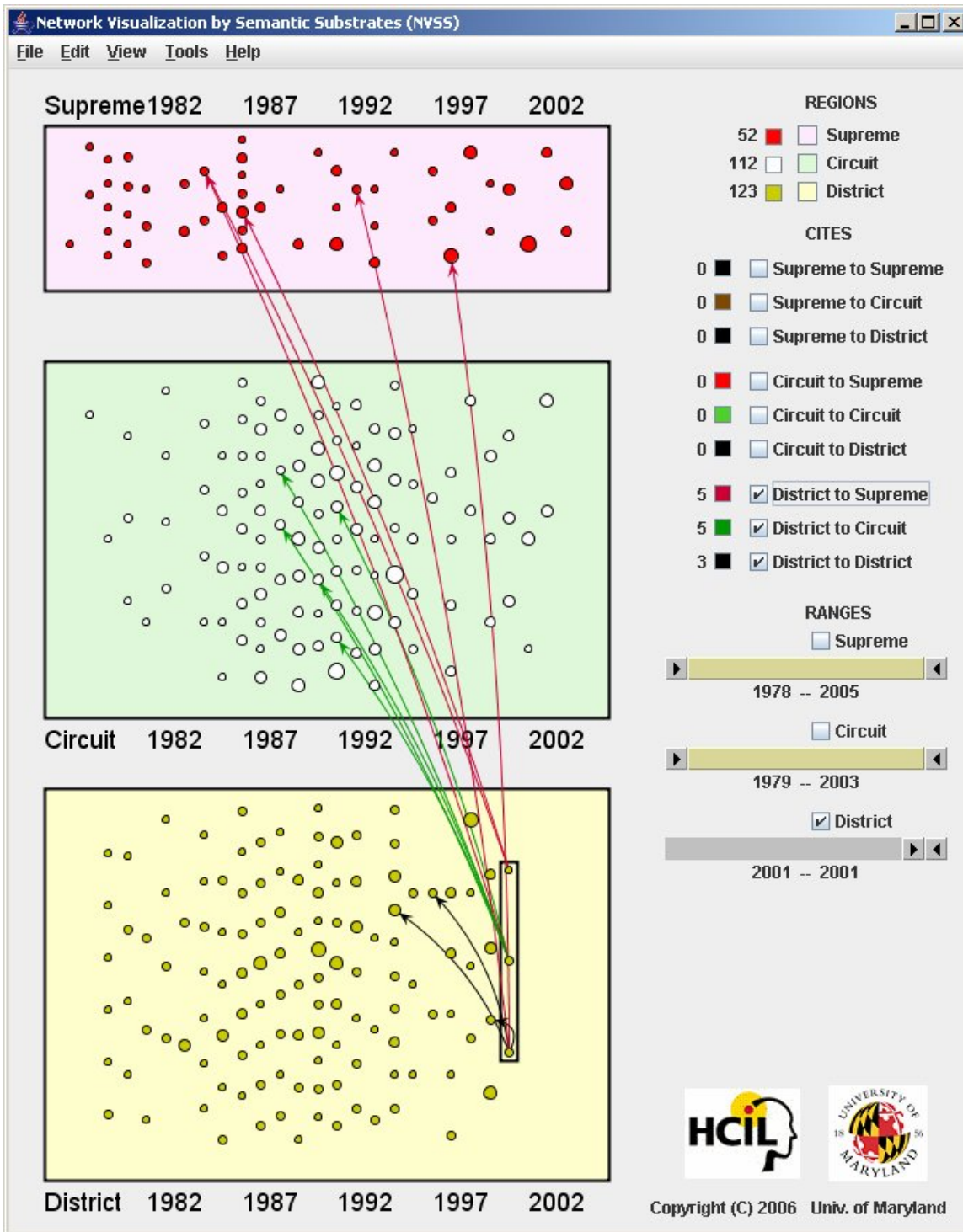


Figure 14 Having District Court cases in a third region, 287 nodes and 2032 links.

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 they were not very clear about 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 this number is called 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 (Figure 15). 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 Figure 15. 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 9th and the Federal circuit were active and important, which was indicated by incoming citations.

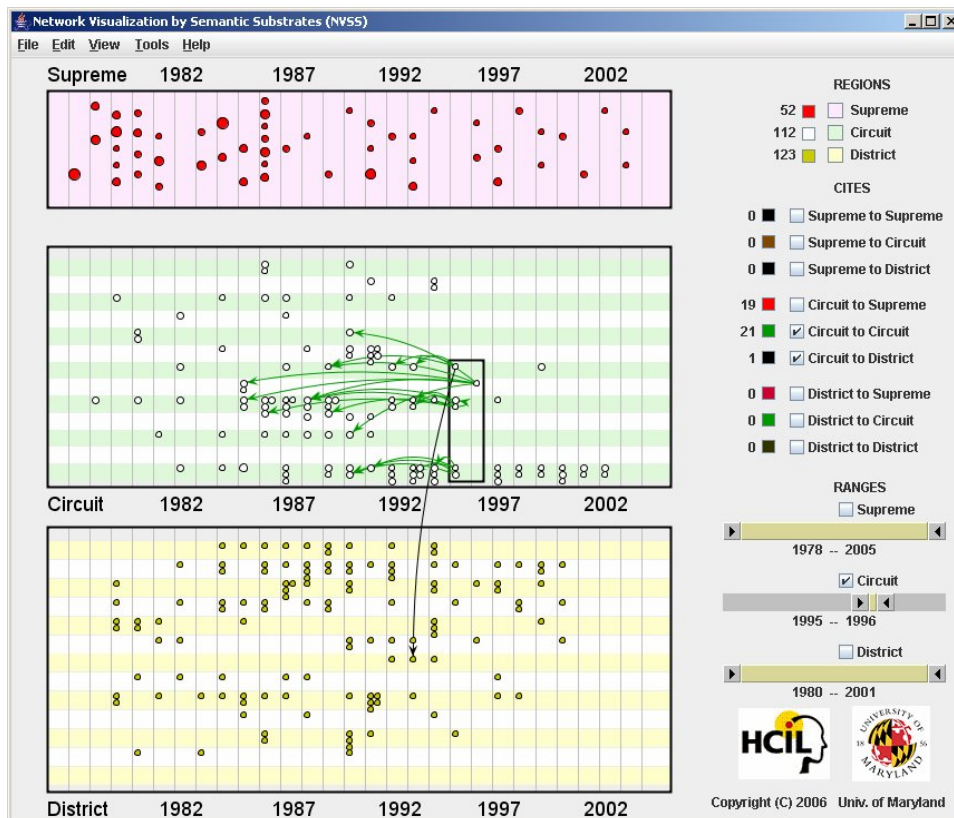


Figure 15 The layout for Circuit Court cases is now organized by the 13 Circuits and the link pattern shows the strong likelihood that cases with reference precedents within the same Circuit.

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 9th circuit were active, they differed in terms of incoming citations from other circuit courts. While the 9th 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 Figure 16. 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. Figure 16 shows a still larger data set with 1,122 nodes and 7,645 links at a 1280x1024 resolution.

