ABSTRACT

Title of Document: VISUALIZING & EXPLORING NETWORKS USING SEMANTIC SUBSTRATES

Aleks Aris, Doctor of Philosophy, 2008

Directed By: Professor Ben Shneiderman
Department of Computer Science

Visualizing and exploring network data has been a challenging problem for HCI (Human-Computer Interaction) Information Visualization researchers due to the complexity of representing networks (graphs).

Research in this area has concentrated on improving the visual organization of nodes and links according to graph drawing aesthetics criteria, such as minimizing link crossings and the longest link length. Semantic substrates offer a different approach by which node locations represent node attributes. Users define semantic substrates for a given dataset according to the dataset characteristics and the questions, needs, and tasks of users. The substrates are typically 2-5 non-overlapping rectangular regions that meaningfully lay out the nodes of the network, based on the node attributes. Link visibility filters are provided to enable users to limit link visibility to those within or across regions. The reduced clutter and visibility of only selected links are designed to help users find meaningful relationships.

This dissertation presents 5 detailed case studies (3 long-term and 2 short-term) that report on sessions with professional users working on their own datasets using successive versions of the NVSS (Network Visualization by Semantic Substrates, http://www.cs.umd.edu/hcil/nvss) software tool. Applications include legal precedent (with court cases citing one another), food-web (predator-prey relationships) data, scholarly paper citations, and U. S. Senate voting patterns. These case studies, which had networks of up to 4,296 nodes and 16,385 links, helped refine NVSS and the semantic substrate approach, as well as understand its limitations. The case study approach enabled users to gain insights and form hypotheses about their data, while providing guidance for NVSS revisions. The proposed guidelines for semantic substrate definitions are potentially applicable to other datasets such as social networks, business networks, and email communication. NVSS appears to be an effective tool because it offers a user-controlled and understandable method of exploring networks.

The main contributions of this dissertation include the extensive exploration of semantic substrates, implementation of software to define substrates, guidelines to design good substrates, and case studies to illustrate the applicability of the approach to various domains and its benefits.

VISUALIZING & EXPLORING NETWORKS USING SEMANTIC SUBSTRATES

By

Aleks Aris

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Professor Ben Shneiderman, Chair
Associate Professor Wayne McIntosh
Associate Professor Douglas W. Oard
Professor Amitabh Varshney
Associate Professor Benjamin B. Bederson

Dedication

To my grandmother, Makruhi Aris

Acknowledgements

I sincerely thank my advisor, Ben Shneiderman, who provided continuous support and encouragement throughout my Ph.D program. He promoted a positive atmosphere with his enthusiasm and a sympathetic attitude. Ben is one of the most enthusiastic people I have encountered in my life. I appreciate his optimistic, gentle, respectful, and graceful approach that manifests itself through his personal interaction. Over the years he has regularly offered a complete, fair, and accurate assessment of my work. He not only pointed out the improvements, but also acknowledged and expressed his appreciation for the parts done well. His approach helped me to understand the big picture accurately and then focus on the parts that needed attention with more assurance. I was surprised when I realized that this is actually an application of his well-known principle (*overview first, zoom and filter, then details on demand*). I have found many such examples. He not only has discovered and promoted these principles but also lives by them.

Besides research, I have learned numerous related things from Ben. These include writing issues, such as using positive expressions, sufficient yet concise and simple descriptions. I greatly appreciate his encouragement to take or attend classes on communication, specifically writing and speaking. In many other matters, he helped me learn how to approach, what to expect, and how to proceed in certain situations. I appreciate his patience, forbearance, and persistence in various circumstances. Finally, I could not help but notice the many similarities he shares with my father, such as being a leader in his profession, having a large number of

social connections, maintaining a friendly and approachable demeanor, thinking and communicating positively, being highly enthusiastic, and deeply enjoying his work.

I would like to extend special thanks to Doug Oard for his extensive comments and generous support during my dissertation revisions. In addition to meeting one-to-one, he provided beneficial comments on written drafts, during the proposal, and the dissertation defense. I feel privileged to know him as I highly benefited and learned from his analytic truth-revealing approach. Through his major contribution, this dissertation increasingly improved over the course of 6-8 weeks both in readability and in clear communication of research outcomes.

I appreciate the ongoing support and collaboration of Wayne McIntosh. He did not only serve as a member of my committee but also provided financial support for several years as a partner for the research in this dissertation. I appreciate his leadership skills that facilitate the progress of a project with realistic and concrete steps and communication during the progress, which kept the plans realistic and enabled to enjoy working on it. In addition, I have been impressed many times by his technical ability and intuitive understanding despite his expertise in a non-technical field.

I would like to thank Amitabh Varshney for serving on my committee. His positive look encouraged a constructive atmosphere throughout the proposal and the dissertation defense steps. Since his early encouragement and communication, I was able to understand this process more clearly, which helped me to be more productive and pass through challenging times.

I would like to thank Ben Bederson for serving on my committee and appreciate his earlier support. His careful attention to important details and ability to communicate them well despite the complexity has been both impressive to me and helpful to improve the dissertation.

I appreciate the valuable collaboration of Noshir Contractor. He has provided partial support and has participated for part of the research in this dissertation. In addition, we plan to continue this work together in Northwestern University. I appreciate our past collaboration including his thoughtful comments and suggestions and I am looking forward to working together in the near future.

I would like to thank the Cite-It team led by Wayne McIntosh, which includes Stephen Simon, Ken Cousins, and other members, who have provided useful comments for the Cite-It case study.

I thank Steven Harper for the early collaboration and efforts for the TobIG dataset and other members of the CI-KNOW team led by Noshir Contractor, which includes Andy Don, Hank Green, and Nat Bukley.

Despite her busy times, Cynthia Parr provided ongoing support by communicating via email to answer questions and clarify ambiguities. She also provided useful comments on written drafts describing the outcomes of initial stages of NVSS case studies and was available to meet for the Food-Web case study session. I appreciate her ongoing commitment to this early case study despite the many challenges we encountered with the data and the visualization.

I thank Chris Wilson for preparing the SenateVotes case study dataset and communicating to address and resolve issues we encountered. I also appreciate his positive communication style.

I would like to thank the IOpener team including Bonnie Dorr, Judith Klavans, Jimmy Lin, Saif Mohammad at the University of Maryland and Vahed Qazvinian and Dragomir Radev at the University of Michigan. This team provided comments on the visualization. Ben Shneiderman was also part of the team. Vahed Qazvinian brought the dataset together under the guidance of Dragomir Radev and Ben Shneiderman. Ben Shneiderman provided guidance, feedback, and structure to the meetings as well as taking part in the preparation of the visualization consisting of Treemap papers.

I would like to thank the PubMed team for taking the NVSS implementation and putting it to use to visualize Biomedical databases. The PubMed team members are Mike Lieberman, Inbal Yahav, Sima Taheri, Huimin Guo, and Fatemeh Mir Rashed. Also, Mike Lieberman provided many useful comments on the NVSS implementation.

I would like to thank friends and colleagues in the HCIL (the Human-computer Interaction Laboratory) and in the Computer Science Department at the University of Maryland. Catherine Plaisant worked together on prior research topics, provided guidance, help, and shared her expertise & intelligence to overcome many issues successfully. She also provided feedback on initial stages on this research work. Many times, Jennifer Golbeck helped on critical issues including comments for the dissertation defense presentation. Her input helped improve the initial

presentation substantially. On my dissertation defense presentation, Sheri Massey pointed out many improvements and provided detailed explanations. Mike Lieberman provided many comments, most of which I found useful and applied to my dissertation defense presentation and the final draft of this dissertation. Many times during the Ph.D., he also offered support as a friend, was a listening ear, shared his wisdom, and gave useful ideas during challenging times, which worked well despite their simplicity. I am impressed with and inspired by his decision-making ability, non-judgmental attitude, simple yet powerful thoughts, accurate judgments, and successful & efficient progress in many areas of his life. Adam Perer provided many thoughtful and educated comments not only on drafts of papers but also on the late stages of the Ph.D. process and has been a wonderful colleague. I have felt so grateful to him. Many times I felt I could not express my gratitude sufficiently and could not return the favor. In addition, I appreciate Adam's humble yet successful, diligent, and generous character. Chang Hu gave feedback and provided countless comments on many conceptual issues and walked with me on difficult paths of theoretical explorations, many times on the whiteboard and the computer screen. He offered his generous help with great efficiency even for a few moments in his busiest times, which made a difference. I thank Jagan Sankaranarayanan for providing comments on the final draft of the dissertation.

I appreciate the support, advice, and guidance from Joanne DeSiato on various issues ranging from how to phrase an email to possible resources on campus and what to do in certain circumstances. I appreciate her positive attitude and educated wisdom, which I learned greatly from in unexplainable ways. Furthermore, her help did not

only provide me a wonderful learning opportunity but also saved time in many instances.

I am very grateful to my mentor Don Jewett on various issues. He has been very supportive and prompt in communicating, which helped in many issues including managing time more effectively for research & programming, increasing productivity in writing, thinking about, and formulating research material. He also helped me overcome many challenging situations and gave encouragement through accurate and fair assessments of my situation. I have been surprised by his correct insights and judgment despite the limited communication we had and his positive helpful responses, many of which were unknown to me before our communication.

I would like to thank other HCIL members who have been of support many times in various situations including Anne Rose, Kiki Schneider, Allison Druin, Ken Fleischmann, Vibha Sazawal, Tejas Khatri, Alex Quinn, David Wang, Jerry Fails, Mona Leigh Guha, and former members including Gene Chipman, Bongshin Lee, Jinwook Seo, Bill Kules, Hyunmo Kang, Harry Hochheiser and visiting members of HCIL including Paolo Buono, Pekka Parhi, Anthony Don, and Romain Vuillemot.

I thank HCI researchers in the field, colleagues, and friends who interacted with me about this research, the Ph.D. process and/or provided useful oral or written comments including Martin Wattenberg, Stephen North, Jeffrey Heer, Johannes Pretorius, Wolfgang Aigner, Sean Smitz, Fritz Ebner, Michael Smitz, Oren Etzioni, Loretta Auvil, Wendy Carter, Abigail Daken, Sean Williamson, Bernard Chabot, Elena Zheleva, Galileo Namata, and Marie desJardins.

Finally, I would like to thank my mother, who provided support during and before the Ph.D. process in various ways, my father, my other family members, friends of my family, and my friends for their various types of support.

NSF and Microsoft provided partial support for the research work in this dissertation. The Cite-It project provided support under the U.S. National Science Foundation grant “Inter-Court Relations in the American Legal System: Using New Technologies to Examine Communication of Precedent II” (SES-0519157). NSF and other funds were provided by agreement between Noshir Contractor and Ben Shneiderman. Ben Shneiderman provided additional support through NSF and Microsoft funds.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Problem Statement

Networks are found in different forms across various application areas, such as social networks, biological food webs, scholarly paper citations (e.g., conference and journal papers citing one another and legal court cases referencing each other). Network visualization enables users to see the pattern of nodes and links; detect interesting cases; and analyze, interpret, and arrive at conclusions.

1.1.1 Definition of Network

The terms *network* and *graph* are used interchangeably in the literature. *Network visualization*, also known as *graph* visualization, is a visual presentation of objects 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. An object in a network will be referred to as *a* node, while a connection between two nodes in the network is referred to as *a link.* A *link* is also referred to as *an edge*. The visual representation of a node is usually a shape such as a circle, a square, or a rectangle while the visual representation of a link is usually a line or a curve connecting the two nodes.

Figure 1 shows a typical network visualization, where nodes are represented as squares and links are represented as gray lines. Nodes represent companies that had a role in Apple's popular product, iPod, in 2001. Red (Apple) is used for accessory

makers, while blue (all except Apple and Inventec) is used for technology providers, and green (Inventec) is used for competitors. Links show communication between companies.

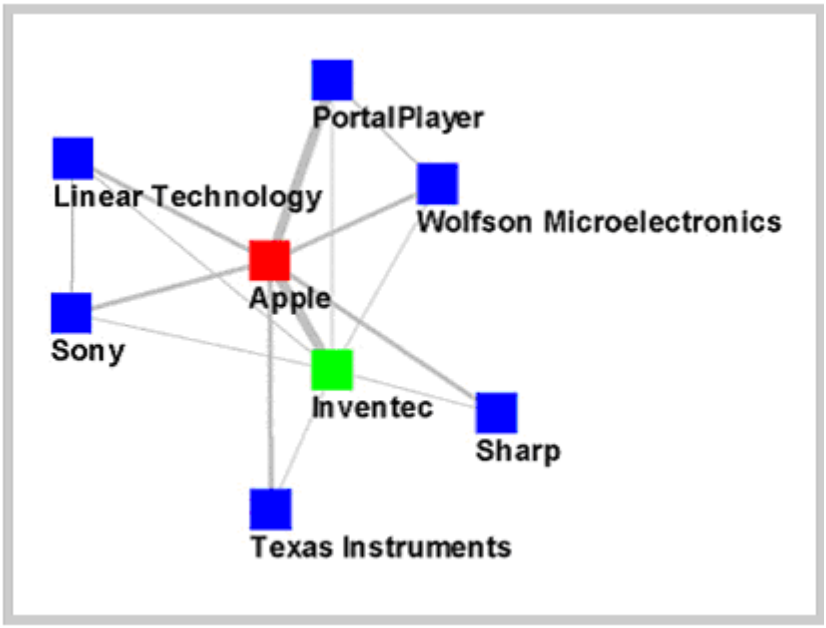


Figure 1 A network visualization that shows companies and communications between them. Source: Matt Woolman, 2005, < http://www.visualcomplexity.com/vc/project_details.cfm/?index_number=152&id=164&DomainName=>.

1.1.2 Motivation

Network visualization is superior to textual representation for some tasks because it utilizes the capabilities of the human's visual perceptual system. A well-designed network visualization enables fast and accurate interpretations and supports overviews. While network visualization facilitates faster and better understanding, presentation becomes a challenge as the number of nodes and links increases. Labels are increasingly harder to show and the display begins to get too cluttered to comprehend (see Figure 2).

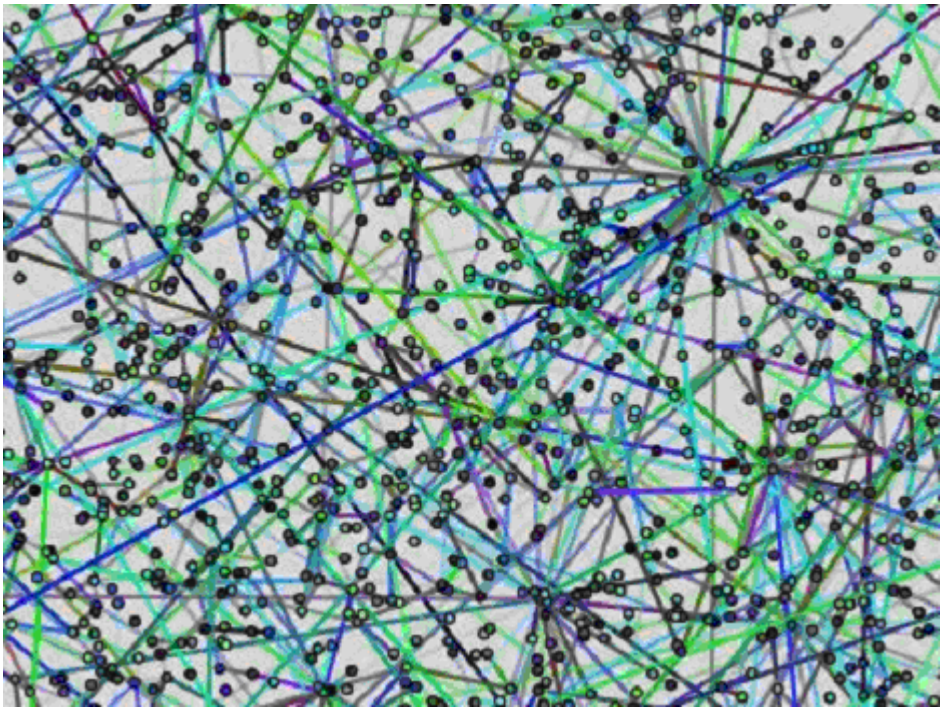


Figure 2 Graphs become harder to understand as the number of nodes and links increase. Source: Matt Woolman, 2005, <http://www.visualcomplexity.com/vc/project_details.cfm/?index_number=135&id=146&DomainName=>.

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, consequently better understanding. Another trail of research uses these aesthetics and provides algorithms that strive to conform to them when drawing networks. Some of these algorithms are concerned with efficiency, as well, since it is computationally expensive to attain the goals defined by these aesthetics (Brandenburg 1988). For example, determining the number of link crossings is NP-complete (Gary 1983).

As the number of nodes and links grows, 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 clear) 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 seems to be the arbitrary placement of nodes on the screen. Users tend to associate the place of a node with the object the node represents. A node consistently placed in the same location is likely to become familiar over time, where users start to use its location to interpret the object it represents. Furthermore, when nodes with similar characteristics are placed close to each other, they can be conceptualized as a group. This 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 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.

When nodes are placed arbitrarily, location is not utilized as a channel of information to convey meaning. Furthermore, users may be confused if nodes look as if they are placed according to a set of human-interpretable rules. The spatial location of a node can be leveraged to convey information. Using node attribute values to determine where to place the nodes, if done in a simple way, will convey information about the nodes. Using placement of nodes in this way, the comprehensibility of networks can be enhanced, which can lead to increases in productivity and user satisfaction.

Other strategies than the placement of nodes, 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 strategies other than placement to deal with the complexity of the network.

1.1.3 Overview of the Solution

The approach used in this dissertation conveys information about nodes by using node attribute values to determine a spatial layout. The author of this dissertation started exploring this possibility with researchers from the Government & Politics Department at the University of Maryland, College Park. A sample network (Figure 3) shows the precedent patterns, where a node represents a court case and a directed link represents a citation from one court case to another.

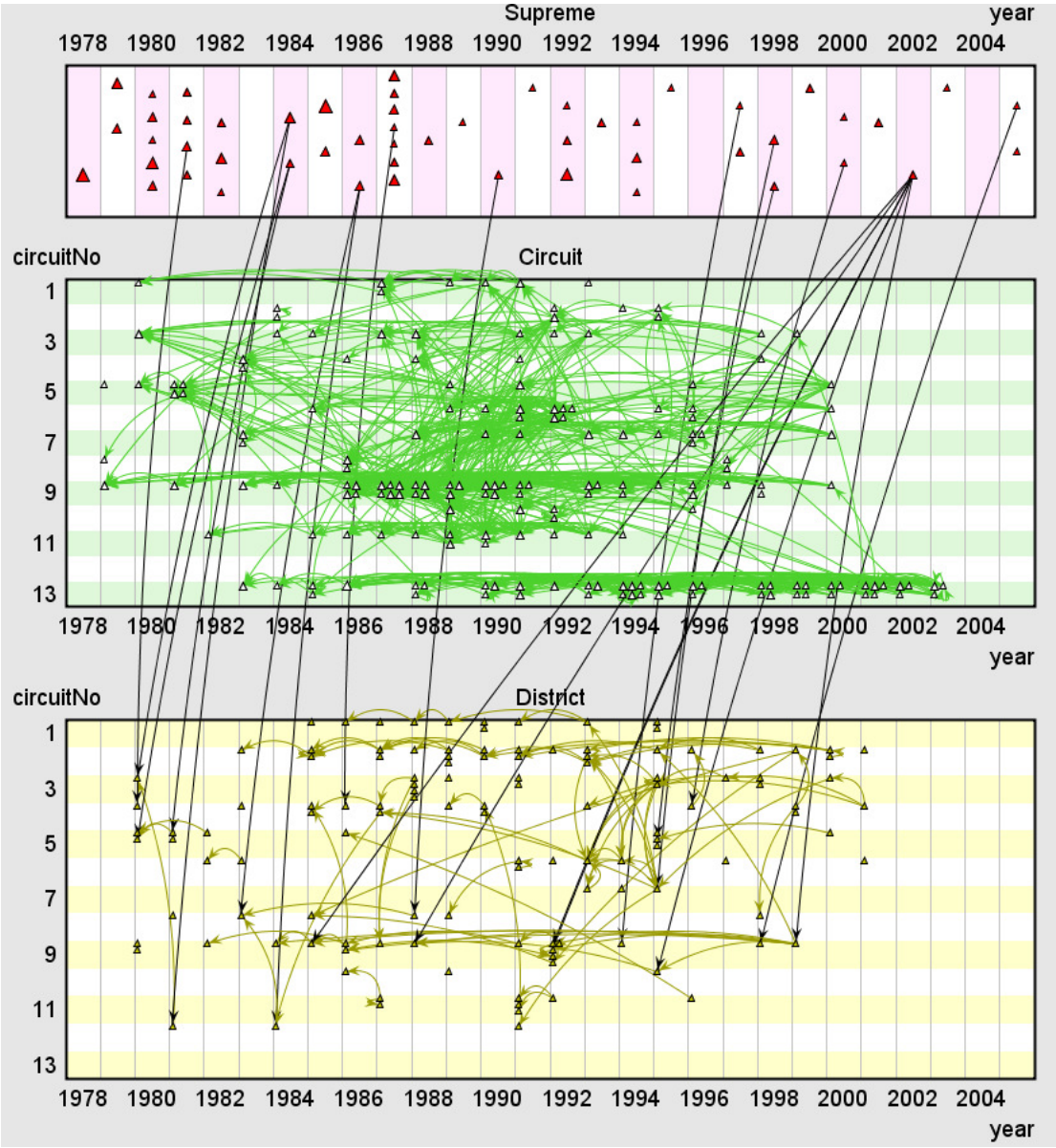


Figure 3 Visualizing legal court cases: regulatory takings cases from 1978 to 2005.

Nodes have attributes such as *date*, *name*, *courtType*, and *name*. *courtType* has values "Supreme," "Circuit," and "District," and the values of *courtType* are used to group the nodes into rectangular regions. The *year* and *circuitNo* attributes are used to further arrange the nodes inside the regions. Visual attributes of regions (i.e., size and location) and the set of nodes they contain (e.g., nodes representing "Supreme Court" cases) 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.

Although there are applications that have the elements of semantic substrates, they have many restrictions either on the dataset, on the user control aspects, or the features that allow exploration. This dissertation emphasizes a much more flexible type of user control on network layout using node attributes via the semantic substrate approach. In addition, the types of networks that this approach is applicable to are quite general.

1.1.4 User tasks

User tasks are highly important in a network visualization domain as they are in any human-computer interaction field. Understanding users and their needs is essential to help them complete tasks to achieve their goals. While exploring one set of users, who are Government & Politics researchers, from the point of view of the author of this dissertation, the user needs and tasks can be summarized in terms of the following three major categories:

- *Understanding the network and its elements (nodes and links).* Attribute selection to define regions and placement methods within regions help fulfill this need.
- *Being able to manipulate the network layout and the visual properties.* Users have control to define their own substrate, and actually define more than one to experiment with different substrates for the same data. A module called *Substrate Designer* is provided to enable this process. Users can visually

define the size and location of regions, the nodes that fall into each region and their placement method along with other visual attributes, such as node size coding and region background color.

- *Scalability.* Users are likely to need to explore larger networks. Challenges arise as the number of nodes and links grow. A metanode mode is provided to address this challenge. In this mode, nodes with the same placement attribute values are represented by a single node, which is called a "metanode". This way, the number of nodes is reduced on the display. As node groups are transformed into metanodes, links between those node groups are also transformed into metalinks. Therefore, the number of links on the display decreases, as well. As a result, this mode provides an aggregated view. Combined with link filters, the reduced overview display allows exploration of larger datasets in the context of semantic substrates.

1.2 Contributions

The implementation and case studies show the applicability of the semantic substrate approach to directed simple networks that have node attributes with single values. This dissertation illustrates this approach in a generalized way applicable to various domains and datasets, shows its feasibility by implementing it, and illustrates the benefits evidenced by the case study outcomes.

Contributions can be listed specifically as follows:

- *The definition of the semantic substrate idea:* The 3-step approach to place nodes visually using node attributes. First, nodes are grouped into regions

by a node attribute; second, they are placed within regions by other node attributes; third, link visibility controls are provided to users.

- *The technical structure and the visual & interactive design of a semantic substrate:* The definition (contents) of a semantic substrate, how it is separated from the data in terms of implementation design and the visual appearance of the substrate, its components, the widgets provided, their functionality, coordination, and organization on the display.
- *The user design process of semantic substrates and guidelines:* How to let users design and use a semantic substrate as well as good substrate design practices, which are the guidelines that resulted from the case studies.
- *Scalability in the context of semantic substrates:* How to visualize datasets as they increase in the number of nodes and links without compromising the benefits of the semantic substrate approach. The node aggregation mechanism provides a solution to explore larger datasets by providing a meaningful overview through the aggregated view. Users' choice of binning determines the categories of nodes in the aggregated view, while the non-aggregated view provides details-on-demand. This solution also accelerates the process of exploration since the aggregated view is a simpler display. Users can process the simpler version of the information better while seeking relationships and facts about the data.
- *Implementation (the NVSS application):* How a system of network visualization using semantic substrates can be implemented. The implementation of NVSS is evidence of the feasibility of the

implementation of the semantic substrate idea. The fact that several datasets could be imported, visualized, and explored in NVSS shows evidence that this idea is quite feasible and sound to implement. The implementation also has left room for extensions. Details are provided in the implementation details chapter (Chapter 4: Implementation Details) of this dissertation.

- *Case study results:* What benefits and results users could gain by using the semantic substrate approach in several domains. The results of several case studies show the types of benefits and insights users could gain by using the semantic substrate approach in network visualization. These results also show that it is a feasible approach for a variety of audiences in terms of *use* (the system is sufficiently understandable and/or easy enough to be used or learned by various users), *general applicability* (this approach can be applied to several domains, not only one), and *useful* (using this approach, users have gained several insights or a useful understanding of the dataset they were exploring. They were not able to gain many of these benefits by their previous methods of exploration).

This dissertation work has also led to two publications ((Shneiderman 2006), (Aris 2007)) and one technical report (Aris 2008). In addition, a team of five graduate level information visualization students at the University of Maryland, College Park used the application to help explore biomedical data (genes, OMIMs (Online

Mendelian Inheritance in Man, a type of document), and publications) from the online database PubMed[1] of NCBI (Lieberman 2008).

In conclusion, semantic substrates accompanied with good substrate design promise a higher quality exploration of networks through increased user control, which leads to better understanding and deeper insights. Furthermore, node aggregation enhances the benefits of using semantic substrates for complex networks by enabling simpler, and therefore, a more efficient and effective exploration of networks.

1.3 Summary of Introduction

Network Visualization using Semantic Substrates is based on visualizing networks based on layouts defined by users. Users use the node attributes to group nodes into (rectangular) regions and select from the available placement methods (and setting node placement attributes as their parameters) to define the placement strategy within each region. This essentially enables users to create their own conceptual map to place the nodes. The expectation is that this will convey the meaning of nodes by their location and placement attributes and users will be able to make sense of the visualization and exploration process. The contributions include the definition of the semantic substrate idea, the technical structure and the visual & interactive design of the interface, the user process to design a semantic substrate and guidelines, the scalability extension (the node aggregation feature) to enable exploration of larger networks, the software (NVSS), and case study results.

[1] http://www.ncbi.nlm.nih.gov/pubmed/

The rest of the dissertation is organized as follows. Chapter 2 provides the previous work, which includes analysis of 75 network visualization applications besides a literature review. Chapter 3 describes the design issues of the semantic substrate approach. It provides the design choices made for the visualization, describes the elements and their utility on the control panel, describes the design of the SubstrateDesigner module, and the details for the node aggregation extension. Chapter 4 provides implementation details, which include statistics on system performance. Chapter 5 describes the evaluation methodology used and gives summary descriptions on the 5 case studies. Chapter 6 lists the guidelines arrived at through the case studies and it provides discussion with examples from the case studies in Chapter 5. Chapter 7 includes future work and concludes with the summary of contributions.

Chapter 2: Previous Work

This chapter provides the previous work that is related to the semantic substrate approach of network visualization. The previous work is divided into 3 major parts:

The first part (section 2.1) is a review of early work that gave rise to a set of principles to guide drawing graphs (networks). This early work provides the foundation of many existing network visualization approaches and it is necessary to review it in order to understand the state-of-the-art visualization approaches as many of them follow these guidelines. These guidelines usually result in arbitrary placement of nodes; and therefore, do not utilize location to convey meaning about nodes, which the semantic substrate approach leverages.

The second part (section 2.2) is an empirical review of 100 visualizations. The focus of this analysis is node placement strategies used in network visualization applications. The review results in a classification of 75 of the 100 visualizations in terms of their node placement strategies. (The remaining 25 did not have enough information to be categorized.) This review is not comprehensive; however, it gives readers an idea of commonly used node placement strategies.

The third part (section 2.3) reviews the literature in terms of node placement techniques in network visualization to situate the semantic substrate approach as well as to compare and contrast with the existing research.

The fourth part (section 2.4) reviews the literature in terms of scalability strategies employed in the literature to situate the scalability extension of the semantic substrate approach, which helps to explore larger datasets.

The fifth part (section 2.5) reviews various evaluation approaches and identifies the reasons of using a case study approach to evaluate NVSS.

2.1 Graph Drawing Aesthetics

This section provides a review of early research to find principles and guidelines on how to draw graphs on a computer screen. This is highly relevant to the semantic substrate approach as drawing graphs is primarily affected by node placement. While this section provides the review of theoretical principles and guidelines for drawing graphs (commonly called "graph drawing aesthetics"), section 2.2 provides an analysis of 75 network visualization applications, and section 2.3 provides the review of the related follow-up literature to the literature in this section. While the literature in this section is theoretical, the follow-up literature in section 2.3 incorporates applications, as well. The review both in this section and in section 2.3 is contrasted with the semantic substrate approach to situate the semantic substrate approach within the existing node placement strategies and how the semantic substrate approach differs from other strategies.

Several early researchers recognized the importance of improved graph presentation as the number of nodes and links increases in a graph. In fact, even with small graphs, a bad layout can make it difficult to comprehend. While it is difficult to comprehend the graph in Figure 4a, it becomes considerably easier to perceive the structure when its layout is improved as in Figure 4b.

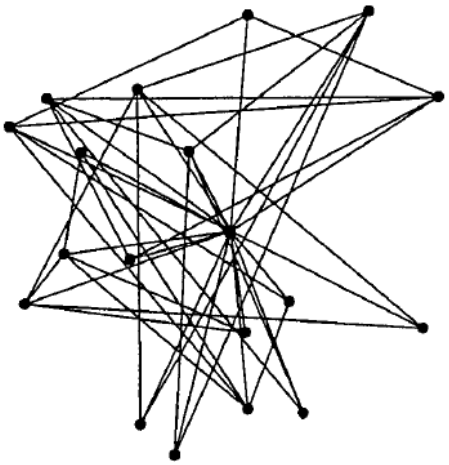


Figure 4a A graph with a bad layout. Source:(Davidson 1996)

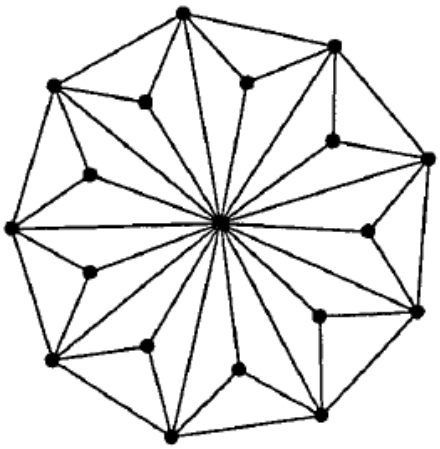


Fig 4b Improved layout of the graph in Fig 3a. Source: (Davidson 1996)

Researchers use 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, each of which is 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 she 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.

Although the graph drawing aesthetics help the perception of the graph, they usually place nodes arbitrarily. Arbitrary placement of nodes does not help identify nodes in terms of their locations. On the other hand, the semantic substrate approach enables the location of a node to be related to its attributes. In this way, location is leveraged to convey information about the nodes, and collectively, the graph.

2.2 Review of Visual Complexity

To gain insight about the placement methods utilized in common practice, 100 network visualizations were analyzed (by the author of this dissertation) 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 currently lists more than 600 graph visualizations. However, at the time of collection, it listed approximately 150 visualizations. At that time, the first 100 network visualizations were selected. The visualizations appear in the order that they are submitted (i.e. the first visualization is the first visualization submitted). Therefore, the review is limited to a report of the first 100 visualizations that were submitted by December 7, 2005 and may not be generalized.

The method for node placement was not specified for 25 of the visualizations. Some of the visualizations 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 (75 of them) 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 Frequency / 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 Visual Complexity.

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%	

In this categorization, the force-directed method emerged as the most frequently used method (33%). One of the applications using force-directed also used a clustering method (Figure 5).

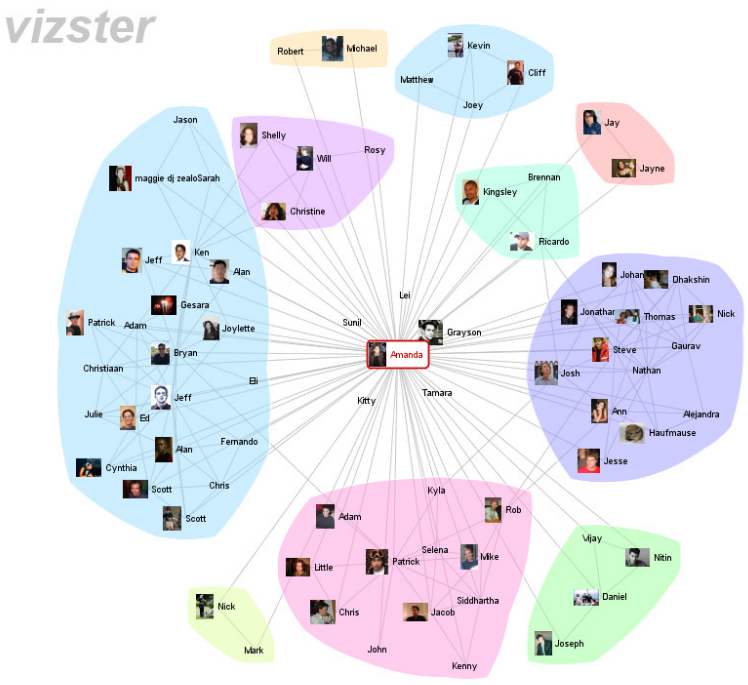


Figure 5 Vizster - One of the visualization using a force-directed approach to place nodes. It also uses clustering to group nodes according to the number of connections between nodes (85th visualization on VisualComplexity.com).

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. An example for the geographical map is illustrated in Figure 6, where the map of the United States is used in the background.

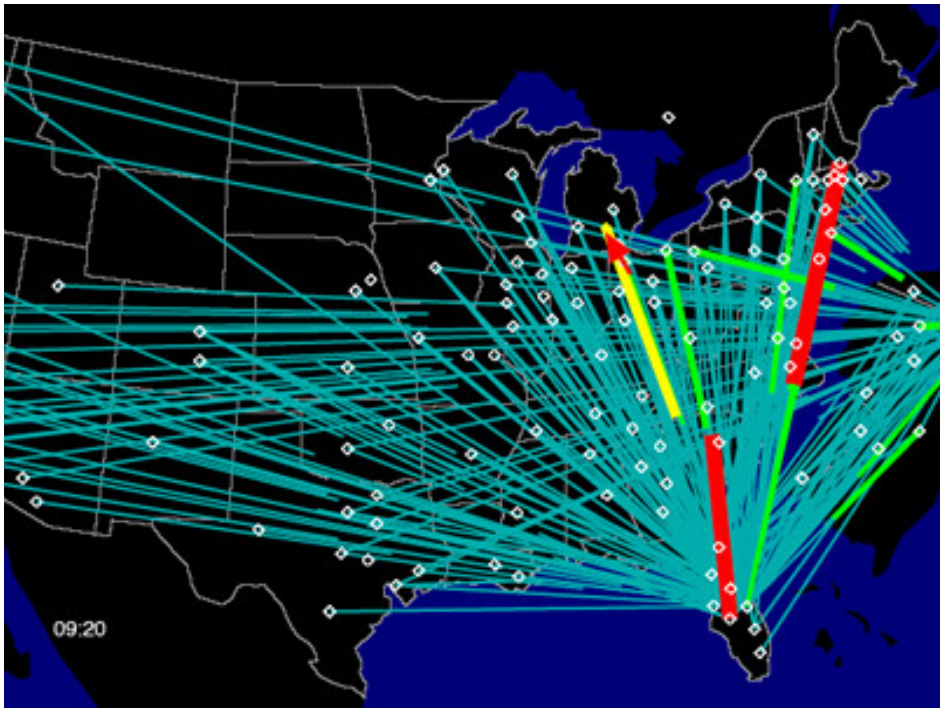


Figure 6 Example for nodes laid out on a geographical map (100th visualization on VisualComplexity.com).

Visualizations in the circular category used concentric circles or a single circle sometimes with a node or a collection of nodes in the center (see Figure 7). 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 visualization sorted the nodes alphabetically around the circle, while another sorted the nodes using an external factor, such as an input file specifying the order of the creation of the objects over time (as in "Social Circles" in Figure 8). Most of the other visualizations using circles are placed in a second category according to the method they use (see Table 1).

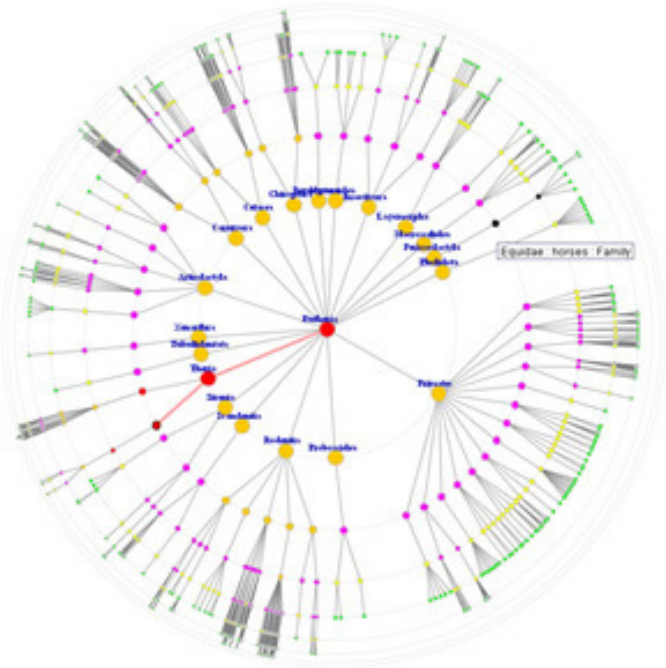


Figure 7 Circular layout is used with a center node (26th visualization on VisualComplexity.com).

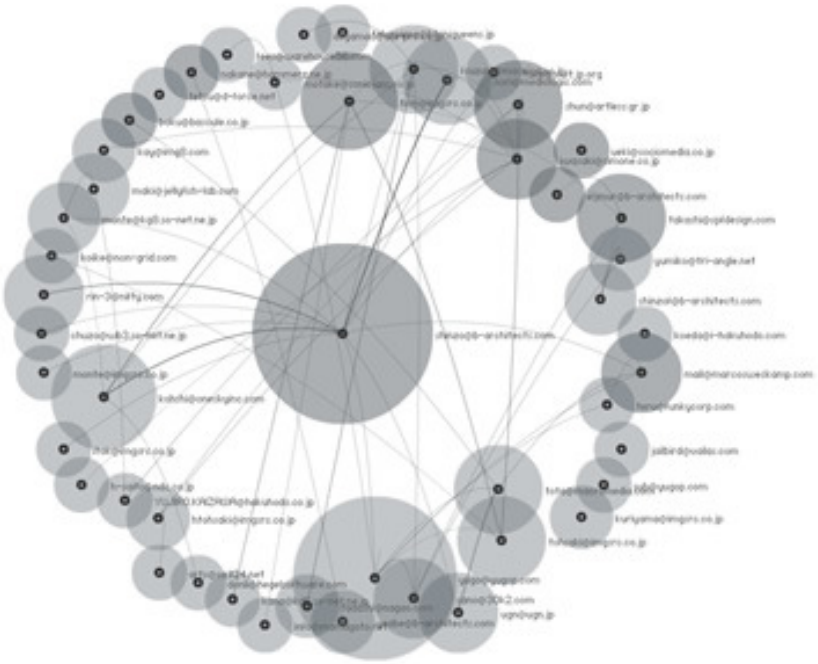


Figure 8 Social Circles - It uses a circular layout to convey the structure and activity of the social network in mailing lists. The central character is placed as the central node (58th visualization on VisualComplexity.com).

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. However, they may not be part of the network, in which case they are not nodes of the network but 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" (Figure 9), the 36th visualization, sorted Dewey classification numbers (these are the *related entities* in this visualization as defined above) 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.

Figure 9 Making Visible the Invisible – nodes are placed according to Dewey classification numbers provided as horizontal strips at the top and bottom, sandwiching the nodes (36th visualization on VisualComplexity.com).

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" (Figure 10), 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.

Figure 10 When users click on the map, nodes adjust according to the clicked distanced proportional to the flight time it would take to get there (92nd visualization on VisualComplexity.com).

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" (Figure 11) 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.

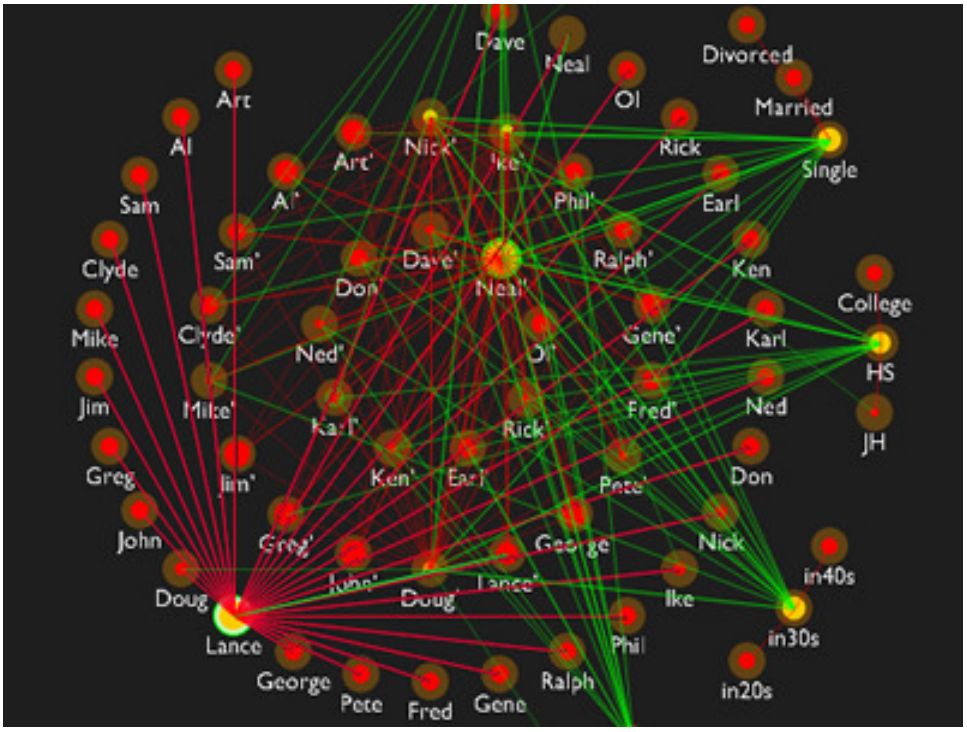


Figure 11 Interactive Activation – a connectionist network which groups the nodes according to their type (79th visualization on VisualComplexity.com).

Time is used to place the nodes in two time-oriented visualizations. In "Time Graphs," (Figure 12) 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. Another visualization (Figure 13) used time to divide the concentric circular layout into twelve sections, where each section represents a month of a year.

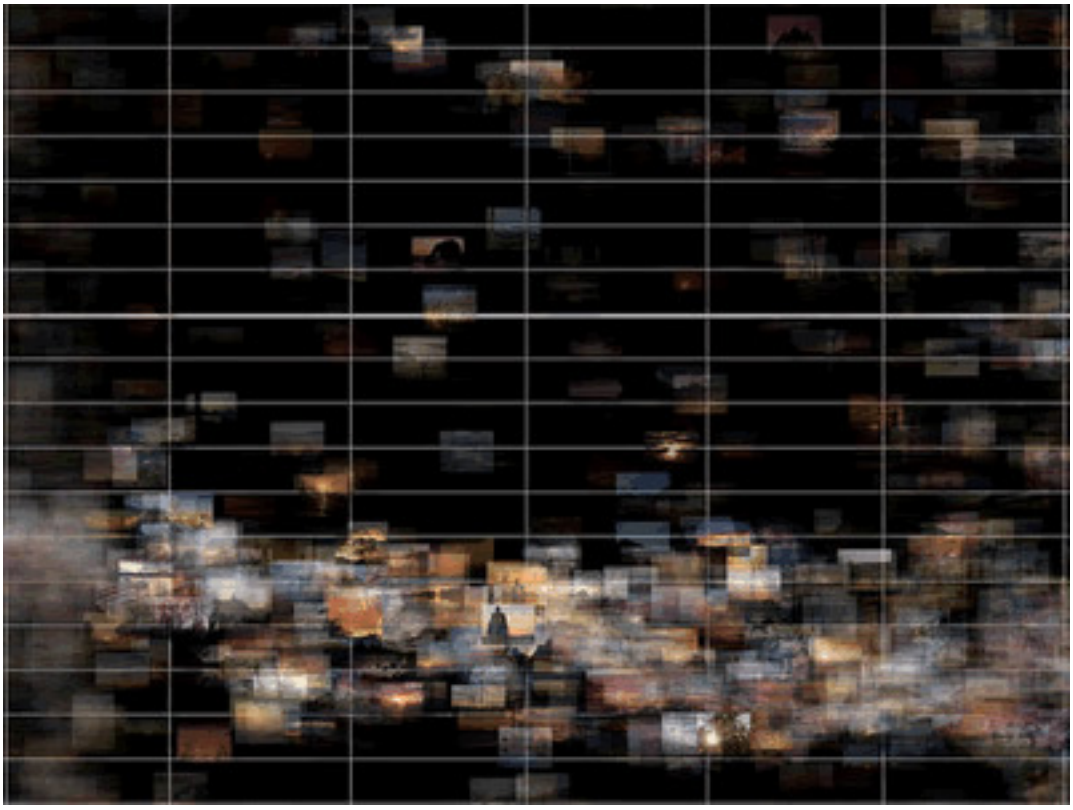


Figure 12 Time Graphs – nodes are ordered according to increasing time from left to right (87th visualization on VisualComplexity.com).

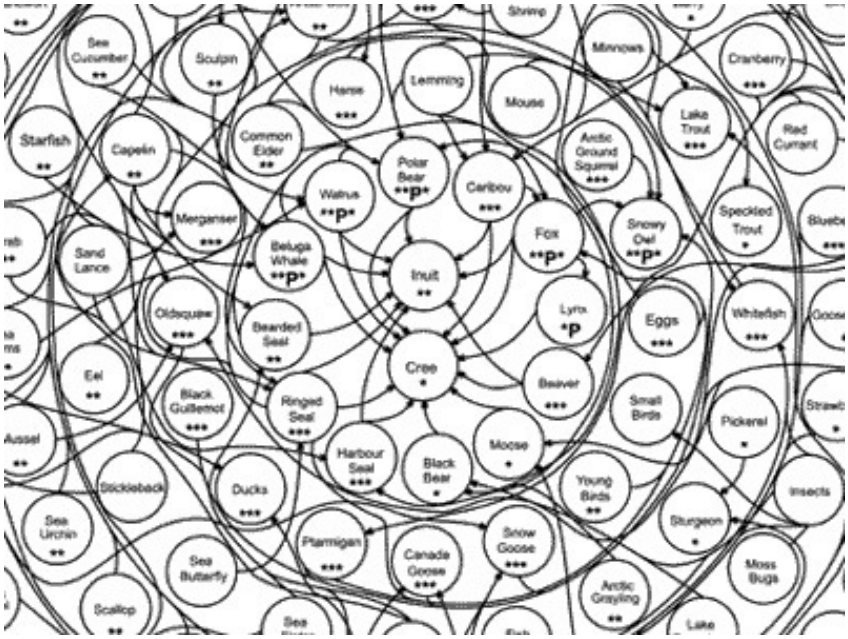


Figure 13 Hudson Bay Food Web – relationships between species are visualized in a circular fashion having 12 imaginary sections, each section representing a month in a year (96th on VisualComplexity.com).

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

From this categorization, there was only one visualization (Figure 11) among 75 that used a substrate-based approach (i.e. placed nodes according to node attribute(s)), which actually used only one attribute to group nodes and did not use node attributes to further place the nodes within groups. From this report, the conclusion arrived was that in the set of network visualizations analyzed, the presence of the substrate-based approach was very rare.

2.3 Node Placement Techniques

This section reviews the literature in terms of the node placement strategies. Most of the time, research in this literature follows the graph drawing aesthetics guidelines in section 2.1. The following review of the literature starts with an overview, categorizes the literature in terms of node placement strategies (note that this is parallel to the categories arrived at in section 2.2; there may be some overlap with the applications categorized in section 2.2; however, the applications in this section are not necessarily the same ones) and contrasts these strategies with the semantic substrate approach, then a transition is made to the research and applications that have elements of or are close to the semantic substrate approach.

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 16-year old International Symposium on Graph Drawing (http://www.gd2008.org/). A taxonomy of tasks for network visualization is provided by Lee et al. (Lee 2006).

Zooming (Bederson 2004) and fisheye (or other distortions) approaches have been used to give users some control, but effective layouts are still needed to minimize link crossings and tunneling under nodes. In addition, dynamic query filters may still be needed to reduce node and link density. NicheWorks included helpful interactive features, such as highlighting nodes, links, and hiding them, for analysis purposes of graphs ranging from 10,000 to 4,000,000 nodes. Using a subset of a telephone network call graph, Wills illustrates how an analyst could narrow the search to find patterns suggesting fraud (Wills 1999).

The literature on network layout has been dominated by force-directed strategies because they produce elegant spreading of nodes and reasonable visibility of links. Nodes are laid out as if there were electrical forces between them, where links determine the attraction between connected nodes. Eades (Eades 1984) proposed the idea but the most common reference is to the refined Fruchterman-Reingold (FR) algorithm (Fruchterman 1991), with further refinements by many others (Gansner 1998). Variations are sometimes called spring-embedding to describe the connections between connected 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 scalable versions of force-directed methods that work on a coarse representation of a large network, which refine the layout locally to achieve remarkably rapid layout for large networks (10^6 nodes in a few seconds).

A second common layout strategy, which generates familiar and comprehensible layouts, uses geographical maps, in which the node locations are fixed, as in cities on a world map (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). In this case, hierarchies are based on nested clustered groups. The leaf levels are at the bottom of the hierarchy, the nodes that represent clusters of the leaf nodes are one level above, nodes that represent clusters of clusters are one more level above, and so on. The clustering is based on the link structure between nodes; and therefore, these hierarchies are based on the structure of the graph. 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 (Agrafiotis 1997; Martin-Merino 2004) (an MDS (Multi-Dimensional Scaling) algorithm developed by Sammon (Agrafiotis 1997); MDS (Buja 2008) algorithms place items on the display based on a similarity measure between nodes) 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).

While these approaches may have elements of the semantic substrate approach, they do not provide a general system where users can choose node attributes to determine node layout. The geographical map approaches use latitude and longitude (or geographical location) of a node to place a node; however, this focuses only on networks that have geographical information. Even though such networks can be effectively visualized using a geographical map approach, there are instances where such networks can also be laid out via other attributes that could lead to useful explorations. Similarly, the approaches that use time are limited to using only time information; often in a way that is specific to the format of the dataset; not allowing users to choose other attributes. In this way, the semantic substrate approach provides users with a more general and flexible way of defining the layout of a network.

A major inspiration for user-defined semantic substrates idea has been the user-defined spatial layouts for photos with shared attributes (Kang 2005). A strategy that was a partial, more manual, and specialized version of the semantic substrate approach proved beneficial in a network visualization tool for author name resolution

in a 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. Although these systems provide inspiration, they are different than the semantic substrate approach (the photo layout is present in an application that does not visualize general graphs and the bibliographic database name resolution tool is specific to the database it uses in terms of the data structures and the node attributes chosen to layout the nodes).

Six recent systems have elements of semantic substrates; however, they are either not equivalent to the semantic substrate approach or they don't provide the generality that is provided in the semantic substrate approach (or both). These systems are compared with NVSS, which is the network visualization application that utilizes the semantic substrate approach.

Jambalaya (Storey 2001) integrates SHriMP views into the Protégé framework. A graph metaphor is used to show links between concepts, similar to regions in the semantic substrate approach, 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 as in the semantic substrate approach. Links are categorized and therefore can be color-coded by source and target classes. Also, this application is not for visualizing general networks but ontologies.

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. 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 supports multiple regions. Users can select node attributes on each of the x- and y-axes.

Pretorius et al. (Pretorius 2005) represent transition systems, systems that have states and transitions between them, as networks and uses the projection of multi-valued node attributes to the 2D plane to position nodes. The projection is parameterized 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 grids. NVSS also enables users to control the size and location of regions (grids in this research work).

Although in the two systems above 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. (Kosak 1994) group the 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. As networks get larger, the approach employed by NVSS will be faster than the manual approach. In addition, Kosak et al. focus on computing the layout while NVSS also provides link-visibility features.

In Constellation (Munzner 1999), 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.

The range sliders in NVSS are dynamic query filters (range sliders are referred to as double-sliders or double-box sliders) were inspired by early concepts of visual information seeking (Ahlberg 1994), which lead to the commercial software called Spotfire[2] (Ahlberg 1996; Shneiderman 1999). While the filtering effect of range sliders in Spotfire are applied to the entire dataset, NVSS has multiple range sliders categorized in terms of regions. In fact, Spotfire does not have a feature equivalent to having multiple regions. The continuous scattergrams in Spotfire are helpful in various situations. In contrast, NVSS has a grid-plot placement method, equivalent to binned scattergrams, which could be helpful in certain situations. Spotfire supports multiple scattergrams; however, this does not have the same effect of having regions in NVSS as they show different views of the same data. In addition, Spotfire does not represent graph data although a new feature to represent graphs will be added in the near future, which has been influenced by the concepts in NVSS (Ahlberg (CEO of Spotfire) 2008).

Dig-CoLa (Dwyer 2005) and IPSep-CoLa (Dwyer 2006) 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 (Dwyer 2006) has the additional

[2] http://www.spotfire.com

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 (Swayne 2003), Tulip (Auber 2003), NicheWorks (Wills 1999), SocialAction (Perer 2006), Visone (Brandes 2003), and Osprey (Breitkreutz 2003), Glide (Ryall 1997) provide other useful features, they do not support graph layouts based on node attributes:

GGobi (Swayne 2003) uses radial, dot, and neato layouts (the radial layout places a designated node at the center and arranges the rest of the nodes in concentric circles around it; the dot layout produces hierarchical layouts by finding optimal coordinates for nodes to maximize the use of graph drawing aesthetics; the neato layout is a force-directed algorithm that produces "spring" model layouts of undirected graphs). GGobi 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 one of the grid-plot placement methods in NVSS in terms of using a node attribute for the x- and y-axes to place nodes on display, there are differences. The scatterplots in GGobi do not contain graphs (they do not have links) whereas in NVSS the graph data is fully represented (as a node-link diagram; links connect the nodes that are placed using the grid-plot placement method). While GGobi scatterplots use jitter to eliminate overlap of nodes (having the same attribute values on both x- and y-axes),

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 (Auber 2003) 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 doesn't include links and it is limited to the hierarchical treemap algorithm) and no link visibility based on node attributes have been reported.

Visualizing large graphs (up to 1,000,000 nodes) is a driving goal for NicheWorks (Wills 1999), 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 support filtering on node attributes.

A type of force directed layout is used in SocialAction (Perer 2006), 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) and also supports node filtering and link visibility by node attributes.

Visone (Brandes 2003) provides a set of different algorithms to layout nodes, such as spectral, layered, and radial layouts.

A domain specific tool, Osprey (Breitkreutz 2003), enables biologists to combine datasets and provides node filters based on attributes.

Glide (Ryall 1997) 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 (He 1998) 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. (Dengler 1993) 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 et al. (Ghoniem 2004), MatrixExplorer (Henry 2006)). MatrixExplorer couples matrices with node-link diagrams (where matrices are subgraphs and the node-link representation connects the matrices) and provides interactive operations such as sorting matrix columns in terms of attributes and filtering. The node-link representation doesn't use layouts based on node attributes.

This review of the literature including application shows that there has not been a system that implemented the semantic substrate approach in a general way although they may have utilized some elements. The semantic substrate approach generalizes the idea of organizing nodes of a network in terms of their attributes and also defines a specific way that can be implemented in terms of three steps leading to the final network layout.

The next section reviews the literature in terms of the scalability approaches to situate the scalability extension of the semantic substrate approach in the context of related research work.

2.4 Scalability

Networks become harder to visualize as they get more complex, especially as the number of nodes and links increases. This section provides a review of the literature in terms of how network visualizations handle larger networks, that is, address the scalability problem. The semantic substrate has an extension to help explore larger networks. This extension is called the *node aggregation feature* of NVSS. It provides an additional mode for the visual display, called the "metanodes" mode. When users switch to this mode, nodes that have the same placement attributes are represented by (in other words, *are aggregated* to) a single node, which is called a metanode. This section compares and contrasts the existing research on how they address the scalability issue in network visualization and situates the node aggregation extension of the semantic substrate idea within this larger context.

Several strategies have been employed to reduce or cope with the complexity of networks. These can be categorized as clustering approaches, focus+context techniques, link aggregation and routing, graph drawing aesthetics, and matrix-based representations.

2.4.1 Clustering Approaches

Among the clustering approaches, PivotGraph (Wattenberg 2006) and Jambalaya (Storey 2001) show the closest characteristics to node aggregation in the

context of semantic substrates. PivotGraph (Wattenberg 2006) "rolls-up" a graph to produce a summary view of nodes and links on a two-dimensional grid based on a node attribute on each of the axes, which is the equivalent of node aggregation in one of the regions in NVSS. While PivotGraph allows users to change the node attribute on the axes and animates the transition, NVSS allows users to change the node attributes via its Substrate Designer interface. PivotGraph also uses size coding for metalinks (besides metanodes).

Jambalaya (Storey 2001) uses a graph metaphor to show connections between concepts via links. Concepts are similar to regions in NVSS. Concepts can contain sub-concepts, which resembles node aggregation (concepts to sub-concepts are similar to metanodes to nodes). Nodes can be placed manually or automatically by a structural property of nodes but not by a node attribute. Links can be color-coded according to their types, which is determined by their source and target classes.

Several approaches use multiscale visualization ((Auber 2003), (Archambault 2007), (Auber 2003)). Such applications usually are able to handle very large graphs and enable users to group nodes to metanodes and ungroup them. In Grouse (Archambault 2007), nodes are grouped into metanodes using hierarchy information and the layout is based on topological features that are computed from the graph structure. Users are able to open and close metanodes on demand and layouts are computed as needed as the user explores the network, which enables fast response for very large networks. Tulip (Auber 2003) enables users to manage clusters (group and ungroup nodes) as well, and although it supports node attributes, there is no layout strategy using node attributes. However, a plug-in capability allows defining new

layout algorithms. Link visibility based on node attributes also has not been reported. SocialAction (Perer 2006), designed primarily for social networks, uses a force-directed layout and visually surrounds clusters of nodes with a convex hull. The clusters are determined using Newman & Girvan's community algorithm, with interactive parameter control by users, on the network's structure and not the node's attributes. Each cluster can be collapsed to a metanode where the links become metalinks. Metanode size and metalink thickness represent the number of nodes and links they represent.

Several applications do not reduce nodes into metanodes but use visual clustering and provide filters to cope with the complexity. NicheWorks (Wills 1999) clusters nodes by placing them close to each other and doesn't reduce them to metanodes. It uses an initial layout (circular layout, hexagonal grid, and tree layout) and through incremental algorithms (e.g. steepest descent and simulated annealing) computes the final layout. NicheWorks was designed to visualize large graphs (up to 1,000,000 nodes) and supports filtering based on node attributes. Osprey (Breitkreutz 2003), a domain specific tool for biological researchers, also handles large datasets and provides node filters based on attributes. However, it doesn't provide layouts based on node attributes and doesn't reduce nodes into metanodes.

The node aggregation extension of the semantic substrate idea, is in principle, a clustering method. The difference is in the fact that none of these clustering approaches provide this functionality in the context of user-defined layouts based on node attributes. The node aggregation extension of the semantic substrate idea is unique in terms of providing the clustering functionality within the context of a

semantic substrate. Furthermore, the node aggregation extension is smoothly integrated with the semantic substrate approach. In other words, it does not interfere with the existing layout methods; on the contrary, it leverages them to provide the aggregated view.

2.4.2 Other Approaches

Other approaches that directly address scalability in network visualization consist of focus+context approaches, link aggregation, and link routing,

The use of hyperbolic geometry in networks (Munzner 1998; Lamping 2005) exemplifies focus+context techniques to cope with complexity. Although these provide greater detail for the focused areas, they may distort the overall view of the network, which leads to difficult navigation. These techniques do not aggregate nodes.

A special example to link aggregation is provided by Becker et al. (Becker 1995), where double-directed links between two nodes are reduced to a single link (visually represented by a straight-line) augmented by color- and thickness- coding. Links could be also drawn partially to reduce display complexity. Holten et al. (Holten 2006) provides an approach where links are spatially organized but not replaced with metalinks. The link aggregation is guided by node hierarchy information.

The primary example to link routing is provided Flow Map Layouts (Phan 2005), which reduce the number of links by combining common parts via edge-routing algorithms.

Approaches that indirectly address the scalability issue consist of the application of graph drawing aesthetics and matrix-based approaches. These help reduce the complexity of the display but are more limited than approaches that address the scalability issue directly.

The application of graph drawing aesthetics (Sindre 1993; Ware 2002) help reduce the display complexity of the network visualization. Principles, such as minimization of link crossings, help organize the display. However, these techniques seem to place nodes arbitrarily and not according to node attributes as in the semantic substrate approach.

Matrix-based approaches ((Ghoniem 2004; Henry 2006) provide an alternate view to the node-link diagrams. They reduce display complexity by avoiding drawing nodes and links in traditional ways. In such representations, the spatial structure of the network is hidden. MatrixExplorer (Henry 2006) combines matrices with node-link diagrams to give a sense of the spatial structure (not based on node attributes) and simplifies the display by keeping matrices for parts of the network. In addition, interactive operations, such as sorting matrix columns in terms of attributes and filtering, are provided.

The approaches in this section either do not aggregate nodes or do not use a node-link representation. In this respect, they are less related to the node aggregation extension of the semantic substrate approach.

2.5 Evaluation Methods

Plaisant (Plaisant 2004) groups the current evaluation practices for information visualization applications into the following groups:

(1) *Controlled experiments comparing design elements.* In this category design elements of applications, such as widgets are compared.

(2) *Usability evaluations.* These evaluations focus on problems of the tool that make it harder to use are reported.

(3) *Controlled experiments comparing two or more tools.* This is the most commonly used approach where a new technique is evaluated against existing approaches using many subjects.

(4) *Case studies.* These are the least common type of evaluation tools. These usually focus on a few users and work with tasks that they users define to solve their own problems. It is flexible and the results are relevant to the real tasks that users have; on the other hand, they are time consuming, and the results are specific to the few users chosen and dependent on their characteristics, such as domain, expertise, and interests.

Controlled experiments usually change one factor at a time while keeping all the other controllable factors the same. One example is where Plaisant et al. compare three visualization tools; SpaceTree, Hyperbolic Explorer, and Windows Explorer with an experiment design of 3 interfaces by 7 tasks (Plaisant 2002). Another paper that reports results of such an experimental evaluation compared five systems that visualized trees (Kobsa 2004). Such experiments usually measure performance, accuracy, and user satisfaction. Performance usually is defined by task completion time to indicate how efficiently the tasks can be completed using the tools by the experiment population. Accuracy measures the percentage of correct answers or correct outcomes subjects could produce. User satisfaction is usually measured with

surveys after the experiment. The result of such experiments can be statistically analyzed via methods, such as ANOVA (Analysis of Variance). The results may be affected by several factors such as the population selected; the tasks chosen; and the domain, size, and complexity of the datasets. Therefore, repeated experiments are often needed to increase the reliability of the outcomes.

The controlled experiments are limited in several ways:

(1) *They need to be short in time.* Long-term experiments are not feasible, difficult, not possible, or not reliable. Perer et al. (Perer 2008) report that out of 132 papers in the 2005-2007 IEEE Information Visualization and the 2006-2007 Visual Analytics Science & Technology Conferences, only 39 papers had any user evaluation and those lasted less than 2 hours of tool usage.

(2) *The set of tasks are limited when designed by the experimenter.* This limits the creativity of the users and possibly ignores some of their expertise in their domain and the tasks that are related to their domain. The outcome becomes less useful as the tasks are less representative.

(3) *Measurements are limited.* Usually performance, correctness, and user satisfaction are measured. However, the purpose of visualizing information extends beyond these measures. Exploration may involve other important goals such as hypothesis generation, creativity, arriving at refined questions from initial questions, and finding insights.

Gersh and Saraiya et al. (Saraiya 2004; Gersh 2006) provide examples of evaluations and the methodology to extend the scope of the evaluation to measuring

insights. However, these encompass short-term evaluations lasting only a few hours. Some of the tasks of researchers extend to weeks or months, which these approaches still fail to address. In addition, there is an initial learning time that researchers/users need to pass to have higher quality and deeper outcomes.

The MILC method provided by Shneiderman et al. (Shneiderman 2006) addresses all of these issues except that it primarily focuses on expert users. MILC stands for multi-dimensional, in-depth, long-term case studies, where the *multi-dimensional* aspect suggests multiple evaluation methods including "observations, interviews, surveys," the *in-depth* aspect suggests "an intense engagement of the researchers/evaluators with the expert users to the point of becoming a partner or an assistant," the *long-term* aspect refers to "longitudinal studies that begin with training in use of a specific tool through proficient usage that leads to strategy changes for the expert users," and *case studies* refers to the "detailed reporting about a small number of individuals working on their own problems, in their normal environment."

The semantic substrate approach is implemented in NVSS. The design of the semantic substrate depends on what users are interested in exploring in the dataset; in other words, it is more meaningful when users define the tasks. In this respect, a case study approach is more amenable. Also, a longer term, in-depth engagement is advantageous and often necessary because substrates may take several iterations to design. Moreover, several sessions or preparations may be needed to understand the dataset characteristics, which might be incorporated with the substrate design process. As a result, an approach similar to MILC is followed in evaluating NVSS. This approach also provided feedback to improve the design of NVSS.

The MILC method is successfully used to evaluate SocialAction (Perer 2008) recently, where a refined methodology is provided with 5 stages in the process (interview, training, early use, mature use, outcome) and to evaluate HCE (Seo 2006) formerly, which started with 6 case study participants and concluded with 3 of them. In these cases and in NVSS, domain experts work for days and weeks to complete exploratory tasks; therefore, a long-term and in-depth engagement is needed to also help the domain experts to use the tool incorporating the new technique. In addition, exploratory tasks are not to be defined by users but by domain experts. Moreover, many tasks are defined as the exploration progresses. Finally, diverse work styles are expected. For all these reasons and the ones mentioned before, an approach similar to MILC was employed to evaluate NVSS.

2.6 Summary of Previous Work

Many node placement techniques follow the graph drawing aesthetics principles, such as minimizing link crossings, listed in section 2.1. The review of 100 network visualizations (section 2.2) revealed that the majority ($1/3^{rd}$) of network visualizations of the set analysed use force-directed layouts that place nodes arbitrarily on the screen. Table 1 provides a summary of all node placement methods. Although many systems have elements of the semantic substrate approach, they either are specific to certain domains / datasets or they differ in other ways from the general approach that NVSS implements, the details of which are provided in section 2.3. The node aggregation mechanism addresses the scalability issue and enables larger datasets to be explored in the context of semantic substrates. Related work on network visualization scalability aims to reduce complexity of networks, as provided

in the subsections of section 2.4, via clustering approaches, focus+context approaches, link aggregation and routing, matrix-based representations of networks, and the application of graph drawing aesthetics. While the node aggregation extension of the semantic substrate approach is a clustering method, no other methods provide the clustering functionality in the context of user-defined layouts based on node attributes. The other approaches are less related as they either do not cluster nodes or represent the network via a matrix rather than a node-link diagram. Finally, a long-term case study approach is used to evaluate NVSS (the tool that uses the semantic substrate approach). This approach is compared with other approaches and reasons for choosing this approach over others are provided. To summarize, due to the lengthy nature of the exploration process with expert users, long-term and in-depth engagements are necessary to have more meaningful and effective results. The results are less general as it is restricted to a few users. However, this approach enables the tasks defined by users. Therefore, the outcome is more relevant to real users.

Chapter 3: Design Issues

The following sections describe and discuss the design choices made for the semantic substrate approach. These design choices are illustrated in NVSS (the tool that uses the semantic substrate approach). Section 3.1 describes the design choices for the visualization and exploration of network data by illustrating it in the NVSS Visualization Module. Section 3.2 describes the design choices for creating a semantic substrate and illustrates it using the Substrate Designer module of NVSS. Finally, section 3.3 describes the design choices for the node aggregation feature and illustrates it in NVSS.

3.1 Semantic Substrates

Semantic substrates first group nodes into regions and then place them within regions based on node attributes. The third (and final) step is to provide link visibility filters to restrict the number of links on the display.

Figure 14 shows a semantic substrate with 3 regions. The regions are chosen to be rectangular although, in principle, they could have other shapes, as well. However, a rectangular shape was the simplest form to implement; and therefore, this shape was chosen. In addition, it would facilitate the use of axes.

The triangles represent the nodes in the dataset that is visualized. This dataset is a legal precedent dataset. Nodes represent court cases, while links are citations from a court case to another. Nodes have attributes such as *year*, *courtType*, *name*, *inCites*, and *outCites*. c*ourtType* has values of "Supreme," "Circuit," and "District." *InCites* represents the number of times a case has been cited, while *outCites* is the

number of times it cites other cases. *name* is the name of the case, while *year* is the year the case was heard.

Nodes are grouped using the *courtType* attribute. The top region has the nodes that have attribute value "Supreme," while the other two have "Circuit" and "District."

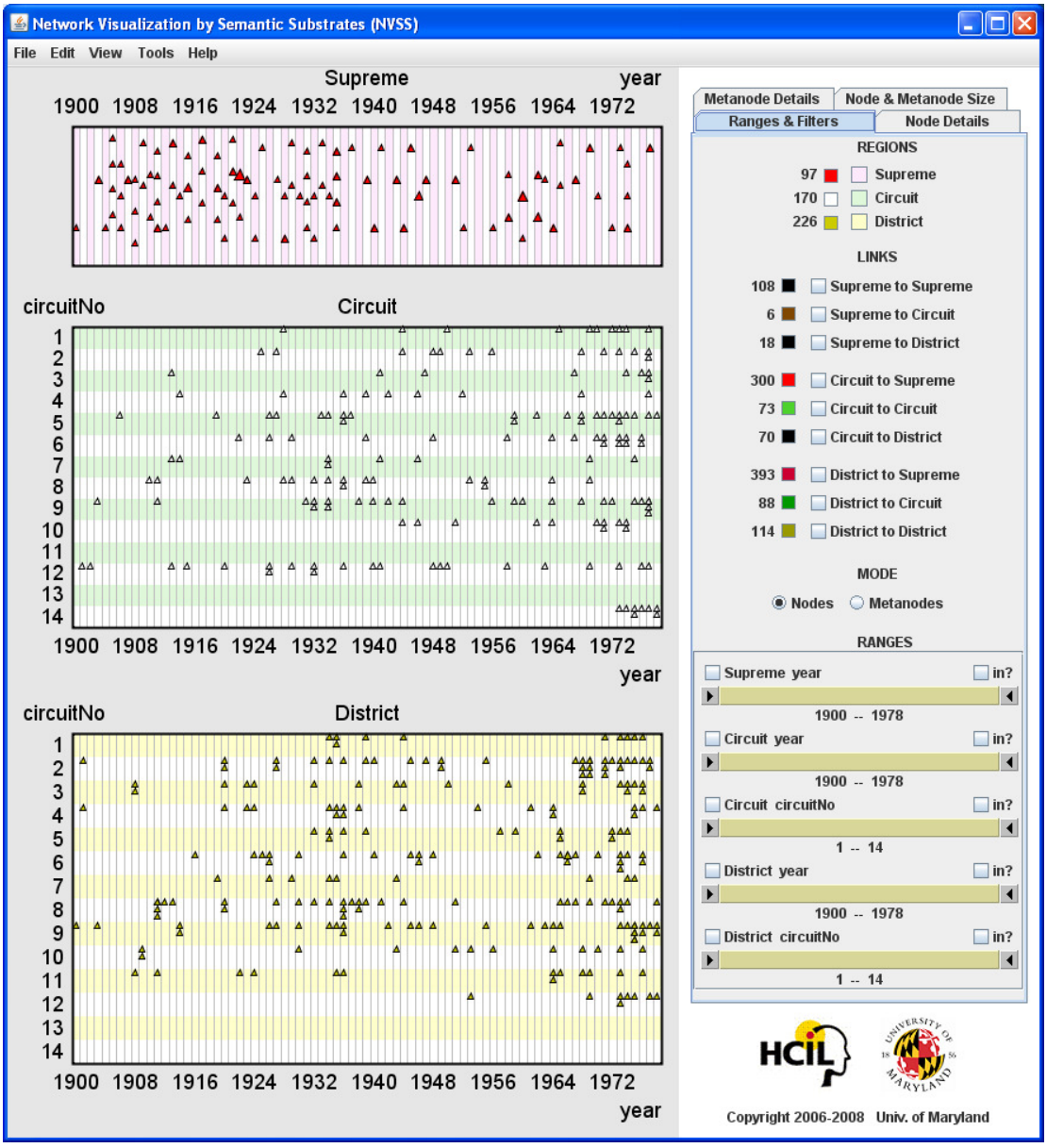


Figure 14 A semantic substrate with 3 regions.

The right-hand side is the control panel of NVSS. The top of the control panel also lists the regions (under the "REGIONS" heading). For each region, from left to right the following is provided: The number of nodes contained within this region, the color of nodes in this region, and the background color of this region (e.g., for the "Supreme" region, the values are 97, dark red, and pink (in a grayscale copy of this dissertation, the dark red will appear as dark gray and pink will appear as light gray)).

The second step is to place nodes using node attributes. In this step, the concept of *placement method* is used to describe the way the nodes are placed. In Figure 14, a placement method is used, which represents the values of certain node attributes using both x- and y-axes. This placement method is called the GridPlotXY placement method. The method takes node attributes as parameters to define the x- and y-axes. In this sense, the placement method is general and it can be used with any dataset (as long as attributes can be provided for x- and y- axes). The placement method also takes binning parameters to determine how many groups to make on the x- and y-axes. In the example substrate in Figure 14, each bin has only one value (e.g., the year 1900, for the first bin on the x-axis); however, bins can group more than one attribute value. Complete details of binning are provided in the next section (section 3.2).

Other placement methods are implemented as well. The complete set of placement methods are as follows:

- *GridPlotX:* Only the x-axis is used. The method takes an attribute as a parameter to define how to represent the x-axis. The y-axis is free. Nodes are placed equidistant from each other along the y-axis. Their order is arbitrary.

- *GridPlotX Jittered:* This is the same as GridPlotX placement method. The only difference is that nodes are jittered along the y-axis. The jittering moves all nodes having the same x-slot up or down, alternating.
- *GridPlot Y:* This is the same as the GridPlotX placement method except that y-axis is used instead of the x-axis.
- *GridPlot Y Jittered:* This is the same as the GridPloy X Jittered placement method except that y-axis is used instead of the x-axis.
- *GridPlotXY:* Both x-axis and y-axis are used. The method takes an attribute for each as parameters to define the representation of the axes.

These placement methods all use x- or y-axes. However, in principle, placement methods could use any strategy to place nodes based on node attributes. NVSS could be extended to accommodate other types of placement methods.

In Figure 14, the GridPlotXY placement method for each region uses the *year* attribute for the x-axis and the *circuitNo* attribute for the y-axis. The *year* attribute ranges from 1900 to 1978 (with 79 bins, each bin containing 1 year) and the *circuitNo* attribute ranges from 1 to 14 (with 14 bins, each bin containing 1 circuitNo). In the example, the placement attributes of all regions are the same. However, they could be different, in general.

The third step is to provide link visibility filters. The filters are provided on the control panel (Figure 14). Two types of filters are provided: link filters (under the "LINKS" heading) and range filters (under the "RANGES" heading).

The link filters filter the links according to their source and destination regions. This way links are categorized in terms of the grouping attributes. Users can

select source and destination region(s) and visualize only those links they are interested in seeing. This design choice was made so that the grouping attribute is being used not only to group links but also to filter them. (The grouping attribute is the attribute used to group nodes into regions.)

