Information Visualization (2007), 1–20
© 2007 Palgrave Macmillan Ltd. All rights reserved 1473-8716 $30.00

www.palgrave-journals.com/ivs

Designing semantic substrates for visual network exploration

Aleks Aris[1]
Ben Shneiderman[1]

[1] *Computer Science Department & Human–Computer Interaction Lab, University of Maryland, College Park, MD, U.S.A.*

Correspondence:
Ben Shneiderman, Computer Science Department & Human–Computer Interaction Lab, University of Maryland, College Park, MD20742, U.S.A.
Tel: +1 301 405 2680;
Fax: +1 301 405 6707;
E-mail: ben@cs.umd.edu

Received: 27 July 2007
Revised: 2 October 2007
Accepted: 3 October 2007

Abstract

A semantic substrate is a spatial template for a network, where nodes are grouped into regions and laid out within each region according to one or more node attributes. This paper shows how users can be given control in designing their own substrates and how this ability leads to a different approach to network data exploration. Users can create a semantic substrate, enter their data, get feedback from domain experts, edit the semantic substrate, and iteratively continue this procedure until the domain experts are satisfied with the insights they have gained. We illustrate this process in two case studies with domain experts working with legal precedents and food webs. Guidelines for designing substrates are provided, including how to locate, size, and align regions in a substrate, which attributes to choose for grouping nodes into regions, how to select placement methods and which attributes to set as parameters of the selected placement method. Throughout the paper, examples are illustrated with NVSS 2.0, the network visualization tool developed to explore the semantic substrate idea.

Information Visualization advance online publication, 22 November 2007; doi:10.1057/palgrave.ivs.9500162

Keywords: Network visualization; semantic substrate design; information visualization; data exploration and analysis; graphical user interfaces

Introduction

Successful network visualization tools enable domain experts to carry out key tasks such as recognizing clusters, identifying interesting nodes, discovering patterns of links, and detecting unusual relationships. The diversity of network types has inspired a rich variety of network visualization tools, often based on force directed layouts.[1] However, many of these produce network visualizations with numerous link crossings, occluded nodes, cluttered displays, and unreadable labels, all of which decrease comprehensibility and interfere with task completion.[2] To reduce these problems, many algorithm designers place nodes to minimize link crossings, minimize the longest link, or maximize the minimum angle between links.[3] As a result, their algorithms position nodes in what seems like an arbitrary location on the display.

The desire to place nodes in a way that was comprehensible to users led to the idea of semantic substrates for node layout. These algorithms use attribute values to place nodes in meaningful stable locations that facilitate discovery by enabling users to see patterns, outliers, and gaps (see Shneiderman and Aris[4] and Lee *et al.*[5] for taxonomies of network visualization tasks). We were inspired by the clear benefits of map-based layouts, such as cities on a familiar map of the United States, so we sought to create meaningful spatial layouts in which users could understand node placement, spot relationships among nodes, and notice regions where nodes were absent or sparse.

Semantic substrates require two conceptual steps to organize nodes. First, nodes are grouped into rectangular regions according to one of their attributes. Second, nodes are placed in each region according to one or more other attribute values. Once the nodes are organized, user control of link visibility according to their source and destination regions reduces the cluttered displays that exist in many implementations. The utility of semantic substrates was illustrated in the legal precedent domain where nodes were court decisions and links were legal precedents.[4]

This paper introduces the Substrate Designer, a tool for users to specify rectangular regions, attributes for grouping nodes into regions, and attributes for placing them within regions. In addition, users can specify the placement algorithm and decide on additional visual parameters, such as node and region colors to complete the design of a substrate. Different data sets and tasks influence the design of effective substrate features such as the number of regions, their spatial relationship, their size, and the type of filters to provide. This paper shows how to leverage user control on the substrate features to accelerate exploration of a network data set. Enabling users to design their own substrates dramatically facilitates the iterative process of creating, applying, editing, and reapplying substrates to a data set. We believe that semantic substrates will be beneficial for many network analysis tasks, but are especially effective when there are at least 2–3 attributes for each node.

The next section reviews relevant work about designing semantic substrates. The subsequent section describes the process of substrate design, while the fourth section provides two case studies based on our work with domain experts. We show how changing the substrate enables users to gather new insights about their data sets and illustrates the process of semantic substrate design. The fifth section provides guidelines for semantic substrate design. The penultimate section provides discussion that leads to future work and the final section concludes the paper.

Relevant work

The notion of user-defined semantic substrates proved beneficial in a network visualization tool for author name resolution in bibliographic databases.[6] Author name nodes were laid out in five distinct regions so users could quickly spot shared and non-shared co-authors for suspected duplicate names. Another inspiration for semantic substrates is the user-defined spatial layout for photos with shared attributes.[7]

Six recent systems have elements of semantic substrates. Jambalaya[8] integrates SHriMP views into the Protégé framework. A graph metaphor is used to show links between concepts, similar to regions in NVSS (NVSS stands for Network Visualization by Semantic Substrates, the name of the visualization tool to explore the semantic substrates idea), which may include sub-concepts (subclasses). Users can manually place the nodes or automatically order them by a structural property of nodes, such as number of children, however, not by node attributes. Links are categorized and therefore can be color-coded by source and target classes. PivotGraph[9] places nodes on a two-dimensional (2D) grid by their node attributes and nicely aggregates nodes by their attributes to present a useful overview. While PivotGraph aggregates nodes, NVSS shows all nodes. Nodes having the same placement attributes are either spread out or put next to each other in NVSS. In addition, PivotGraph has only one region in NVSS terminology, while NVSS has many regions. Users can select node attributes on the *x*- and *y*-axis. Pretorius *et al.*[10] represent multi-dimensional transitional systems as networks and uses the projection of multi-valued node attributes to the 2D plane to position nodes. The projection is parametrized and user adjustable, which users could experiment with to arrive at a good projection that fits their needs. The visualization utilizes a grid–plot arrangement algorithm with the extension of nested and rotated grid. NVSS also enables users to control the size and location of regions (similar to grids). In all these systems, nodes are arranged in a grid–plot layout. NVSS allows multiple regions and allows users to choose a different node placement method for each region. Kosak *et al.*[11] group nodes according to their type and show two ways of organizing the nodes within each group: rule-based and using genetic algorithms. The rule-based layout may be used to group and place nodes in terms of their node attributes; however, the specification is manual. NVSS uses node attributes directly and lets users specify the attributes that the nodes will be placed by. In a way, this provides a faster and more intuitive approach to users. In addition, Kosak *et al.* focus on computing the layout while NVSS also provides link-visibility features. In Constellation,[12] horizontal and vertical positions of nodes are based on the specific attribute value of 'pathway importance'. Then, a further optimization pass is done to increase information density. Dig-CoLa[13] and IPSep-CoLa[14] extend the force-directed approach by layout constraints, which can have the same effect of placing nodes according to their node attributes. Constraints also include separation constraints, which enhance the visual representation of the graph, such as avoiding overlaps of nodes and clusters. IPSep-CoLa[14] has the additional capability to cluster nodes into rectangles according to an attribute value, such as all cereals of a given manufacturer.

Although many systems; such as GGobi,[15] Tulip,[16] NicheWorks,[17] SocialAction,[18] Visone,[19] and Osprey,[20] Glide[21] provide other useful features, they do not support layouts based on node attributes:

GGobi[15] uses radial, dot, and neato layouts; allows users to manually edit node locations and categorize links (by creating 'edge sets'); supports different views, such as scatterplots, barcharts, and parallel coordinate charts, and provides brushing between linked views. The (jittered) scatterplots in GGobi do not show links although they can be brushed to the node–link diagram. Although

GGobi scatterplots are similar to the GridPlotXY placement method in NVSS in terms of using a node attribute for the *x*- and *y*-axis to place nodes on display, there are differences. While GGobi scatterplots use jitter to eliminate overlap of nodes (having the same attribute values on both *x*- and *y*-axis), NVSS simply places them next to each other within a cell (columnwise, from top to bottom, starting another column from left to right as needed).

Tulip[16] supports node attributes, user interaction to manage clusters (group and ungroup nodes), and has plug-in capability for defining new layout algorithms. Although it may be theoretically possible, no layout based on node attributes (there is a treemap rendering; however, it does not include links and it is limited to the hierarchical treemap algorithm) and no link visibility based on node attributes has been reported.

Visualizing large graphs (up to 1,000,000 nodes) is a driving goal for NicheWorks,[17] which uses several initial layouts (circular layout, hexagonal grid, and tree layout). Its incremental algorithms, such as steepest descent and simulated annealing, compute the final layout and supports filtering on node attributes.

Force directed layouts are used in SocialAction,[18] but it can show clustered groups of nodes called 'communities' that are determined by using a structural clustering algorithm with user-controlled parameters. SocialAction filters nodes using rankings on statistical information (such as betweenness-centrality) but does not control link (or node) visibility based on node attributes.

Visone[19] provides a set of different algorithms to layout nodes, such as spectral, layered, and radial layouts.

A domain-specific tool, Osprey,[18,20] enables biologists to combine data sets and provides node filters based on attributes.

Glide[21] provides users with Visual Organization Features (VOFs) to apply to the graph to organize node locations. The VOFs are based on spatial placement principles (i.e. building blocks of aesthetic principles) and the graph is updated as users apply them manually. He and Marriott[22] provide a way to layout nodes according to user-defined constraints that assigns nodes suggested values and places them accordingly. Their system takes in constraints over the *x* and *y* positions of the nodes and a partial assignment of suggested values for the node coordinates. Dengler *et al.*[23] provide a similar framework that deals with visual features and does not use node attributes to place nodes.

A different approach to network visualization is matrix-based (Ghoniem *etal.*[24] MatrixExplorer[25]). MatrixExplorer couples matrices with node–link diagrams and provides interactive operations such as sorting matrix columns in terms of attributes and filtering. The node–link representation does not use layouts based on node attributes.

Substrate design

The notion of semantic substrates was introduced and its utility illustrated using NVSS 1.0.[4] This paper uses NVSS 2.0 to demonstrate how users can design their own substrates.

NVSS 2.0 has a module called the 'Substrate Designer' (called 'the designer' for brevity). The Substrate Designer module enables users to create, save, and edit semantic substrates for their own network data sets, which can be applied to their data via NVSS 2.0.

Regions in a substrate have spatial properties, such as location and size; visual properties, such as background color and node color; and algorithmic properties, such as the criterion that determines which nodes will fall into and how nodes will be placed within this region. In both of the algorithmic properties, attributes are used to determine node inclusion and node placement criteria within a region. Therefore, in a substrate definition, at least the attributes that are being used need to be defined. If the substrate could be edited, then all possible attributes must be derivable from the stored substrate representation used by the editor. Additional settings of the substrate include its size, node sizes, and link colors.