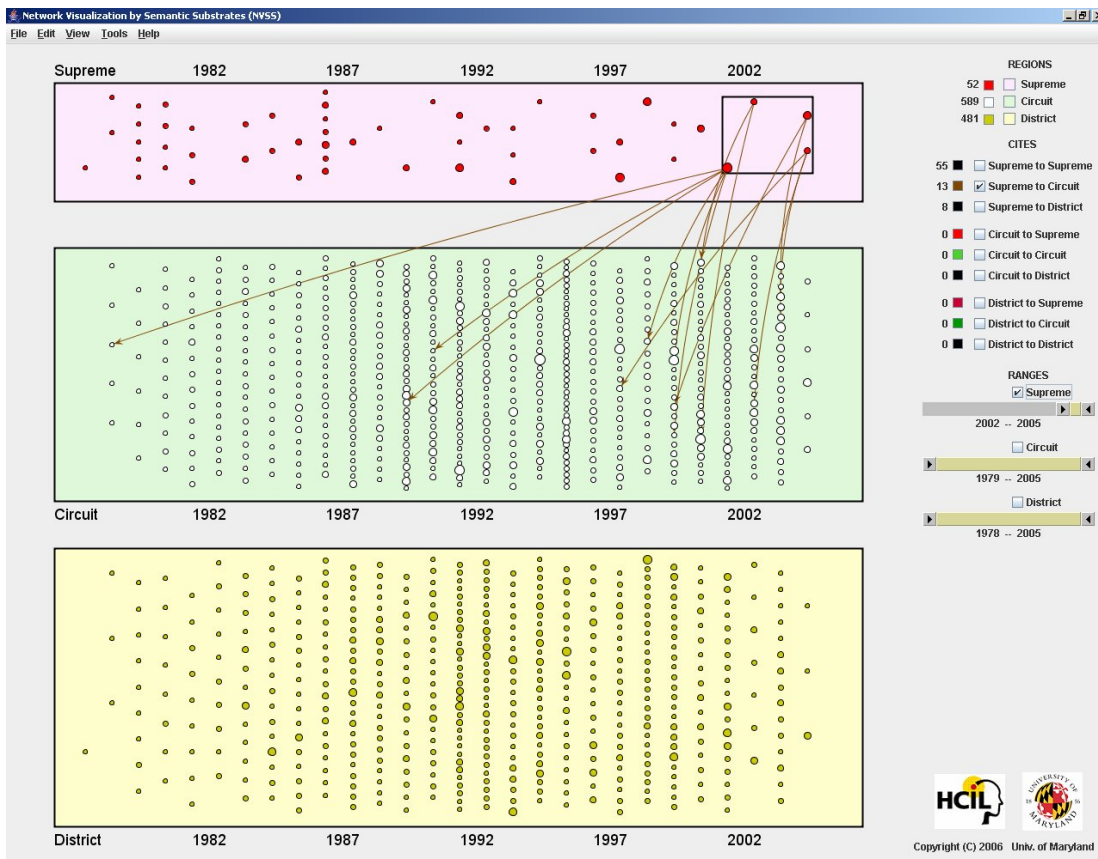


Figure 16 Displaying 1,122 nodes and 7,645 links at a 1280x1024 resolution.

3.3 Experience and insights regarding user tasks

So far, the research on semantic substrates has shaped according to the communication with users from the court precedent domain. Section 3.3.1 summarizes the insights for user tasks from this communication. Section 3.3.2 briefly describes possibilities for a second domain.

3.3.1 Experience from court precedent users

According to our communication, court precedent users are looking for influence over the history of cases. The motivation is to spot patterns or relationships that look causal, which are worth further investigation with other tools. In this respect, the role of network visualization in their setting is to accelerate the process of hypothesis generation, while a secondary role is to enhance a general understanding of the network. The following tasks are either cited by our collaborators or reflect our understanding of what is useful.

1. *Get a sense of the distribution of cases among different types of courts.*

This helps for the general understanding of the dataset. Seeing how many nodes are in each group (visualized as a region) conveys the characteristics of the dataset. In a dataset that is a subset of highly cited court cases, users see that most of the cases are Supreme Court cases.

While users benefit from the distribution among general groups, they later would like to see the distribution among more specific groups. For example, they may want to see the distribution of Circuit Court cases among the 13 circuit courts. In the case that where the District courts are shown, they would like to see subgroups of district courts according to their jurisdiction.

2. *Get a sense of the connectivity between groups.*

Users are looking for the number of links across groups. The connectivity information has a value in terms of the specificity of groups. Connectivity between general groups has a different value from the connectivity between specific groups. However, both seem to be useful. Users may benefit from the general and then seek for the specific type. For example, the connectivity between Supreme Court cases and Circuit Court cases is general, while the connectivity between Circuit Court 5 and Circuit Court 9 is specific. Users are also interested in the direction of the connectivity, especially for the part of the network that they have not acquired previous knowledge from somewhere else. The direction conveys prestige. The cited cases are believed to be important and influential.

Sometimes users ask less specific questions such as “Which district courts cite out of jurisdiction?” or more complicated, such as “Which circuit courts contain cases that are cited by cases outside of their jurisdiction?” This is another indicator for prestige. Another relevant question users look to answer is “Are the cases influential by themselves, or is it more that the court produces influential cases?”, which in turn may mean that there are factors that make this court influential, which is manifested by its cases.

3. *Factoring in time.*

While users first seem to seek for patterns or relationships without time, they later want to add “time” into the equation. This behavior seems to show itself in various types of tasks. To illustrate with the “connectivity exploration task”, first they may look for the connectivity between Supreme Court cases and Circuit Court cases. Later, they may want to see the same relationship over time, that is, “connectivity between Supreme Court cases and Circuit Court cases for each year from 1978 to 2005”.

Factoring in time is likely to increase the number of possible combination. The tendency of users is to cluster them. For example, they may want to see “the connectivity between Supreme Court and Circuit Court cases from 1978 to 2005 clustered in 3 year groups”.

Their reason seems to be to reduce the number of possibilities by decreasing the specificity. In other words, they may want to switch to a more general level when the task becomes overwhelming for better understanding and faster navigation.

4. *Chains of influence.*

Users are interested how one case evolves over time and how other cases are influenced by it. This partly can be understood by the links of one past case over time. Users also try the reverse, which is “Which set of cases influenced this case?” This creates a reaction to want to see the citation tree that originates from the recent case. In the former case, the length of the chain may convey “the longevity of a case”.

Sometimes, they are also interested to see this information in proportion to the whole with the aim to generate an informal metric for its importance. This is partially for comparison purposes also, which is motivated by the need to find “the most important”, and “next important” cases in history.

5. *Citation patterns determined by node properties.*

Users are interested in seeing the citation patterns given a case. However, the citation pattern itself may not be enough. The links seem to enhance understanding when they are qualified by a property of the nodes they are connected to. For example, when users inquire the citation pattern for a case, they may ask, “Is this case cited by a large number of small cases, or a small number of large cases?” In this question, “small” and “large” is a property of the nodes connected on the other side of each link, namely the in-degree.

Users would like to see this property of all connected nodes at once, such as via the size of those nodes on the display. An exact measure of each is usually not critical, as the overall understanding seem to be more important.

6. *Generate-store-use cycle.*

At times, users seem to process and make sense of data in an incremental way. They look for some information, find it, interpret it (usually an abstraction), which is the *generate* phase. Then, they seem to be willing to store the result of this information to be used as an input for other similar kinds of exploration. For example, first they may be looking for cases that were cited the most between 1978 and 1985. Then, they would be willing to store or mark these cases to be used for another exploration, such as finding the cases in 2005 that eventually cite (directly or indirectly cite) from these cases. Then, they may want to store these cases, do the same for 2004 and then find which of the 2005 cases previously marked cite the cases marked in 2004. This process may continue stepwise, where one task depends on the result of another task.

3.3.2 Other Domains

Currently, there seem two possibilities for a second domain: (1) Biological food web data, and (2) protein interaction networks.

The biological food web shows which species is the predator of another species (where this other species is called prey). Users of this dataset are interested to see the species that are solely predators (have no predators) and the species that are solely preys (have no preys themselves). There are networks, the links in which users estimate. Another task is to compare the generated networks with the real networks. A third task is to discover new links in the network. A related task is to find which parts of the dataset (characterized by an attribute such as phylum) are easier to find links in.