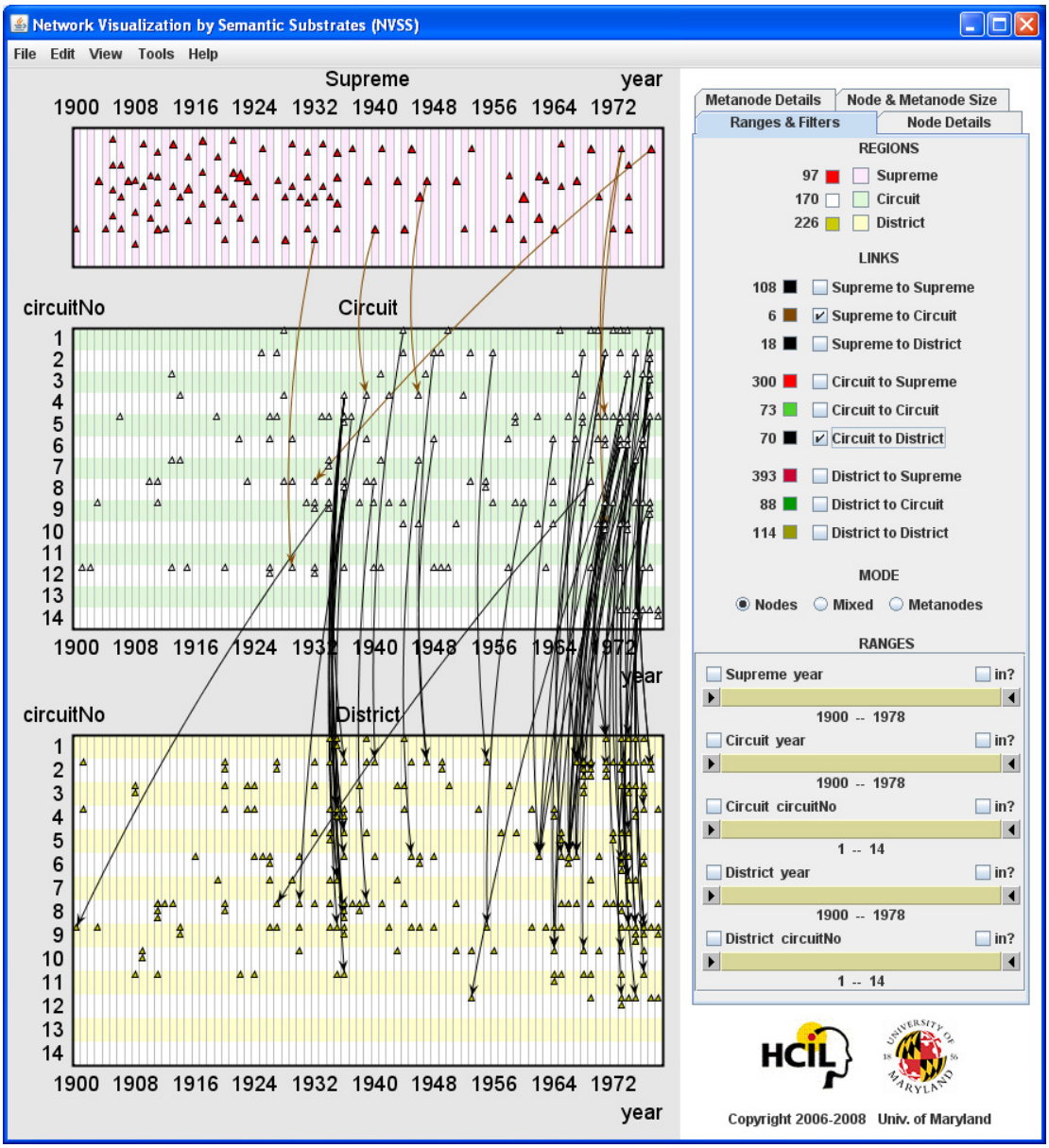


Figure 15 Links from the Supreme to Circuit and from the Circuit to the District regions are enabled.

In Figure 15, the links from Supreme to Circuit and from Circuit to District regions are enabled by clicking the corresponding checkboxes. A checkbox is provided for each combination and organized by first source and then destination region so that users can easily find them to enable or disable the links. The number of links each filter enables is available on the left of the checkbox (e.g., 6 Supreme to Circuit links in Figure 15).

The range filters are based on the placement attributes within each region. (The placement attributes are the attributes used by the placement method.) This way an association is created between placement of nodes and filtering the links connected to them.

In Figure 16, two range filters on the Circuit region are used to restrict the links. One of the filters is on the *year* attribute and restricts the period to 1973-1978 and the other filter is on the *circuitNo* attribute and restricts the circuit to the 2nd Circuit Court. (In the resulting view, links from this selection point to the 2nd, 6th, and the 9th circuit cases in the District region)

Filters are inspired by the range filters of Spotfire and the early ideas of using dynamic queries on starfield displays (Shneiderman 1999). These filters are using an AND logic. NVSS also adopted this strategy. However, users frequently have asked for the OR functionality. Although this has not been implemented in NVSS, it is a future possibility (included in future work, section 7.1.5). The filters limit the links to the outgoing links from the intersection area defined by the range filters. Checking the “in?” checkbox switches the functionality to the incoming links.

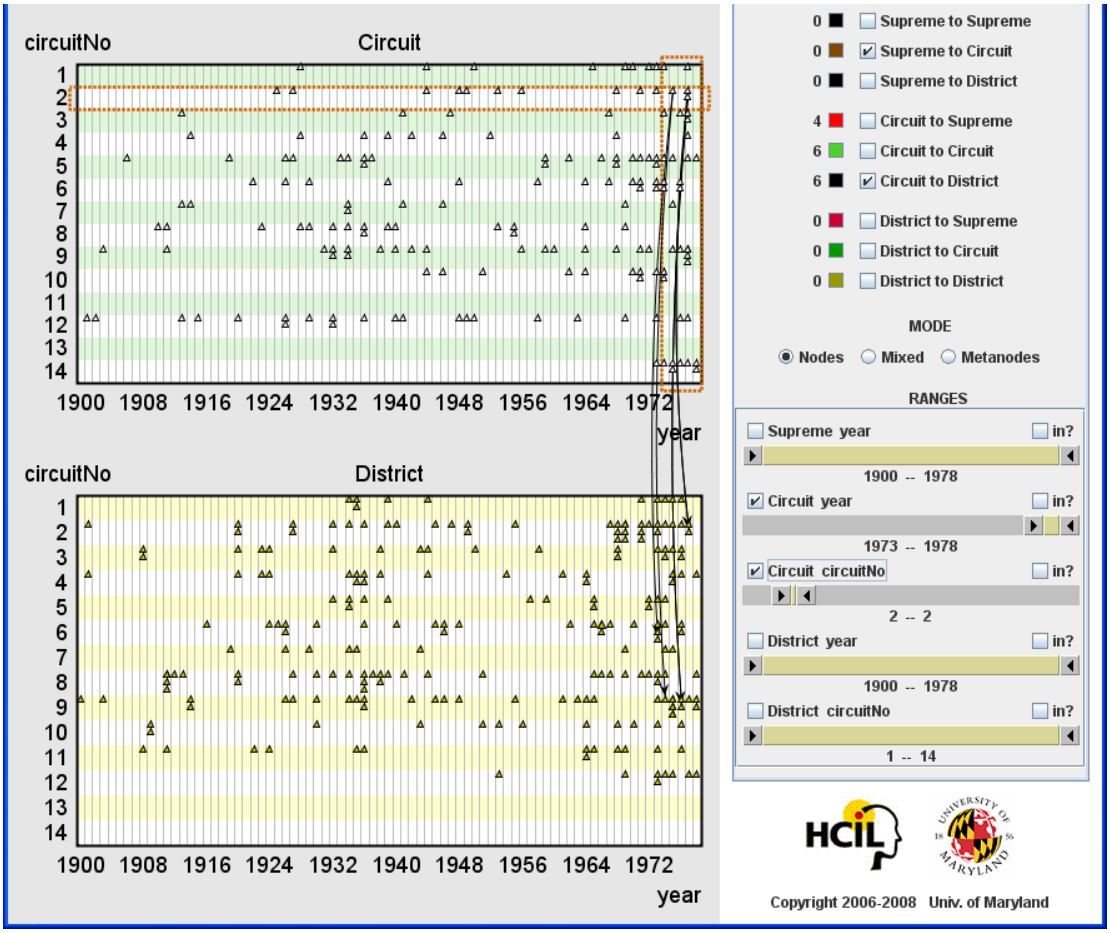


Figure 16 Range filters on circuitNo and year for the Circuit region are used to restrict the links to the 2nd Circuit Court during the years 1973-1978. (Only the bottom part of the display is shown.)

The semantic substrate approach is open to include other types of filters, such as link filters for non-placement attributes. There are numerous possibilities that have not been implemented in NVSS.

3.2 Substrate Designer

Substrates in the semantic substrate approach (as exemplified and described in section 3.1) are user-defined. NVSS has a module called Substrate Designer to enable users design a semantic substrate. For complete details on creating substrates, see The

process of creating a substrate in NVSS. After a semantic substrate is created, it is applied to a dataset and the dataset is visualized via the NVSS Visualization Module.

In order to create a substrate, a Data Model file is needed. This file contains the set of node attributes. This design choice was made so that a substrate could be reused by multiple datasets having the same set of node attributes. The file also expects *attribute labels*, which are used by the NVSS Visualization Module whenever the name of the attribute needs to be displayed. For example, if the attribute name is "circuitNo", and the attribute label is defined to be "Circuit Number", "Circuit Number" will appear (instead of "circuitNo") on the range filters if used as a placement attribute. This helps improve the readability of the display in certain cases while keeping the original / shorthand notation of attribute names within the datasets for direct association with the original source of the data. Since the Data Model file is a readable TAB delimited text file, it can be edited manually (Figure 17).

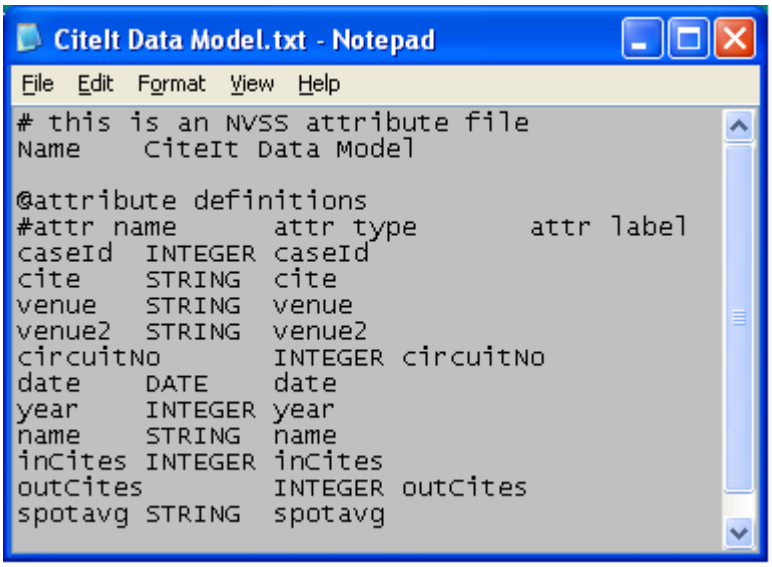


Figure 17 A sample Data Model file.

NVSS Main is the main module that takes a substrate and a dataset and launches the NVSS Visualization Module. To do this, users load a substrate (by pressing "Load"), specify the nodes and links files (these are the dataset files) and then press the "Create Graph" and "Launch" buttons (Figure 18).

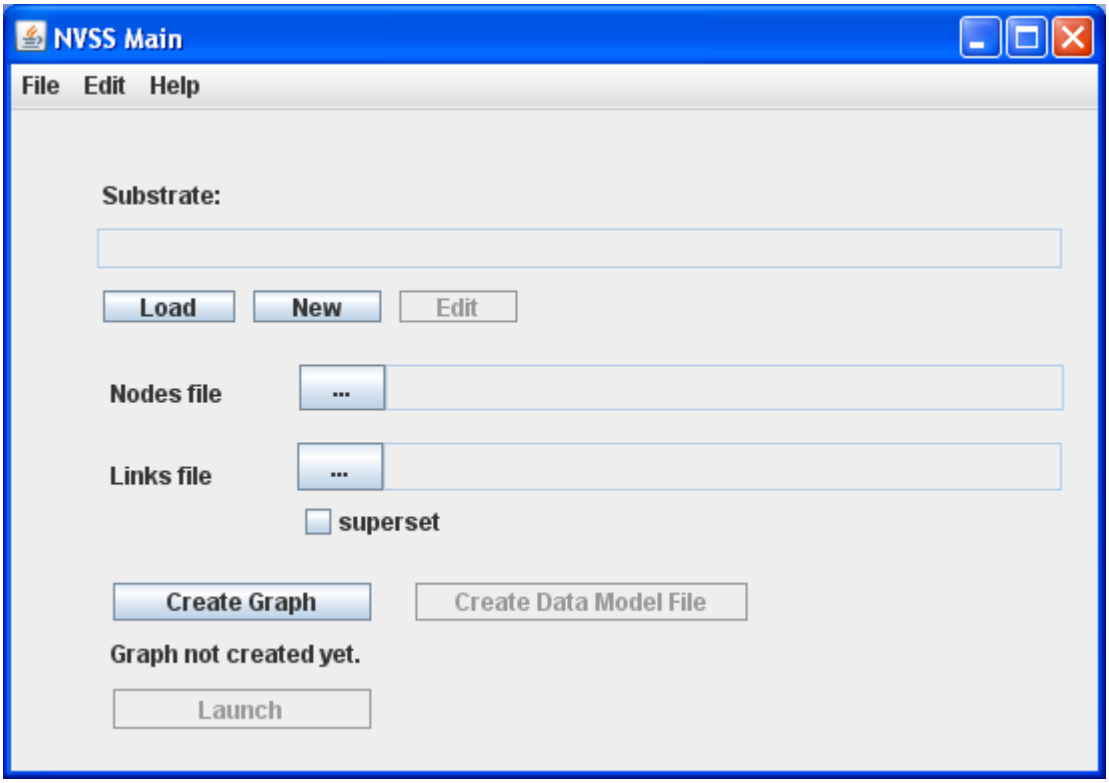


Figure 18 NVSS Main.

This way, the substrate is kept separate from the dataset. The dataset is represented by two files (nodes and links). The format of these files are illustrated in Figure 19 and Figure 20. The nodes file contains the attribute names, the attribute types, and the attribute values of all nodes. The first attribute is used as a key (in database terms) to be used in the links file.

	A	B	C	D	E	F	G
1	caseId	cite	venue	venue2	circuitNo	date	year
2	INTEGER	STRING	STRING	STRING	INTEGER	DATE	INTEGER
3	4003	94 S.Ct. 2291	Supreme	NULL	0	6/10/1974	1974
4	4004	54 S.Ct. 599	Supreme	NULL	0	4/2/1934	1934
5	4005	1930 WL 1063	District	Territory of	9	4/4/1930	1930
6	4006	55 S.Ct. 333	Supreme	NULL	0	1/14/1935	1935
7	4007	565 F.2d 338	Appeals	Fifth Circu	5	12/27/1977	1977

Figure 19 Example nodes file to be used as input to NVSS.

	A	B	C
1	caseId1	caseId2	dateDiff
2	INTEGER	INTEGER	DOUBLE
3	4003	4004	40.21644
4	4003	4291	53.39178
5	4004	4291	13.17534
6	4005	4291	9.178082
7	4006	4004	0.786301
8	4006	4291	13.96164

Figure 20 Example links file to be used as input to NVSS.

The choice of defining the dataset in terms of nodes and links files helps to keep the format of these files simple and allows users to easily import from databases into spreadsheet formats. Once the data is in a spreadsheet format, with little effort, it can be modified to conform to the format that NVSS accepts. In addition, this allows the links file to be modified, which allows easier outside link filtering. Outside link filtering is helpful to explore large datasets that are slow to visualize in NVSS.

NVSS expects adjacent nodes of every link to be present. To allow outside node filtering without updating the link files (to eliminate links that become invalid), a “superset” checkbox is provided (Figure 18). When checked, NVSS ignores the invalid links.

“Create Graph” will report any inconsistencies among the substrate’s Data Model, nodes file, and the links file.

To create a substrate, users click the "New" button in NVSS Main (Figure 18). NVSS Main will require the location of the data model file. The Data Model becomes an integral part of the substrate, which is visible on the left hand side of the Substrate Designer (see Figure 21, first line, where it says "Data Model" and "CiteIt Data Model." In this case, the name of the data model is "CiteIt Data Model", which was specified in the Data Model file.).

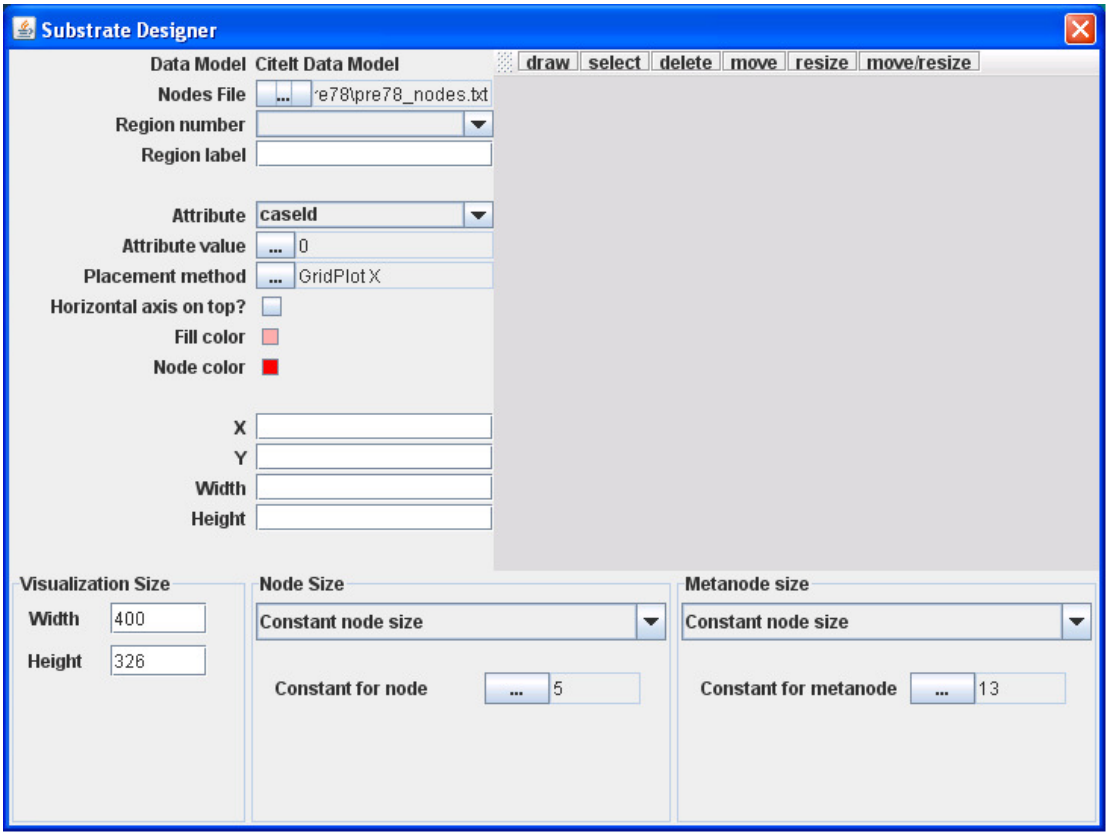


Figure 21 New substrate after pressing "New" and loading the Data Model from a file.

The Substrate Designer (Figure 21) consists of 3 panels. The panel on the top right is where users can visually define regions, i.e., their location and size properties. The buttons on the top allows users to change the "mode" of operation. These buttons behave the same way as radio buttons. To create a region, users press the "draw" button. Once in the "draw" mode, the first click of the mouse defines the upper left

corner of the region and the second click after a drag defines the bottom right corner (Figure 22). To alter a mode, users simply click another button at the top.

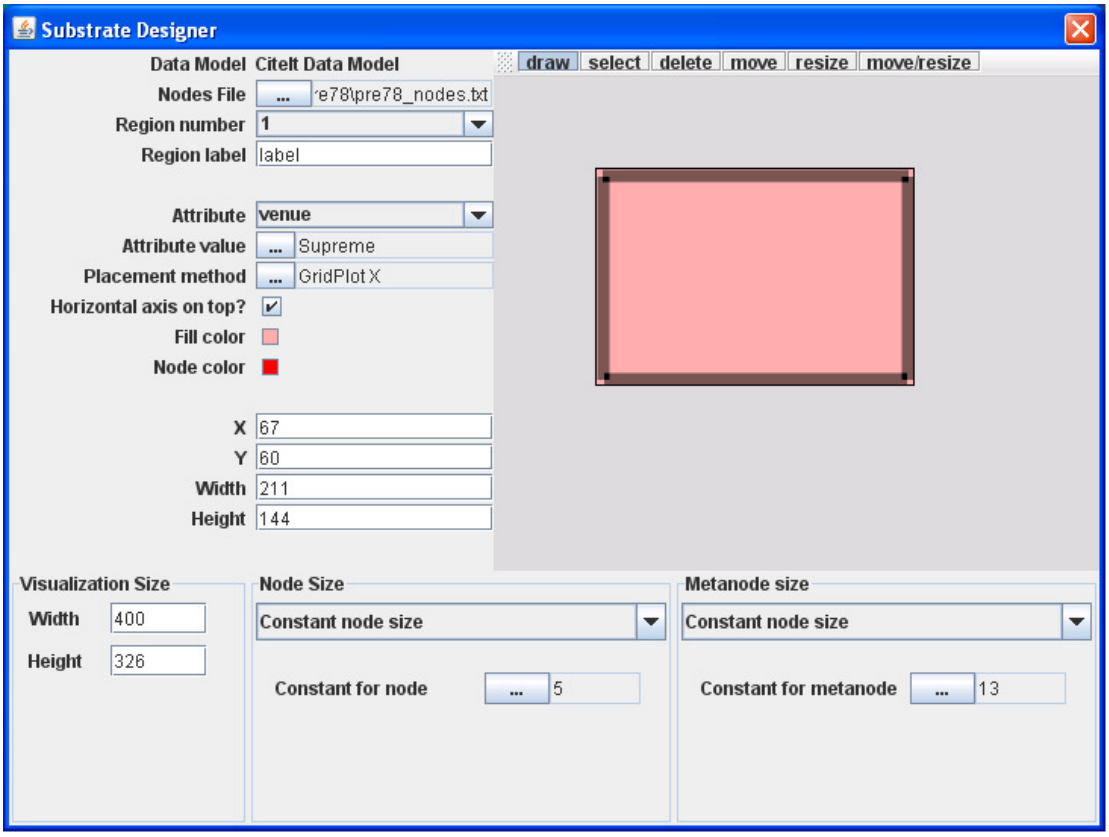


Figure 22 Creating a region in "draw" mode in the Substrate Designer.

When a region is created, its visual properties are assigned in the details view (Figure 22). These are X and Y, which determine its location, and Width and Height, which determine its size. The default fill color and node color are used once the region is created. To modify colors, users click on the color and select the desired color from a new window providing color choices.

The newly created region is assigned a region number (in this case, it is “1”) and its default region label is “label” (Figure 22). Users type a new label to override

the default region label. The label of a region appears at the top centered in the visualization (not displayed in the Substrate Designer).

"Attribute" on the left panel determines which attribute will be used to place nodes within this region. In Figure 22, the *venue* attribute is selected. The "Attribute value" determines the attribute value that the nodes within this region will have for the selected attribute. In this case, it is selected to be "Supreme." As a result, nodes having attribute value "Supreme" for the *venue* attribute will be placed within this region.

To set the placement method within a region, users need to press the "…" button next to "Placement method", which opens the Placement Method Selector dialog (Figure 23). As previously described in section 3.1, NVSS supports five placement methods:

- GridPlot X
- GridPlot X Jittered
- GridPlot Y
- GridPlot Y Jittered
- GridPlotXY

The Placement Method Selector enables users to set the placement method, and specify the node attributes for placement (as parameters; e.g., "Attribute along X-axis" is defined as *year* in Figure 23).

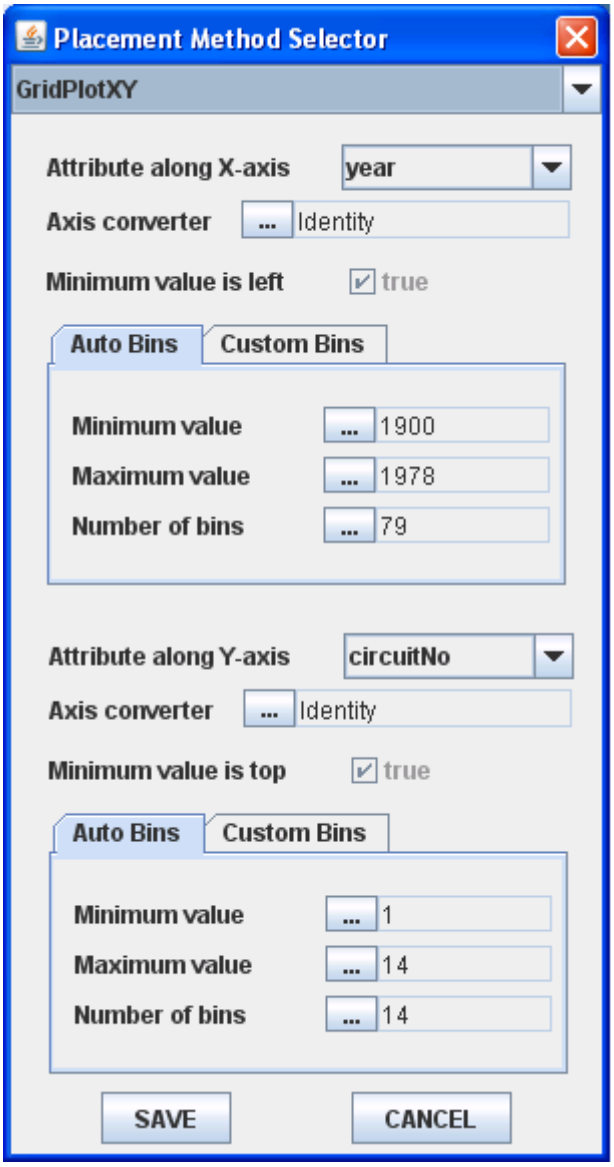


Figure 23 Placement Method Selector launched from the Substrate Designer.

After selecting a placement method and specify its attribute parameters, a binning strategy is chosen: "Auto Bins" or "Custom Bins."

In the "Auto Bins" strategy, users set the `Minimum value`, the `Maximum value`, and the `Number of bins` (Figure 23). The minimum value is the left (smallest) value in the first bin and the maximum value is the right (largest) value in the last bin. The number of bins determines how many bins there will be between the

minimum and the maximum values (these values are inclusive). For instance, with a minimum value of 1, maximum value of 100, and 10 bins, the left value in the first bin will be 1, the right value in the tenth (last) bin will be 100. The first bin will contain the values 1..10, the second bin will contain 11..20, and so on.

In the "Custom Bins" approach, users provide the boundary values for bins. The boundary values are the left values of all bins in increasing order and the right value of the last bin. For instance, with boundary values "1,5,7,10", there will be three bins, where the first bin will contain values 1..4, the second bin will contain 5..6, and the last bin will contain 7..10.

The "Auto Bins" approach facilitates quick definition of bins for evenly spaced groupings. On the other hand, the "Custom Bins" approach allows users to define unevenly spaced groupings. This option is helpful to visualizing data when there are gaps that take space on the display when spaced evenly (i.e., when using the "Auto Bins" approach).

For each placement attribute, an axis converter is used, where the default one is the identity axis converter (see Figure 23, "Axis converter"). The axis converter provides a mapping from the values in the dataset to STRING values to be presented on the display. For example, the circuit numbers 12, 13, and 14 in the earlier example of section 3.1 (see Figure 14) actually stand for the DC, Federal, and Temporary Circuit Courts, respectively. By providing an AVC that maps these appropriately (see Appendix A for details), the view in Figure 24 can be obtained. The mapped values are visible both on the axes (in the visualization) and on the labels of the range filters

(on the control panel). This enables meaningful presentation without modification of the dataset.

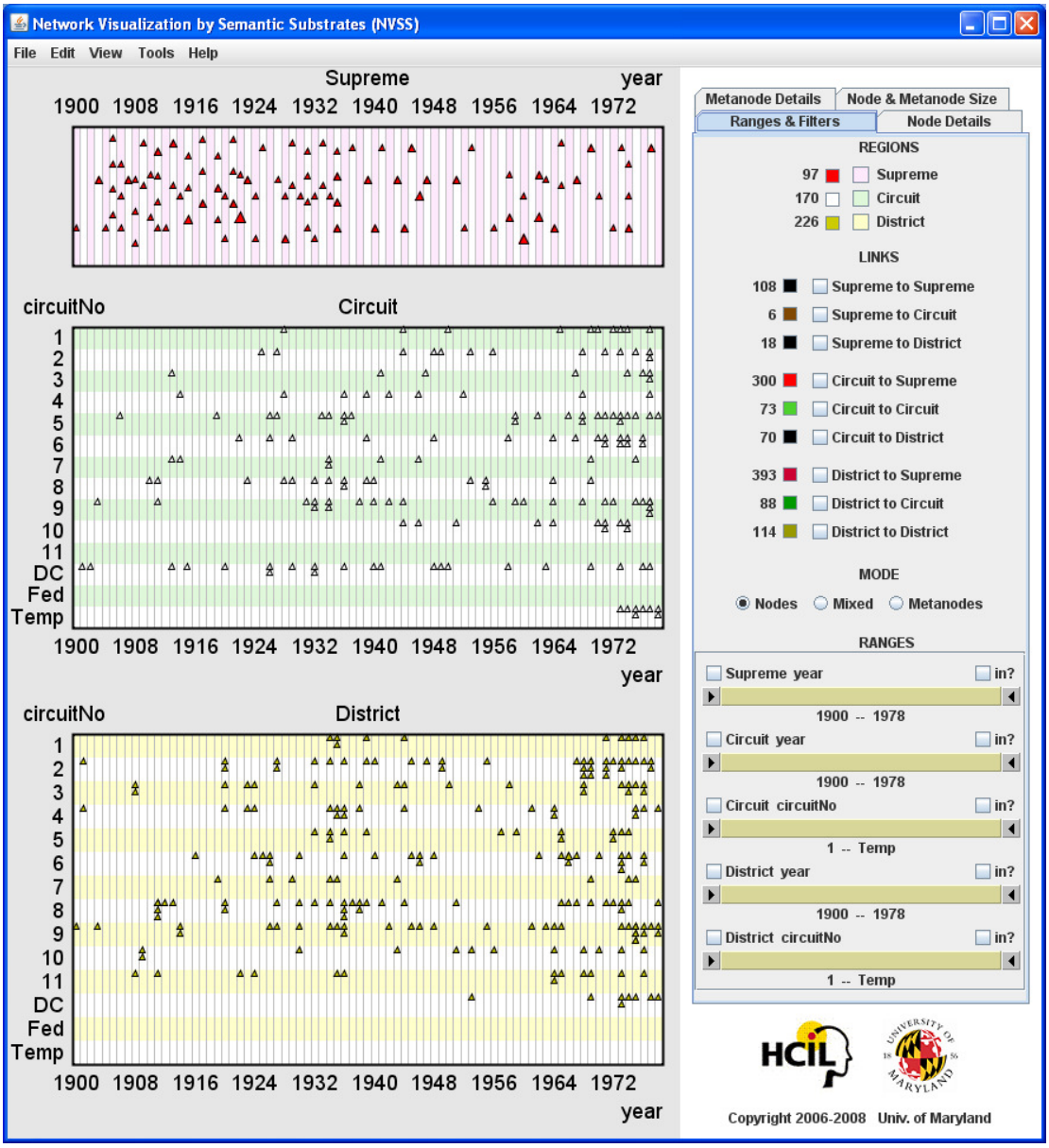


Figure 24 DC, Fed, and Temp has replaced the circuitNo values 12,13, and 14.

The Substrate Designer also allows for node size coding (the metanode size coding will be explained separately in the next section). NVSS allows users to specify the node size as constant (e.g., 5) or in terms of the value of a node attribute. A transformation function can be applied along with scaling and translation, which is in

the form of the formula y = m * f(x) + n, where x is the attribute value to be transformed, f(x) is the transformation function, m is the scale, n is the intercept, and y is the transformed value, which will be used directly to determine the size of the node. In Figure 25, the *inCites* attribute is used to determine node size, while a transformation y = 0.2 * sqrt(X) + 5 is applied. Currently, the only transformation function available is sqrt(X); however, the modular approach in NVSS will allow adding other transformation functions if needed in the future (new code will need to be written; but it will be possible to add to the existing code base without affecting other parts of the code).

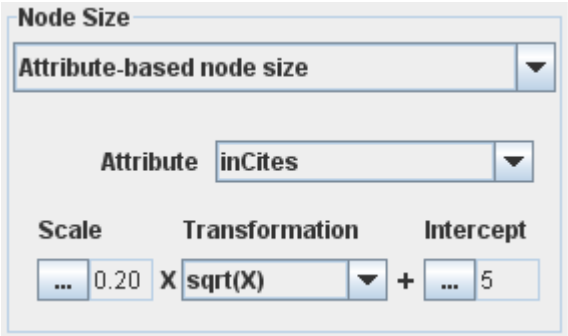


Figure 25 "Node Size" section of the bottom panel of Substrate Designer in NVSS.

The transformation feature is useful as many times attribute values do not correspond to screen pixel amounts and they need to be transformed, scaled, and translated. The transformation adapts the attribute values to reasonable pixel amounts on the display.

Although the Substrate Designer doesn't allow users to specify the link colors, users can specify them in the NVSS Visualization Module and save the modified substrate via the menu.

3.3 Node Aggregation

The node aggregation feature is an extension on the semantic substrate approach to help visualize large graphs. Large graphs are hard to visualize for two reasons. The first is that the performance decreases below a level that allows smooth interaction. The second reason is that there is not enough space on the display to fit the large number of nodes, which lead to overlaps and occlusions. The node aggregation extension helps with both of these problems while maintaining the benefits of the semantic substrate approach.

The node aggregation extension introduces a 2nd mode of visualization in which nodes that have the same placement attributes are aggregated, that is, they are replaced by a single node. This single node is called a *metanode* and it represents the nodes that it has replaced. Accordingly, links are replaced with *metalinks*, which represent the links they replaced.

In NVSS, the node aggregation extension is implemented by a 2nd mode of visualization in the NVSS Visualization Module named as "the metanodes mode." The normal mode (the un-aggregated view) is named as "the nodes mode." The modes are provided as options (radio buttons) to the user in the control panel of NVSS under the "MODES" heading (Figure 26).

There is also a third mode (not shown in Figure 26), named as "the mixed mode," in which aggregation of groups is performed on a group basis. A group of nodes (having the same placement attribute values) are aggregated only if the available space is not sufficient; otherwise, they are not aggregated. This mode has

been found to be confusing and not supporting user tasks well. As a result, it has been made optional. It can be enabled or disabled via a menu option.

The dataset in Figure 26 is a document citation dataset, which contains nodes representing authors, documents, and keywords and links between these (for complete details refer to the TobIG Case Study in section 5.2.3). The dataset contains 4,296 nodes and 16,385 links. With this large number of nodes, both the Document and Keyword regions have many node occlusions.

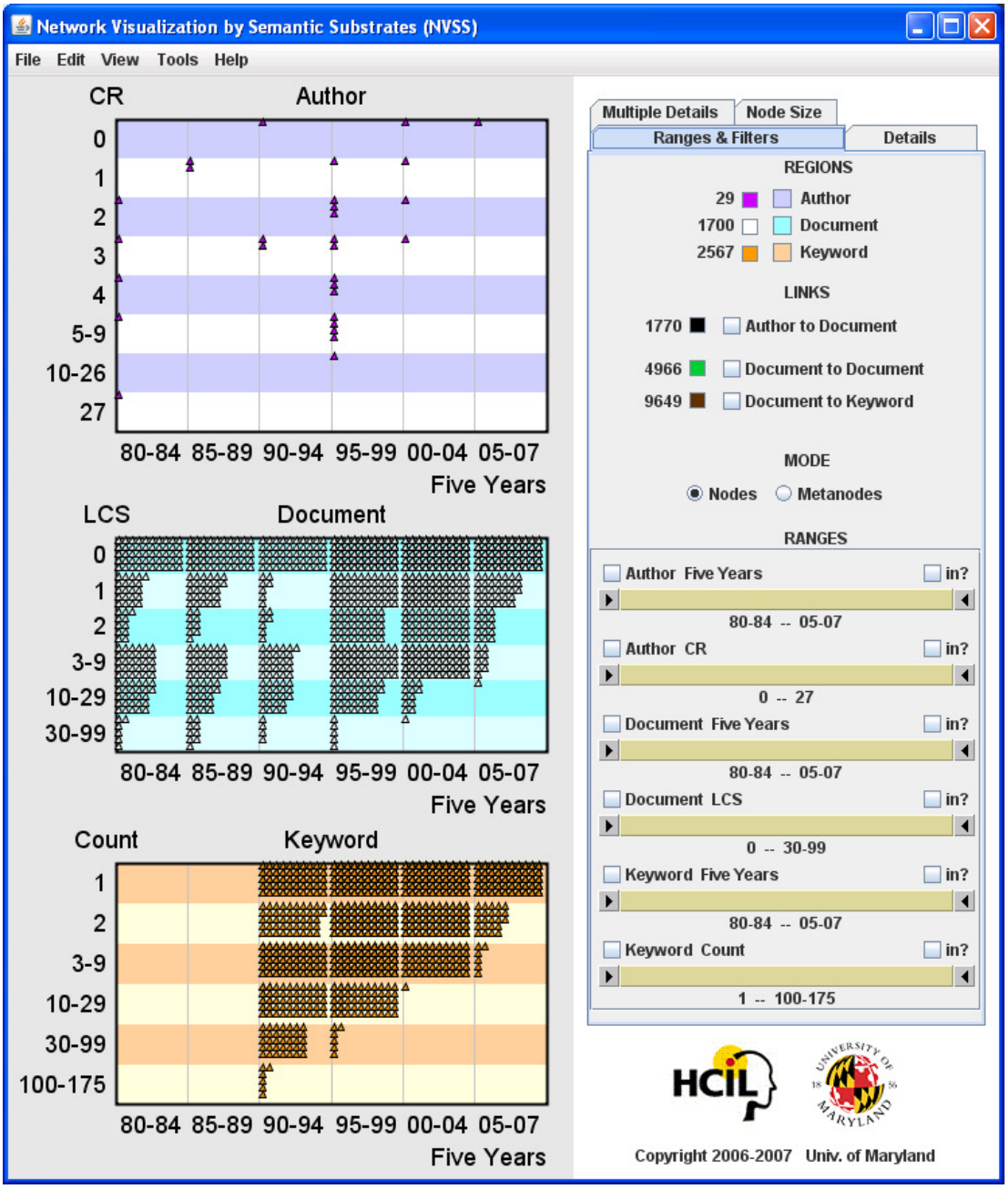


Figure 26 A dataset that contains 4,296 nodes and 16,385 links causes many node occlusions in the Document and Keyword regions.

Figure 27 shows the 9,649 Document->Keyword links on the left hand side, which reduce to 221 metalinks on the right hand side. The 4,267 nodes in the Document and Keyword regions reduce to only 53 metanodes (Figure 27).

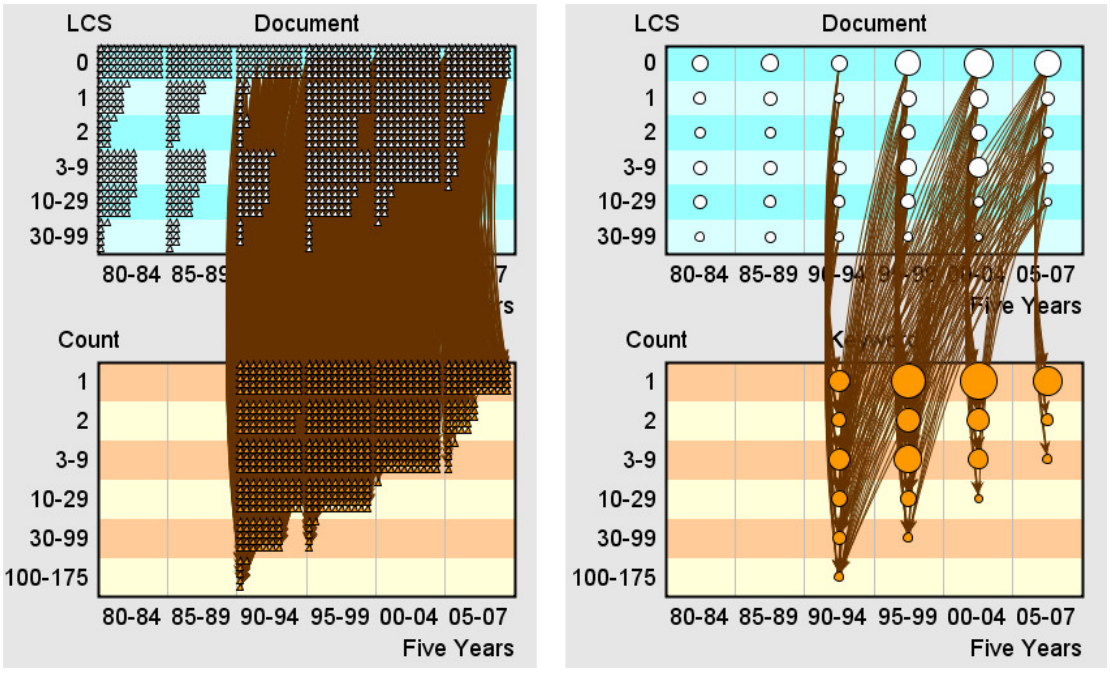


Figure 27 Switching the aggregation mode to the "metanodes" mode reduces the 9,649 Document->Keyword links to 221 metalinks.

Users can use the link and range filters that they used in the nodes mode with the same functionality in the metanodes mode. The only difference is that the filters operate on the metanodes and metalinks instead of the actual nodes and links. This enables filtering on the "overview" of the graph. The consistency of the semantic substrate approach is preserved both for the placement of metanodes and filtering of the metalinks. After links are filtered in the metanodes mode, users can switch to the nodes mode for more details, i.e., to see the actual nodes and links, and their counts (categorized by region) on the control panel.

The Substrate Designer allows for metanode size coding as it did for node size coding. The same settings as the node size coding are available for metanodes. The attribute-based size coding is adapted to metanodes by providing an aggregation strategy. NVSS supports "Sum" and "Average." For example, if "Sum" is chosen, the

sum of the attribute values of the aggregated nodes is used. Metanodes can also be size coded by the number of nodes they represent. NVSS provides this option with the transformation functionality (Figure 28).

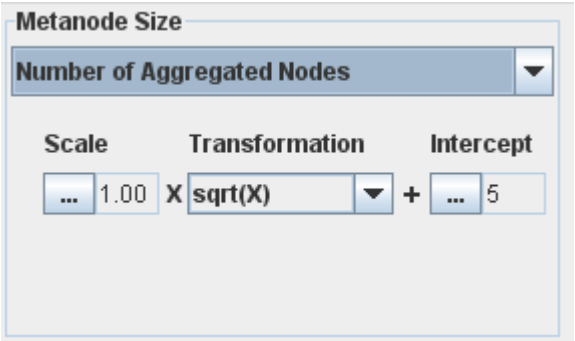


Figure 28 "Metanode size" section at the bottom of Substrate Designer in NVSS.

3.4 Summary of Design Issues

This chapter provided the design choices for the semantic substrate approach and illustrated them in NVSS. Some important design choices include the following.

When grouping nodes into regions, rectangular regions are used. This helped also having axes in the regions defined by the placement methods. Placement methods that are implemented include axes to represent attributes and provide binning. Filters are categorized by regions, which makes the filtering associated with placement of nodes.

The semantic substrate is kept and stored separately from the data files. This allows substrates to be reused for many datasets having the same set of node attributes (and also conform to the binning parameters). The nodes and links are kept in separate data files to allow use of subsets after external filtering. The Substrate Designer of NVSS provides visual controls to design a substrate and specify node

size coding depending on node attributes. NVSS Main is a gateway to the Substrate Designer, where the substrate is designed or edited, and allows a substrate to be loaded and then applied to a dataset to launch the NVSS Visualization Module, where the data can be explored.

Node aggregation extends on the semantic substrate approach to help explore larger datasets. It provides overviews with aggregated nodes and links and performance at the level of the aggregated dataset. Nodes that have the same placement attributes are aggregated, which preserve the benefits of the semantic substrate approach: placement and filtering. Users can switch to the un-aggregated view for details (possibly after filtering in the aggregated view). Node size coding extends to the aggregated nodes consistently by adaptation of features.

Chapter 4: Implementation Details

This chapter provides implementation details of NVSS, the software developed to explore the semantic substrate approach. Section 4.1 provides the data structures, algorithms, and the modular design of NVSS while section 4.2 provides details on system performance. Section 4.3 summarizes the chapter by highlighting important details.

4.1 Data Structures, Algorithms, and Modular Design

This section provides the data structures, algorithms, and the modular design of NVSS. Section 4.1.1 provides a comprehensive overview, while section 4.1.2 provides statistics on the implementation and gives an account of the history of NVSS development. The following sections provide details on certain parts of the NVSS implementation. Section 4.1.3 and 4.1.4 explain the class hierarchy for the placement methods and the algorithm memento (details specific to a region produced by a placement algorithm), respectively. Section 4.1.5 provides implementation details for the node aggregation feature and section 4.1.6 has details for miscellaneous issues.

4.1.1 Overview

NVSS has three modules: NVSS Main, the Substrate Designer, and the NVSS Visualization Module. Using the Substrate Designer, users generate a substrate and save it as a file. NVSS Main provides users an interface either to launch the Substrate Designer or the NVSS Visualization Module. NVSS Main takes the input dataset (it takes the nodes file and links file, then generates an internal graph structure) and the

semantic substrate and gives it to the NVSS Visualization module. The NVSS Visualization Module displays it for visualization and exploration.

NVSS uses JUNG[3] (O'Madadhain 2008) to create, maintain, and visualize the underlying graph of the visualized network and uses Piccolo.Java (Bederson 2004) for the visual interaction in the Substrate Designer module.

From NVSS Main, users can create new substrates, load, and edit existing substrates as explained in detail in Appendix A.

A substrate contains the following:

- Settings for each region (RectRegionSettings).
- The size of the visualization.
- The data model (contains the set of node attributes and attribute labels).
    - An attribute contains a name and a type. The available types are INTEGER, DOUBLE, DATE, and STRING.
    - The attribute labels are displayed on the screen in the place of the names of the attributes. The attribute names are used internally and in the substrate designer.
- The link colors.
    - The default link colors are provided by the Substrate Designer and modified from within the NVSS Visualization Module by users. Users save the substrate during visualization to have the link colors updated.
- The node size definitions for nodes and metanodes.

[3] JUNG stands for Java Universal Network/ Graph (JUNG) (http://jung.sourceforge.net/) Framework, an open source software library, widely used by network visualization researchers.

- Node sizes can be constants or attribute based. When they are based on an attribute, users can choose a transformation function to be applied. The transformation function is in the form of y = mx +n, where m is the scale and n is the translation (or intercept). These values could be defined in terms of floating point numbers by users in the Substrate Designer and also are available to view and to temporarily modify and apply within the visualization instance.
- Metanode size can additionally be coded by the number of the nodes represented by the metanodes. In this case, a transformation could be applied, as well.

The region settings (RectRegionSettings) contains settings for a region, which are as follows:

- The region attribute and the value (to select the nodes that will fall into this region).
  - For example, the attribute could be "courtType" and the value could be "Supreme."
- The region label.
  - This label is displayed on top of the region in centered form during the visualization.
- The placement method (RectRegionAlgo).
  - This is one of GridPlotX, GridPlotXJittered, GridpPlotY, GridPlotYJittered, and GridPlotXY.

- The node color (nodeColor).
- The background color of the region (fillColor).
- The coordinates to determine location and size (x, y, width, height).

For every placement method there is a separate Java class containing the algorithm to place nodes within a given region (e.g. GridPlotXYRectRegionAlgo.java for the GridPlotXY placement method) and a memento (e.g. GridPlotXYRRAlgoMemento; memento is a design pattern (Gamma 1994)). This makes NVSS extensible in the sense that for every new algorithm, one will need to write an algorithm and a memento class, which are independent from the other algorithms or mementos. One may need to refactor (Fowler 1999) and extend certain features of NVSS if the new algorithm is very different from the existing ones in order to support an uncommon feature.

The algorithm class is designed to be free of state so that it could be reused once it is created regardless of the current specific UI characteristics. The memento class is to hold the current specific UI characteristics and state such as the locations of the axis labels (which change when the application resizes). Therefore, the memento (but not the algorithm) is regenerated every time an operation is made that changes the current UI characteristics or state (such as resize). This design follows the memento design pattern (Gamma 1994).

The algorithm class follows the strategy design pattern (Gamma 1994). Every algorithm is a leaf node of the algorithm inheritance hierarchy, where the root class (RectRegionAlgo) has an abstract method (placeSGVs()) that specifies the placement

algorithm. This helps to reuse methods and avoid duplication leading to concise, clean, and comprehensible code.

The algorithm class encapsulates the algorithm to paint the background so that this algorithm can be customized for each specific placement method. For example, the GridPlotXY placement method paints the background in terms of horizontal bands and vertical lines. In addition, it encapsulates the computation of the visual rectangle for a given region, placement attribute, and minimum and maximum values for a given visual link filter. Therefore, the algorithm class is independent of regions and placement methods (and attributes).

The algorithm's major function is to calculate node locations. Then, NVSS uses a JUNG layout that places nodes according to their coordinates. The method of placing nodes takes a region, vertices, and an algorithm memento (RRAlgoMemento) and places nodes according to these parameters. Depending on the aggregation mode and cell sizes, it performs node aggregation operations (aggregates nodes and links or decomposes metanodes and metalinks).

Each node in NVSS contains its attribute values. Therefore, nodes (or collection of nodes) are passed via parameters to various methods, where their attribute values are processed using the attribute values when necessary. There is a manager class (SGGraphManager) that holds the graph, that is, the graph structure in JUNG as well as all nodes in the graph. It also contains the logic to aggregate to metanodes and decompose from metanodes. To perform these operations, it also holds structures to remember which nodes are represented as metanodes and the mapping from the former to the latter. This is necessary when forming or

decomposing metalinks driven by the node aggregation operations. NVSS stores the list of nodes that each metanode represents and uses this information at times.

Node aggregation is performed on demand and incrementally. In other words, the entire graph is not regenerated when a node aggregation operation is needed. Instead, several nodes are taken out and replaced with a metanode or vice versa. During this process, associated links need to be recomputed.

Regions are represented with the region class (RectRegion). This class connects necessary entities that are related to a region. Every region has a unique index at run time for internal identification and communication. For example, the substrate layer class (SubstrateLayer), which is associated to all regions and indexes them, recognizes regions by their index. Every region holds a set of information including state information. A region contains the following:

- Its unique region index
- The region settings (RectRegionSettings)
- The substrate (SubstrateSettings)
- A layout class of JUNG (extended and named SGLayout)
- The class that holds the graph (SGGraphManager)
- A collection of nodes that fall into this region
- For each link visibility filter
    - Whether it is active
    - The visual representation of it
        - This is computed by the algorithm memento and stored in the region (RectRegion)

- Its minimum and maximum values

- Order providers (OrderProvider)
  - These are classes that provide the order of a value of an attribute in this region. These are useful to synchronize the range sliders with their visual representation on the graph (a hollow rectangular box) and many other things.
  - NVSS has two different order providers:
    - Bin order providers
      - Given the minimum and maximum value and the number of bins (BinParams), these calculate the values and their orders on an attribute value line. This concept is kept abstract to accommodate various attribute value types (that is, INTEGER, DOUBLE, STRING, and DATE, and possibly others in the future).
    - Custom bin order providers
      - They take the order values from the custom bin structure (CustomBinParams) directly (they are called boundary values in that context).

- The original bounds (of the visual representation) of the region.
  - This is used when the region is resized. The factor is found and is multiplied with the original bounds values to avoid accumulated error from the repetitive floating point multiplications, which cause residual round-off errors due to computational limits.

- The algorithm mementos

Link filters and link visibility range filters are generated by the control panel class (SGControlPanel) of NVSS automatically depending on the substrate (specifically regions and the grouping and placement attributes they use). The control panel has a connection to the substrate layer class (SubstrateLayer), which has connections to all regions in the substrate. The filters in the control panel message the associated region (RectRegion) the new parameters (minimum and maximum value of a range filter), which makes the region compute new filter rectangles using the associated algorithm and set the minimum and maximum values of the range filter.

The region also provides service methods to determine whether a node is within a given range, that is, between a minimum and a maximum value. This service is used the link predicate class (provided by JUNG, extended by NVSS, SGEdgePred) and links associated with these nodes are viewed accordingly.

All attribute values in NVSS are treated generically depending on their type defined in NVSS. NVSS has a module consisting of only type information and processes (the datamodel package). It is possible to extend the types of NVSS systematically by adding several classes to this package and making sure they are supported by other procedures correctly.

NVSS has a UI package that provides shared functionalities in terms of the user interface. Some of these are as follows:

- The transformation formula component (TransformationPanel)
    - The associated computational class is TransFormula.

- The attribute and bin selector component (AttributeAndBinSelector)
    - It is used once or twice inside the placement method selector (PlacementMethodSelector) for a region.

NVSS uses a few code fragments or packages that were freely available on the internet. These are the double-slider (revised) and the extended file chooser (EFileChooser). Using the extended file chooser, users can view a preview of a selected file and can select a previously used file from a combo box to replace a deep browse function. Also, NVSS has a few convenience features such as remembering the last file locations used, accepting file locations on the web instead of on the local disk of the computer (useful for easy distribution of a dataset by avoiding sending the dataset over email to each client/user), and silent recoveries from corrupted files. In addition, NVSS gives useful diagnostic messages when some of the file formats are violated rather than simply crashing.

NVSS has undergone many versions of substrate file formats. It is backwards compatible and can provide options for upgrade in many cases for older files. It also uses likely default values whenever possible.

The double-sliders have a continuous range of interaction. When the double-slider values are discrete and users leave a slider in a non-discrete location, it is automatically and silently moved to the nearest discrete location when needed (e.g. when a range sweep occurs by dragging the middle of a range). Still, double-sliders could be improved. For example, introducing a minimum length that is long enough to be dragged will increase the usability of the double-sliders for points and short ranges.

4.1.2 Implementation Statistics and History Overview

This section provides implementation statistics of NVSS.

NVSS consists of 9 packages that group the code into logical components that are highly related to each other. Two of these packages contain contributed code (doubleslider and util.contrib, which contains EFileChooser and SpringUtilities for spring layout that is partially used in the control panel to arrange items orderly). Some metrics provided by the Metrics plug-in in Eclipse are provided in Appendix B.

NVSS has undergone many revisions during its history. The initial major revision occurred to enable representing more than two regions because the initial version was a prototypical application allowing only one or two regions in the substrate. A major revision occurred when NVSS introduced its own data types and the existing data types needed to be transformed to those data types. The NVSS data types allow general processing and facilitate holding various kinds of datasets. This improvement enabled visualizing different datasets. This revision was tested with the food-web dataset. (The initial dataset was the court case dataset by CiteIt.) Another revision occurred when the substrates were generated by the Substrate Designer and saved in files. During these revisions some source files changed name and locations. The class SubstrateSettings was introduced to more clearly separate the substrate from the data. Finally, another revision occurred to support the node aggregation feature. Since NVSS had a modular design, this revision had fewer challenges than the others. Still, functionalities were designed with the assumption of a static (unchanging) graph. Therefore, during this revision, all parts of the implementation that used the graph needed to be revisited and those that were affected by this

assumption needed to be generalized to assume that the graph could change. However, this made NVSS more flexible and these changes were performed in a controlled and careful way.

Since CVS is used for version control, at instances where files were renamed or changed location, revision history for those files was lost and the revision number restarted (from 1). Nevertheless, the current statistics on the version numbers will still give an idea about classes that are revised frequently. Table 2 provides the most frequently revised classes:

Table 2 Actively revised classes of NVSS.

Class	Number of Revisions	Lines of code	Comments
RectRegion	204	271	This class is very central as RectRegion partially acts as a communication hub for entities and processes associated with or related to a region. When improvements and generalizations were made, it has undergone major simplifications.
SGApplication	111	372	The class is responsible for initializations and launching the visualization instance. Before SGMain and SubstrateDesigner existed, it was the main class and its name has undergone changes causing huge revision histories to be lost (its very first name was TwoGroupsGraph and remained from the times when NVSS could support only 2 regions.)
SGGraphManager	40	227	This class is a simplified and generalized version of a former class called SGVertexManager, which was a communication hub and processing agent for the nodes of the graph visualized. The class had undergone many revisions (probably above 100) and consisted of around 1000 lines of code. Its current role has much

			simplified and generalized. It is the major class holding the graph, i.e., all nodes and links and provides services (mostly to the algorithm classes) to perform the calculations necessary for node aggregation to occur.
SubstrateLayer	70	179	This class has connections to all regions and provides services to whomever that needs access to a region. It also plays a role when the regions are created in memory and resize calculations of the visualization component.
GridPlotXYRectRegionAlgo	78	153	This is the most popular placement method. There have been many revisions, alterations, and corrections on its algorithm, and additions / changes due to the addition of node aggregation feature.
RectRegionAlgo	81	136	Being the superclass of all placement method classes, this class has changed over time as common code moved into it and out of it and as parameters of the common methods and its constructor have been revised.
SGMain	103	601	The main component of NVSS communicates both with the Substrate Designer (SubstrateDesigner) and the visualization instance (SGApplication). Also, there have been changes in its UI and improvements in silent error handling and default value usage for improved user experience.
SubstrateDesigner (SD)	175	1,097	The largest (non-contributed) class of NVSS (with 1097 lines of code) holds many UI components and communicates to subcomponents. SD has undergone many revisions as the interface of the designer improved and changed as well as when subcomponents were extracted as classes (most of them widgets visually operable via the Visual Editor (VE) of Eclipse) to lighten the burden and simplify this class. Most

			of the code in this class is automatically generated by VE. The predecessor of SD was SubstrateEditor, which did not use Piccolo and was not designed visually using VE. It was deleted to be replaced completely by SD.

NVSS history officially starts on 10/15/2005 although it has a previous short unofficial history, from the time it originated within the Cite-It project to the time that it was extracted as a separate Eclipse project. The official start date is the date when it became a separate Eclipse project. Since then, the project has been tagged (snapshot of the code base taken and its state preserved by CVS, the version control system) many times (1-0-0 to 1-0-10, 1-1-0 to 1-4-4, 2-0-1 to 2-4-1), totaling to 87 project level versions (tags). Many of these have been uploaded onto the web and shared with collaborators.

4.1.3 Placement Method Class Inheritance Structure

The following chart (Figure 29) shows the inheritance structure of placement methods in NVSS.

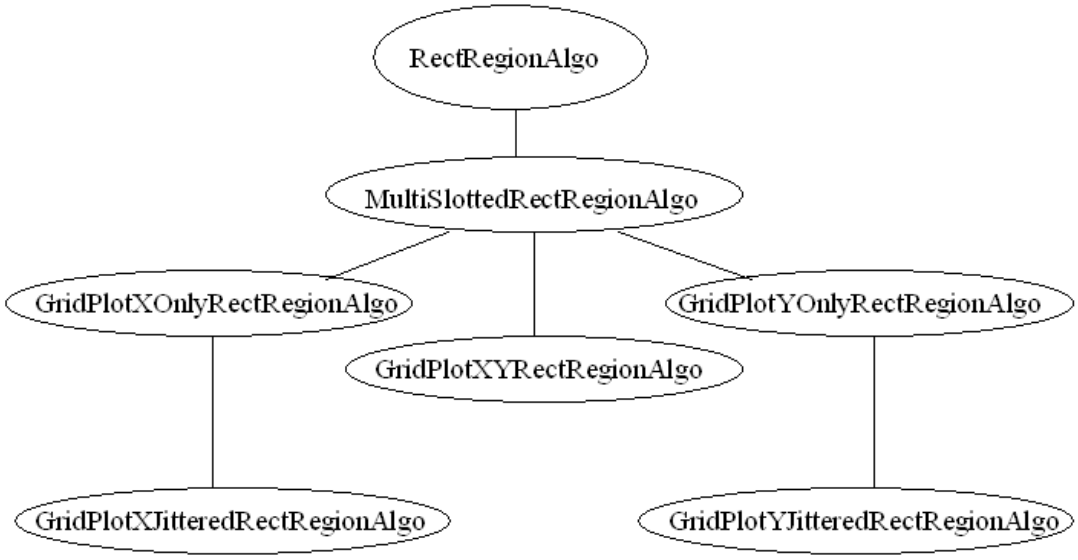


Figure 29 Inheritance hierarchy of the placement method classes in NVSS.

The placement method follows the strategy design pattern and is a stateless class with respect to the region and nodes in that region applied. The classes are structured according to an inheritance hierarchy in order to maximize reuse of code / functionality. RectRegionAlgo is the abstract superclass of all placement methods.

The five placement methods in NVSS are the GridPlot- classes. Since the Jittered versions are very similar except the implementation of the node placement, they have an inheritence relationship with their non-jittered versions so that they reuse everything else. The MultiSlottedRectRegionAlgo abstract class incorportates the methods that are more general than a specific placement method. For example, it has the methods to paint the horizontal and vertical grids of the region, and methods to paint the X- and Y- axes. In addition, it overrides the abstract method to produce the rectangular filter structure (getFilterRect(RectRegion, int plAttrIdx, int minOrder, int maxOrder)) for the X- and Y- axes (getFilterRectX(...) and getFilterRectY(..) which are used by subclasses). It also implements the method to calculate the local

order structure (getLocalOrdersFor(…)) that defines the mapping between discrete filtering orders (which are used in the visualization and by the getFilterRect() method) and attribute values, which are used as the range of values on the range sliders.

The classes for the Jittered versions of the placement methods are very short and simple as they override only a small method that is used by the placement algorithm (placeSGVs()), which is "getDisplacement(int order)", and is a 1-line method that returns the amount of shift for a node so that jitter occurs. It has two other simple methods that are fundamental and define the name and an internal id of the placement method.

The RectRegionAlgo, which is at the top of the hierarchy has elements common to all algorithms, such as encapsulating the placement attributes, creating a memento given a region (RectRegion), encapsulating the attribute value converters, and a default background paint function, which could be overridden as necessary by the placement method classes.

This structure is amenable to be enhanced in order to support additional placement methods in the future. For instance, to support a force-directed layout, one would extend (inherit) the RectRegionAlgo class, override the abstract method placeSGVs(), which will define where the nodes will be placed. If this class needs elements to appear on the background (oval shapes to indicate clusters, for instance), it could override the default background paint method. A modified force-directed layout could use an attribute to further group nodes within a region. In that case, the class shall override the filter producing method (getFilterRect(…)) and the local order

map producer method (getLocalOrdersFor(…)) with non-trivial definitions. This will automatically add support to the user interface to filter using the placement attribute (using the range sliders which is in synch with a filter rectangle that appears on the visualization).

To introduce another GridPlot-like placement method, one may introduce the placement method most probably under MultiSlottedRectRegionAlgo and possibly reuse some of the methods already defined there.

4.1.4 Algorithm Memento Class Inheritance Structure

The following chart (Figure 30) shows the inheritance structure of algorithm mementos in NVSS. An algorithm memento class encapsulates details specific to a region produced by a placement algorithm. This way an instance of the algorithm class only encapsulates the node placement strategy and remains independent from any (specific) region.

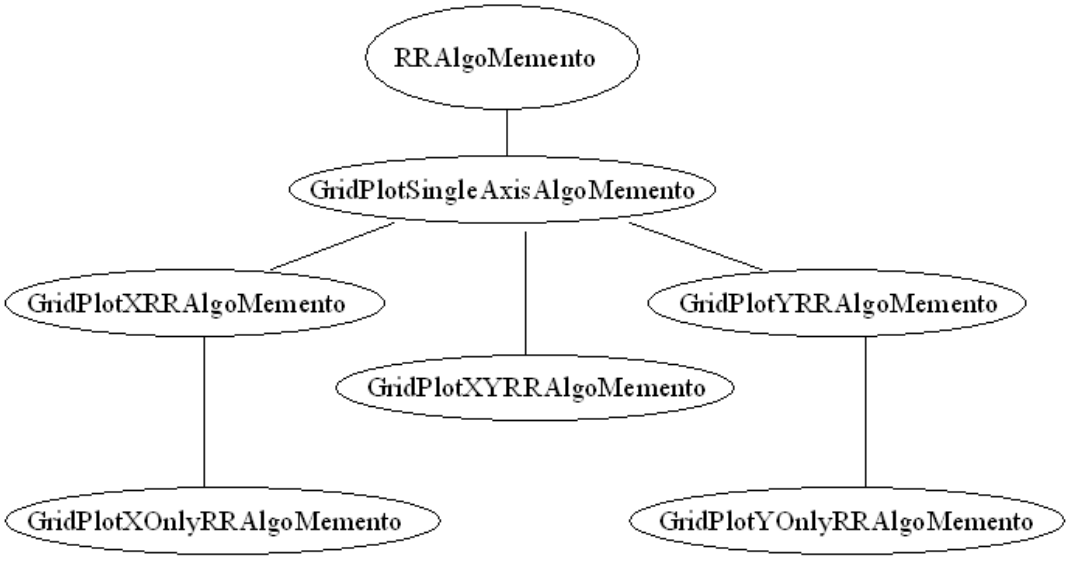


Figure 30 Inheritance hierarchy of algorithm memento classes in NVSS.

The placement method class (RectRegionAlgo) generates a memento given a region (RectRegion). The memento calculates and remembers node information related to the specific region. The major processing involves group and element order. Nodes that are within the same cell (GridPlotXY) or within the same slot (all the other GridPlot placement methods) belong to the same group. Each node within a group is ordered by the element order. These are calculated according to the specific algorithm. The mementos also hold the axis information (in terms of a reference to an axis class created; different classes exist for the vertical (y-) and horizontal (x-) axes). RRAlgoMemento is the superclass of all and implements a default version of the required methods. The GridPlotSingleAxisAlgoMemento abstract class has many reusable methods and a constructor. Consequently, the 3 leaf classes need to add or modify very little. Each memento is created by its corresponding placement method class. For instance, GridPlotXYRRAlgoMemento is created by GridPlotXYRectRegionAlgo. There is no distinction among the mementos with respect to jittering. Therefore, both the jittered and non-jittered versions of a placement method class produce and use the same memento. For example, both GridPlotXOnlyRectRegionAlgo and GridPlotXJitteredRectRegionAlgo use GridPlotXOnlyRRAlgoMemento.

4.1.5 The Node Aggregation Algorithm

The node aggregation algorithm requires processing the internal graph structure. The other option of regenerating the graph and substituting with the existing one was considered, too, which didn't prove to be a better solution.

Therefore, the existing graph structure was chosen to be modified as node aggregation operations take place. This solution also is a very efficient one as nodes that do not undergo aggregation operations do not cost additional execution time.

The node aggregation mechanism is integrated with the placement method classes, specifically the placeSGVs() method. This is because aggregated nodes need to be placed. A new class called SGMetaNode was created to represent a metanode. Since it would not be modular for node aggregation to be defined by a placement method, the algorithms for aggregation operations needed to move to a common location. This location resulted to be the SGGraphManager class. This class holds the graph structure, that is, the complete collection of nodes and links.

The SGGraphManager class supports two public methods to be used by placement method classes. These are aggregateToMetaNode(nodes, nodeData) and decomposeMetaNode(metanode). The former one takes a collection of nodes and the attribute values for the metanode to be created and returns the metanode. The latter one takes a metanode and returns a collection of nodes (the nodes that this metanode represents). NVSS doesn't require all nodes to be aggregated or non-aggregated all at once (This is due to the existence of the "Mixed" mode in NVSS, in which some groups of nodes are transformed into a metanode while others aren't). The initial link structure of the completely non-aggregated graph is preserved. Without this addition, a mapping from nodes to metanodes seems to be sufficient; however, this was found not to be true as some cases cannot be covered. The current solution is simpler and sound.

The placement method checks to see the aggregation mode (set by the user on the control panel) and accordingly decides to aggregate node groups or decompose metanodes one by one. At this stage, the operations provided by the SGGraphManager class are used. After this step, the placement method places nodes and metanodes. The placement algorithms for the two currently differ (metanodes are centered while nodes start from the upper left corner); however, these could be combined in the future with an elegant algorithm that unifies both cases.

The placement method tracks whether the graph has changed due to aggregation operations. If not, it does not do any further updates. Otherwise, the placement method updates the structures of the associated region (only the structures that are influenced by the node aggregation). For example, the local nodes of a region are re-queried (which could be thought of as a cache copy of nodes that fall into that region. The cache of nodes local to a region saves time as these nodes are used frequently. Otherwise, every time these nodes are needed, they have to be searched among all nodes in the graph). In addition, the order providers that are within the region are updated. The order providers hold a mapping from an NVSS node to its order (this corresponds to the order on the range slider) depending on the placement attribute the order provider represents. This class is an abstract class, whose subclasses are BinOrderProvider and CustomBinOrderProvider. Currently, the only orders in NVSS are through the binning process; however, this can be extended to other types of logic if needed in the future.

The node aggregation is performed incrementally on every group that needs to be aggregated. However, this process does not lead to many paint call. In other words,

it is done efficiently, i.e., the paint method is called only once after the node aggregation operations are completed.

4.1.6 Miscellaneous

There are separate classes for the horizontal (x-) and vertical (y-) axes. They are responsible for mapping tick marks to the region as well the placement of the labels. They have paint methods that are eventually used by the background paint method of the algorithms, which in turn are called from the region's (RectRegion) paint method. The paint methods of all regions are called by the SubstrateLayer's paint method, which implements VisualizationViewer.Paintable. SubstrateLayer is registered to the JUNG's VisualizationViewer's instance as the pre-render paintable object (VisualizationViewer.setPreRenderPaintable(SubstrateLayout)). This way, regions are painted before the graph is painted by JUNG. JUNG also has a channel to accept a post-render object, which NVSS doesn't use.

The classes for axes make the calculations (for the tick marks and label locations) based on the available space (i.e. the size of the associated region). Depending on the available space and the space that labels take, if the space is not enough to paint all labels, every other label or every third label (and so on) is painted. This way, when users resize the application, the labels never overlap but accommodate to the given situations. The axis classes also provide services to other components in NVSS. For example, in computing the coordinates of the filter box on the visualization, their X- and Y- calculation methods are used by the placement method classes. The placement method classes compute the vertical and horizontal grids, similarly, by using these services.

4.2 System Performance and Scalability

NVSS provides smooth interaction up to 1000-2000 nodes on a 3GHz Dell 8400 with 3GB RAM. Larger networks that reduce to this size in the metanodes mode could be smoothly explored. The transition time from and to metanodes mode is 2 seconds for the dataset in the TobIG case study (section 5.2.3). The TobIG dataset contains 4,296 nodes and 16,385 links. The metanodes mode has 72 metanodes and 804 metalinks. NVSS was also given a dataset containing 29,555 nodes and 352,807 links. Creating the graph takes approximately 40 seconds and launching the visualization 30 seconds. The transition time from nodes to metanodes mode is 70 seconds and the transition from metanodes to nodes mode is 155 seconds. The metanodes mode has 158 metanodes and 10,234 links. In general, the transition times depend on the network size (number of nodes and links), the substrate applied (how nodes are distributed among regions and cells), and perhaps slightly on the structure of the network.

Large transition times hinder exploration when tasks require frequent switches between the nodes and the metanodes modes. However, large transition times (such as 155 seconds) might be still acceptable for visualizations that have a small enough network to produce smooth interaction in the aggregated view. Two situations that this might be acceptable are as follows: The first situation is if the aggregated view will exclusively be used for the exploration. The second situation is if the aggregated view will be used to find parts of the network to focus on. In that case, those parts can be externally filtered and the subset of the dataset consisting of those part(s) can be explored.

4.3 Summary of the NVSS Implementation

In brief, NVSS is written in Java using JUNG (O'Madadhain 2008) (a Java library to represent and visualize networks) to hold the graph data structure and for the visualization and Piccolo.Java (Bederson 2004) for the visual operations in the Substrate Designer module. NVSS Main is a connection platform between the Substrate Designer module (the output of which is a substrate) with the NVSS Visualization Module (which expects a substrate and a dataset).

NVSS was developed with good programming practices of refactoring (Fowler 1999) and design patterns (Gamma 1994). Hence, the code consists of reusable components that define unique operations, is modular and therefore easily extendible with appropriate refactoring practices. The code contains meaningful variable, function, and class names and therefore has minimal documentation (as it is not needed, which is the application of another principle from (Fowler 1999)). Functionalities are defined so that they are efficient without compromising good design and good comprehensibility of code.

Chapter 5: Evaluation

Evaluation of the semantic substrate idea is performed by case studies using NVSS to explore various datasets with interested users.

The case studies in this dissertation are conducted according to guidelines described in (Shneiderman 2006). These are:

- **Multi-dimensional**: Various ways are used to assess user performance, interface effectiveness, and utility. Some of these ways are interviews, observations, and surveys.
- **In-depth**: Over time, evaluators are engaged deeply in the tasks of the domain experts to the point that they become a partner or an assistant.
- **Long-term:** The case studies take 4-6 weeks on the average and possibly more, usually at the pace of a 45-minute or a 1-hour session per week.
- **Case studies:** Notes were taken to report in detail about a small number of individuals, who usually are domain experts, while they are trying to solve problems they have or define in their natural setting.

The case study participants are researchers collaborating with the author of this dissertation. Many of them provided funding for the exploration process and were interested in the outcomes. Therefore, they were interested in successful outcomes. Other dissertations that use a similar evaluation method are by Jinwook Seo (Seo 2005) and by Adam Perer (Perer 2008).

The case study participants were researchers, such as Prof. Wayne McIntosh (GVPT) who has provided NSF support to partially sponsor the work in this dissertation research. They also include Prof. Noshir Contractor, whose research includes social networks, at Northwestern University and Dr. Cynthia Parr, a research biologist formerly at University of Maryland.

The role of the author of this dissertation has been to build software to help these researchers analyze networks for social, citation, or predator-prey networks. The software NVSS has been developed for a wide variety of applications, but working with these collaborators over weeks and months provided insight to the strengths and weaknesses of the software. The software was being tested and not the collaborators. These collaborators have been potential co-authors. Their role is to help validate the efficacy of the software (NVSS). The working strategy was to meet with them, and then the author of this dissertation wrote down their comments (positive and negative) about the software to guide revisions. These comments were available to the collaborators for comment and correction. No private personal information was collected. As a result, the case studies in this dissertation did not require the formal approval of the IRB (Internal Review Board) office at the University of Maryland, College Park. The IRB office declared this decision with a formal letter (see Appendix C).

Although all the case studies were intended to be long-term, some of them became short-term due to various reasons. In the case studies, various versions of NVSS were used. A summary of the significant differences among major versions is as follows:

- *NVSS 1.0:* This version is the initial version of NVSS. It was designed as a prototype to show the utility of semantic substrates. The input was specialized to accept only court cases. Also, the input was directly read from an online database. The details of a node (the node attribute values) were available on the console output of the application rather than on the control panel of NVSS. To define the substrate and input it to the software required programming effort, which was provided by the author of this dissertation.
- *NVSS 2.0*: This version is the next major version of NVSS. It is generalized to represent any network data. The data is input in the form of nodes and links files (see section 3.2 and Appendix A for details on the format of these files). This type of input does not restrict the data in terms of node attributes, i.e., to a specific domain. The details of a node are provided in the "Details" tab in the control panel. This version also separated the substrate design part by providing the Substrate Designer module.
- *NVSS 3*.0: This version is the final major version of NVSS. It added the node aggregation extension. The nodes and the metanodes modes were introduced.

After the case study guidelines in the next secion (section 5.1), the long-term and short-term case studies are grouped into separate sections (section 5.2 and section 5.3, respectively). Section 5.4 provides a summary of all case studies and section 5.5 reflections on the case studies.

5.1 Case Study Guidelines

For each case study, a case study document is provided. Case studies typically were performed in the form of 45-minute to 1-hour sessions. A session was held every time enough progress was made and a mutually convenient time could be arranged with the case study participant. During these sessions, the author of this dissertation took notes and used them to write the case study documents. These documents were shared with the participant for possible corrections and additions. The documents conform to a common format. This format is provided in terms of case study document guidelines in section 5.1.1. The guidelines for questions to ask during a case study are provided in section 5.1.2.

5.1.1 Case Study Document Guidelines

The following are sections in a case study document and explanations of what they contain.

Participants

This section gives information about one or more participants of the case study.
For each participant, include the following information:

- Their name and their position
    - Example: Wayne McIntosh holds a faculty position in the Government and Politics Department (GVPT) at the University of Maryland, College Park (UMD).
- A statement that gives a sense of their interest in this data (this could be described in the description of the project they are involved in instead).

- (If any) specific things that they are interested in the data or they are looking for
- (Whenever possible) Their motivation to look at this data
    - Will they publish papers and books based on their exploration? It could be a starting point of shaping their motivation, as well.

Dataset Description

Describe the dataset.

In the first paragraph, explain the origin of the data. Where is this data coming from? Which project uses this data? What is the nature of that project?

Add additional paragraphs as necessary to continue to explain the origin of the data.