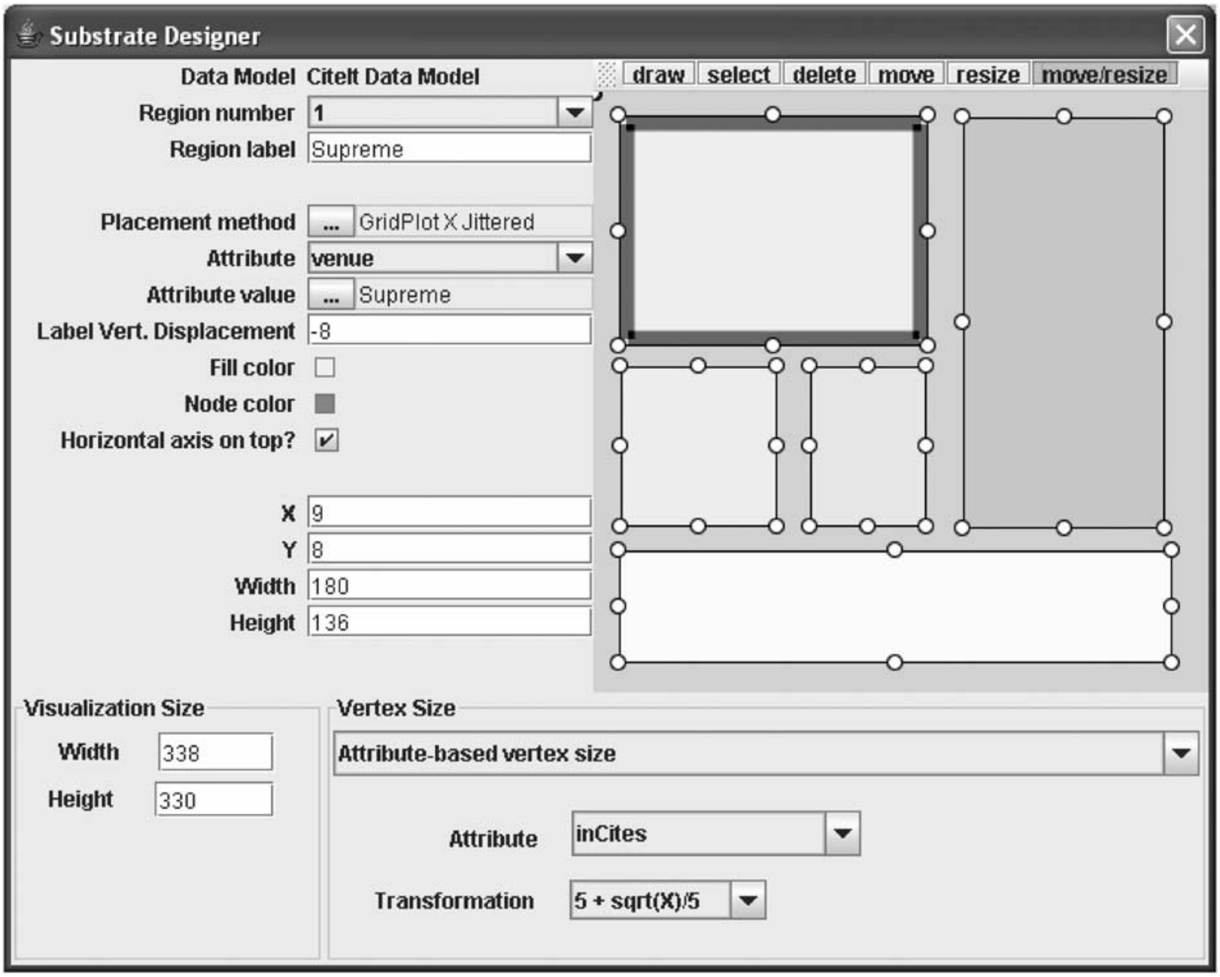
NVSS 2.0 allows users to set all the region properties within the designer (Figure 1). In addition, users can set the substrate and node size within the designer, while they define link colors by modifying the default link colors from within the instance of the visualized data set that uses this substrate. Once users define link colors, they can save the substrate.

NVSS 2.0 allows users to create a new substrate or load, edit, and save an existing substrate. Once a substrate is loaded, users can specify the data set files (consists of nodes and links file) and launch the visualization using that substrate. Each attribute in the data model of a substrate has a name and a type (INTEGER, DOUBLE, STRING, or DATE). Substrates in NVSS 2.0 are independent of the data, and therefore can be reused for other data sets, as long as the data model is compatible.

The top part of the designer displays features of regions, while the bottom part displays features of the entire substrate. At the bottom part on the left, users can set the visualization size by either entering the width and height or resizing the designer window by the usual mouse dragging to resize windows. At the bottom part on the right, users can select the method to determine node size. Currently, the two available methods are 'constant size' with a specified diameter, and 'attribute-based' size, where users select an attribute and optionally apply a square root transformation on it. Users can also create an external attribute and use that attribute to determine node sizes.

Users can visually create the regions on the top right-hand side once they are in 'draw' mode. The other modes are 'select', 'delete', 'move', 'resize', and 'move/resize'. In 'select' mode, users can select a region and modify its details (elaborated below). Users delete regions in 'delete' mode, while they relocate or resize the regions in 'move' and 'resize' modes, respectively. In 'move/resize' mode, they can do either.

The top left-hand side shows the details of the selected region. Users can select visual attributes of a region, such



Data Model CiteIt Data Model
Region number 1
Region label Supreme
Placement method ... GridPlot X Jittered
Attribute venue
Attribute value ... Supreme
Label Vert. Displacement -8
Fill color □
Node color ■
Horizontal axis on top? ☑
X 9
Y 8
Width 180
Height 136

Visualization Size
Width 338
Height 330

Vertex Size
Attribute-based vertex size
Attribute inCites
Transformation 5 + sqrt(X)/5

Figure 1 The NVSS 2.0 module that enables users to design their own semantic substrates. The user has created five regions, assigned their background colors and is in the process of moving/resizing them.

as the location (X and Y), the size (Width and Height), the color of nodes, and the background color (Figure 2).

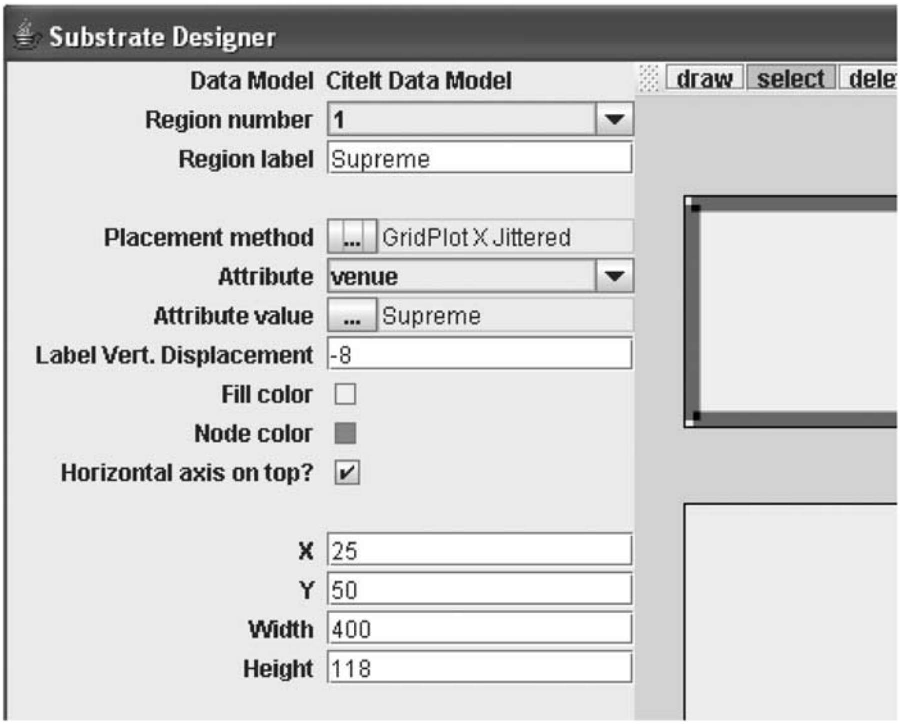
The rest of the settings involve placement of nodes: (1) which region and (2) how they are placed within this selected region. Users select an attribute and an attribute value to place nodes into the selected region. The *venue* attribute and its 'Supreme' value is set for the purple selected region (the grey highlighting inside edges of the purple region indicates it is the selected region) (Figure 3).

The dialog shows available algorithms and choices to select from available attributes to be set as parameters for the selected algorithm. The available algorithms are GridPlotX, GridPlotY, and GridPlotXY. GridPlotX and GridPlotY allow users to select an attribute for the *x*- and *y*-axis respectively, while leaving the other axis free. The jittered versions of these two algorithms introduce jitter along the free axis (up/down shifts of all nodes in alternating horizontal and vertical slots for *x*-jittered and *y*-jittered, respectively). GridPlotXY allows users to set attributes for both axes. For all five GridPlot algorithms, users select a minimum value, a maximum value and the number of bins between these values to define the attribute values on the axis along *x* or *y*. For nominal attributes (STRING type), NVSS 2.0 uses only alphabetical ordering. To use another ordering, users can create a derived attribute (based on the nominal attribute) externally and make the order of the derived attribute conform to the desired order to get a similar effect.

When users are done, they close the designer, which will prompt a dialog to save the substrate to a file.

Examples

The following examples show how substrate design helps users explore their data. The examples show how user needs influence substrate design and how the outcomes enable certain insights. There is a division of labor when designing substrates: Our users, who are knowledgeable



Substrate Designer

Data Model CiteIt Data Model

Region number 1

Region label Supreme

Placement method ... GridPlot X Jittered

Attribute venue

Attribute value ... Supreme

Label Vert. Displacement -8

Fill color □

Node color ■

Horizontal axis on top? ☑

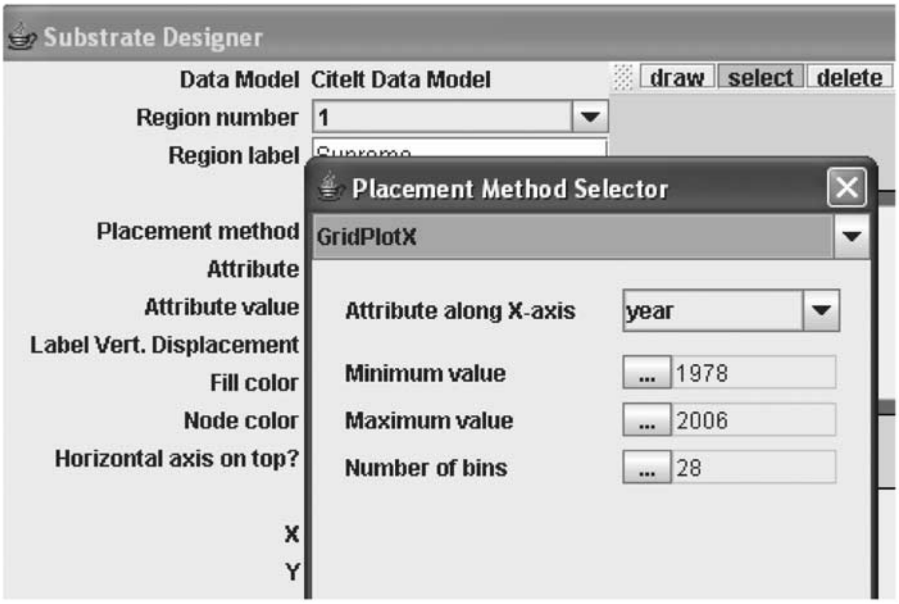
X 25

Y 50

Width 400

Height 118

Figure 2 Users can set visual properties of a region in the Substrate Designer. The pink selected region is highlighted by an inside shaded border to indicate it is the selected region. The settings of the pink region are displayed on the left-hand side.



Substrate Designer

Data Model CiteIt Data Model

Region number 1

Region label

Placement method

Attribute

Attribute value

Label Vert. Displacement

Fill color

Node color

Horizontal axis on top?

X

Y

Placement Method Selector

GridPlotX

Attribute along X-axis year

Minimum value ... 1978

Maximum value ... 2006

Number of bins ... 28

Figure 3 Users define the placement method for the selected pink region (i.e. the 'Supreme' region in Figure 2). The placement method is comprised of a placement algorithm and its parameters. In this case, GridPlotX, a grid plot placement algorithm that uses only the *x*-axis to place the node, is selected. The attribute along the *x*-axis is defined to be year with binning parameters that define the minimum (1978) and maximum (2006) values and the number of bins (28).

in the domain of their data sets, are the 'domain-experts'. Whoever designs the substrates for them are the 'tool-experts'. In both cases, our users were the domain experts and we were the tool experts. We listened to them to understand their needs and designed the substrates for them. However, we envision that domain experts can become (with increased exposure, some instruction and practice) the tool experts.

Two data sets are shown: the first one is legal precedent data (the same data set used in[4]) and the second example is a food-web data set.

Legal precedent data

The first example is a legal court case (Throughout the paper, 'case' is used to refer to a legal court decision.) data set that our collaborators wanted to explore. We, as the tool experts, designed the substrates to meet the needs of their exploration. One of our collaborators is Professor Wayne McIntosh, who holds a faculty position in the Government and Politics Department at the University of Maryland and is the leader of the Cite-It project, which aims to analyze and understand the evolution of regulatory takings cases over the years. Other team members include Ken Cousins and Stephen Simon. Ken Cousins is a visiting assistant professor of the Political Science Department at Western Washington University. Stephen Simon currently is an assistant professor of political science at the University of Richmond. In this multidisciplinary project, our domain experts are knowledgeable in different aspects of the domain. Our work over 16 months covered data identification, data collection and filtering, followed by problem analysis to develop requirements for visualization.

We spent a dozen sessions of 10–60 min with our collaborators (sometimes one person, sometimes all three people). Each of our domain experts spent time with the tool by themselves and showed it to their colleagues. They also use screenshots of the data set to communicate facts among themselves and other domain experts through presentations and research papers. We (as the tool experts) and our domain experts agreed on the design of the substrate quickly, (usually in 1–3 major iterations), deciding on the regions, their placement, the grouping and placement attributes for nodes. The approach we used to arrive at the initial substrate could be considered as a trial and error approach, which ended quickly with a satisfying substrate. The other substrates were created via design-by-example. We copied the initial substrate and modified it until we achieved the other types of arrangements we envisioned.