The protein interaction network domain is a very recent possibility under exploration. There are two possible users. One of them is a graduate student doing PhD research (in this Computer Science Department) involving analysis in protein interaction networks. Another possibility is to collaborate with a prospective researcher in TIGR (The Institute for Genomic Research) at Rockville, MD. The plan is to get in touch with him to explore this possibility as soon as the prospective researcher's full time position starts at TIGR.

4. Planned Work

The following sections detail the next steps in this work in terms of user tasks and details the user interface design considerations. Each section describes a major task. After or during each phase, there will be communication with our collaborators to get their input, which will contribute to the case study. Other datasets could be used to visualize by NVSS to see whether the functionalities satisfy other users and to get their input, which may reflect other user tasks. By performing more than one case study, the common tasks and different tasks across domains can be determined. Knowing which tasks are common, and which tasks are domain-specific, certain features may be determined more valuable and NVSS can be designed (or adapted) accordingly.

4.1 Substrate generation

The main approach in this proposal is to use substrates containing regions, in which nodes are placed according to their attribute values. Nodes are placed into a corresponding region according to one attribute, and their placement inside the region is further defined as a function of another attribute (or more than one attribute). In section 3.2, the court type attribute determined in which region a node is to be placed and the date attribute determined the placement of a node within its region.

Given a dataset and its attributes, it would be ideal to have a semantic substrate generated automatically for the user. However, there are many factors contributing to the design of a good substrate:

- *The complexity of the nodes and links in each region.* The more complex the subset of the network within a region, the more challenging it becomes for users to understand this region. One way to alleviate the complexity within a region is to allocate larger size to it. However, the sizes of the other regions may need to be considered, as well.
- *The importance of a region.* The more important a region is, the more of the limited resources it deserves. One of those resources might be screen space, another might screen location (some locations are better than others, such as the middle, where it has the largest opportunity to connect with other regions).
- *The node placement method used within a region.* There may be many node attributes to choose from when determining which one(s) should be used to place the nodes. One attribute can be good to use for determining the x-coordinate of nodes (e.g. time), another might be better suited for the y-coordinate (e.g. circuit in precedent data, category in a general sense). Other types of arrangements (simpler or more complicated) are possible.

It would be ideal to have an algorithm that generates the optimal substrate for user understanding that can take into account the factors above and perhaps others. However, this is difficult to automate. It may be hard to come up with a clear method to measure the importance factor or it may depend on changing user opinions and their tasks. For this reason, the suggested approach is to let users define it, however, also provide good controls to minimize the effort.

There are some simple properties of a substrate to define, such as:

- the location of each region
- the size (width & height) of each region
- the attribute value of each region (i.e. the attribute value of nodes that this region will contain)
- the title and background color of each region

The more complicated properties are:

- the node placement method used in each region
- the complexity measure that determines the size of each region

There are several user interface issues regarding having users generate substrates, some of which might be challenging. A number of those challenges can be expressed in terms of questions:

- How does a user know what attributes there are to choose from? How does the user interface represent their types? (A set of types are: categorical, ordinal, numerical (integer, real, etc.), and boolean.)
- What criteria will be available for users to define what each region will contain? How will they define them? (How will they define them given a region? And, will there be quick methods to define what a number of regions will contain at once?)
- How will the size and location of each region be defined? Will there be some automatic methods, too?
- What factors do users need to know when determining the location of regions? (Static locations for each region are assumed. Each region will expand proportionally when the application is resized. There are more complex layout algorithms such as the ones provided in Java libraries. However, the scope is limited to static regions for the moment.)
- Will there be a preview of the network laid out according to the generated substrate? What will this preview contain? When will this preview be updated? Is interactive preview tractable and feasible? Will it work when the network gets large?
- Once a user spends effort to generate a substrate, will it be saved? In what form, using what types of data structures, and where will it be saved? How will it be reused? Can it be reused across networks? If so, what are the criteria of compatibility?
- Can the application assist with user's strategical choices? What information needs to be gathered and presented to the user? How can this information be presented?

Some of these tasks may be quite involved and may generate further questions. The initial design choices for a substrate editor need to be carefully thought through and the context of the application needs to be considered because the two are likely to influence each other.

4.2 LinkViz Methods: An alternative approach to drawing links

With the introduction of human interpretable places of nodes, alternatives to drawing links emerge that are perhaps more effective due to the placement strategy of nodes. Furthermore, due to these placement methods, drawing links decreases in its effectiveness when there are many of them. Since the placement of nodes conforms to some other rules than aesthetic features of the graph, it is likely that networks laid out this way will conform less to the graph drawing aesthetics. This is expected to lead to more link crossings, links tunneling under nodes, and links with very small angle in between them, etc., which make hard to follow the links and sometimes see the nodes. As a result, thinking of other alternatives becomes promising.

Among the methods to show links, perhaps drawing them provides the most information. When links are drawn in a directed network, users can see the source, the target, and determine which node is connected to another node. Drawing all links theoretically would be ideal to perform all tasks related to links. However, in practice, space becomes quickly depleted and links become incomprehensible due to “overdrawing” (drawing more than is reasonable).

It seems there is no alternative method that will convey the same amount of information as drawing the links. However, this is not a sufficient reason to disregard alternative methods. Users have various tasks that they accomplish via interpreting drawn links. However, they are probably not using all the information available to them by the drawing. They may concentrate on certain aspects, which are sufficient for the task they accomplish.

Defining a taxonomy of tasks related to links, *link tasks*, and investigating the needs for each of those and creating a list of methods, *linkviz methods* (interactions, visualization features, etc.), satisfying those will have the same effect as drawing the links given a task. Furthermore, since each linkviz method is a more specific solution, it is likely that it will perform even better than drawing the links. Therefore, the overall value of this network visualization will be higher because:

- nodes are at interpretable places (more effective than other placements)
- link tasks can be performed as well as or better than the most aesthetically pleasing layout

This is possible whenever alternative methods can be found that can take place of drawing links. When there is none, “links can be drawn.” Since only some of the instances will perform better when the links are drawn (compared to other visualizations), the characteristics of such instances could be reported. In short, this approach may not solve all problems; however, it has potential in contributing better solutions for many existing problems.

An initial (incomplete) taxonomy for link tasks (assuming directed networks) follows. For each link task, a linkviz method is provided and illustrated via an example figure, where the left of the figure uses the conventional way, i.e. drawing links, and the right of the figure uses the linkviz method.

Link Task 1: Nodes linked from a group. Select a group of nodes and see the nodes that this group refers to.

Linkviz Method 1: Color by type. Define three colors: `color_src`, `color_tgt`, and `color_src_tgt`. Color the source nodes, target nodes, and nodes that are both source and target by these colors, respectively. Since this solution doesn’t use size coding, that channel is available for further information.

In Figure 17, the source nodes are enclosed in a box. `color_src`, `color_tgt`, and `color_src_tgt` are chosen as red, yellow, and orange, respectively. While link crossings and links that overlap nodes make it hard to understand which nodes are referred to, it is much easier to perceive this with the linkviz method. The author claims that this linkviz method is superior to even improved link routing and links using a branching metaphor due to less ink used to represent the same information (Tufte 1983). Also, improved link routing or using a branching metaphor when drawing links (Phan 2005) may provide good solutions for small datasets or sparse links, but when the data becomes large and links become dense, it will degrade in its effectiveness, while the linkviz method 1 will preserve its effectiveness.