Describe any transformations or processing done to the data (maybe from a base dataset). Also, if significant and important to know, describe any pre-processing done to the data to get it into NVSS.

Describe what nodes and links represent. Describe regions and placement methods.

Give the total number of nodes within each region and the total number of links.

Case Study Notes

This section contains notes and explanation from the case study (session, meeting, etc.)

Include figures (recommended: one per page with explanation following the figure).

For every figure, describe anything that is not obvious (sufficient to describe once in the earliest Figure), such as "13 stands for the Federal Circuit."

In the figure caption, briefly describe the nature of the data, such as, what links are visible and under what filtering parameters, such as "Incoming citations to Giants".

Follow by notes. Notes can be quotes or a summary of what happened and/or what they said. Whenever possible, qualify notes by time and participant.

Outcome

Describe what happens after this case study is over.

Are there any papers or books published due to this exploration? Or are there any pending publications?

Will they continue to use NVSS (where, why, and how)?

Briefly describe general changes such as did NVSS change their way of thinking? Did it make them realize something in general that they did not think of before?

Did the exploration help confirm (high-level) beliefs and/or facts about the dataset?

5.1.2 Guidelines to Follow during a Case Study Session

These are the questions to consider that form the guidelines during a case study session:

1. Who are the participants?
2. What is the dataset that the participants are looking at? Ask them and write it down. Include all file names (nodes, links, and substrate).
3. (optional) Take notes on their mental state (excited, eager, confused, delighted, etc.).
4. What is the task?
5. How will they accomplish this task? (Either ask or observe.)

   What are their questions to you?
6. How are they accomplishing their task? Try to capture the operations and/or the outcome (as the case study observer, capture figure if possible or take notes to reproduce later).
7. What do they want to do that they couldn't do now?

   (optional) What features would need to be present in the software to do that? (Ask the case study participant. If they don't know, suggest to them if you can to see what their reaction is. Otherwise, explain what you think it would be useful.)
8. What are their findings? (If they have any comments on their findings, write them down, too.)
9. What suggestions do they have? (feature requests, suggestions, and brainstorming)

5.2 Long-Term Case Studies

The following subsections provide three long-term case studies. These case studies have lasted from 2 months to more than 1.5 years.

5.2.1 Cite-It Case Study

The following subsections divide the Cite-It case study in terms of the data that the participants were exploring.

The participants were Wayne McIntosh, Ken Cousins, and Stephen Simon, members of the CITE-IT Team, an NSF funded project that aims to analyze and understand the evolution of regulatory takings cases over the years.

Regulatory takings: The U.S. Constitution requires the government to provide "just compensation" when it physically appropriates private property for a public use (building a highway, for example). A "regulatory taking" requiring the payment of "just compensation" may also occur when the value of private property is destroyed by government action that falls short of actual appropriation, such as when a zoning ordinance has the indirect effect of depriving the owner of any viable use of the property.

The dataset was collected by a team of researchers, called Cite-It, from the Department of Government and Politics at the University of Maryland and has changed and grown over time.

The data originated from the efforts of the CITE-IT project, which collects potential regulatory takings cases from the online Westlaw[4] database and determines whether they are regulatory takings cases. The dataset consists of federal cases and their citations as well as metadata about the cases. First, the post78 dataset is collected, which has federal cases (Supreme Court, Circuit Courts, District Courts, and other courts) from 06/26/1978 to 10/15/2005. Then, the pre78 dataset is downloaded from 12/31/1899 to 06/25/1978. Later, they are combined to form the post1900 dataset.

The following sections use subsets of the Cite-It datasets. Cases from courts other than the Supreme, Circuit and District Courts (bankruptcy, tax, etc.) are not included in the data that is explored.

5.2.1.1 Initial Post78

This section reports exploration on the initial version of the post78 dataset. The next section reports exploration on the final version of the post78 dataset. In these two sections, the data is similar and the goals are the same. In this section, NVSS 1.0 is used and the dataset contains 2780 federal judicial cases from the period 1978 to 2005 concerning the legal issue known as "regulatory takings."

Node placement is tied to the temporal attribute (year for the case) in which the oldest cases (nodes) are on the left and the newest on the right, organized into discrete vertical slots, as in historiographs (Garfield 2004), where they are usually horizontal. Within a year, a vertical jittering function spreads out the cases to reduce link crossing and tunneling under nodes. The jittering function, which moves nodes

[4] http://wic.westlaw.com

up every 2nd and 4th slot and down every 1st and 3rd slot within a 4-slot period, was arrived at experimentally and was found to decrease link overlaps.

The cases, ranging from 1978 to 2005, were carefully selected by political science researchers to study patterns of precedents. Their numerous questions involve issues such as changing patterns of reference over time. For example, they seek to understand whether Supreme Court cases rely more heavily on lower courts (Circuits and Districts) now than in the past. Another task is to study evolving patterns of reference to a key Supreme Court 1978 case by later court cases at each level. The problem is complicated by distinctions among the 13 Circuit Courts, and 90+ District Courts, but to start only Supreme and Circuit Court cases were shown. To make it easier to comprehend, the dataset is kept small: It consists of 36 Supreme Court and 13 Circuit Court cases that were cited at least 45 times by other cases in this 2780 case corpus, thereby indicating their importance. Within these 49 cases there are 368 citations from 1978-2002. This is a modest sized network, but is already difficult to draw in a way that preserves visibility. Figure 31 shows the hopelessly cluttered display as a result of using the JUNG's layout that uses Fruchterman–Reingold's (FR) layout algorithm. Larger node sizes indicate greater number of citations to previous cases in the text of the case, but other attributes can be used. This layout is additionally problematic because the interesting cases with many in and out links are tightly woven together in the center and temporal patterns are difficult to assess.

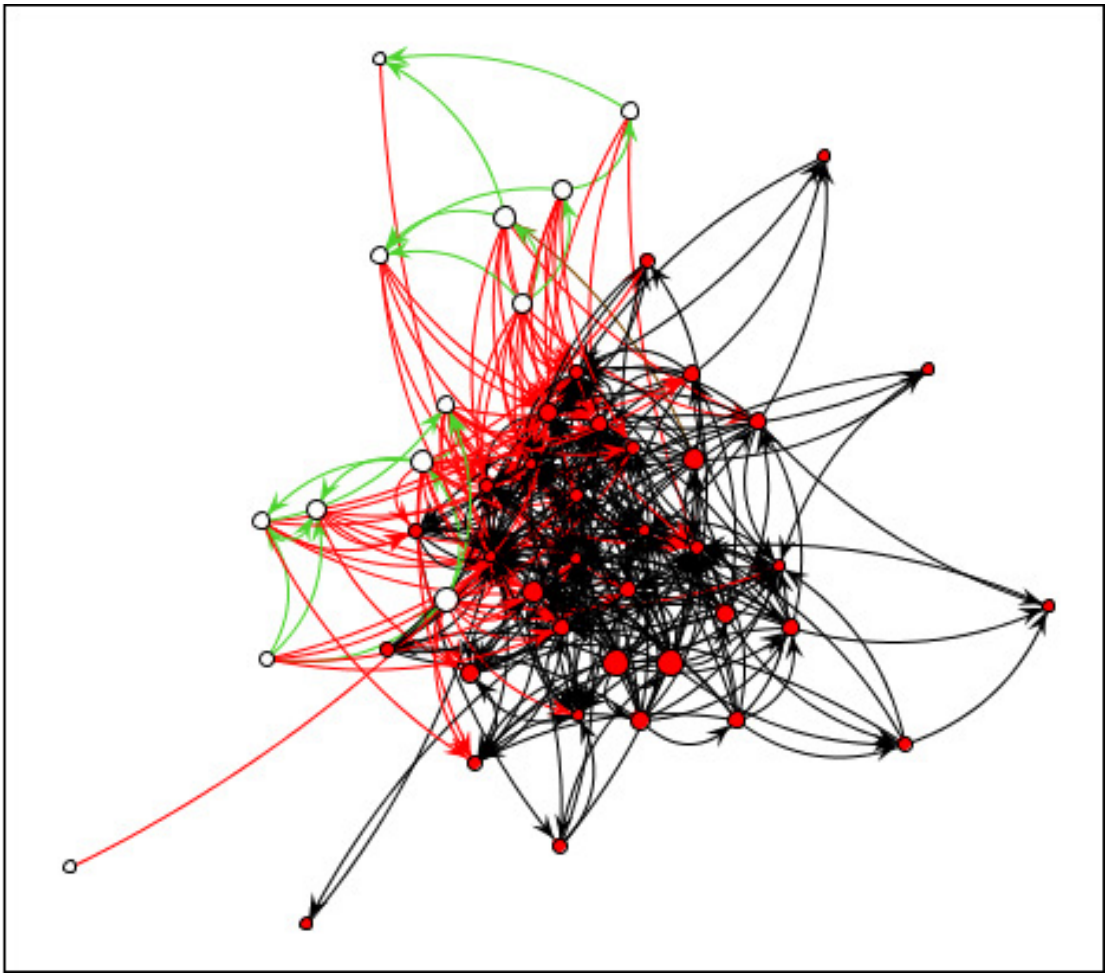


Figure 31 Using JUNG's FR algorithm to place the 49 cases with all 368 citations makes it impossible to follow citations from source to destination or to see temporal patterns.

Using NVSS 1.0, we created regions for the Supreme and Circuit Court cases in temporal order with oldest on the left (Figure 32). The controls for link visibility allow users to see the four categories of citations: Supreme to Supreme (260 citations), Supreme to Circuit (1), Circuit to Circuit (18), and Circuit to Supreme (89). In this visualization, there is a highly asymmetric citing relationship, since Circuit Court cases are more likely to cite Supreme Court cases (89 times) than the other direction (only 1 time). To expose the number of citations across regions, NVSS

includes numbers in the control panel, and includes a color key for the different kinds of citations.

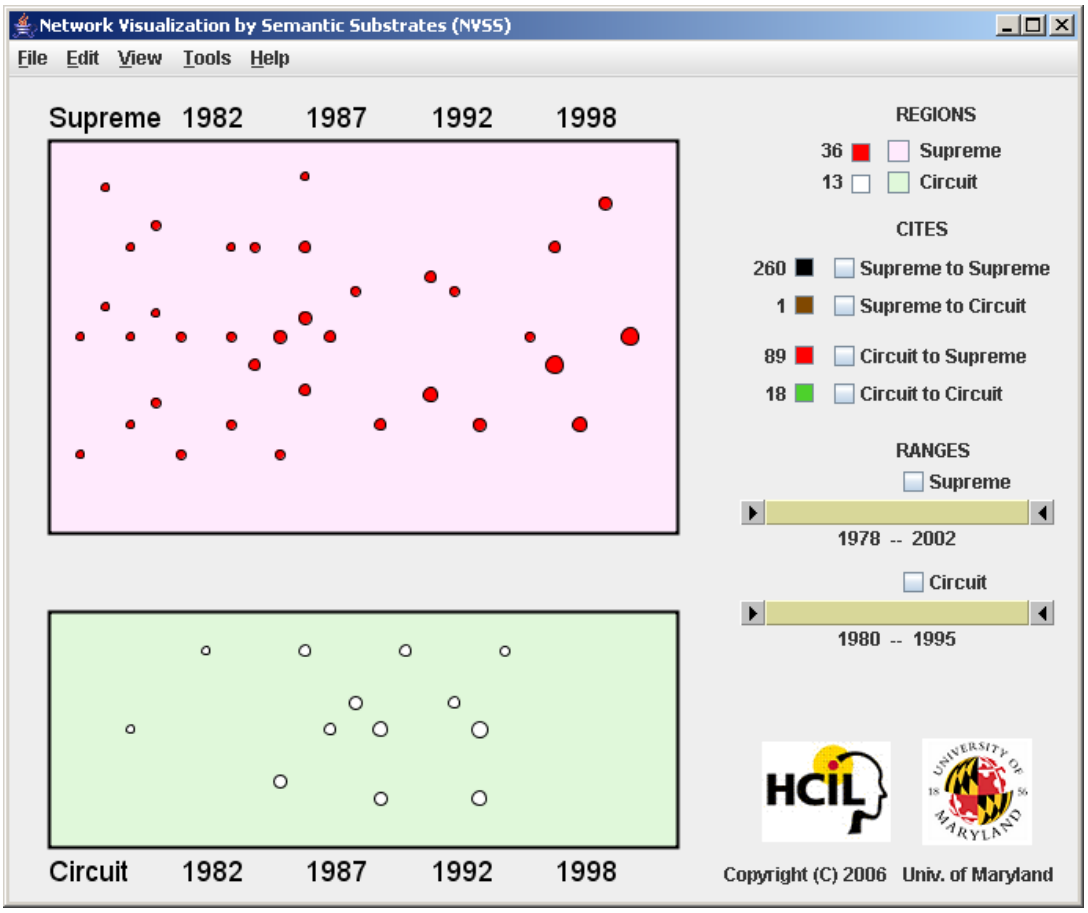


Figure 32 Step 1 in simplification places nodes in regions without links. Supreme Court region has 36 cases from 1978-2002. Circuit Court region has 13 cases from 1980-1995.

In this visualization, the user controlled link visibility is best utilized to clearly display the single Supreme to Circuit citation and the 18 Circuit to Circuit citations (Figure 33).

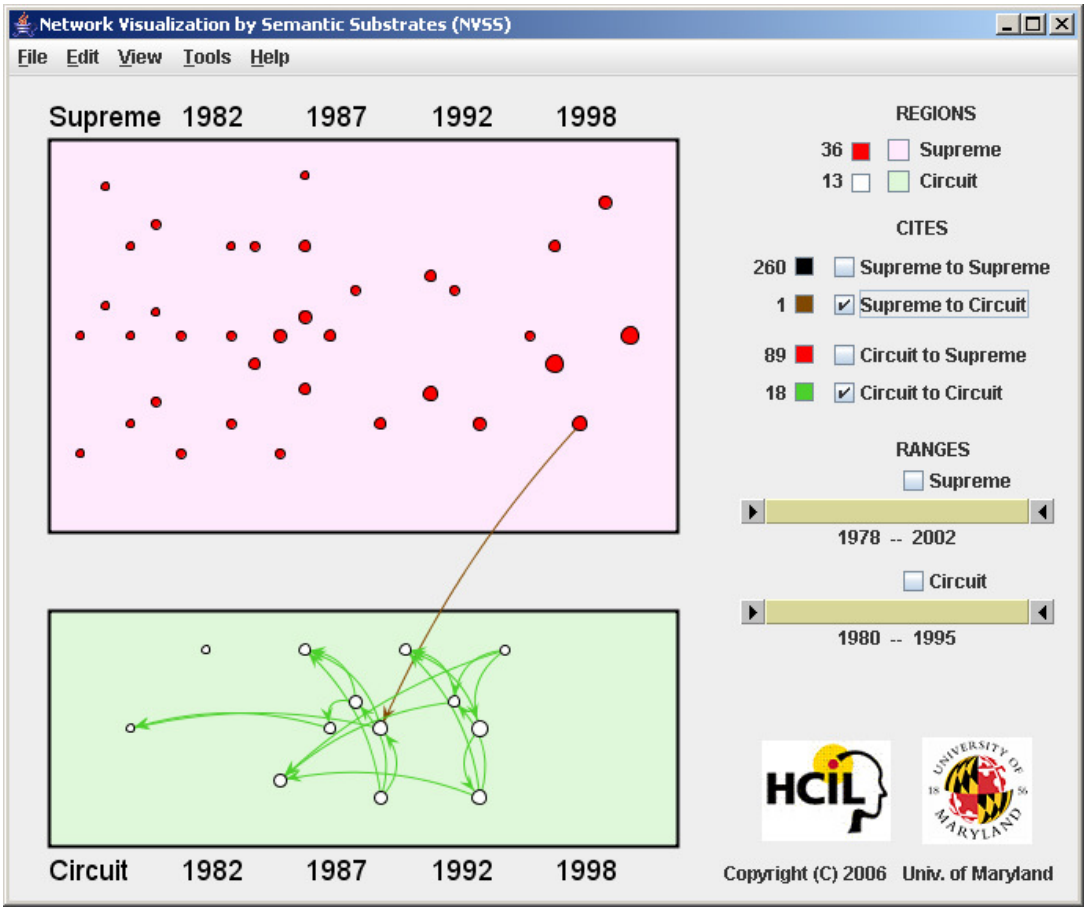


Figure 33 Step 2 of applying interactive control with check boxes simplifies the display and shows just one brown Supreme to Circuit and 18 green Circuit to Circuit citations.

To cope with the clutter of the 260 Supreme to Supreme links, NVSS provides users with double-box dynamic query sliders to filter the year range for cases whose citations are displayed. Users can tightly limit the year range and then sweep through the full range of years for an animated overview (Figure 34).

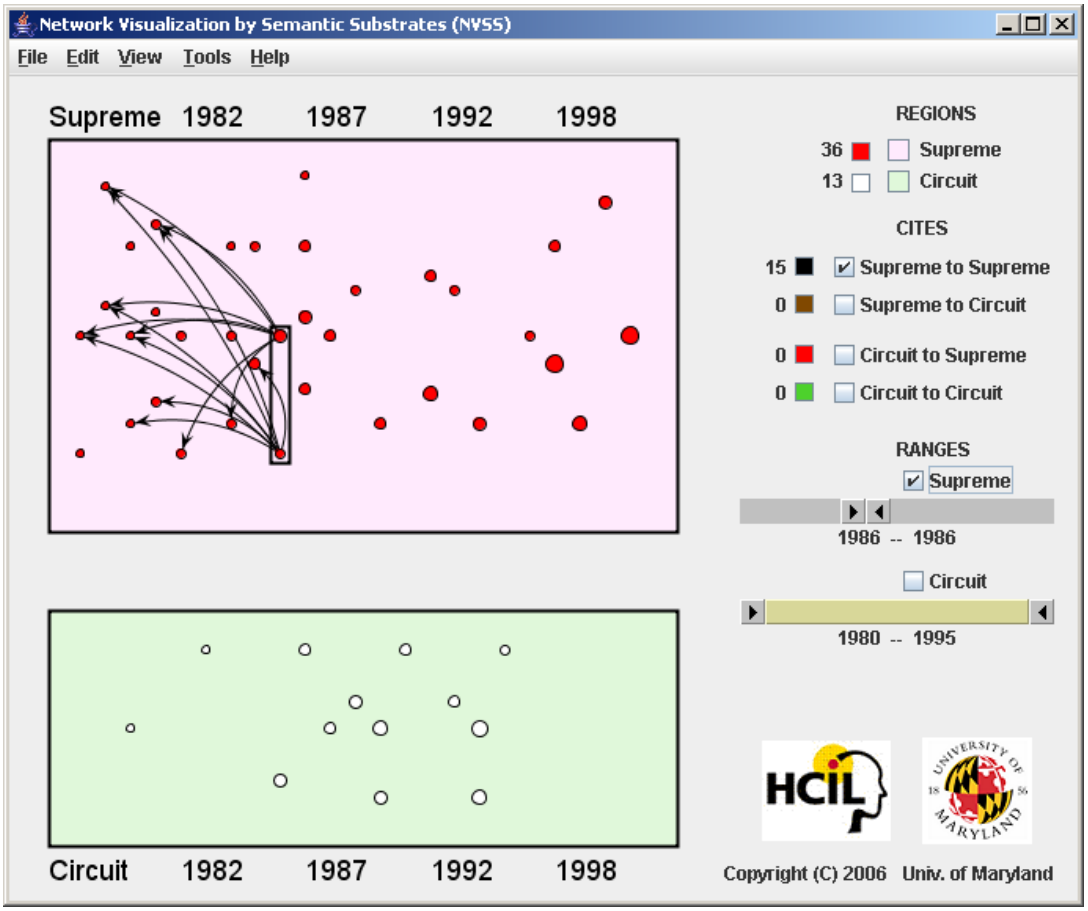


Figure 34 Step 3 shows that even the clutter of Supreme Court cases is controlled by limiting to the 2 in 1986 with just 15 citations. Five cases are cited twice and 5 cases are cited once.

Even after filtering, links may be hard to distinguish, that is, follow from source to destination.

The range selection works well across regions. By selecting the 1990 to 1991 Circuit Court cases using the Circuit Court slider, users can see the two citations to Circuit Court cases and the 18 to Supreme Court cases (Figure 35).

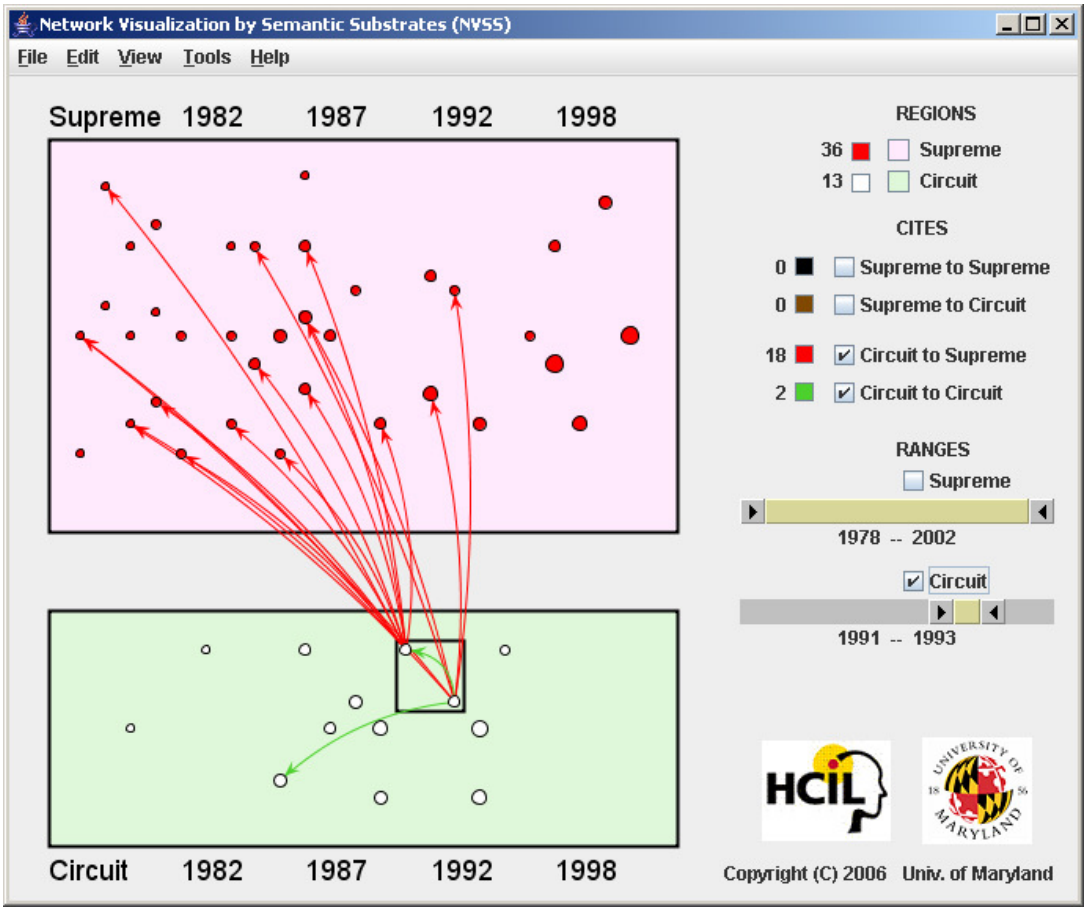


Figure 35 Limiting the selected Circuit Court cases to the 2 in 1991-1993 generates a comprehensible display of the 18 red Supreme Court and the 2 green Circuit Court citations.

While citations in Figure 35 are still comprehensible, sometimes the current link drawing strategy will need to be improved (Figure 36). The close alignment of just the two Circuit Court cases makes the red citation links overlap, undermining visibility. Animated node movement or improved link routing are possible improvements.

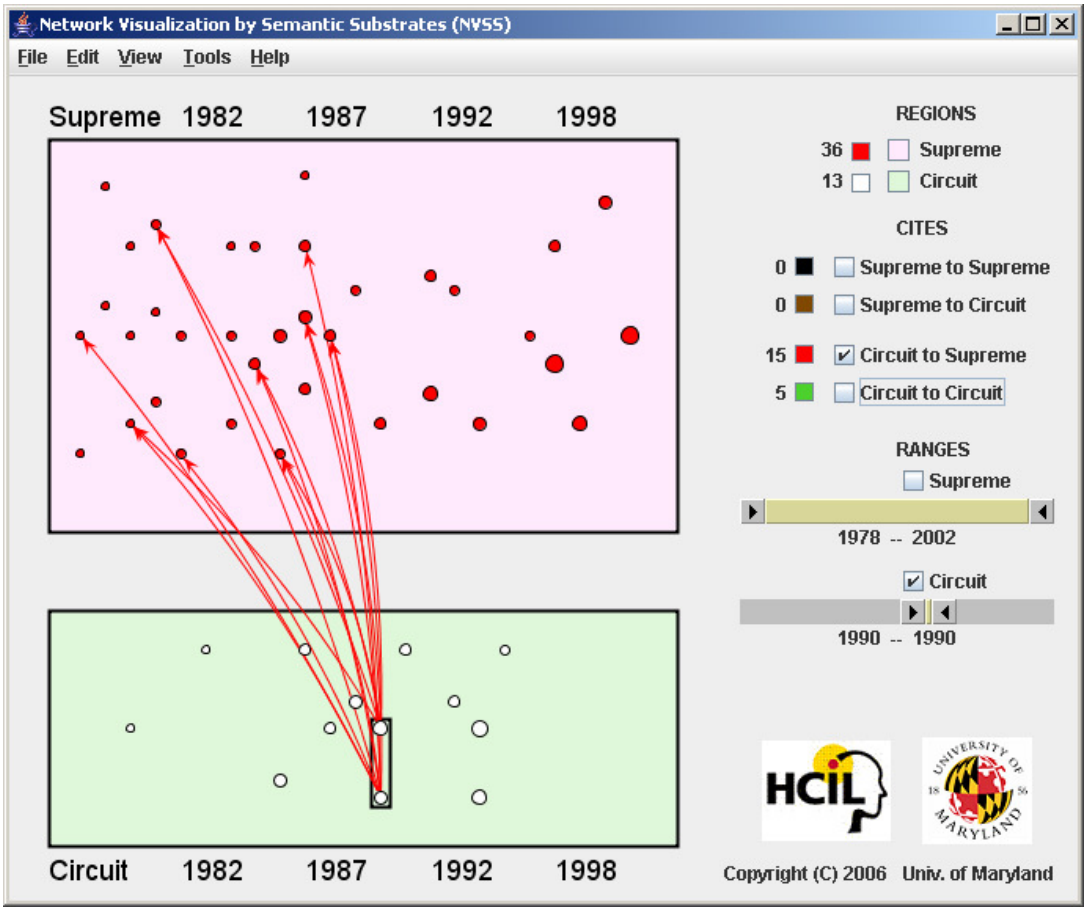


Figure 36 Limiting the selected Circuit Court cases to the two in 1990 generates overlapped links to Supreme Court cases, suggesting the need for improved link routing strategies.

Having more than two regions reveals more information (Figure 37). In this court case example, a natural choice for the third region is to include the District Court cases. In Figure 37, 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).

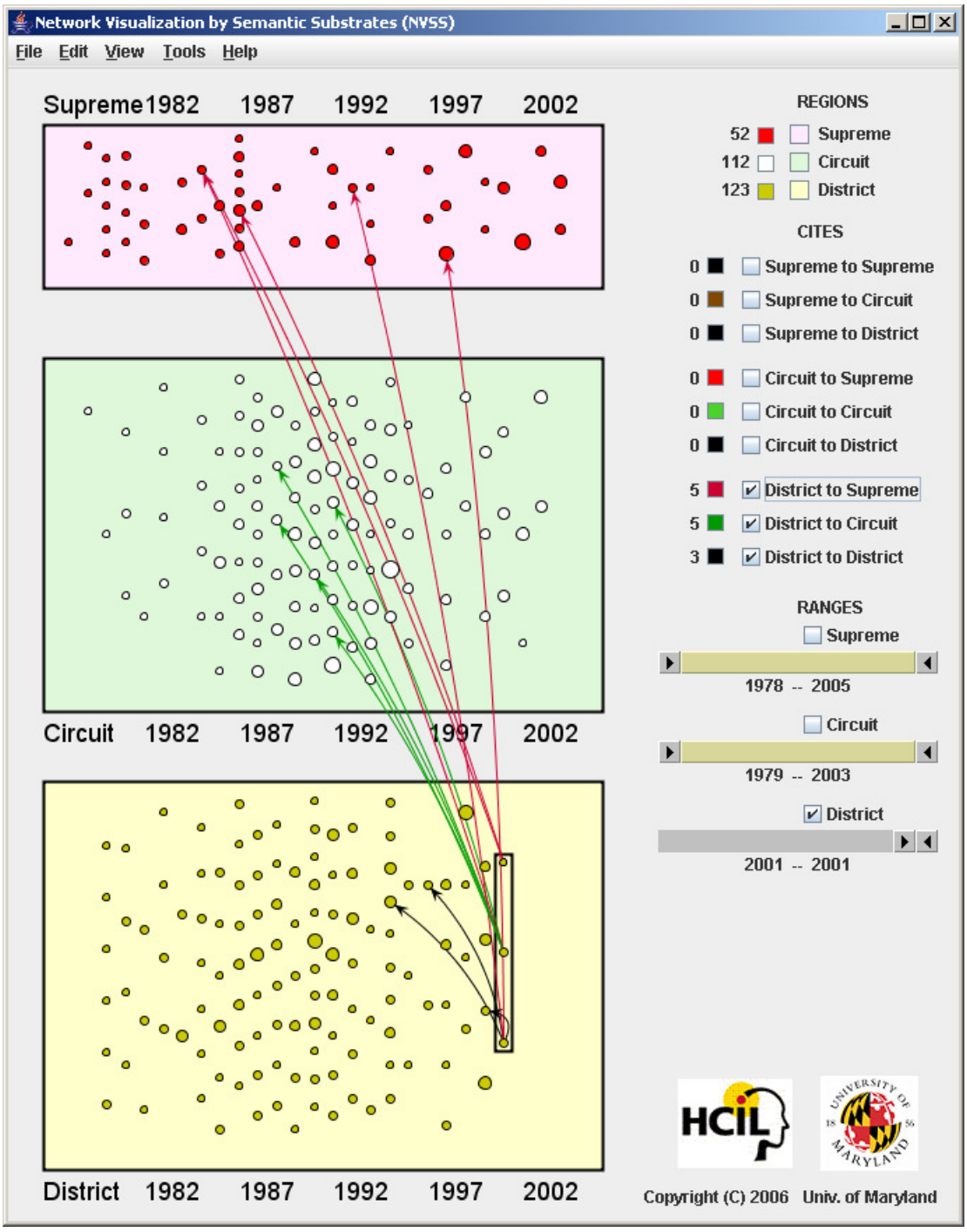


Figure 37 Having District Court cases in a third region shows an anticipated referencing pattern, that is, District Court cases have a short reference half-life. This display shows 287 nodes and 2032 links.

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." The political science researchers were pleased to see that the visual display added support to some of their conjectures such as this one about citation patterns for precedents. Furthermore, they were surprised to detect patterns that were not apparent 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 as to what factors make the exception to the rule occur. One question that the 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 38). This immediately confirms the expectation of the 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 38. A similar hypothesis for the District Courts is also confirmed 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 38 The layout for Circuit Court cases is now organized by the 13 Circuits and the link pattern shows the strong likelihood that cases will reference precedents within the same Circuit.

The 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 39. 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 see nodes and links more clearly. Figure 39 shows a still larger data set with 1,122 nodes and 7,645 links at a 1280x1024 resolution.

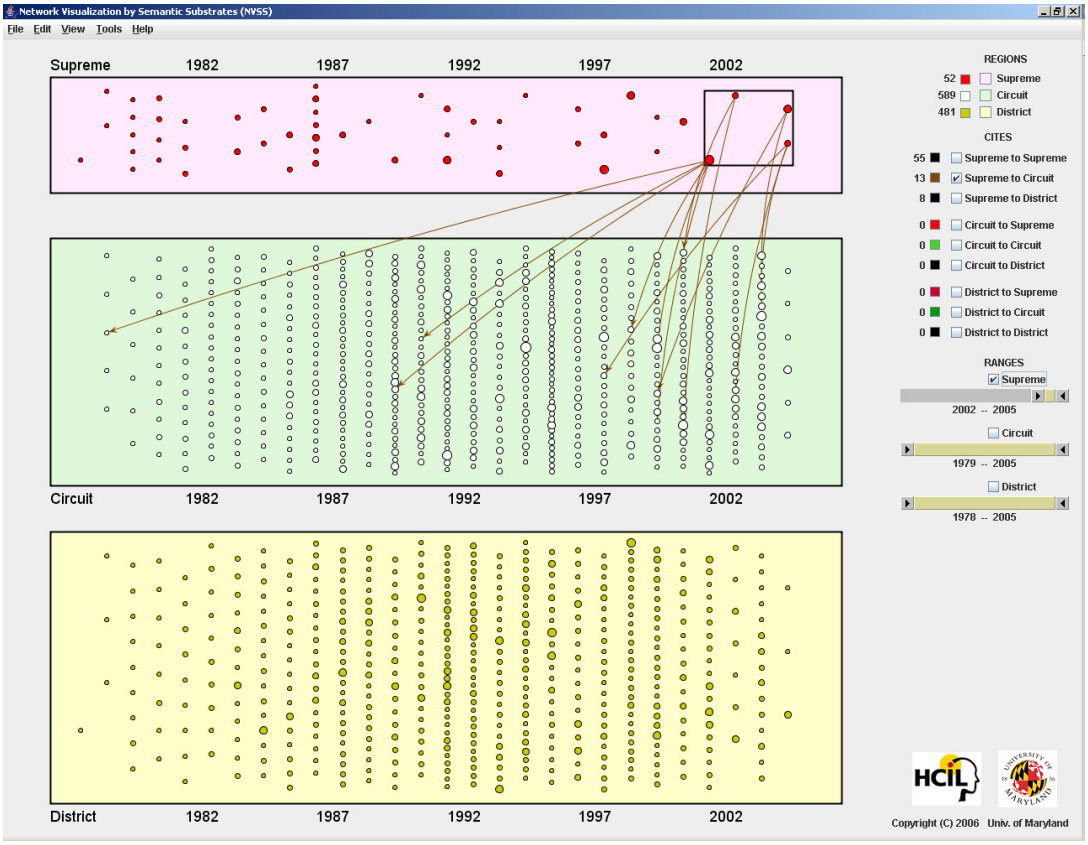


Figure 39 Displaying 1,122 nodes and 7,645 links at a 1280x1024 resolution. The relatively small number of Supreme Court cases is apparent, as is the similar number of Circuit and District Court cases.

5.2.1.2 Final Post78

In this section, the collaborators explore the final version of the post78 legal court case dataset. In this section, NVSS 2.0 is used (a later version of the software than the version in the previous section).

The author of this dissertation, as the tool expert, designed the substrates to meet the needs of their exploration. In this multidisciplinary project, the domain experts are knowledgeable in different aspects of the domain. The collaboration over 16 months covered data identification, data collection and filtering, followed by problem analysis to develop requirements for visualization.

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

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 40 shows the result of applying the first substrate to this dataset. The dataset is a subset of a larger dataset with 2345 nodes and 14,401 unique links and contains 287 nodes and 2032 links. The subset was selected by selecting all Supreme Court cases, Circuit Court cases that were cited more than 10 times, and District Court cases that were cited more than twice.

The link counts are as follows (from row to column, Table 3):

Table 3 Link counts for Initial Post78.

	Supreme	Circuit	District
Supreme	328	25	20
Circuit	775	757	55
District	395	240	96

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 heard 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 heard. For the "District" cases, *circuitNo* indicates the jurisdiction of the District Court that the case was heard. *inCites* is the number of citations to a case in the larger dataset. *outCites* is the number of outgoing citations from the case in the larger dataset. *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."

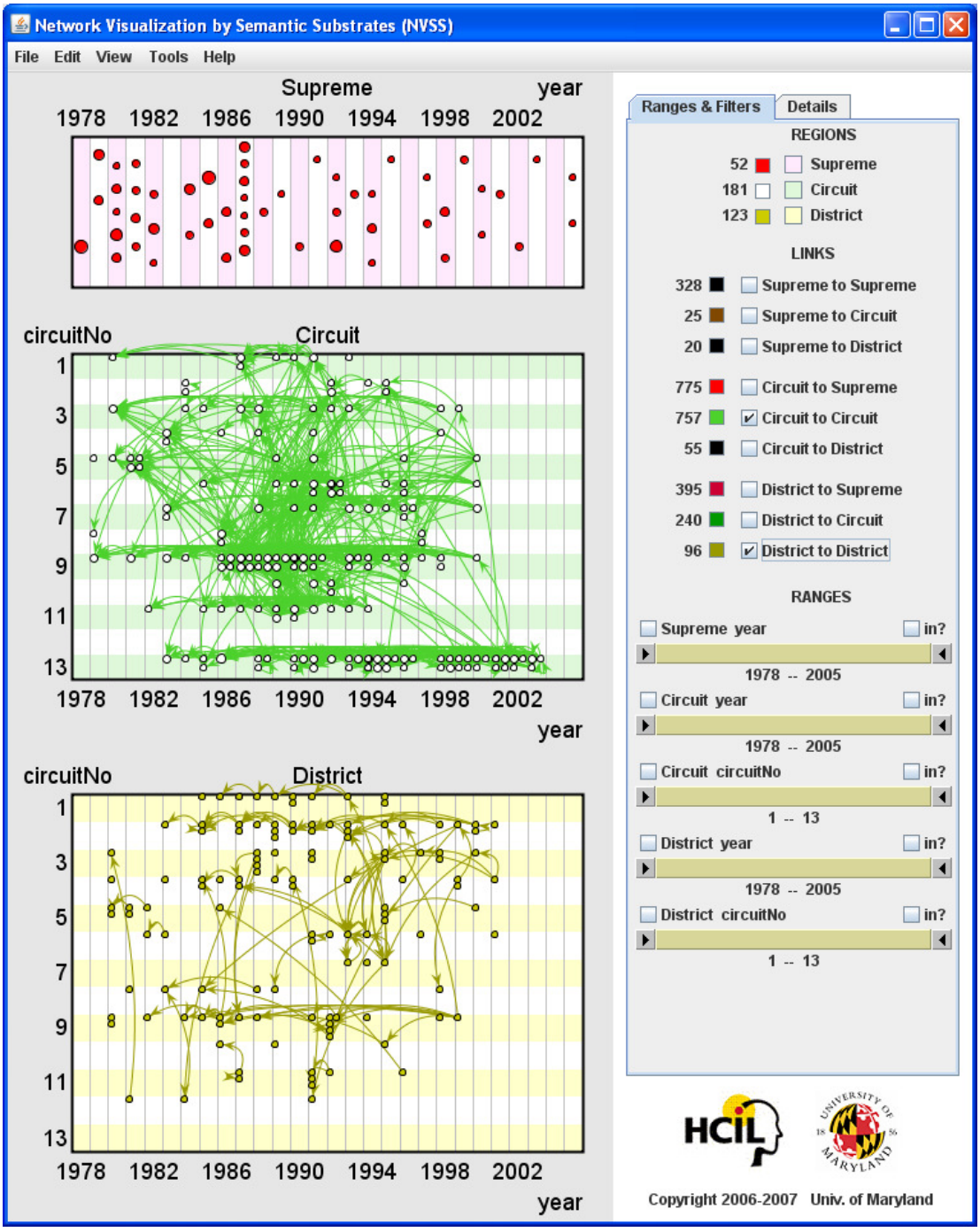


Figure 40 An initial semantic substrate is applied to a court case dataset, where nodes are court cases and links are citations from one case to another. Nodes are grouped into regions using the venue node attribute with "Supreme", "Circuit", and "District", while they are placed using *year* along the 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.

The semantic substrate in Figure 40 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.

The 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) would be shown (see Figure 40). 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 the 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 41).

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 41,

respectively). The filters work conjunctively (rather than disjunctively). As a result, the more filters applied on a region, the fewer links are shown. The filters restrict links either to incoming or outgoing links. In Figure 41, 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, an interesting result is achieved using filters. 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 to the other end of the range (this takes a few seconds by dragging the double-slider from the middle, between the arrows). 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.

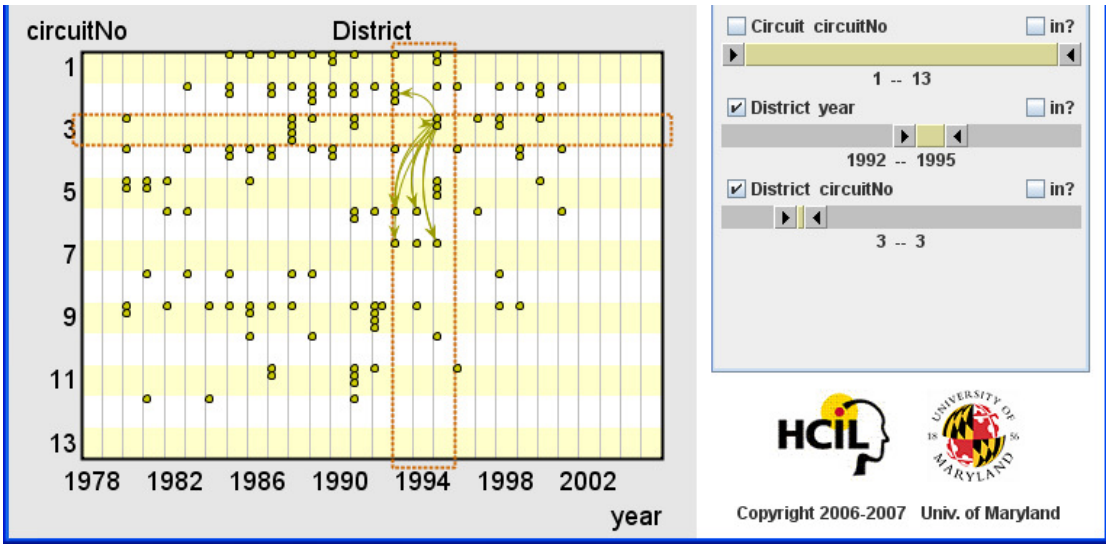


Figure 41 Upon seeing the tendency that court cases tend to cite within their circuit in Figure 40, a diversion from this tendency is isolated using link filters on the District region, which helps users clearly see them.

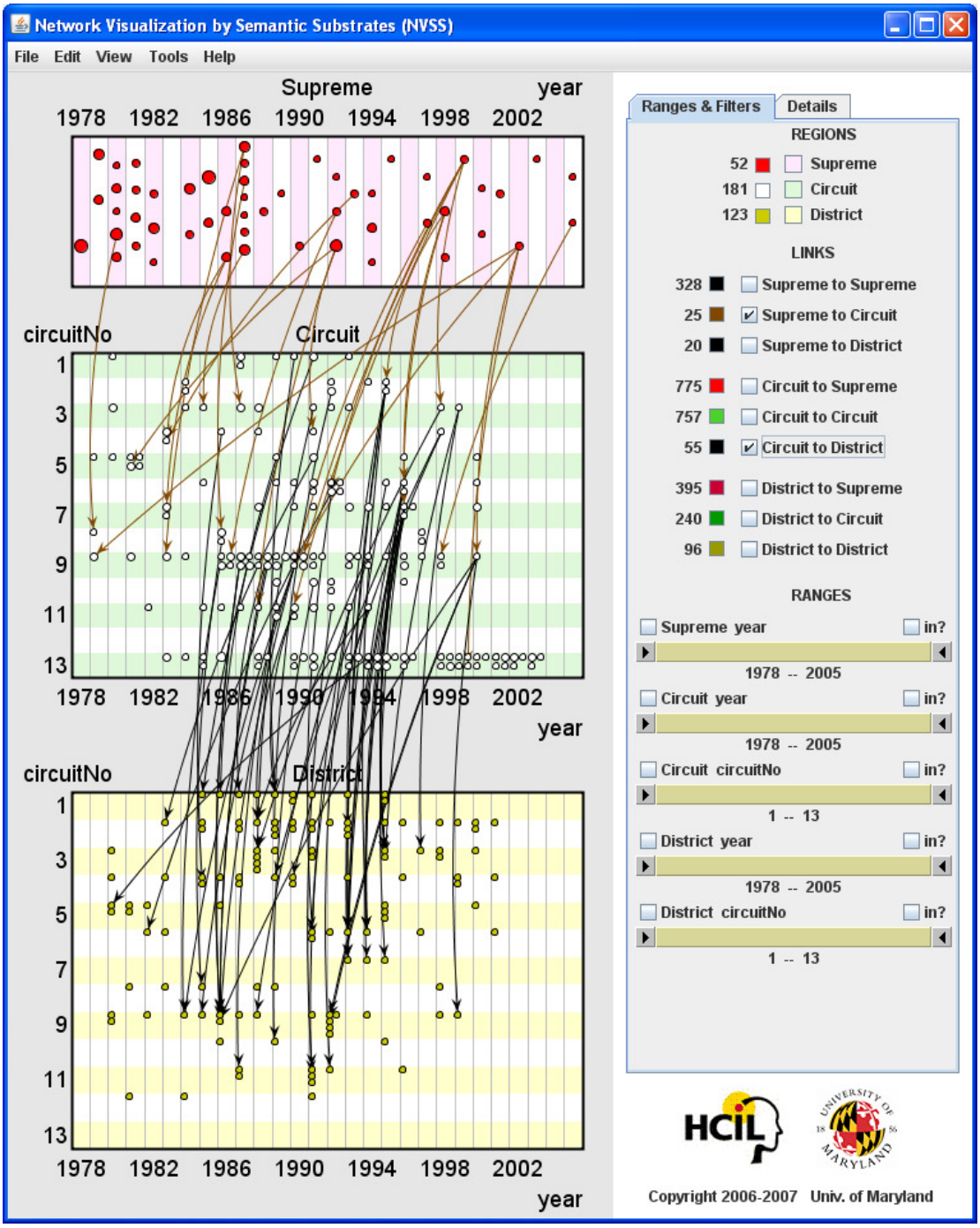


Figure 42 Due to having used the *year* attribute across regions consistently on the x-axis, it is easy to compare the citation patterns according to *year* across regions. The citation patterns indicate that although Circuit Courts tend to follow-up immediately after a case is appealed, it takes a longer time to do so for the Supreme Court, possibly due to their lengthy appeals process.

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 42).

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 42, 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 (cite) 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 42, 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, a different substrate was used on the same data.

To satisfy further requests from the domain experts, the initial substrate was opened, the attributes for the x- and y-axis for the Circuit and District regions were swapped, and the edited substrate was saved as a new substrate. Then, it was applied to the data to see Figure 43. 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.

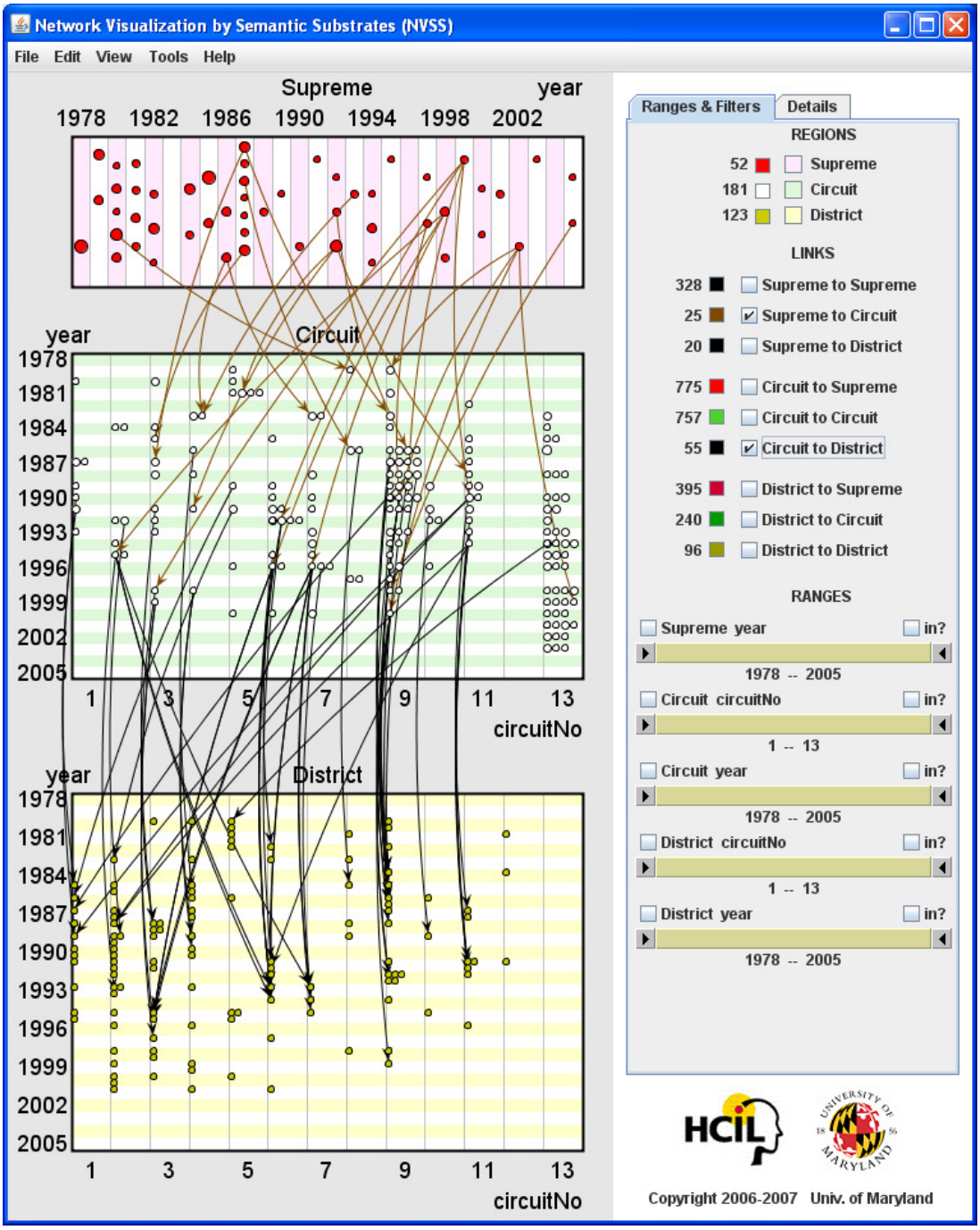


Figure 43 Looking at the same data using a modified substrate (the substrate in Figure 42 with swapped axes for Circuit and District regions) that aligns circuits using *circuitNo* along the x-axis of Circuit and District Court regions helps comparison between cases from these regions cases in terms 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.

Figure 43 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 the domain experts to see that there are Circuit Court cases that cite District Court cases 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 the 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 44). Using this substrate, citations from Supreme Court to Circuit and District Court cases are easily comparable. The domain experts saw no dramatic differences in the citation patterns.

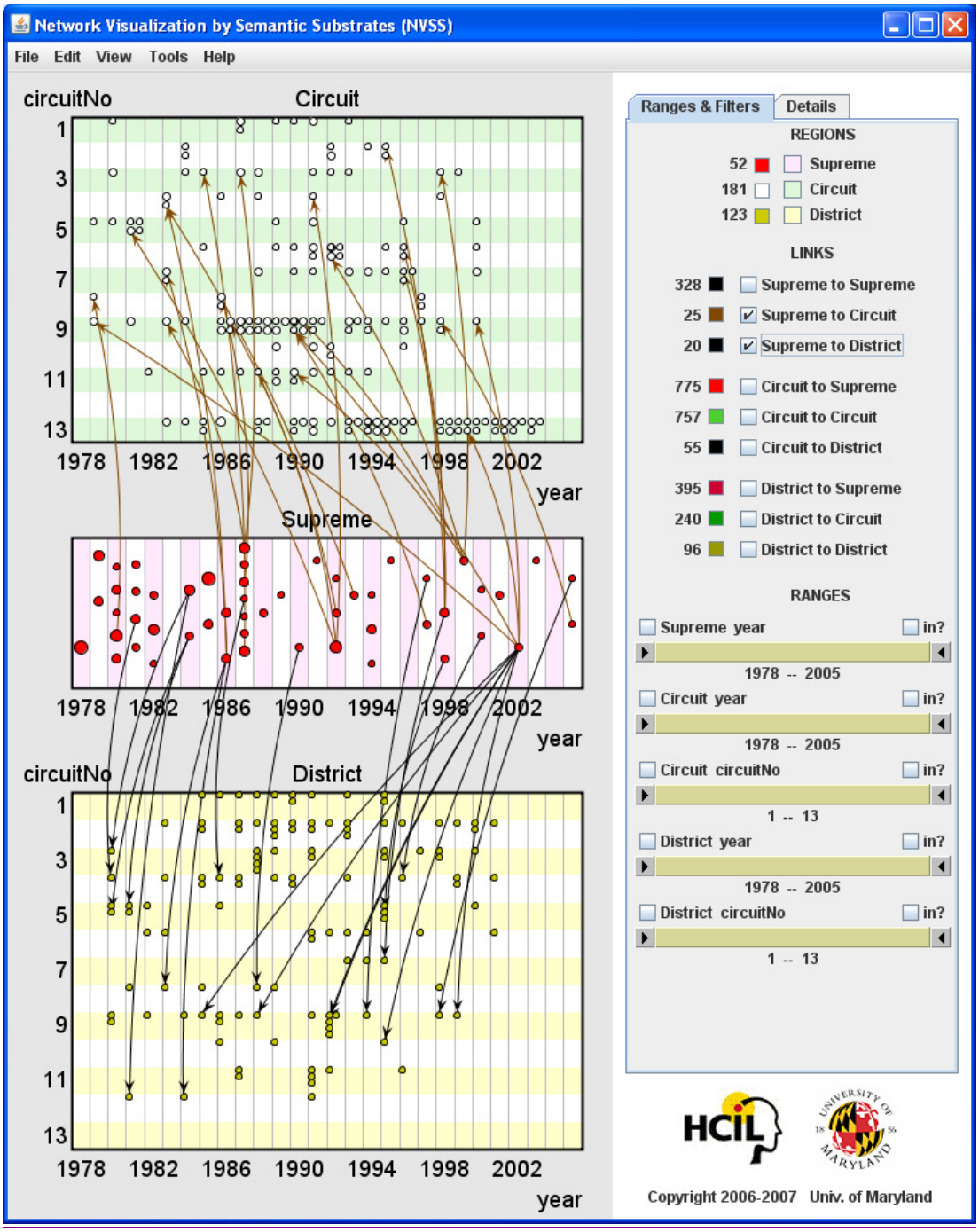


Figure 44 The substrate in Figure 42 modified to align the Supreme region with the other regions. The pattern of Supreme Court citations to Circuit and District Courts appear to be similar.

The domain experts were intrigued by the fact that there are Supreme Court cases that cite several District Court cases at once. They found this quite unexpected and worthy of further exploration. This is so, although somewhat differently, with the

Circuit Court cases, too. A possible explanation for the Supreme Court citing Circuit Court cases is that Circuit Courts participate in forming the law in their jurisdictional region and they have the potential to derive constructive legal responses to a difficult set of issues. When they do so, the Supreme Court may find their decisions useful as they address similar questions and cite them. This might indicate the Circuit Court cases that influenced the Supreme Court cases, an interesting phenomenon worthy of further research according to the domain experts.

The domain experts also wanted to explore the citations from the District region to the Circuit region. To help with this task, the previous substrate in Figure 43 was loaded and applied the data to generate Figure 45. 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 46). The domain experts did not expect to see Circuit Court citations to District Court cases outside their circuit. Circuit Courts are more authoritative than the District Courts and therefore are not expected to cite District Courts (Circuit Courts bear immediate responsibility for reviewing decisions handed down by District Courts within their jurisdictions. Thus, to avoid having their rulings

overturned, District Court judges are expected to adhere to established caselaw within their own Circuits, rather than drawing upon decisions handed down in other Circuit jurisdictions). One possible explanation is that the District Courts in the 2nd circuit may have specialized in a particular topic that the Circuit Courts found worthy of citing.

As the domain experts became familiar with the Substrate Designer's features, the semantic substrates became a new language of discourse for them, enabling them to generate new hypotheses. The visual presentation and user control of link visibility supported discussion, exploration, and communication within and beyond the group. The domain experts mentioned that NVSS is useful for them to look at the data quickly to find interesting phenomena and then narrow down to the cases to investigate. This would allow them also read those cases in a targeted way to answer the questions they formed while exploring the dataset. Overall, the 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. The domain experts have captured states of their exploration via screenshots 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.).

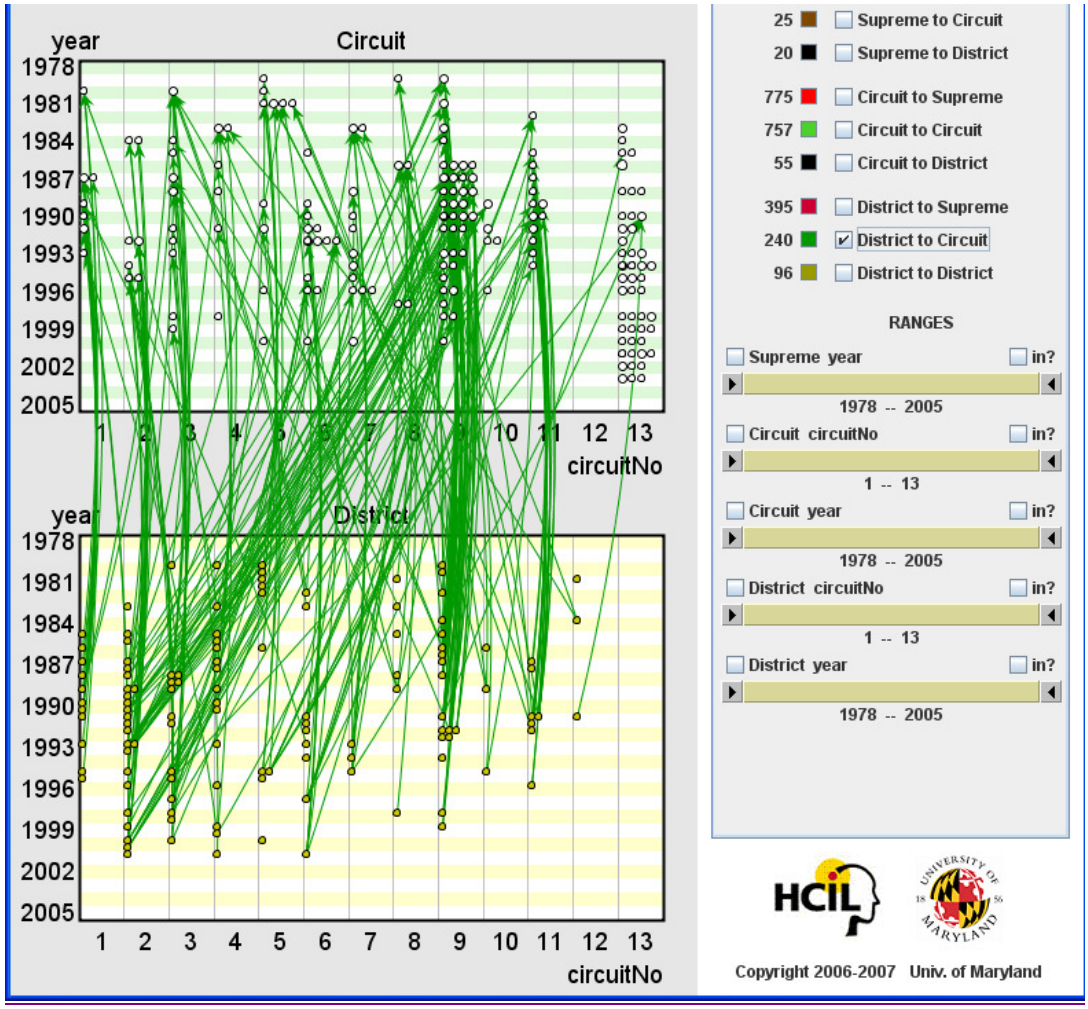


Figure 45 Applying the previous substrate to see District to Circuit citations reveal many parallel citations with a major exception of frequent citations from the 2nd to the 9th circuit. (Only the bottom part of the display is shown.)

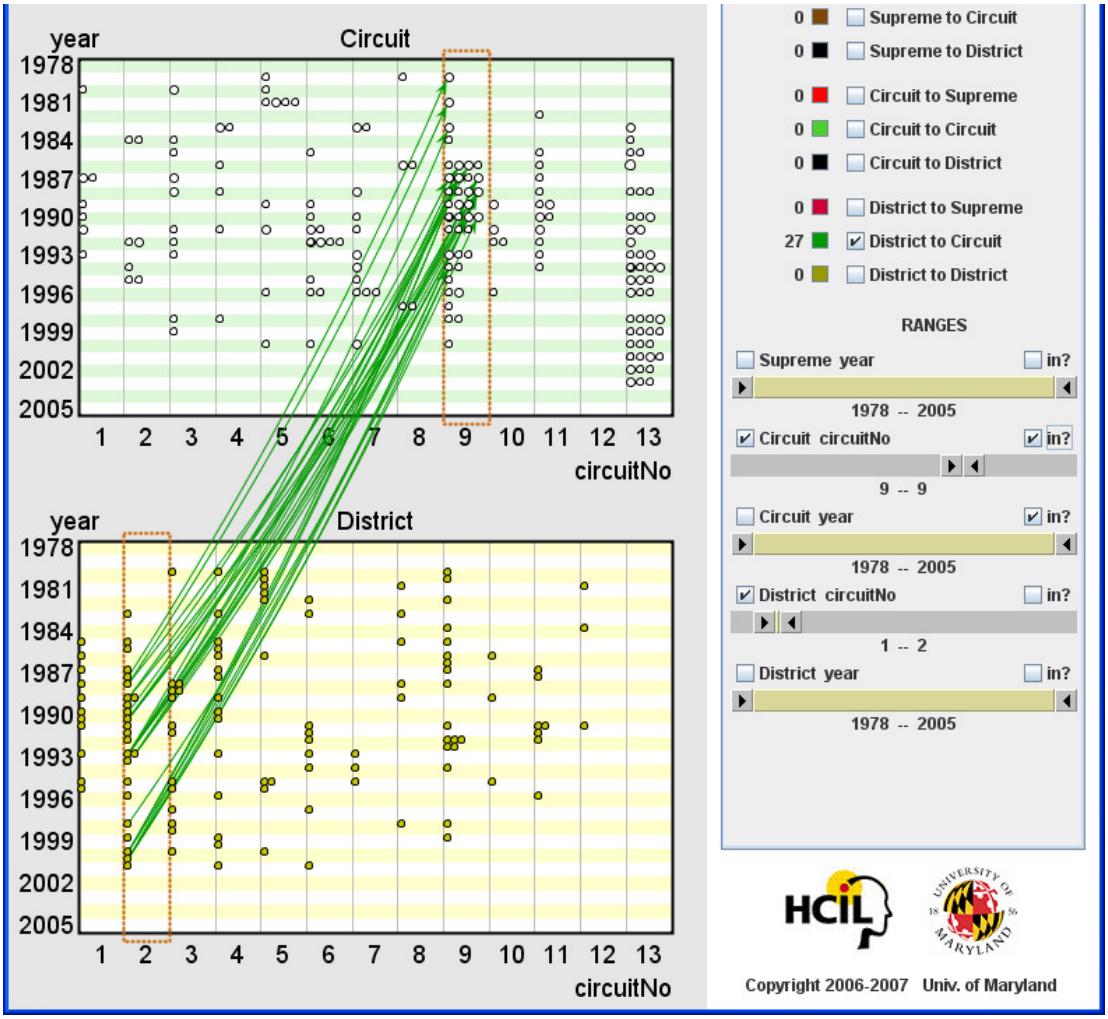


Figure 46 Restricting links in Figure 45 between the 2nd and 9th circuits using link filters reveal that citations are concentrated in three periods, namely 1989, 1993, and 2000. (Only the bottom part of the display is shown.)

5.2.1.3 Giants

This section uses a dataset called the Giants dataset. It is a processed version of the post1900 dataset. Giants were identified as the Circuit Court cases falling more than one standard deviation above the mean (average) number of incites (incoming citations) to all Circuit decisions overall. In other words, all cases were normalized within a given population (Circuit Court cases) to standard deviations of the mean for

those populations. The selection resulted in 87 Circuit Court cases, which were re-classified as "Giants" and treated separately from the other Circuit Court cases.

Regions are defined using *venue* attribute having values "Supreme," "Circuit," or "District." The Supreme region has Supreme Court cases. The Giants region has the mostly cited cases from the Circuit Courts of the United States. The Circuit region has the rest of the cases, which are less cited.

The dataset has 149 Supreme Court, 87 Giants, and 937 Circuit Court cases. This is the complete dataset derived from the post1900 dataset (no minimum incite restrictions). The link counts are as follows (from row to column, Table 4):

Table 4 Link counts for Giants.

	Supreme	Giants	Circuit
Supreme	643	17	57
Giants	436	265	133
Circuit	2948	755	1384

The dataset contains a total of 1173 nodes and 6638 links. The following sections illustrate the exploration of this dataset using NVSS 2.0. This case study contains a complete set of 24 figures. Since this is a summary example, only a few of those figures along with notes are included.

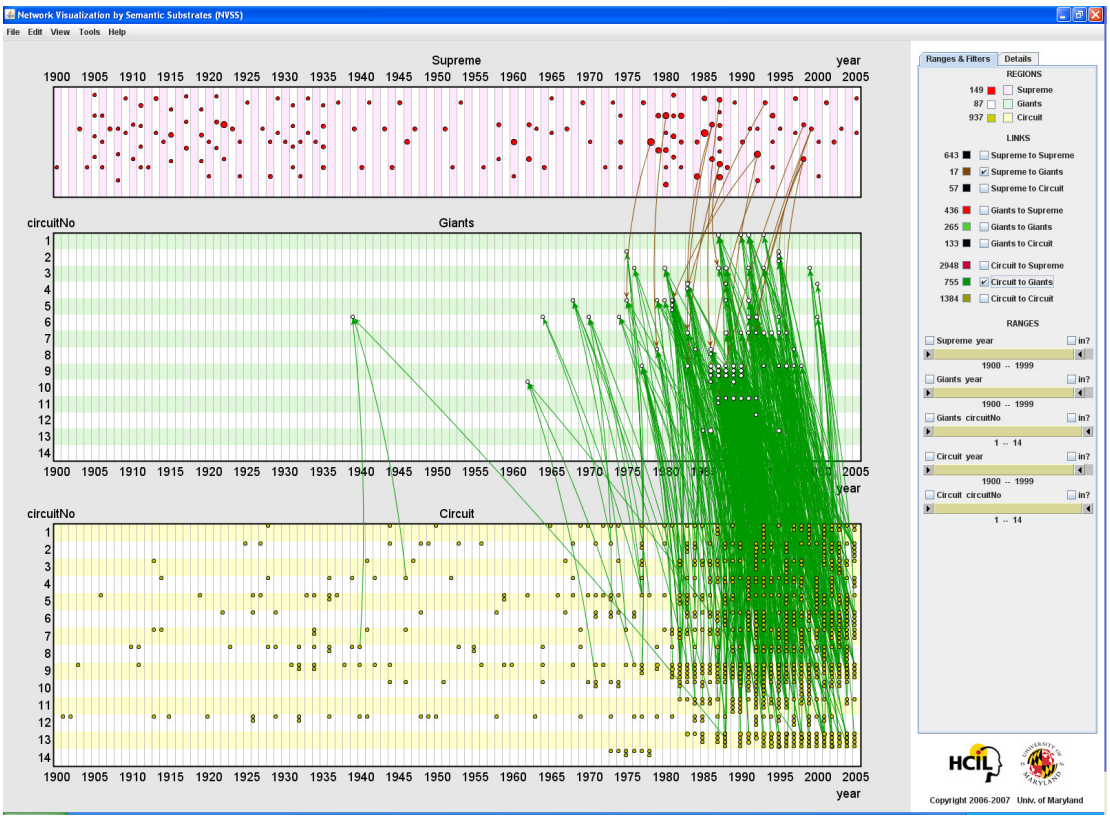


Figure 47 Incoming citations to Giants.

Figure 47 shows incoming citations to giants. A substrate is used such that the x-axis represents the year increasing from left to right, while the y-axis represents the circuit number increasing from top to bottom. In this dataset, 12 stands for the DC Circuit, 13 for the Federal Circuit, and 14 for the Temporary Emergency Circuit Court.

From this arrangement, there seems to be inactivity in the Supreme Court around 1936-1955. In the Giants region, there is only one case (around 1935), before 1960. Before 1965, there is another early Giants case.

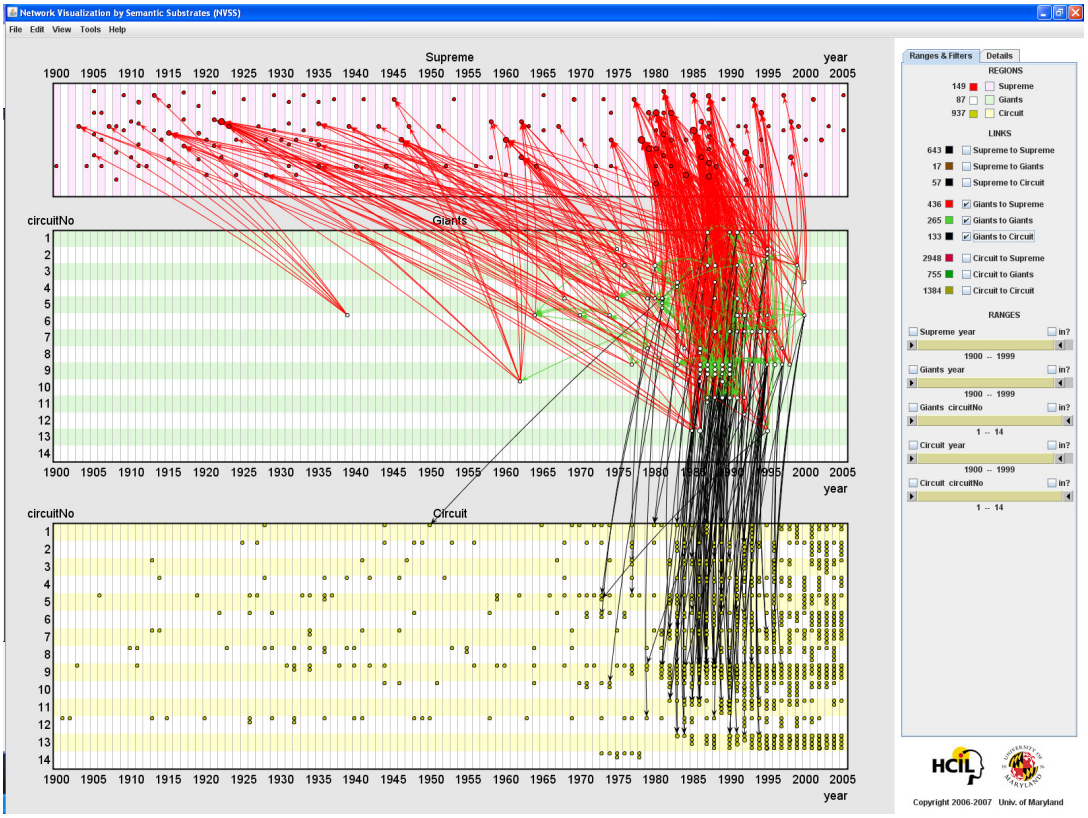


Figure 48 Outgoing citations from Giants.

Figure 48 shows outgoing citations from Giants. The outgoing links from the Giants region are enabled. There are 436 Giant to Supreme, 265 Giant to Giant, and 133 Giant to Circuit links.

Giants cite Supreme Court cases much more than the Supreme Court cites Giants. There are only 17 Supreme to Giants links as compared to the 436 Giant to Supreme links.

Next, the participants wanted to concentrate on the Giants to Supreme links. By disabling the other links, we arrived at the display in Figure 49.

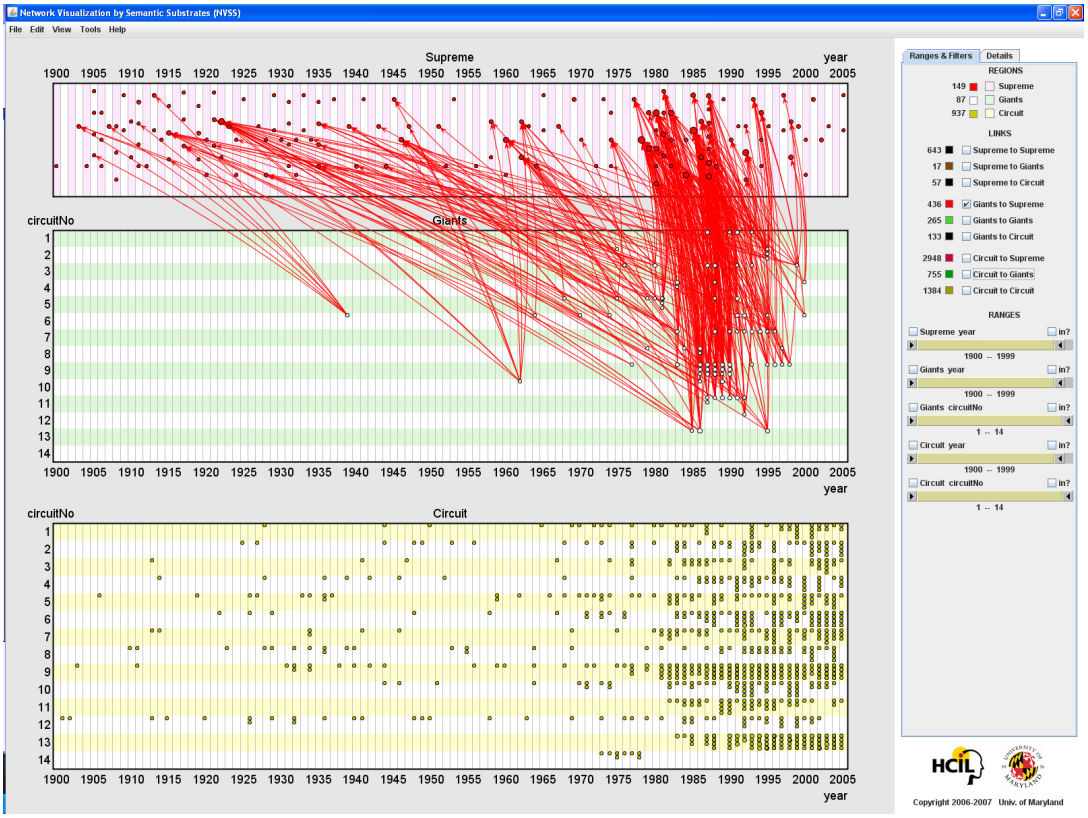


Figure 49 Giants to Supreme citations.

In Figure 49, only the Giants to Supreme links are observed. One of the case study participants (Ken) was interested to see in this view that not all Supreme Court cases were cited by the cases in the Giants region. The lack of connecting links to some nodes in the Supreme region shows this phenomenon.

The case study participant also found it interesting to be able to distinguish Supreme Court cases by the number of incoming citations from the Giants. Specifically, he was able to distinguish cases that were not cited at all, that were cited once, and more than once. It seemed that this categorization would be useful to them because this way they could distinguish more popular Supreme Court cases from less popular ones. In addition, this display helped them to think what the causes might be.

In a way, this display helped them to generate some hypotheses. One of the hypotheses was that language use affected the popularity of a Supreme Court case.

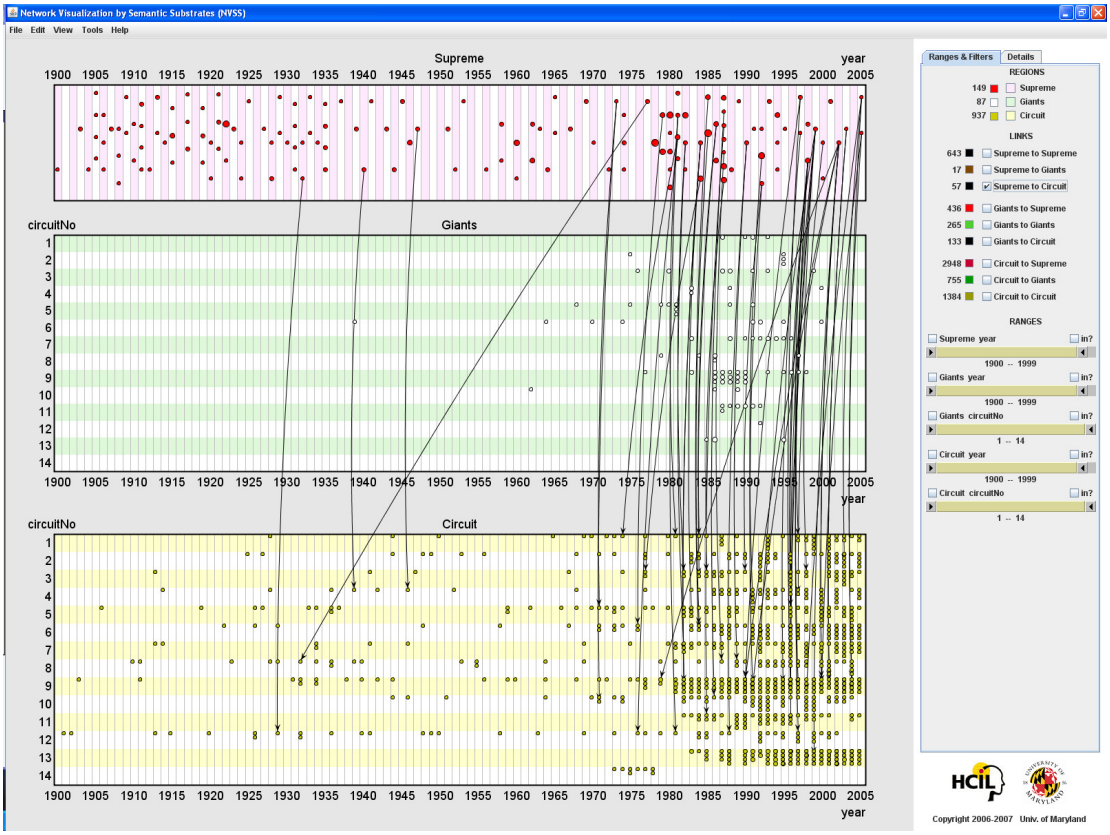


Figure 50 Supreme to Circuit citations.