Nodes represent legal court cases from 1978 to 2005 concerning the legal issue known as 'regulatory takings' and links represents legal citations from one court case to another. Figure 4 shows the result of applying the first substrate to this data set. The data set is a subset of a larger data set with 2345 nodes and 14,401 unique links and contains 287 nodes and 2032 links.

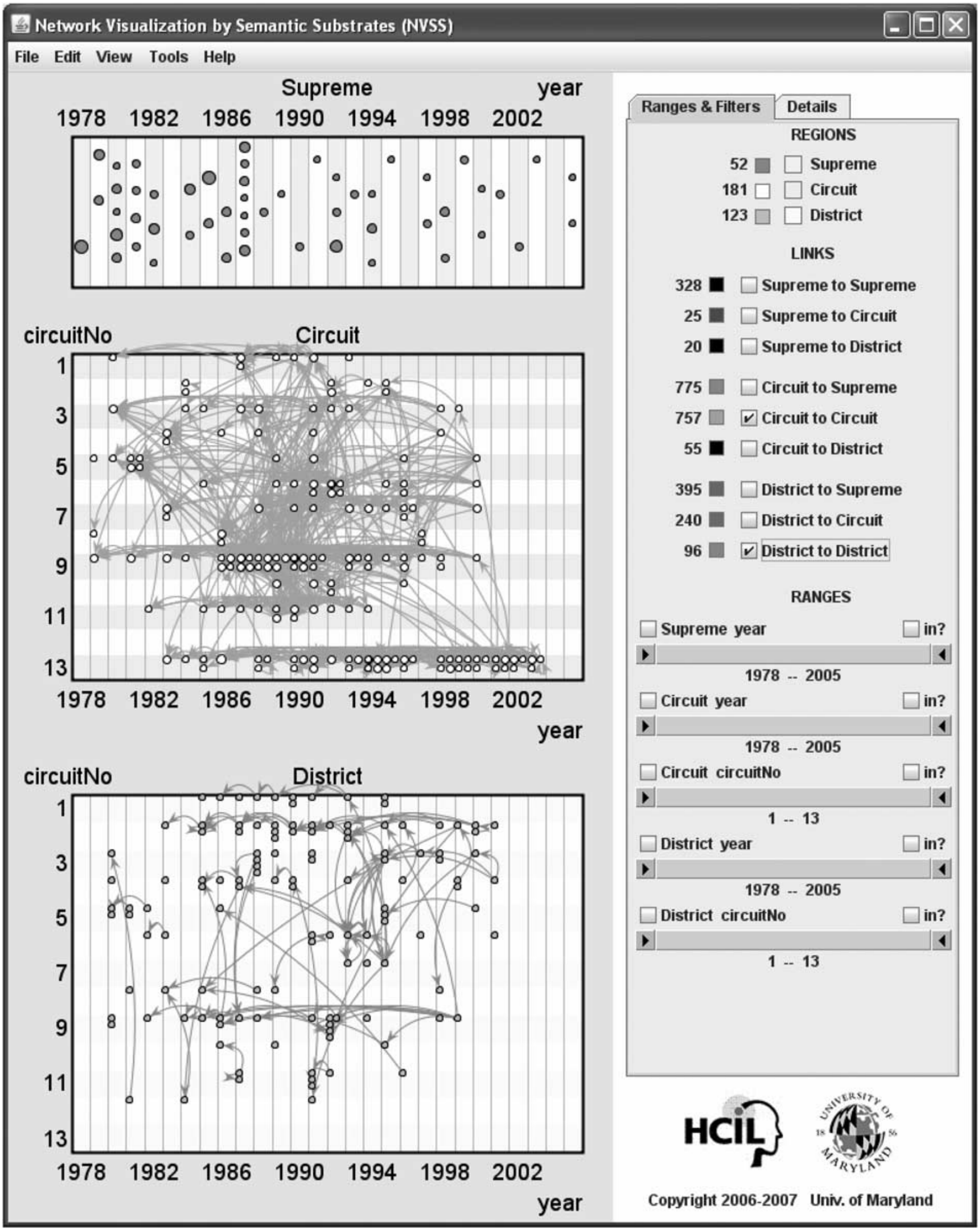
Nodes have the following attributes in this subset: *caseId*, *date*, *year*, *venue*, *venue2*, *circuitNo*, *inCites*, *outCites*, *cite*, and *name*. *caseId* is a unique integer to uniquely identify each case. *date* is the date of the case. *year is* the year part of *date*, a derived attribute. *venue* is the type of court the case was held with values 'Supreme', 'Circuit', and 'District'. *venue2* is the court name. *circuitNo* is a derived attribute that ranges from 1 to 13 for Circuit and District Court cases, where 1 to 11 indicate 1st to 11th circuit, 12 represents the D.C. circuit, and 13 represents the Federal Circuit. For the 'Circuit' cases, *circuitNo* indicates in which Circuit Court the case was held. For the 'District' cases, *circuitNo* indicates the jurisdiction of the District Court that the case was held. *inCites* is the number of citations to a case in the larger data set. *outCites* is the number of outgoing citations from the case in the larger data set. *cite* is the citation of the case. *name* is the name of the case usually indicating the two involved parties, such as 'Penn. Cent. Transp. Co. v. City of New York.'

The semantic substrate in Figure 4 has three regions, each using a value of the *venue* attribute. The location of the regions from top to bottom is also in line with the hierarchical system of courts in the United States, where the Supreme Court has the most power, followed by the Circuit Courts and then District Courts. This way the link directions also indicate the hierarchy of the source and target cases, where upward indicate higher and downward indicate lower hierarchy in the court system. *year* is used along the *x*-axis of all regions consistently. This is achieved by using the same parameters (minimum and maximum values, and the number of bins) for the *x*-axis when designing the substrate. The same is true for the *y*-axis of the Circuit and District regions, where the *circuitNo* attribute is used.

Our domain experts more or less expected to find that by using the *circuitNo* attribute for placement, the tendency to cite within a circuit (both within Circuit and District Court cases) is shown (see Figure 4). This tendency is better perceived when link filters are used to look at subsets of links at a time quickly and consecutively on the Circuit region (i.e. users limit outgoing links on the Circuit region by *year* to a few years and drag the double-slider from left to right to inspect consecutive ranges). What our domain experts found interesting were the diversions from the general tendency, which can be isolated using link filters and investigated for further analysis (see Figure 5).

Every region has associated link filters for each placement attribute used. Since the 'District' region uses attributes *year* and *circuitNo* to determine node placement, there is a filter for the *year* attribute, and another filter for the *circuitNo* attribute (the second and third filters from the top on the right hand side in Figure 5, respectively). The filters work conjunctively (rather than disjunctively). As a result, the more filters applied on a region, the more links are restricted. The filters restrict links either to incoming or outgoing links. In Figure 5, links are restricted to outgoing links. To make a filter restrict to incoming links, users check the 'in?' checkbox that belongs to that filter (at the far right).

Sometimes during exploratory tasks, users can adjust filters to produce interesting results. Initially, users get a sense of looking at the unfiltered data; then, they try one filter, usually narrow it down and sweep it from one end



REGIONS

52 ▨ □ Supreme
181 □ □ Circuit
123 ▨ □ District

LINKS

328 ■ □ Supreme to Supreme
25 ■ □ Supreme to Circuit
20 ■ □ Supreme to District

775 ▨ □ Circuit to Supreme
757 ▨ ☑ Circuit to Circuit
55 ■ □ Circuit to District

395 ■ □ District to Supreme
240 ■ □ District to Circuit
96 ▨ ☑ District to District

RANGES

□ Supreme year □ in?
1978 -- 2005

□ Circuit year □ in?
1978 -- 2005

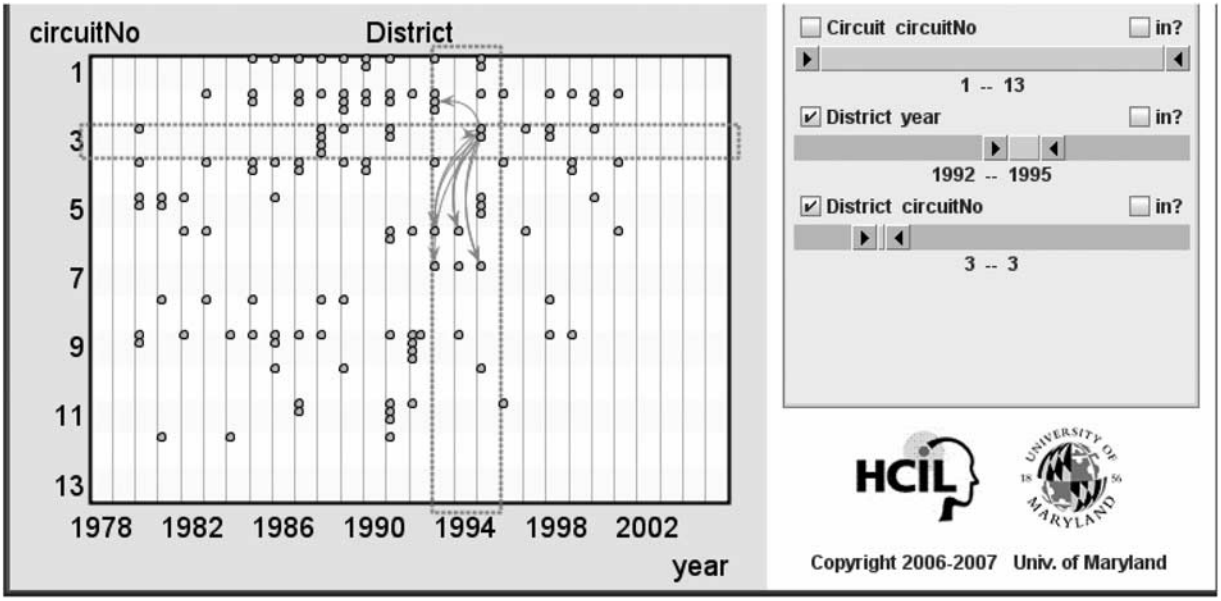
□ Circuit circuitNo □ in?
1 -- 13

□ District year □ in?
1978 -- 2005

□ District circuitNo □ in?
1 -- 13

Copyright 2006-2007 Univ. of Maryland

Figure 4 An initial semantic substrate is applied to a court case data set, where nodes are court cases and links are citations from one case to another. Nodes are grouped into regions using the *venue* node attribute with 'Supreme', 'Circuit', and 'District' values, while they are placed using *year* along the *x*-axis and *circuitNo* along the *y*-axis (except the Supreme Court cases), indicating the hierarchy of court cases in the legal system. Enabling links within Circuit and District regions shows a tendency for courts to cite within their circuit.



□ Circuit circuitNo □ in?
1 -- 13
☑ District year □ in?
1992 -- 1995
☑ District circuitNo □ in?
3 -- 3

Figure 5 Upon seeing the tendency that court cases tend to cite within their circuit in Figure 4, a diversion from this tendency is isolated using link filters on the District region, which helps users clearly see them.

to the other end of the range (could be done in a few seconds by dragging the double-slider from the middle). Then, depending on the visual feedback, users either can expand the range or activate another filter and do a similar procedure to arrive at an interesting result.