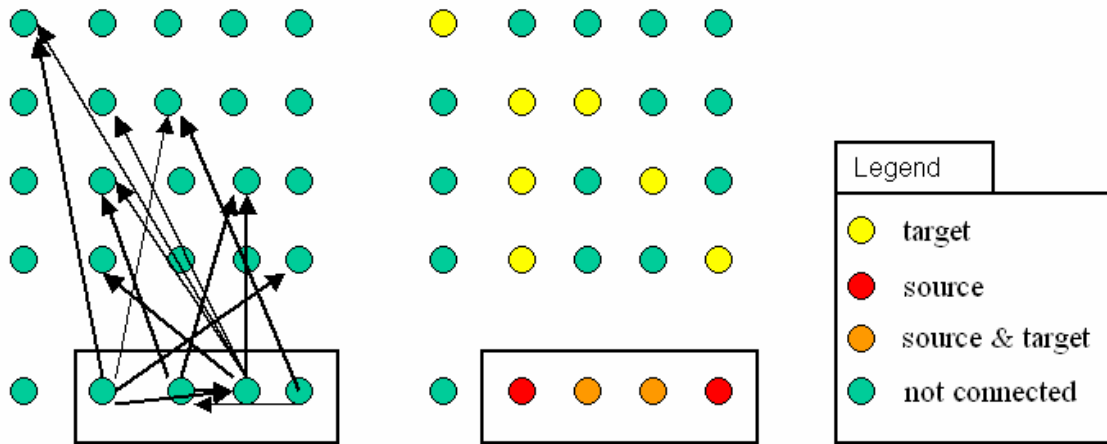


Figure 17 An illustration of linkviz method 1, where source nodes are red, target nodes are yellow, and nodes that are both source and target are orange.

Link Task 2: Distance N from node. Select a node and see the paths originating from this node.

Linkviz Method 2: Color by type. Define as many colors as the number of separation available in the graph. Assuming it is n , define $color_1, color_2, \dots, color_n$. Color the immediately connected nodes to the source node with $color_1$, then color the nodes that are connected to the 1st level nodes by $color_2$, and so on. (For large n , use other visual properties of nodes, such as border thickness, in addition to the color of nodes.)

In Figure 18, $color_1, color_2,$ and $color_3$ are chosen as yellow, orange, and magenta, respectively. It is much clear to see the pattern of levels visually, where yellow nodes (first level) form partial vertical bands, orange nodes are farther and tend to be diagonal, and magenta nodes (level 3) are farther than level 1 and 2 and form a triangle (or a symmetrical V shape to the left). These patterns of levels are likely to make sense especially when the node locations are interpretable.

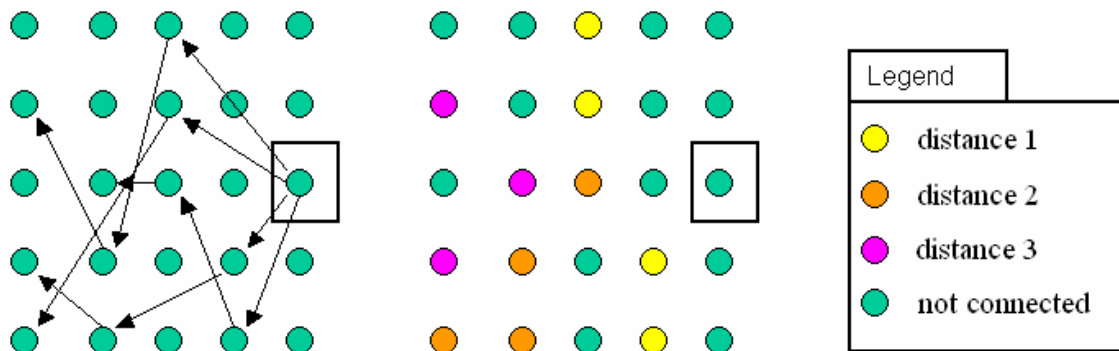


Figure 18 An illustration of linkviz method 2, where level 1, level 2, and level 3 are indicated via yellow, orange, and magenta, respectively. It is much clearer to see the pattern/distribution of the levels, which is likely to make even more sense when node locations are interpretable.

Link Task 3: Nodes linked N times from a group. Select a group of nodes and see how many times other nodes are referred by a node in this group.

Linkviz Method 3: Color by type, size by N. In addition to linkviz method 1, size the target nodes by how many times they are referred by the selected group.

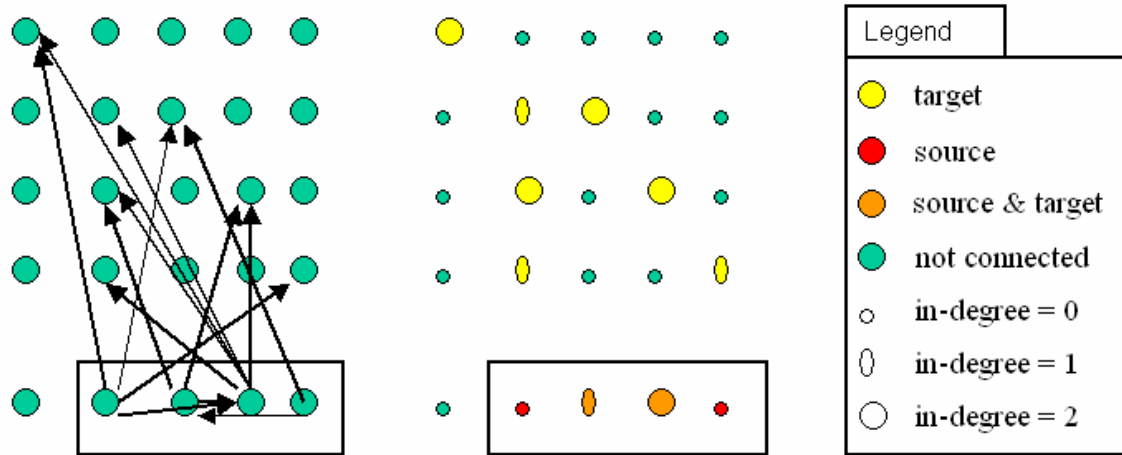


Figure 19 An illustration of linkviz method 3, where source nodes are red, target nodes are yellow, and nodes that are both source and target are orange. In addition, nodes are sized by in-degree due to incoming links from the selected nodes.

Link Task 4: Nodes linked by nodes 1-N: Select a small number of nodes (n , where $n \leq 6$) and see the referred nodes by this group in terms of by which nodes they are referred to.

Linkviz Method 4: Color by multi-type 1-N: Represent each of the source nodes by a color. Let those colors be color₁, color₂, ..., color_n. Color each target node by the colors of source nodes that they are referred to. Use a pie-chart metaphor when coloring. For $n > 6$, binning of the selected group might be a solution to make $n \leq 6$.

In Figure 20, three nodes are selected (designated as source nodes). It is hard to see any pattern when drawing the links. On the other hand, the display is much clearer with the linkviz method 4. One can even see patterns and summarize the behavior. For instance, the target nodes referred by the magenta node form an X shape, while the orange nodes form an almost complete elongated L and the yellow ones form two groups of an L shape that fell to the left side. It is very likely that being able to see the distributions clearly and the ability to notice patterns will reveal strong meaningful information when node locations are interpretable.