In one of the displays, we were looking at Supreme Court cases citing Circuit Court cases (Figure 50). One of the citations had a large time difference (from around 1977 to 1933). When a citation is from a higher level to a lower level court (higher to lower level order is: Supreme, Circuit, and District), citations with such time differences have been very interesting to theparticipants. They usually wonder why a higher level court would cite a lower level court back in time as one possibility for this is that the higher level court may be seeking guidance from a lower level court. These instances usually would make them curious about the cases and they would

look up the information using the "Details" tab of NVSS and then the text of the cases to investigate the reason for the citation. This seemed to help them in their goal of trying to find influences among courts over time.

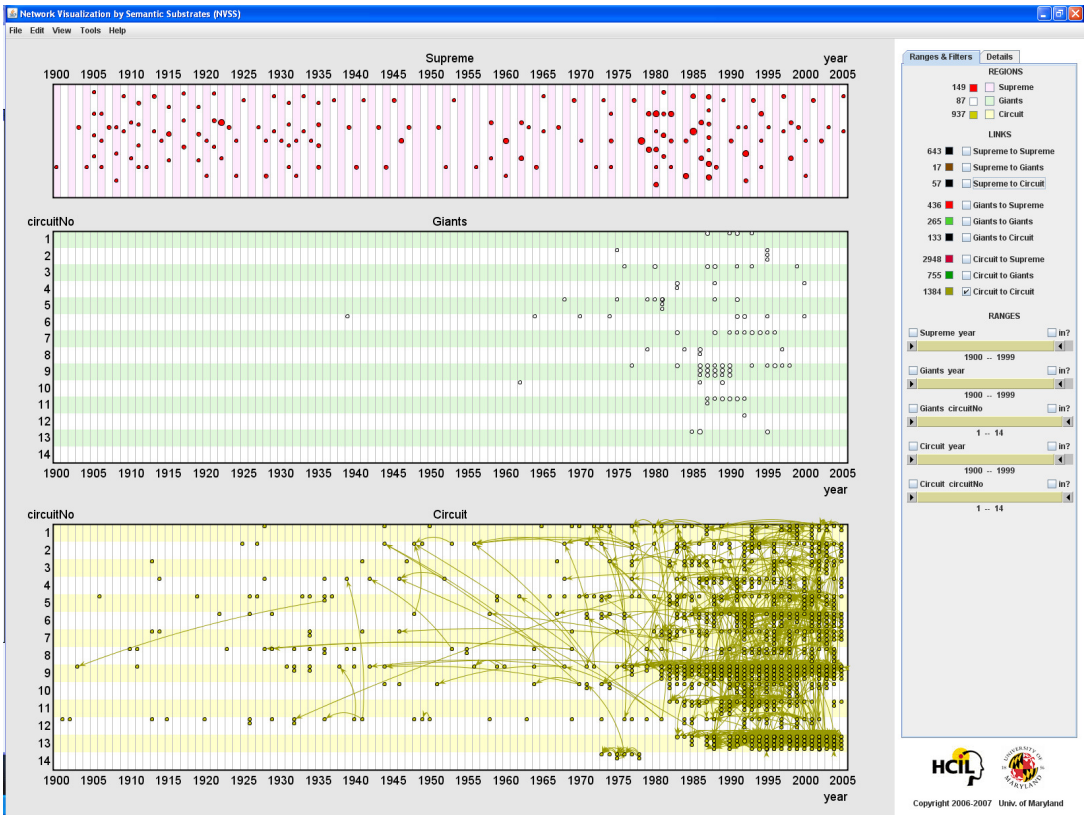


Figure 51 Circuit auto-citations.

Figure 51 shows Circuit auto-citations (Circuit Court cases citing Circuit Court cases). These Circuit Court cases are not Giants (i.e., the highly cited portion of Circuit Court cases determined by a metric mentioned before).

In the figure, parallel citations are observed. In other words, Circuit Court cases cite other Circuit Court cases that are in the same circuit (indicated by horizontal links). The non-horizontal links indicate citations to other circuits. The earlier periods have sparse citations.

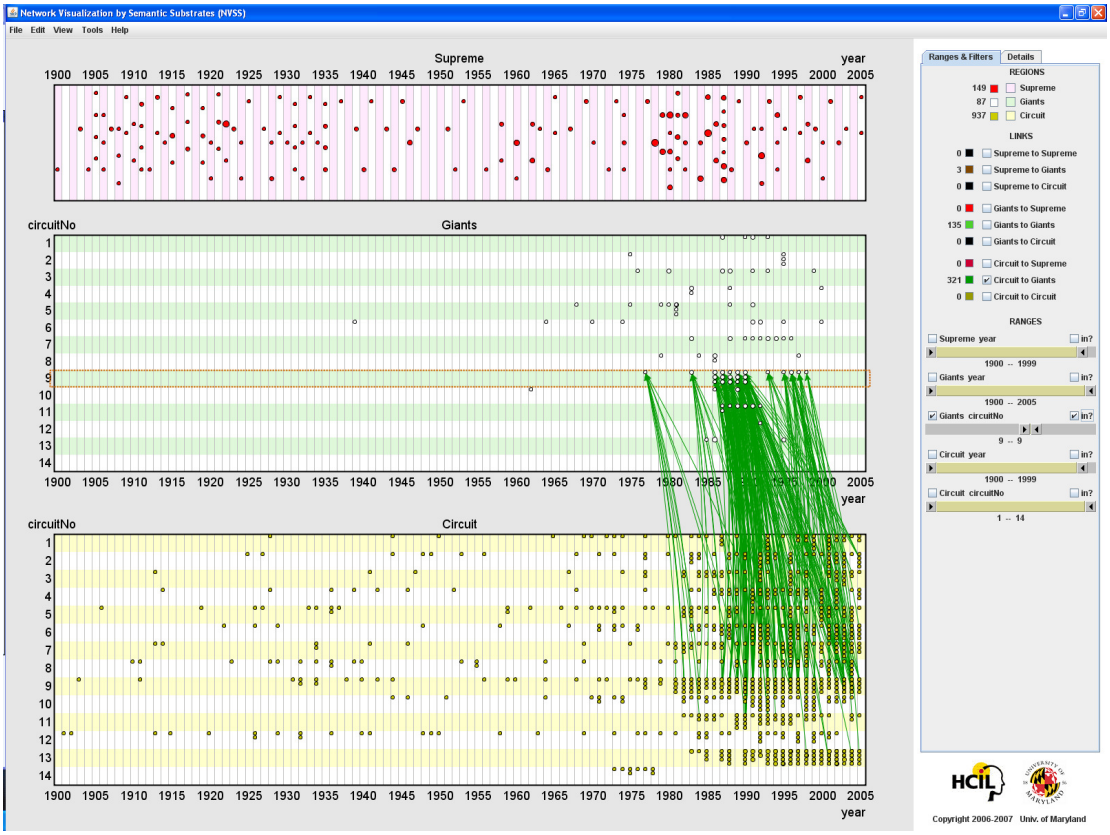


Figure 52 Circuit to 9th Circuit Giants citations.

The case study participants were interested to see the Giants in the 9th Circuit and the citations to them. Using the Circuit to Giants link filter and restring the incoming links in the Giants region to the 9th Circuit resulted in the citations to be visualized they wanted to see (Figure 52). However, they could not identify and categorize the many citations coming from the Circuit region. The author of this dissertation suggested that they could see which circuits the citations are coming from by modifying the substrate to align circuits vertically. They took the suggestion and were interested to see in the modified substrate. As a result, the author of this dissertation modified the substrate and applied the same filters to show the data as in the next figure (Figure 53).

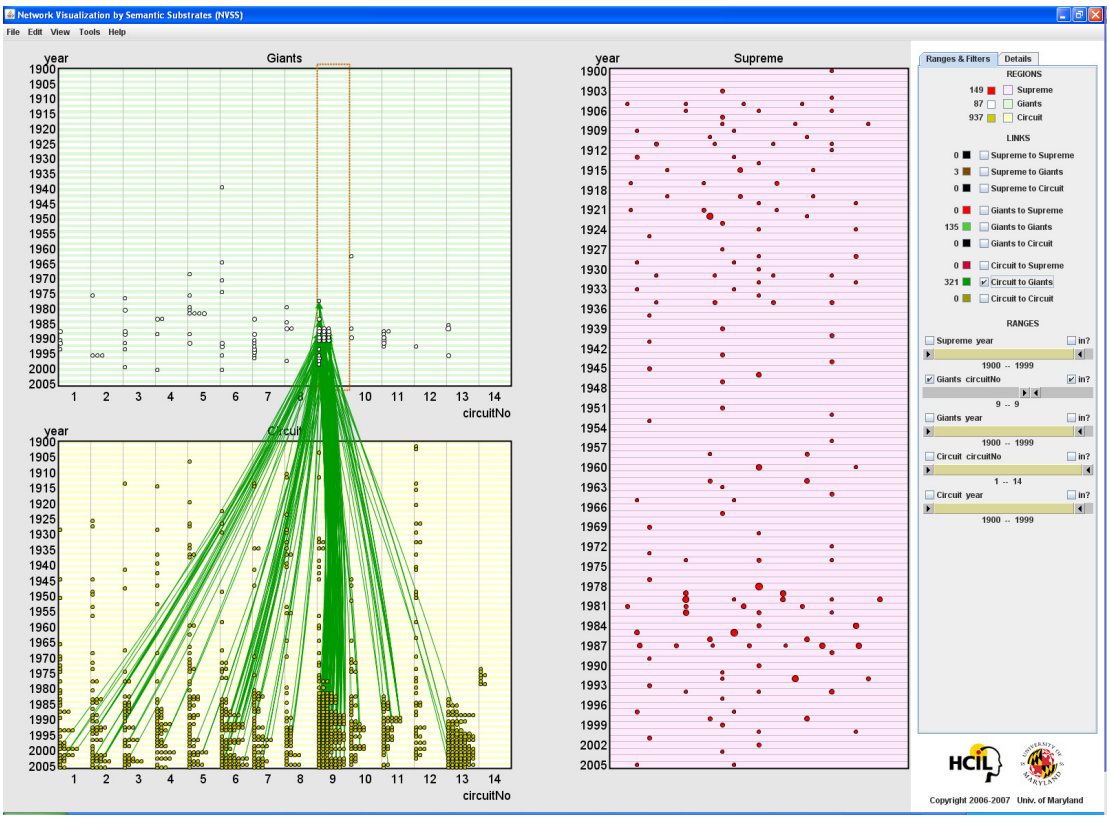


Figure 53 Circuits to 9th Circuit Giants citations with circuits aligned.

This figure shows circuits to 9th Circuit Giants citations with Circuit Courts aligned (using the modified substrate). Using this modified substrate allowed the case study participants to see which circuits cite the 9th Circuit Giants as well as the distribution. They were satisfied to see where the external citations were coming from (by circuit).

The case study participants wanted to see the relationship of the two early Giants cases to the Supreme Court and other Circuit Court cases. The following two figures show each situation separately.

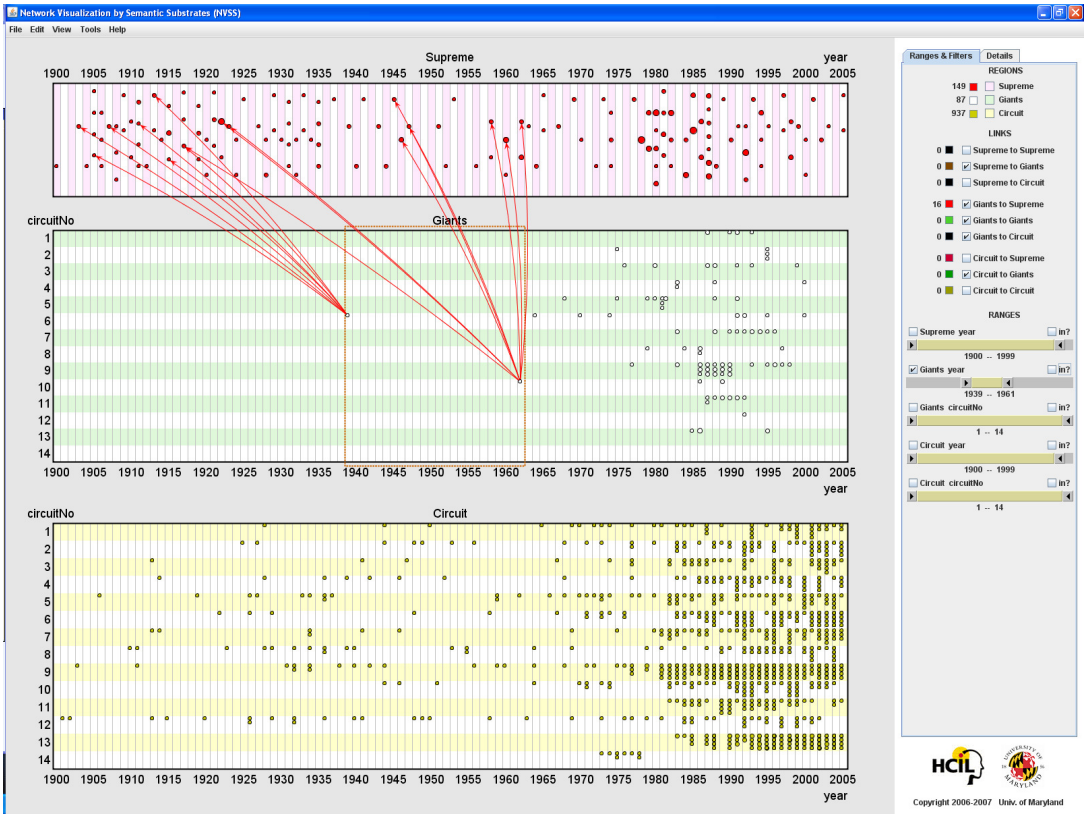


Figure 54 The citations from two early Giants cases to the Supreme Court cases.

Figure 54 shows citations from the two early Giants cases to the Supreme Court cases. Two of the three case study participants (Ken & Wayne) commented on this display and said they were interested to see this pattern as what they saw was unexpected: the two Giants cases cited a different set of Supreme Court cases. They said that it would be expected if they cited the same set of cases. Their citation of different Supreme Court cases was unexpected to them.

In addition, they were surprised that the earlier Giants case was not citing a well-known Supreme Court case in 1939 (this was communicated as "Mahon" among the case study participants; the case is: Pennsylvania Coal Co. v. Mahon, 260 U.S. 393 (1922)). One of the case study participants (Steve) offered two possible explanations. One explanation was that the earlier cases were actually not regulatory

takings cases. Another explanation was that the later cases might have superseded the earlier cases.

The next figure (Figure 55) shows the incoming citations to the two early Giants cases. This view revealed that there were only 3 citations to the earlier Giants case. They were surprised that with only 3 citations a case could become a Giant. Consequently, they questioned the validity of the selection of the cases as Giants. However, their investigation showed that their selection was valid and 3 was a sufficient number for incoming citations for a Circuit Court case to be categorized as a Giant.

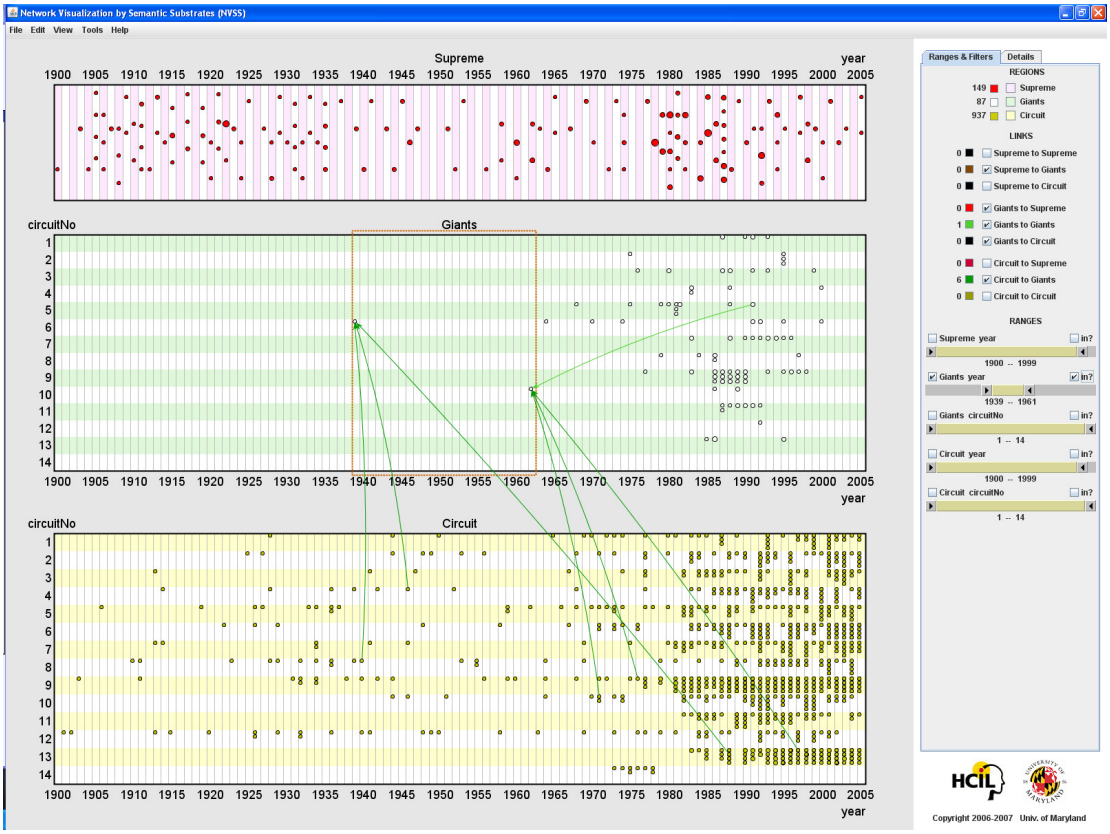


Figure 55 Incoming Circuit Court citations to the two early Giants cases.

5.2.1.4 Pre78

In this section, a subset of the Pre78 dataset of the CITE-IT project is used. The Pre78 dataset has court cases from the legal domain, specifically court cases believed to be regulatory takings cases.

Regions are defined using the *venue* attribute having values "Supreme," "Circuit," or "District." The Supreme region has cases from the Supreme Court. The Circuit region has cases from the Circuit Court. The District region has cases from the District Court.

This subset has 97 Supreme Court, 170 Circuit, and 226 District Court cases. This dataset is a subset because it does not contain the other court cases (bankruptcy, tax, etc.); however, it contains all Supreme, Circuit, and District Court cases in the dataset. The links are as follows (from row to column, Table 5):

Table 5 Link counts for pre78.

	Supreme	Circuit	District
Supreme	108	6	18
Circuit	300	73	70
District	393	88	114

This dataset contains a total of 493 nodes and 1170 links. The following sections illustrate the exploration of this dataset using NVSS 2.0.

This case study contains a complete set of 20 figures. Since this is a summary example, only a few of those figures along with notes are included.

Figure 56 provides an overview of the dataset. In this figure, a substrate is used such that the x-axis represents the year increasing from left to right, while the y-axis represents the circuit number increasing from top to bottom.

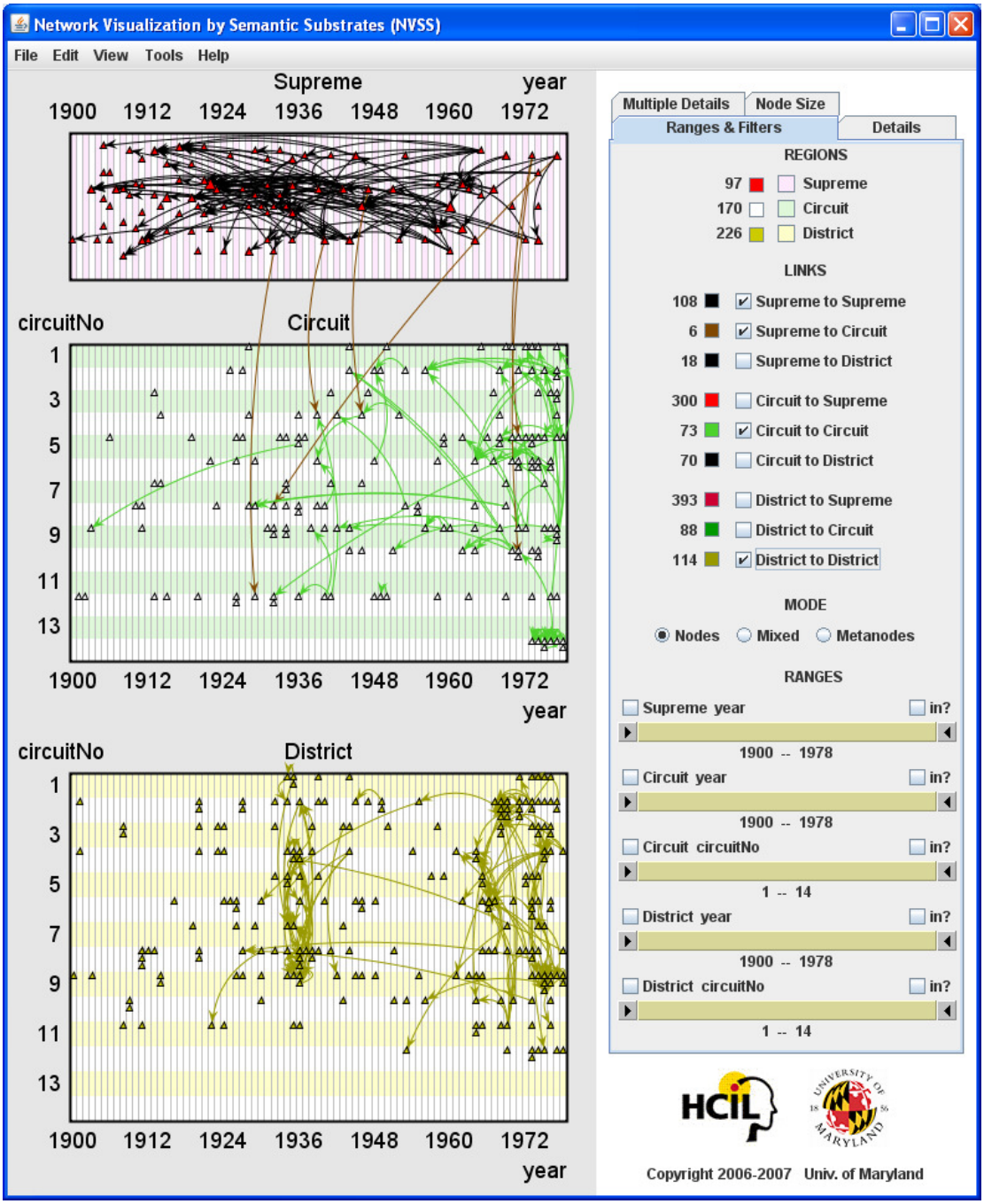


Figure 56 The dataset with links within regions and Supreme to Circuit links.

In this dataset, 12 stands for the DC Circuit, 13 for the Federal Circuit Court, and 14 for the Temporary Emergency Circuit Court. One Supreme Court case cites a very early Circuit Court case. There is high activity among District Court cases around 1939. Figure 57 shows the Supreme to District citations. The case study

participant (Wayne) noticed and commented that they are concentrated around 1939 and after 1965. He said it was interesting that they were concentrated around 1930s and that they were to District Courts. Also, he thought that in 1930s there were not many Supreme Court cases. At the same time, he was communicating his observations and thoughts to two other team members over a voice connection.

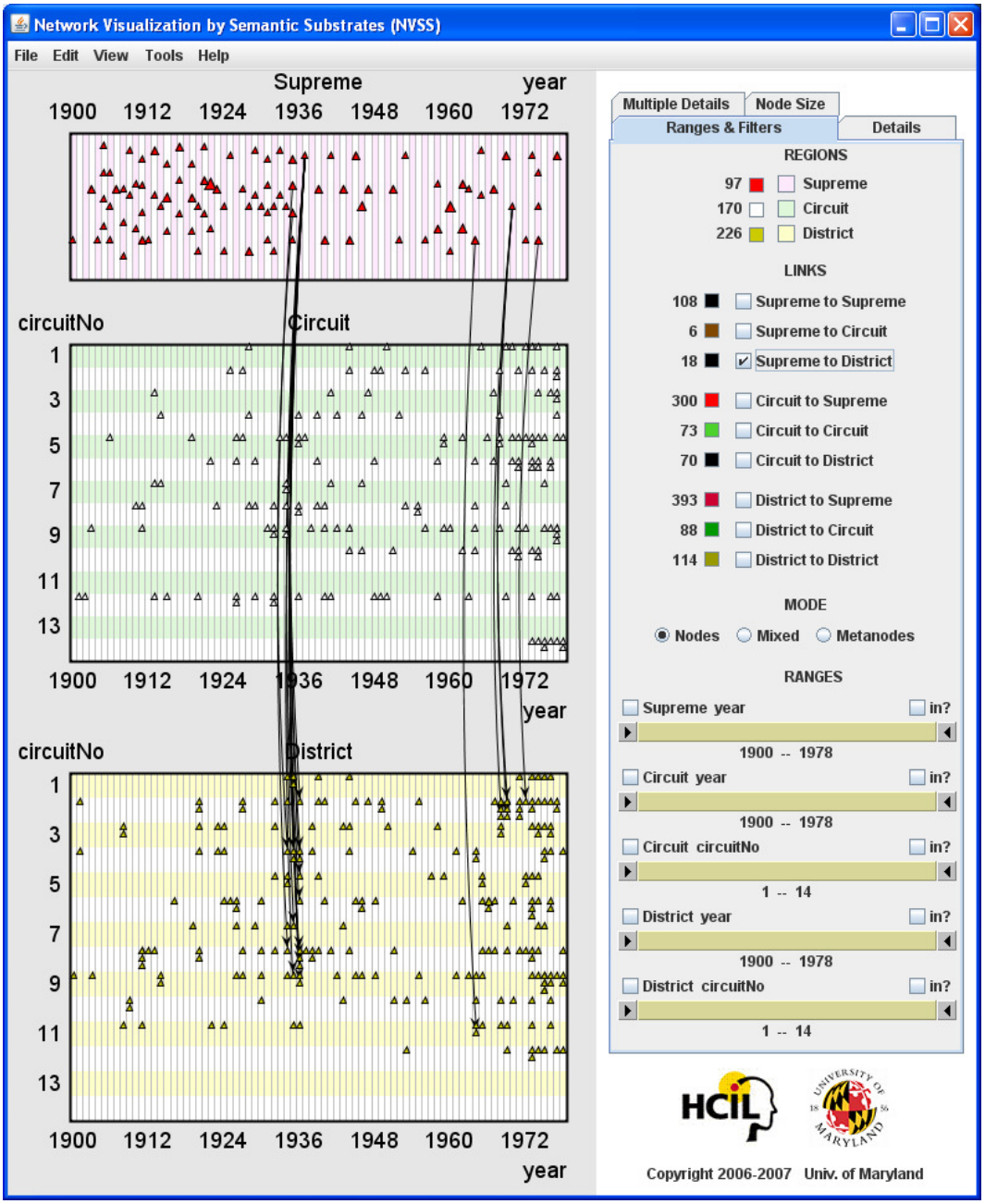


Figure 57 Supreme to District citations are visible.

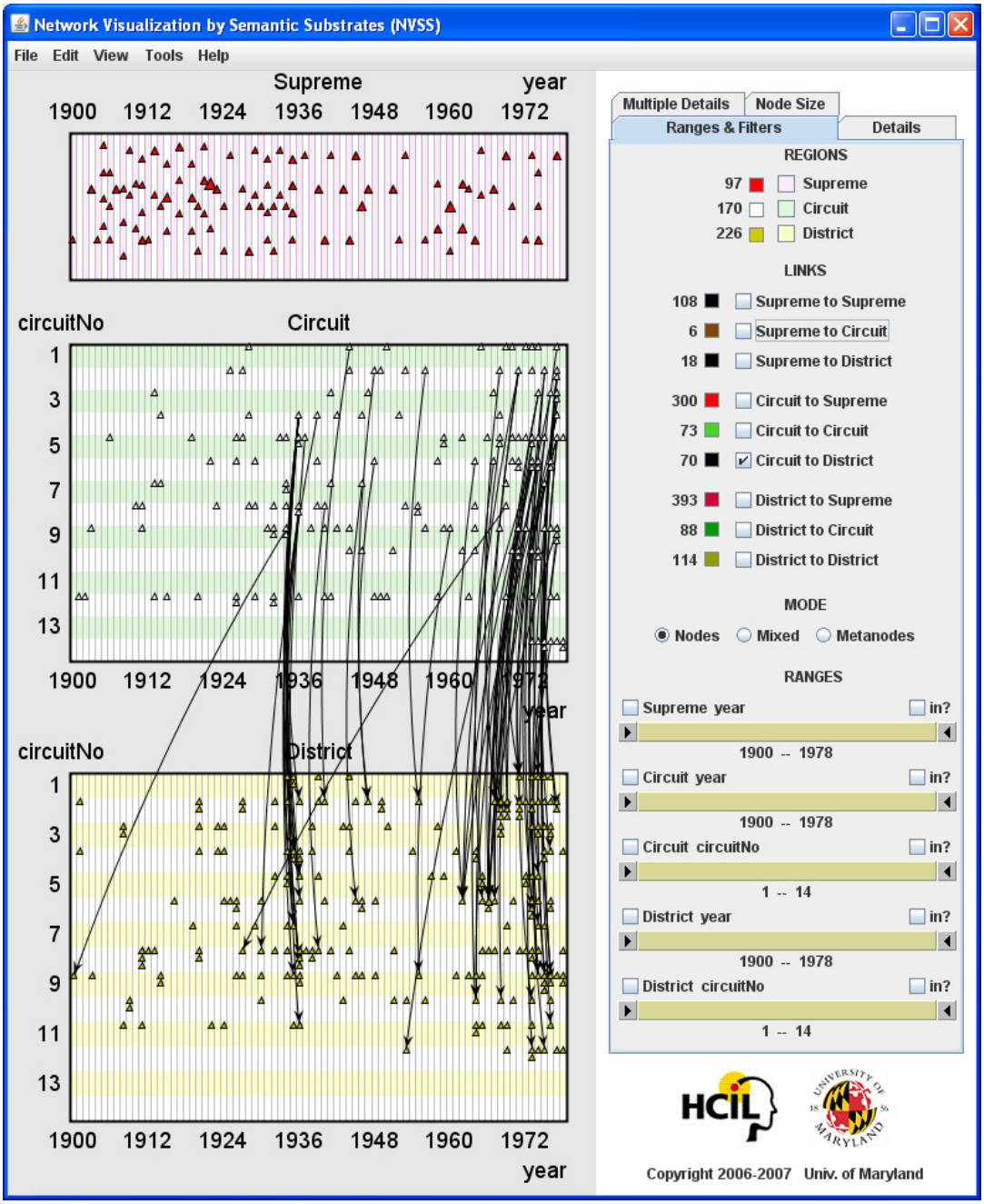


Figure 58 District to Circuit citations.

Next, the case study participant wanted to see whether those District Court cases were being cited by Circuit Courts. We enabled the Circuit to District links and disabled the others to see the patterns (Figure 58). We saw that also Circuit Courts

were citing District decisions around the year 1939. He commented that the period of 1930s had distinct references.

Next, the case study participant wanted to see how the Supreme Court cases were being cited. We were going to enable the Circuit to Supreme links. Since there were many, we applied a filter on the year to first look at the earlier period (Figure 59).

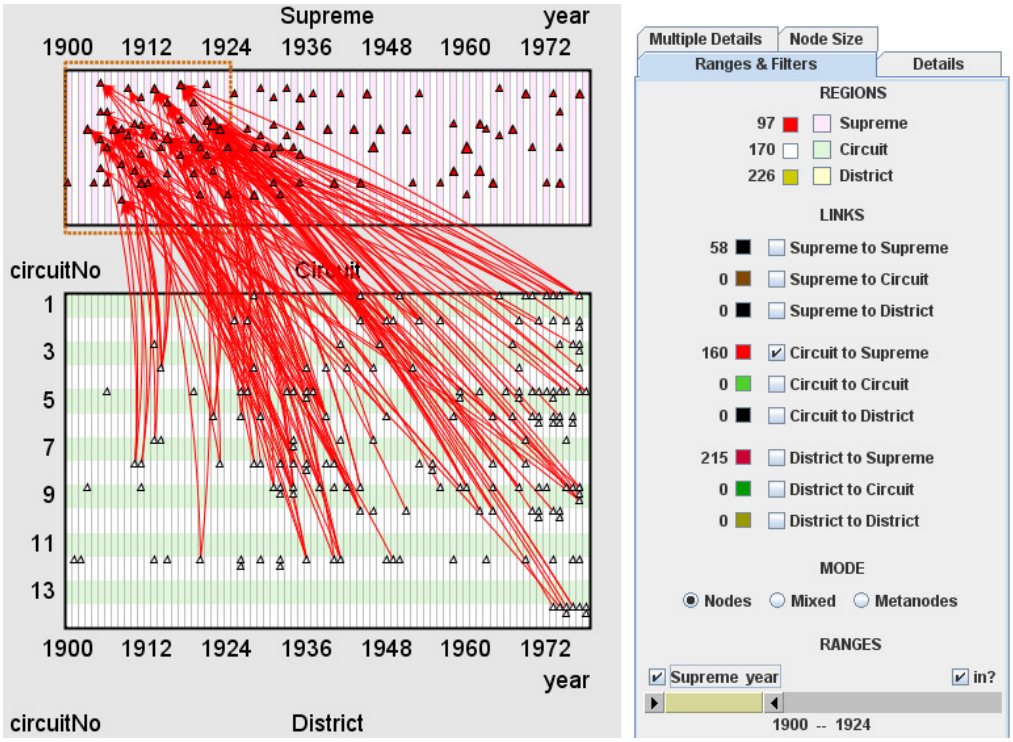


Figure 59 Circuit to Supreme citations in the earlier period.

He revealed his thoughts and said that the Supreme Court was active making decisions in the entire pre78 period. Next, the case study participant wanted to see all links also to get an overall view. Hence, we disabled the filter on the year to see all Circuit to Supreme links (Figure 60).

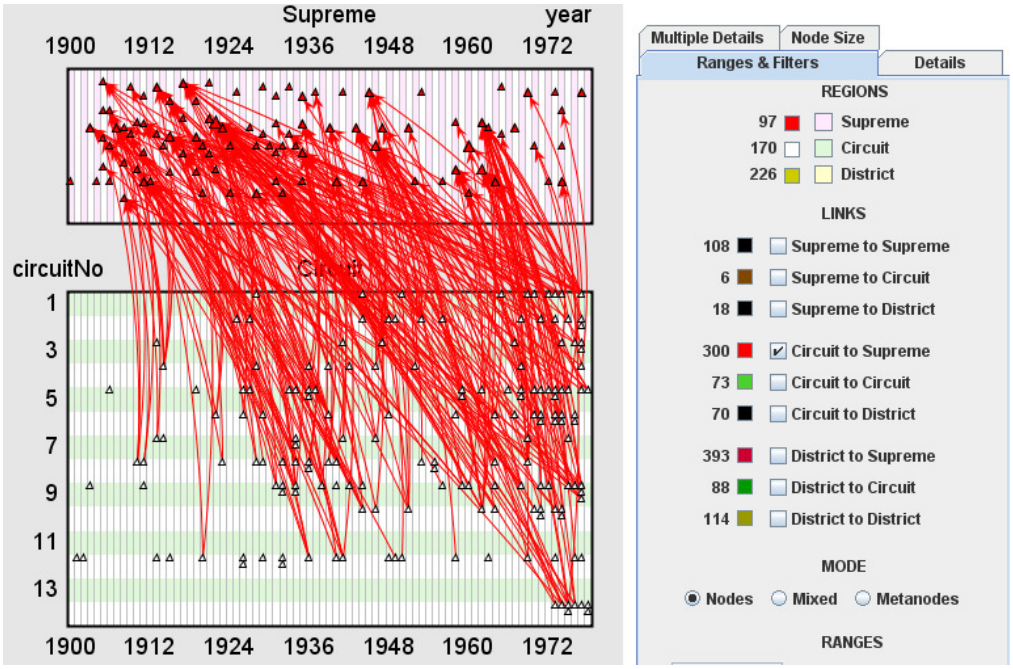


Figure 60 All citations to Supreme Court from Circuit Court.

From this view (Figure 60), the case study participant concluded that the references to the Supreme Court was lower in the later period than the earlier period. He also wanted to see the District to Supreme Court citations. The following figure (Figure 61) shows one such display that we looked at together. There were many citations as expected. Although the number of citations is available on the control panel, it seems that the case study participant wanted to see the links visually. Perhaps, this gave him an idea about the distribution of the links, possibly identifying patterns of gaps, outliers, and high or low activity period. Another possibility is that the case study participant may have simply forgotten that the numbers are provided on the control panel. It seems that he did not see or could not identify any patterns of interest in this view (Figure 61). However, it seems that the large volume of District to Supreme citations were normal when asked his expectations and any reactions about the display.

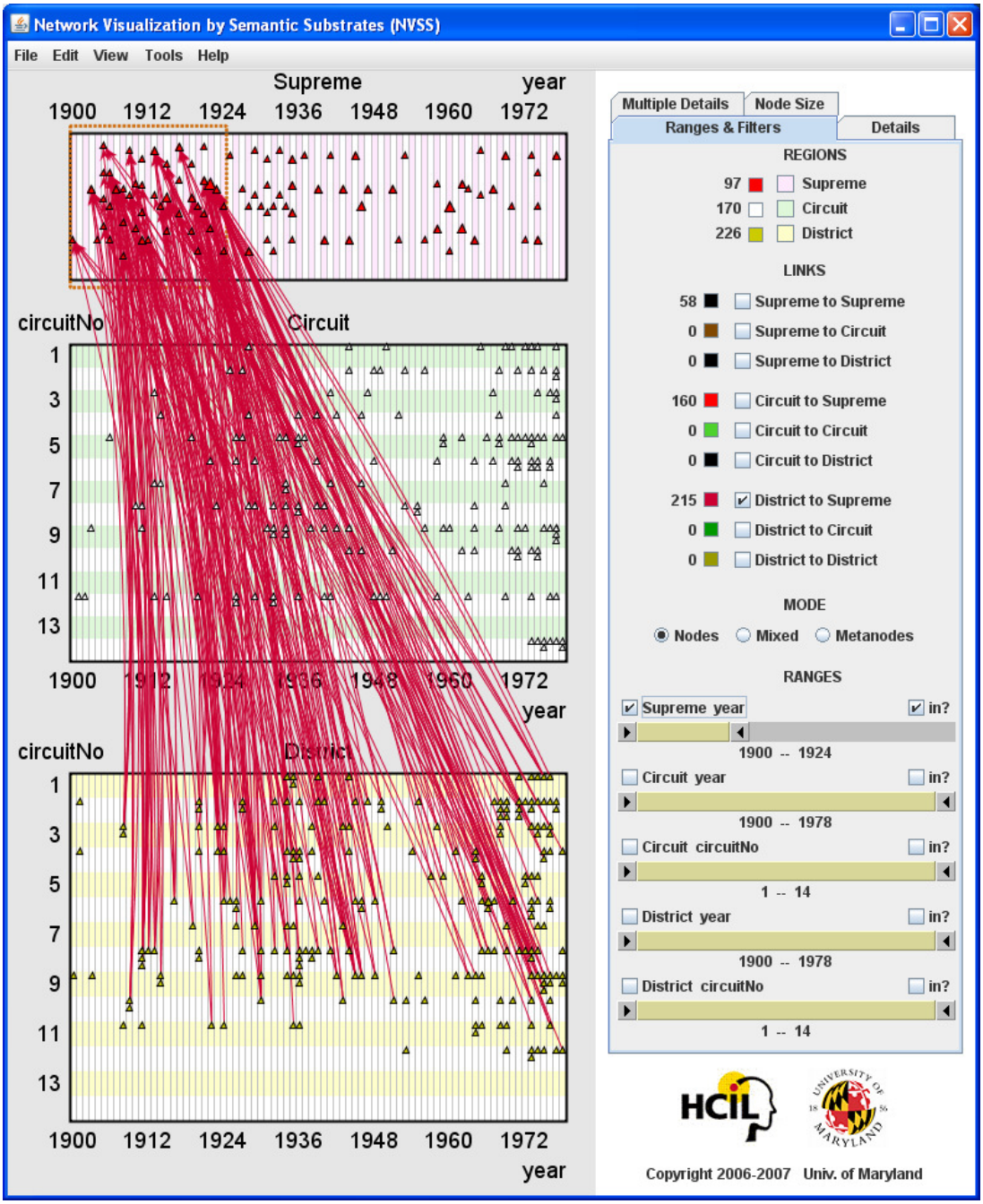


Figure 61 District Court citations to earlier Supreme Court cases.

Then, we looked at the Supreme to Circuit citations (Figure 62). He was surprised to see that there were only 6 citations. He was very skeptical about this view

and asked to check for the correctness of the visualization, and the dataset. Those were checked and found to be correct.

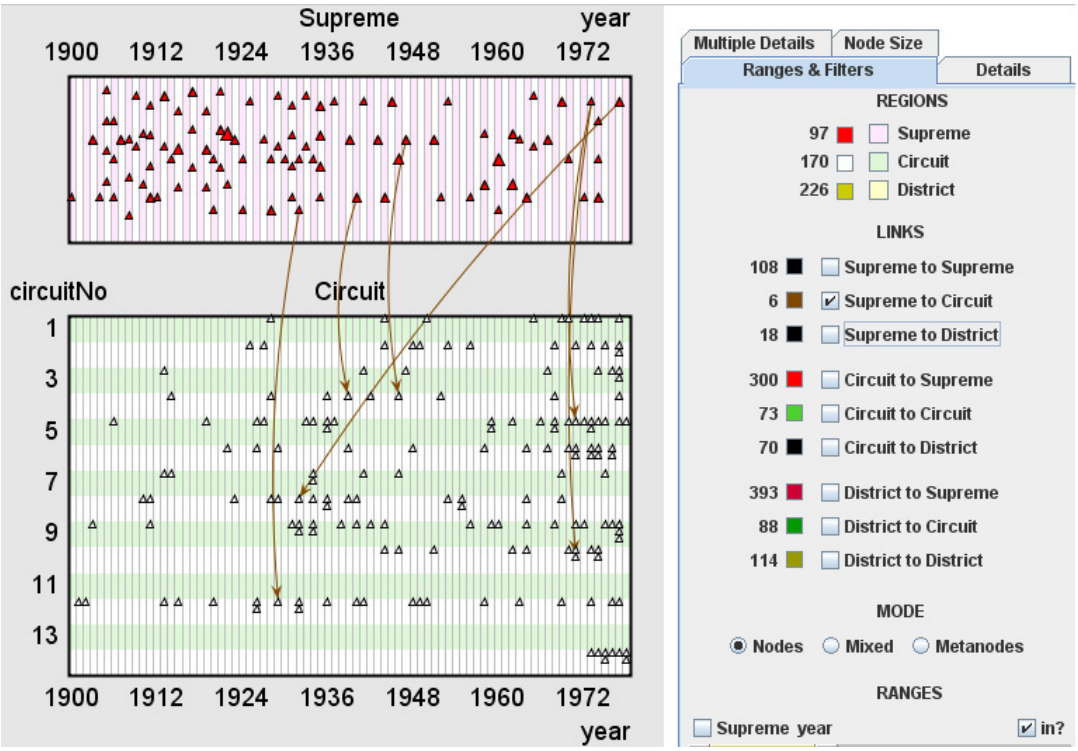


Figure 62 Supreme to Circuit citations.

After this view, the case study participant wanted to look at a combination of links. Hence, we enabled the Supreme to District and Circuit to District links (Figure 63). Since the colors of these last two were the same, the Circuit to District link color was chosen a different color (red if this page is printed in color, lighter gray if this page is printed in grayscale).

In this new view (Figure 63), the case study participants communicated that they were expecting that the Circuit Courts would become the source of authority since the Supreme Court was not assuming it. However, by looking at this display, he commented how strongly cases were citing the District Court cases and that their

previous hypothesis remained unsupported. He concluded that they were overlooking some interesting possibilities as the important case law was developed by District Courts at this period (1930s).

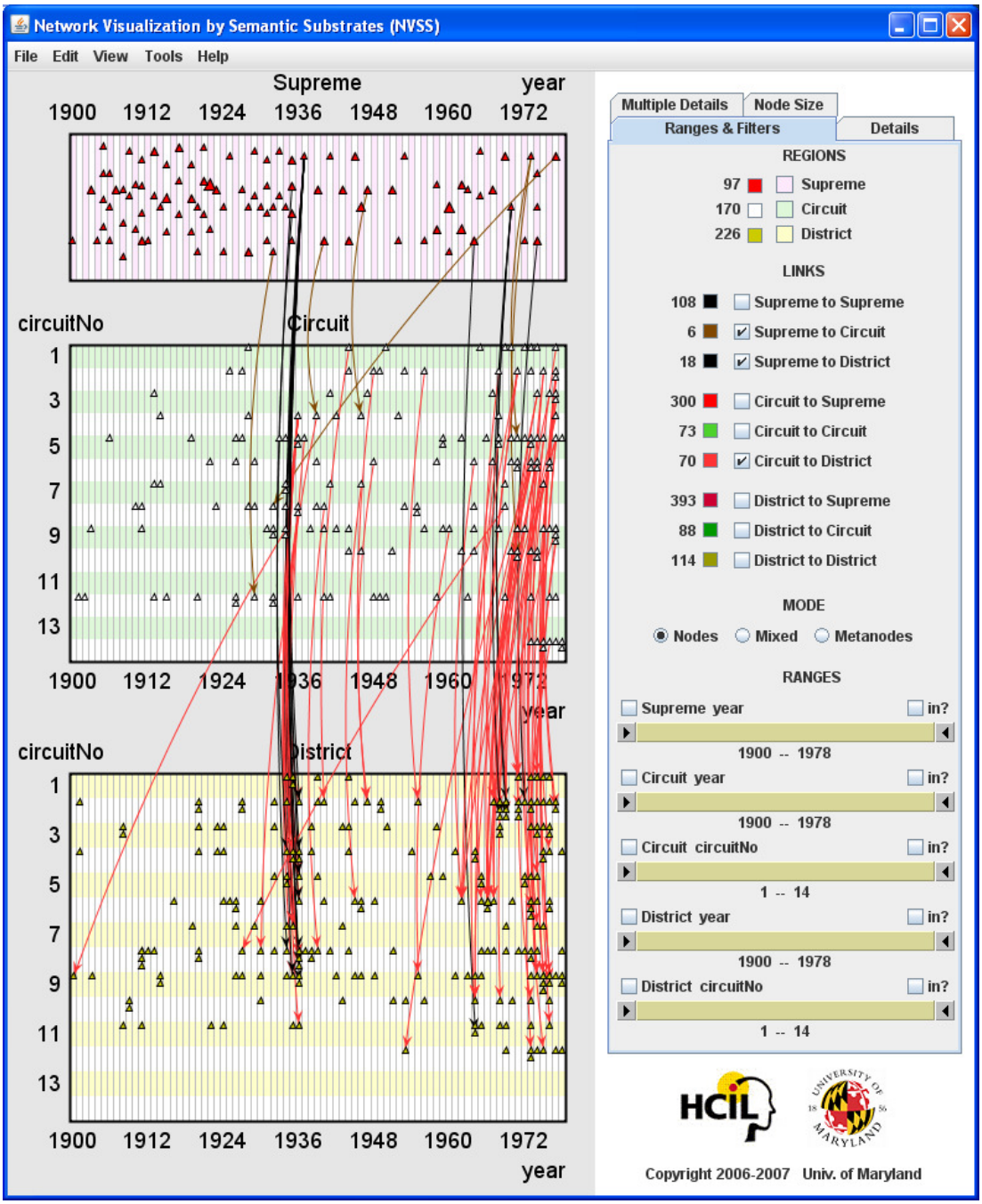


Figure 63 Added the Circuit to District citations to see how courts cite the District Court cases.

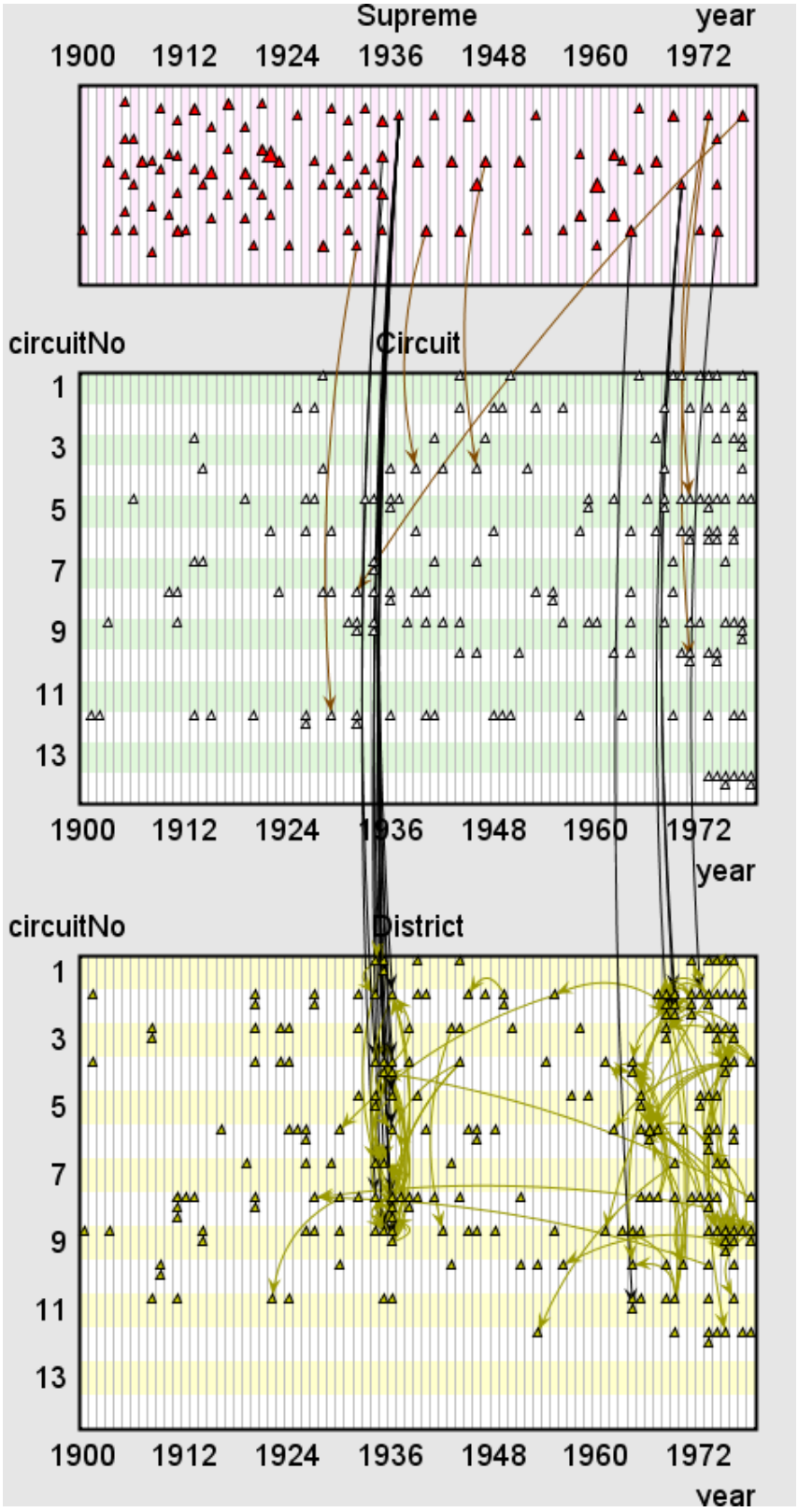


Figure 64 Added the District to District citations.

Next, the case study participant wanted to also see the District to District citations; hence, we enabled those (Figure 64). We also disabled the other links to clearly see only the District to District citations (Figure 65).

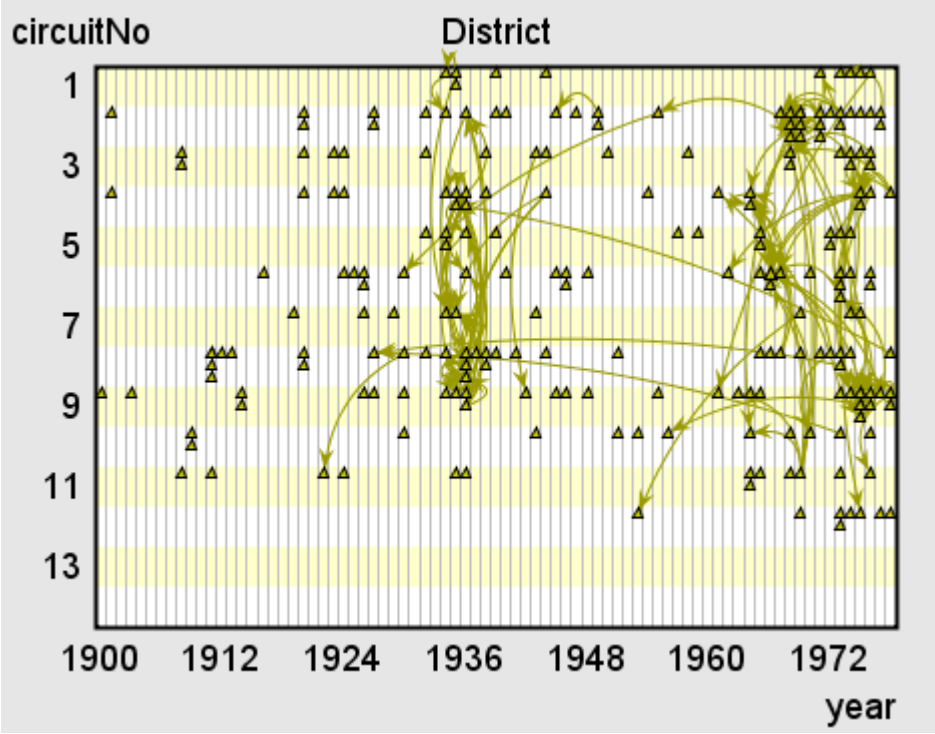


Figure 65 Looking at the District to District citations only.

On this view, the case study participant commented that there were a lot of intra-citations during the period of 1930s and concluded that the District Courts were referring to each other. He communicated to the other two team members that they might want to identify these cases and some numbers about these cases. The case study participant looked happy and enthusiastic. It seemed that they had found something interesting in the dataset. He continued to say that he had not notice this phenomenon until he opened this dataset the previous night to visualize it. He said that previously he was looking at only citations to the Supreme and Circuit levels. However, at the District level, cases are cited by other Circuit Courts. Furthermore,

the citations were not only from within the circuits but also from outside of the circuits, which seemed to further intensify the importance of this discovery. Citing outside of a circuit usually seems to be an interesting phenomenon.

The numbers of outgoing Supreme Court citations except the Supreme to District Court citations surprised the case study participant. Considering that there are 97 Supreme Court cases, he found the 108 Supreme to Supreme, the 6 Supreme to Circuit citations to be low. He suggested double-checking the accuracy of these numbers. The numbers of all outgoing Supreme Court citations (including Supreme to District) were checked and found to be correct.

At this point, the voice connection to the other team members was ended. We continued to look at the data together; and at the end he suggested to prepare screenshots and send them to him so he could circulate them to the other team members to have a more meaningful discussion.

5.2.2 Food-Web Case Study

Biologists study predator-prey networks, which are called food webs. The domain expert, Dr. Cynthia Parr, is a biologist and researcher associated with the Human-Computer Interaction Lab. There are several visualization tools she had been exposed to before NVSS. One of them is the Webs on the Web 3D ball and stick visualization created by Rich Williams, currentlyat Microsoft Research in England. Another one is a visualization that a former Ph.D. student Bongshin Lee created with her help, TreePlus. Finally, a third visualization she was exposed to used a standard force-directed layout. She used these visualizations primarily to understand local connections in food webs for a given species, to answer questions, such as: Who are

the predators and who are the prey? What are the typical chain lengths, what species are very densely connected, and what kinds of characteristics are seen for some of these animals in the tree? Were there any patterns in those characteristics?

She was interested in exploring a food web dataset (seven aquatic webs from Brose et al. (Brose 2005)) and we agreed on a substrate design to facilitate her understanding of the data. NVSS 2.0 was used to explore the data. She was the domain expert for this dataset, while the author of this dissertation was the tool expert. The results here were arrived after 5 sessions over 6 weeks, each lasting 45-60 minutes. In the first two sessions, we determined 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 feedback. Email communication with the domain expert helped discuss specific aspects of the data and its presentation.

Communication with the domain expert about the dataset led to the initial substrate. Later, as the tool expert, 3-4 iterations were needed to arrive at an initial substrate satisfying to the domain expert. These iterations were guided by the 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 2 iterations to arrive at the second substrate. This time, we used design-by-example. The first substrate was reused and modified to arrive at the second one, a much faster process. As in the Cite-It dataset, iterations resulting from minor adjustments, NVSS software updates, etc. are not counted.

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 66). The dataset combines results from seven studies of aquatic food webs. When we visualized the network by studies, we discovered that the nodes do not refer to the nodes in other studies. This might be because each study is self-contained to a certain place and time. Some of the available node attributes in this dataset 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 dataset).

The dataset consists of a total of 640 nodes and 1978 links. The missing values in the dataset were represented with negative values as NVSS did not have a capability to represent missing values (and it was not straightforward to add this feature). Although this choice was not ideal and the domain expert did not find the presentation intuitive, it provided a workable solution. To avoid misinterpretation of the display, this point was clearly communicated to the domain expert.

With the 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 $3.57e^{-18}$ to $9.46e^{-8}$ grams while the average length ranges from 0 to 0.1 meters

(most values are small ranging from $4e^{-6}$ to $9.8e^{-5}$ meters; the rest are 0, 0.0001, 0.0005, 0.005, and 0.01 meters; 0 may indicate that the length is immeasurably small). Since the range of these attributes is so small and hard to analyze, they were not used for placement for this first region. 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 the collaborator, we assumed 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, i.e., missing data.

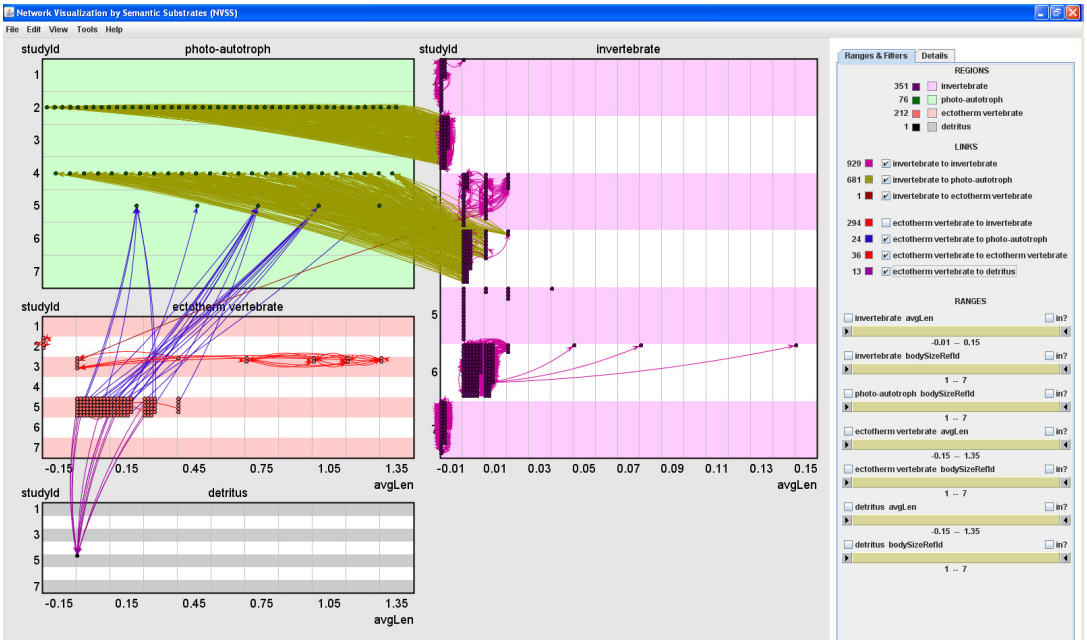


Figure 66 Using semantic substrates with food web datasets. Displaying data from seven studies with length (in meters) on the x-axis for all except photo-autotroph. Negative values indicate missing attribute values.

The domain expert reported that the most striking conclusion was that the seven datasets differ considerably in the metabolic categories of organisms they sampled, and hence the kinds of links that were possible. However, there are some patterns relating to size.

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*, an invertebrate in study 6 having avgLen 0.08 (the 2nd longest in terms of avgLen of all invertebrates), is prey of *Agabus bipustulatus* (Coleoptera), also an invertebrate in study 6 having avgLen 0.01 (one of many in the cell identified by the measurements). Invertebrates are also prey of ectotherm-vertebrates, such as *Daphnia rosea* (water flea) is prey of *Salmo trutta* (brown trout) (identified by one of the many 294 ectotherm vertebrate to invertebrate links, which are not in visualized in Figure 66). Only in study 5, invertebrates do not consume invertebrates (the domain expert reported back that "at least, their consumers are not in these datasets").

It appears that photo-autotrophs are the 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 (mostly by short and medium, some from long and non from the longest ectotherm vertebrates 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.

The missing predator-prey links are questionable. To determine the reason, further investigation to the data source and studies is needed.

To gain further understanding, a different substrate was used to visualize the same dataset (Figure 67). We hoped that this different point of view would help to attain new insights and understandings. By the suggestion of the domain expert, log transformations on the axis attribute values are used to have a better spread of the nodes. In this substrate, all regions except detritus, whose avgLen is specified as 0, 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. The y-axis attribute values and the number of bins are not the same although they are close. In the new substrate (Figure 67), the study number is not represented. Instead, the mass is represented. We hoped that this would enable the domain expert to see overall tendencies without distinguishing by study. Specifically, combining the data from all studies, visualizing general tendencies in terms of mass and length were facilitated. This provided evidence to support the earlier hypothesis that mass and length are usually proportionate to each other (the nodes (except the ones having missing data) in Figure 67 are usually located on the diagonal from the upper left to the lower right corner). Shorter and lighter photo-autotrophs are consumed only by 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 due to missing values in the data) of photo-autotrophs and detritus; while mostly the heavier and longer ectotherm-vertebrates eat others in their own metabolic category.

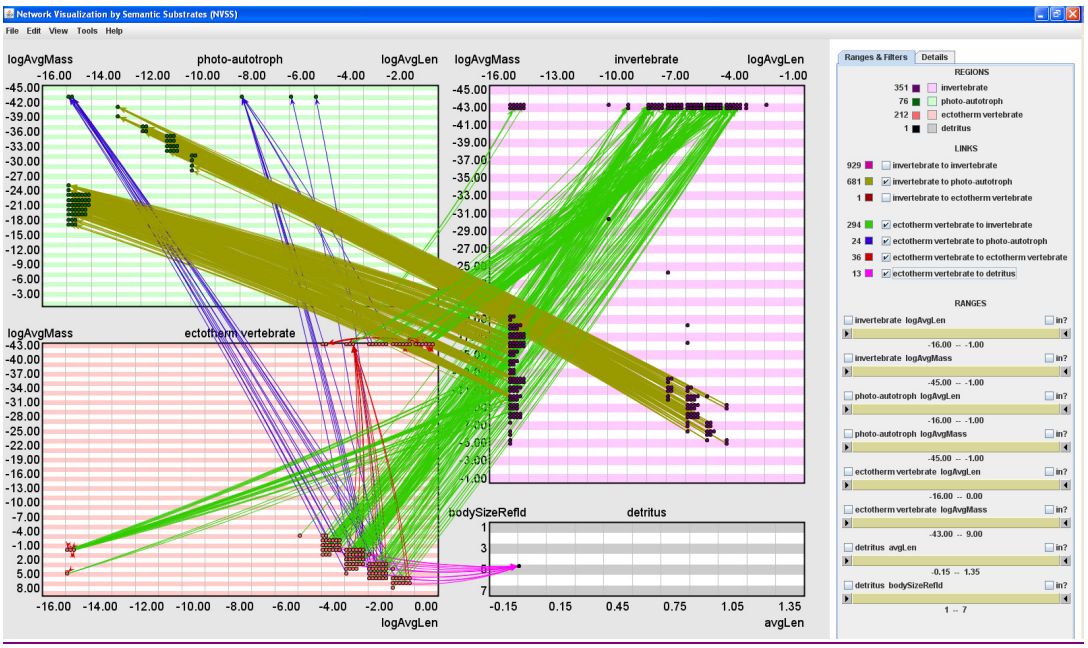


Figure 67 Using a different semantic substrate with the same food web dataset as in Figure 66. Displaying combined data from seven studies with log(length in meters) on the x-axis and log(mass in grams) on the y-axis. Missing/unknown mass is denoted by –43, while length is denoted by –15.

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. This is mostly perceived by looking at the distribution of the destination nodes connected by links originating from the bottom left of the ectotherm-vertebrate region. By using an incoming link filter on mass on the invertebrate region, one can clearly see that this is the case (Figure 68).

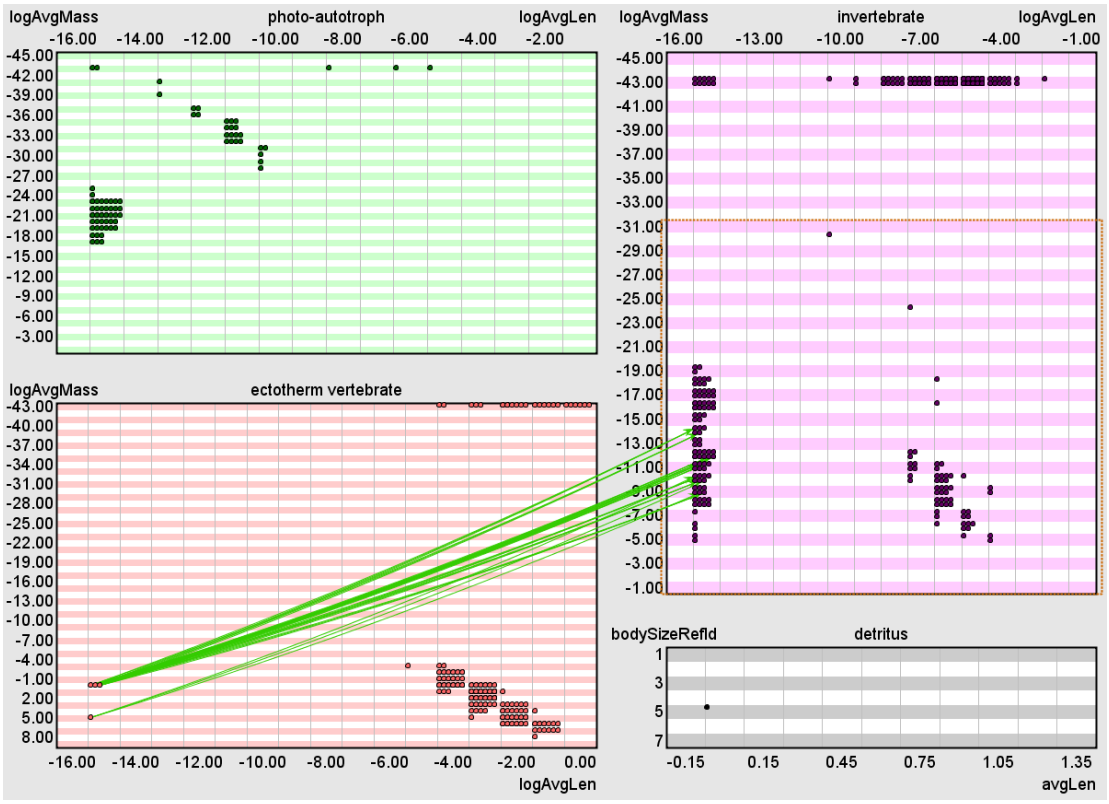


Figure 68 Using an incoming link filter on mass on the invertebrate region shows that among the known-mass invertebrates, the ones that are eaten are those that have a medium weight. In other words, the lightest and the heaviest invertebrates are not consumed.

From these two views on the food web dataset, at one glance, with a little bit of focus on each area on the display, almost all interactions within the dataset of seven studies are visible in terms of study and length, and then mass and length. Filters help focus on areas to reveal relationships more clearly.

The semantic substrates enabled the domain expert to understand her dataset better. She realized that the seven datasets the seven datasets differ considerably in the metabolic categories of organisms they sampled, and hence the kinds of links that were possible, a fact unknown to her before looking into the data in NVSS.

The highly skewed distributions and the missing values in the data presented challenges, but these features stood out clearly in the 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 (missing data can be identified easily by looking at the edges of the regions because the extreme values are used to represent missing attribute values). This case study also revealed a possible improvements and challenges. One possible improvement was the support to define and represent missing values. One challenge was that the directions of links were not clear when there were many of them drawn close to each other (e.g., see Figure 66, the directions of the links connected to the photo-autotrophs are not clear).

The 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.

5.2.3 TobIG Case Study

This section will summarize the 7th session of the TobIG case study and will contain examples where the node aggregation feature was used to explore a document citation dataset called TobIG (For complete details on the 7th session, see Appendix D). The author of this dissertation designed the substrates in consultation with the collaborators. The main collaborators are Prof. Noshir Contractor at Northwestern University and Assistant Prof. Steve Harper at James Madison University. The earlier sessions and communications were mostly conducted by both of the collaborators while the later sessions were conducted with Prof. Noshir Contractor. Noshir

Contractor has experience with various network visualization tools. He reported that they (he and his team) have used NetDraw, Pajek, as well as a suite of Java and PREFUSE visualization tools that they have developed in their lab. These tools were used to help distinguish different types of links (by color and/or thickness) or different types of nodes (by color, size or shape). They used these to help interpret what their analytics were telling them about key roles played by individual nodes in the network as well as overall global properties of the network (density, centralization, for instance).

The dataset name TobIG stands for Tobacco Behavioral Informatics Grid and contains Tobacco researchers, the documents they wrote, and the keywords of these documents. Prof. Noshir Contractor is not the actual end user for this dataset; however, he interacts with the end users and domain experts and has a good knowledge of the types of tasks and goals. In addition, he has personal contacts with at least a few of the authors in the dataset; and therefore, can interpret some of the results well. Prof. Noshir Contractor was treated as if he was a domain expert in this case study. Further validation of results with end users and domain experts is appropriate and recommended.

We started exploring the data in NVSS 2.0 and in later sessions, NVSS 3.0 was used. The collaborators were interested in finding patterns and relationships to answer several questions including what topics emerged over time in this field; when and how they emerged, which authors wrote on Tobacco research and in what other topics these authors are knowledgeable.

In this session, the feedback and experiences from earlier sessions was used to design the substrate. In addition, several exploration paths that could be interesting to the case study participant were found prior to the session. The case study participant was allowed to lead the path of exploration during the session. The paths of exploration found earlier were presented either when he was not leading or when he asked questions that could be answered by illustrating these exploration paths. At those instances, these exploration paths were presented and the reactions of the case study participants were noted.

Nodes represent authors, documents, and keywords. There are 29 authors, 1,700 documents, and 2,567 keywords, totaling to 4,296 nodes. Links in the dataset are directed and represent the following relationships according to their source and destination: Authors write documents, documents cite documents, and documents use keywords. The data had to be pre-processed to convert it to the format that NVSS expects. The original data contains different set of attributes for authors, documents, and keywords. However, NVSS assumes that all nodes have the same set of attributes. For this reason, a pre-processing step generated common attribute types for these three types of nodes. A new attribute was introduced called *type* with possible values of "Author," "Document," and "Keyword." The *name* attribute is the name for authors, the title for documents, and the keyword itself for keywords. The *year* attribute is the first year of publication for authors, the year of publication for documents, and the year of the first document in which it appeared for keywords. This process of combining attributes of different types of nodes to a single attribute is defined as *the unification of attributes.* Another way to accommodate attributes that

are specific to each type of node is to duplicate attributes of one type of node to the other types of nodes. This process is defined *the duplication of attributes*. (These pre-processing steps can be eliminated by having NVSS support nodes having multiple sets of attributes, a major programming effort that could not be accommodated at the time and left as future work.)

The following attributes were additional attributes that were specific to the type of node:

The attribute *CR* is only applicable to authors; it stands for "Citations Received" and it actually represents the author's H-score (an index to characterize the scientific output of a researcher, where the researcher has h papers that are cited h or more times, (Hirsch 2005)). The attribute *LCS* is only applicable to documents; it stands for "Local Citation Score" and it represents the number of times the document was cited by other documents in this dataset. The attribute *Count* is only applicable to keywords and it represents how many documents used this keyword in this dataset.

At the time of the 7th session of this case study, these attributes were unified to the attribute *CR_LCS_Count.* Later, this was found confusing and these attributes were duplicated instead.

The semantic substrate in Figure 69 was not shown to the case study participant (this substrate is designed by duplicating *CR*, *LCS*, and *Count*). A substrate that was equivalent to this substrate (the only difference was that the attributes *CR*, *LCS*, and *Count* were unified to *CR_LCS_Count* at that time instead of being duplicated.) was designed as an initial substrate. It partially revealed the distribution of the data, which helped design the final substrate. Only the final

substrate was presented to the case study participant (to save time; the case study participant had a busy schedule). The substrate (Figure 69) has three regions, each using a value of the *type* attribute. The location of the regions from top to bottom is in line with the directionality of the links: authors write documents and documents use keywords. *Year* is used along the x-axis of all regions consistently. *CR, LCS,* and *Count* are used along the y-axes with a consistent bin height, 5 for authors and 10 for documents and keywords. Most of the nodes seem to have lower values and tend to overlap in small places. This revealed the need for an unevenly spaced binning strategy on the y-axis values to spread the nodes more evenly on the display. The dataset is dense in terms of the links as there are 1,770 author-to-document, 4,966 document-to-document, and 9,649 document-to-keyword links (total = 16,385).

To improve upon the node overlaps in lower values of *LCS* and *Count* (Figure 69), an unevenly-spaced binning strategy was used for the y-axis of all regions. Consequently, the substrate in Figure 70 was attained (for strategies to create uneven distributions see Aris et al. (Aris 2005)). To determine the binning, a trial-and-error method was used (by the author of this dissertation) and the boundary values that seemed to result in better distributions were chosen. When there was more than one possible binning that had a similar spread, personally intuitive choices were made hoping these choices would be close to the choices that the case study participant would prefer. For example, authors with H-score between 5 and 9 were grouped together causing the next group to start from 10. During the session, no indication to a different preferred binning was observed from the case study participant.