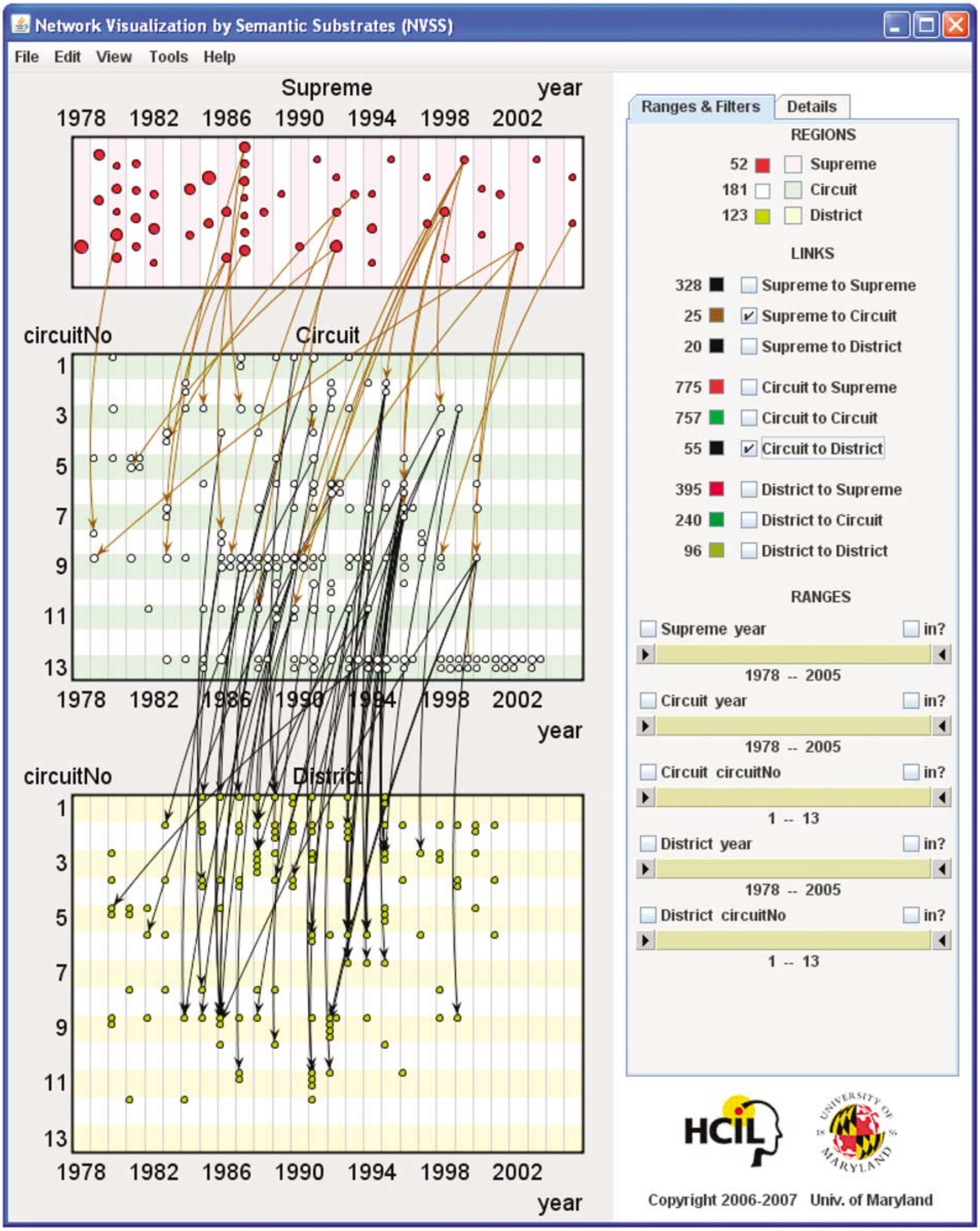
By switching the visibility of links to 'Supreme to Circuit' and 'Circuit to District', the design of the substrate reveals the citation patterns with respect to the 'year' attribute (Figure 6).

Since the *year* attribute is used along the *x*-axis of all three regions consistently, visual comparisons in terms of *year* are facilitated across regions. In Figure 6, citations from 'Circuit' to 'District' tend to follow immediately after (almost parallel citations), while citations from 'Supreme' to 'Circuit' are more diverse (not nearly as parallel, rather spread out over time). This might give insight into the nature of citations. Circuit Courts seem to follow up cases that are appealed promptly, while it takes a while for the Supreme Court to do so. A reason might be that the Supreme Court's decision-making process takes more time. Looking into Figure 6, one critical question that comes to mind is how cases of Circuit Court and District Court cite one another in terms of their circuit. It is hard to tell whether the citations from 'Circuit' to 'District' tend to be within the same circuit or not. To perceive this easily, we used a different substrate on the same data for our users.

To satisfy further requests from our domain experts, we opened the initial substrate, swapped the attributes for the *x*- and *y*-axis for the Circuit and District regions, and saved the edited substrate as a new substrate. Then, we applied it to the data to see Figure 7. With this modified substrate, the same data is viewed from a different point of view that favors comparisons in terms of the *circuitNo* attribute across the Circuit and District regions.

Figure 7 reveals that many citations from Circuit to District are within the same circuit although there are quite a few cross citations outside their circuits (which happens to be with District Courts of Circuits 1, 2, 3, 5, and 7. This becomes clearly visible with a sweep of incoming links filter on the District region). At the same time, the Supreme Court seems to cite various circuits with no particular attention to a few. It was interesting for our domain experts to see that there are Circuit Court decisions that cite District Court decisions in a different circuit. They noted that as worthy of further investigation. They were also curious to see whether Supreme Court citations have a different pattern when they cite Circuit and District Court cases. However, it is hard to perceive this using this substrate. Enabling only those links and coloring them distinctively helps; however, modifying (and reapplying) the substrate produced a much better display. We opened the substrate to edit, moved the Supreme region so that it is in the middle of the Circuit and District regions. This helped our domain experts to compare the citation half-life of Supreme versus District and Circuit Courts. We saved the modified substrate and applied it to the same data (Figure 8). Using this substrate, citations from Supreme Court to Circuit and District Court cases are easily comparable. Our domain experts saw no dramatic differences in the citation patterns.

Our domain experts were intrigued by the fact that there are Supreme Court decisions that cite several District Court decisions at once. They found this quite unexpected and worthy of further exploration. This pattern of citations from the Supreme Court to multiple Circuit



Ranges & Filters | Details

REGIONS

52 🟥 □ Supreme
181 □ □ Circuit
123 🟨 □ District

LINKS

328 ■ □ Supreme to Supreme
25 ■ ☑ Supreme to Circuit
20 ■ □ Supreme to District

775 🟥 □ Circuit to Supreme
757 🟩 □ Circuit to Circuit
55 ■ ☑ Circuit to District

395 🟥 □ District to Supreme
240 🟩 □ District to Circuit
96 🟩 □ District to District

RANGES

□ Supreme year □ in?
1978 -- 2005

□ Circuit year □ in?
1978 -- 2005

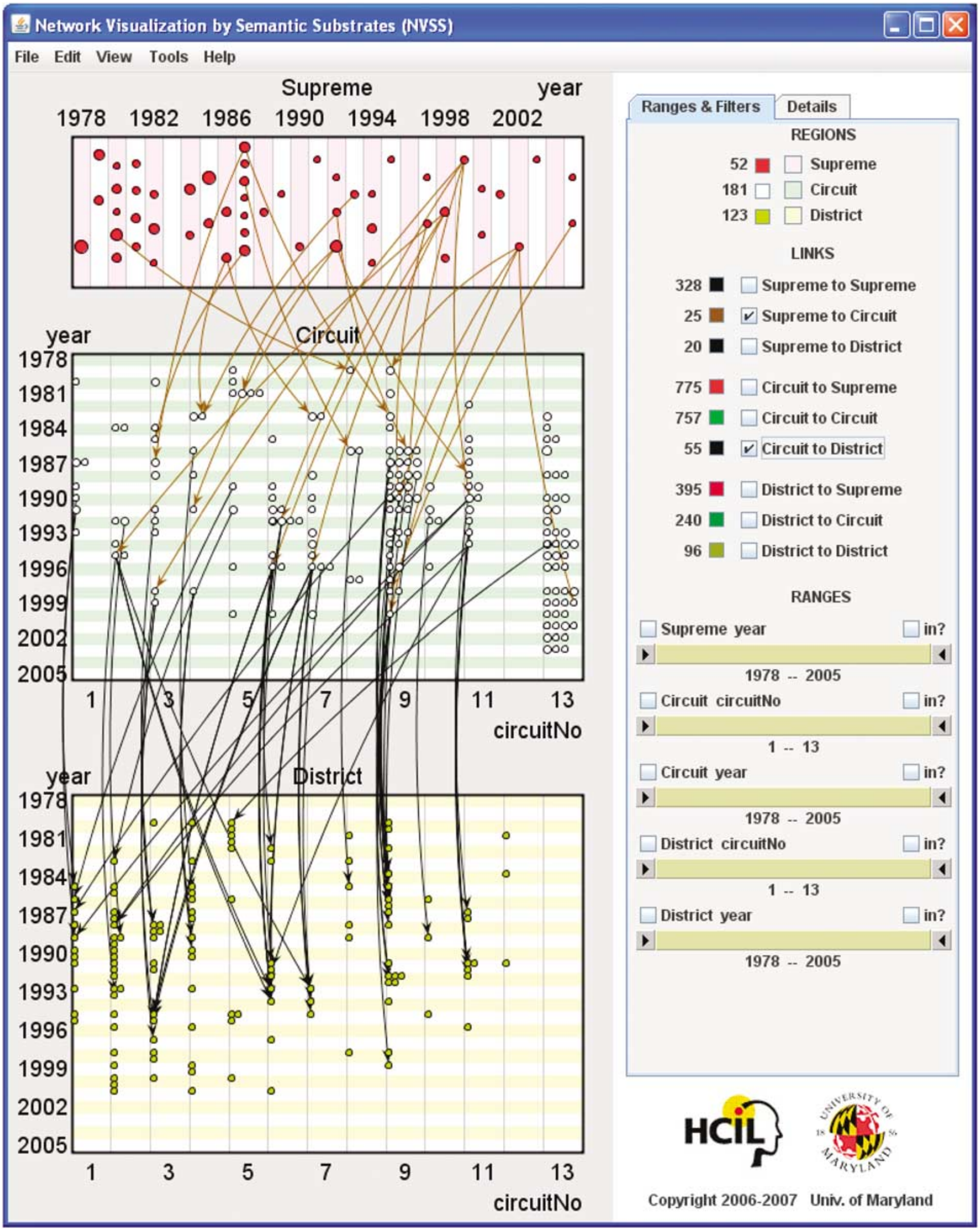
□ Circuit circuitNo □ in?
1 -- 13

□ District year □ in?
1978 -- 2005

□ District circuitNo □ in?
1 -- 13

Copyright 2006-2007 Univ. of Maryland

Figure 6 Due to having used the *year* attribute across regions consistently on the *x*-axis, it is easy to compare the citation patterns according to *year* across regions. The citation patterns indicate that although Circuit Courts tend to follow-up immediately after a case is appealed, it takes a longer time to do so for the Supreme Court, possibly due to their lengthy appeals process.



Ranges & Filters | Details

REGIONS

52 🟥 □ Supreme
181 □ □ Circuit
123 🟩 □ District

LINKS

328 ■ □ Supreme to Supreme
25 ■ ☑ Supreme to Circuit
20 ■ □ Supreme to District

775 🟥 □ Circuit to Supreme
757 🟩 □ Circuit to Circuit
55 ■ ☑ Circuit to District

395 🟥 □ District to Supreme
240 🟩 □ District to Circuit
96 🟩 □ District to District