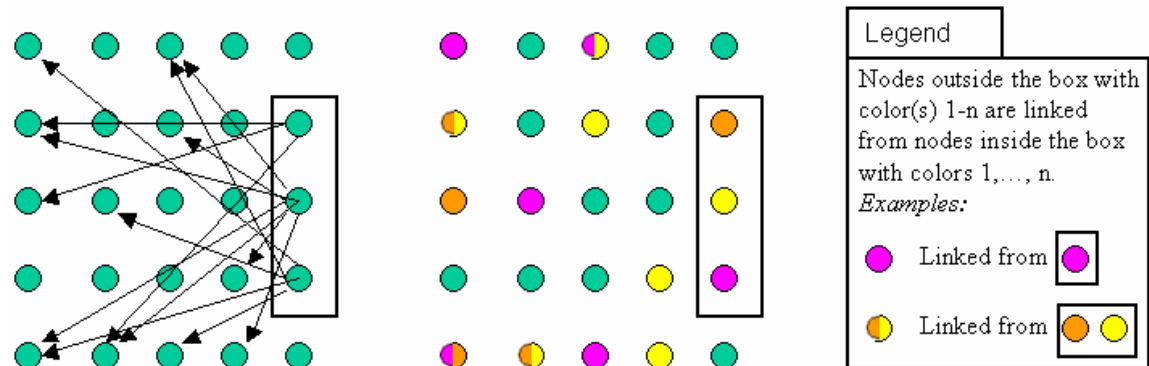


Figure 20 An illustration of linkviz method 4, where source nodes are orange, yellow, and magenta, and target nodes are colored according to the colors of the incident source nodes.

The challenges of this approach include designing the structure that allows such interaction and defining the user interaction. Once they are available, users should be able to quickly perform the task they want to accomplish.

It is likely that there will be many link tasks (more than 20) and possibly more than one linkviz method for each link task. Currently, however, the prediction is that usually there will be a dominating linkviz method that will be more effective than others given a link task. In addition, each linkviz method has parameters, such as choice of color, shape and possibly drawing style (such as whether to draw concentric circles or to use a pie-chart metaphor for multi-colored nodes). Also, the collection of linkviz methods may depend on the features of dataset under exploration (type and/or size) and the collection of linkviz methods may vary from dataset to dataset. Since link tasks are many, there could be choices for linkviz methods, and parameters need to be specified for each linkviz method, a mechanism will be needed to organize this functionality. This mechanism will be one major contribution of this proposal. The plan is to first develop a prototype mechanism and a collection of linkviz methods with one dataset and then apply it to another dataset. As it is applied to another dataset, the mechanism will need to be enhanced and this will reveal general characteristics of the mechanism. These are to be reported and illustrated with examples in the final dissertation. In addition, the implementation of the mechanism is to be provided as software.

4.3 Scalability

There are various types of scalability in a network visualization tool. Being able to handle larger networks is one type of scalability, while being able to increase the number of regions and keep them usable is another type of scalability. The scalability issues are categorized in the following subsections.

4.3.1 Node and link scalability

As the number of nodes and links grow, various challenges arise. First, it becomes harder to see and understand nodes and links. Next, screen space begins to be insufficient. At very large scales, load time, initial drawing, and updates following interactive operations become bottlenecks. Some of those bottlenecks can be alleviated, such as drawing only the most important nodes in a limited amount of time. In fact, for more than 100,000 elements (nodes and links), these are the problems encountered in NicheWorks (Wills 1999), H3 (Munzner 1998), and Tulip (Auber 2002). All three use partial drawing of the network in a limited amount of time to alleviate this situation. Some mechanisms to reduce complexity is clustering, reducing clusters to nodes, zooming in and out, panning, and filtering.

Although node and link scalability is grouped in this section, there is a difference between node scalability and link scalability. For example, networks containing a small number of nodes but a large number of links (dense networks) pose different challenges than networks containing a large number of nodes but a small number of links (sparse networks). Yet, they are analyzed under the same category as there are many common challenges posed by both.

The methods of clustering, zooming, and filtering are existing strategies employed in network visualization literature to deal with the complexity of networks. If those are considered and implemented in NVSS, they may or may not bring new insights. There is the question whether the application of these would be different in the context of semantic substrates. For example, filtering has already brought some insights when used on a time attribute. The continuous spatial representation of time allows users to manipulate a range filter and associate with the spatial

location of nodes, whereas this would not be possible in a network laid out without considering the time property.

In terms of complexity, there are many possibilities of filtering features to consider. Nodes can be filtered according to their type. Links can be filtered according to the properties of their source and target nodes. Users could select a set of nodes and filter adjacent¹ nodes (a link task). Similarly, there are many possibilities for selection. Nodes can be selected according to their attributes, a combination of those, according to the attributes of nodes connected to, a structural property such as in-degree or an out-degree, or a computed property, such as betweenness-centrality. The number of possibilities of the combination of selection and filtering are multiplicative. Among so many possibilities, the challenge is to spot the most valuable, effective, simple strategies to choose and design the interface accordingly. The question is what features and combinations would be most useful to have for users and where (in which domains, under what circumstances) do these features have this effect.

4.3.2 Scalability of the number of regions

Having more than 5 regions is expected to cause problems, especially when links are drawn. Filtering links accordingly may provide a partial solution, whereas rearranging regions may also do so but at the expense of reduced stability of the network. Using linkviz methods instead of drawing links might be the solution and seems very promising. Furthermore, the number of chunks for the user to deal with increases, which may lead to cognitive challenges. One example line of thought is whether hierarchical clustering would be a good solution, or whether it is better to have the user hide and show regions, perhaps dynamically cluster and offer multi-zoom capability, while hiding or collapsing the unimportant or unfocused ones. Since most of these methods are not new, the value of these methods in terms of their research contribution in this setting is questionable. How to deal with such issues requires further investigation, thinking, prototyping, and exploration.

4.3.3 Scalability in terms of node attribute values

Since nodes fall into regions and are laid out according to node attributes, the types and ranges of the attributes become important. First, the former is considered, and then the latter.

A categorical attribute having up to 5 values is suitable for use in determining regions. For more than 5, some intermediate strategies may be helpful, such as grouping them together. (How should they be grouped?) Or else, this will become a challenge for the scalability of the number of regions (see 4.3.2). An ordinal type may suggest orderly locations for regions; otherwise, it might be confusing to users. Numerical attributes tend to have a too long range (integer attributes) or a non-discrete character (real-valued attributes), and therefore may not be suitable to be used directly to determine regions. However, they could be binned and represent non-overlapping consecutive ranges. Determining the cut-off point might be a challenge for users (and confusing when two close elements into different regions) when there are no natural gaps on the continuum of values.

Attributes when used to place nodes within a region also pose challenges. When it is natural to use an attribute on the x-axis (e.g. time), long ranges pose a problem. Time, a common attribute, has the potential of posing challenges in terms of distribution. Certain periods may be inactive (even may have no nodes), while certain periods may be very active. Representing unevenly-spaced data over time (Aris 2005) may become an important issue for certain domains or datasets.

¹ Nodes connected to the selected set of nodes

Even with discrete attributes challenges arise when, for example, they are used to separate nodes within a region horizontally. Too many categories may be intractable to use, in which case, a method of choosing the important ones, binning, or other techniques (e.g. focus+context, zoom, etc.) may need to be employed. These are going to have (possibly disadvantageous) implications on other interactions, such as aligned regions may not be comparable any more.

4.4 Other issues

There are other issues that do not fit into the categories defined by previous sections. For example, if regions are aligned due to a time or time-like continuous attribute, having a common scale increases the complexity of the design. It is simpler to have all regions have a common scale or all regions have a different one, and it is more complicated to have some regions have a common scale. Defining those regions (by the user) as groups and calculations of the common scales need to be handled carefully not to impact the design (both internal & user interface) so that the consistency is not compromised.

Another issue is the generality of NVSS. There are many aspects of generality. One aspect is to be general in terms of the number of regions. Another one is to be general in terms of the data. Other aspects of generality are likely to appear as new features are added. It may be wise to limit the scope in some of those aspects. For example, currently the plan is to stay with simple directed networks.