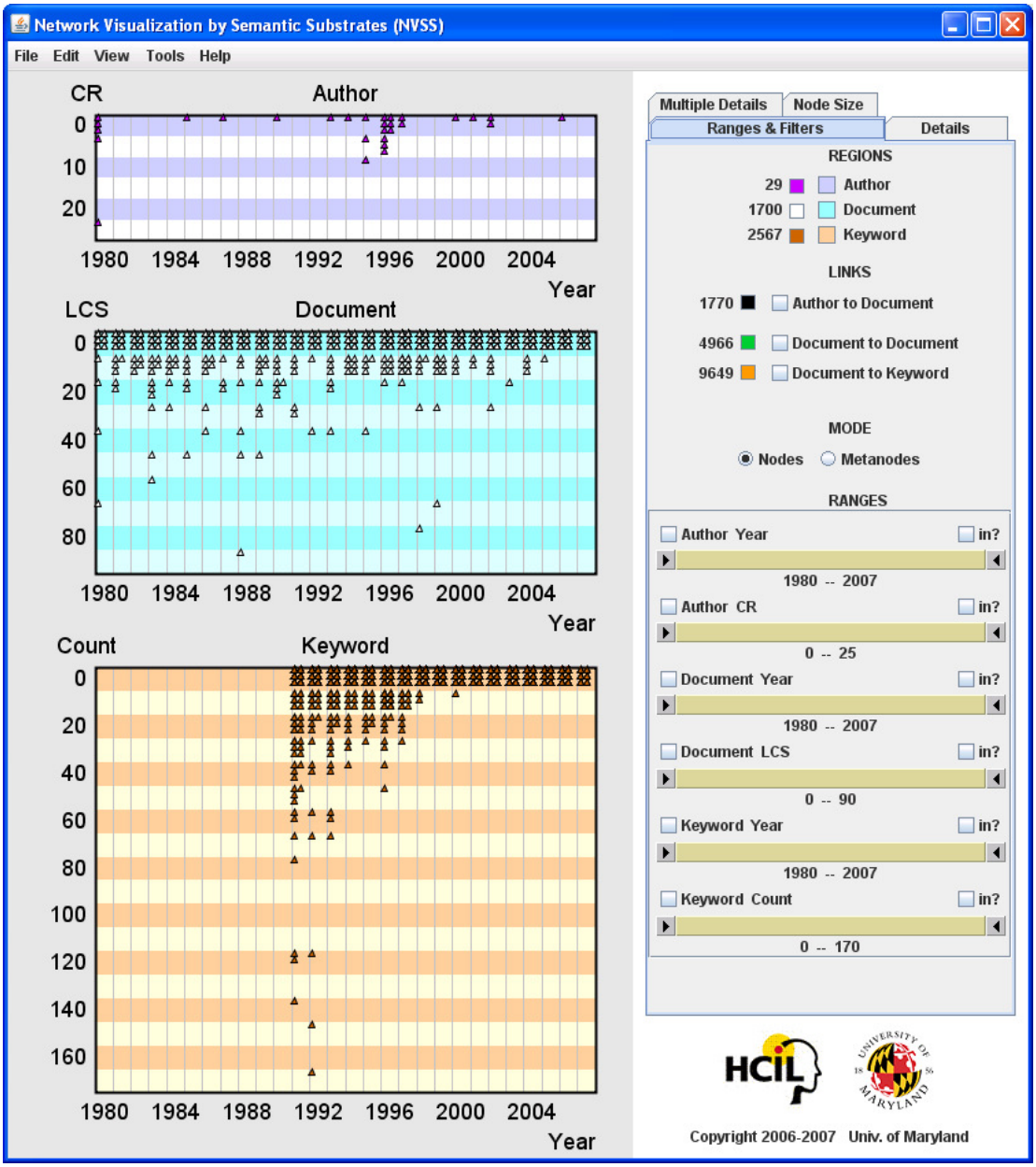


Figure 69 An initial semantic substrate is applied to the TobIG dataset, where nodes represent authors, documents, or keywords, and links represent "Author writes Document", "Document cites Document", or "Document uses Keyword". Nodes are grouped into regions using the type attribute with "Author", "Document", and "Keyword" values while they are placed using Year along the x-axis and CR, LCS, and Count along the y-axes.

There were still cells that have more nodes than the available space in Figure 70 (e.g., cells where keywords have *Count* = 1); however, the display is much improved when compared to Figure 69. The height of the author region is minimal to

allow the visibility of all labels on the y-axis in this version of NVSS (in later versions, the sensitivity to available space was improved and the author region could be made shorter in terms of height and still view all the labels). Since there are more than 4,296 nodes, it is very hard (if not impossible) to design a substrate to avoid node overlap completely.

The size of metanodes in Figure 71 represents the sum of the *CR_LCS_Count* attribute values for the nodes they represent. The case study participant asked whether it represents the number of nodes and later suggested that this would be a useful feature. This point was noted and this feature was implemented later. The use of the unified attribute *CR_LCS_Count* seemed to slow down the case study participant when interpreting the data initially.

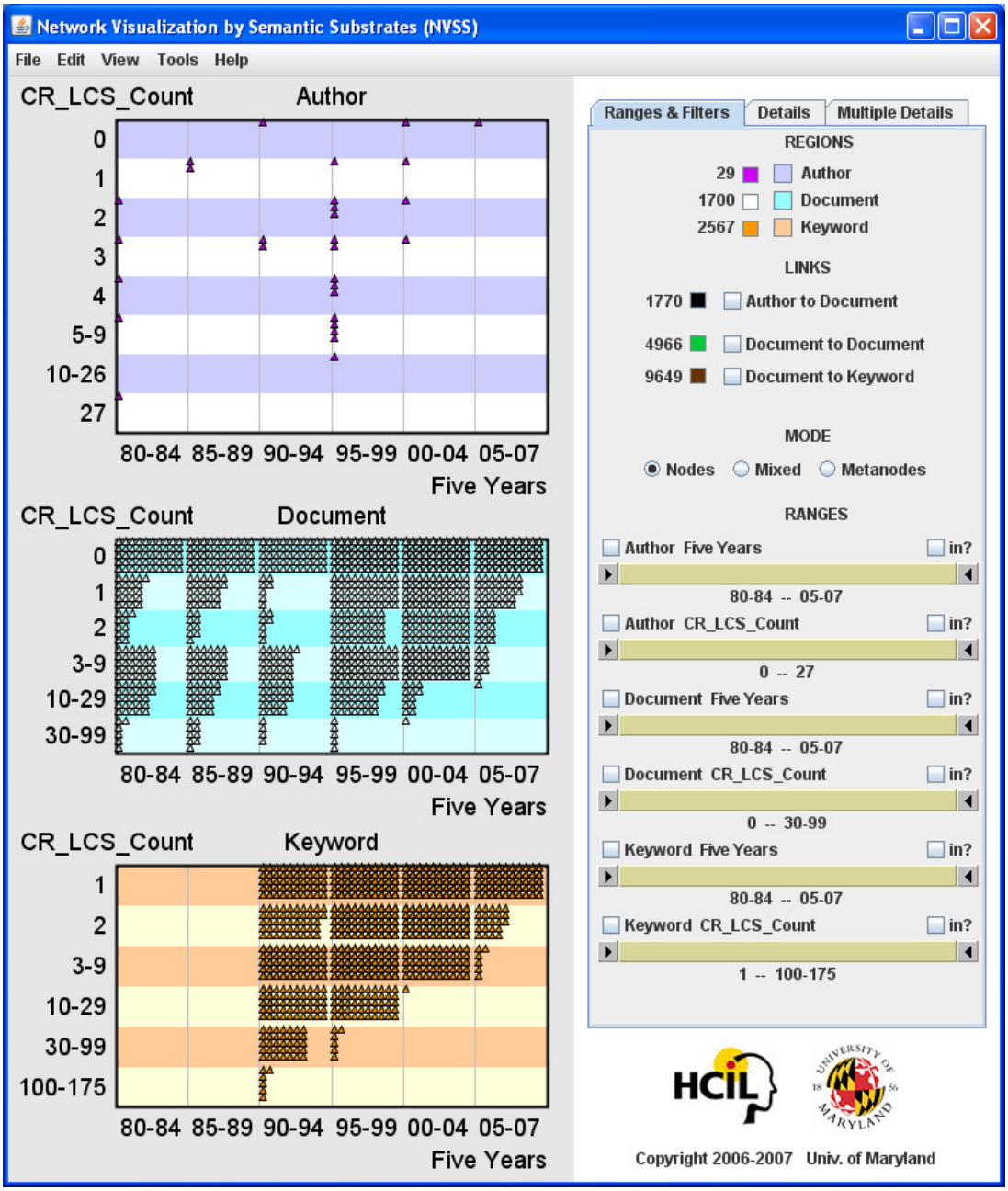


Figure 70 A different substrate is applied to the data in Figure 69 upon seeing node overlap in lower y-values. Year on the x-axis consistently binned into 5-year periods, while a custom binning is applied for CR, LCS, and Count on the y-axes different for each region.

The filters on the Keyword region were applied to show the usage of the highly used keyword during 1990-1994. When it was pointed out that not all documents were using the highly used keywords, the case study participant was very interested and found this intriguing.

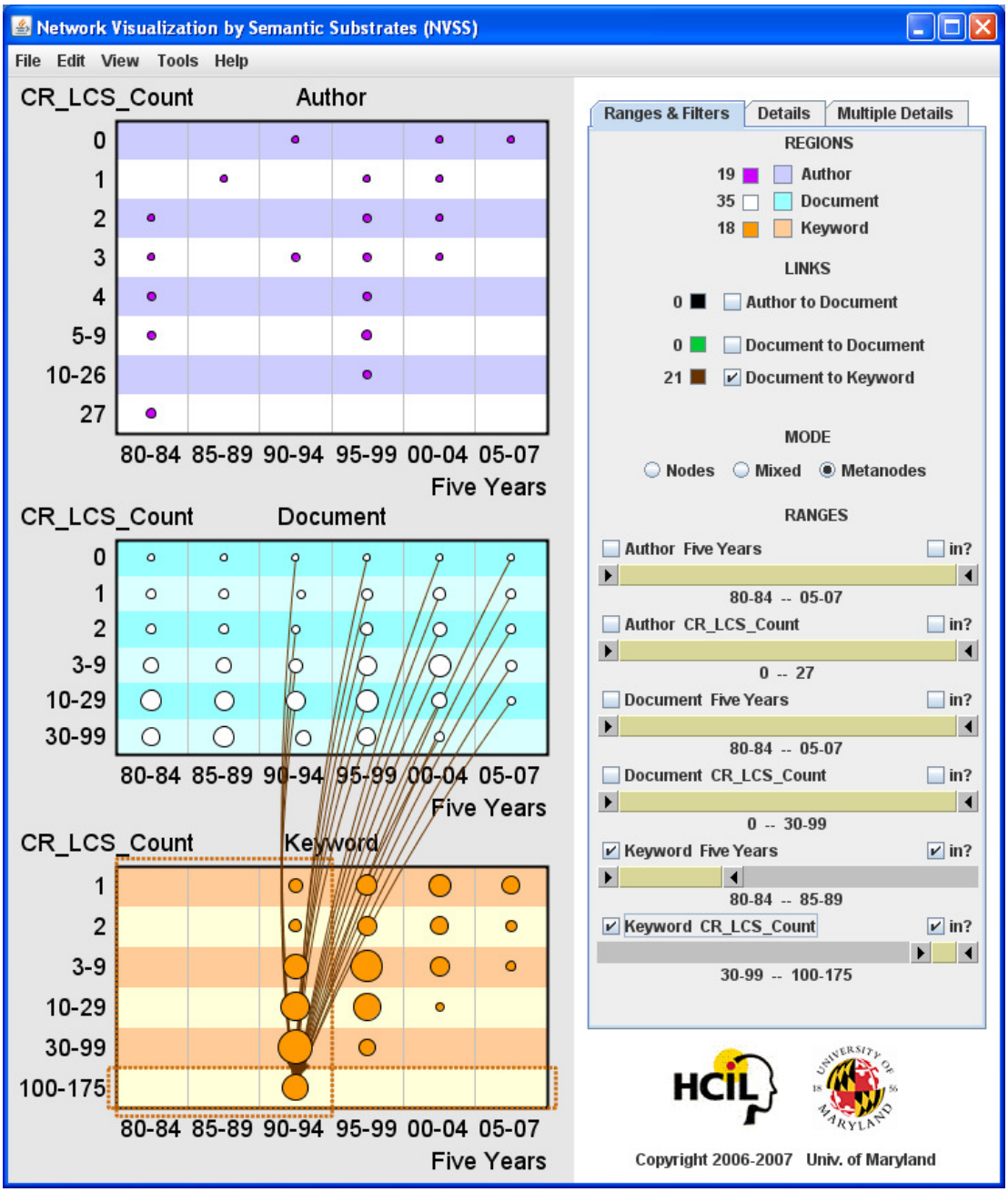


Figure 71 Among the documents in 90-94, the ones that are cited once and 30-99 times did not use the mostly used keywords in the same period (90-94).

Upon the request of the case study participant, we looked at the highly used keywords (Figure 72). He seemed to find all keywords very relevant to tobacco research. However, he wanted to look further to determine whether the lack of use of

the highly used keywords by the documents in the same period was an interesting fact.

Ranges & Filters | Details | Multiple Details

ID	NAME	TYPE	YEAR
5382	tobacco	Keyword	1992
5264	study	Keyword	1992
3423	cigarette	Keyword	1991
5190	smoking	Keyword	1991
4559	nicotine	Keyword	1991
5189	smoker	Keyword	1992

Figure 72 Details for highly used keywords in 1990-1994.

When we looked at the documents (Figure 73) and compared their years with the years of the keywords, the fact became less interesting. However, the case study participant appreciated that NVSS enabled him (or us) to investigate this.

Ranges & Filters | Details | Multiple Details

TYPE	YEAR	FIVE_YEARS	CR_LCS_C
Document	1991	2	32
Document	1991	2	37
Document	1992	2	41
Document	1993	2	40

Figure 73 The most cited documents of 1990-1994.

This discovery led the case study participants to possible hypotheses. He said it could be that two different groups were in this dataset, who works on different topics, where one group uses the highly used keywords and the other group on other keywords. Another reason could be that there was not enough time for the most cited documents to use the newly introduced keywords.

The case study participant found it useful to explore the relationship between documents and keywords and asked whether NVSS can help explore other types of relationships in this dataset. He specifically asked whether we could involve the

authors. Consequently, we switched our attention to the author to document links (Figure 74). We defined *top documents* in this dataset to be the documents that were cited 10 times or more. First, we looked at the period 1980-1989 (Figure 74). In this view, only one group of authors are writing top documents. Clicking on the metanode and looking in multiple details, we discovered that this group consisted of only one author called "Steve Hecht" (Figure 75).

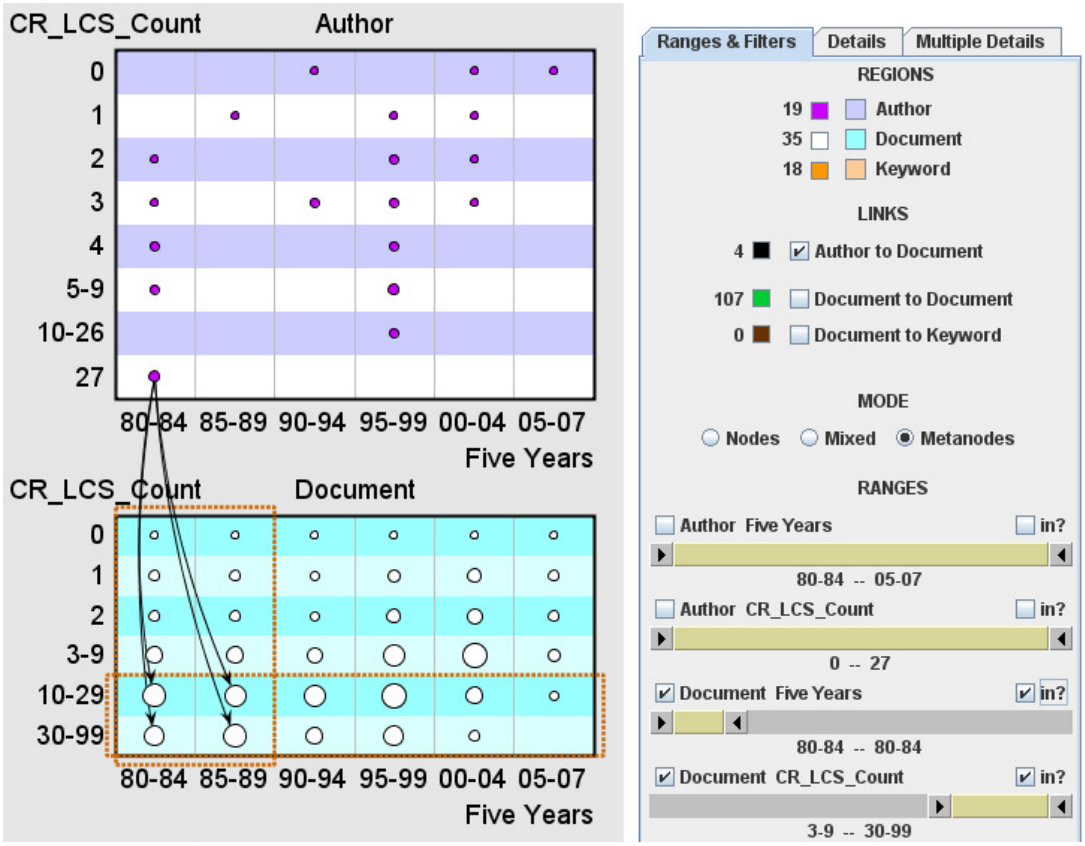


Figure 74 Showing the relationship between authors and top cited documents (documents that are cited 10 times or more).

Figure 75 The author in Figure 74 who has H-score = 27.

Advancing the time period to 90-94 led to another group of authors to be added (Figure 76), namely a group of authors from the 5-9 range (CR). Clicking on the metanode, we saw that it was only one author, “Ashley David” (Figure 77).

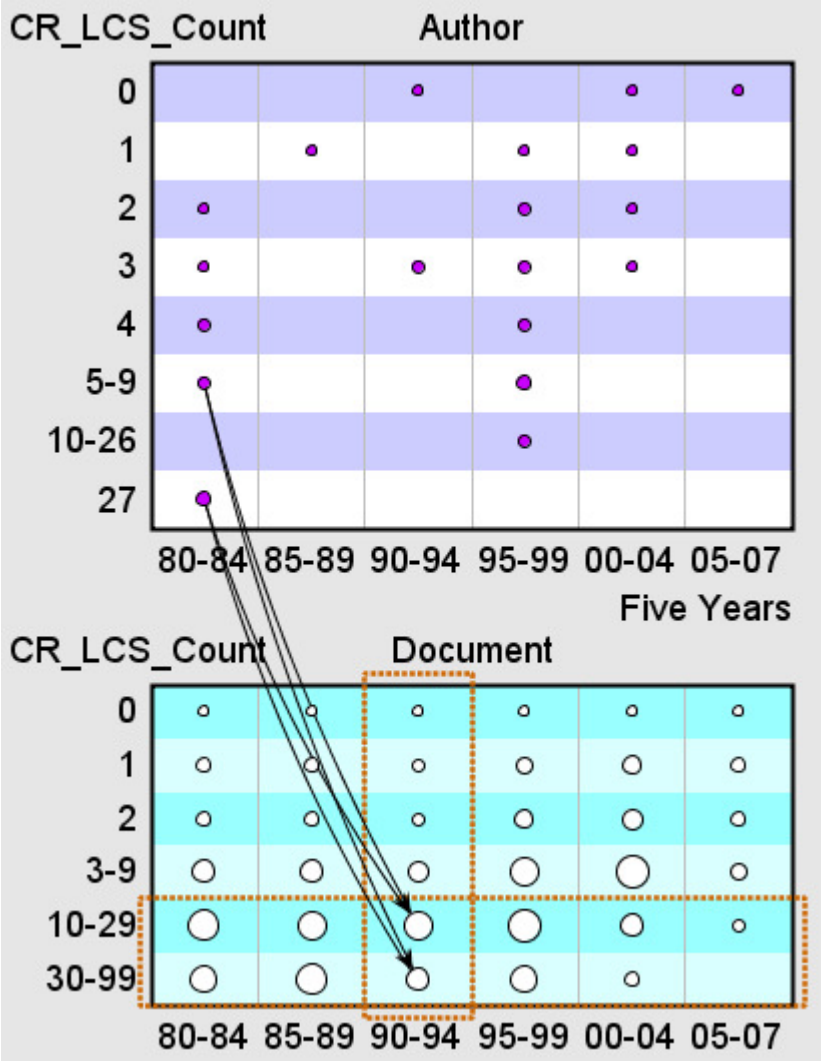


Figure 76 Looking at the period 1990-1994 shows authors writing top documents.

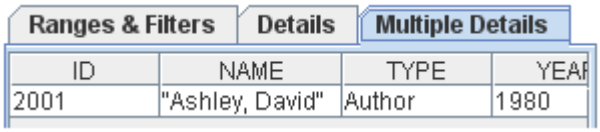


Figure 77 The author in Figure 76 who has H-score 5-9.

Advancing the period to 95-99, we saw a lot of other groups of authors join (Figure 78). At this point, the case study participant asked how many authors there were. Since NVSS didn't have the feature to show this directly, each metanode in the new 95-99 period was clicked (in the Author region that have a link outgoing) to see the list of the authors quickly (Figure 79, Figure 80, Figure 81, and Figure 82). The case study participant was very interested. He actually knew Fran and Scott personally (Figure 79). When I showed the next period, where they did not write top documents any more, he found this reasonable because he knew that Fran and Scott changed their positions and probably were not writing as much in the field for this reason (Figure 83).

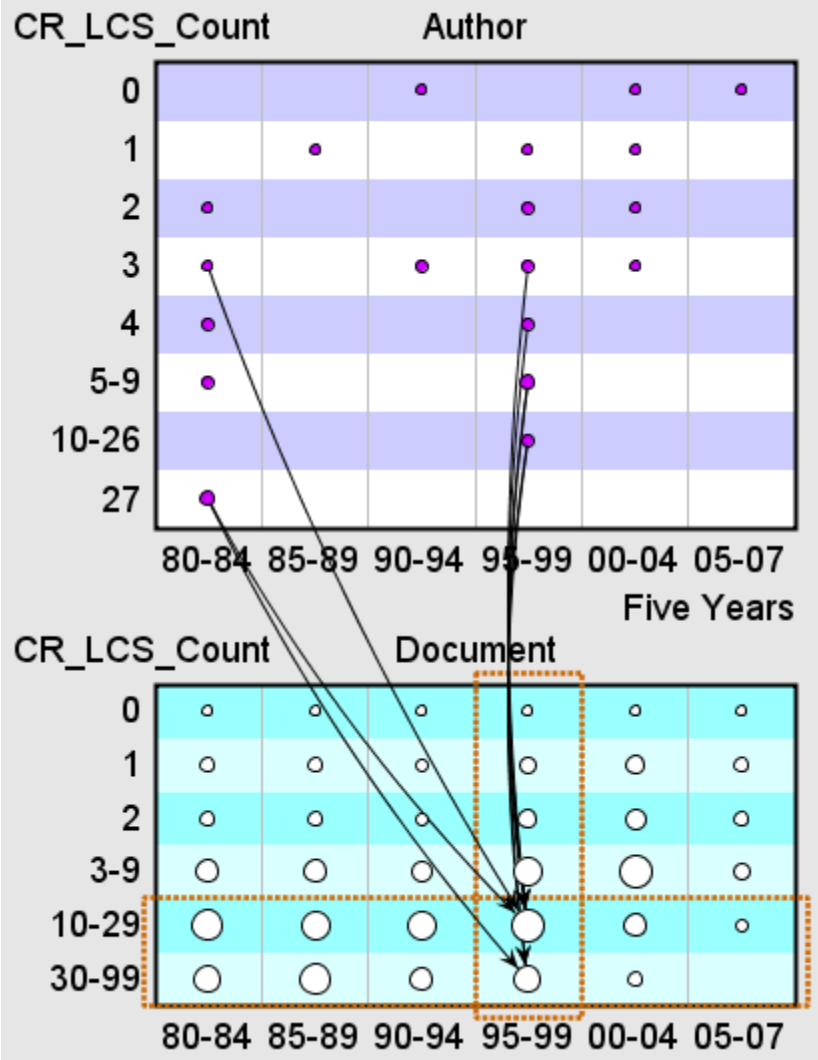


Figure 78 Looking at the period 1995-1999 shows authors writing top documents. There are many new categories, many from the same period 95-99.

Ranges & Filters | Details | Multiple Details

ID	NAME	TYPE	YE
2030	"Stillman, Fran"	Author	1996
2019	"Leischow, Scott"	Author	1996

Figure 79 The authors in Figure 78 who have H-score 3.

Ranges & Filters | Details | Multiple Details

ID	NAME	TYPE	YEAR
2003	"Biener, Lois"	Author	1997
2009	"Djordjevic, Mirjana"	Author	1996
2031	"Tomar, Scott"	Author	1996

Figure 80 The authors in Figure 78 who have H-score 4.

Ranges & Filters | Details | Multiple Details

ID	NAME	TYPE	YE
2013	"Giovino, Gary"	Author	1996
2029	"Shields, Peter"	Author	1995
2015	"Hatsukami, Dorothy"	Author	1996
2011	"Eissenberg, Tom"	Author	1996

Figure 81 The authors in Figure 78 who have H-score 5-9.

Ranges & Filters | Details | Multiple Details

ID	NAME	TYPE	YE
2025	"Benowitz, Neal"	Author	1995

Figure 82 The authors in Figure 78 who have H-score 10-29.

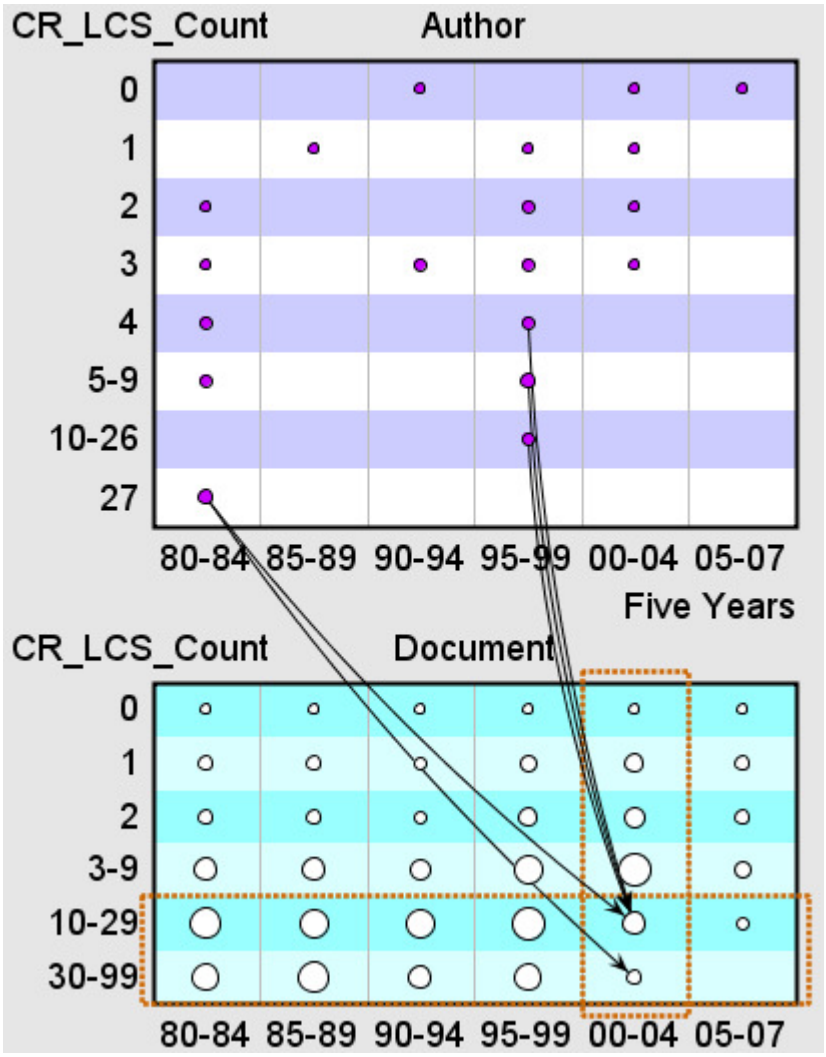


Figure 83 Looking at the period 2000-2004 shows authors writing top documents. The authors having H-score and writing top documents in the previous period vanished.

The case study participant was cautioned that the measure of "top-documents" might be naturally getting stricter because there is less time in this period (2000-2004) for other documents to cite documents produced in this period (Figure 83). To check this assumption, we switched to the node view, which showed that there were only 16 author-to-document links (Figure 84).

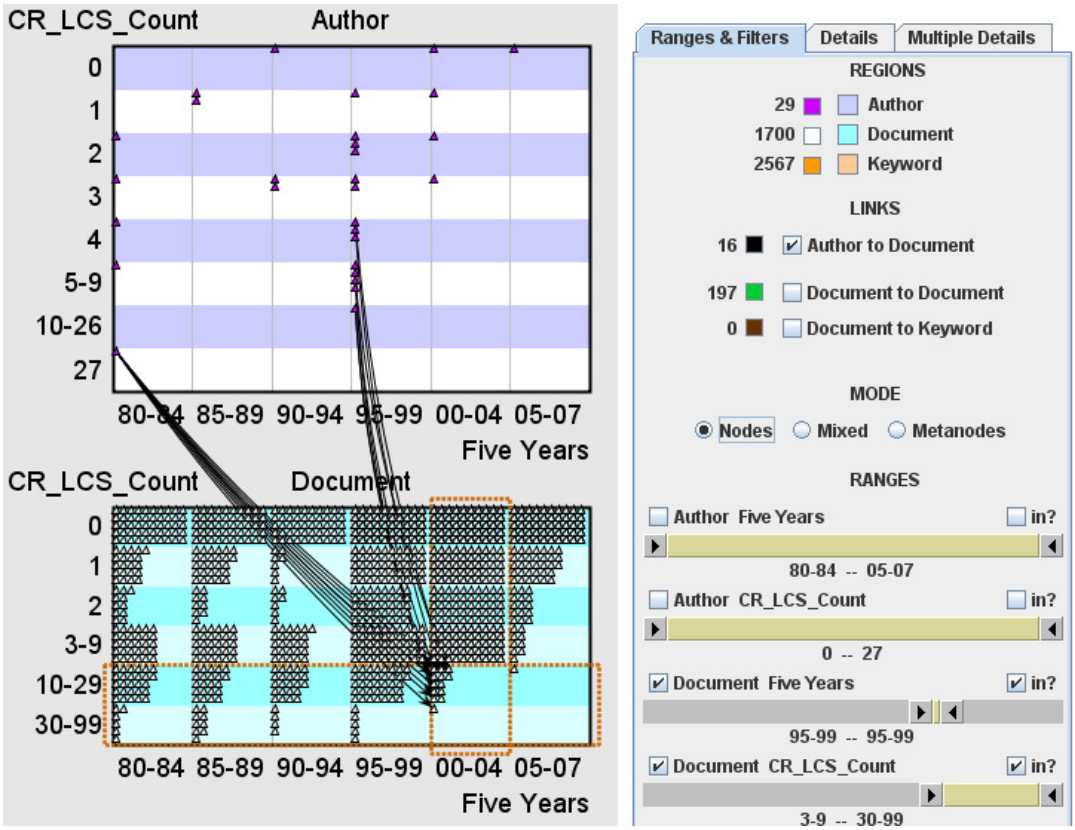


Figure 84 Switched to the node mode to see the number of actual author-to-document links in 2000-2004. There are only 16 such links, which users can see on the right hand side to the left of "Author to Document" checkbox.

Comparing the number of links (16) in the period 2000-2004 (Figure 83) with the links in the earlier period 1995-1999 (Figure 85) provided some evidence for the reasoning that the measure for "top-document" in 2000-2004 might indeed be getting stricter, thus revealing a useful reason to use the nodes mode, which is: to check

assumptions and either correct them or confirm their validity). Furthermore, we also quickly looked at the earlier period, which is 1990-1994, which had 27 links. We thought that that was ok.

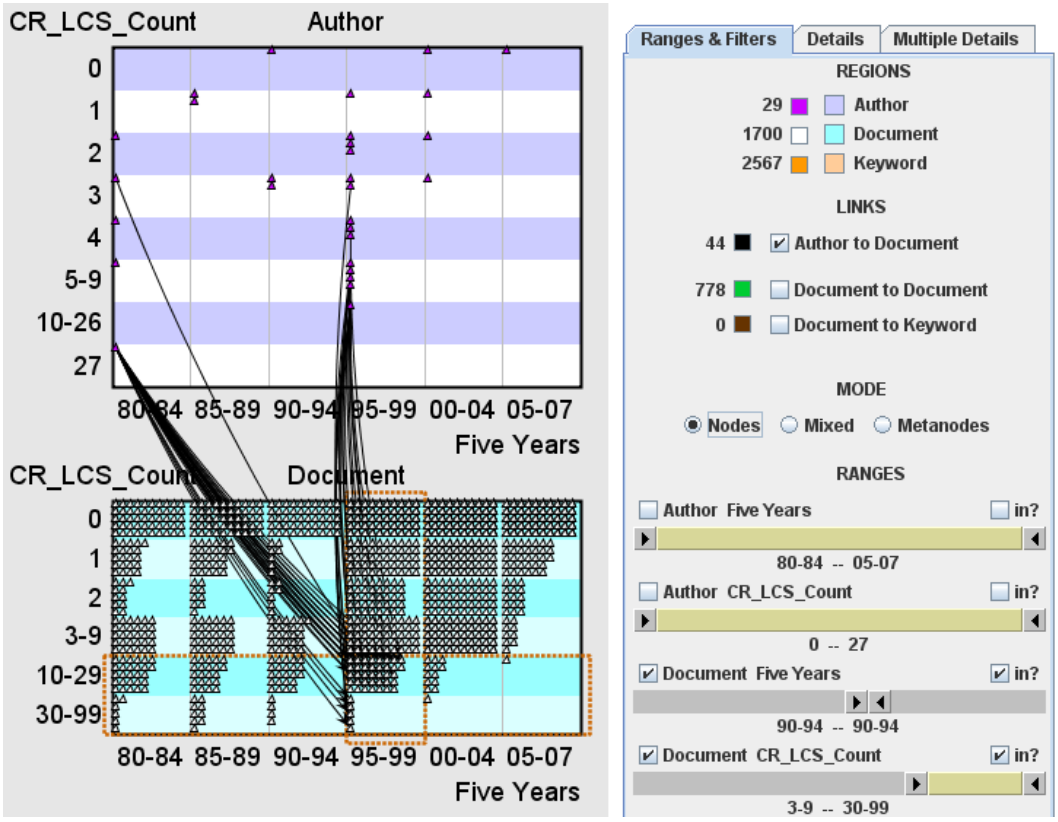


Figure 85 In the nodes mode, shifting to the earlier period to see how many actual author-to-document links there are in 1995-1999. There are only 44 such links, which is significantly higher than 16 links in 2000-2004.

Switching to the nodes mode momentarily also revealed that there was only one document in the 2005-2007 period. In fact, there was only one incoming link in that period (Figure 86). Then, we looked at a new view that displayed links from "Steve Hecht", the author with H-score=27 (Figure 87). In this view, we saw that Steve Hecht, the most prolific author of the dataset, did not write only top-documents

but all kinds of documents (except the last category, which was insignificant because it contains only one document as illustrated in Figure 86).

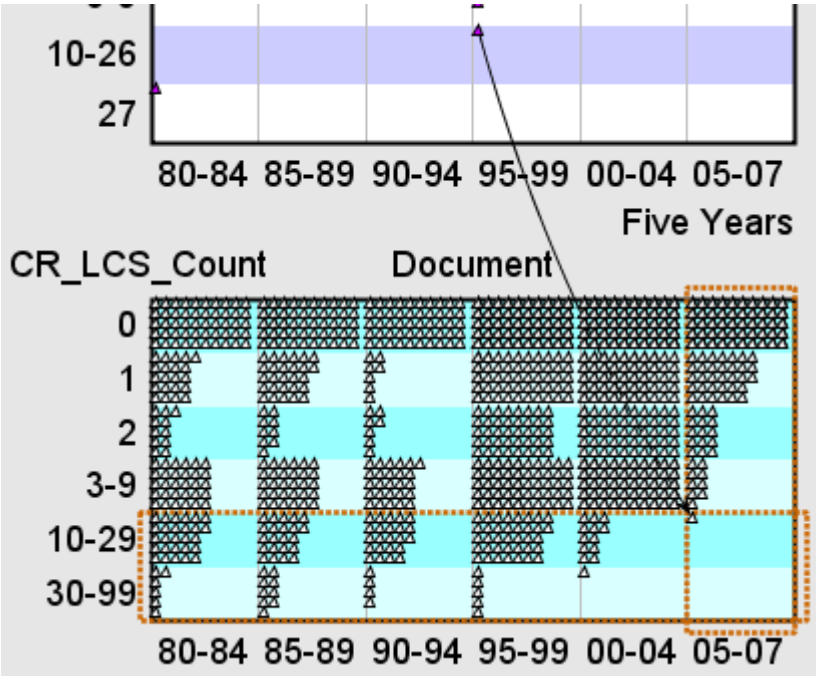


Figure 86 There is only one incoming link into the 2005-2007 period for documents.

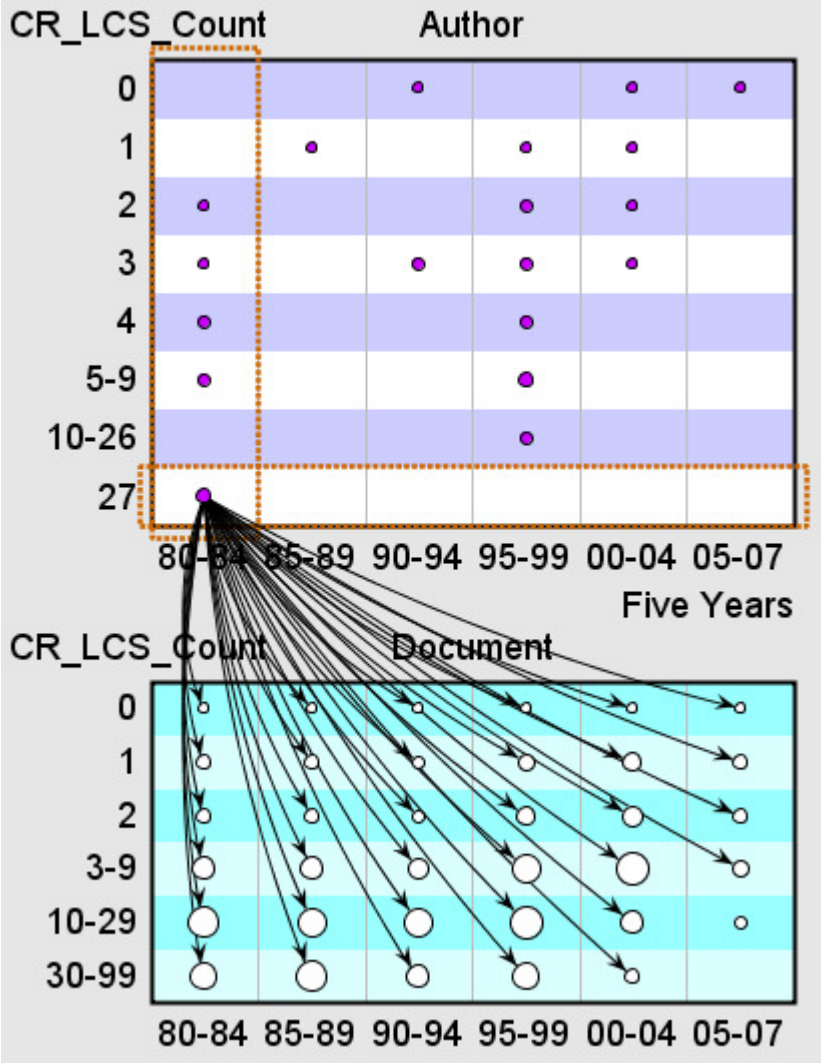


Figure 87 Showing the categories of documents that Steve Hecht wrote.

The case study participant was pleased by the exploration process that NVSS enabled on this dataset. He said "This is wonderful" and asked what we could do next. He said that we could have insights by looking at datasets like this in NVSS. He also said "I like this." He thought that NVSS was in a stage that could be used to explore datasets with domain experts and was enthusiastic to get in contact with potential experts that could benefit from this type of exploration.

The results of this case study showed some validity to the utility of the exploration process with NVSS as the case study participant has used several other

network visualization tools. Still, he found the approach in NVSS novel, useful, interesting, and worthwhile. He also suggested further directions to analyze various datasets. Some of these details can be found in the detailed case study notes. This case study also resulted in feedback for NVSS features. One of the motivators for the node aggregation feature was due to earlier exploration in this case study. Some of the future work ideas were actually suggested by the case study participant (e.g., cascaded type of exploration in section 7.1.4). Several features were suggested by the situations we encountered (e.g., multiple node types as address in the future work chapter, section 7.1.5).

5.3 Short-Term Case Studies

The following subsections provide two short-term case studies. These case studies have lasted less than 4 sessions.

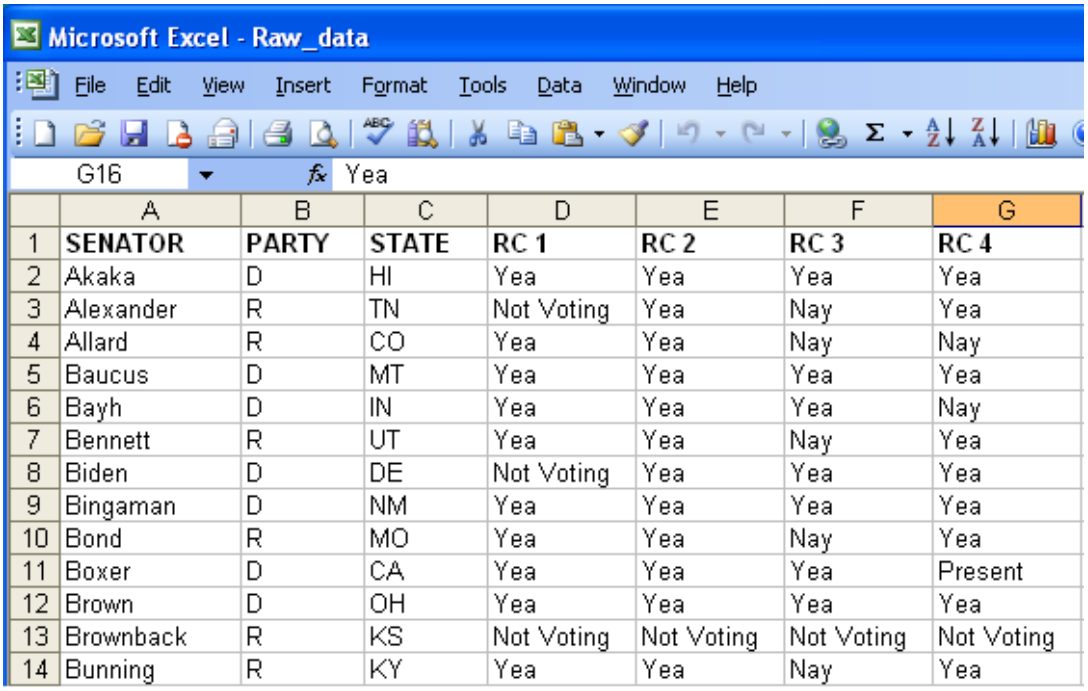
5.3.1 SenateVotes Case Study

This case study remained incomplete because the collaborator discontinued interest in exploring this dataset further. However, this is an interesting case study due to the familiar nature of its dataset and meaningful layout formed in NVSS.

In this case study, the domain expert is Chris Wilson. He was interested in exploring shared votes among the U.S. Senators in NVSS. NVSS 2.0 was used to explore this dataset. Preprocessing steps of the data was needed to input into NVSS and the domain expert helped with this step in a major way.

Chris Wilson is with U.S. News & World Report in Washington, DC. He was interested to see interesting patterns in the dataset. The SenateVotes dataset is based

on 98 U.S. senators (Figure 88) voting on 247 issues between 8 January 2007 and 13 July 2007 (Figure 89).



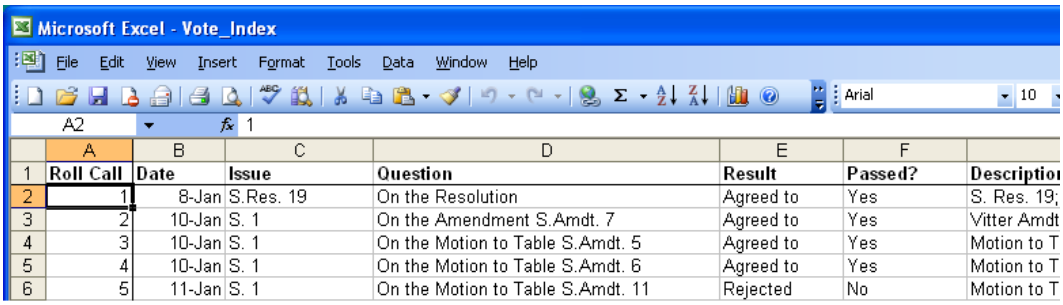
Microsoft Excel - Raw_data

File Edit View Insert Format Tools Data Window Help

G16 fx Yea

	A	B	C	D	E	F	G
1	SENATOR	PARTY	STATE	RC 1	RC 2	RC 3	RC 4
2	Akaka	D	HI	Yea	Yea	Yea	Yea
3	Alexander	R	TN	Not Voting	Yea	Nay	Yea
4	Allard	R	CO	Yea	Yea	Nay	Nay
5	Baucus	D	MT	Yea	Yea	Yea	Yea
6	Bayh	D	IN	Yea	Yea	Yea	Nay
7	Bennett	R	UT	Yea	Yea	Nay	Yea
8	Biden	D	DE	Not Voting	Yea	Yea	Yea
9	Bingaman	D	NM	Yea	Yea	Yea	Yea
10	Bond	R	MO	Yea	Yea	Nay	Yea
11	Boxer	D	CA	Yea	Yea	Yea	Present
12	Brown	D	OH	Yea	Yea	Yea	Yea
13	Brownback	R	KS	Not Voting	Not Voting	Not Voting	Not Voting
14	Bunning	R	KY	Yea	Yea	Nay	Yea

Figure 88 Ninety eight (98) U.S. Senators voting on 247 issues. RC stands for Roll Call. D stands for Democrats and R stands for Republicans.



Microsoft Excel - Vote_Index

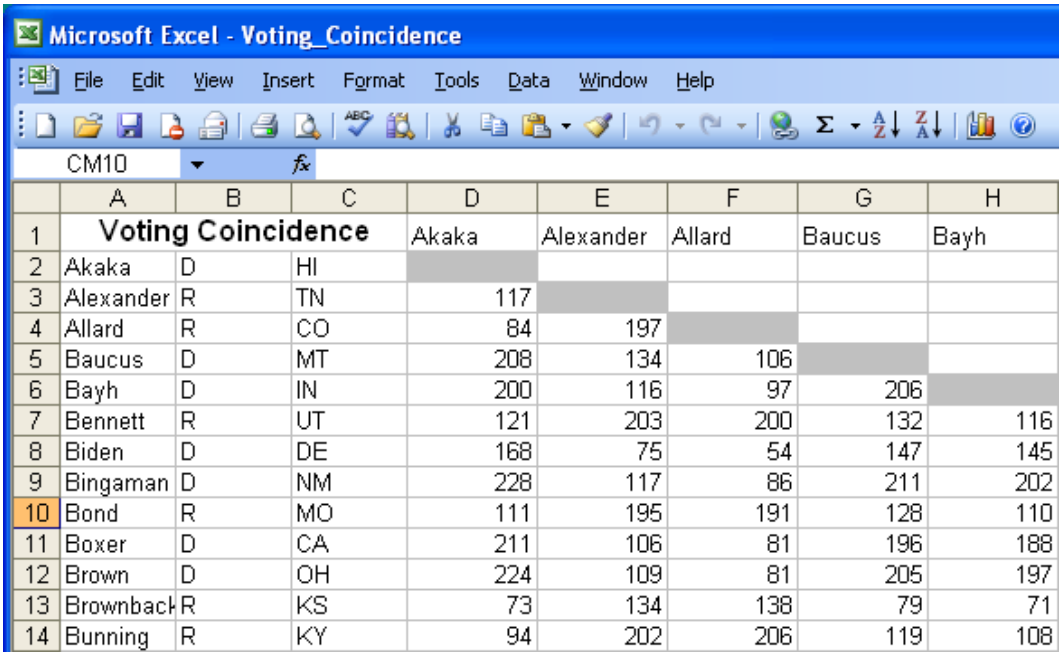
File Edit View Insert Format Tools Data Window Help

A2 fx 1

	A	B	C	D	E	F	
1	Roll Call	Date	Issue	Question	Result	Passed?	Description
2	1	8-Jan	S.Res. 19	On the Resolution	Agreed to	Yes	S. Res. 19;
3	2	10-Jan	S. 1	On the Amendment S.Amdt. 7	Agreed to	Yes	Vitter Amdt
4	3	10-Jan	S. 1	On the Motion to Table S.Amdt. 5	Agreed to	Yes	Motion to T
5	4	10-Jan	S. 1	On the Motion to Table S.Amdt. 6	Agreed to	Yes	Motion to T
6	5	11-Jan	S. 1	On the Motion to Table S.Amdt. 11	Rejected	No	Motion to T

Figure 89 Votes of senators on 247 issues.

Chris Wilson generated a voting coincidence matrix that shows the number of times each two senators voted the same way (Figure 90).



Microsoft Excel - Voting_Coincidence

File Edit View Insert Format Tools Data Window Help

CM10

	A	B	C	D	E	F	G	H
1	Voting Coincidence			Akaka	Alexander	Allard	Baucus	Bayh
2	Akaka	D	HI					
3	Alexander	R	TN	117				
4	Allard	R	CO	84	197			
5	Baucus	D	MT	208	134	106		
6	Bayh	D	IN	200	116	97	206	
7	Bennett	R	UT	121	203	200	132	116
8	Biden	D	DE	168	75	54	147	145
9	Bingaman	D	NM	228	117	86	211	202
10	Bond	R	MO	111	195	191	128	110
11	Boxer	D	CA	211	106	81	196	188
12	Brown	D	OH	224	109	81	205	197
13	Brownback	R	KS	73	134	138	79	71
14	Bunning	R	KY	94	202	206	119	108

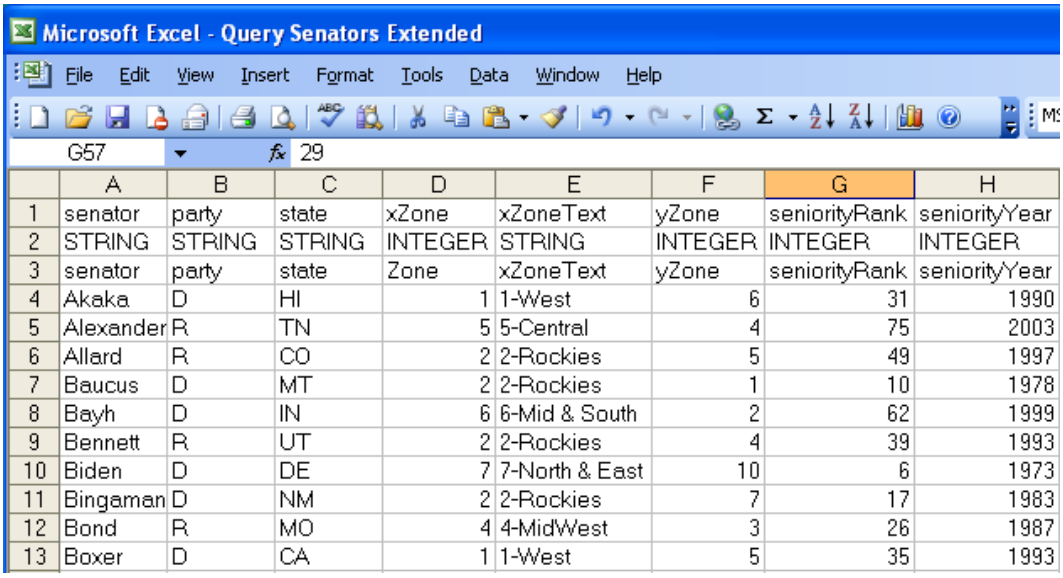
Figure 90 The voting coincidence between each pair of senators based on 247 issues.

The data in Figure 88 and seniority data from Wikipedia[5] were used to create the nodes file containing nodes and their attributes (Figure 91).

Nodes represent senators and links represent shared votes among senators. Links are undirected but were represented with directed links in NVSS.

The author of this dissertation collaborated with the domain expert to create three files containing link information for NVSS. One link file contains all votes that have been shared 250 times or more between two senators (Figure 92), while the other files have the threshold set to 200 and 150. In this section, examples are provided from the visualizations that include 250 and 200 shared votes between senators.

[5] http://en.wikipedia.org/wiki/List_of_current_United_States_Senators_by_seniority accessed Aug 17, 2007



	A	B	C	D	E	F	G	H
1	senator	party	state	xZone	xZoneText	yZone	seniorityRank	seniorityYear
2	STRING	STRING	STRING	INTEGER	STRING	INTEGER	INTEGER	INTEGER
3	senator	party	state	Zone	xZoneText	yZone	seniorityRank	seniorityYear
4	Akaka	D	HI	1	1-West	6	31	1990
5	Alexander	R	TN	5	5-Central	4	75	2003
6	Allard	R	CO	2	2-Rockies	5	49	1997
7	Baucus	D	MT	2	2-Rockies	1	10	1978
8	Bayh	D	IN	6	6-Mid & South	2	62	1999
9	Bennett	R	UT	2	2-Rockies	4	39	1993
10	Biden	D	DE	7	7-North & East	10	6	1973
11	Bingaman	D	NM	2	2-Rockies	7	17	1983
12	Bond	R	MO	4	4-MidWest	3	26	1987
13	Boxer	D	CA	1	1-West	5	35	1993

Figure 91 Nodes file containing senators and their attributes.

The *zone* (or xZone) attribute of senators is a derived attribute. It is computed based on the senator's state. States are divided into 7 vertical groups in terms of their geographical location. Each group is referred to as a *zone* and is designed to be as vertical as possible with some exceptions (e.g., Alaska is in Zone 1).

The definition of zones in terms of the states they cover is as follows:

Zone 1: Alaska, Washington, Oregon, Nevada, California, Hawaii

Zone 2: Montana, Idaho, Wyoming, Utah, Colorado, Arizona, New Mexico

Zone 3: North Dakota, South Dakota, Nebraska, Kansas, Oklahoma, Texas

Zone 4: Minnesota, Iowa, Missouri, Arkansas, Louisiana

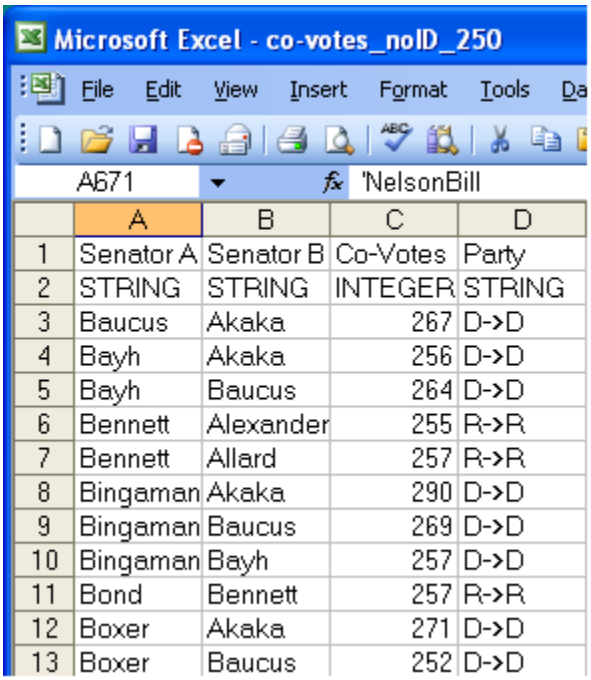
Zone 5: Wisconsin, Illinois, Kentucky, Tennessee, Mississippi, Alabama

Zone 6: Michigan, Indiana, Ohio, West Virginia, Virginia, North Carolina, South Carolina, Georgia, Florida

Zone 7: Maine, New Hampshire, Vermont, Massachusetts, Rhode Island, Connecticut, New York, New Jersey, Pennsylvania, Delaware,

Maryland, Washington DC

The two independent senators (Lieberman & Sanders) were treated the same as Democrats for simplicity of data pre-processing and analysis. This point was consulted with the collaborator and he found this acceptable and preferable. He mentioned that the independent senators were closer to the Democrats than the Republicans and therefore, this choice would be acceptable.



	A	B	C	D
1	Senator A	Senator B	Co-Votes	Party
2	STRING	STRING	INTEGER	STRING
3	Baucus	Akaka	267	D->D
4	Bayh	Akaka	256	D->D
5	Bayh	Baucus	264	D->D
6	Bennett	Alexander	255	R->R
7	Bennett	Allard	257	R->R
8	Bingaman	Akaka	290	D->D
9	Bingaman	Baucus	269	D->D
10	Bingaman	Bayh	257	D->D
11	Bond	Bennett	257	R->R
12	Boxer	Akaka	271	D->D
13	Boxer	Baucus	252	D->D

Figure 92 Links file for NVSS contains shared votes (for this file the threshold is set to 250 votes).

A derived attribute is computed based on seniority and used to size the nodes (senators) in the visualization. The larger the node, the more senior (higher rank, which is smaller integer) a senator is.

The notes for this case study contains 23 figures. In this section, only a few of those figures along with notes are included.

5.3.1.1 Shared votes are limited to 250 or more:

In the following figures (Figure 93 - Figure 99) the link file that includes shared votes of 250 or more is used.

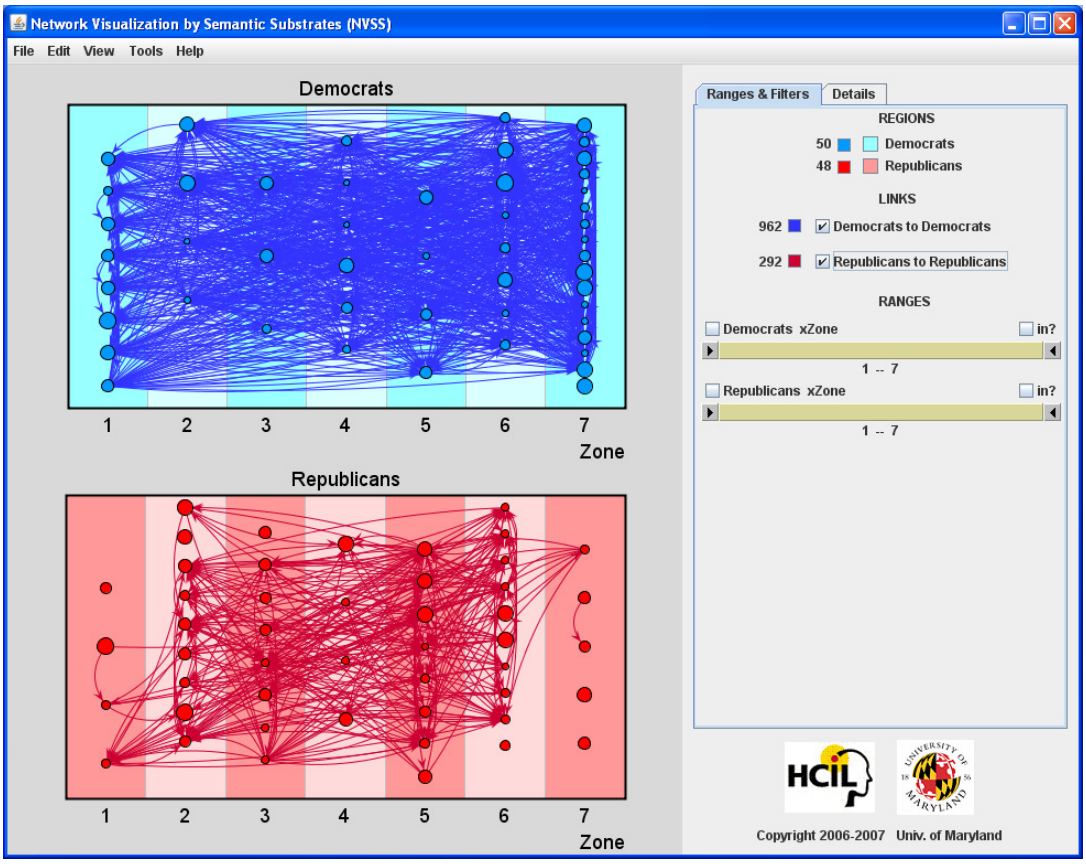


Figure 93 Democrats share more votes among themselves than republicans.

There are no votes shared between Democrats and Republicans in Figure 93. This means no two senators from different parties have shared more than 250 votes. Democrats share more votes among themselves than republicans (962 vs. 292). There are considerably more democrats in zone 7 than republicans (note the number of circles in zone 7 of the Democrats region in Figure 93).

By looking at the following visualizations in this section, some evidence can be found that location may play a role for the number of shared votes between

senators. Closer zones tend to share more votes. See the following figures in pairs (Figure 94 and Figure 95).

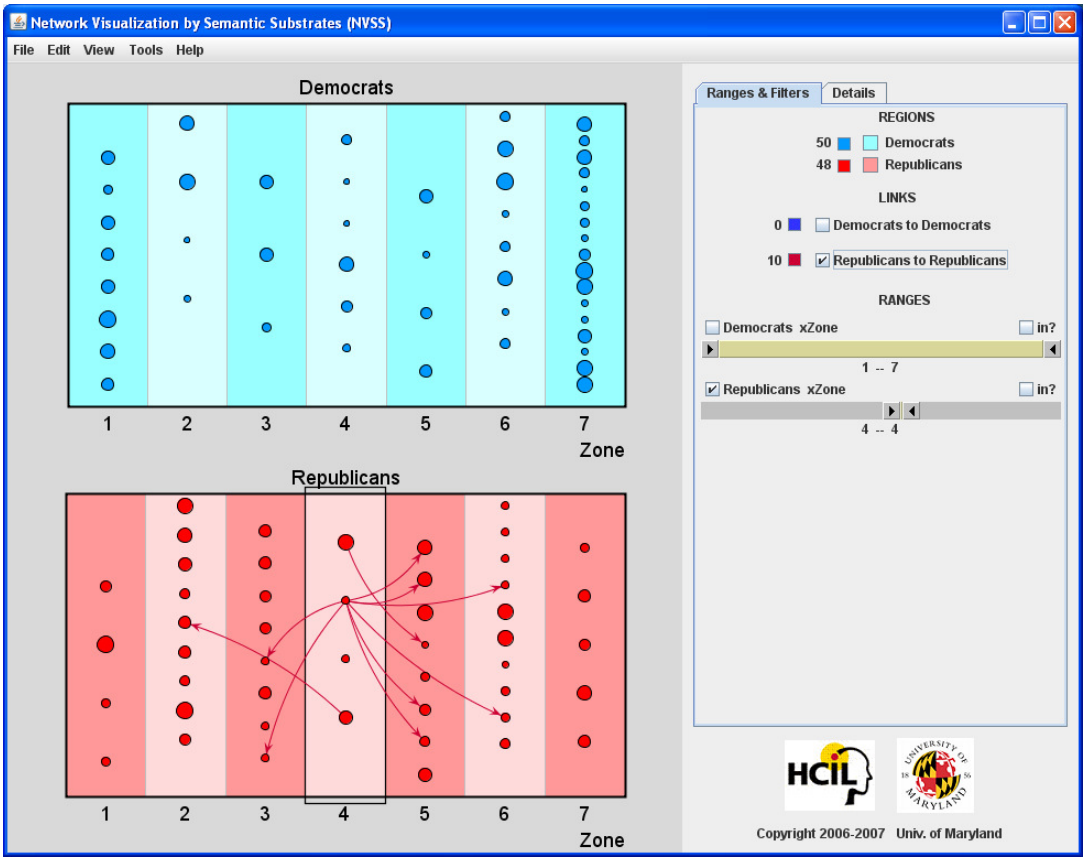


Figure 94 Shared votes in zone 4 are closer among republicans - part 1 of 2.

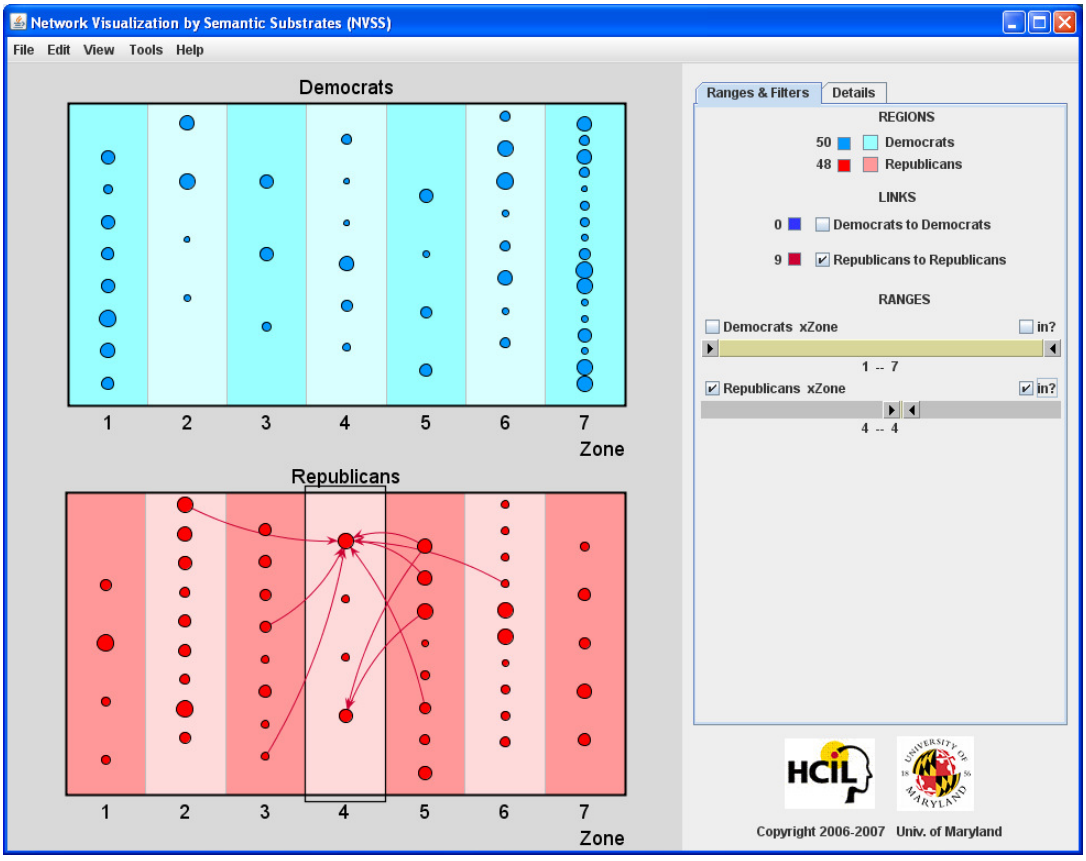


Figure 95 Shared votes in zone 4 are closer among republicans - part 2 of 2.

Some more evidence from zone 5 is in the following two figures (Figure 96, Figure 97). In this case, 4 out of 5 senators in zone 7 are not involved at all. Another observation is that more votes are shared in zone 5 than zone 4 and the sharing seems to be wider geographically.

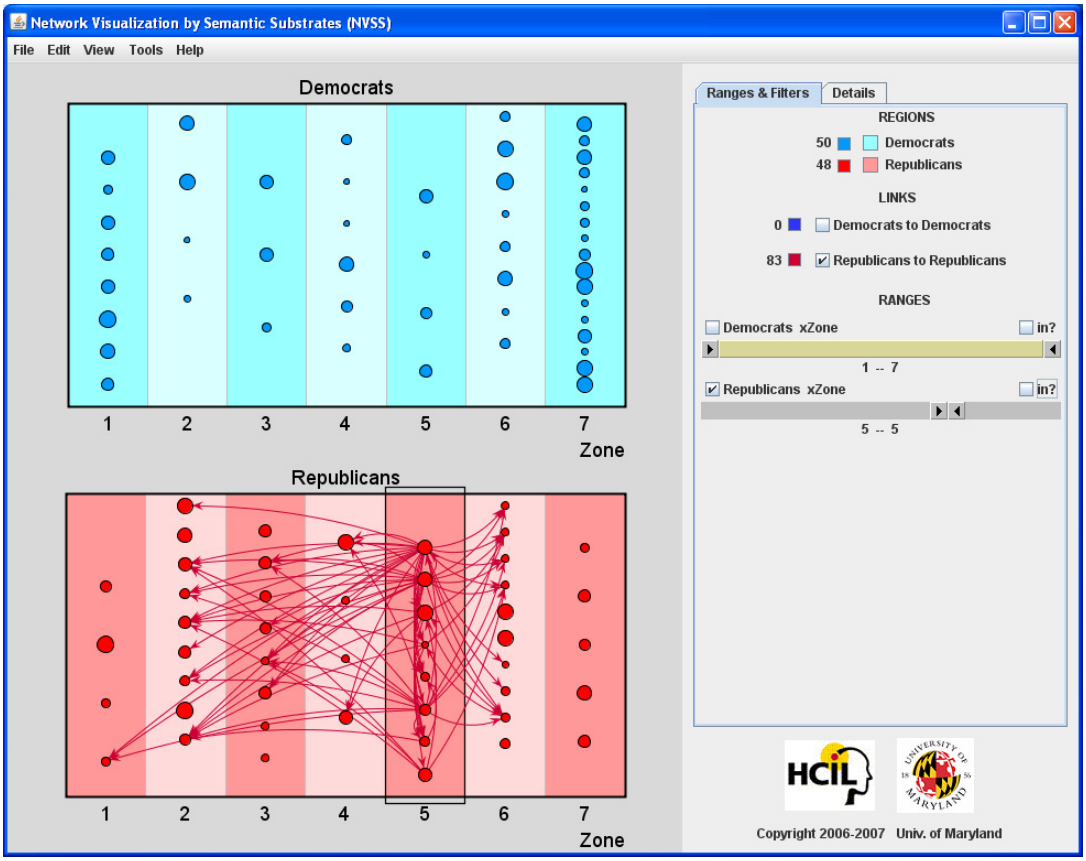


Figure 96 Republicans in zone 5 - part 1 of 2.

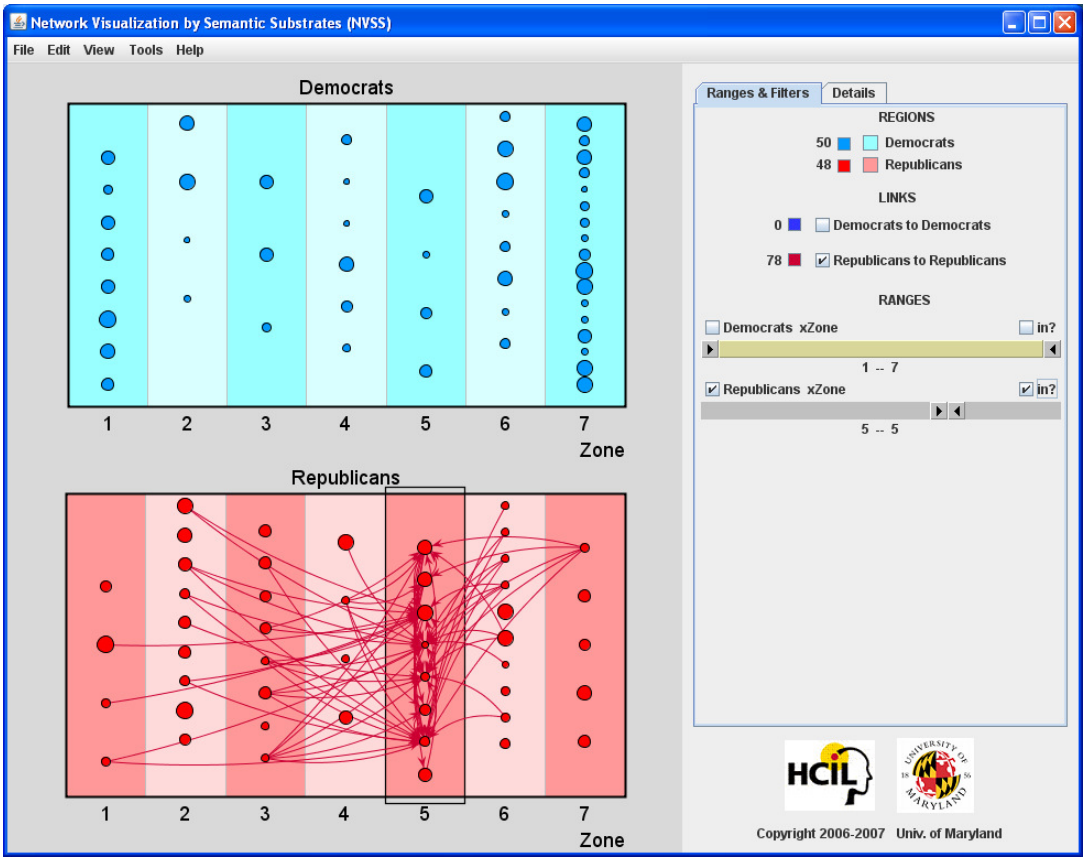


Figure 97 Republicans in zone 5 - part 2 of 2.

Looking at zone 7 provides additional support for the hypothesis that geographical location may play a role in the number of shared votes. Mostly zones 5 and 6 are involved. Not many shared votes exist with the other zones (Figure 98 and Figure 99).

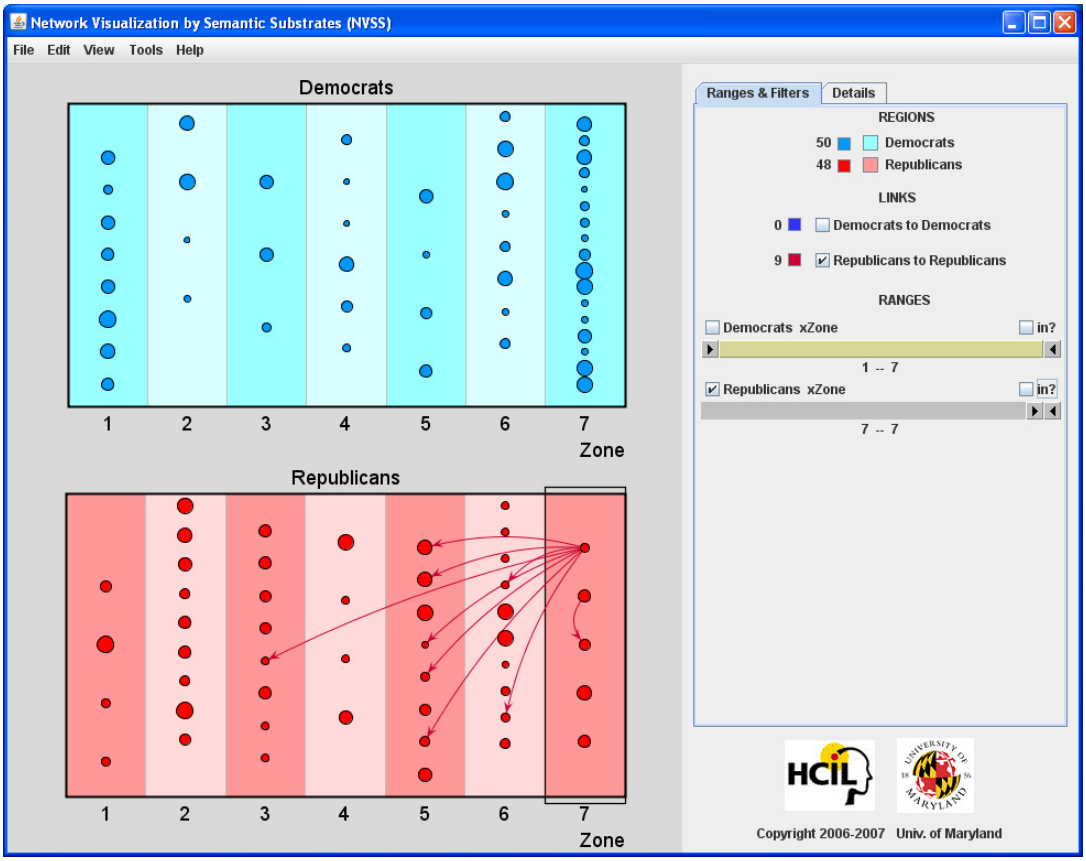


Figure 98 Republicans in zone 7 – part 1 of 2.

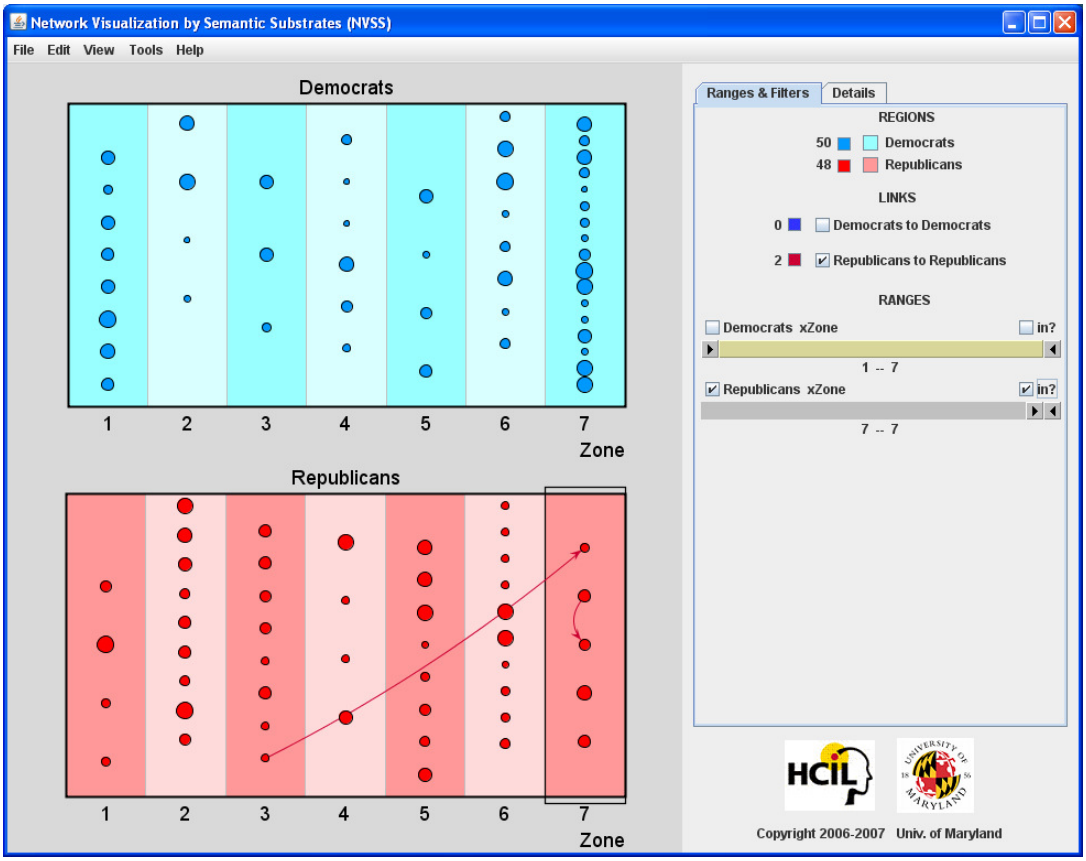


Figure 99 Republicans in zone 7 part 2 of 2.