RANGES

□ Supreme year □ in?
1978 -- 2005

□ Circuit circuitNo □ in?
1 -- 13

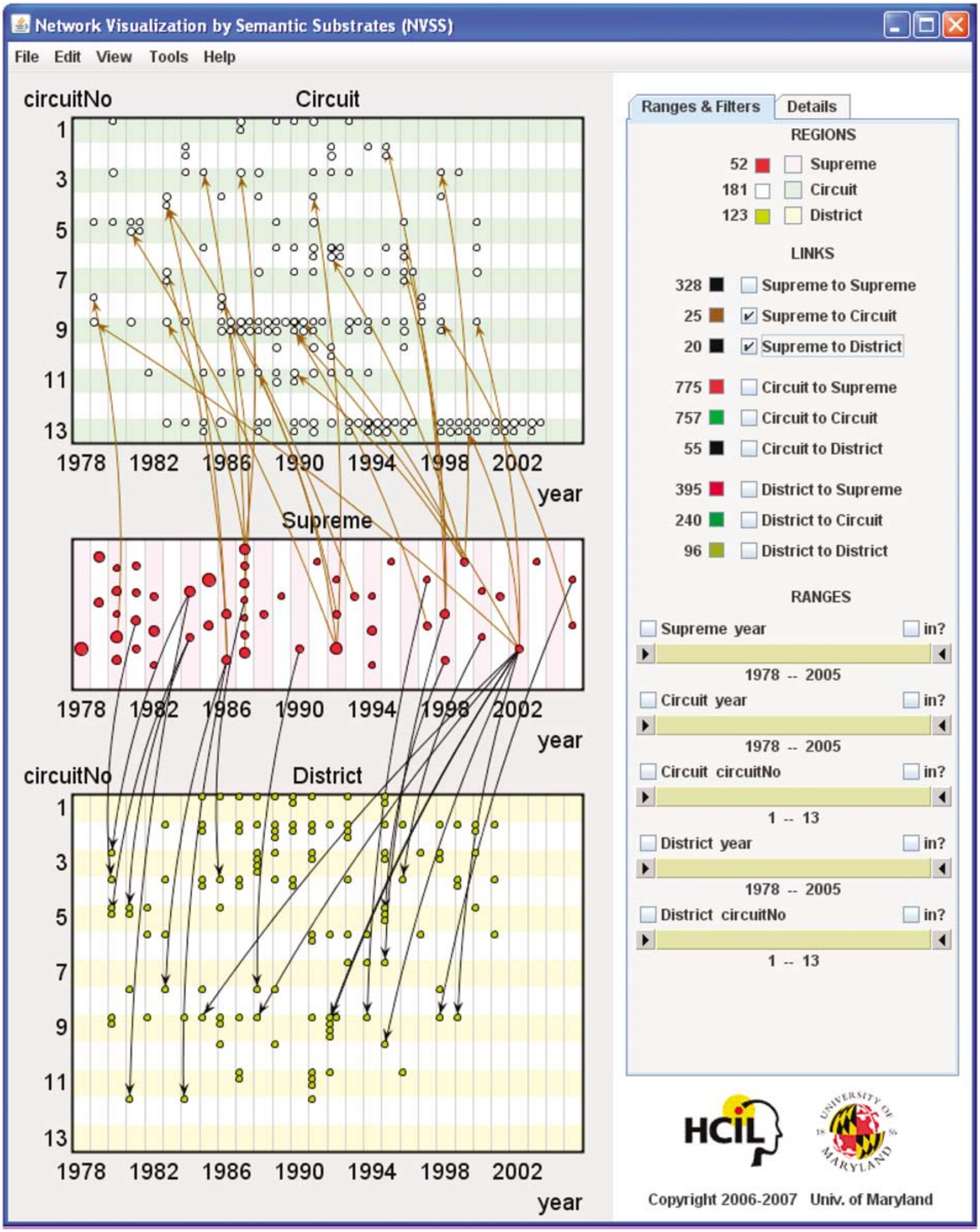
□ Circuit year □ in?
1978 -- 2005

□ District circuitNo □ in?
1 -- 13

□ District year □ in?
1978 -- 2005

Copyright 2006-2007 Univ. of Maryland

Figure 7 Looking at the same data using a modified substrate (the substrate in Figure 6 with swapped axes for Circuit and District regions) that aligns circuits using *circuitNo* along the *x*-axis of Circuit and District Court regions helps comparison between cases from these regions in terms of their circuits. Most citations from the Circuit Courts to the District Courts are within the same circuit with a few exceptions from the 1st, 2nd, 3rd, 5th, and 7th Circuit Courts.



Network Visualization by Semantic Substrates (NVSS)

REGIONS

52 🟥 □ Supreme
181 □ □ Circuit
123 🟩 □ District

LINKS

328 ■ □ Supreme to Supreme
25 ■ ☑ Supreme to Circuit
20 ■ ☑ Supreme to District

775 🟥 □ Circuit to Supreme
757 🟩 □ Circuit to Circuit
55 ■ □ Circuit to District

395 🟥 □ District to Supreme
240 🟩 □ District to Circuit
96 🟩 □ District to District

RANGES

□ Supreme year □ in?
1978 -- 2005

□ Circuit year □ in?
1978 -- 2005

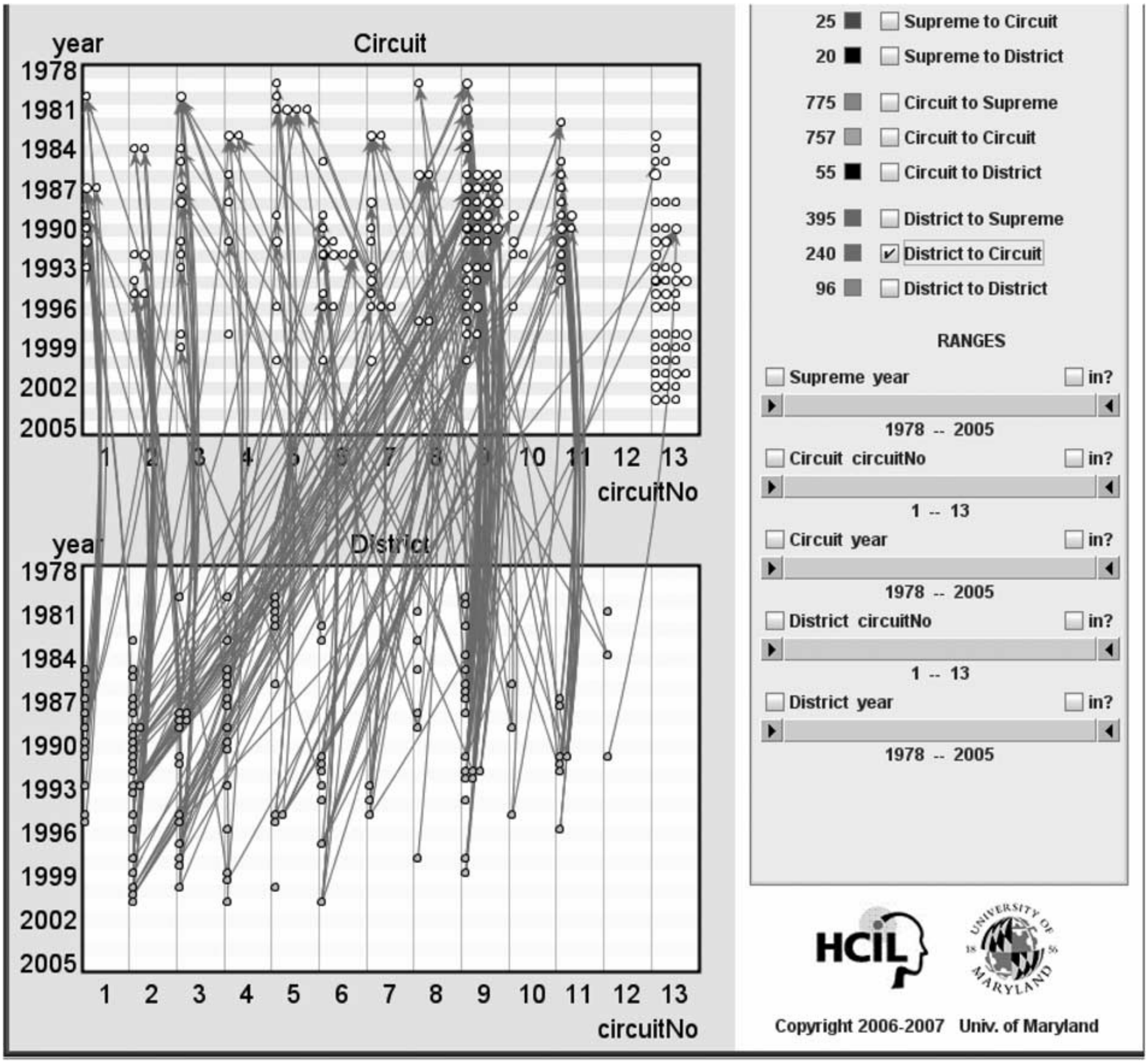
□ Circuit circuitNo □ in?
1 -- 13

□ District year □ in?
1978 -- 2005

□ District circuitNo □ in?
1 -- 13

Copyright 2006-2007 Univ. of Maryland

Figure 8 The substrate in Figure 6 modified to place the Supreme region between the other regions. The pattern of Supreme Court citations to Circuit and District Courts appears to be similar.



25 ■ □ Supreme to Circuit
20 ■ □ Supreme to District

775 ■ □ Circuit to Supreme
757 ■ □ Circuit to Circuit
55 ■ □ Circuit to District

395 ■ □ District to Supreme
240 ■ ☑ District to Circuit
96 ■ □ District to District

RANGES

□ Supreme year □ in?
1978 -- 2005

□ Circuit circuitNo □ in?
1 -- 13

□ Circuit year □ in?
1978 -- 2005

□ District circuitNo □ in?
1 -- 13

□ District year □ in?
1978 -- 2005

Copyright 2006-2007 Univ. of Maryland

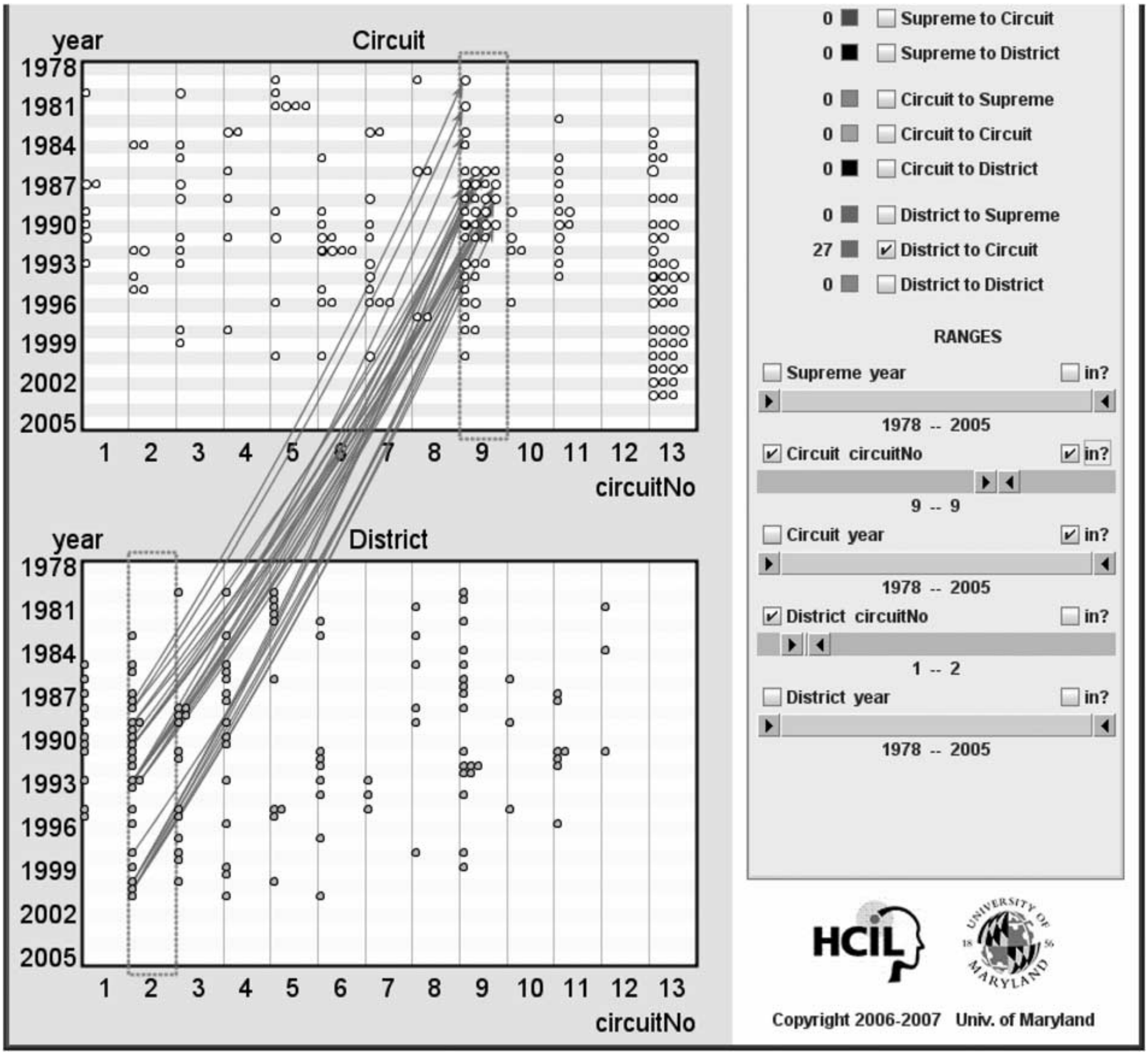
Figure 9 Applying the substrate from Figure 7 reveals District to Circuit citations here many parallel citations with a major exception of frequent citations from the 2nd to the 9th circuit.

Court cases is interesting since it shows how Circuit Court cases can set precedents that influence even the Supreme Court. This interesting circumstances was deemed worthy of further research by our domain experts.

Our domain experts also wanted to explore the citations from the District region to the Circuit region. To help with this task, we loaded the previous substrate, applied the data to generate Figure 9. Enabling 'District to Circuit' links reveals many citations, the majority of which appear to be parallel. A major exception seems to be between the District Court cases in the Second Circuit and the Circuit Court cases of the Ninth Circuit. The Ninth circuit also seems to receive many citations from the district courts in the Ninth Circuit.

By isolating links from the district courts in the Second Circuit to the Ninth Circuit (using an outgoing links filter on the District region, an incoming links filter on the Circuit region, and by restricting values to the desired range), most of the citations appear to be concentrated in three periods, namely 1989, 1993, and 2000 (Figure 10). Our domain experts did not expect to see Circuit Court citations to District Court decisions outside their circuit. Circuit Courts are more authoritative and therefore are not expected to cite other Districts. What might be happening in this situation is that the District Courts in the 2nd circuit may have specialized in a particular topic that the Circuit Courts found worthy of citing. Our domain experts mentioned that they might look into those decisions to find out whether this is the case and if so which topics these are.