A final issue is in terms of software design. Sometimes, there is an optimal order to add a feature into the software. When added in the wrong order, it may affect all components and require too many modifications at once, which both increase the time of implementation and risk of unintended inconsistencies in the application. In addition, there is a trade-off between functionality and complexity. The more functionality added to the software, the more complex it becomes, which makes the software harder to maintain. Even on the path to make the best design choices, the potential to make a costly error increases. Therefore, the best minimal set of features, which provide the most value, is essential to implement. Some features are modular and do not affect so much the main structure of the software, which are plausible to implement time permitting. In any case, a good design approach and refactoring principles for the development are to be employed.

5. Evaluation

As in any endeavor of exploration of novel user interface design principles and information visualization techniques, evaluation plays an important role to confirm or reveal the benefits of the novel contributions. Evaluation has two main roles in this proposal: (1) confirmation of validity and utility of new features, design decisions, and novel approaches (“novelty” in short), (2) feedback to improve as the work is in progress. Due to the second role, the author plans to evaluate intermittently as the work progresses. As a result, although the former role can theoretically take place at the very end, most of it will take place intermittently. As the novelty is evaluated, successful results are likely to have the first role, while the rest are to be considered for the second role.

Among qualitative and quantitative evaluation choices, a qualitative approach is more promising for novel user interfaces, especially when many design choices are involved (Shneiderman 2006). Given that the proposed work in this proposal involves many design decisions, there are many network visualization applications in use (Woolman 2005), and that many factors are involved

when comparing one approach to another; the author chooses a qualitative approach. Among several well-known qualitative approaches (Creswell 1998; Saraiya 2004; North 2006; Shneiderman 2006), the case study approach fits the best to this proposed work. To generalize results, more than one case study (3-6) will be conducted. The author strongly believes that the collective² case study approach will provide a wider range of deeper insights (on a number of different issues), will generate more useful feedback, and is likely to lead to principles and guidelines, which designers, practitioners, and researchers in information visualization could benefit from. One successful example of this approach is HCE (Seo 2006).

Each case study will consist of 4-6 meetings of at least 30 minutes consisting of participant observations and interviews. The following are potential questions to users during the case studies:

- What data sets are useful to visualize with the semantic substrate concept?
 - o This question may reveal that semantic substrates are useful for data sets according to certain criteria such as type of attributes, number of attributes, range of attribute values, and the size of the network (number of nodes, number of links, density, and distribution).
- What grouping strategies and placement methods of nodes are particularly useful to you? What makes those strategies and methods useful?
 - o This question may reveal important or common themes. For example, a method of arrangement may be particularly useful when one attribute is time or has the characteristics of time.
- How do you think of defining the substrate?
 - o This question may reveal features that could be particularly useful during the substrate creation process.
- What linkviz tasks do you need?
 - o This question is in search of factors affecting the tasks users need. For example, factors might be the type of domain or the type of dataset. The purpose is to arrive at statements, such as “When the dataset is of type X, linkviz tasks A, B, and C are very likely to be needed.”
- What issues do you encounter as the network grows in size? Do the solutions provided help you? What other solutions might be useful?
 - o This question addresses scalability issues and seeks feedback from users.

6. Research Outcomes and Expected Contributions

The outcomes of this research will be as follows:

- *Design*. How to design a system to support the tasks in section 4. There are two different kinds of design and strategic concerns for both.
 - o *Design of the interface*. This aspect explains the design of the interface to support the tasks. When more than one possible solution exists, the reasons for the chosen solution could be given (due to user understanding, performance, simplicity, familiarity, or feasibility in terms of implementation or system performance). The design of the substrate editor, the filtering, selection mechanisms on nodes and links, how the link tasks are supported (linkviz method mechanisms) and the choice of implementation of features supporting scalability (large number of nodes, dense networks, more than five regions, possibly clustering techniques and their integration). The implications of the design

² A term used to refer to more than one case study (Creswell, 1998).

choices to effectiveness (does it provide functionality to solve problems that are not solvable by other methods, which problems are they?), to efficiency, usability, and simplicity (how much time/effort does it take to learn them). The initial plan for the design of each task is as follows:

- For the substrate editor: Enable the user to define the size, the location of each region, and the placement method used within each region. Provide a mechanism to store and modify these settings. The placement method can be very diverse to define by the user. To begin with, provide an interface to define the x and y values of each node by attribute values.
 - For the linkviz methods: enable users to choose a linkviz method and then perform it. Provide a mechanism to choose an appropriate linkviz method given the dataset and possibly other factors. Enable the user to switch between drawing links and using a linkviz method.
 - For the scalability: Provide filters for nodes and links according to node attributes and/or structural properties such as in-degree and out-degree. Provide a mechanism to select a working set of the filters. Implement an overview mechanism mostly to be used when the number of nodes so large that many occlusions occur and/or it is very hard to draw them all and perceive. The overview could show the distribution of nodes rather than drawing them individually.
- *Design of the system.* This aspect explains the high level components in the system and how they work together. What parts of the system are chosen to be extendible, which parts are modular and how can they be used or (when applicable) reused by others as well as an insight to how they were realized? What is the general picture of the design that supports the complex, coherent, and highly interactive parts of the software? The trade-offs that were made to keep the complexity manageable and the software efficient. The restrictions that were made and the limitations of the software with possible comparison to other systems.

The initial plan for the system design by each task is as follows:

- For the substrate editor: Implement the substrate editor as a separate module that produces or works on an object representing the “substrate settings.” Enable this object to be stored to and read from a file. Make the settings specific to a dataset and consider reusing for variation of the same dataset.
- For the linkviz methods: Determine the collection of properties to be used by linkviz methods (node color, special node coloring (e.g. to use pie-chart metaphor), node size, etc.). Implement a generic interface to connect to and update these properties so that various linkviz methods or the user manually (e.g. size/color of node by one of the node attributes) can use them. Implement a working set mechanism for the linkviz methods to hide the ones that are not in use and to eliminate the ones that are not applicable. The generic mechanism to use linkviz methods will help to add new linkviz methods later. In this respect, the linkviz method collection will be extendible.
- For the scalability: Define modes for selected nodes, filtered nodes, and filtered links. Implement/augment the filtered set of nodes and links as a central mechanism and connect it to the active set of filters in use. The filters are to be generic up to the type of attributes. Provided that it is feasible to do so, implement the overview as a pluggable algorithm so

that another type of overviews is possible to add/replace with in the future.

- *Algorithms.* The following algorithms are to be provided.
 - Algorithms for the placement of the nodes and how the place updates with interaction (resize, etc.)
 - Linkviz methods and how they integrate with the other mechanisms such as selection and filtering whenever applicable.
 - The mechanism of linkviz methods and selection, maintenance of different collection of linkviz methods per datasets and compatibility detection algorithms as implemented.
 - Algorithms of features used for scalability such as overview, filtering, etc.
 - Optimization enhancements or special algorithms – if any – general or specific to datasets for large networks (scalability issues).
- *Software.* The final version of NVSS to be made available as an application on the web.
- *Insights.* The knowledge obtained from the case studies will be provided. Specifically, the list of link tasks in each domain (or case) and their relative importance or frequency may benefit other researchers. Looking over many case studies, predictions for common tasks or similar tasks across domains (or cases) are to be pointed out. This has the potential for other researchers or practitioners to provide knowledge from this experience and facilitate the discovery of tasks in another domain (or case).
- *Scientific advances.* Novel designs, ideas, algorithms if any that prove to be effective or seem to have potential to do so are to be provided.
- *Practical benefits.* Examples from each domain (or case) are to be provided with the explanation of how it benefited them to use NVSS with specific examples along with the users' reasoning during the example are to be provided at the end. This will illustrate the usefulness of this research in concrete terms.