This section concludes with the overview in terms of zones. In the next section, the threshold is set to 200 votes between senators.

5.3.1.2 Shared votes are limited to 200 or more:

In the following figures (Figure 100 - Figure 101) the link file that includes shared votes of 200 or more is used.

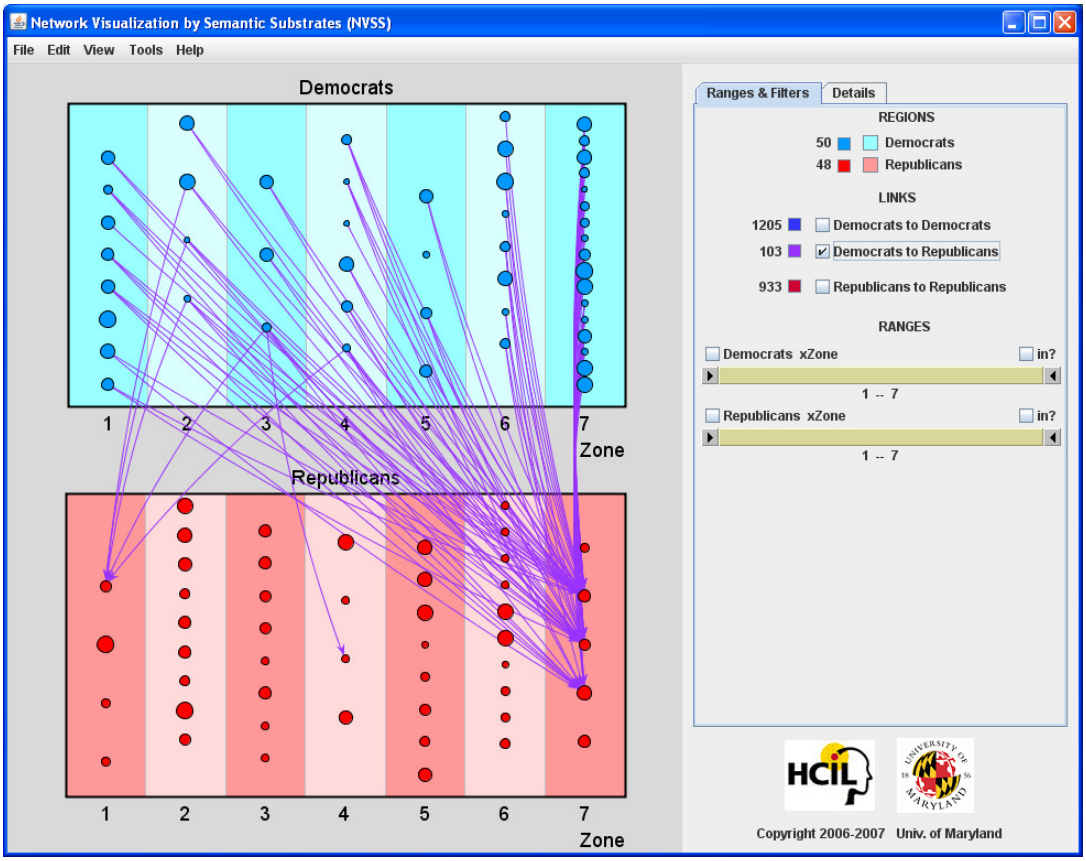


Figure 100 Shared votes between Democrats and Republicans for more than 200.

When shared votes are limited to 200 or more, shared votes between Democrats and Republicans appear. There are 933 shared votes between the two parties. The links Democrats->Republicans and Republicans->Democrats were unified to Democrats->Republicans for simplicity of analysis.

Looking at the votes shared between the two parties (Figure 100), 4 republicans appear to share multiple (more than one) votes with many Democrats.

3 of these 4 republicans are in Zone 7 and they share votes with more Democrats than the other republican senator. These 3 republican senators are:

1) Snowe from ME

2) Collins from ME

3) Specter from PA

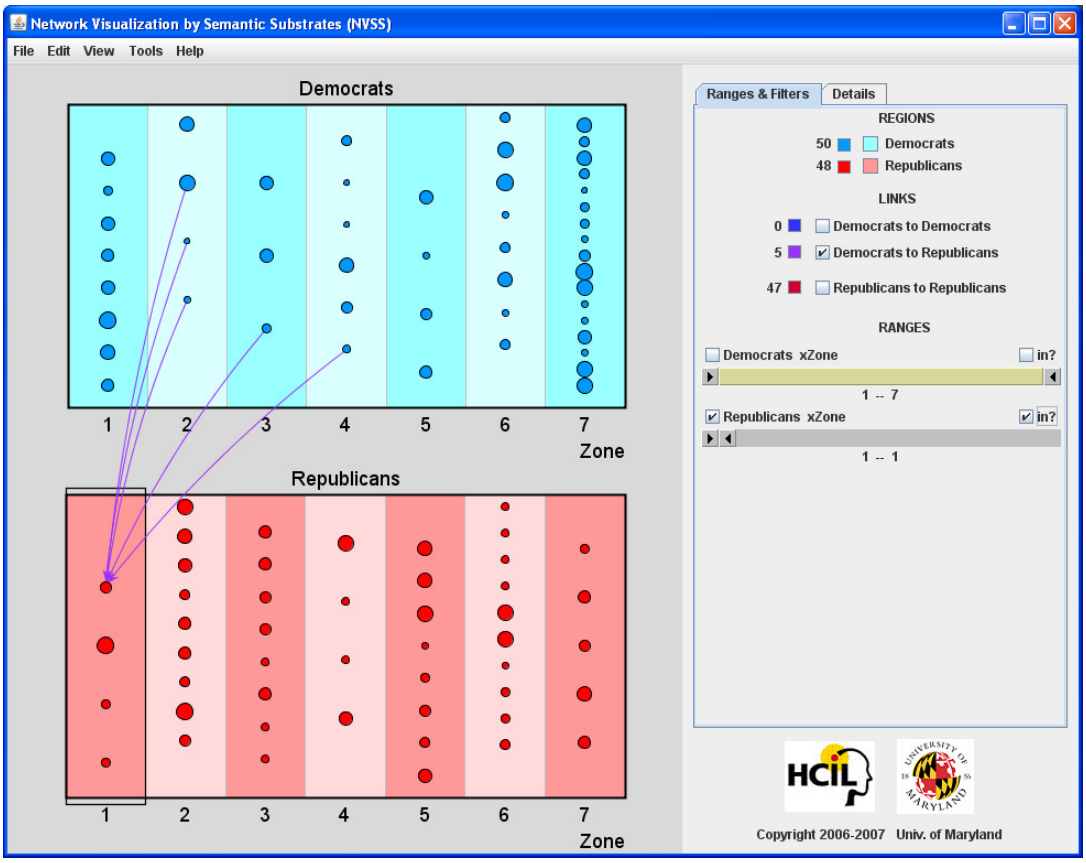


Figure 101 Shared votes restricted to zone 1 in Republicans.

The remaining republican is in zone 1 (Figure 101) is:

4) Smith from OR

Smith shares votes with 5 democrat senators:

- Zone 2
    - Baucus from MT
    - Tester from MT
    - Salazar from CO
- Zone 3
    - Ben Nelson from NE
    - Pryor from AR

In this section, the visualization helped to identify the Republicans who share more than 200 votes with the senators in the Democrat party.

Since the visualizations are observations in which the collaborator could not participate, this case study has no research results in terms of the use of the semantic substrate approach by domain experts.

5.3.2 IOpener Case Study

The case study participants are the IOpener research project group members initiated at the University of Maryland, College Park. Participants from University of Maryland are Prof. Bonnie Dorr, Dr. Judith Klavans, Asst. Prof. Jimmy Lin, Dr. Saif Mohammad, and Prof. Ben Shneiderman. The other participants are Master's student Vahed Qazvinian and Professor Dragomir Radev from the University of Michigan. The participants are highly experienced with data analysis either in information retrieval or in natural language processing. Some of the participants (e.g., Jimmy Lin) are also experienced in using visualization tools.

The Topics4 dataset is a citation dataset that contains papers in four topics. They were interested in exploring the dataset for various reasons including understanding the citation patterns, terminological uses that can contribute to the understanding of deep semantic relationships between documents, and for the purpose of summarization for survey creation.

This case study started with a small dataset that contained Treemap papers (publications from the Computer Science literature that use the Treemap concept (Johnson 1991), (Shneiderman 1992)) to illustrate to the IOpener group meaningful

exploration of papers in NVSS (this was illustrated to them in the 1st session by Ben Shneiderman, the only session where the author of this dissertation was not present in the session). It continued with exploring another small dataset of 27 papers and 39 citations consisting of PBMT (Phrase-Based Machine Translation) papers from the Computer Science literature as in Figure 102 (this was illustrated to them in the 2nd session).

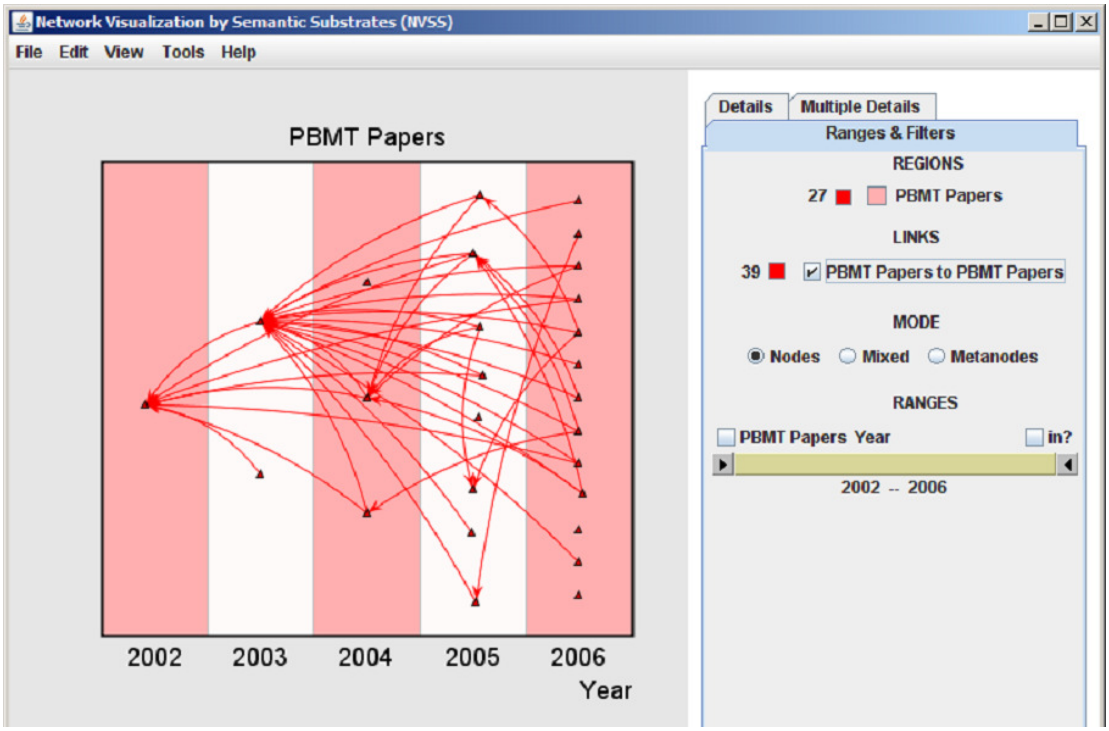


Figure 102 PBMT papers in NVSS with Year attribute on the x-axis.

While Figure 102 spreads the PBMT papers on the y-axis for link visibility, Figure 103 uses the number of incites on the y-axis to separate highly cited papers (lower on the figure) from the others. The key paper by Koehn in 2003 (the one enclosed in a circle) and citations to it are salient in Figure 103.

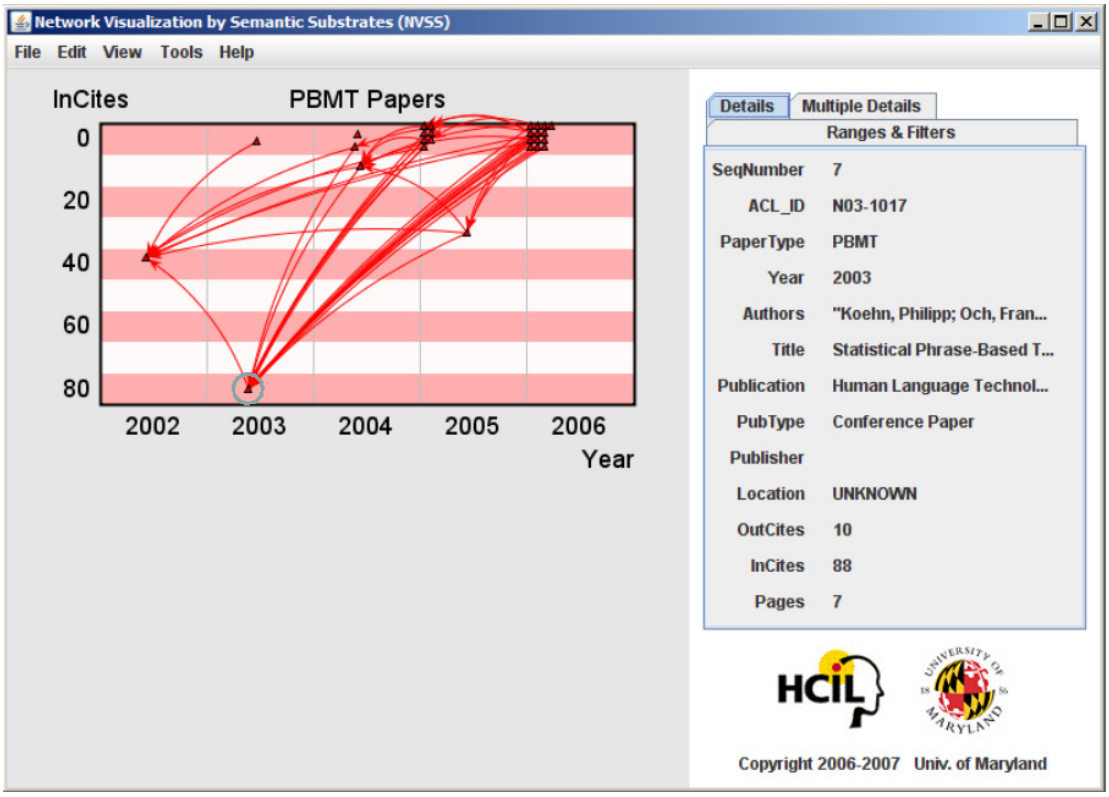


Figure 103 PBMT papers with highlight on key paper by Koehn et al. in 2003.

The Topics4 dataset includes these PBMT papers and papers from three other topics. In the third session, the Topics4 dataset was shown to the IOpener group to get their reactions. First, two of the researchers (Bonnie Dorr and Judith Klavans) tried to answer the questions on a quiz that was prepared by Ben Shneiderman beforehand. After seeing the visualization, they said they understood the questions better. Visualizing the data (compared to talking about it in abstract terms) seemed to help them understand the questions.

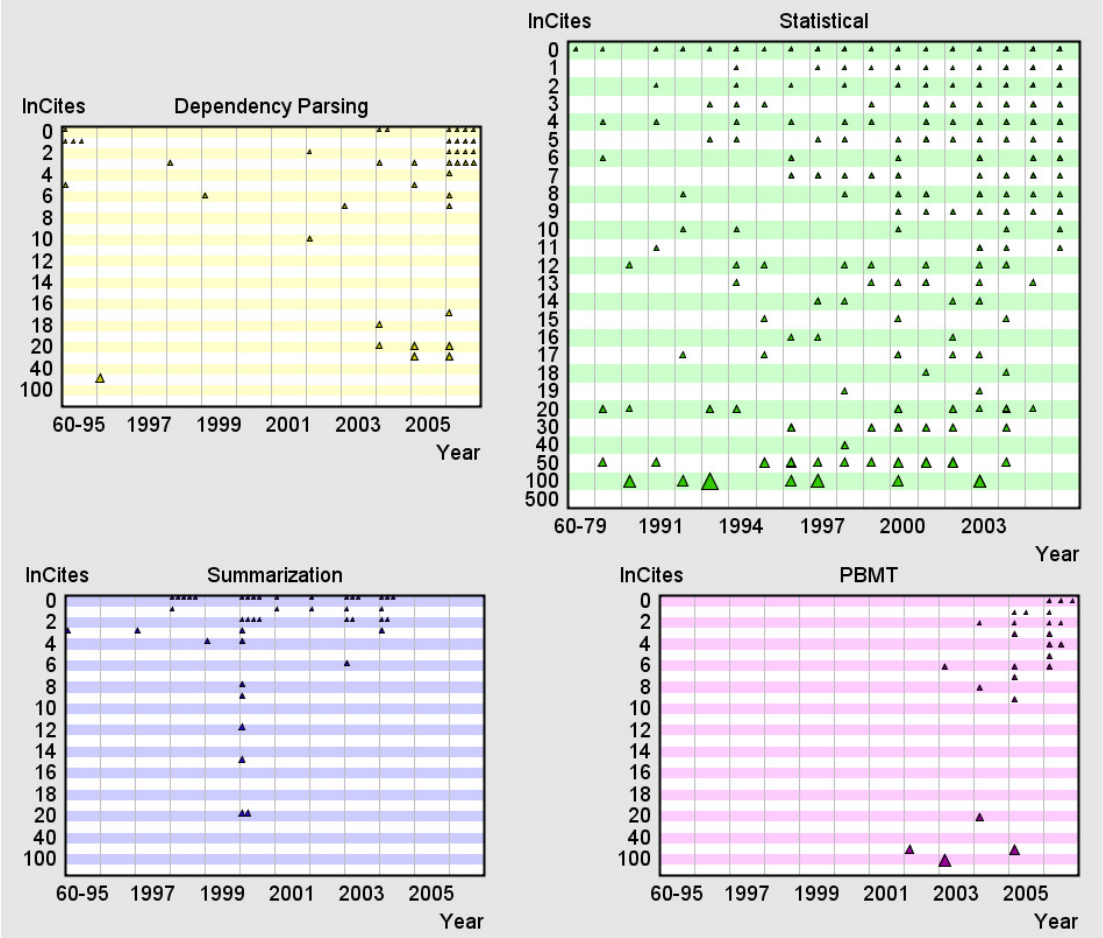


Figure 104 The Topics4 dataset without any links.

Figure 104 shows the four topics, each in a region. As in the previous PBMT example, the x-axis represents years while the y-axis represents the number of citations a paper has received (*inCites*). In this substrate, there is a difference on the x-axis from the previous ones in that there is custom binning enabling the first column of the small region to represent the years 1960-1995, and the first two columns in the Statistical region represent the year ranges 1960-1979 and 1980-1989, respectively. All other columns represent the following single years.

Figure 105 shows citations within each topic area (enabled using the link filters in NVSS).

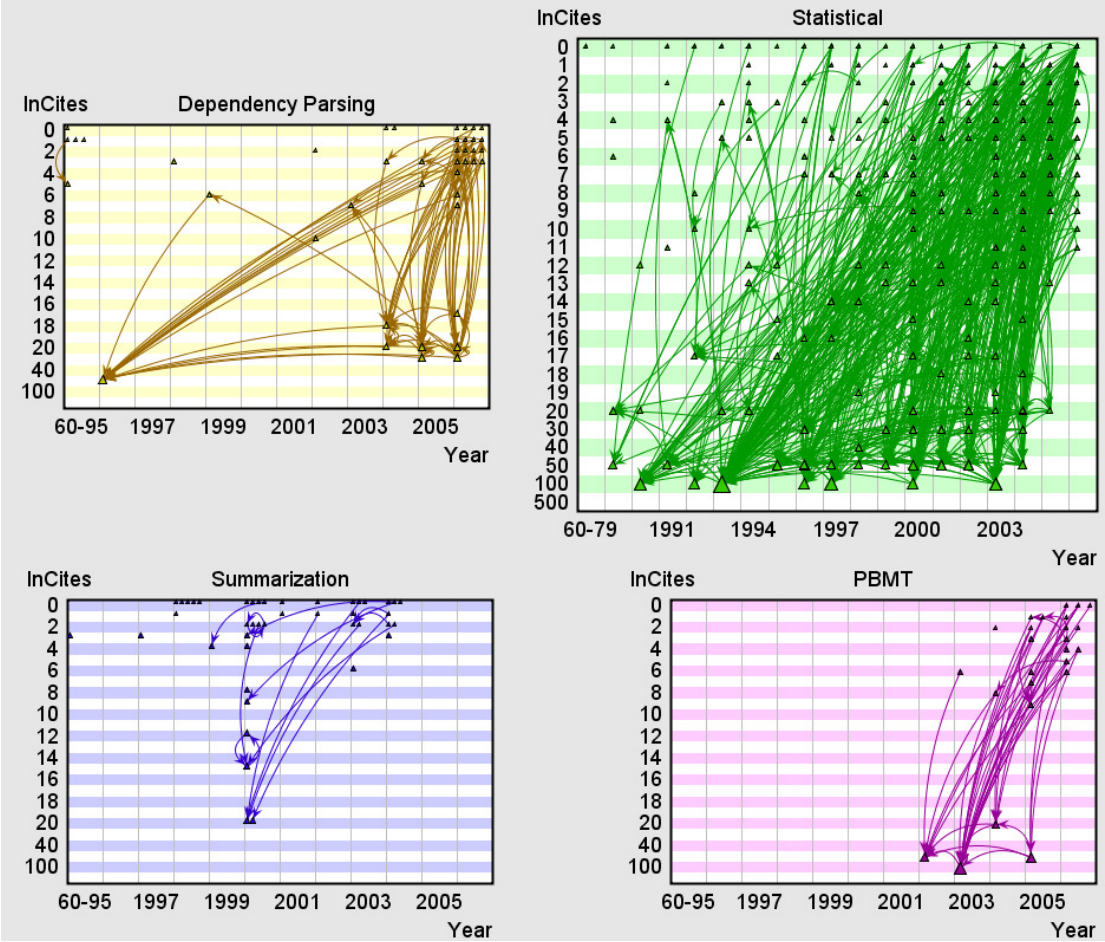


Figure 105 Enabling links within each topic area.

The collaborators easily found the highly cited paper in the DP (Dependency Parsing) region and they recognized it from their domain knowledge. They did so for a few other highly cited papers, and recognized them, too. Then, they also looked at the Summarization region to find one of the team member's (Dragomir Radev) papers and found it among the highly cited ones as they expected. This confirmed their expectations. They did so for the remaining regions as well. When looking into the PBMT (Phrase-Based Machine Translation) region, a paper was found to be highly cited and the reason was not understood by the team. One of the team members

(Jimmy Lin) provided an explanation for it, which was introducing a methodology that other papers used and, therefore, frequently cited.

With the Statistical region, filters became useful as this region contains more nodes and more links than the others exceeding the threshold of comprehensibility when all links are viewed. I started showing them how to apply the filters and the team members quickly understood it. In fact, they started to use it and wanted to explore the dataset in this way. One of the researchers (Judith Klavans) expressed enthusiasm to continue exploring this region as it was in her domain of expertise. She wanted to see where the patterns match her understanding and whether there was more information that she could discover. She also provided an explanation for the contrast of citation patterns with respect to time between the Dependency Parsing and the Statistical fields. She commented that in the Statistical field the papers cite the recent papers and that was good and expected, while this behavior would be considered undesirable in the Dependency Parsing field, and therefore, they would cite the early papers instead more often. Klavans attributed this to different cultural properties regarding citation between the two groups of researchers working on different topics. She also commented that this way of visualizing could be used as a way to compare citation cultures across fields in general.

Next, we looked at the distribution of papers from other fields that cite papers in the Statistical region (Figure 106).

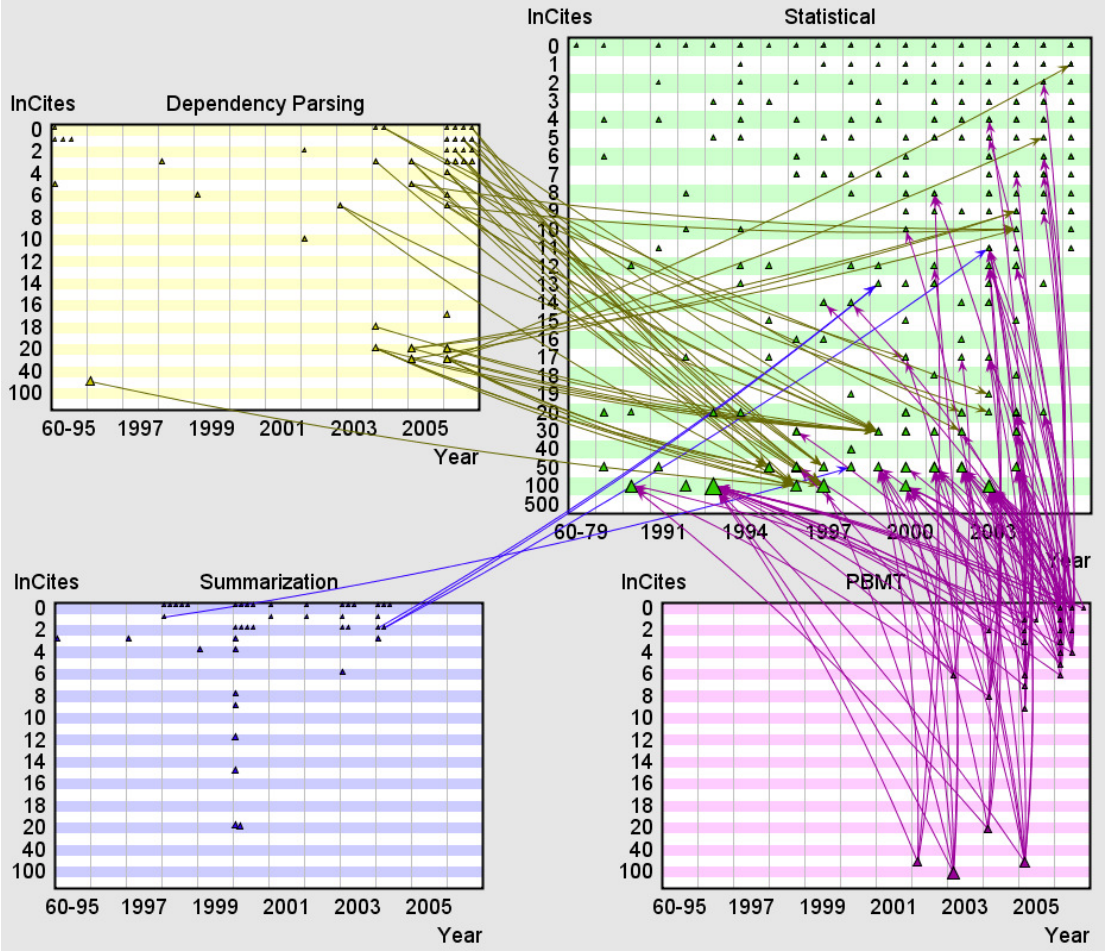


Figure 106 Citations from other topics to the Statistical region.

In this view (Figure 106), their expectations were confirmed that DP and PBMT made more use of the Statistical papers while the Summarization papers cited only 4 Statistical papers. Also, they confirmed that PBMT is more statistical in nature than DP and that DP has a history that is unrelated to the Statistical field.

In the meantime, they were asked what the relationships among the small regions were (i.e., DP, Summarization, and PBMT). At first, they didn't understand and had no idea. (Perhaps, this wasn't a point that they thought of but when brought to their attention, they started to think about it.) When they realized that there is no

relation due to the lack of links between regions in NVSS, they found this fact interesting; however, this didn't seem something that would be useful for them.

Next, we looked at what papers were cited from the papers in the Statistical region (Figure 107).

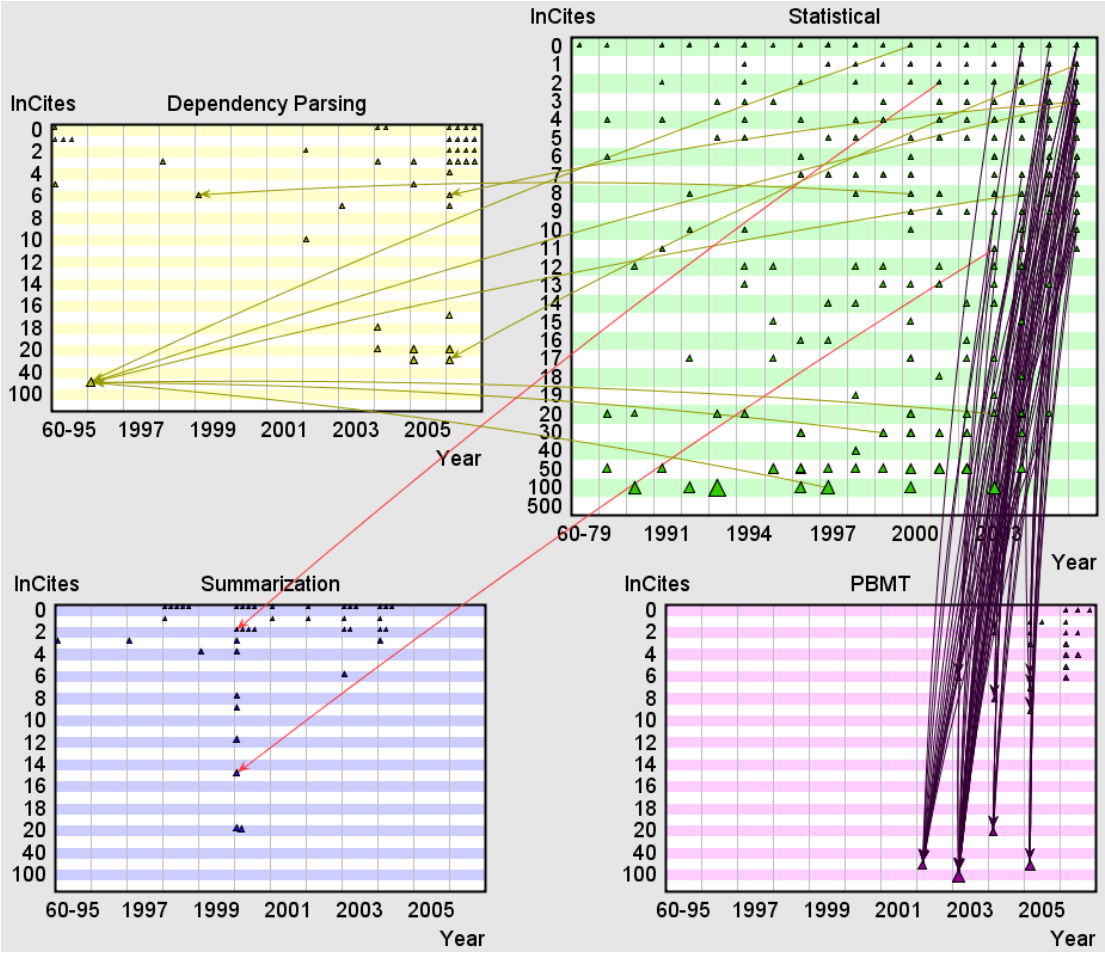


Figure 107 Displaying links from the Statistical region to other topic areas.

Judith Klavans was surprised by the fact that the papers in the Statistical region cite only 2 papers in the Summarization field. She said that the summarization field generally is based less on statistical modeling, so fewer references would be expected. However, she would expect more than 2 citations and was surprised by the few citations.

Lastly, we explored links within the statistical region by applying filters (Figure 108).

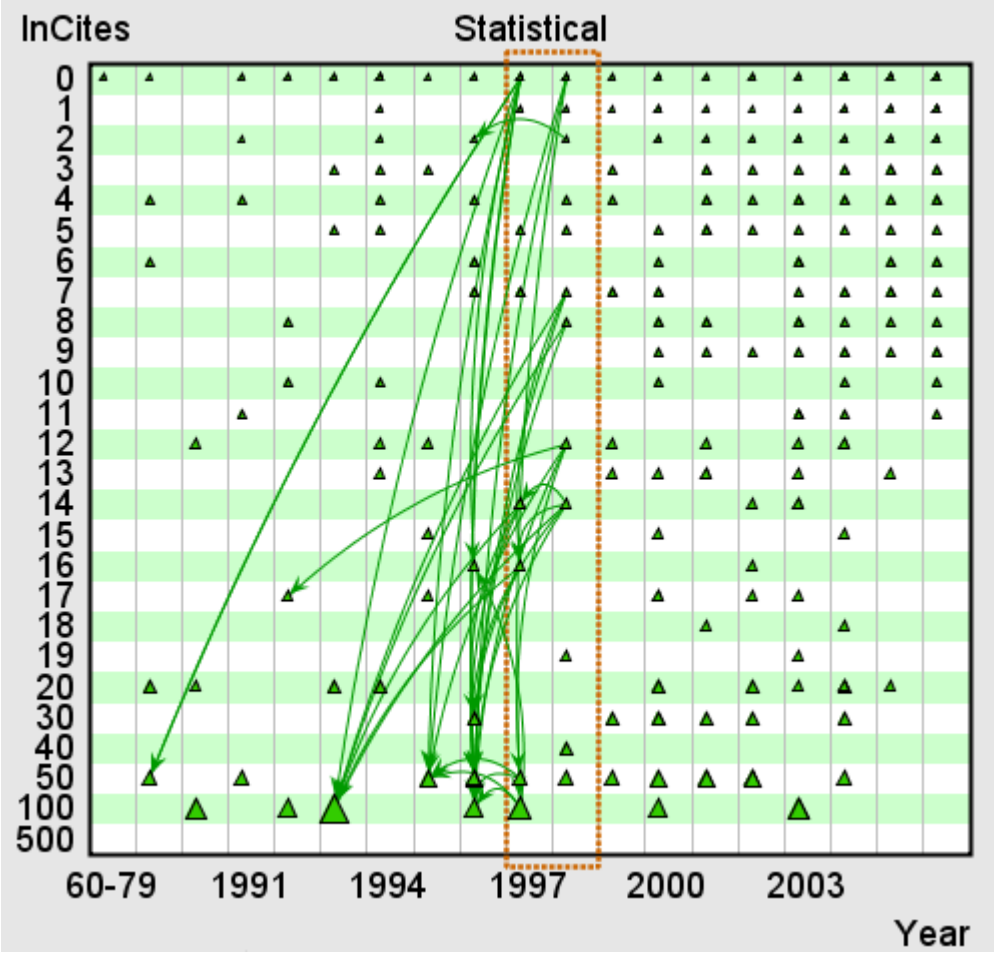


Figure 108 Filtering links according to Year in the Statistical region.

Although time didn't permit for more exploration, seeing the link patterns led to excitement. Judith Klavans said that NVSS is great and the team expressed that it would be interesting to explore a dataset like this using NVSS.

Since this was a short-term case study, there was little chance to obtain many research results. However, it revealed possible improvements and challenges in the user interface design: node placement inside cells could be improved by considering the node sizes in each cell rather than considering the node sizes in the entire region

(see Figure 108; each cell at the top raw suffers from node occlusion; however, there is space to minimize the occlusion). When nodes are occluded, they do not look very different from a single node (see Figure 108, each cell in the top row looks as if it contains a single node; however, in fact, each cell contains more than one node and they are occluded). The presentation of the occluded nodes can be improved.

Nevertheless, the exploration with the case study participants gave an idea of whether this approach would be interesting to researchers highly experienced in data analysis and whether they would consider using it. The (reported) conversations indicate that they would.

5.4 Summary of the Case Studies

This section gives a standardized summary of all case studies for ease of comparison while providing an overview.

The following table (Table 6) provides an overview of case studies.

Table 6 Summarizing features of all case studies.

Case Study or Dataset	Number of people	Number of nodes	Number of links	Regions	Placement attributes	Approx. span
Init. Post78	3	287	2032	3, courtType	Year & circuitNo	1 year
Post78	3	287	2032	3, courtType	Year & circuitNo	1 year
Giants	3	1173	6638	3, courtType	Year & circuitNo	2-3 months
Pre78	3	493	1170	3, courtType	Year & circuitNo	1 month
Food-Web	1	640	1978	4, metabolic category	Length & mass	2-3 months
TobIG	2	4,296	16,385	3, type (node	Year, CR, LCS,	7-8 months

				type)	Count	
SenateVotes	1	98	2915	2, party	Zone, Seniority	2 months
IOpener	7	487	1079	4, topic	Year, InCites	3-4 months

The following table (Table 7) provides strengths and weaknesses identified in each case study:

Table 7 Stregths and weaknesses of the semantic substrate approach identified in each case study.

Case Study	Identified Strengths	Identified Weaknesses
Init. Post78	The semantic substrate (Figure 37) combined with the year filter reveals the longevity of court cases by making within and across court citations visible.	The restrictive node location leads to link overlap and makes it hard to separate them visually (Figure 36).
Post78	The substrate helps identify the parallel links in Circuit and District region that attest the tendency for courts to cite within their circuit (Figure 40). The use of the year attribute makes cases with longer temporal difference in their citations pronounced (Figure 42, Figure 44). Aligning circuits within Circuit and District regions on the x-axis shows the within and across circuit citations well across these court types (Figure 43, Figure 45).	When there are many links, it is hard to follow each individually (Figure 45). Sometimes, this is so even with the help of filters (Figure 46).
Giants	The compact representation of the regions and the year on the x-axis allows the visualization of long-range (100 years) attribute values (Figure 48).	When there are many links within a cell, it is hard or impossible to identify each link (Figure 47).
Pre78	The use of time on the x-axis spotlights gaps and high-activity areas (Figure 56, Figure 64, Figure 65).	The random placement of nodes on the y-axis may accidentally draw attention to some cases due to long connections over time but may fail to reveal others (Figure 56, note the Supreme Court case at the top of the region after 1960

		and the earlier cases it cites).
Food-Web	The semantic substrate approach allows looking at the same data from different point of views. The substrate in Figure 66 gave a good overview of the 7 datasets showing which are the predators, what their prey are, and the volume of the relationships. On the other hand, the substrate in Figure 67 revealed relationships in terms of length and mass.	When there are many links, the directions of the links are hard to discern (Figure 66; see links connected to photo-autotrophs).
TobIG	The node aggregation feature simplifies the network (Figure 27) and makes it easy to identify interesting phenomenon in the dataset (Figure 71).	The node aggregation view lacks summary information on what portion of the aggregated nodes is cited leading to only partial knowledge (Figure 87, Steve Hecht could have written all documents or one document from each aggregated node, users can't discern).
SenateVotes	Placement of nodes help finding patterns and conceptualize the description of the pattern in terms of the meaningful placement. Among all republicans, east coast (zone 7) republicans seem to behave the most similar to democrats (Figure 100).	The inability to view both incoming and outgoing links at the same time breaks down (naturally what would be) one view into two views (e.g. Figure 94, Figure 95). The random assignment of nodes on the y-axis could be misleading. Alignment of nodes with the links leads to a poor display (Figure 100, see zone 7 links).
IOpener	The lack of many citations becomes immediately apparent (when viewing links across summarization and statistical regions in Figure 106 and Figure 107).	The presentation of occluded nodes is similar to the presentation of a single node (Figure 108, nodes in the top row), which may be misleading. Also, when there is node occlusion, available space is not used efficiently if node sizes vary within a region (Figure 108, nodes in the top row). This could be improved by modifying the node placement algorithm.

The following section provides the reflections of the author of this dissertation on the case study approach used in this dissertation.

5.5 Reflections on the Case Studies

The case studies in this dissertation followed the guidelines of the MILC method (Shneiderman 2006). The following is a summary of the experiences gained and suggestions to researchers willing to conduct a case study evaluation method similar to the one in this dissertation.

The feedback from the case study participants indicating the utility of the semantic substrate approach was encouraging. They believed this method was novel and they produced insights that they believed they would not be able to get using other methods or the methods they tried before trying the semantic substrate approach. This conjecture needs to be validated in future case studies and perhaps partially by controlled experiments.

The MILC evaluation method offers wide flexibility to evaluate a new interface design. However, this benefit came with the trade-off that the MILC method provides little or no guidance on how to structure the evaluation. Therefore, many times, it has been hard to structure and plan the process and decide about what to do next. There has been a beginning phase of using the case study evaluation method, where little structure existed. This was perhaps comparable to a pilot study. From the experience gained, case study guidelines were formed (section 5.1). This greatly helped to capture the case study results. Capturing results requires diligence and sometimes it has been impossible to capture every interesting event in the case studies due to the high pace of occurrences and the inability to keep up with the pace. In

addition, notes had to be taken carefully and briefly to allow the flow of the events while capturing the essence of the interactions to be able to reproduce and write it later in detail.

It is quite difficult to conduct a case study session while taking notes and also remember the details later. To researchers contemplating of conducting such case studies, it is recommended that they have another person assist them if possible, or use a voice recorder (and get permission from the case study participants to record their voices). However, transcribing the voice recordings may take a very long time. Perhaps, combining this with writing notes will be helpful. Then, notes could be the primary source of reproducing the case study events and voice recordings can be referred to only when necessary as a secondary resource to either remember or confirm details.

Case studies require conducting sessions, adding to the case study documents, reproducing events, and communication with case study participants through meetings, email, and phone. Furthermore, it is advisable to reproduce and write events early and check with the participants as soon as possible (while their memory is still fresh) and resolve any questions. These activities together require a high-level of time commitment from the researcher conducting the case studies. Therefore, it is highly recommended to limit the number of case studies conducted at the same time. Two or at most three case studies that have the pace of 1 sessions per week is likely to be sufficient and perhaps the limit. In addition, other factors, such as, developing the implementation, correcting errors, adding or modifying features take more time as the number of case studies or sessions per week increase. It is also recommended that the

implementation used in the case studies be in a good state in terms of functioning correctly. Incorrect algorithms, inadequate error handling and error messages (e.g., for the input files), misleading labels and information is advised to be minimized or eliminated soon after they are discovered as these lead to not only low-quality and sparse outcomes, but also may discourage and frustrate the case study participants.

Although usability of the implementation may not be focus, some issues need to be addressed to increase the accuracy of results with respect to the user interface design being evaluated. The researcher is advised to be prepared to address and handle usability issues despite that this is not the focus of the evaluation. For example, a lengthy process of inputting data could discourage some participants or obtain a negative impression on the software they are using.

The validity of the case study evaluation can be questioned when only one person is taking notes and reproducing the information later. Checking the results with the case study participant could abate this threat to validity. However, the case study participants may not give this task enough attention. The option of having another person to report is to be considered for additional validity or finding the common points in two varying results. Many times, however, having a second person may not be feasible.

Interaction with case study participants in the context of evaluating and developing a novel user interface design is challenging. It requires social and communication skills. Setting expectations too early may harm the future of the case study, while not setting enough expectations may create frustrations, a sense of disconnectedness from the case study, or the lack of control for the case study

participants. Communication and an incremental approach with feedback from the case participant when defining the next steps and expectations are suggested.

Since the course of the case studies are usually undefined and the results unknown, the stability of the relationship with the case study participants is also unknown. Sometimes, it is not possible to address the needs of a case study participant. Accordingly, a case study may start with no or little commitment and may end after a few sessions. The researcher is encouraged to be prepared for such possibilities. Furthermore, external factors, i.e., factors other than the case study, may contribute (e.g., changing jobs, moving to another place, project funding status changes (for the case study participants), commitment to too many projects, lack of time, etc.). However, external factors could be overcome. The author of this dissertation experienced successfully completed case studies whose participants moved and changed jobs. However, he also experienced case studies terminating due to changing jobs and/or lack of interest. It is suggested to consider such instances as normal and handle them with a calm and understanding attitude.

The MILC method is in need for more application examples to provide researchers with options for strategies. A few dissertations that used this method (by Jinwook Seo (Seo 2005) and by Adam Perer (Perer 2008)) provide some additional information or guidelines. In addition, Perer et al. (Perer 2008) structures the case study in stages. These are useful additional guidelines and more guidelines may be needed for various situations or additional validation.

Finally, the MILC method seems to have provided the most benefits considering the available resources and the variety of outcomes. This variety would

not be possible with a controlled study. On the other hand, a case study evaluation method doesn't have the same authority as a focused and controlled study. However, it seems that the case studies benefited the participants and provided them with a new way of thinking for their problem. In this respect, the MILC method seems to have been successful for this dissertation and the method could benefit from further improvements (e.g., a set of guidelines for structure, perhaps categorized or indexed in terms of influencing factors).

Chapter 6: Guiding Principles of Design

The following sections provide the design guidelines arrived at through the experience gained from the case studies. Examples will be given from the long-term case studies (section 5.2).

6.1 Semantic Substrate Design Guidelines

The experience of designing semantic substrates with domain experts let to an initial set of 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 attributes to be explored.

(2) Favor attributes with uniform distributions to spread out nodes evenly. Transform attributes (e.g., by using sqrt(X) or log(X)), if necessary, to make their distribution more uniform.

(3) To save screen space, minimize or eliminate gaps (by transforming or selecting attribute values) and avoid outliers (possibly by deleting them or setting a maximum value).

(4) Align regions to facilitate comparison.

(5) Locate regions to minimize link length and link overlaps while facilitating comparison.

These guidelines will be referred in the following sections as (1), (2), (3), (4), and (5).

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

6.1.1 Selecting a Grouping Attribute

Experience suggests choosing 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, dataset 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 dataset (see sections 5.2.1.1 and 5.2.1.2) and the *metCat* attribute (metabolic category) with the food web dataset (section 5.2.2).

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).

Knowledge of what attributes are available in the dataset, 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 66, which was useful to reveal incoming links. Users who are knowledgeable in the aspects described above will have an advantage. Otherwise, they can acquire 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 in section 7.1.2).

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 decide what attributes to choose for placement ahead of time and not choose these attributes as grouping attributes.

Users may want to select values of different attributes for each region. In that case, as long as each node falls into a unique region and attribute values for grouping together cover all nodes in the dataset, NVSS will be able to display the dataset in this type of a substrate.

To summarize, experience suggests that a grouping attribute that has 2-5 values divides the dataset into meaningful subgroups or categories. Users will need to know what attributes are available in the dataset, 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 dataset.

6.1.2 Determining the Placement Method

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

A useful practice is 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 dataset, *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 dataset (Figure 40 - Figure 46) (2). Outliers may pose a challenge as in the invertebrate region in Figure 66 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 66; 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 useful if the values of the selected attributes have a uniform distribution (2) across the selected range. Although this is ideal, it is not always necessary to gain insights. There are uniform distributions in the legal cases dataset but non-uniform ones in the food web dataset. For instance, photo-autotrophs are not distributed in a balanced way across studies in Figure 66. 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. discuss several options for transformations (Aris 2005).

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 acquire this information by iterative design and application of substrates or the application can present this information to users (see future work in section 7.1.2).

6.1.3 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 is 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 (4). 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 dataset, they can iteratively refine the design of their substrate until desired insights are gained.

Among choosing node, background, and link colors, choosing the link colors seems to be the most crucial issue, especially when there are many links on the display. Experience suggests choosing contrasting colors (such as blue with purple and blue with red as in links associated with the *ectotherm-vertebrate* region in Figure 66). Node and background colors are less important but still significant. Lighter colors are better for the background.

Determining the size of nodes is another issue. Additional information can be represented by node size via using 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 (i.e. a few high values) logarithmic transformation produces a more uniform

distribution. In the event log(X) doesn't produce a uniform distribution, users could try other transformations such as sqrt(X). In fact, the transformation used for the legal cases dataset was 5 + sqrt(X)/5 on the *inCites* attribute (indicating the number of citations to a case in the larger dataset of 2,345 cases), which produced good results, especially on the Supreme region.

6.2 Guidelines for Node Aggregation

To reduce complexity in dense networks, node aggregation in the context of semantic substrates is introduced. This addition, when combined with the filtering and details-on-demand functions enables users to detect patterns, gaps, outliers, and clusters in large datasets.

In this strategy, node aggregation is based on replacing all the nodes in a grid cell with a single metanode. Grid cells are the result of using the GridPlotXY placement method of NVSS (Aris 2007), where x- and y-axes are used and they each represent the values of a node attribute. Node aggregation is illustrated in the context of the GridPlotXY placement method in the TobIG case study (section 5.2.3); however, it could be generalized to other placement methods. Users can switch between the nodes and the metanodes modes. In the nodes mode, all nodes are displayed. In the metanodes mode, nodes in grid cells are aggregated into a single large metanode. The TobIG case study illustrates how these modes were useful in scaling up to explore much larger datasets than was possible without this approach. This broadly applicable strategy depends only on aggregating nodes with similar attribute values, avoiding costly clustering algorithms.

The following sections discuss how to apply node aggregation in the context of semantic substrates.

6.2.1 Simplified Exploration through Node Aggregation

Node aggregation helps to attain a more comprehensible display and also facilitates understanding by simplifying the display. When users select meaningful attribute values to group nodes, aggregated nodes become meaningful overviews of the groupings the user made. The simplicity makes the exploration effective and efficient. Facts stand out, especially surprising ones. The fact that the highly cited documents during 1990-1994 have not used the most popular keywords in the dataset in Figure 71 is an example. Another example is the fact that Steve Hecht wrote all types of documents in Figure 87.

Some substrates may be better than others in answering a specific question or in exploring a dataset from certain perspectives. Regarding the dataset in the TobIG case study (section 5.2.3), we were interested in the activity (for authors: writing papers, for documents: being cited, and for keywords: being used) and time aspects of nodes (authors, documents, and keywords) as well as the relationship between them. The attributes used in the substrates (the attribute *Five Years* and the unified attribute *CR_LCS_Count* representing *CR, LCS, and Count* in Figure 70) supported the exploration from the perspectives above.

6.2.2 Binning Attribute Values into Ranges

The dataset in section 5.2.3 looked from the perspective of 5-year periods and *CR, LCS,* and *Count* binned in a certain way. This is a good arrangement when users know and want to see the data in this way. In other words, it makes sense to users to look at the data in those specific 5-year periods and in the *CR, LCS,* and *Count* ranges that were used. For example, it is assumed that the local citation score (LCS) of a document does not make much difference within the range 5-9 to users; hence, the range of 5-9 is given a specific slot and separated from other ranges. Similarly, there is (or at least could be) a difference when LCS is 3 and 4; hence, the different slots were allotted.

There is a trade-off in how to bin values into ranges. The more bins, the more detailed information revealed, and the more effort needed in managing it (remembering and comparing them to each other). On the other hand, too few bins lead to a crude division, which lead to a shallow understanding. A balanced view is desired and can be attained by iterative substrate design (see also Aris et al. (Aris 2007)). In the example dataset, a 6-part binning for the documents and a 7-part binning for the authors were used. Figure 69 shows one of the earlier substrates on the same dataset. The latter binning arrangement arose after perusing the distribution of the data through a few iterations. A certain amount of time may be necessary to achieve a satisfactory result. This seems to depend on many factors, such as the complexity of the dataset, how much users know about the dataset, and how experienced they are in terms of having explored the dataset.

Certain tasks are better with certain substrates (and binning) than others. In the TobIG case study (section 5.2.3), the latter substrate performed well in terms of providing insights and understanding the data. The 6-7 bins on both axes facilitate to go over the different slots and get overviews quickly as well as compare them to one another. If deeper or other types of questions arise, substrates could be iteratively modified to look for deeper insights and more precise facts.

6.2.3 Details-on-Demand

Being able to switch between the metanodes and the nodes modes allows users to look at details-on-demand. This way, users get more information only when needed, which leads to a cleaner, and therefore a more comprehensible and efficient, process of exploration.

Details-on-demand have several benefits: They (1) enrich understanding due to the additional information, (2) help to check assumptions, and/or (3) prevent incorrect inferences and sometimes compensate for when the representation of the overview is misleading.

Examples for the above points are as follows:

(1) For Figure 78, switching to the nodes mode revealed that there are 10 authors writing 44 documents (Figure 85) in the 2000-2004 period and what their distribution is in terms of H-score.

(2) In Figure 83, it is assumed that it is harder to write top-documents in the 2000-2004 period, as it is a recent period. Switching to the nodes mode and comparing Figure 84 with Figure 85 supported this assumption. There were only 16 author-to-document links in the later 5-year period while there were 44 author-to

document links in earlier 5-year period. In addition, it is visible that there are more nodes (documents) in the earlier 5-year in the nodes mode.

(3) In Figure 86, looking at the nodes mode reveals that there is only one top document in the 2005-2007 period, which is written by only one author. This prevents treating this last period the same as (or close to) the previous ones as there is substantial difference.

6.3 Summary of the Design Guidelines

The summary of the design guidelines fall into two:

- *Guidelines to design effective semantic substrates*. The 5-item list in section 6.1 provides a good summary for this. These items are used to select a grouping attribute, to determine the placement methods, and miscellaneous issues.
- *Exploration guidelines using the node aggregation feature.* Through node aggregation, the network can be simplified (decreased number of nodes and links, see Figure 27). Then, users can filter to focus on the parts of the network and switch to the de-aggregated mode for details. The boundary values chosen for binning determine the groups in the aggregated view. Therefore, if users intend to use the aggregated view, they are encouraged to set the boundary values so that the metanodes in the aggregated view will represent meaningful groups (for the exploration tasks).

Chapter 7: Future Work & Conclusions

The following sections list future work (section 7.1) and end with conclusions (section 7.2).

7.1 Future Work

By engaging the remarkable human capabilities for spatial perception and analysis, semantic substrates enable users to control the layout of the network visualization. This opens the way to more comprehensible displays to support a variety of user tasks. The case studies demonstrated benefits for domain experts, but more needs to be done to refine and extend the implementation and features. The following sections address future work by area of application.

7.1.1 User Interface Design Issues

As the number of regions in the substrate increases, the complexity of the display and the control panel increase. Future work includes how to simplify these displays or find strategies to make them manageable and comprehensible.

The number of checkboxes for link filters grows quadratically with the number of regions. Possible solutions include a selection mechanism to define which filters to keep on the control panel and replacing the checkboxes with an iconic representation that succinctly represents regions and interregional link connections.

The NVSS implementation could be improved in terms of user interface features to simplify or accelerate region parameters, such as size, location, color,

labels, and node layout strategy. In addition, flexibility could be added or improved for node, link, and label properties such as placement, size, color, font, and background. Being able to specify the node colors by attribute would help for certain datasets and tasks. Node and link visibility could be made dynamic and enhanced by tooltips, excentric labels (Fekete 1999), and window panes containing their textual representation. Additional filters for nodes and links, and perhaps widgets for various visual interactions could be designed and included.

Furthermore, elastic window strategies could be implemented that would enable users to enlarge one region while shrinking the others in a smooth animation (Kandogan 1998).

7.1.2 Substrate Design Issues

The process of substrate design could benefit from tools, modules, or features that 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 the substrates could easily be shared among many users. A meaningful substrate captures domain knowledge and enables easy comparison of datasets, identification of attribute value changes, new nodes, and links.

In terms of substrate design issues, future work falls into the following categories: (A) the visual presentation, (B) facilitating the substrate creation process, and (C) miscellaneous issues.

(A) The Visual Presentation: 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 Figure 46. Nodes with similar attribute values are placed close to each other. The trade-off is between meaningful node locations and perceivable link display. A specific form of link overlap is with links concentrated within a small space. This usually happens when the source and destination links are packed closely together as in GridPlotXY placement method cells, such as in the invertebrate region in Figure 66. Another challenge is to clearly display the links between regions, especially when there is another region between two regions that are connected with links. Possible solutions include link routing (Phan 2005) and link clustering such as using hierarchy to organize edges (Holten 2006).

(B) Facilitating the Substrate Creation Process: 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. To create a good substrate, an iterative substrate design process (using a trial-and-error approach) could be used. In addition, familiarity with the dataset and the domain is helpful. Substrates could be stored and reused for similar datasets. A module that helps users store substrates, calculates compatibility between a substrate and a dataset, 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. To help users decide about which attributes to use for grouping or placement method within a region, providing the range or distribution of attribute

values for the associated region can eliminate several iterations. In addition, previewing nodes and links within a region and links between regions might help.

In some datasets, such as in the initial post78 (section 5.2.1.1), the collaborators knew beforehand which attributes were important, and therefore could be used to determine effective placement. In general, however, there may be many attributes that users have little awareness of. In such cases, they will not know which attributes are best to use. A user interface to help users explore combinations of attributes could lead to better designs faster.

(C) Miscellaneous Issues: Opportunities for improvement include more expressive region specifications (allowing the use of operators other than equality to a single attribute value (e.g., *venue* = ”Supreme”), and in general supporting a Boolean expression with a complete set of operators), a way to select filters (to make available during exploration), and different types of node placement methods 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 datasets) and creating derived attributes.

7.1.3 Scalability Issues

In terms of scalability issues, one possible future work is to assist users in determining boundary values to group attribute values into bins. One way to do this would be to implement a visual module that shows the distribution of attribute values and suggests binning intervals to achieve a balanced distribution of nodes and links. Furthermore, it would be beneficial to allow users to adjust the suggested bin intervals to facilitate the bin creation for many custom bin intervals users would like

to have. A sophisticated interface would be capable of providing several alternatives by conveying to the user the trade-offs in each alternative. This way users could make informed decisions (for a similar idea in forecasting time-series interfaces, see (Buono 2007)).

Explorating data usually involves filtering and narrowing down to an interesting subset. Node aggregation provides overviews for improved understanding. Assuming that two links, A and B, are pointing to an aggregated node, there are situations that only link A is visible due to filtering. Currently, the aggregated node remains the same. It has the meaning "some nodes that the aggregated node represents are linked." However, it would be helpful to give information about the nodes that are linked. One way to do this is to have two aggregated nodes instead of one, one that represents the linked nodes and the other the rest. This way, users can set the size coding for the aggregated nodes in the Substrate Designer to represent any attributes that they want to appear in the visualization.

Metalinks in the aggregated view could be improved by size, color, or texture-coding to indicate the number of links they represent.

One application-level improvement would be scalability in terms of response time as the numbers of nodes and links increase. For networks with millions of nodes, dynamic queries could be made more efficient (in terms of response time) to limit node visibility while preserving comprehensibility.

7.1.4 Interactive Exploration Issues

Currently, filters apply to all nodes or a subset of nodes within regions. Support for a dynamically selected set of nodes would enable users to do cascaded

types of exploration. In other words, after applying filters to a set of source nodes users will be able to select the nodes that links point to. Then, users will be able to use those selected nodes as source nodes, apply filters, and produce another set of destination nodes. This will enable users to arrive at more complex meaningful subsets of the data. For example, in Figure 76, if the documents could be selected to see the set of keywords these documents use, this would have been a cascaded exploration. The exploration could continue by finding the authors that used those keywords. As such cascaded explorations get longer in the number of steps, it becomes harder to keep track of the meaning of the selected subset. A visual representation that reminds users of the meaning would remedy this problem. A history mechanism would enable users to undo steps in their cascaded exploration and choose other paths (Shrinivasan 2007). Being able to define more than one dynamic selected set of nodes may expand the types of explorations users could do.

Another type of improvement would be the capability to compare two (or more) link patterns that resulted from different paths of exploration. Currently, this can be achieved by having two instances of the visualization side by side. With additional features, there may be benefits to show more than one exploration view in the same application.

In certain user tasks, the complete information that the current link display provides (for every link: source node, destination node, direction of link) may not be necessary. In such cases, link rendering can be replaced via representations that are simpler and more scalable to facilitate comprehension.

7.1.5 Application Level and Other Issues

Further future work includes supporting and representing link attributes, different node types (each node type having a different set of attributes), and allowing multiple-valued attributes.

Non-rectangular or overlapping regions might help in some datasets. Zooming and panning features could help to focus on certain parts of the visualization. Allowing dynamic changes to bin limits and placement attributes could facilitate exploration of data. Supporting OR logic between filters would enable users to represent more sophisticated queries. Customization of the node details feature would help with certain datasets and tasks; for example, a web page (the link of which is defined by an attribute) could be opened when a node is clicked.

Networks changing over time could be supported and other data sources, such as external databases, could be added. An export facility for subsets of data that are arrived at through exploration would facilitate external analysis. Being able to store and retrieve a path of exploration would enable users to reproduce and communicate their explorations to others. Support could be added for undirected networks. Additional case studies with the semantic substrate approach in other domains would increase the reliability of current results and provide more feedback for further improvements (Shneiderman 2006).

7.2 Conclusions

While all these challenges remain, attractive new possibilities emerge for network visualization based on semantic substrates. User-defined regions create some

new problems, such as restricting some of the links into smaller spaces, but have proved to be beneficial in several application domains.

Among the many previously existing network visualizations, some contain the elements of the semantic substrate approach. However, none of them had a complete approach to enable users to specify the network layout using node attributes in a systematic and flexible way. They either expected domain-specific datasets or limited the ability to define layouts using node attributes. One advantage of the semantic substrate approach is being able to look at the data in many different ways through the use of various node attributes. Another advantage is that it is applicable to network datasets from different domains, which is not supported by some of the visualizations that contain the elements of the semantic substrate approach. The implementation and case studies provided evidence for this applicability and its benefits.

Contributions are provided in detail in section 1.3 and summarized here as follows:

- *The definition of the semantic substrate idea:* The 3-step approach to place nodes visually using node attributes.
- *The technical structure and the visual & interactive design of a semantic substrate:* The definition (contents) of a semantic substrate, its technical design and visual design.
- *The user design process of semantic substrates and guidelines:* How to design a semantic substrate and guidelines for good substrate design practices, which resulted from case studies.

- *Scalability in the context of semantic substrates:* The node aggregation extension to enable or facilitate the exploration of larger datasets.
- *Implementation (the NVSS application):* The implementation of NVSS both shows how the semantic substrate approach is applied and provides evidence for its utility.
- *Case study results:* Case studies show the types of benefits and insights users could gain by using the semantic substrate approach.

In conclusion, semantic substrates enable users to specify the network layout in terms of node attributes. With the addition of filters based on placement attributes, users can explore the data in terms of the attributes they selected. This leads to increased user control, which may lead to better understanding and deeper insights. Furthermore, node aggregation enhances the benefits of using semantic substrates for complex networks by enabling a simpler exploration of networks. This approach could be applied to many domains including citation datasets, food-webs, and the rising field of social networks.

Appendices

The following sections are appendices. Appendix A describes the process of creating a semantic substrate in NVSS. Appendix B provides software engineering metrics on the NVSS implementation. Appendix C includes the letter from IRB that shows the evaluation method in this dissertation did not require an IRB approval. Appendix D includes the TobIG case study document (A summary of section 7 is provided in section 5.2.3 for the TobIG case study).

A. The process of creating a substrate in NVSS

NVSS consists of two modules: the Substrate Designer and the NVSS Visualization Module. The substrate designer module enables users to create a semantic substrate, while the visual exploration module applies a semantic substrate to a dataset, and visualizes the module to be explored by users.

First, users prepare their dataset in terms of nodes and links files. The nodes file is a TAB delimited file, which could be opened with a spreadsheet program, such as Microsoft Excel (Figure 109). The first column is the unique key for nodes, while the rest of the columns are the rest of the attributes for that node. Each row in the nodes file represents a node except that the first row contains attribute names and the second row contains the attribute types, which are INTEGER, DOUBLE, STRING, and DATE.

	A	B	C	D	E	F	G
1	caseId	cite	venue	venue2	circuitNo	date	year
2	INTEGER	STRING	STRING	STRING	INTEGER	DATE	INTEGER
3	4003	94 S.Ct. 2291	Supreme	NULL	0	6/10/1974	1974
4	4004	54 S.Ct. 599	Supreme	NULL	0	4/2/1934	1934
5	4005	1930 WL 1063	District	Territory of	9	4/4/1930	1930
6	4006	55 S.Ct. 333	Supreme	NULL	0	1/14/1935	1935
7	4007	565 F.2d 338	Appeals	Fifth Circu	5	12/27/1977	1977

Figure 109 Example nodes file to be used as input to NVSS.

The links file (Figure 110) uses the key attribute of nodes (the first column in the nodes file) to represent the links. The first two rows are similar to the nodes file in that the first two columns in the first row contain modified names of the key attribute (that was in the nodes file) and the second row contains the type. These rows are only for users and are not used by NVSS. Data beginning at the third column represent link attributes. Although they are currently ignored by NVSS, the format allows the columns to be present for future use in NVSS.

	A	B	C
1	caseId1	caseId2	dateDiff
2	INTEGER	INTEGER	DOUBLE
3	4003	4004	40.21644
4	4003	4291	53.39178
5	4004	4291	13.17534
6	4005	4291	9.178082
7	4006	4004	0.786301
8	4006	4291	13.96164

Figure 110 Example links file to be used as input to NVSS.

Once the nodes and the links files are ready, the next step is to generate a substrate. Once the substrate is created, it can be applied to the dataset to visualize it in the NVSS Visualization Module.

In order to create a substrate, users will need a Data Model file that contains the node attribute data. Users can automatically generate the Data Model file by using

the nodes file. Since the Data Model file is a text file, it can be edited manually for adjustments (Figure 111).

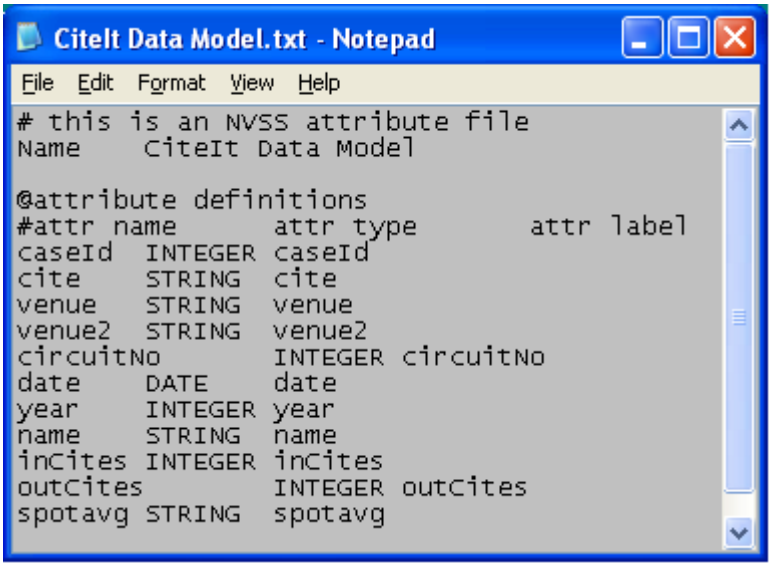


Figure 111 A sample Data Model file.

When the Data Model file is ready, users can start creating a substrate by clicking the "New" button in NVSS Main (Figure 112). NVSS Main will require the location of the data model file to load it. The Data Model becomes an integral part of the substrate, which is visible on the left hand side of the Substrate Designer (see Figure 113, first line, where it says "Data Model" and "CiteIt Data Model." In this case, the name of the data model is "CiteIt Data Model", which is specified in the Data Model file.).

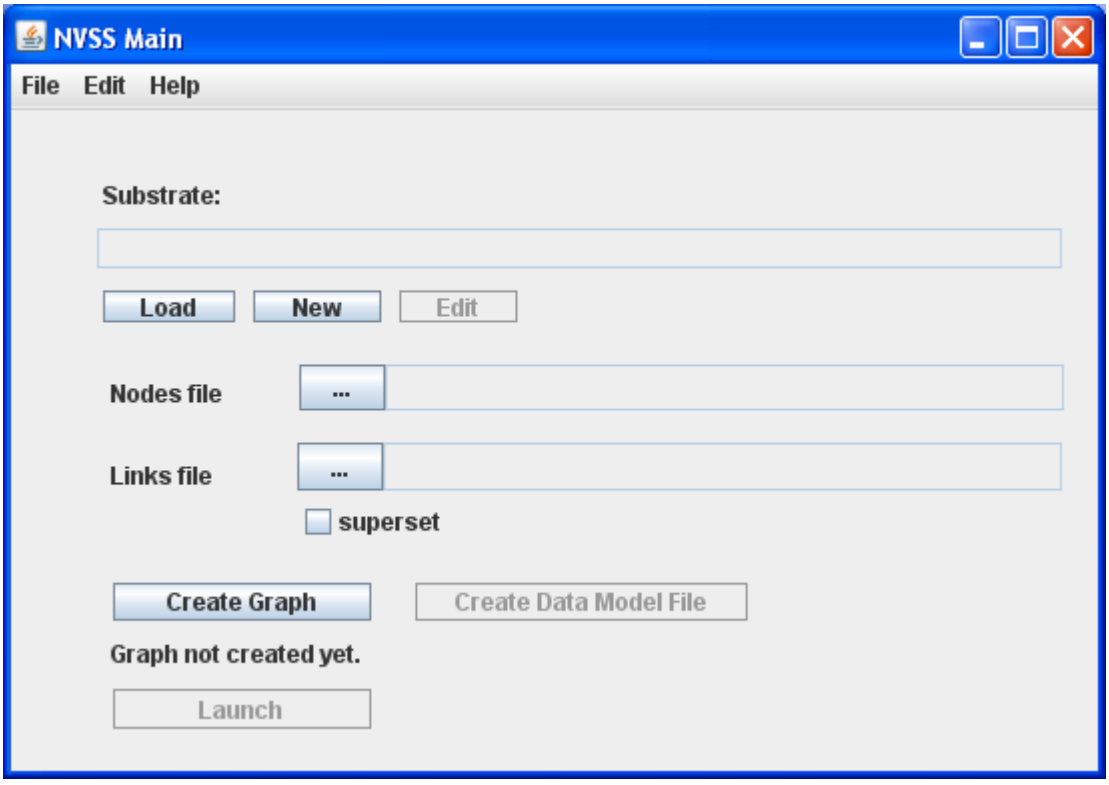


Figure 112 NVSS Main.

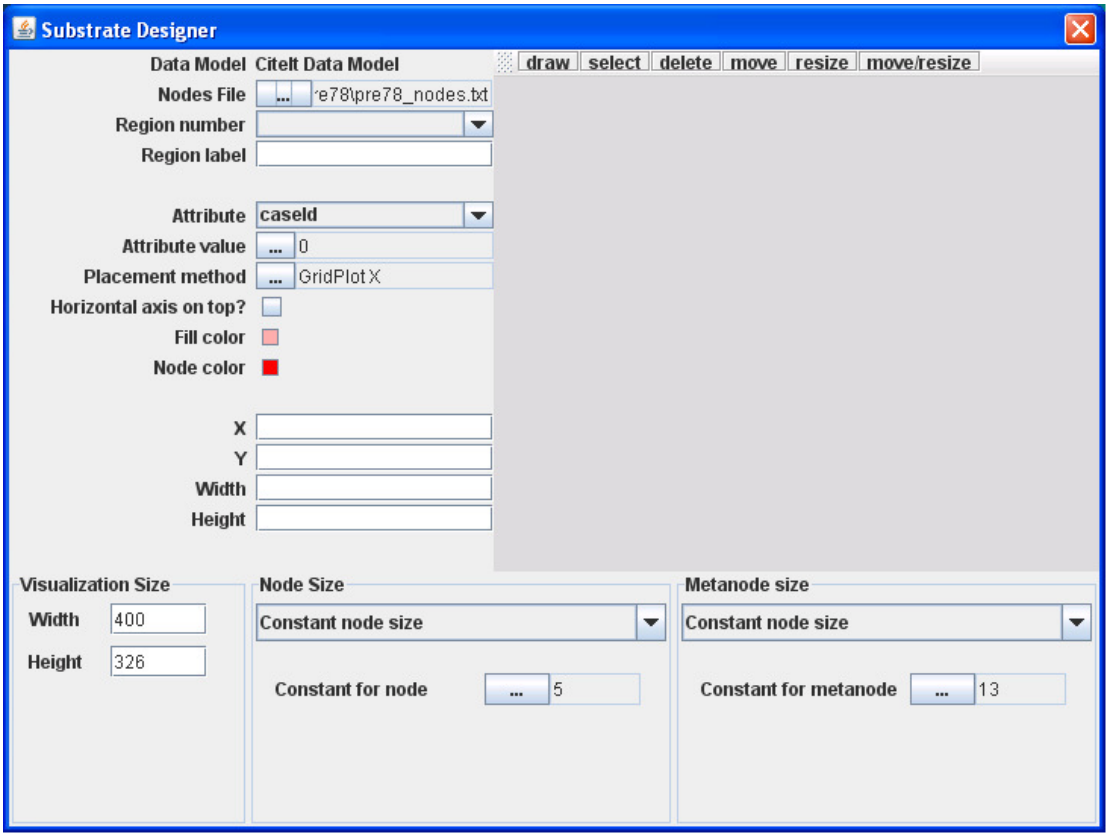


Figure 113 New substrate after pressing "New" and loading the Data Model from a file.

The Substrate Designer (Figure 113) consists of 3 panels. The panel on the top right is where users can visually define regions and their location and size properties. The buttons on the top allows users to change the “mode” of operation. To create a region, users press the “draw” button. Once in the “draw” mode, the first click of the mouse defines the upper left corner of the region and the second click after a drag defines the bottom right corner (Figure 114). To alter a mode, users simply click another button (these buttons have the same behavior as radio buttons or tabs) at the top.

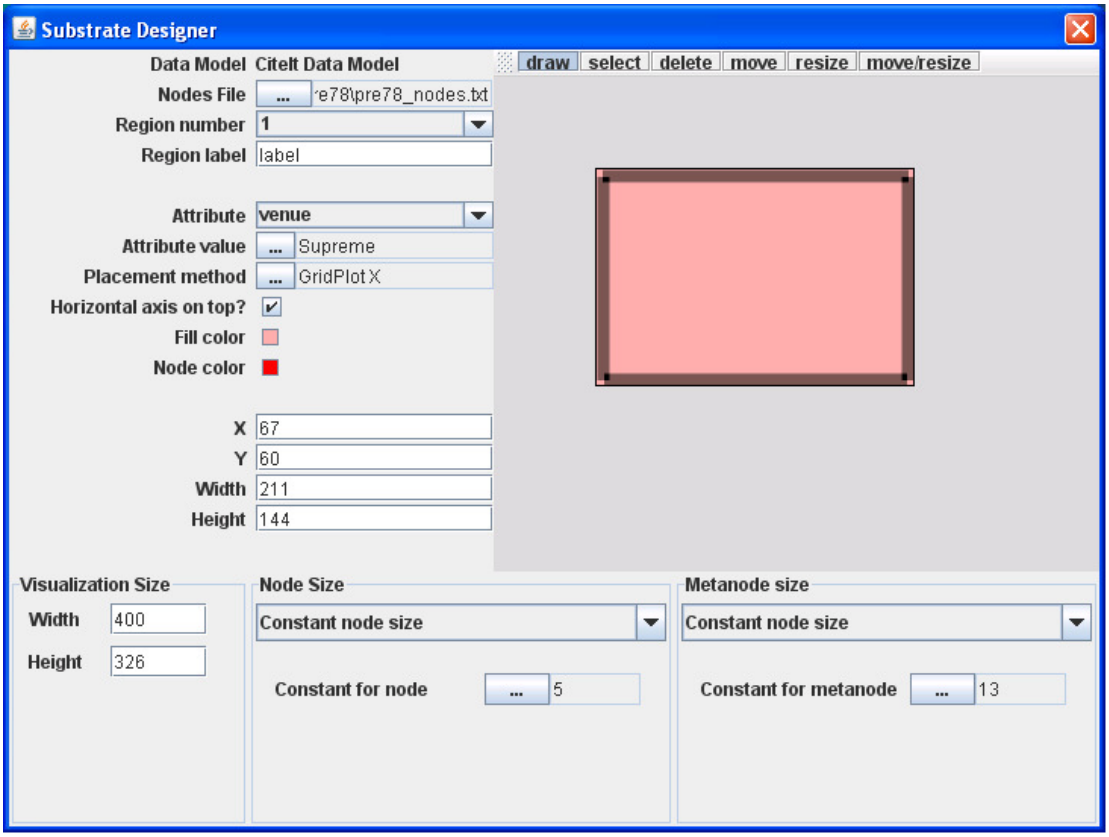


Figure 114 Creating a region in "draw" mode in the Substrate Designer.

When a region is created, its visual properties are assigned in the details view (Figure 114). These are X and Y, which determine its location, and Width and Height, which determine its size. The default fill color and node color are used once the region is created. To modify colors, users click on the color and select the desired color from a new window providing color choices.

The newly created region is assigned a region number (in this case, it is “1”) and its default region label is “label” (Figure 114). Users type a new label to override the default region label. The label of a region appears at the top centered in the visualization (not displayed in the Substrate Designer).

"Attribute" on the left panel determines which attribute will be used to place nodes within this region. In Figure 114, the "venue" attribute is selected. The "Attribute value" determines the attribute value that the nodes within this region will have for the selected attribute. In this case, it is selected to be "Supreme." As a result, nodes having attribute value "Supreme" for the "venue" attribute will be placed within this region.