As our domain experts became familiar with the Substrate Designer's features, the semantic substrates



0 ■ □ Supreme to Circuit
0 ■ □ Supreme to District
0 ■ □ Circuit to Supreme
0 ■ □ Circuit to Circuit
0 ■ □ Circuit to District
0 ■ □ District to Supreme
27 ■ ☑ District to Circuit
0 ■ □ District to District

RANGES

□ Supreme year □ in?
1978 -- 2005
☑ Circuit circuitNo ☑ in?
9 -- 9
□ Circuit year ☑ in?
1978 -- 2005
☑ District circuitNo □ in?
1 -- 2
□ District year □ in?
1978 -- 2005

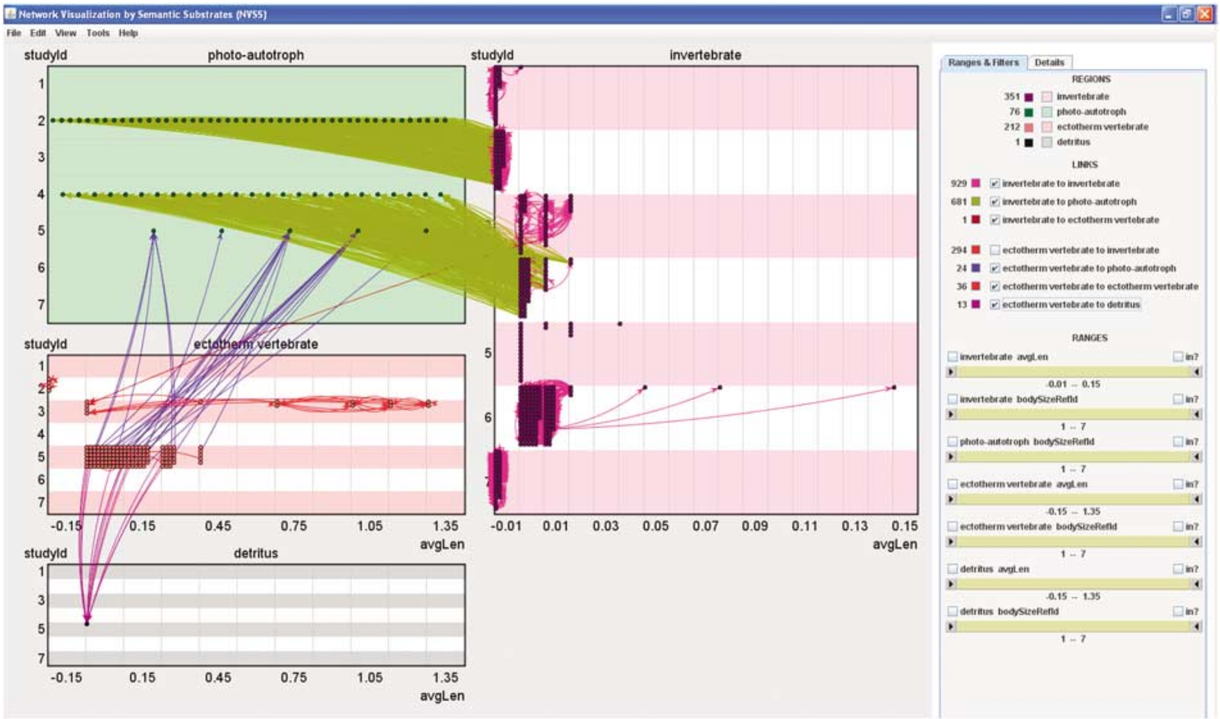
Copyright 2006-2007 Univ. of Maryland

Figure 10 Restricting links in Figure 9 between the 2nd and 9th circuits using link filters reveals that citations are concentrated in three periods, namely 1989, 1993, and 2000.

became a new language of discourse for them, enabling them to generate many new hypotheses. The visual presentation and user control of link visibility supported discussion, exploration, and communication within and beyond the group. Our domain experts mentioned that NVSS is useful for them to overview the data quickly to find out interesting phenomena and narrow down to the cases to investigate. This would allow them also to read those cases in a targeted way to answer the questions they formed while exploring the data set. Overall, our domain experts found NVSS useful because it enabled them to look at the temporal (*year*) and circuit (*circuitNo*) dimensions at the same time, which they found comprehensible as opposed to looking at a spreadsheet. Our domain experts have captured states of their exploration via images and used those to communicate ideas within their group and with their colleagues. They are planning to further explore their data using NVSS and complement it with other methods (reading cases and statistical measures) to finally produce results to be published in academic research venues in their field (conferences, journals, etc.).

Food web data

Biologists study predator–prey networks, which are called food webs. Our collaborator, Dr. Cynthia Parr, is a biologist and researcher associated with the Human–Computer Interaction Lab. She was interested in exploring a food web data set (seven aquatic webs from Brose *et al.*[26]) and we designed our substrates to facilitate or improve the



REGIONS

351 🟪 □ invertebrate
76 🟩 □ photo-autotroph
212 🟥 □ ectotherm vertebrate
1 ■ □ detritus

LINKS

929 🟪 ☑ invertebrate to invertebrate
681 🟩 ☑ invertebrate to photo-autotroph
1 🟥 ☑ invertebrate to ectotherm vertebrate

294 🟥 □ ectotherm vertebrate to invertebrate
24 🟪 ☑ ectotherm vertebrate to photo-autotroph
36 🟥 ☑ ectotherm vertebrate to ectotherm vertebrate
13 🟪 ☑ ectotherm vertebrate to detritus

RANGES

□ invertebrate avgLen □ in?
-0.01 – 0.15
□ invertebrate bodySizeRefId □ in?
1 – 7
□ photo-autotroph bodySizeRefId □ in?
1 – 7
□ ectotherm vertebrate avgLen □ in?
-0.15 – 1.35
□ ectotherm vertebrate bodySizeRefId □ in?
1 – 7
□ detritus avgLen □ in?
-0.15 – 1.35
□ detritus bodySizeRefId □ in?
1 – 7

Figure 11 Using semantic substrates with food web data sets. Displaying data from seven studies with length (in meters) on the *x*-axis for all except photo-autotroph. Negative values indicate missing or unknown attribute values.

process of understanding (finding facts or insights) of her data. She was our domain expert for this data set, while we were the tool experts. Her results were arrived at after five sessions over 6 weeks, each lasting 45–60 min. In the first two sessions, we discussed how to compile the data and the data characteristics. In the latter 2–3 sessions, we looked at the data in NVSS together and she gave us her feedback. We also communicated with our domain expert via email to discuss specific aspects of the data and its presentation.

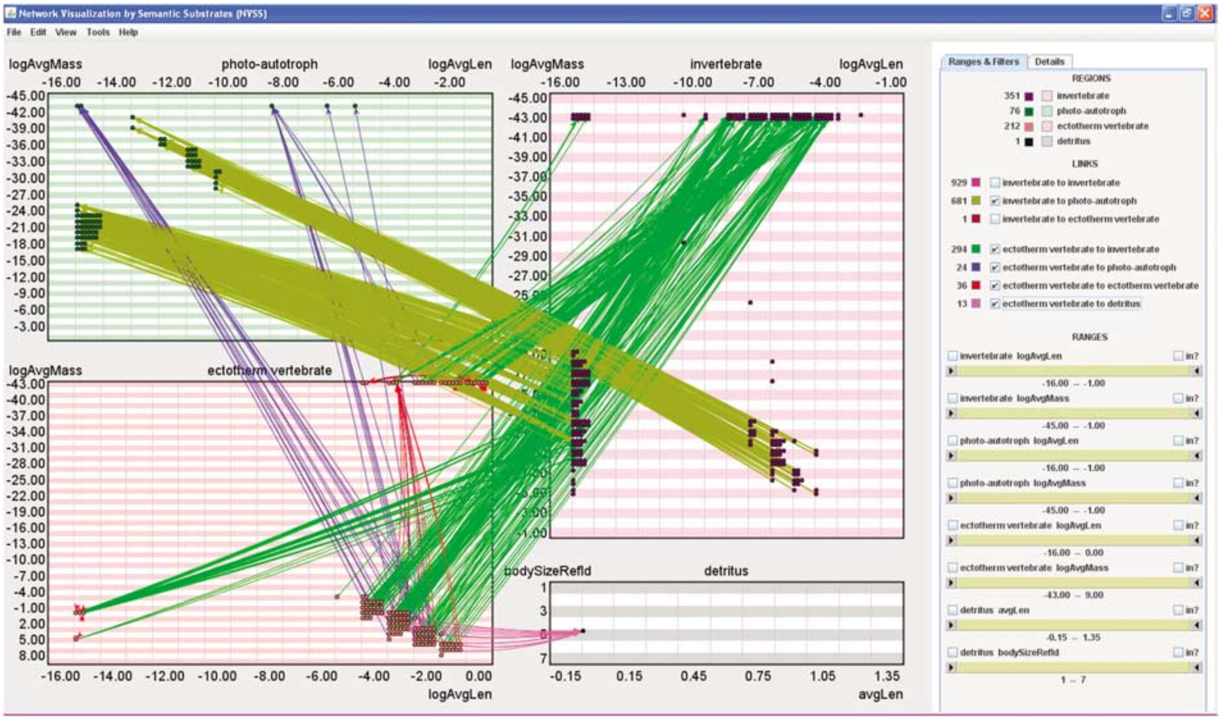
Communication with our domain expert about the data set led to our initial substrate. Then, 3–4 iterations were needed to arrive at a satisfying initial substrate. These iterations were guided by our domain expert's comments. We quickly arrived at the grouping attribute (metabolic category), however, the placement attributes took several iterations because of our joint lack of knowledge about the data distribution. After the first substrate, it took two iterations to arrive at the second substrate. This time, we used design-b*y*-example. We reused the first substrate and modified it to arrive at the second one, a much faster process. As in the other data set, we do not count iterations resulting from minor adjustments, NVSS software updates, etc.

In this food web example, nodes are taxa (species or higher level classifications for living entities) and links are predator to prey (also called 'consumer' and 'resource', respectively) (Figure 11). The data set combines results from seven studies of aquatic food webs. Studies do not have links across each other because each is self-contained to a certain place and time. Some of the available node attributes in this data set are *avgLen* (average length of the taxon in meters), *avgMass* (average mass of the taxon in grams), *studyId* (the study that the taxon was observed in; ranges from 1–7), and *metCat* (metabolic category of the taxon, which has values 'invertebrate', 'photo-autotroph', 'ectotherm-vertebrate', and 'detritus' in this data set).

The data set consists of a total of 640 nodes and 1978 links.

With our domain expert, we made a series of design choices for this semantic substrate. The metabolic category was selected to group nodes into regions. This attribute determines what type of living entity the taxon is in terms of its metabolism.

Photo-autotrophs, such as *Peridinium cinctum* and Din*obryon bavaricum*, are usually very small in length and mass. The average mass in photo-autotrophs ranges from 3.57e–018 to 9.46e–008 g while the average length ranges from 0 to 0.1 m (most values are small ranging from 4e–006 to 9.8e–005 m; the rest are 0, 0.0001, 0.0005, 0.005, and 0.01 m). Since the range of these attributes is so small and hard to analyze, they were not used for placement for this first substrate. Instead, the *studyId* was used to organize the nodes along the *y*-axis. In fact, for consistency, *studyId* is used along the *y*-axis in all regions. With the educated guess of our collaborator, we assumed



Ranges & Filters | Details

REGIONS
351 ■ □ invertebrate
76 ■ □ photo-autotroph
212 ■ □ ectotherm vertebrate
1 ■ □ detritus

LINKS
929 ■ □ invertebrate to invertebrate
681 ■ ☑ invertebrate to photo-autotroph
1 ■ □ invertebrate to ectotherm vertebrate

294 ■ ☑ ectotherm vertebrate to invertebrate
24 ■ ☑ ectotherm vertebrate to photo-autotroph
36 ■ ☑ ectotherm vertebrate to ectotherm vertebrate
13 ■ ☑ ectotherm vertebrate to detritus

RANGES
□ invertebrate logAvgLen □ in?
-16.00 -- -1.00
□ invertebrate logAvgMass □ in?
-45.00 -- -1.00
□ photo-autotroph logAvgLen □ in?
-16.00 -- -1.00
□ photo-autotroph logAvgMass □ in?
-45.00 -- -1.00
□ ectotherm vertebrate logAvgLen □ in?
-16.00 -- 0.00
□ ectotherm vertebrate logAvgMass □ in?
-43.00 -- 9.00
□ detritus avgLen □ in?
-0.15 -- 1.35
□ detritus bodySizeRefId □ in?
1 -- 7

Figure 12 Using a different semantic substrate with the same food web data set as in Figure 11. Displaying combined data from seven studies with log(length in meters) on the x-axis and log(mass in grams) on the y-axis. Missing/unknown mass is denoted by -43, while length is denoted by -15.

that the *avgLen* attribute would be a pretty good indicator of how large a taxon is. Hence, *avgLen* was used along the x-axis for all regions except photo-autotrophs. Negative length indicates unknown length measurement for that taxon, that is, missing data.