7. Detailed Plan of Work

The following work plan describes a proposed schedule for future work that leads to a completed dissertation by December 2007 graduation:

Date	Implementation/Evaluation	Reporting
May 2006	Look for other possible datasets to visualize. Find a solution for the issue of common x-axis.	Present NVSS with the precedent dataset in HCIL SOH 2006.
September 2006	Generalize NVSS to accept another dataset (generalize internal structures, implement input/output formats, etc.) Implement substrate editor. Find & train case study users. Prepare them to make report their experience during October if possible.	Prepare to write a paper on substrate editor issues and experience of visualizing two datasets using the semantic substrate idea. Prepare/plan to collect user experiences. Submit to CHI 2007 (deadline: 29 Sep 2006).
October 2006	(If possible) case study users using NVSS and taking notes.	If paper accepted, attend InfoVis 2007 & present paper.
November 2006	Collect case study user experiences. Make adjustments to substrate editor. Implement linkviz methods for one dataset.	IEEE Visualization/VAST Doctoral Colloquium (November 2, 2006)
December 2006	Collect user experiences & evaluate the linkviz methods. Prepare for the doctoral consortium.	
January 2007	Generalize the linkviz methods mechanism and its integration with other mechanisms. Improve selection & filtering mechanisms.	Aim to attend CHI Doctoral Consortium (12 Jan 2007). Topic: linkviz method mechanism & possible taxonomy and applicability on two domains (optional: predictions for other domains).
February 2007	Generalize the linkviz method mechanism to a second dataset.	Begin writing paper for InfoVis 2007 (probably deadline is March 31, 2007) on linkviz method mechanism.
March 2007	Collect user experiences, revise and improve implementation. Integrate the linkviz method mechanism to selection & filtering mechanisms, if applicable.	Incorporate user experiences, revisions and improvements on the implementation to the InfoVis 2007 paper and submit it.
April 2007	Generalize/improve important mechanisms if needed. Publish NVSS on the web.	Write manual. Publish NVSS on the web.
May 2007	Start thinking about the scalability issues. Think about how mechanisms	Present NVSS with new features in HCIL SOH 2007.

	will adapt to the scaling, specifically from 1,000 to 10,000 nodes.	
June-August 2007	Implement scalability. Test under different circumstances and with different datasets. Implement optimizations. Implement new techniques to overcome scalability problems. These techniques may involve overview, aggregation, clustering, filtering and others as well as a combination of them.	(Optional) Write a survey paper on scalability issues in network visualization. Write a paper for CHI 2008.
September 2007	Perform evaluations on the methods that are implemented for scalability issues. Either conduct user studies or collect user experiences.	Revise and submit to CHI 2008 (possible deadline: 29 September 2007)
October 2007	Write, revise, and defend dissertation.	Write a journal paper on the methods, mechanisms and user experiences (Information Visualization or IEEE TVCG).
November 2007	Final adjustments to the NVSS software. Look for further enhancements, possible future directions, etc.	May write a paper on future directions, guidelines for designers and practitioners who use network visualization.
December 2007	Graduation	Complete & submit possible paper, revise dissertation and finalize.

8. Bibliography

Aris, A., Ben Shneiderman, Catherine Plaisant, Galit Shmueli, Wolfgang Jank (2005). "Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration." Proceedings of the International Conference on Human-Computer Interaction (INTERACT 2005) **LNCS 3585**: 835-846.

Auber, D. (2002). "Tulip an information visualization software for huge graphs." IEEE Computer Graphics Forum.

Becker, R. A., Stephen G. Eick, and Allan R. Wilks (1995). "Visualizing Network Data." IEEE Transactions on Visualization and Computer Graphics **1**(1): 16-28.

Best, C., and Hans-Christian Hege (2002). "Visualizing and Identifying Conformational Ensembles in Molecular Dynamics Trajectories." Computers in Science and Engineering **4**(3): 68.

Bilgic, M., Louis Licamele, Lise Getoor, and Ben Shneiderman (2005). "D-Dupe: An Interactive Tool for Entity Resolution in Social Networks." Poster at 13th International Symposium on Graph Drawing.

Borner, K., Chaomei Chen, and Kevin W. Boyack (2003). "Visualizing Knowledge Domains." Annual Review of Information Science and Technology **37**.

Brandenburg, F. J. (1988). "Nice Drawing of Graphs are Computationally Hard." Visualization in Human-Computer Interaction **LNCS 439**: 1-15.

Brandes, U., and Dorothea Wagner (2003). Visone - Analysis and Visualization of Social Networks. Special Issue on Graph Drawing Software, Springer Series in Mathematics and Visualization. M. Juenger, P. Mutzel, Springer-Verlag: 321-349.

Breitkreutz, B.-J., Chris Stark, and Mike Tyers (2003). "Osprey: a network visualization system." Genome Biology **4**(3): R22.

Creswell, J. W. (1998). Qualitative Inquiry and Research Design: Choosing Among Five Traditions, Sage Publications.

Davidson, R., and David Harel (1996). "Drawing Graphs Nicely using Simulated Annealing." ACM Transactions on Graphics **15**(4): 301-331.

De Nooy, W., Andrej Mrvar, Vladimir Batagelj, and Mark Granovetter (2005). Exploratory Social Network Analysis with Pajek Cambridge University Press, UK.

- Di Battista, G., P. Eades, R. Tamassia, and I.G. Tollis (1999). Graph Drawing: Algorithms for visualization of graphs, Prentice Hall.
- Eades, P. (1984). "A Heuristic for Graph Drawing." Congressus Numerantium **42**: 149-160.
- Eades, P., and Qingwen Feng (1996). "Multilevel Visualization of Clustered Graphs." Proceedings of Graph Drawing LNCS 1190: 101-112.
- Fruchterman, T. M. J., and E.M.Reingold (1991). "Graph Drawing by Force-directed Placement." Software-Practice and Experience **21**(11): 1129-1164.
- Gansner, E., and S. North (1998). "Improved Force-Directed Layouts." Proceedings of Graph Drawing LNCS 1547: 364-373.
- Garfield, E. (2004). "Historiographic Mapping of Knowledge Domains Literature." Journal of Information Science **30**(2): 119-145.
- Gary, M. R., and D. S. Johnson (1983). "Crossing number is NP-complete." SIAM J. Algebraic and Discrete Methods **4**: 312-316.
- Ghoniem, M., Jean-Daniel Fekete, and Philippe Castagliola (2004). "A Comparison of the Readability of Graphs using Node-Link and Matrix-Based Representations." Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04): 17-24.
- Hadany, R., and D. Harel (2001). "A Multi-Scale Algorithm for Drawing Graphs Nicely." Discrete Applied Mathematics **113**(1): 3-21.
- Harel, D., and Y. Koren (2000). "Drawing Graphs with Non-Uniform Vertices." Proc. Conf. on Advanced Visual Interfaces (AVI2000): 157-166.
- Harel, D., and Y. Koren (2000). "A Fast Multi-Scale Method for Drawing Large Graphs." Proceedings of Graph Drawing 2000 LNCS 1984: 183-196.
- Heer, J., and D. Boyd (2005). "Vizster: Visualizing Online Social Networks." IEEE Symposium on Information Visualization.
- Herman, I., Guy Melançon, and M. Scott Marshall (2000). "Graph Visualization and Navigation in Information Visualization: A Survey." IEEE Transactions on Visualization and Computer Graphics **6**(1): 24-43.
- Huffaker, B., Evi Nemeth, and K. Claffy (1999). "Otter: A general-purpose network visualization tool." Internet Society INET'99 Conference.
- Kamada, T., and S. Kawai (1989). "An algorithm for drawing general undirected graphs." Information Processing Letters **31**(1): 7-15.