To set the placement method within a region, users need to press the "…" button next to "Placement method", which opens the Placement Method Selector dialog (Figure 115). NVSS supports five algorithms:

- GridPlot X
- GridPlot X Jittered
- GridPlot Y
- GridPlot Y Jittered
- GridPlotXY

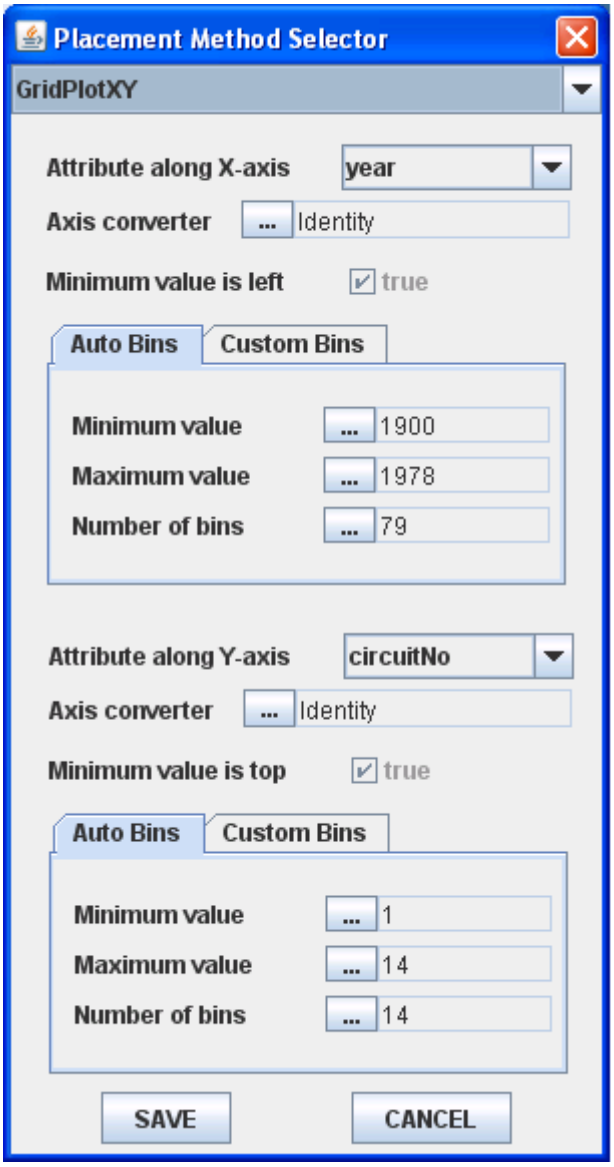


Figure 115 Placement Method Selector launched from the Substrate Designer.

The first four algorithms affect only one axis, where they leave the other axis free or introduce jitter. For example, `GridPlot X` will define the x-axis in terms of an attribute. A node's x-coordinate will be determined by this x-axis setting. The node's y-axis will be arbitrary in the sense that it is not derived from an attribute (specifically, nodes will be evenly spaced on the y-axis). `GridPlot X Jittered` is similar to `GridPlot X` except that nodes are jittered along the y-axis (moved up or

down, alternating, with no change in the vertical distance between nodes that are aligned vertically with each other).

After selecting a placement algorithm, users need to define settings for each axis. First they choose an attribute along that axis, and then they choose a binning strategy: “Auto Bins” and “Custom Bins”.

In the “Auto Bins” approach, users set the `Minimum value`, the `Maximum value`, and the `Number of bins` (Figure 115). By default, the minimum and maximum values are set according to at the data file provided ("Nodes file" in Figure 114) and the "Attribute Value" set for the region. Users may need to override these values, especially if they want a common axis between two or more regions. For the “Auto Bins”, the minimum value is the left (smallest) value in the first bin and the maximum value is the right (largest) value in the last bin. The number of bins determines how many bins there will be between the minimum and the maximum values (these values are inclusive as just described). For instance, with a minimum value of 1, maximum value of 100, and 10 bins, the left value in the first bin will be 1, the right value in the tenth (last) bin will be 100. The first bin will contain the values 1..10, the second bin will contain 11..20, and so on.

In the “Custom Bins” approach, users provide the boundary values for bins in comma-separated form. The boundary values are the left values of all bins in increasing order and the right value of the last bin. For instance, with boundary values “1,5,7,10”, there will be three bins, where the first bin will contain values 1..4, the second bin will contain 5..6, and the last bin will contain 7..10.

For each placement attribute, an axis converter is used, where the default one is the identity axis converter (see Figure 115, "Axis converter"). The axis converter is used when more meaningful values could be presented to the user. For example, if 5-year periods are grouped and each group is represented by a number, these numbers will not be meaningful to users. Users can map these numbers to meaningful STRING values by using an AVC (Attribute Value Converter). An AVC defines the mapping. Currently, the only way to define a mapping for users is via a file as in Figure 116. An AVC file is a TAB delimited text file. Users can create this file first in a spreadsheet program, such as Microsoft Excel, and then save it as a TAB delimited text file. The resulting file is loaded by clicking the "…" button next to "Axis converter" (Figure 115).

	A	B
1	FIVE_YEARS	5YR-RANGE
2	INTEGER	STRING
3	0	80-84
4	1	85-89
5	2	90-94
6	3	95-99
7	4	00-04
8	5	05-07

Figure 116 Example AVC file, where simple integers (group numbers) are converted to 5-year ranges.

Although all algorithms use x- and/or y-axes, this is theoretically not necessary. In principle, any type of placement method could be used. NVSS is designed in a modular way to facilitate addition of new placement methods.

Since all algorithms supported by NVSS currently have axes, there is a feature to show the x-axis on top or bottom. (This feature could be moved into the algorithm in the future if other algorithms that do not have axes are added to NVSS.) Checking

“Horizontal axis on top?” causes the horizontal axis to show on the top of the region when checked. This is useful when there are many links that go through the bottom of the region occluding the x-axis labels. Putting it on the top relieves this problem.

The Substrate Designer also allows for node size and metanode size coding. Node size coding is useful in the nodes mode and metanode size coding is useful in the metanodes mode, when nodes are aggregated to metanodes. Currently, NVSS allows users to specify the size as constant or in terms of the value of a node attribute. A transformation function can be applied along with scaling and translation, which is in the form of the formula y = m * f(x) + n, where x is the attribute value to be transformed, f(x) is the transformation function, m is the scale, n is the intercept, and y is the transformed value, which will be used directly to determine the size of the node. In Figure 117, the *inCites* attribute is used to determine node size, while a transformation y = 0.2 * sqrt(X) + 5 is applied. Currently, the only transformation function available is sqrt(X); however, the modular approach in NVSS will allow adding other transformation functions if needed in the future (new code will need to be written; but it will be possible to add to the existing code base without affecting other parts of the code). The metanode size coding has an additional option for size coding, namely using the “number of aggregated nodes.” This is the number of nodes a metanode represents. The transformation functionality is provided at this option, as well.

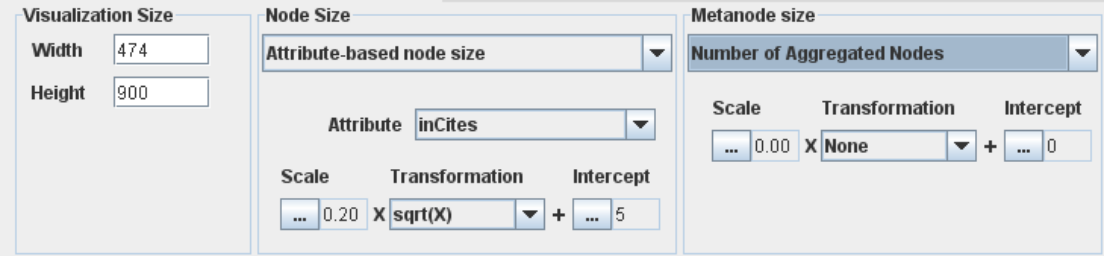


Figure 117 Bottom panel of the NVSS Substrate Designer.

Although the Substrate Designer doesn't allow users to specify the link colors, they can specify them in the NVSS Visualization Module (users can use the substrate with a dataset to launch the NVSS Visualization Module, modify the link colors they would like to modify, and finally save the substrate via the menu File->Save Substrate).

Figure 118 illustrates a completed substrate, where the top region will have Supreme Court cases, the middle region will have Circuit Court cases, and the bottom region will have District Court cases. All three regions use the Year attribute to define their x-axis with the same binning parameters, while the Circuit and District regions also use the circuitNo attribute at their y-axis (and they use the same binning parameters). While the Supreme region uses a GridPlotX Jittered placement method, the other two regions use GridPlotXY as a placement method.

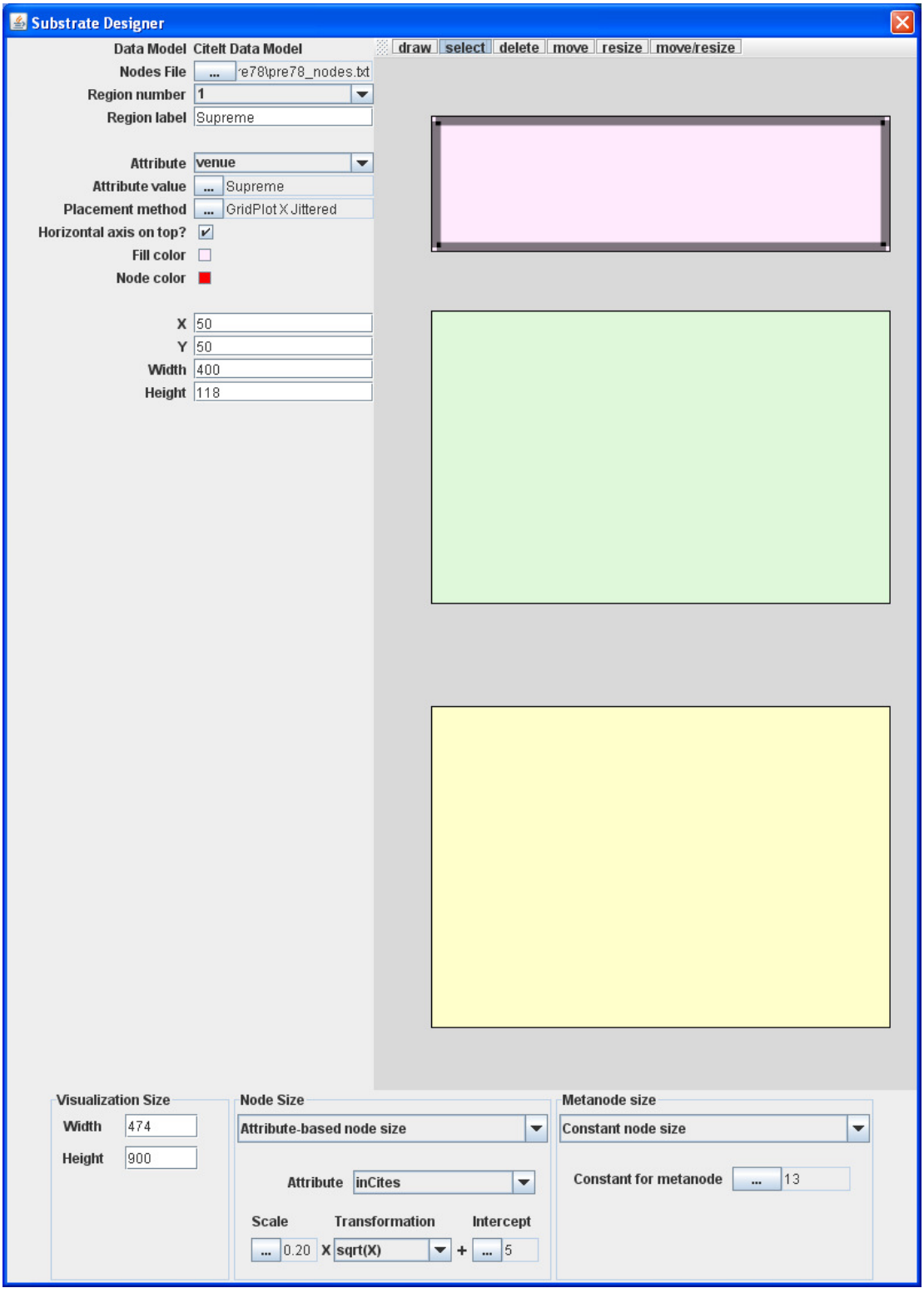


Figure 118 The completed substrate.

Closing the Substrate Designer window will prompt the user to save it as a file. Once it is saved, the focus returns to NVSS Main, where the substrate file is instantiated.

Figure 119 shows the status of NVSS Main window after the substrate is created, and the user has specified the nodes and links files. These are the 3 elements that are needed by the NVSS Visualization Module. Once these are available, users can click the "Create Graph" button, which will internally create the graph. At this stage, any incompatibilities between the nodes & links file and between the nodes and the substrate will be reported, which helps users identify errors and correct them. The "superset" checkbox enables users to use the subset of the nodes file without changing the links file. This is a great convenience as it eliminates the step for users having to externally filter the links according to the nodes file to create the subset of the links file. When unchecked, NVSS requires the nodes file to contain all the nodes that are specified in the links file.

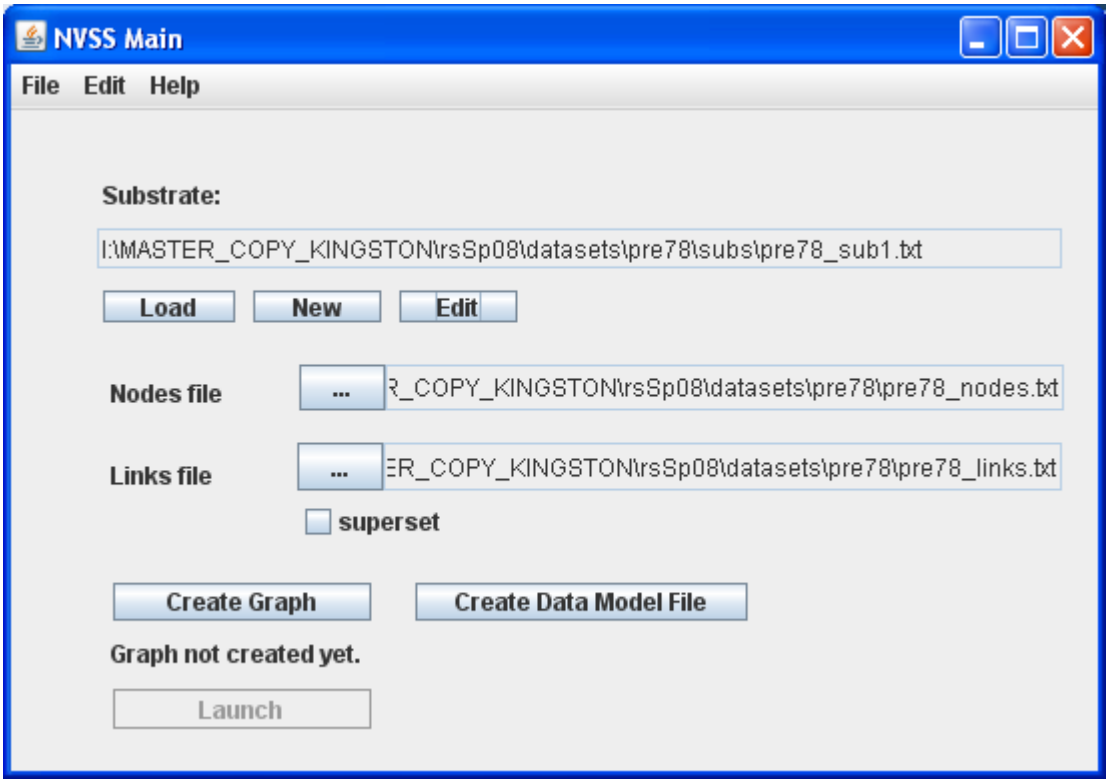


Figure 119 After creating a substrate and selecting nodes and links files.

There is a status line below the "Create Graph" button that will report the completion of the creation of the graph. At the same time, the "Launch" button will be enabled (Figure 120).

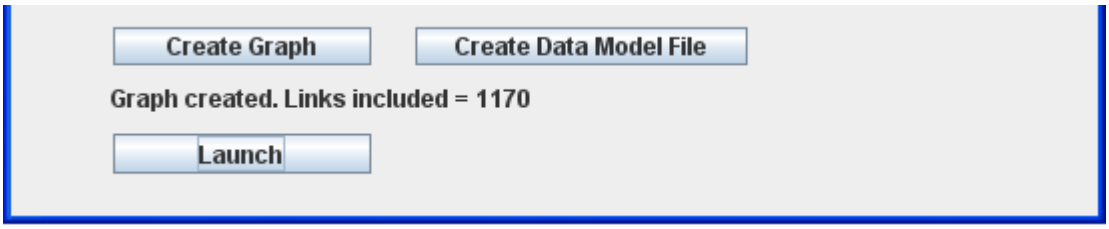


Figure 120 The status message confirms the creation of the graph after users press "Create Graph." In addition, it informs how many of the links are used just in case users have pressed the "superset" checkbox for the links file.

Once the user presses "Launch", the NVSS Visualization Module will launch with this data specified (Figure 121).

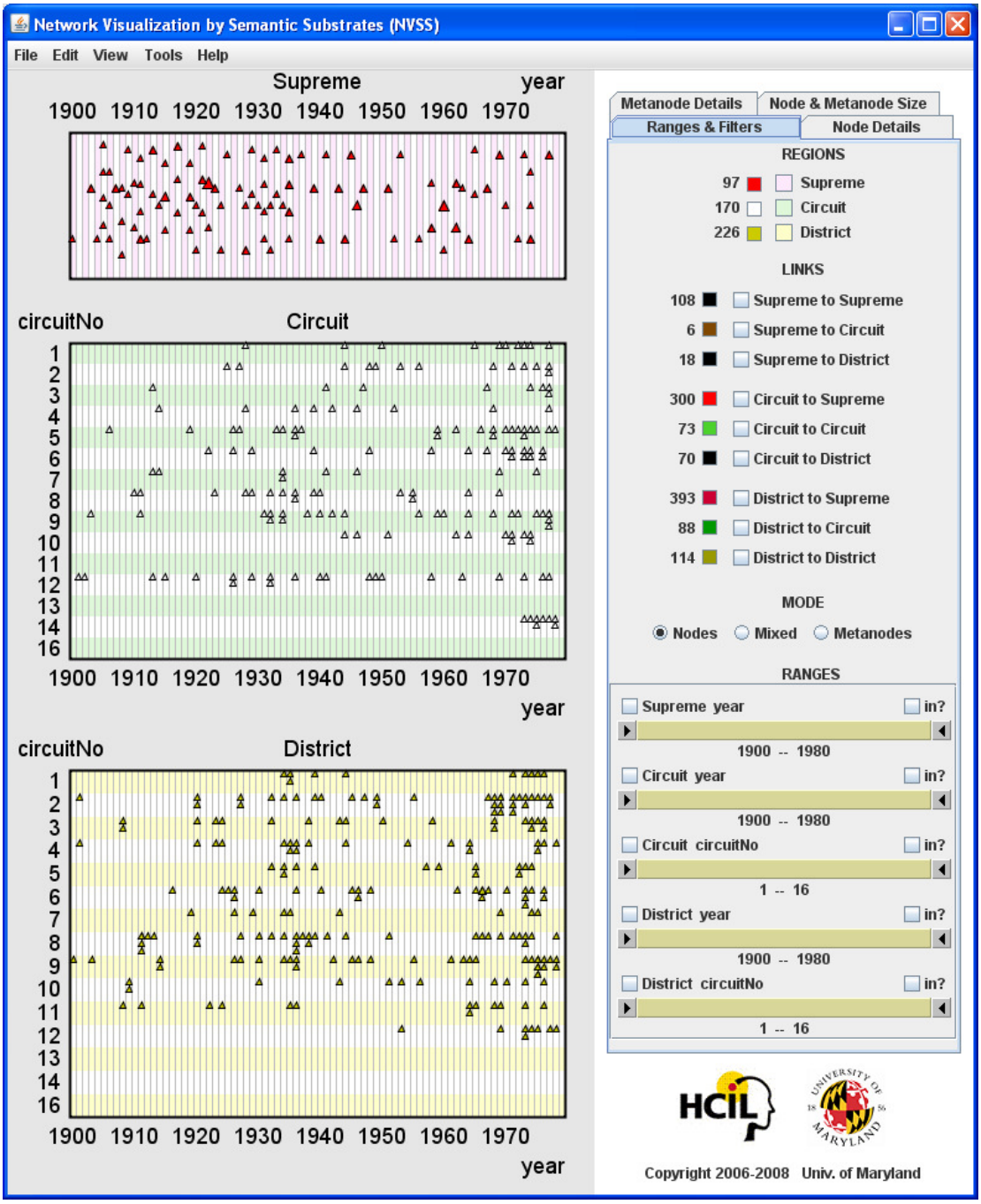


Figure 121 NVSS Visualization Module after the user has pressed the "Launch" button.

B. Software engineering metrics of NVSS

Some metrics provided by the Metrics plug-in in Eclipse 3.2 are as follows (Table 8). The metrics are based on NVSS version 2-4-1, April 23, 2008. Note that some of the maximums are caused by contributed code.

Table 8 Software engineering metrics on the NVSS implementation.

Metric	Total	Mean	Std. Dev.	Max	Resource causing Maximum / method AND/OR (comments)
Total Lines of Code	13,244				(The contributed part is 2036 making the non-contributed part 11,208.)
Number of overridden methods	60	0.42	0.831	3	GridPlotYJitteredRRA.java
Number of Attributes (avg/max per type)	537	3.755	8.021	67	SubstrateDesigner.java
Number of Children (avg/max per type)	41	0.287	0.85	5	AttrType.java
Number of Classes (avg/max per package Fragment)	143	13	6.633	26	edu.umd.cs.sg.ui (package name)
Method Lines of Code (avg/max per method)	7,990	6.402	9.823	118	SubstrateDesigner.java / custom_init
Number of Methods (avg/max per type)	1142	7.986	9.909	69	SubstrateDesigner.java
Nested Block Depth*		1.471	0.817	6	RectRegion.java / getNumEdgesTo
Depth of Inheritance Tree (avg/max per		2.545	1.733	7	SGMetaNode.java (most of the inheritance path remains in the JUNG

type)					package)
Number of Packages	11				(2 of the packages are not effectively used. One of them is recycled code (recycle), the other is used for debugging purposes (debug))
Afferent coupling (avg/max per package Fragment)		19.455	17.095	46	edu.umd.cs.sg.debug (note that this package is not one of the 9)
Number of Interfaces (avg/max per package fragment)	14	1.273	1.135	3	edu.umd.cs.sg.altviz.algo (this package contains the placement methods)
McCabe Cyclomatic Complexity (avg/max per method)*		1.716	1.501	21	EfileChooser.java / accept (note that this class is contributed)
Instability (avg/max per package fragment)		0.376	0.301	1	edu.umd.cs.sg.recycle (note that this package is not one of the 9)
Number of Parameters (avg/max per method)*		0.805	1.105	11	RectRegionSettings.java /RectRegionSettings (constructor)
Lack of cohesion of methods (avg/max per type)		0.314	0.387	1.417	EfileChooser.java (note that this class is contributed)
Efferent coupling (avg/max per package fragment)		9.182	7.964	25	edu.umd.cs.sg.ui
Number of static methods (avg/max per type)	106	0.741	2.17	19	SGUtils.java
Normalized Distance (avg/max per package		0.55	0.271	0.979	edu.umd.cs.sg.debug

fragment)					
Abstractness (avg/max per package fragment)		0.152	0.131	0.429	edu.umd.cs.sg.recycle (note that this package is not one of the 9)
Specialization index (avg/max per type)		0.23	0.61	4	DoubleSlider.java (note that this class is contributed)
Weighted methods per class (avg/max per type)	2142	14.979	19.143	136	SubstrateDesigner.java
Number of Static Attributes (avg/max per type)	315	2.203	3.334	19	SettingsRayner.java

C. IRB determination of IRB approval

The following figure is a formal letter and shows that the evaluation method in this dissertation did not require an IRB approval from the IRB office of the University of Maryland, College park (Figure 122).

UNIVERSITY OF MARYLAND

INSTITUTIONAL REVIEW BOARD

2100 Lee Building
College Park, Maryland 20742-5121
301.405.4212 TEL 301.314.1475 FAX
irb@deans.umd.edu
www.umresearch.umd.edu/IRB

Notice: Review of Request for Determination of Non-Human Subject or Non-Research IRB Form

Date: June 3, 2008

To: Dr. Ben Shneiderman
Department of Computer Science

From: Roslyn Edson, M.S., CIP
IRB Manager
University of Maryland, College Park

Re: *Request for Determination of Non-Human Subject or Non-Research IRB Form #08-NHS-0031*
Project Title: *Visualizing & Exploring Networks using Semantic Substrates*

The *Request for Determination of Non-Human Subject or Non-Research Form* for the above-cited project was reviewed. The IRB determined that your project does not involve human subjects as defined in the Federal regulations. Therefore, your research does not require IRB review and approval. Please contact the IRB Office at 301-405-0678 if you have any IRB-related questions or concerns.

Figure 122 Letter from the IRB office of the University of Maryland, College Park, which shows that the evaluation method in this dissertation does not require an IRB approval.

D. The TobIG Case Study Document

Participants

The case study participants that we have been in close contact with are Steven Harper and Noshir Contractor. I (Aleks Aris) collaborated with them since the beginning of the summer 2007. During the summer of 2007, Steven Harper and Noshir Contractor both were at the same team at the University of Illinois at Urbana-Champaign (UIUC). Other team members that were included in some email communications are Andy Don, Nat Bukley, and Hank Green. During Fall 2007, Steven Harper relocated to become an Assistant Professor in the Management Program of James Madison University[6]. During the same time, Noshir Contractor moved to Northwestern University[7]. His main title is Jane & William White Professor of Behavioral Science.

Noshir Contractor reported that they have used NetDraw, Pajek, as well as a suite of Java and PREFUSE visualization tools they have developed in their lab. These tools were used to help distinguish different types of links (by color and/or thickness) or different types of nodes (by color, size or shape). They used these to help interpret what their analytics were telling them about key roles played by individual nodes in the network as well as overall global properties of the network (density, centralization, for instance).

Noshir Contractor is interested in the TobIG dataset as part of one of his funded projects. He is collaborating to see how NVSS can be useful and at the same time trying to understand how he can be a mediator between domain experts and NVSS. He also has a network of connections where he can get in touch with domain experts.

Steven Harper has been a valuable close collaborator to Noshir Contractor and team member at the UIUC team. I (Aleks Aris) communicated with him many times through email and a few times over the phone to resolve dataset issues (understanding attributes, transformation of the dataset to a form that fits NVSS format, spotting errors and correcting them, handling missing values, etc.). He has also contributed in valuable ways in terms of user needs, tasks, get involved in brainstorming and gave feedback to us for the existing state of NVSS.

Dataset Description

The TobIG dataset is a set of scholarly papers written by Tobacco researchers. This dataset is also used in the CI-KNOW[8] tool that is developed at the NCSA research center at UIUC.

The dataset has 3 different nodes: Authors, Documents, and Keywords.

---

[6] http://www.jmu.edu/management/harper.shtml
[7] http://nosh.northwestern.edu/
[8] http://sonicserver.ncsa.uiuc.edu:9061/ciknow4tobig/ncsa.sonic.ciknow.WebApp/WebApp.html

Each of these nodes has different attributes. NVSS expects all nodes to have the same set of node attributes. For this reason, Steven Harper has preprocessing the nodes to unify them with common attributes. The core unified node attributes are:

- ID: a field that is generated so that each node has a key field (an integer value).
- TYPE: The type of node. Values are: Author, Document, Keyword.
- NAME: The name of the document, author or keyword.
- YEAR: The year of the document, the keyword that first appeared, or the first year the author wrote a document in this dataset
- CR_LCS_Count: A unified field for each three types of nodes.
    - For Authors: CR: Citations Received. This is the author's H-score, explained better here[9].
    - For Documents: LCS: Local citation score. The number of times this document was cited by other documents within this dataset.
    - For Keywords: Count: The number of times this keyword appears in documents in this dataset.
- SQRT_Count: SQRT(CR_LCS_Count)
    - This is a derived attribute. I (Aleks Aris) generated it in Excel and used in sub13 and sub 14. It has the effect of providing fewer bins along the y-axis.
- Custom_Count: Custom_function(CR_LCS_Count)
    - This is a derived attribute. I (Aleks Aris) generated it in Excel using 3 different nested =IF formulas (max bins supported by Excel: 7), one for each type of node:
        - For Authors: 0,1,2,3,4,5-9, 10-26, 27-(27)
        - For Documents: 0,1,2,3-9, 10-29, 30-(99)
        - For Keywords: 0,1,2,3-9,10-29,30-99, 100-(175)

The links in this dataset represent different things according to the nodes they connect. The links are directional and their meaning are as follows:

- Author -> Document: Author writes Document.
    - Authors in this dataset are pretty prolific. There is one author that wrote around 550 documents (email communication with Steve).
- Document -> Document: Document cites another document.
- Document -> Keyword: Document contains keyword.

Regions are defined using TYPE having values "Author," "Document," and "Keyword." GridPlotXY algorithm is used in each region with YEAR on the x-axis (1980-2007) and CR-LCS-Count or one of its derived attribute on the y-axis.

[9] http://dx.doi.org/10.1073/pnas.0507655102

The counts of nodes and links in the dataset are as follows:

Node	**Count**
Author	29
Document	1,700
Keyword	2,567
Total	2,969

Link	**Count**
Author -> Document	1,770
Document -> Document	4,966
Document -> Keyword	9,649
Total	16,385

The following sections illustrate the exploration of this dataset using NVSS 2.

Case Study Notes

Session 1:

The first session took place over the phone (1 hour conversation) between me (Aleks Aris), Steven Harper, and Noshir Contractor.

The following are the outcome:

1. Errors in the data / visualization
    a. Authors have back references in terms of dates (year only) to documents (see Figure 123). In fact, Ben Shneiderman noticed this before this session. This was thought as an error in the data at first and communicated to the UIUC group.
        i. After the meeting, the UIUC group (Steve) reported that nodes weren't placed correctly. After the meeting, I (Aleks Aris) looked into this and found that the algorithm had a minor error that caused misplacements of nodes at times and fixed it (see Figure 124). This also fixes some of the problems that seem to be double-slider problems.
2. Missing data
    a. Missing data had been replaced by –1. Binning was adjusted to visually separate (have a separate bin) for values that have –1. However, this was done by looking at the Author region. It didn't work for the other regions. Via analysis of the data, the TobIG group realized that missing data existed only for 4 authors. Through the visualization, the TobIG group realized that these 4 authors had no incident links. Hence, they decided that it is best to remove these 4 authors. This also eliminated the need to handle missing data.
3. Meaning of attributes & links
    a. CR_LCS_Count had a different meaning for nodes in different regions. At first, this seemed to be the same by a Noshir Contractor. However, he challenged the assumption and asked for its meaning. Steven Harper explained its varying meaning to Noshir Contractor on the phone.
    b. The meaning of links was not apparent to Noshir Contractor. Every link has a slightly different meaning in this dataset (depending on the source & destination region). Noshir C. asked for the meaning of the Author->Document links. The meaning was Authors writing documents. The date attribute for authors was questioned. Steve H. explained the meaning. The meaning didn't match the visualization and we all questioned the correctness of the data (see 1).
4. Features that would be useful
    a. A switch (of value order) in the y-axis was mentioned that it could be useful for this dataset as the low values are at the top. Users mentioned it would be more intuitive if they were at the bottom on the y-axis.
    b. After the meeting Steve thought that the data had exponential distribution (by looking at the visual representation of the data and seeing a lot of empty space). He suggested that a transformation to be applied to the values to distribute nodes more evenly on the display but also have a way to show the non-transformed values on the y-axis.
    c. A TobIG team member (Hank Green?) couldn't see all sliders (only 4 of them instead of 6) on his screen. This was detected to be due to a lower screen

resolution on that user's computer. After the meeting, it was mentioned that it would be useful to plan NVSS's visual display to be usable in 1024 x 768 resolution also considering the projectors that are used for presentations. (The user may have a higher resolution than 1024x768 but definitely lower than 1600x1200.)

5. Reported problems
    a. Links could not be displayed on a computer that had Java 1.6 but they could with Java 1.5.
    b. Problems with teleconferencing were experienced (lab computer in HCIL couldn't launch and display NVSS through WebEx on a user's computer in UIUC).

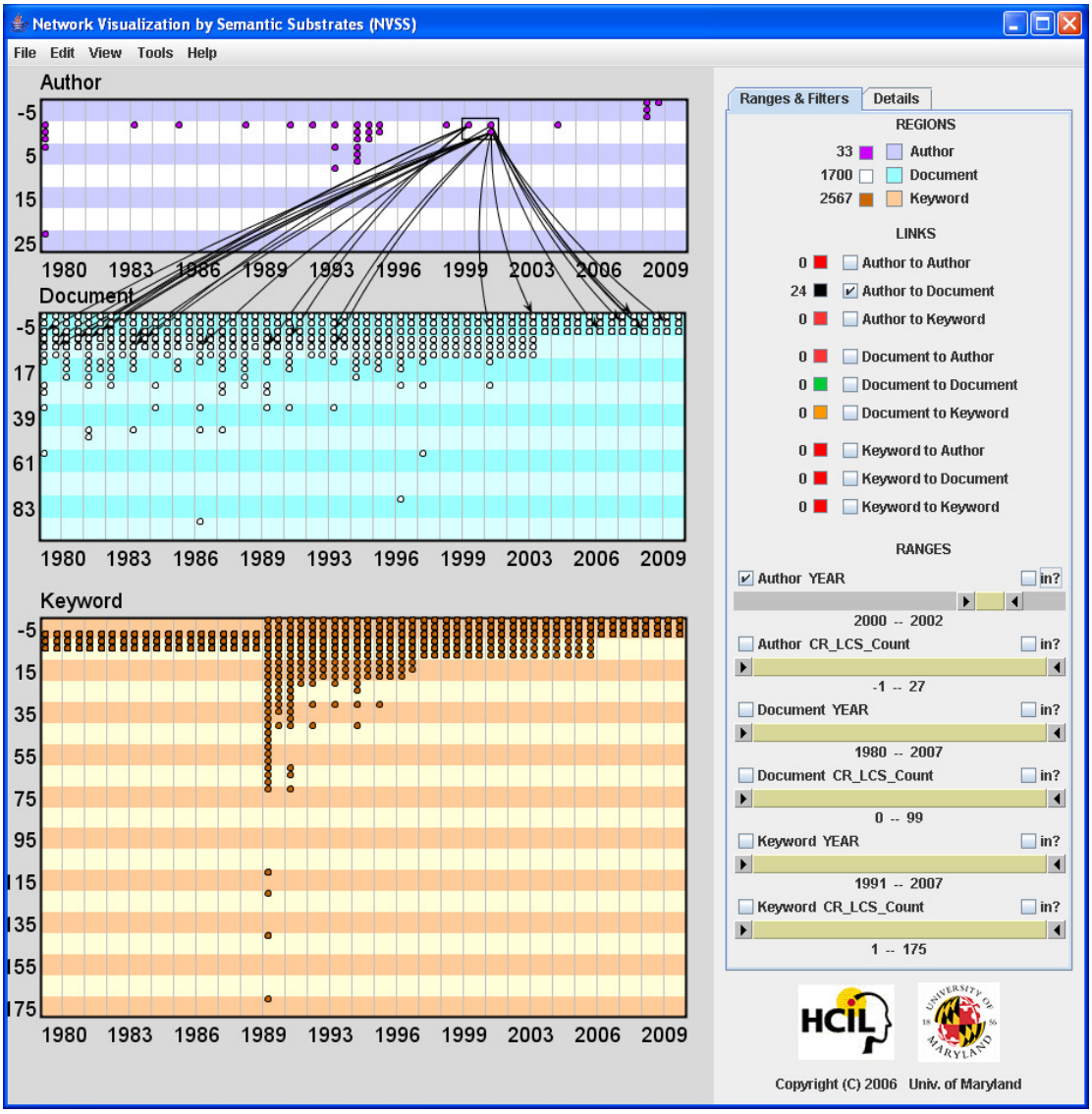


Figure 123 First version when looking at the TobIG dataset. Authors have back references, which is an error due to recent modifications in the placement algorithm of NVSS.

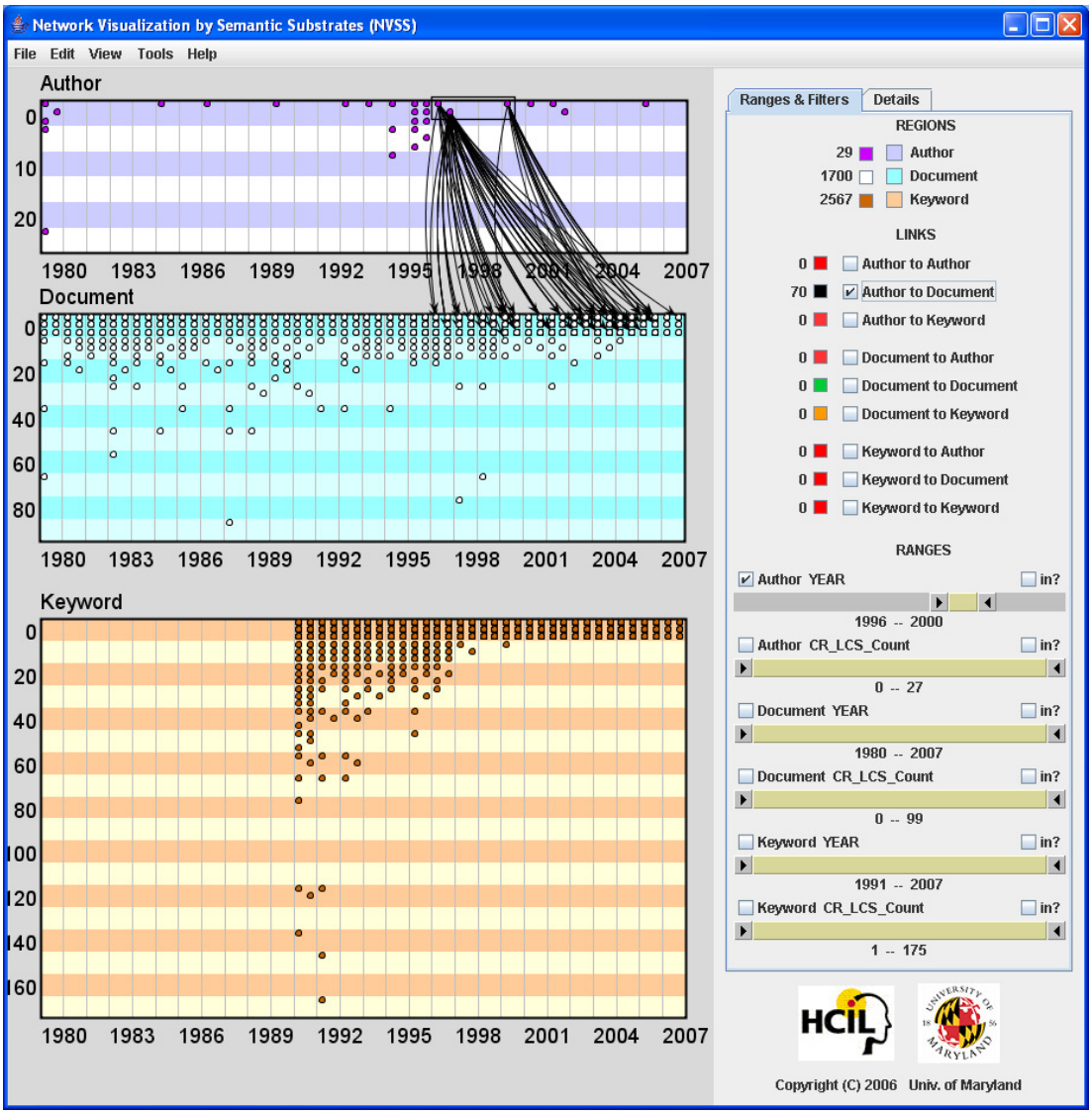


Figure 124 The corrected first version of looking at the TobIG dataset.

Session 2:

The second session also took place over the phone (1 hour conversation) between me (Aleks Aris), Steven Harper, and Noshir Contractor. Hank Green was on the background possibly listening or helping to Steven H.

This meeting was held online using WebEx and it lasted about an hour.
The visualization is generated using sub12.nsf & TobIG_v1 dataset using NVSS 2.0.1.

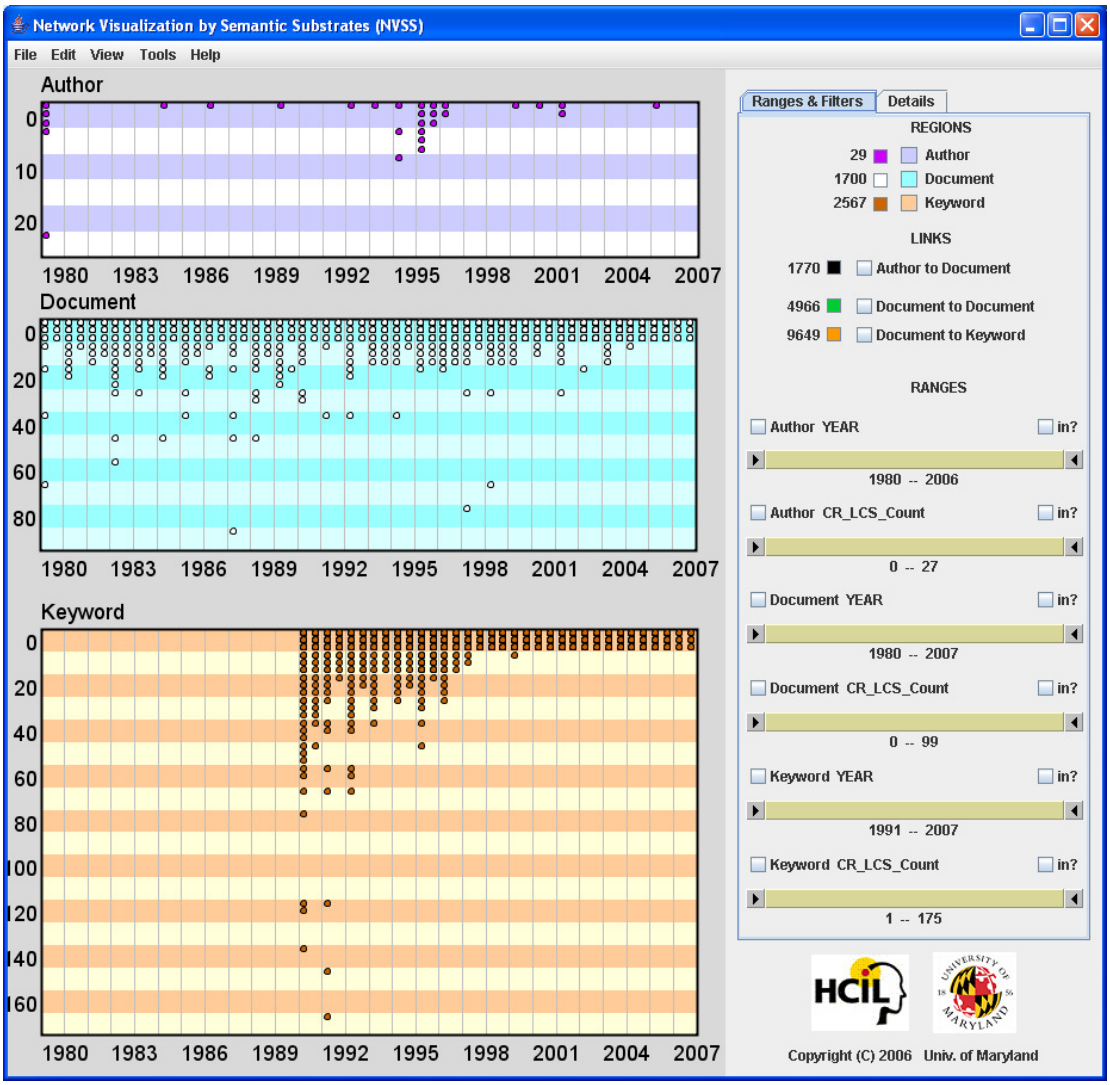


Figure 125 The complete TobIG dataset in NVSS.

1. Observations
    a. Steven H. checked the checkboxes and then used the filters exploring different combinations. He realized that links become 0 when he used multiple filters (this is due to the AND logic of the filters.) Also, in this data, there are no outgoing

links from keyword. When the user activated the filter for the keyword region, links became 0 (by default outgoing links are shown, and "in?" checkbox is unchecked). He realized this and checked "in?" to see the incoming links.

b. The user said the following when thinking out loud. He wondered what the keywords are. It seemed that he wanted a quick understanding of the keywords. He mentioned that it would be nice to see a list of the keywords and not having to click on the "keyword" nodes to see their details one by one. He suggested that labels could help if they were on the graph. He wanted to see which keywords are here the most and perhaps also which ones are the most cited ones. He said the display is very link intensive and noted that the top of the filtering box wasn't visible due to links coming in to the keyword region. (He wanted to see the top of the box.) (This happened when the keyword filter for CR_LCS_Count was set to 48-175.)
c. Steven H. used 2 filters in the document and 1 filter in keyword. It seemed he had a good understanding of how to activate filters and activating more than once at a time and also being able to control the role of the filter (incoming/outgoing) very well after brief explanations.
d. Steven H. identified that there were no keywords before 1991 (see Figure 125). He had asked before whether this is an error in the data or visualization. I (Aleks Aris) had checked this before and this was not an error. This became a small fact that the user learned by looking at the data. Steven H. later reported that there had been no abstracts in the ISI records prior to 1991 and thus no keywords until this date.
e. Steven H. wanted to find a node by providing an attribute value (such as the name of a keyword). He suggested that a text search could provide this functionality.
f. The user asked how to explore this dataset. I asked what the motivation was to explore the dataset (giving an idea that it could either be a combination of specific goals or a more general one). He provided two motivations:
    i. Being able to give recommendations to other people in terms of who to work with or to whom to ask about a specific topic or area in this research field(s) in the dataset. He mentioned, for example, that exploring keywords would help. He probably would see which documents are cited by the keywords and also see then which authors are writing those documents.
    ii. General understanding and exploration of the data.
g. When Steven H. asked for further guidance on how to use the system, I mentioned creating different substrates to favor the attributes of interest. The substrate we were looking at favored interpretations in terms of the year attribute. We talked about the case of creating a substrate where the axes would be switched favoring the CR_LCS_Count. He mentioned that a button to do this quickly on the interface could be useful.
h. Steven H. suggested a feature: being able to cascade links. In other words, being able to show links from the nodes that have visible links. This way, the user would be able to see how one node affects or spreads activation / influence in the dataset. Another use of this could be to find authors given a set of keywords (selected keywords activate documents, which in turn activate authors).
i. Steven H. realized nodes are movable and inquired whether this had a specific purpose. I sensed that the user had a concern of losing the initial arrangement and I mentioned that resizing the application window would restore the placement.

This seemed to diminish the concern. Our discussion continued with other topics / points.

j. In the meantime, Steven H. suggested labels on the axes (towards the end of the axis and with direction along the axis). He also requested that the label to be specified by the user rather than be automatically assigned to be the attribute name by the system. One motivation for this was the different meaning that the attribute CR_LCS_Count has for each of the three different regions (h-score, count, etc.)
k. Steven H. inquired whether nodes could be size coded. (They are. I quickly mentioned that they are size coded by a selected attribute's value.) He wondered whether the links are valued. (They are not. NVSS doesn't support link attributes at this moment.)
l. Steven H. detected and mentioned the discrepancy between the double slider status and the visualization.
m. Steven H. stressed upon the desire to see the details of selected nodes. (There is no feature of node selection in NVSS. He meant the nodes that have visible links after applying a filter(s).) The motivation for this was the question "what are these nodes?" He wanted to select the nodes and see a list, at least their names and maybe more information including frequency, etc. almost like an embedded spreadsheet.
n. Several other possible features mentioned:
    i. Ability to connect the visualization to a database.
    ii. Being able to provide OR logic with multiple filters.
    iii. Popping labels when hovering over nodes.
o. I inquired short-term goals for presentation of this dataset and suggested a subset to attain increased / usable performance during a live demo. The group mentioned that they are working on a subset, where they exclude keywords used only once and too general ones (e.g. research, study, complicated).
p. The group gave a quick overview of their work / software. It presents data from a database in a structured way and has a graph visualization component (with labels) that uses a force-directed type of layout.
q. I gave a quick overview of the Substrate Designer. Due to limited time, I pointed out the major points and mentioned the user manual as a resource / guide.

2. Other observations & thoughts
    a. It was useful to eliminate the filters with 0-links. Less clutter eliminated unnecessary confusion.
    b. The visualization might be misleading due to overlapped nodes. What are visible as 10 nodes might be actually 500 nodes overlapped in a small area. (The planned node aggregation feature could be a good solution (for the visualization part and not necessarily for performance).)
3. Self evaluation (my impressions)
    a. Did the visualization confirm their expectations?
        i. I think it did somewhat (they saw the data in terms of attributes). On the other hand, I sensed that there might have been a lack of not knowing how to proceed. I think users need more guidance and maybe also knowledge (but then this could be considered as a natural part of the process.) Overall, it seems to me a good start with the expectation of more support on details (node info) and features (selection, etc.) in the future.
    b. What did they see that they never noticed before?
        i. That keywords start from 1991.

c. Did they think NVSS was worse or better than their current tools?
    i. I think it provides better and different (and therefore useful) functionalities but it seems they would like to have (some of the) functionalities that support exploration.

Session 3:

This session took place over the phone between Ben Shneiderman, me (Aleks Aris), and Noshir Contractor. It took about 45 minutes. We did not share a display during this conversation.

Ben S. asked the goals for the exploration. Noshir C. answered as follows:
There are 2 goals:

1) There are 2 audiences:
    a. People at NCI (managing networks)
        i. They are looking for funding opportunities and what the priorities are for funding. NVSS could help them in that decision making process.
        ii. Some of the questions they may ask are as follows:
            1. Can we look at the data and find insights for priorities?
            2. Can we identify new topics?
            3. Are there groups of people working on similar things and are they not communicating?
            4. Are there areas that are overly funded and some areas are not funded so well? (Although Ben S. suggested that that would be a hard problem because how well an area should be funded is not necessarily constant.)
            5. What funding / networking opportunities are there?
    b. Users and members of network. These fall into 2 categories:
        i. Regular patron researcher
            1. These are well-established people who have complementary skills and they try to identify people, documents and journals.
            2. They are looking for either common or complementary characteristics.
        ii. Junior researchers
            1. These are people new to the area, such as graduate students or 1& 2nd year junior faculty.
            2. They are trying to understand who is who and what the structure of the network is.
2) The 2nd goal is general exploration.
    a. In an area of research (using MESH, Medline, PubMed) users may want to look for relationships:
        i. Temporal relationships: Is the relationship between two objects getting stronger?
        ii. Key network leaders: Who are the key people in the area of research?

Ben S. asked Noshir C. whether TobIG is the only dataset or whether there are others.

Noshir C. answered that a possible 2nd dataset is the NIH funding database called CRISP.

Session 4:

This session took place over the phone between Ben Shneiderman, me (Aleks Aris), and Noshir Contractor. It took about 30-45 minutes. We did not share display during this conversation. However, Ben S. was exploring the dataset while talking with Noshir C.

Ben S. and Noshir questioned the meaning of Author->Document links as there were only 29 authors and 1,770 links to documents from these authors. It seemed unlikely that the meaning of this type of link is "author writes document." After the meeting, I (Aleks Aris) sent an email to Steven H. and he found out that the original meaning was correct (that authors really write documents).

There are four authors missing since they had no papers (TobIG members that have not published). S. Hecht has a lot of papers (556).

Author: Last Name	First Initial	Documents authored	NVSS Database number of Authors
ashley,	d	122	2001
backinger,	c	13	2002
biener,	l	41	2003
christen,	a	49	2004
clark,	p	16	2005
clayton,	r	60	2006
cole,	g	16	2007
djordjevic,	m	17	2009
duke,	j	5	2010
eissenberg,	t	55	2011
fagan,	p	14	2012
giovino,	g	86	2013
hatsukami,	d	121	2015
hecht,	s	**556**	2016
hesse,	b	16	2017
husten,	c	43	2018
leischow,	s	36	2019
mabry,	p	10	2020
malson,	j	5	2021
markus,	s	13	2022
olster,	d	17	2024
benowitz,	n	**220**	2025

parascandola,	m	20	2026
schad,	p	13	2028
shields,	p	125	2029
stillman,	f	25	2030
tomar,	s	45	2031
vallone,	d	4	2032
zeller,	m	8	2033

It has been discussed that keywords can help determine areas.
One idea to make the visualization more focused was to make a subset and have the100 more important keywords.
Through keywords, the dataset can be explored to see the evolution of themes or topics over time.

Session 5:

This session took place over the phone between me (Aleks Aris) and Noshir Contractor. It took 1 hour. I shared my display during this conversation over Vyew.
Sub13.nsf was used with the nodes & links file of TobIG_v1 dataset. This substrate (sub13.nsf) uses YEAR on the x-axis and SQRT_Count on the y-axis.
NVSS 2.2.0 was used to visualize TobIG. This version is different from previous demonstrated versions in that it contains the aggregation feature (the option on the control panel between Nodes, Mixed, and Metanodes).

Noshir C. suggested to keep in mind the possible option to reduce this dataset by eliminating the years before 1991 as there is no keywords before that. He mentioned that we would lose a couple of authors and many documents but this still may be acceptable.

Noshir C. was looking at the detail of a node and there was confusion of which node was selected. A possible solution is to have a "select" mode for a node that is clicked and being viewed in the details view.

Noshir C. also suggested that there may be a need to look for a node in terms of one of its attributes. In this dataset, an example could be that the user knows a keyword (such as "tobacco") and would like to see where that node is (that represents the keyword). He suggested that there could be a list of nodes and the user could quickly go through them to find it.
My (Aleks Aris) addition to this thought is that sorting might be needed. An alternative is to have text-search capability.

Noshir C. also asked whether there is a current mechanism to do this type of search. My answer was equivalent to "No, however, using filters on the placement attributes, interesting nodes could be found." This seemed to offer a partial solution to the problem; however, a better solution is as it is suggested above. In fact, these two support different tasks that sometimes overlap (so one technique could be used for the other when lucky).

Noshir C. wondered what we could explore with this dataset. I suggested that we could look at the highly used keywords. We could use SQRT_Count as direct indicator of the high-use of a keyword. He concurred and we start looking at the keyword by clicking on them and seeing their name in the "Details" tab. The highest used keyword was obvious as it was the only keyword having SQRT_Count = 13 (highest on the keyword display, see Figure 126). Noshir C. wondered the next. So, we looked at the one on top of that (same year). That one was "tobacco." Noshir C. wondered the next one, which was "nicotine." He thought that was interesting and we stopped and focused on this keyword.

He wondered whether we could look at which documents cite nicotine. He said he knew we could see all links from Documents to Keywords by clicking "Document to Keyword" checkbox; however, since there would be so many things, he wondered whether there was a way to isolate the links to the "nicotine" keyword.

In general, there is no way in NVSS to filter incoming links to a single node. However, in this case, I said we were lucky that it fell by itself to a single cell. Since every cell could be isolated by using the range filters, I suggested doing that, which took us to Figure 126.

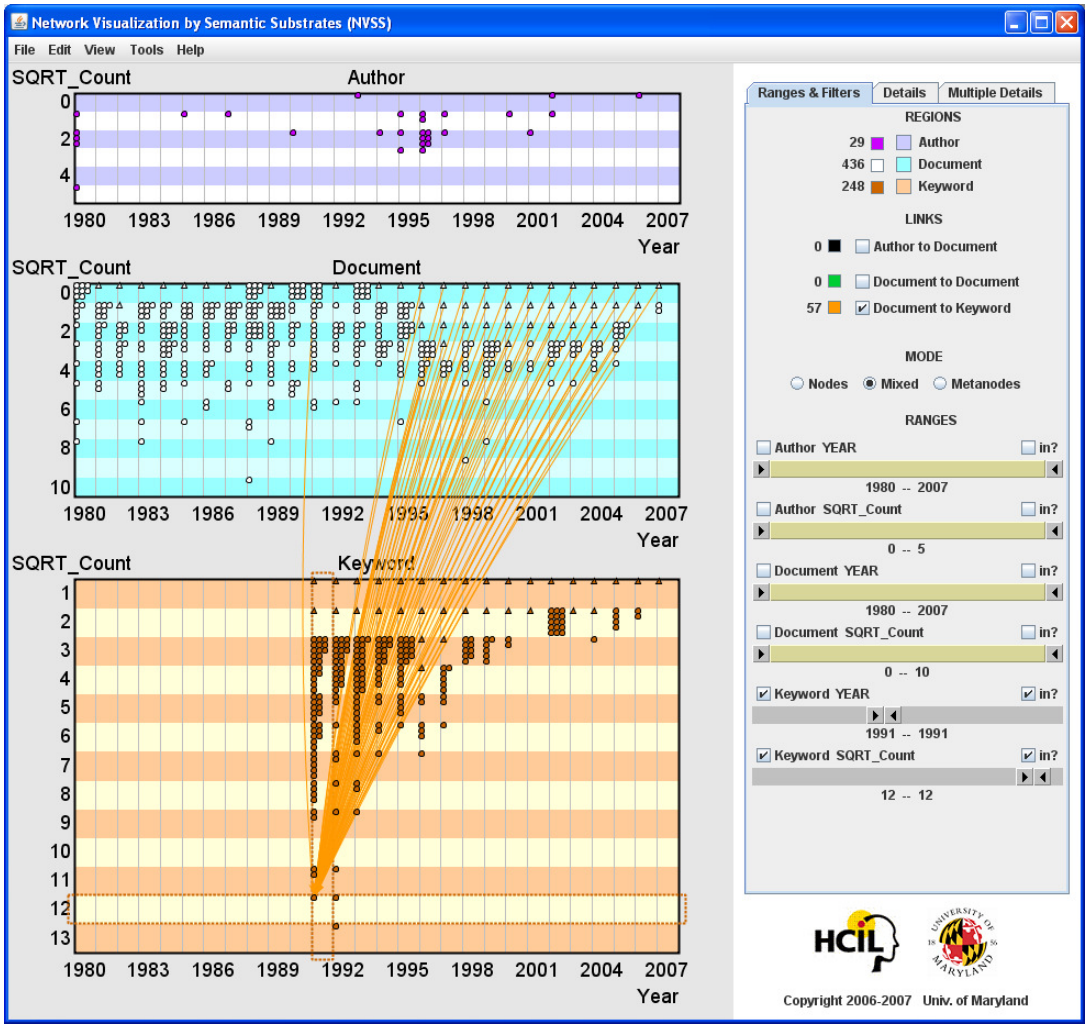


Figure 126 Focusing on the keyword "nicotine" and looking at the incoming links from the "documents."

Noshir C. was intrigued by the distribution of documents citing nicotine. The earliest document that used “nicotine” appeared to be isolated from the latter ones by several years (and then consistently “nicotine” appeared in the literature seemingly every year after that).

Noshir C. wondered whether we could see the details of the earliest document using “nicotine.” Although it took a bit of time to find it, I found the node. (This indicates a room for improvement for NVSS in that there could be a way for users to quickly identify which node is the adjacent (source in this case) given a link.)

The details of this node are as follows:

Attribute Name	Attribute Value
ID	356
NAME	TRANSDERMAL NICOTINE FOR SMOKING CESSATION - 6-MONTH RESULTS FROM 2 MULTICENTER CONTROLLED CLINICAL-TRIALS
TYPE	Document
YEAR	1991
CR_LCS_Count	0
SQRT_Count	0

This was not very surprising to Noshir. When I asked, he said that was because he was not the domain expert on this but he had contacts and was looking to understand before he possibly gets in touch with them.

Noshir C. wanted to see what year the other documents were starting. I used the filter to identify that. It was the year 1996.

Then, Noshir C. mentioned that it would be interesting to see how many documents use this keyword by year. Using filters and dragging them, I mentioned that we could get a sense. Although, I didn't show this during our conversation, Figure 127 illustrates the way. Users will need to switch to the "Nodes" option from "Mixed" to get the actual count of links.

Noshir C. mentioned it would be useful which authors used this keyword at their documents. Since NVSS doesn't have a direct feature for this task, we could not do this. However, we identified this as a potential room for improvement for NVSS. This either could be accomplished by "cascaded links" or a "selection mechanism," where the user can select the documents and then finds all authors that write these documents (author -> document links for these selected documents).

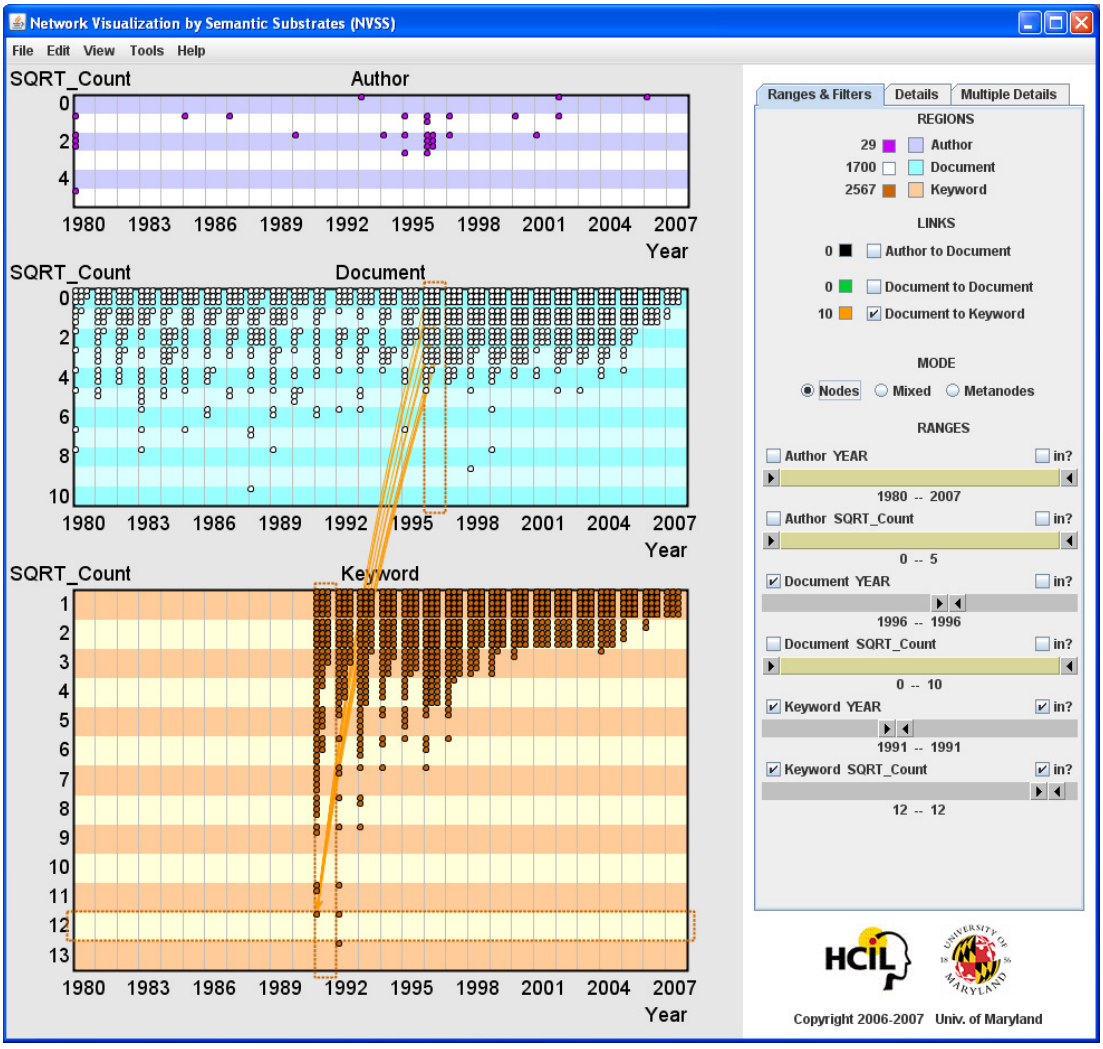


Figure 127 The number of links by years is visualized using filters and dragging them one year at a time from left to right. The count appears on the control panel at the right hand side (10 on the left of the checked "Document to Keyword" checkbox.) The "Nodes" option is selected to get the actual count of links.

Noshir C. then asked whether there is a way that we could look at which documents cite the isolated 1991 document that uses "nicotine." And whether other documents that are using "nicotine" cite them or not.

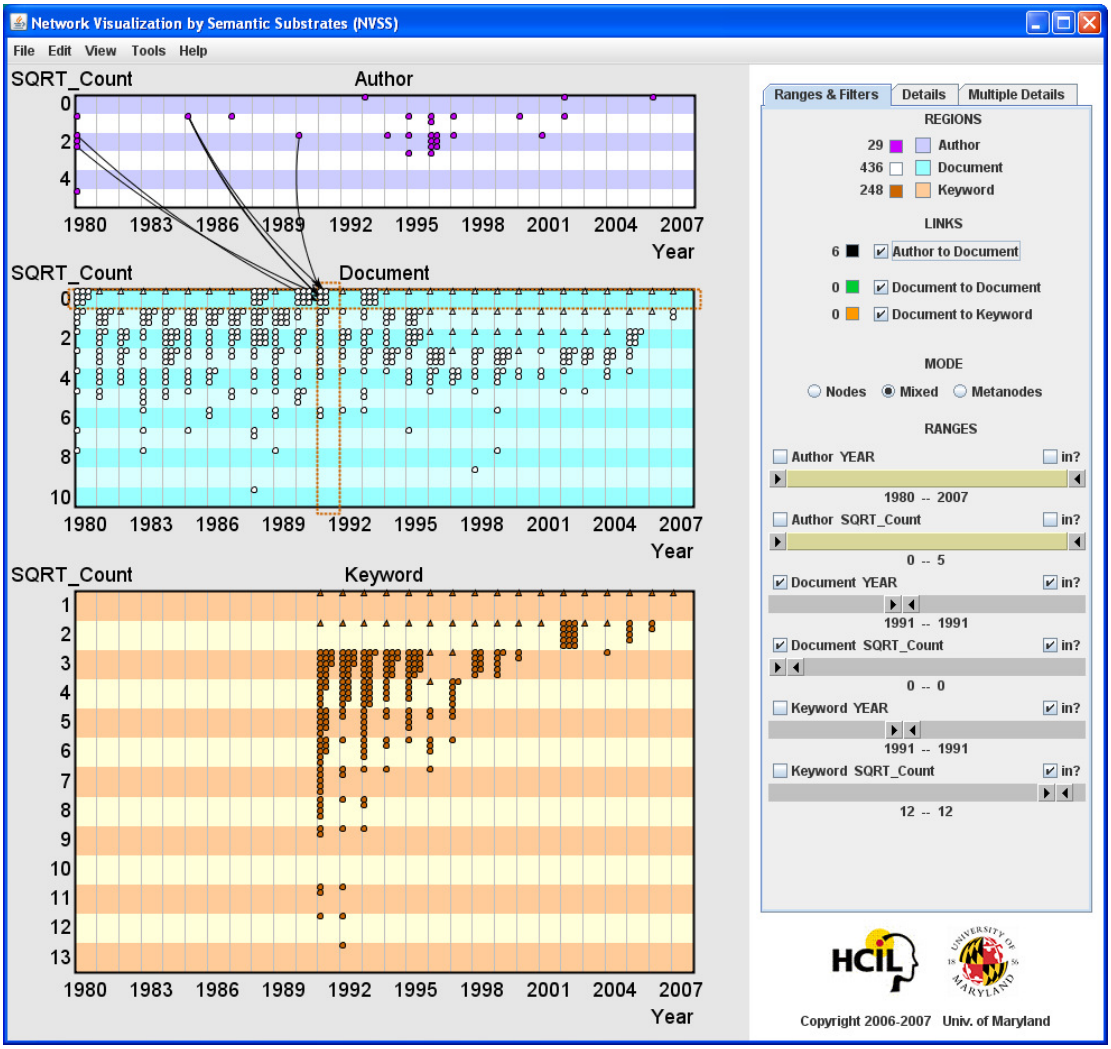


Figure 128 Trying to find whether other documents cite the isolated first document using "nicotine" in 1991 for the first time.

I isolated the document by using filters on the Document region (still I could not isolate fully as NVSS doesn't support finer than cell isolation). By checking "Document to Document" and checking "in?" for the Document filters, the count appeared to be 0. This meant that no document cited this document in the entire dataset. We were lucky in this case. If there were links to this document, we would not be able to answer whether there are documents that use "nicotine" cite this document or not. In this case, we knew that they were not. This could potentially be interesting to domain experts.

Noshir C. wondered which author has written this document. Since the filter was not fine enough to isolate the document itself, checking "Author to Document" filter caused all authors that have written documents in this cell. Noshir C. suggested that the "Document" nodes could have the "Author" information in them; however, NVSS doesn't support nodes of varying set of attributes, and therefore, this could not be done within the present NVSS design. If NVSS had the finer filtering, though, this is

not needed. (We can find the author, click it and see its details.) However, if varying sets of node attributes are allowed this dataset could have a richer representation in NVSS (the unification step would not be needed when transforming the original data into NVSS format).

We also wondered whether this document cites other documents in the dataset. We found only one document (Figure 129).

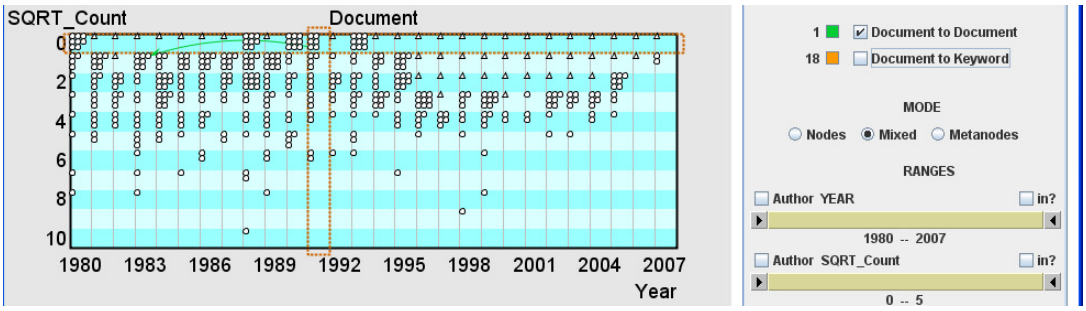


Figure 129 The isolated "nicotine" document cites exactly one document in 1983.

We could get to the details of the cited document via the “Details” tab.
The details of this document is:

Attribute Name	Attribute Value
ID	98
NAME	TUMOR PROMOTERS AND COCARCINOGENS IN TOBACCO CARCINOGENESIS
TYPE	Document
YEAR	1983
CR_LCS_Count	2
SQRT_Count	1

Noshir C. then asked whether there is a way to see all “nicotine” documents so that we can isolate the ones that are cited the most. We realized we could achieve this task by using the filter on the SQRT_Count of the Document region (Figure 130).

Among those, we looked at the earliest document, the details of which are below:

Attribute Name	Attribute Value
ID	465
NAME	Pharmacology of nicotine: Addiction and therapeutics
TYPE	Document
YEAR	1996
CR_LCS_Count	26
SQRT_Count	5

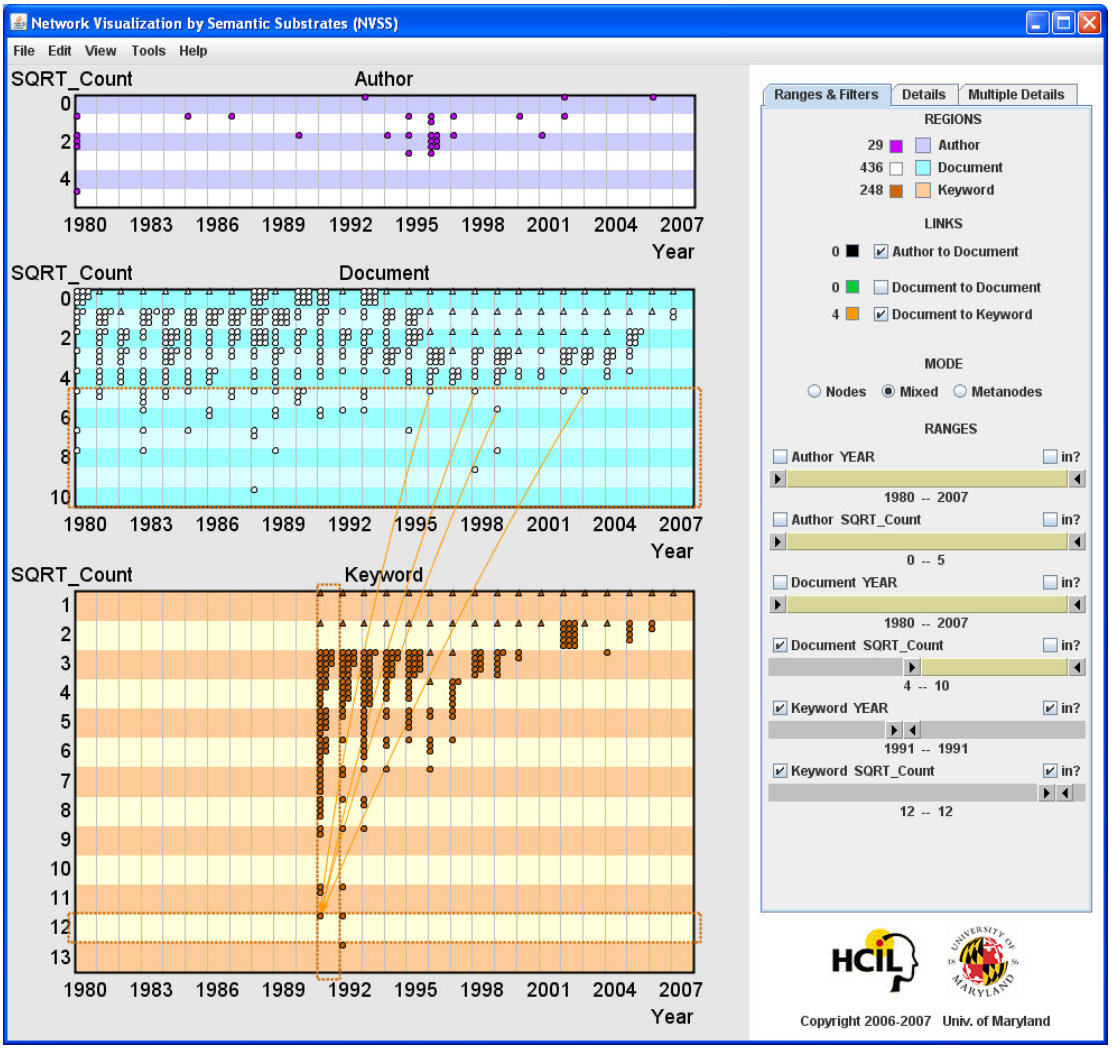


Figure 130 Highly cited documents that use the "nicotine" keyword.

According to the details of this document, it was cited 26 times. By using the filters, we could see which 26 documents these are (Figure 131).

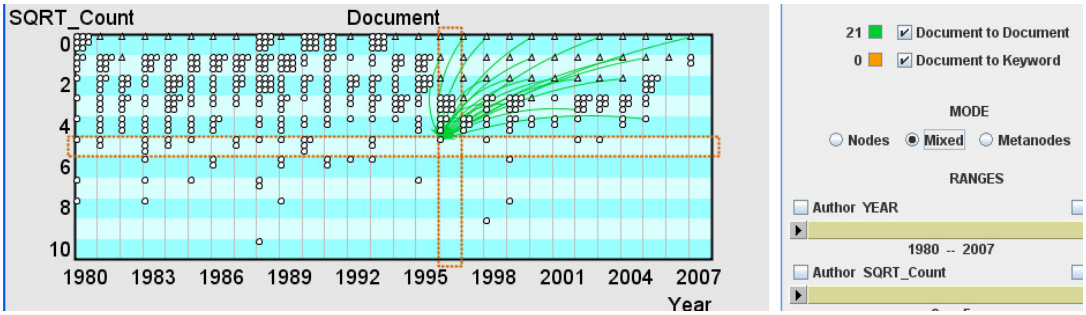


Figure 131 Documents that cite this highly cited document using "nicotine." Aggregation is applied.

Comparing this with Figure 132, we can see the benefit of node aggregation. When nodes are aggregated, it is easier to comprehend the citing documents, especially when users are not interested in seeing the individual documents but the years of the citing documents and how highly cited they are.

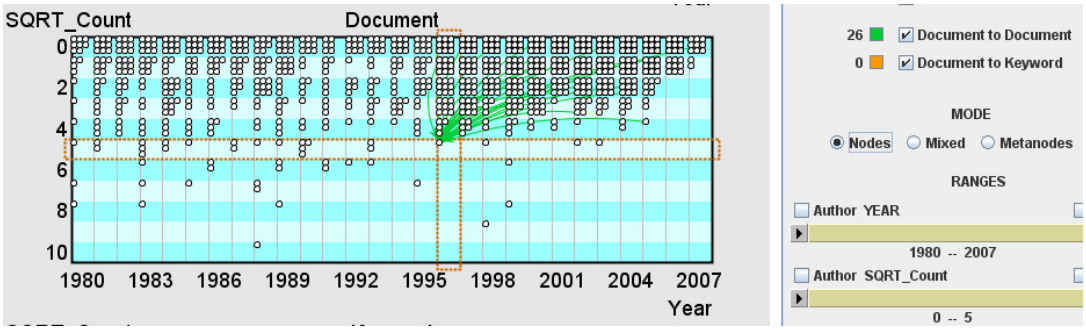


Figure 132 Documents that cite this highly cited document using "nicotine." This time no aggregation is applied. The display in the previous Figure is much more comprehensible.