In general, the most striking conclusion is that the seven data sets differ considerably in the metabolic categories of organisms they sampled, and hence the kinds of links that were possible. However, some patterns relating to size are possible to discern.

Most of the invertebrates are very small animals. Looking at the invertebrate region, study 6 reveals that some longer invertebrates are prey of much shorter ones, for example *Sigara nigrolineata* is prey of *Agabus bipustulatus* (Coleoptera). Invertebrates are also prey of ectotherm-vertebrates, such as *Daphnia rosea* (water flea) is prey of *Salmo trutta* (brown trout) (294 links, unchecked not to visualize the links). Only in study 5, invertebrates do not consume invertebrates.

It appears that photo-autotrophs are sole producers and are only in studies 2, 4, and 5. In studies 2 and 4, photo-autotrophs are heavily consumed by invertebrates, while in study 5 they are solely consumed by ectotherm-vertebrates of relatively shorter taxa. Only one study included detritus, which is solely consumed by ectotherm-vertebrates as well (by relatively shorter ones only from study 5).

It appears invertebrates never consume ectotherm-vertebrates with one exception in study 3. The prey, in this case, happens to be one of the shortest ectotherm-vertebrates, which is reasonable when considering that most ectotherm-vertebrates are much larger than invertebrates.

To gain further understanding, we used a different substrate to visualize the same data set. We hoped that this different point of view would help to attain new insights and understandings (Figure 12). In this substrate, all regions except detritus use log(*avgLen*) on the x-axis, while they use log(*avgMass*) on the y-axis. Length increases from left to right, while mass increases from top to bottom.

Combining the data from all studies, we helped our collaborator see general tendencies in terms of mass and length. This would also provide evidence to support the earlier hypothesis that mass and length are usually proportionate to each other. Shorter and lighter photo-autotrophs are consumed merely by the heaviest and mostly longer invertebrates, while heavier photo-autotrophs are consumed by mostly not-so-heavy invertebrates. Ectotherm-vertebrates (of known length and mass) consume various length (but unknown mass) of photo-autotrophs and detritus; while mostly the heavier and longer ectotherm-vertebrates eat others in their own metabolic category.

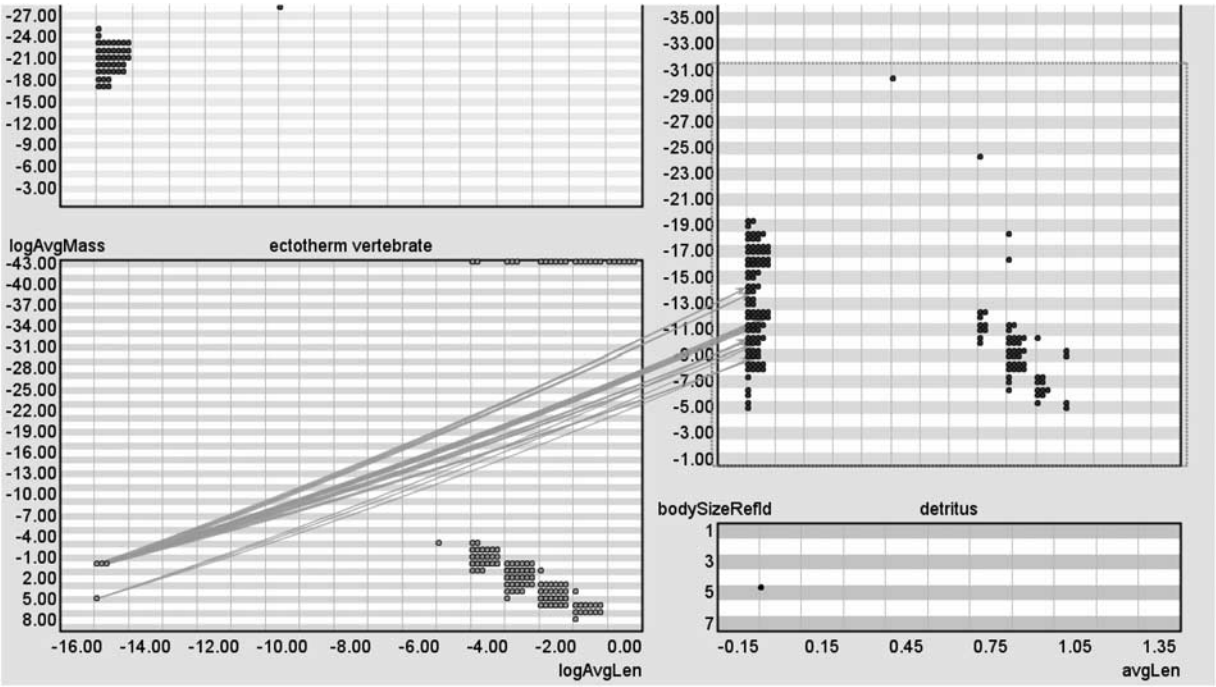


Figure 13 Using an incoming link filter on mass on the invertebrate region shows that among the known-mass invertebrates, the ones that are eaten are those that have a medium weight. In other words, the lightest and the heaviest invertebrates are not consumed.

Looking at the relationship between ectotherm-vertebrates and invertebrates, the ectotherm-vertebrates are always consumers and they tend to consume medium-weight invertebrates rather than light or heavy invertebrates. One can vaguely perceive this by the slope of the links (the ones originating from the bottom left of the ectotherm-vertebrate region); however, using an incoming link filter on mass on the invertebrate region, one can clearly see that this is the case (Figure 13).

From these two views on the food web data set, at one glance, with a little bit of focus on each area on the display, almost all interactions within the data set of seven studies are interpretable in terms of study and length, and then mass and length. Although axes have been used consistently to have the same attributes in each substrate, this is not a restriction and different attributes could be used in general. Filters help focus on areas to reveal relationships more clearly.

The semantic substrates enabled our domain expert to understand her data set better, especially to recognize what other data would be needed for her desired analyses. She also realized that the seven data sets were not comparable with each other, a fact unknown to her before looking into the data in NVSS. She realized some patterns that merit further investigation, such as why ectotherm-vertebrates might avoid the lightest and heaviest invertebrates.

The highly skewed distributions and partial nature of the data presented challenges, but these features stood out clearly in our visualizations, supporting the process of discovery. These insights would not have emerged from a simply defined force-directed layout of nodes, because skewed distributions of attribute values and missing data would not be visible.

Our domain expert found NVSS useful to explore her food-web data and envisions using NVSS to continue food web analysis work. She would use it to compare relationships and attribute patterns of real food webs with patterns of simulated food webs. This would help her refine the models used to make them more realistic.

Semantic substrate design guidelines

As a result of our experience of designing semantic substrates for these two domain expert groups and others we began to develop design guidelines. They are more or less in priority order and aim to provide efficient and effective exploration of network data using semantic substrates:

(1) Choose grouping and placement attributes based on key variables of nodes.
(2) Favor attributes with uniform distributions to spread out nodes evenly. Transform attributes by using log(X) or sqrt(X), if necessary, to make their distribution more uniform.
(3) Minimize or eliminate gaps (by choosing or transforming attribute values) and avoid outliers (possibly

by deleting them or setting a maximum value) to save screen space.

(4) Align regions to facilitate comparison.

(5) Locate regions to minimize link length and link overlaps while facilitating comparison (by aligning regions to be compared and placing them close to each other).

Semantic substrate design guidelines can be applied to selecting attribute values to group nodes into regions, determining the placement method for nodes within region, and other smaller but still significant issues. The following subsections illustrate how the design guidelines are applied.

Selecting a grouping attribute

It is best to choose the attribute that is of most interest (1) and most suitable (2), (3) for grouping. If there are many attributes of interest and their levels of interest are not very different from each other, data set characteristics determine how easy it is to choose an attribute (or attributes) for grouping. Sometimes, there is a best attribute to choose for grouping. Usually an attribute with 2–5 values that separates nodes into meaningful categories is appropriate (1) as was the case with the *venue* attribute with the legal cases data set and the *metCat* attribute (metabolic category) with the food web data set.

If users have an idea of what the best attribute is, they may use it to see if it produces the desired understanding. If not, they can choose another attribute and iterate.

When there is no attribute with 2–5 values, users may create a derived attribute that will have 2–5 values (2). Effective grouping (binning) of attribute values is almost always possible.

Knowledge of what attributes are available in the data set, their types and range of values helps when choosing a grouping attribute, while knowledge of frequency and distribution of attribute values help to create a substrate containing regions with a balanced number of nodes within each region (2). However, exceptions do not violate the rule as in the detritus region with a single node in Figure 11, which was useful to reveal incoming links. Users who are knowledgeable in terms of these aspects will have an advantage. Otherwise, they can accumulate this type of knowledge by iterative design and application of substrates to their data. Another way is to assist users by making this type of knowledge available in the Substrate Designer (see future work section).

The selection of a grouping attribute value for a region eliminates it from the pool of attributes available to determine the placement method for that region. As a result, users may take into account what attributes to choose for placement later and accordingly not choose those attributes as grouping attributes.

In some rare circumstances, users may want to select values of different attributes for each region; however, they need to make sure that nodes fall into a unique region and attribute values for grouping together cover all nodes in the data set (or they can create a subset having only those nodes that they cover).

To summarize, selecting an effective grouping attribute is usually best with an attribute having 2–5 values that divides the data set into meaningful subgroups or categories. Users will need to know what attributes are available in the data set, their type and their meaning. They are likely to make better choices and have fewer substrate design iterations if they have a good idea of the frequency and distribution of nodes in terms of various attributes in the data set.

Determining the placement method

Determining the placement method involves selecting a placement algorithm and providing attributes as parameters.

It is best to first determine which attributes to use for placement. Attributes of high interest should be given priority (1). The placement algorithm should be selected according to the characteristics of the chosen attribute (2), (3), (4).

GridPlotXY is suitable whenever there are two meaningful attributes to choose for placement. For the legal cases data set, *year* and *circuitNo* are meaningful (1) as *year* helps make temporal inferences, while *circuitNo* subcategorizes cases in addition to refining the hierarchy of courts and enabling comparison between Circuit and District Court cases (4), (5) . A fairly balanced distribution of nodes across these attributes helps the visualization as in the legal cases data set (Figures 4–10) (2). Outliers may pose a challenge as in the invertebrate region in Figure 11 due to unused space (3). Nevertheless, it is still possible to get an idea of the distribution of nodes in terms of this attribute (as it is useful to see how invertebrate taxon sizes compare across studies) and compare relationships with other regions that use the same attribute (4) (as it is revealing to see that smaller ectotherm-vertebrate consume photo-autotrophs).

SingleAxisGridPlot algorithms (GridPlotX, GridPlotX Jittered, GridPlotY, and GridPlotY Jittered) are appropriate when there is not a meaningful or a useful second attribute to place the nodes by. Another reason not to use a second attribute is to have a good spread of nodes on the display (2), (3) (as GridPlotXY may cause too many nodes to fall into a cell causing them to overlap, as in ectotherm-vertebrate in Figure 11; a good spread is achieved with photo-autotrophs along the *x*-axis with the bad alternative of overlapped nodes on the far left if the same *x*-axis was used as the ectotherm-vertebrate or even invertebrate region).

In general, it is best if the values of selected attributes have a uniform distribution (2) across the selected range. Although this is ideal, it is not necessary to gain insights. There are uniform distributions in the legal cases data set but non-uniform ones in the food web data set. For instance, photo-autotrophs are not distributed in a balanced way across studies in Figure 11. In fact, studies

$1, 3, 6,$ and 7 have no nodes at all. Still, the lack of nodes in those studies conveys useful information. The cost is unused space; however, the advantage is that the standardization in terms of study facilitates comparison between regions. For attributes that have non-uniform distributions, users also have the option of creating derived attributes (2) (using external tools) that have more uniform distributions by applying transformations and then use those derived attributes. Aris *et al.*[27] discuss several options for transformations.