Kamps, T., J. Kleinz, and J. Read (1995). "Constraint-Based Spring-Model Algorithm for Graph Layout." Proceedings of Graph Drawing 95 LNCS 1027: 349-360.

Kang, H., and B. Shneiderman (2005). Personal Media Exploration: A Spatial Interface to User-defined Semantic Regions, University of Maryland.

Kosak, C., Joe Marks, and Stuart M. Shieber (1994). "Automating the Layout of Network Diagrams with Specified Visual Organization." IEEE Transactions on Systems, Man and Cybernetics 24(3): 440-454.

Lee, B., Mary Czerwinski, George Robertson, and Benjamin B. Bederson (2005). "Understanding research trends in conferences using paperLens." CHI '05 extended abstracts on Human factors in computing systems: 1969-1972.

Misue, K., P. Eades, W. Lai, and K. Sugiyama (1995). "Layout Adjustment and the Mental Map." Journal of Visual Languages and Computing 6(2): 183-210.

Munzner, T. (1998). "Drawing Large Graphs with H3Viewer and Site Manager." Proc. Symp. Graph Drawing GD'98: 384-393.

Nardi, B., S. Whittaker, E. Isaacs, M. Creech, J. Johnson, and J. Hainsworth (2002). "Integrating Communication and Information Through ContactMap." Communications of the ACM 45(4): 89-95.

North, C. (2006). "Toward Measuring Visualization Insight." IEEE Computer Graphics and Applications 26(3): 6-9.

Phan, D., Ling Xiao, Ron Yeh, Pat Hanrahan, and Terry Winograd (2005). "Flow map layout." Information Visualization: 219-224.

Purchase, H. C. (1997). "Which aesthetic has the greatest effect on human understanding?" Proc. Symp. Graph Drawing GD'97: 248-261.

Purchase, H. C., R.F. Cohen, and M. James (1996). "Validating Graph Drawing Aesthetics." Proc. Symp. Graph Drawing GD'95: 435-446.

Saraiya, P., Chris North, Karen Duca (2004). "An Evaluation of Microarray Visualization Tools for Biological Insight." IEEE Symposium on Information Visualization (INFOVIS'04): 1-8.

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

Seo, J., and Ben Shneiderman (2006). "Knowledge Discovery in High-Dimensional Data: Case Studies and a User Survey for the Rank-by-Feature Framework." IEEE Transactions on Visualization and Computer Graphics **12**(3): 311-322.

Shneiderman, B., and Catherine Plaisant (2006). "Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies." Proceedings of the BELIV'06 workshop, Advanced Visual Interfaces Conference.

Sindre, G., B. Gulla, and H. Jokstad (1993). "Onion Graphs: Aesthetic and Layout." Proc. 1993 IEEE Symposium on Visual Languages: 287-291.

Storey, M. A., M. Musen, J. Silva, C. Best, N. Ernst, R. Ferguson, and N. Noy (2001). "Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protege." Workshop on Interactive Tools for Knowledge Capture.

Sugiyama, K. (1987). "A cognitive approach for graph drawing." Cybernetics and Systems **18**(6): 447-488.

Sugiyama, K., S. Tagawa, and M. Toda (1981). "Methods for visual understanding of hierarchical system structures." IEEE Transactions on Systems, Man and Cybernetics **SMC-11**(2): 109-125.

Tufte, E. R. (1983). The Visual Display of Quantitative Information. Cheshire, Connecticut, Graphics Press.

Ware, C., Helen Purchase, Linda Colpoys, and Matthew McGill (2002). "Cognitive measurements of graph aesthetics." Information Visualization **1**(2): 103-110.

Wattenberg, M. (2006). "Visual exploration of multivariate graphs." CHI 2006.

Wills, G. J. (1999). "NicheWorks -- Interactive Visualization of Very Large Graphs." Journal of Computational and Graphical Statistics **8**(2): 190-212.

Woolman, M. (2005). "Visual Complexity." <http://www.visualcomplexity.com/vc/index.cfm> Retrieved December 7, 2005.

9. Reading Lists

9.1 Information Visualization

C. Ahlberg and B. Shneiderman, Visual information seeking: tight coupling of dynamic query filters with starfield displays. *Conference on Human Factors in Computing Systems*, 313-317, 1994.

B. Shneiderman, Dynamic queries for visual information seeking. *IEEE Software*, 11(6), 70-77, 1994.

B. Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the Symposium on Visual Languages (VL '96)*, 336-343, 1996.

M.C. Chuah, Dynamic aggregation with circular visual designs. *IEEE Symposium on Information Visualization*, 35-43, 1998.

John Lamping, Ramana Rao, and Peter Pirolli, A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *Proc. ACM Conf. Human Factors in Computing Systems, CHI*, 2005.

C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, LifeLines: Visualizing personal histories. *ACM CHI '96 Conference Proc.*, 221-227+518, 1996.

H. Hochheiser and B. Shneiderman, Dynamic query tools for time series data sets: timebox widgets for interactive exploration, *Information Visualization*, 3(1), 1-18, Spring 2004.

Colin Ware and Robert Bobrow, Motion to support rapid interactive queries on node-link diagrams. *ACM Trans. Appl. Percept.* 1(1), 3-18, July 2004.

9.2 Network Visualization

Richard A. Becker, Stephen G. Eick, and Allan R. Wilks, Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics* 1(1), 16-28, 1995.

Peter Eades and Qingwen Feng, Multilevel visualization of clustered graphs. *Proceedings of Graph Drawing*, volume 1190 of LNCS, 101-112, 1996.

J. Heer and D. Boyd, Vizster: Visualizing online social networks. *IEEE Symposium on Information Visualization*, IEEE Press, Piscataway, NJ, 2005.

Ivan Herman, Guy Melançon, and M. Scott Marshall, Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 24-43, 2000.

Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola, A comparison of the readability of graphs using node-link and matrix-based representations. *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, 17-24, 2004.

Bongshin Lee, Mary Czerwinski, George Robertson, and Benjamin B. Bederson, Understanding eight years of InfoVis conferences using PaperLens. *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04) - Volume 00*, 216.3, 2004.

Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst, Animated exploration of dynamic graphs with radial layout. *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, 43-50, 2001.

Graham J. Wills, NicheWorks - Interactive visualization of very large graphs, *Journal of Computational and Graphical Statistics*, 8(2), 190-212, 1999.

9.3 Measurement/Evaluation Methods, Examples, and Criteria

Jim Blythe, Cathleen McGrath, and David Krackhardt, The effect of graph layout on inference from social network data. *Proceedings of the Symposium on Graph Drawing*, Lecture Notes in Computer Science; Vol. 1027, 40-51, 1995.

H. C. Purchase, Which aesthetic has the greatest effect on human understanding? *Proc. Symp. Graph Drawing GD'97*, 248-261, 1997.

H. C. Purchase, R.F. Cohen, and M. James, Validating graph drawing aesthetics. *Proc. Symp. Graph Drawing GD'95*, 435-446, 1995.

Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill, Cognitive measurements of graph aesthetics. *Information Visualization* 1(2), 103-110, 2002.

Catherine Plaisant, The challenge of information visualization evaluation, *Proc. Of Conf. on Advanced Visual Interfaces*, AVI'04 (2004), p.109-116.

A. Kobsa, User experiments with tree visualization systems. *Proceedings of InfoVis 2004, IEEE Symposium on Information Visualization*, 9-16. 2004.

Purvi Saraiya, Chris North, and Karen Duca, An evaluation of microarray visualization tools for biological insight. *IEEE Symposium on Information Visualization (INFOVIS'04)*, 1-8, 2004.

Chris North, Toward measuring visualization insight. *IEEE Computer Graphics and Applications* 26(3), 6-9, May/Jun, 2006.