When selecting an attribute for the chosen algorithm, challenges similar to selecting a grouping attribute arise. In other words, users need to know what attributes are available, their type and range of values. Knowledge of their distribution and frequency helps; however, users can find out this information by iterative design and application of substrates or the application can present this information to users (see future work section).

Miscellaneous issues

Size and alignment of regions can facilitate comparison of nodes in terms of attributes. Using a common axis (x- or y-) across regions and aligning them on that axis are effective (4). When there are many alignment possibilities, users must choose. Locating regions to decrease (and in certain situations increase) link length and link overlap will increase the visualization's effectiveness (5) . If users have specific questions, they can set the attributes of interest as parameters of the placement methods and align the regions of interest (1). If users are exploring the data set, they can iteratively refine the design of their substrate until desired insights are gained.

With respect to colors, choosing the link colors seem to be the most crucial issue, especially when there are many links on the display. It is best to choose contrasting colors (such as blue with purple and blue with red as in links associated with the *ectotherm-vertebrate* region in Figure 11). Node and background colors are less important but still significant. Lighter colors are better for the background.

Determining the size of nodes is another seemingly small issue but still significant. Additional information can be represented by tying node size to an attribute. Unbalanced distributions and outliers can decrease the effectiveness of size coding as well as make it hard for users to find a good transformation to apply to the attribute of interest. In such cases, it may help to create a derived attribute (using other applications) from the existing attributes and then use that attribute for size coding (2), (3). For example, for distributions with long tails a logarithmic transformation produces a more uniform distribution. In the event $\log(X)$ does not produce a uniform distribution, users could try other transformations such as sqrt. In fact, the transformation used for the legal cases data set was $5 + sqrt(X)/5$ on the 'incites' attribute (indicating the number of citations to a case in the larger data set of 2345 decisions), which produced good results, especially on the Supreme region.

Discussion and future work

User-defined semantic substrates appear to be effective in organizing data in meaningful ways. Insights gained from the data sets explored in this paper provide concrete examples of the usefulness of this approach. However, there is room for improvement. These fall into the following categories: (1) the visual presentation, (2) facilitating the substrate creation process, and (3) miscellaneous issues.

The visual presentation can be improved in terms of node and link representations. There are many situations where the number of nodes to be represented exceeds the limit of the available space. This happens especially with the more restrictive algorithms, such as GridPlotXY, where cell space is small and it becomes a challenge to display more than a few nodes. Either non-uniform distribution of nodes or simply that there are many nodes to visualize lead to node overlaps.

A way to handle this is to clump nodes into metanodes (also known as clusters; Tulip[16] clusters nodes using density functions of attributes and structural information such as node degree and segment length). This will help with the scalability by enabling display of larger data sets. In addition, it will help reduce the number of links (especially helpful for data sets with dense links).

There are several opportunities to improve link display. Links tend to overlap due to originating from or pointing to close nodes. An example is the links from the District region to the Circuit region in . Nodes with close attribute values are placed close to each other and those nodes are usually of interest. The trade-off is between understanding the data and perception of links. A specific form of link overlap is with links concentrated within a small space. This happens usually when the source and destination links are packed together in a small space as in GridPlotXY algorithm cells, such as in the invertebrate region in Figure 11. Either link routing[28], link clustering such as using hierarchy to organize edges,[29] or other methods might provide better visual representations.

The substrate creation process can be improved in terms of two criteria: (1) a good substrate at the end of the process, and (2) a faster process. A good substrate is one that helps users gain useful insights. Trial and error plus past experience are good ways to get started. Substrates can be stored and reused for similar data sets. A module that helps users store substrates, calculates compatibility between a substrate and a data set, and provides a score in terms of perceptual advantages might help users find a good substrate.

One way to accelerate the substrate creation process is to reduce the number of iterations. When users are deciding which attributes to use for grouping or placement methods within a region, providing the range or distribution of attribute values relevant to the context can eliminate several iterations. Previewing nodes within a region and links whenever possible might help. Opportunities include more expressive region specification (allowing the use of other operators than equality to a single attribute value, to the limit to support a Boolean expression with a complete set of operators), a way to select filters (especially if there are many possible ones), and more algorithms for node placement within regions. Other improvements would be facilities to help users with pre-processing tasks, such as narrowing down to an interesting subset (especially for large data sets) and creating derived attributes.

Application-level improvements would include scalability (both performance and visual representation as the number of nodes and links increase), additional filters for nodes and links, and perhaps other widgets for visual interactions.

Further future work includes allowing multiple valued attributes and supporting more than one type of node (e.g. bimodal networks). Non-square and overlapping regions might be helpful in some problems. Evaluation of semantic substrates in several domains by case studies is also needed.[30]

Conclusion

By engaging the remarkable human capabilities for spatial perception and analysis, we believe that semantic substrates will accelerate many network visualization tasks, enabling domain experts to make more frequent and important insights. Our two case studies demonstrated benefits for domain experts, but much more needs to be done to refine and extend our tools. The process of substrate design could benefit from tools, modules, or features that will both expedite the process and increase the effectiveness or suitability of the substrate. Automated and semi-automated substrate designs are likely to be tuned to the needs of specific domains, but these could easily be shared among many users. A meaningful substrate captures domain knowledge and enables easy comparison of data sets, identification of attribute value changes, and the detection of new nodes or links.

The two case studies in this paper showed how the process of iterative substrate design helps explore network data from different point of views, which resulted in fresh interpretations and outcomes. Another contribution is the proposed guidelines for designing semantic substrates. In the discussion and future work section, areas for improvement are addressed with possible solutions.

We believe that semantic substrates accompanied with good substrate design promise more effective exploration of networks through increased user control that leads to better understanding and deeper insights.

Acknowledgements

We appreciate the invaluable collaboration of Professor Wayne McIntosh (Department of Government & Politics, University of Maryland) and his students Ken Cousins and Stephen Simon. The U.S. National Science Foundation grant 'Inter-Court Relations in the American Legal System: Using New Technologies to Examine Communication of Precedent II' and Microsoft provided partial support. We also appreciate the invaluable collaboration of Cynthia Parr working with us on the food web data set.

References

1 Herman I, Melançon G, Scott Marshall M. Graph visualization and navigation in information visualization: a survey. *IEEE Transactions on Visualization and Computer Graphics* 2000; **6**: 24–43.

2 Ware C, Purchase H, Colpoys L, McGill M. Cognitive measurements of graph aesthetics. *Information Visualization* 2002; **1**: 103–110.

3 Sindre G, Gulla B, Jokstad H. Onion graphs: aesthetic and layout, *Proceedings of the 1993 IEEE Symposium on Visual Languages* 1993 (Bergen, Norway), IEEE Computer Society Press; Silver Spring, MD, 1993; 287–291.

4 Shneiderman B, Aris A. Network visualization by semantic substrates. *(Proceedings of IEEE Visualization/Information Visualization) IEEE Transactions on Visualization and Computer Graphics* 2006; **12**: 733–740.

5 Lee B, Plaisant C, Parr CS, Fekete J-D, Henry N. Task taxonomy for graph visualization, *Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization* 2006 (Venice, Italy), ACM: New York, NY, USA, 2006; 1–5.

6 Bilgic M, Licamele L, Getoor L, Shneiderman B. D-Dupe: an interactive too for entity resolution in social networks. *IEEE Symposium on Visual Analytics Science and Technology* 2006 (Baltimore, MD), IEEE Computer Society; Silver Spring, MD, 2006; 43–50.

7 Kang H, Shneiderman B. Personal media exploration: a spatial interface supporting user-defined semantic regions. *Journal of Visual Languages and Computing* 2006; **17**: 254–283.

8 Storey MA, Musen M, Silva J, Best C, Ernst N, Fergerson R, Noy N. Jambalaya: interactive visualization to enhance ontology authoring and knowledge acquisition in Protege. *Workshop on Interactive Tools for Knowledge Capture* 2001 (Victoria, BC, Canada).

9 Wattenberg M. Visual exploration of multivariate graphs. *CHI 2006*, 2006 (Montréal, Québec, Canada), ACM: New York, NY, USA, 2006; 811–819.

10 Pretorius AJ, vanWijk JJ. Multidimensional visualization of transition systems, *Proceedings of the Ninth International Conference on Information Visualization* 2005 (London, UK), IEEE Computer Society: Washington, DC, USA, 2005; 323–328.

11 Kosak C, Marks J, Shieber SM. Automating the layout of network diagrams with specified visual organization. *IEEE Transactions on Systems, Man and Cybernetics* 1994; **24**: 440–454.

12 Munzner T, Guimbretiere F, Robertson G. Constellation: a visualization tool for linguistic queries from MindNet. *The 1999 IEEE Symposium on Information Visualization* 1999 (San Francisco, CA); 132–135 + 154.

13 Dwyer T, Koren Y. Dig-CoLa: directed graph layout through constrained energy minimization. *Proceedings of the 2005 IEEE Symposium on Information Visualization* 2005 (Minneapolis, MN), IEEE Computer Society: Washington, DC, USA, 2005; 65–72.

14 Dwyer T, Koren Y, Marriott K. IPSep-CoLa: an incremental procedure for separation constraint layout of graphs. *IEEE Transactions on Visualization and Computer Graphics* 2006; **12**: 821–828.

15 Swayne DF, Buja A, Lang DT. Exploratory visual analysis of graphs in GGobi. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)* 2003 (Vienna, Austria).

16 Auber D. A huge graph visualisation framework. In: Mutzel P and Jünger M (Eds). *Graph Drawing Softwares Mathematics and Visualization*. Springer-Verlag: Berlin, 2003, 105–126.

17 Wills GJ. NicheWorks – interactive visualization of very large graphs. *Journal of Computational and Graphical Statistics* 1999; **8**: 190–212.

18 Perer A, Shneiderman B. Balancing systematic and flexible exploration of social networks. *(Proceedings of IEEE Visualization/information Visualization) IEEE Transactions on Visualization and Computer Graphics* 2006; **12**: 693–700.

19 Brandes U, Wagner D. Visone – analysis and visualization of social networks. In: Juenger M, Mutzel P (Eds). *Special Issue on Graph Drawing Software, Springer Series in Mathematics and Visualization*. Springer-Verlag: Berlin, 2003, 321–349.

20 Breitkreutz B-J, Stark C, Tyers M. Osprey: a network visualization system. *Genome Biology* 2003; **4**: R22.

21 Ryall K, Marks J, Shieber SM. An interactive constraint-based system for drawing graphs. *ACM Symposium on User Interface Software and Technology* 1997 (Banff, Alberta, Canada), ACM: New York, NY, USA, 1997; 97–104.

22 He W, Marriott K. Constrained graph layout. *Constraints* 1998; **3**: 289–314.

23 Dengler EF, Marks JM. Constraint-driven diagram layout. *Proceedings of IEEE Symposium on Visual Languages* 1993 (Bergen, Norway), IEEE Computer Society Press; Silver Spring, MD, 1993; 330–335.

24 Ghoniem M, Fekete J-D, Castagliola P. A comparison of the readability of graphs using node-link and matrix-based representations. *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)* 2004 (Austin, Texas), IEEE Computer Society: Washington, DC, USA, 2004; 17–24.

25 Henry N, Fekete J-D. MatrixExplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics* 2006; **12**: 677–684.

26 Brose U, Cushing L, Berlow EL, Jonsson T, Banasek-Richter C, Bersier L-F, Blanchard JL, Brey T, Carpenter SR, Blandenier M-FC, Cohen JE, Dawah HA, Dell T, Edwards F, Harper-Smith S, Jacob U, Knapp RA, Ledger ME, Memmott J, Mintenbeck K, Pinnegar JK, Rall BC, Rayner T, Ruess L, Ulrich W, Warren P, Williams RJ, Woodward G, Yodzis P, Martinez ND. Body sizes of consumers and their resources. *Ecology* 2005; **86**; Ecological Archives E086-135.

27 Aris A, Shneiderman B, Plaisant C, Shmueli G, Jank W. Representing unevenly spaced time data for visualization and interactive exploration. *Proceedings of the International Conference on Human-Computer Interaction (INTERACT 2005)* 2005 (Rome, Italy), Springer Berlin/ Heidelberg; 835–846.

28 Phan D, Xiao L, Yeh R, Hanrahan P, Winograd T. Flow map layout. *Proceedings of the 2005 IEEE Symposium on Information Visualization* 2005 (Mineapolis, MN), IEEE Computer Society, Washingon, DC, USA, 2005; 219–224.

29 Holten D. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics* 2006; **12**: 741–748.

30 Shneiderman B. Plaisant C. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. *Proceedings of the BELIV'06 Workshop, Advanced Visual Interfaces Conference 2006* (Venice, Ialy), ACM: New York, NY, USA, 2006; 1–7.