We concluded the session due to time limit; however, we talked about 3 possible needs/improvements:

1. Finer filtering: Node level filtering is needed sometimes. NVSS provides filtering in terms of the cells defined by the placement algorithm. Within those, there is no filtering capability.
2. The ability to see details of a set of nodes: In this case, Noshir C. wanted to see all available keywords and then select "nicotine" among them. He suggested this could be accomplished by providing a table similar to the one in the "Multiple Details" tab and also to provide a highlighting functionality once the user selects a keyword from the table.
3. Selection: The ability to select a set of nodes in terms of "linked to" or "linked from" and then use this set to continue with further queries such as "which nodes are citing this set."

Self-evaluation:

1. Did the visualization confirm their expectations?
    a. Partially. However, the user expects more navigation support, i.e. the areas of improvements above.
2. What did they see that they never noticed before?
    a. That "nicotine" was one of the highly used keywords (3rd).
    b. The document first using "nicotine" is isolated by several years from others and is not cited by the other ones (in fact, not cited at all).
    c. The highly cited documents that use "nicotine."
3. Did they think NVSS was worse or better than their current tools?
    a. I don't know. I assume they see some difficulties with NVSS; however, I sense that despite that they see the value and usefulness of the concepts used in NVSS (regions, locating in terms of attributes, and filters).

Session 6:

This session took place over the phone between me (Aleks Aris) and Noshir Contractor. It took 1 hour. I shared my display during this conversation over Vyew.
Sub17.nsf was used with the nodes & links file of TobIG_v1 dataset. This substrate (sub17.nsf) uses "Five Years" on the x-axis and Custom_Count on the y-axis.
NVSS 2.2.1 was used to visualize TobIG. This version is different from previous demonstrated version (NVSS 2.2.0) in that it contains the attribute value converter feature, which is specified in the Substrate Designer by loading a file (an .alc file that contains values and converted STRING values for each value; those converted values are displayed on the region axes and on the RangeSliders). The Substrate Design process was completed by me (Aleks Aris) before the start of the session.

I (Aleks Aris) showed the data on this new substrate to Noshir C. as in Figure 133 and explained the axes. The x-axis is binned 5 years (except the last one is 3 years) while the y-axis is binned CR_LCS_Count. This was not understandable first and therefore, I explained that I chose the values. In other words, the values (or ranges) do not conform to a rule or formula but they are decided by the user (me in this case). This explanation helped to understand the view. I also mentioned that this is an example given my understanding of good ranges (based on my experience with the data so far and my guess of how a domain expert could categorize) and hopefully this represents or exemplifies how a domain expert could customize region axes on a substrate. Noshir C. found this reasonable as an example.

Then, I proceeded to suggest a main motivation for today's session: I acknowledged the previous input on navigational features and suggested that we focus on the node aggregation capabilities today. Specifically, I mentioned that the navigation could be a further contribution in the future and currently it seems to me the best would be to focus on how node aggregation helps, could help, or could be improved to help.

Noshir C. understood the motivation and gave the input that the navigational features would really be beneficial and that we consider them for not far but nearer feature and give them high priority if possible. He mentioned that the navigational features would make NVSS much more useful to his contacts who are potential domain experts for this data.

Looking at Figure 133, Noshir C. commented that this view (looking at the Author region) suggested that the year range 95-99 seems to stand out with many prolific authors. He actually said the most prolific authors are found in this region. I commented that it seems most of them to be there and the most prolific one is in the 80-84 region (whose visibility is not well due to the tight alignment to the left of the cell). He noticed and agreed and still thought that this is interesting. (I also mentioned that although we don't see, the rows on the Author region that have no labels actually have them but they don't appear due to the small size (height) of the region. I provided the info that the one above 10-26 is 5-9 and the one below is 27. I also added that the most prolific author is not very distant as his score is exactly 27. I made a mental note to myself to resize the Author region so that the labels on the y-

axis are all visible. (Later, I produced sub18.nsf and found that the height needs to be approx. twice as much for all the labels to appear). Overall, sub18.nsf utilizes space better and looks nicer.)

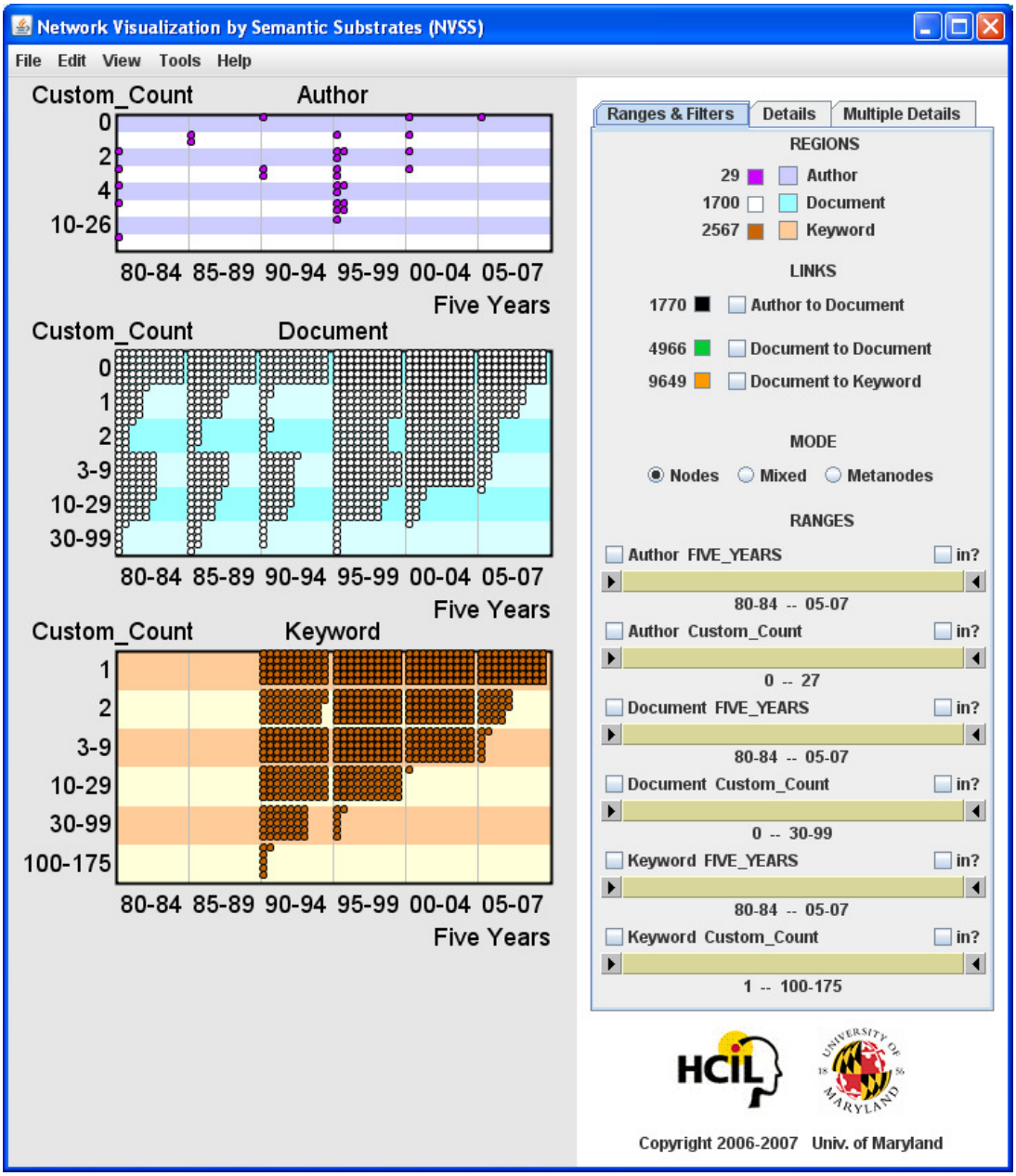


Figure 133 Looking at the TobIG dataset with fewer values on the axes.

I switched the view to metanodes (Figure 134).

Noshir C. asked what we could infer from this view. He mentioned that in the Documents view, every cell had a triangle and that did not seem very informative. Then, he proceeded to interpret the triangles in the Keywords region and noted the incrementally missing triangles toward to the right of the region. He contemplated

whether that could have any specific meaning. I suggested that this is pretty normal as the x-axis represents years. I mentioned that there is less opportunity for a keyword's use as the first time they appear progresses in time because this naturally leaves less time for them to be used. I added that I would find it interesting if the triangles were not missing because then this would mean the more recent keywords were also popular for some reason.

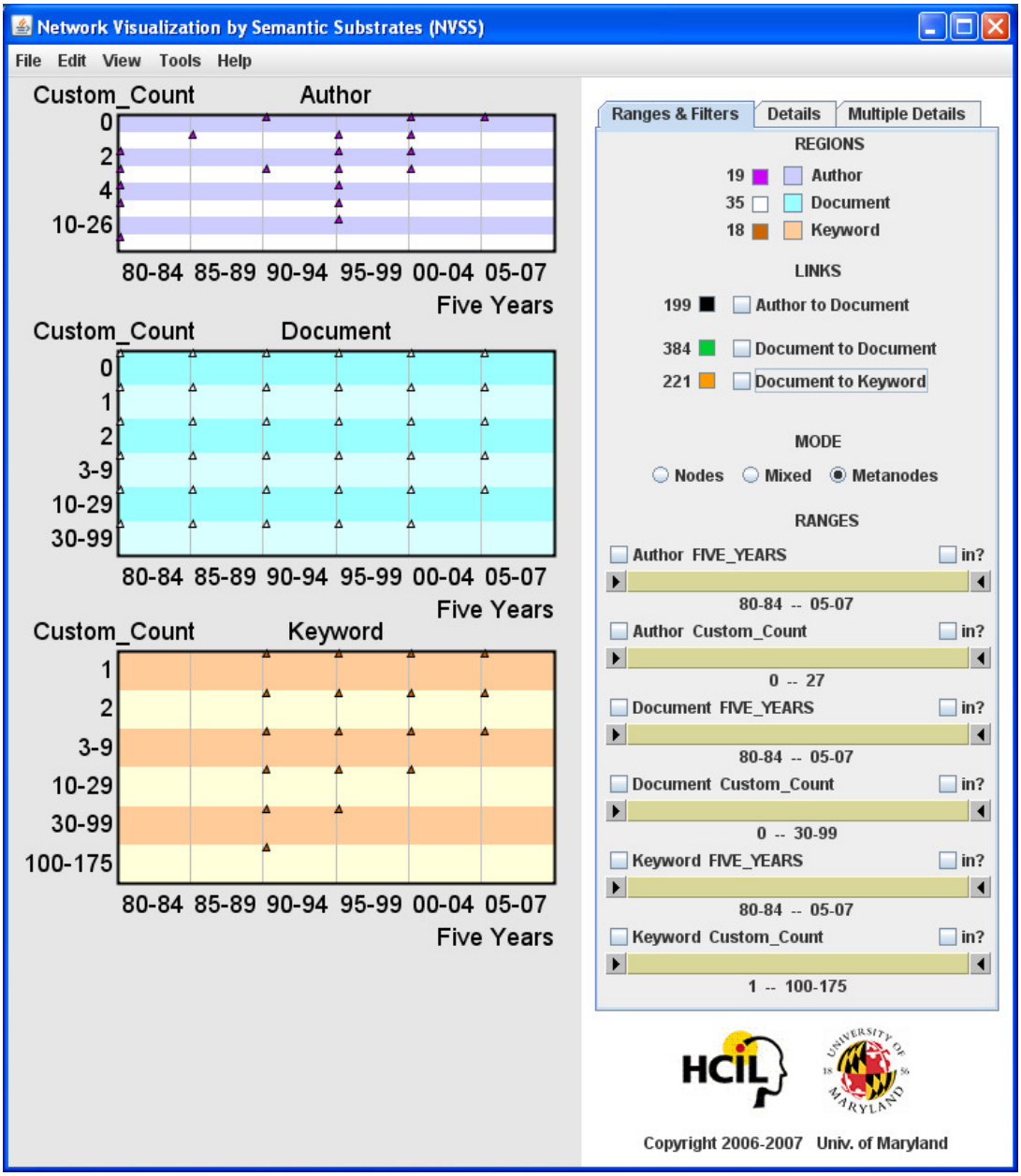


Figure 134 Nodes fully aggregated in metanodes view.

Then, I switched the view to the mixed mode (Figure 135).

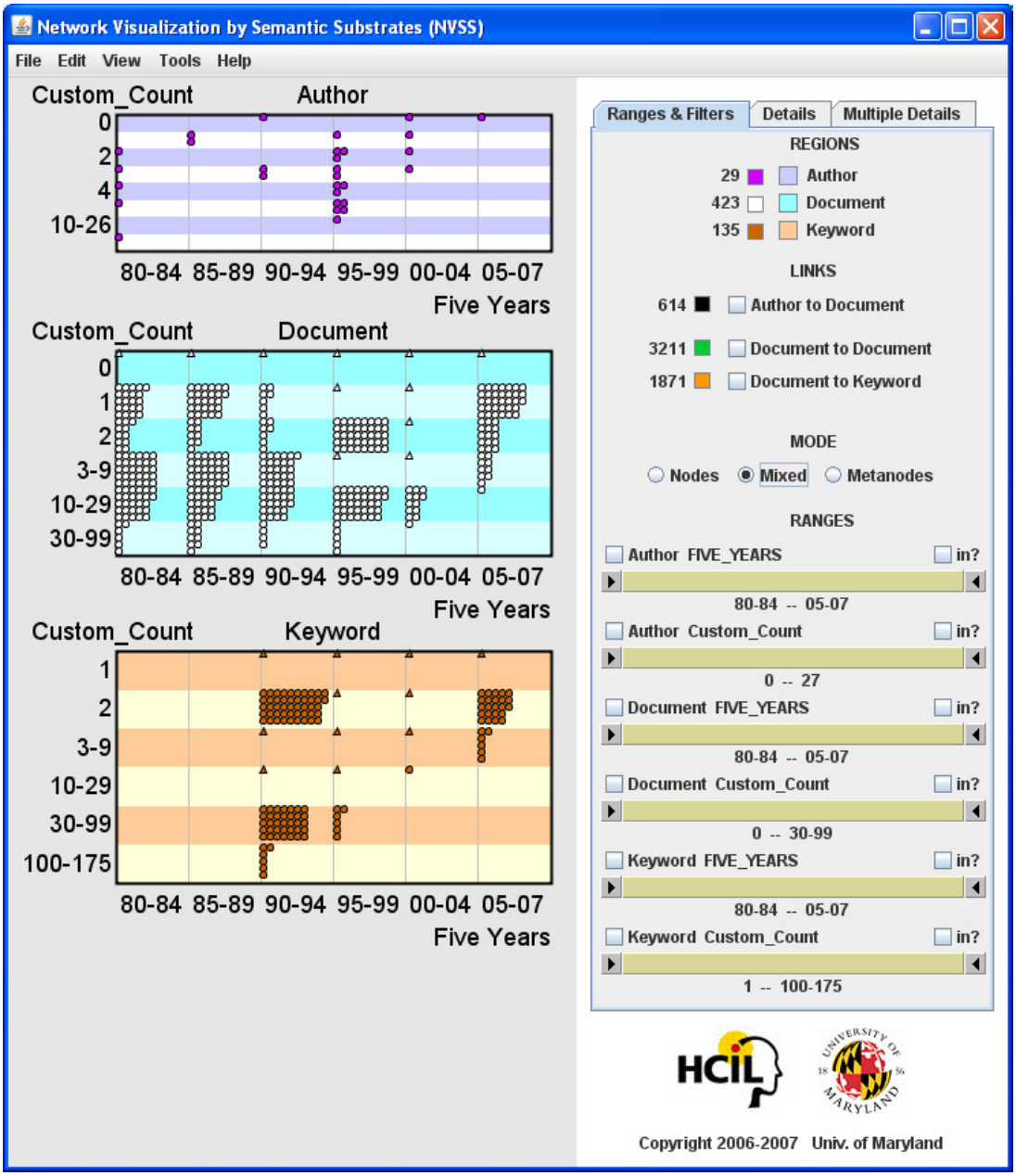


Figure 135 Looking at the TobIG data in mixed mode.

Noshir C. counted the nodes in a cell and asked whether a triangle represents around 40 nodes. I reminded that the triangles in the mixed mode appear when the nodes don't fit into the cells and I said they are 40 or more. He commented that an indication how many nodes each triangle represents (e.g. via size coding) would be useful.

Noshir C. also noted that it would be better if the size of the triangles themselves could be larger and suggested that the fact that they appear small on the screen is misleading. I provided the explanation for the current implementation: The triangles are also nodes and all nodes are sized either via a constant number or via the value of

a node attribute. Since in this substrate I used a constant number to size nodes (5), all nodes are the same size and they are small including the triangles. However, if they were sized according to a node attribute, then most likely the triangles would naturally be displayed as larger. For example, if nodes were sized according to the CR_LCS_Count attribute, this probably would happen. He was excited and concurred that that would be a very good idea to do (i.e. to size using CR_LCS_Count). I also mentioned that in the case of an aggregated node (triangle), the value would specifically be based on the sum of the attribute values of the nodes it represents. I also mentioned that we hope that long-term users will get accustomed to this and the triangle size is an issue only for novice users of NVSS. Noshir C. acknowledged this point.

I suggested the idea that perhaps triangles (aggregated nodes) could be sized differently in the event that the user chooses constant size for node size. He suggested that it would be best if they are the largest, that is, they cover the bottom of the cell (he used the term "cell" to describe a cell, which exactly coincided with the terminology I selected to describe them) and the top point would be the mid-point of the topside of a cell.

Then, I introduced links and filtered them (Figure 136).
I mentioned that our hope is that the node aggregation feature minimizes clutter on the display but still in this example (even though filters were applied) we had many links (as this is a dense dataset in terms of links).

Noshir C. acknowledged the idea and mentioned that the links that are connected to a triangle in effect are aggregated links; however, they are represented the same way as links between two non-aggregated nodes. He suggested that the aggregated links could be visualized thicker. I agreed with his view in terms of lack of distinction and representation for aggregated links. I suggested that link thickness is one way. Another way could be to color them differently. Noshir C. found this reasonable (the alternative way of coloring).

Noshir C. concluded by saying that if the triangle size is bigger and aggregated links are distinguishable, then he may be able to show the TobIG data in NVSS to domain experts. He mentioned that he would meet NIH researchers at a workshop on 12/3/2007 (the same day that we scheduled our next phone conversation).

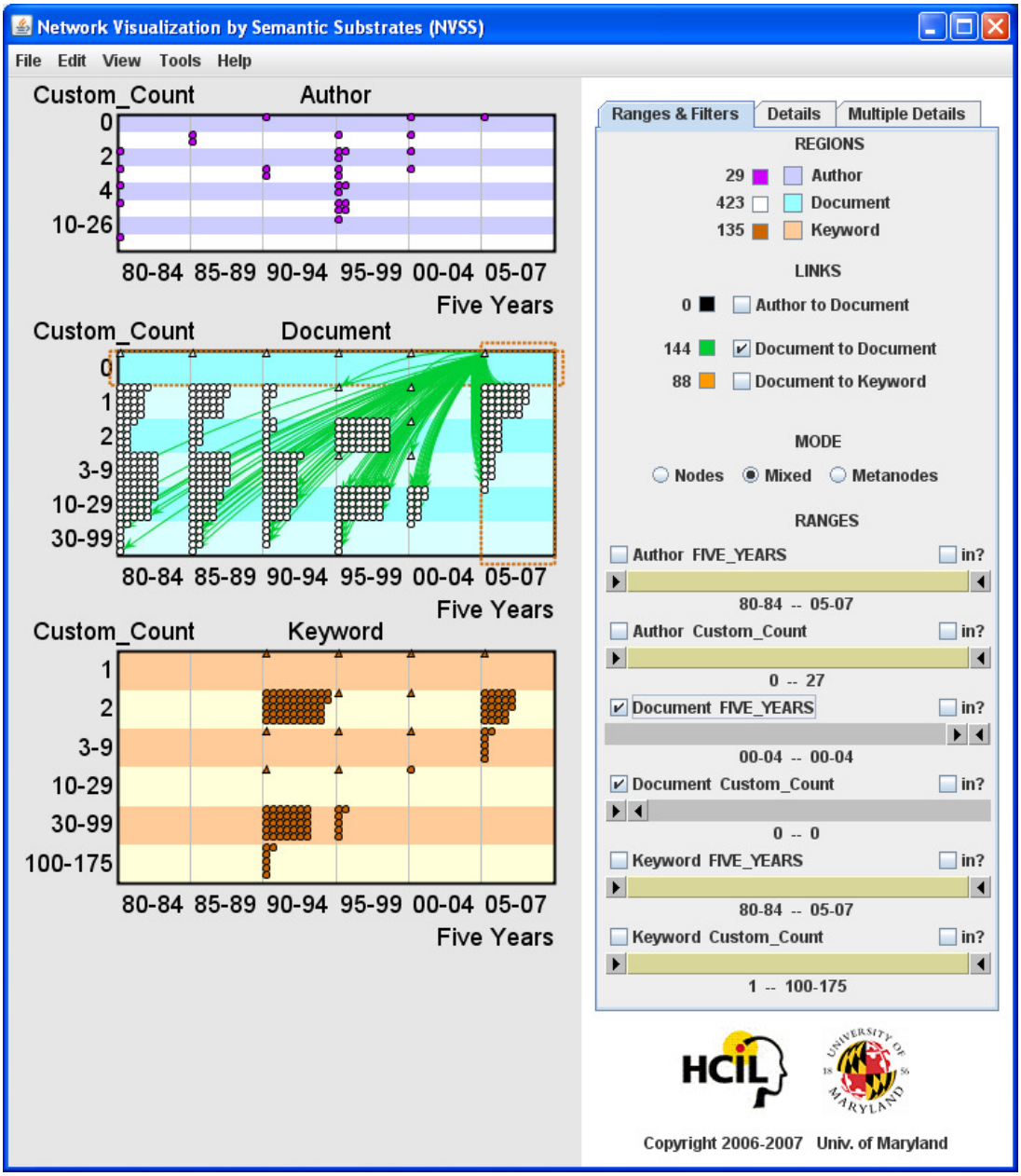


Figure 136 Looking at filtered links in mixed mode.

Self-evaluation:

1. Did the visualization confirm their expectations?
    a. They expected larger triangles for aggregated nodes.
    b. They expected aggregated links to be thicker or distinguishable from non-aggregated links.
    c. Their understanding for the values on the region axes was first that they would be even but with a little explanation they were informed that the ranges were created manually (this capability was appreciated).

2. What did they see that they never noticed before?
    a. That the authors in 95-99 are the most prolific ones.
3. Did they think NVSS was worse or better than their current tools?
    a. My (Aleks Aris) impression is that they thought with the navigation capability it would be better than their current tools. The type of visualization (arrangement, filters, etc.) is a plus that doesn't exist in their current tools but navigation is a minus that they want to have it here too. I guess they have some or all their suggested navigational features in their current tools. (I might be wrong but this is my impression.)

Session 7:

This session took place over the phone between me (Aleks Aris) and Noshir Contractor. It took 1 hour. I shared my display during this conversation over Vyew. Sub19_sumOfCR_LCS_Count.nsf was used with the nodes & links file of TobIG_v1 dataset. This substrate (sub19_ sumOfCR_LCS_Count.nsf) uses "Five Years" on the x-axis and CR_LCS_Count on the y-axis (In fact, Custom_Count is used with an Attribute Value Converter as in Session 6. In addition, "CR_LCS_Count" is used as the attribute label for Custom_Count. Therefore, "CR_LCS_Count" appears in Figure 137 as the label of the y-axis attribute.). NVSS 2.2.5 was used to visualize TobIG. This version is different from previous demonstrated version (NVSS 2.2.1) in that it contains metanode size coding separately from node size coding. Also, the nodes are represented by triangles and the metanodes are represented by circles. In addition, the metanodes are centered within a cell. The substrate design process was completed by me (Aleks Aris) before the start of the session.

I showed Noshir C. the aggregated form of the dataset and focused on keywords to show how documents use the highly used keywords during 90-94 (Figure 138).

When I switched the view to the "Metanodes" mode, he asked what the size of the metanodes represent. He asked whether it represents the number of nodes aggregated. I explained that it represents the sum of CR_LCS_Count attribute value of the nodes they represent. After thinking about it carefully and for a while (it seemed to a bit hard to make sense for him at first), he understood the meaning. It seemed that Noshir C. tried to conceptualize the size of the metanodes as clearly and as precisely as possible to interpret the visualization (the distribution of the nodes into cells). He thought about the meaning for Documents and Keywords, separately (as CR_LCS_Count has a different meaning for each of these types of nodes). It took more to think about the Documents (the first one), and it took less time for him to think and understand the meaning for Keywords (as it was similar to the Document region).

At some point in the discussion (either I asked and he confirmed, or he mentioned) he suggested that a useful size coding for metanodes would be to size them by the number of nodes they represent. I noted this and implemented this feature later.

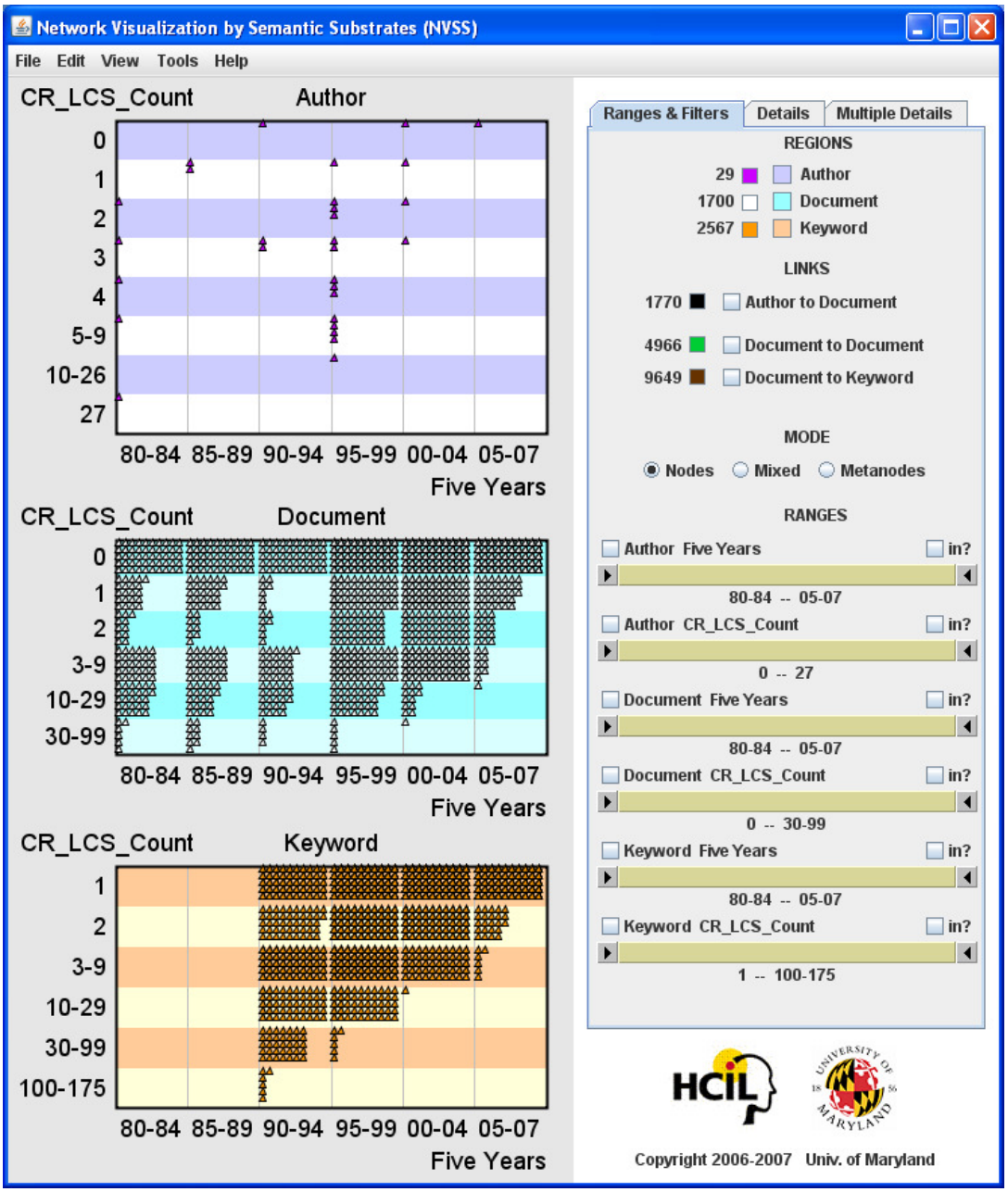


Figure 137 The TobIG dataset in NVSS 2.2.5.

I applied the filters on the Keyword region to show Noshir C. the usage of the highly used keyword during 1990-1994. I explained and pointed out specifically that not all documents are using the highly used keywords. He was very interested and found it intriguing.

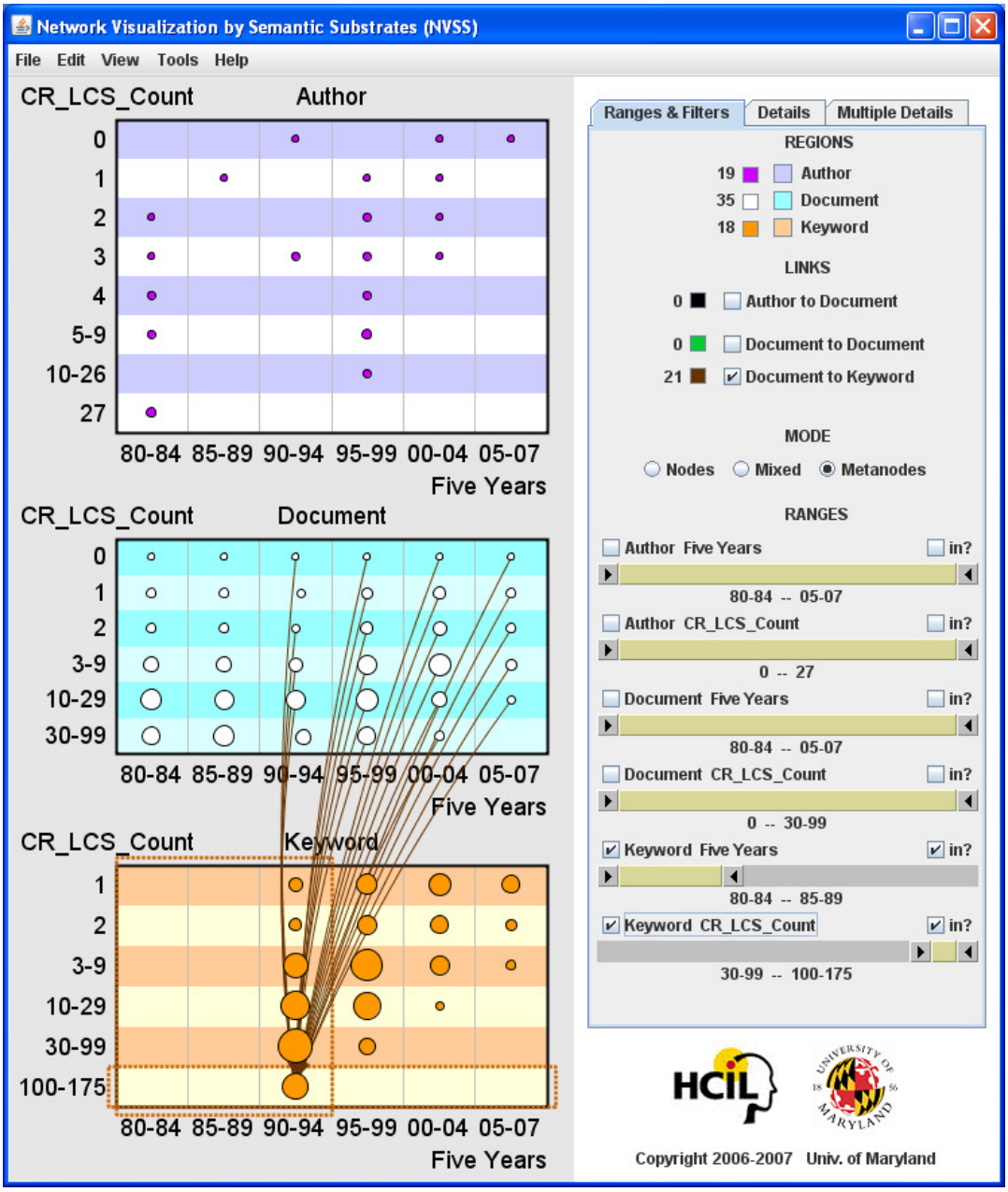


Figure 138 Looking at which documents use the highly used keywords of 90-94.

Noshir C. asked me what those highly used keywords were. I used the "Multiple Details" tab in the control panel to show him (Figure 139). I pointed out that one of them was "tobacco." He seemed to find them all very relevant keywords for the tobacco research.

Then, I pointed out one more time that the most cited documents were not using these keywords and I mentioned that this I found very interesting. Also, I mentioned that this view (Figure 138) shows which "type" (determined by the placement attributes) of documents used the highly used keywords of 90-94. I also pointed out that the

documents cited once (CR_LCS_Count = 1) did not used the highly used keywords of 90-94, too. I mentioned that this is difficult to spot in the normal non-aggregated view. Also, in this view, we can see that the sum of CR_LCS_Count of highly used keywords of 90-94 is not the greatest (see Figure 138 The metanode of CR_CLS_Count = 30-99 is larger!). He asked me for the explanation of what I am referring to. I briefly described.

Ranges & Filters | Details | Multiple Details

ID	NAME	TYPE	YEAR
5382	tobacco	Keyword	1992
5264	study	Keyword	1992
3423	cigarette	Keyword	1991
5190	smoking	Keyword	1991
4559	nicotine	Keyword	1991
5189	smoker	Keyword	1992

Figure 139 Details for highly used keywords in 1990-1994.

Then, Noshir C. questioned about the validity of the interestingness of this phenomenon. He wondered whether the keywords were introduced after the most cited documents were published. I showed the year of those documents using the "Multiple Details" tab in the control panel on the right (see Figure 140). We compared them with the years of the keywords. The interesting fact seemed still reasonable (i.e. it remained interesting although it might be a bit less so than before due to overlapping years). In any case, the ability of NVSS to show such a phenomenon increased Noshir C.'s interest and motivation to use NVSS to explore such data.

Ranges & Filters | Details | Multiple Details

TYPE	YEAR	FIVE_YEARS	CR_LCS_C
Document	1991	2	32
Document	1991	2	37
Document	1992	2	41
Document	1993	2	40

Figure 140 The most cited documents of 1990-1994.

Noshir C. tried to come up with possible hypothesis for this phenomenon. He said it could be that two different groups are in this dataset, who work on different topics, where one group uses the highly used keywords and the other group other keywords. Another reason could be that there was not enough time for the most cited documents to use the newly introduced keywords (which is similar to the concern when we looked at the years of the keywords and documents).

Noshir C. liked the feature that NVSS can show all the keywords in the Multiple Details tab when a metanode was clicked (and similarly for the documents, too).

Noshir C. found it useful to explore the relationship between documents and keywords and asked whether NVSS can help explore other types of relationships in this dataset. He specifically asked whether we can involve the authors. I responded that we could explore the relationship between authors and documents.

Then, I made the adjustments on the control panel of NVSS to show the relationship between authors and top documents, where top documents were identified by documents that are cited 10 or more times. First, I focused on the period 1980-1989 (Figure 141). In this view, only one category of authors are writing top documents. Clicking on the metanode and looking in multiple details, we discovered that this is only one author called "Steve Hecht" (Figure 142).

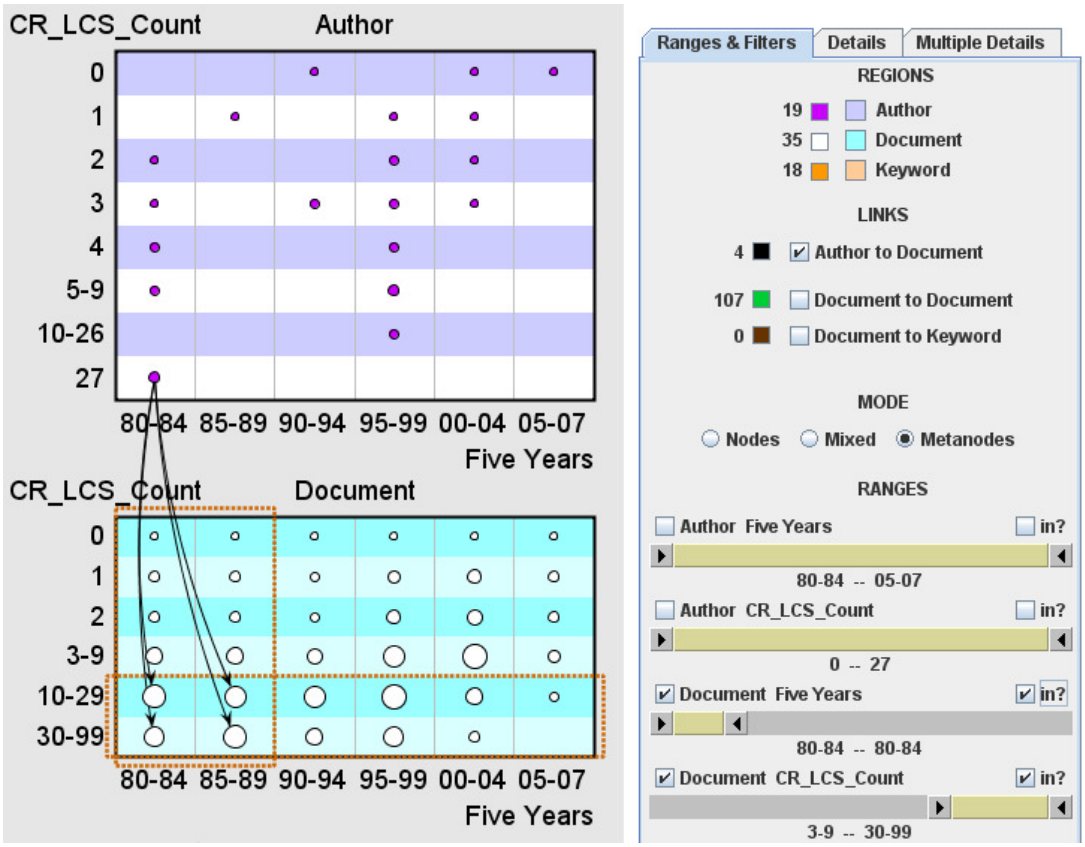


Figure 141 Showing the relationship between authors and top cited documents (documents that are cited 10 times or more).

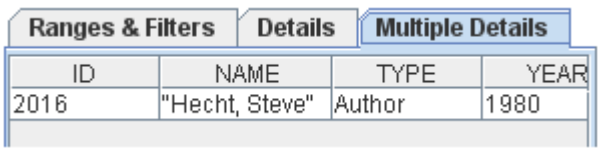


Figure 142 The author in Figure 141 who has H-score = 27.

Advancing the time period to 90-94, another category of authors was added (Figure 143), namely an author category from the 5-9 range. Clicking on the metanode, we saw that it was only one author, “Ashley David” (Figure 144).

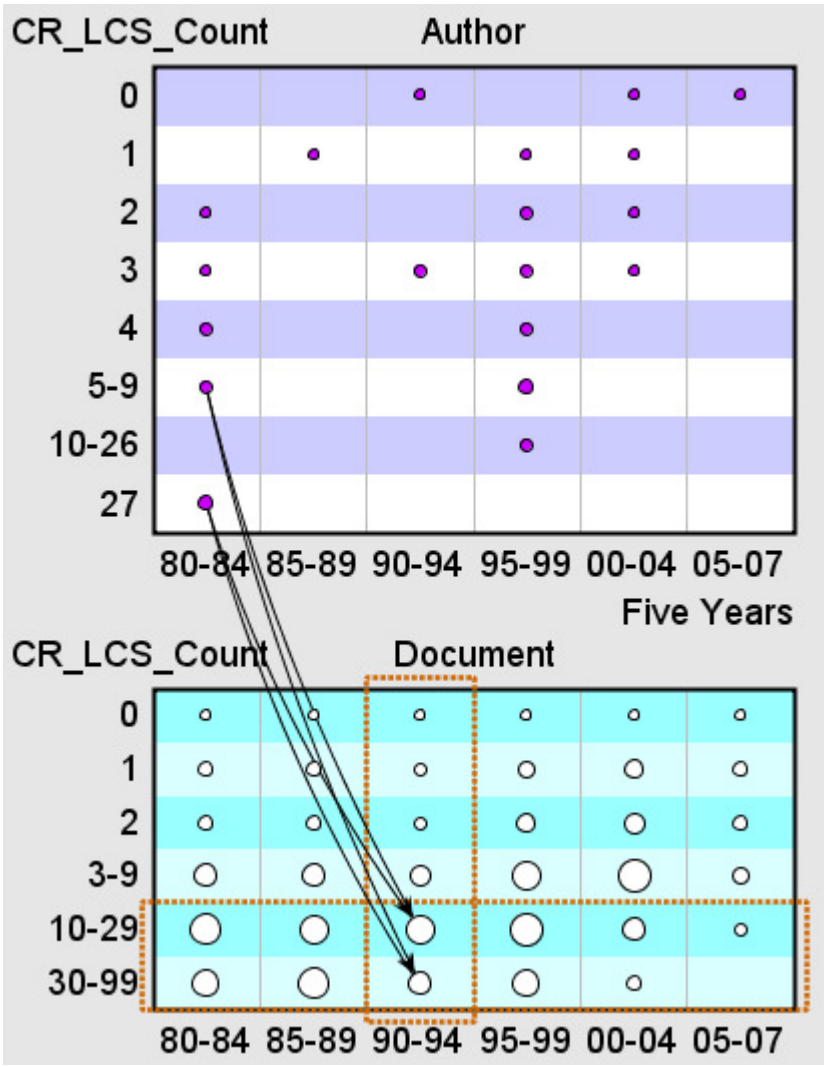


Figure 143 Looking at the period 1990-1994 shows authors writing top documents.

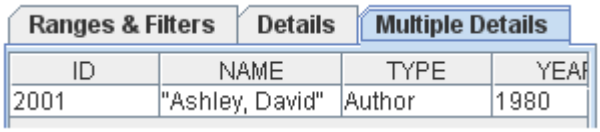


Figure 144 The author in Figure 143 who has H-score 5-9.

Advancing the period to 95-99, we saw a lot of other author categories join (Figure 145). At this point, Noshir C. asked how many authors there are. Since NVSS doesn’t have the feature to tell this directly, I offered to click on each metanode in the new 95-99 period (in the Author region that have a link outgoing) to see the list of the authors quickly (Figure 146, Figure 147, Figure 148, and Figure 149). Noshir C. was

very interested. He actually knew Fran and Scott personally (Figure 146). When I showed the next period, where they do not write top documents any more, he found this reasonable because he knew that Fran and Scott changed their positions and probably were not writing as much in the field for this reason (Figure 150).

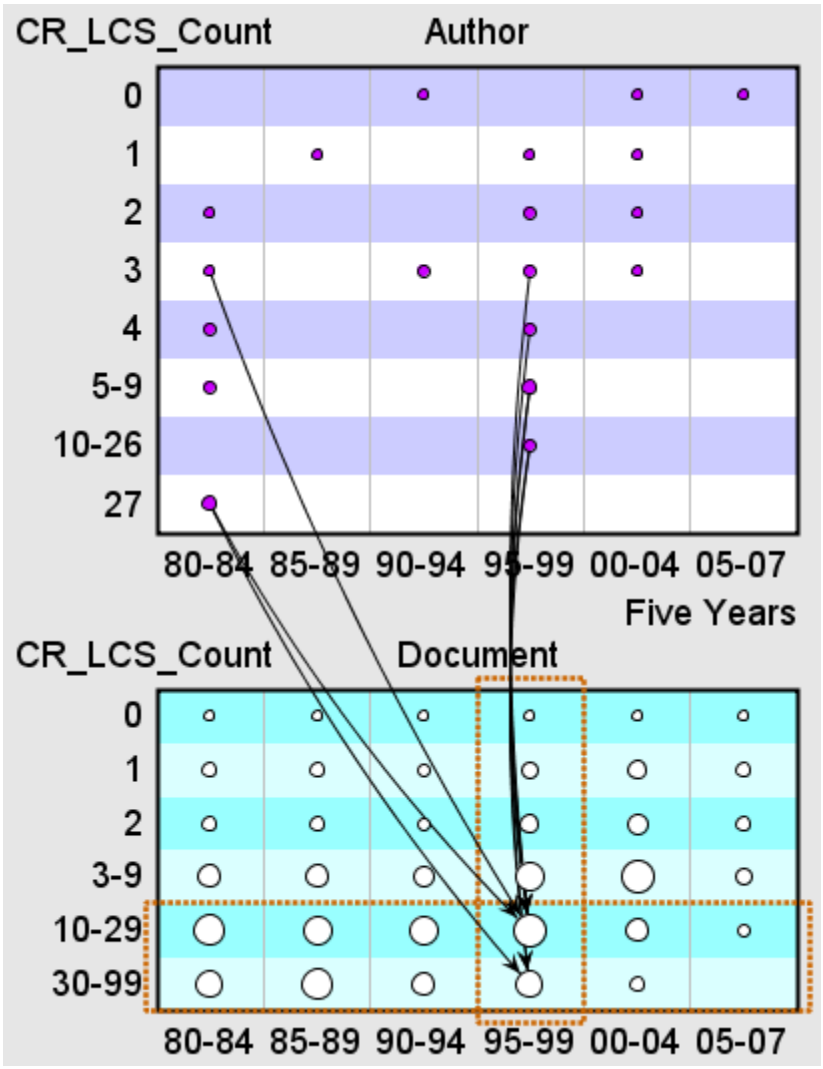
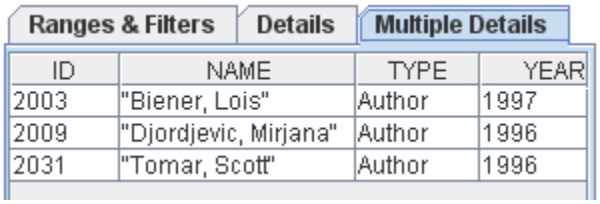


Figure 145 Looking at the period 1995-1999 shows authors writing top documents. There are many new categories, many from the same period 95-99.

Ranges & Filters | Details | Multiple Details

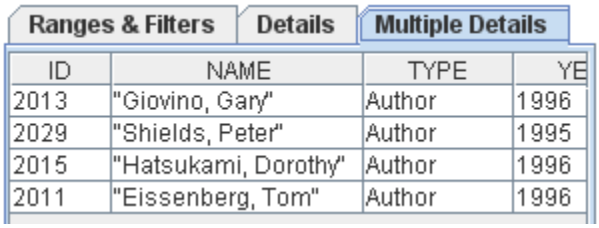
ID	NAME	TYPE	YE
2030	"Stillman, Fran"	Author	1996
2019	"Leischow, Scott"	Author	1996

Figure 146 The authors in Figure 145 who have H-score 3.



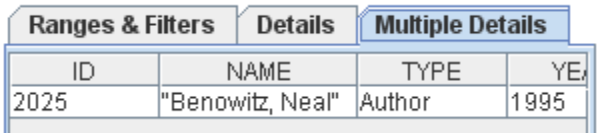
ID	NAME	TYPE	YEAR
2003	"Biener, Lois"	Author	1997
2009	"Djordjevic, Mirjana"	Author	1996
2031	"Tomar, Scott"	Author	1996

Figure 147 The authors in Figure 145 who have H-score 4.



ID	NAME	TYPE	YE
2013	"Giovino, Gary"	Author	1996
2029	"Shields, Peter"	Author	1995
2015	"Hatsukami, Dorothy"	Author	1996
2011	"Eissenberg, Tom"	Author	1996

Figure 148 The authors in Figure 145 who have H-score 5-9.



ID	NAME	TYPE	YE
2025	"Benowitz, Neal"	Author	1995

Figure 149 The authors in Figure 145 who have H-score 10-29.

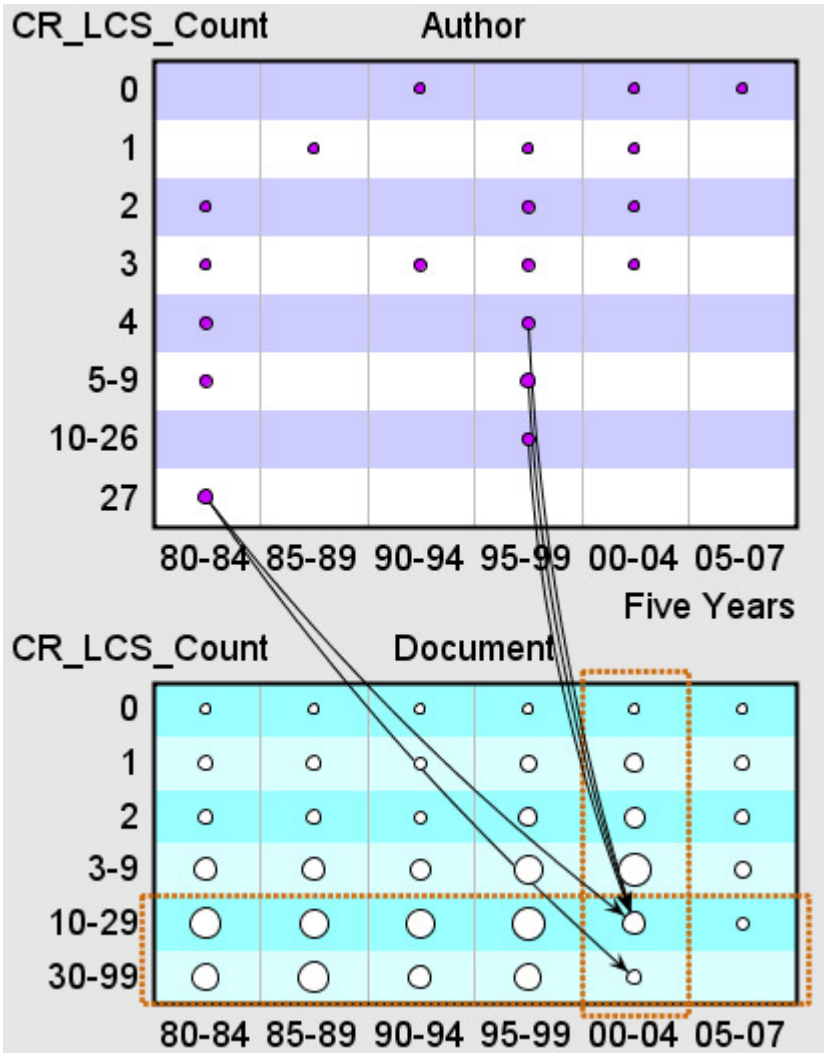


Figure 150 Looking at the period 2000-2004 shows authors writing top documents. The authors having H-score and writing top documents in the previous period vanished.

I cautioned that the measure of "top-documents" might be naturally getting stricter because there is less time in this period (2000-2004) to cite documents produced in this period (Figure 150). To check this assumption, we switched to the node view, which showed that there were only 16 author-to-document links (Figure 151).

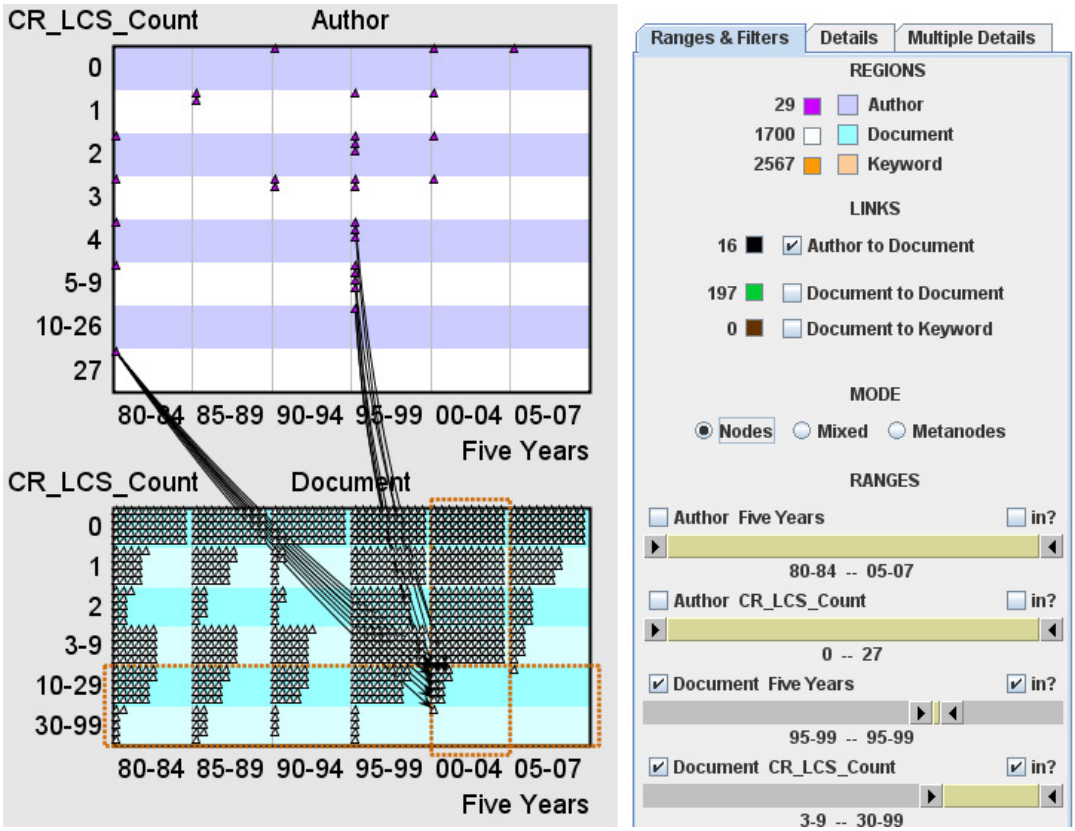


Figure 151 Switched to the node mode to see the number of actual author-to-document links in 2000-2004. There are only 16 such links, which users can see on the right hand side to the left of "Author to Document" checkbox.

Comparing the number of links (16) in the period 2000-2004 (Figure 151) with the links in the earlier period 1995-1999 (Figure 152) provided some evidence for the reasoning that the measure for "top-document" in 2000-2004 might indeed be getting stricter, thus revealing one useful use of the nodes mode (to check assumptions and either correct them or confirm their validity). Further, we also quickly looked at the earlier period, which is 1990-1994, which had 27 links. We thought that this was ok.

Switching to the nodes mode momentarily also revealed that there was only one document in the 2005-2007 period. In fact, there was only one incoming link in that period (Figure 153). I mentioned this to Noshir C. Then, I switched the view to display links from "Steve Hecht", the author with H-score=27 (Figure 154). In this view, we saw that Steve Hecht, the most prolific author of the dataset, did not write only top-documents but all kinds of documents (except the last category, which is insignificant because it contains only one document as illustrated in Figure 153).

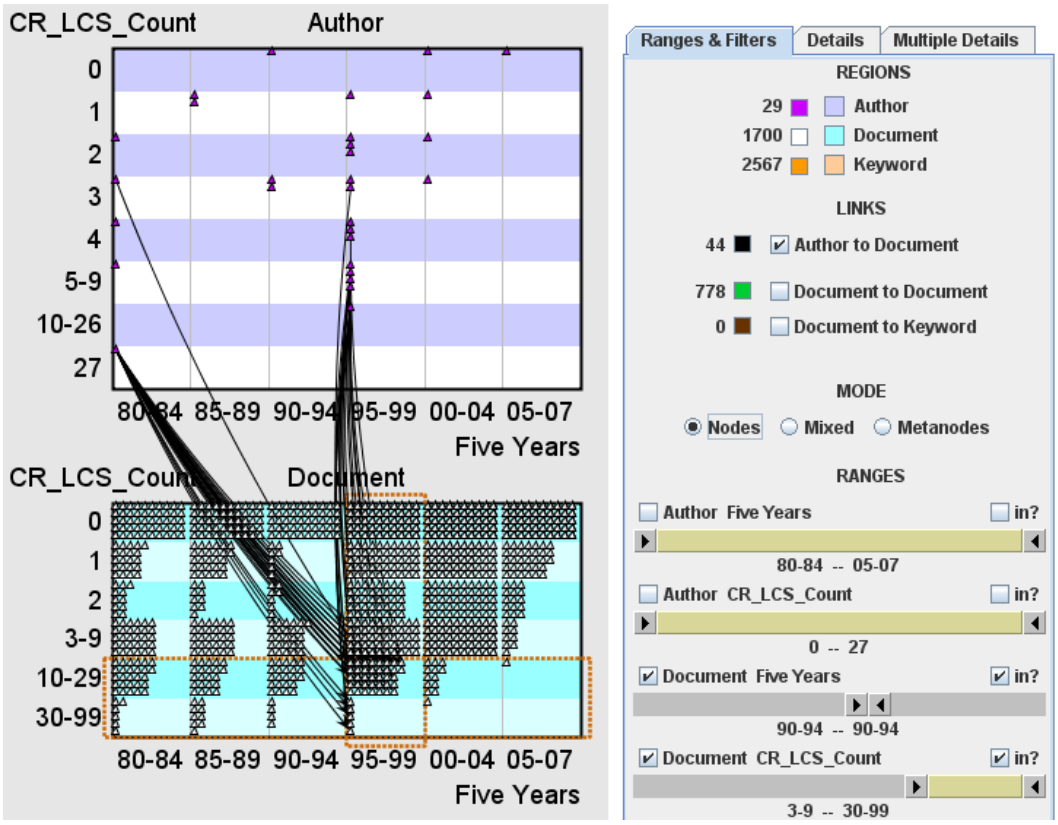


Figure 152 In the nodes mode, shifting to the earlier period to see how many actual author-to-document links there are in 1995-1999. There are only 44 such links, which is significantly higher than 16 links in 2000-2004.

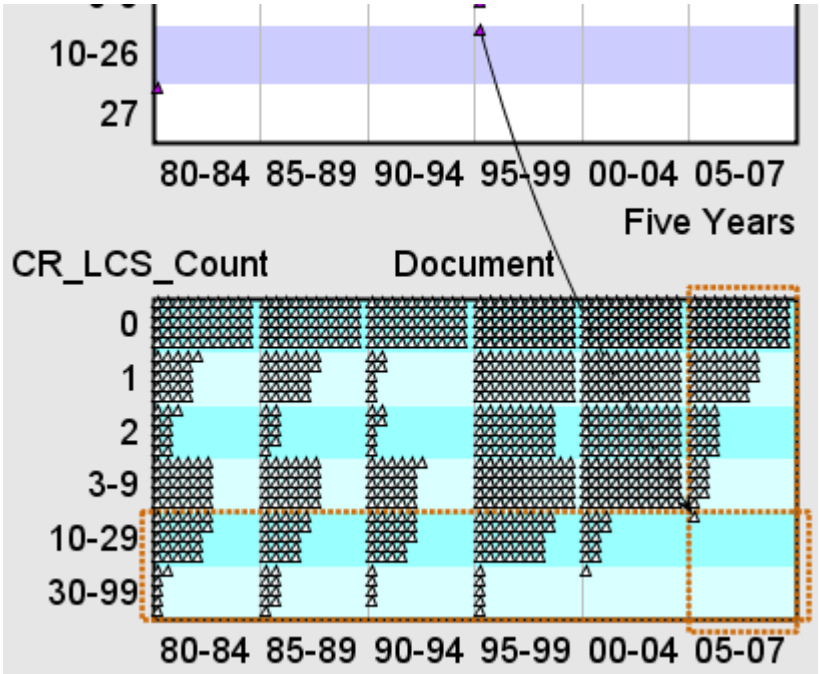


Figure 153 There is only one incoming link into the 2005-2007 period for documents.

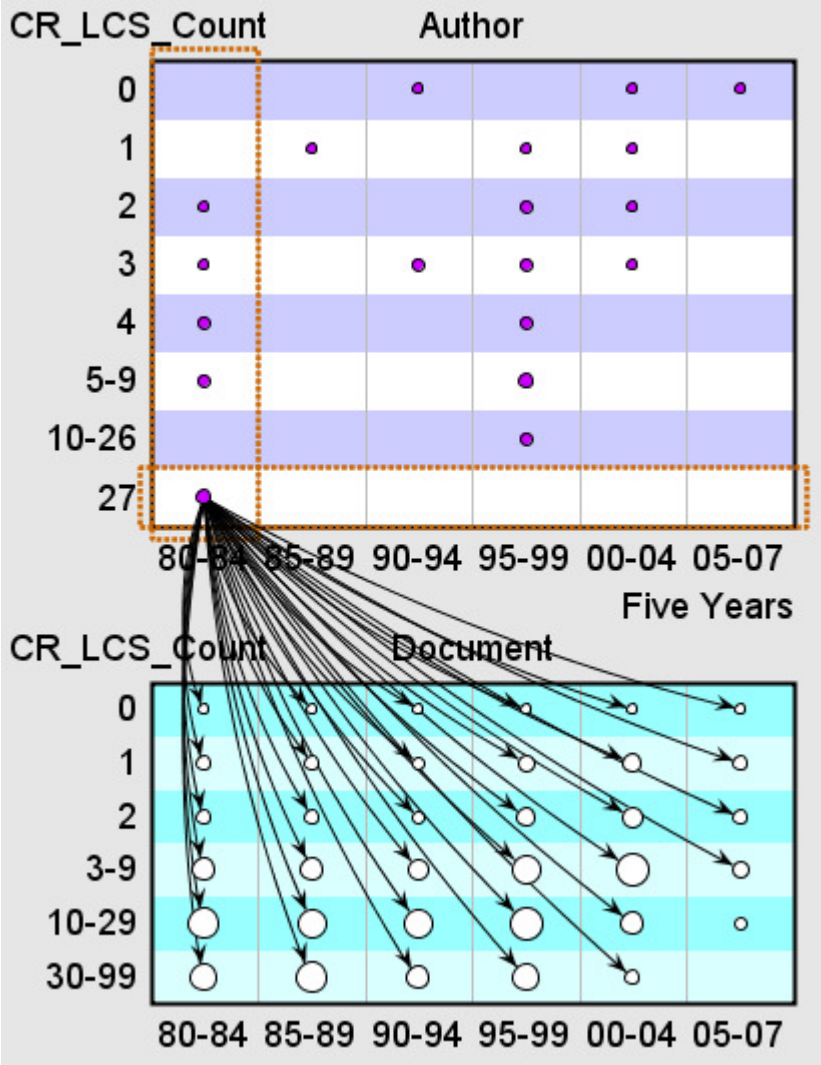


Figure 154 Showing the categories of documents that Steve Hecht wrote.

Noshir C. was pleased by the exploration process that NVSS enabled on this dataset. He said "This is wonderful" and asked what we can do next. He said that we can have insights by looking at datasets like this in NVSS. He also said "I like this." He thought that NVSS was in a stage that could be used to explore datasets with domain experts and was enthusiastic to get in contact with potential experts that could benefit from this type of exploration.

Session 8:

This session took place over the phone between me (Aleks Aris) and Noshir Contractor. It took almost 2.5 hours. I shared my display during this conversation over Vyew.
Sub20_sizedByNumNodes.nsf was used with the nodes & links file of TobIG_v1 dataset.

The substrate is updated from the last time to use separate attributes of Custom_CR, Custom_LCS, and Custom_Count. These attributes have the index values for binned CR, LCS, and Count values. AVC (Attribute Value Converter) files are applied to the substrate to map indices to meaningful bin labels, such as 80-84 to denote the first Five Year period. NVSS 2.3.5 was used to visualize TobIG. This version is different from the previous demonstrated version (NVSS 2.2.5) in that there is a "Node Size" tab allowing users to alter node size and/or use it as a legend, where the transformation function is generalized (contains intercept, the sqrt function, and scaling; separating components of a standard formula). In addition, the metanodes can be sized by the "number of aggregated nodes" they represent. The nodes file format is simplified by removing the attribute labels row (3rd row). SGMain operations are made more robust by detecting common errors and reporting them (instead of crashing). A few other minor improvements are made, such as adding SAVE and CANCEL buttons in the placement method dialog in the Substrate Designer. The Substrate Design process was completed by me (Aleks Aris) before the start of the session.

The main purpose of this session was to help Noshir C. install NVSS on his laptop (running on Windows Vista) so that he can demo NVSS with the TobIG dataset to others. However, this resulted in many questions from Noshir C. that turned into a useful interaction. Consequently, I found it worthwhile to describe it as a session in this document.

I prepared a version of NVSS 2.3.5 to be installed on a computer (as an executable jar file). This way, Noshir C. would not have to access the internet to use NVSS (usual way is via Java Web Start). Also, I prepared a zipped folder that contains TobIG data (v1) with a few substrates (sub12, sub18c, sub19, sub20) and necessary files.

First, I explained how to run NVSS (by clicking the .jar file of NVSS). When NVSS was up, first we loaded sub20 and I said this was the most sophisticated substrate and therefore, he would probably want to show this one. While selecting the substrate, I explained him how to use the navigation functionality (to go to previously visited locations quickly) in the Extended JFileChooserDialog. It took a bit time to communicate where to click. Also, we discovered that it takes a lot of time on his computer when he chooses a previously visited location from the drop-down list. To avoid the long waiting time (5-7seconds), he preferred to locate the file every time from the root directory. (And he continued to prefer this way later on although he tried a couple times the drop-down list.)

Then, he specified the nodes file. While specifying the links file, he realized the preview in the Extended JFileChooserDialog. He saw link id's going up to around 77 and inquired whether the preview is complete. We quickly discovered that the preview showed a truncated version of the file. Then, he wanted to see the content of the files.

He went to the directory structure and saw the .txt file. Then, I explained that the "Other" directory contained the Excel files and that the Excel files are for humans to read and the tab delimited files are for NVSS to read and that the contents of both are the same.

We looked together at the contents of the nodes file. He quickly previewed the document by looking at the TYPE attribute and inquiries about what the types are to make sure. I confirmed that they are *Document*, *Author*, and *Keyword*. While he previewed the file, it was hard to see the *Author*s as they are few and are located between *Document*s and *Keyword*s. After confirming the TYPE, he quickly confirmed the meaning of the YEAR attribute. Then, he inquired about the FIVE YEARS attribute. I explained that the values are the index of the bins and they are converted to meaningful strings (such as 80-84 for bin index 0) by a mapping. At this time, I remembered that I had forgotten to include the AVC files in the dataset directory structure. Then, he quickly confirmed about the CR_LCS_Count and asked what the SQRT_Count is. I mentioned that at some point, I was exploring whether a square root distribution works better. Then he asked about the last three attributes and whether they are important or to be ignored (Custom_CR, Custom_LCS, Custom_Count).

I explained that the last three attributes are present for different regions (a workaround for supporting different types of nodes) and that they are basically the same. He asked why they are not the same as the CR_LCS_Count. Then, I mentioned the attribute value conversion again and he asked whether he could know the mapping of the indices to the attribute labels. At that point, we gave a little break (although we kept our phone connection) and I resent the updated directory structure, this time including the AVC files.

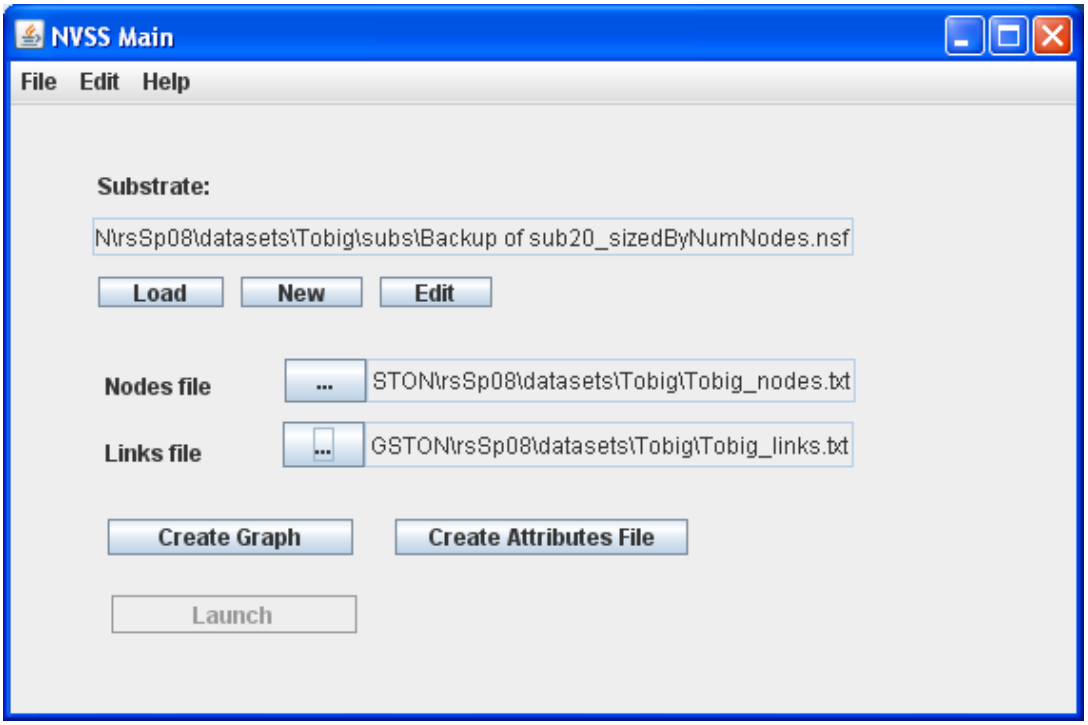


Figure 155 Looking at NVSS Main with Noshir C.

When we returned back to our conversation, Noshir C. got the updated directory structure. I noted the AVC directory. When I explained the structure of AVC files, it was easy for Noshir C. to understand this mapping. Also, it seemed that now it made more sense to him how things are connected as the left column in the AVC file contained the indices and the right column contained the attribute value label to be presented. I also briefly mentioned that this (an AVC file) is used in the Substrate Designer. Then, he noted the "Other" directory and asked how and when the "TobIG Data Model v6.txt" file was used. I explained that it is used when one creates a new substrate. At some point, NVSS was open and Noshir C. also asked about the "Create Attributes File" button (Figure 155). I explained that it creates the Data Model file and he understood that it creates a file of the type that he saw in the other directory.

Noshir C. wanted to see how to use the Data Model file to create a substrate. I directed him to press the "New" button and it asked him to "Load an Attributes File", which Noshir C. noted and told me. He specified the Data Model file from inside the "Other" directory. Since the nodes file was specified in SGMain, it was also specified in the Substrate Designer when it opened. (He asked about that and I explained it briefly.)

Next, I helped Noshir C. to create 2 regions and edit their details including the placement method. When creating the regions, accidental creation of many regions in the "Draw" mode occurred, which I clarified and introduced the "Delete" mode to delete them. It was easy for Noshir C. to quickly delete them. In fact, he used the "Delete It!" button to delete them all and start again. I also explained the "Select"

mode and the coupling of the details on the left panel and the canvas on the right in the Substrate Designer. After drawing the regions, he asked what to do next. I said that we were to set the details of each region. I explained how to set the region label, (grouping) attribute (for the selected region). He intuitively discovered the meanings of fill color and node color. Lastly, the placement method was to be set. In the placement method dialog, I also showed him how to apply an AVC file. At first, there were a few moments of confusion between the grouping attribute and the X- and Y-axis placement attributes; however, after I made a brief explanation, Noshir C. completed specifying the parameters of the placement algorithm. And, then, he quickly completed specifying the parameters of the placement algorithm of the second region. In fact, I was surprised as I did not expect the acceleration in applying the newly learned concepts.

Finally, I guided to set the node sizes. I suggested and he used "attribute-based node size" for node size and used the transformation. The meaning of the transformation was clear. When I mentioned that it was like a formula, Noshir C. mentioned the formula format $y = mx + c$. We thought it might be a good idea to put the intercept to the right ranther than to the left and/or add "*" and "+" signs in between to facilitate user understanding (that the transformation is in the form of a formula). For metanode size, I suggested him to select the "Number of Aggregated Nodes."

After all this, Noshir C. was content that he could build his own substrate. By closing and saving it as a file, the substrate design process was completed. Then, he pressed "Create Graph" and "Launch" buttons. However, nothing happened and we decided to proceed with one of the substrates that I prepared beforehand (sub20).

Noshir C. was reviewing thinking what he could demo. I reminded that he could show the highest keyword example (that were explored in the previous session) and gave guidance.

I also mentioned that nodes can be sized within the visualization using the control panel on the right (as now, there was the "Node Size" tab). He wanted to change the node size. He asked me what I suggest and I suggested that he changed node size and used attribute-based size coding with the CR_LCS_Count attribute. In addition, I suggested that he uses a transformation. He asked me for good parameters and I suggested 5 + srqt(X) * 1.0. Noshir C. accidentally pressed "Refresh" button and we lost the settings; however, he quickly did it the second time and we visualized the display (Figure 156).

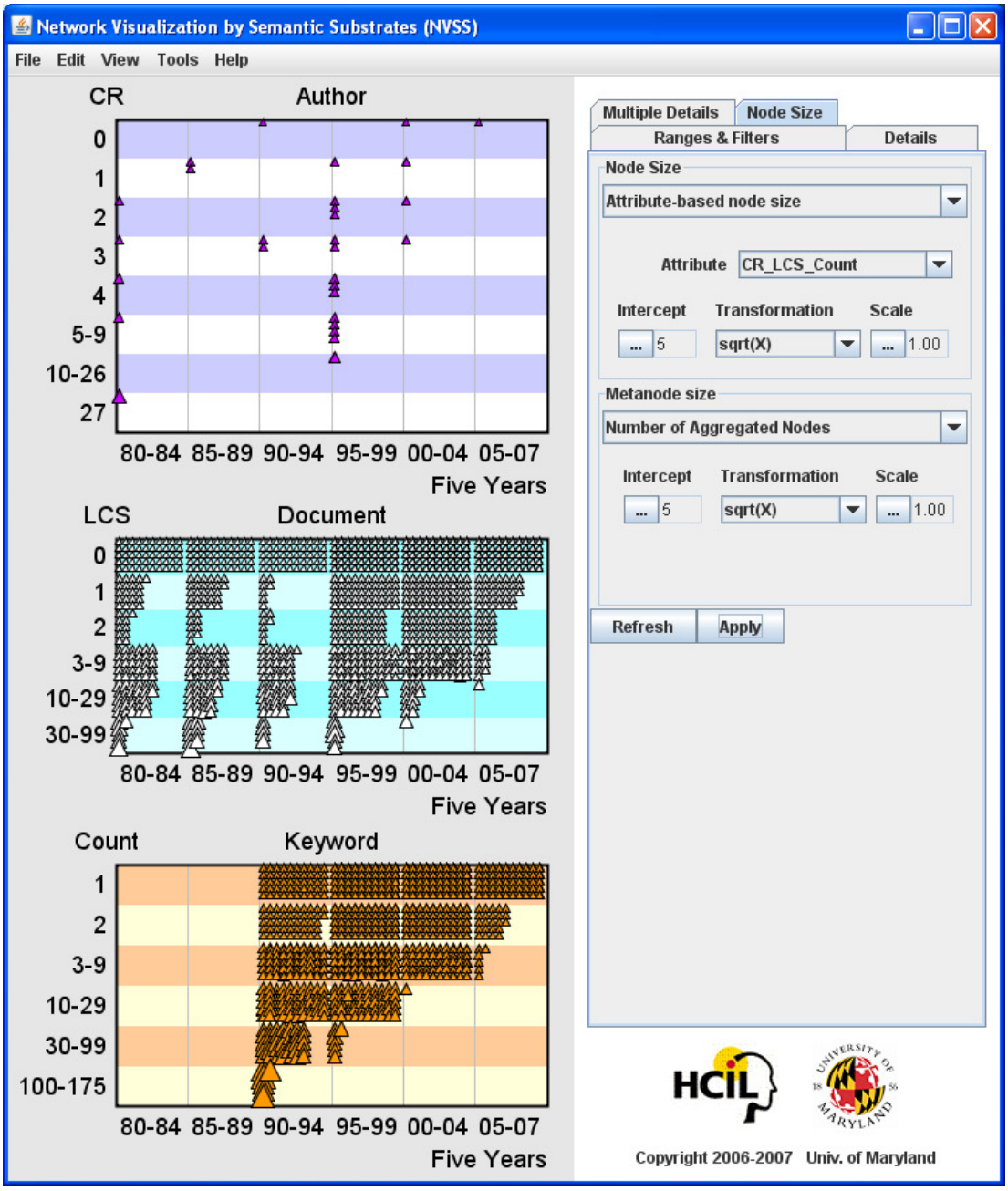


Figure 156 Using node size coding on TobIG dataset.

After this view, I said the triangles perhaps are too large and I suggested that Noshir C. tried 0.5 as the scale. Noshir C. did that and commented that the does not see much difference between triangle sizes suggesting that perhaps this setting was not distinguishing enough (see Figure 157).

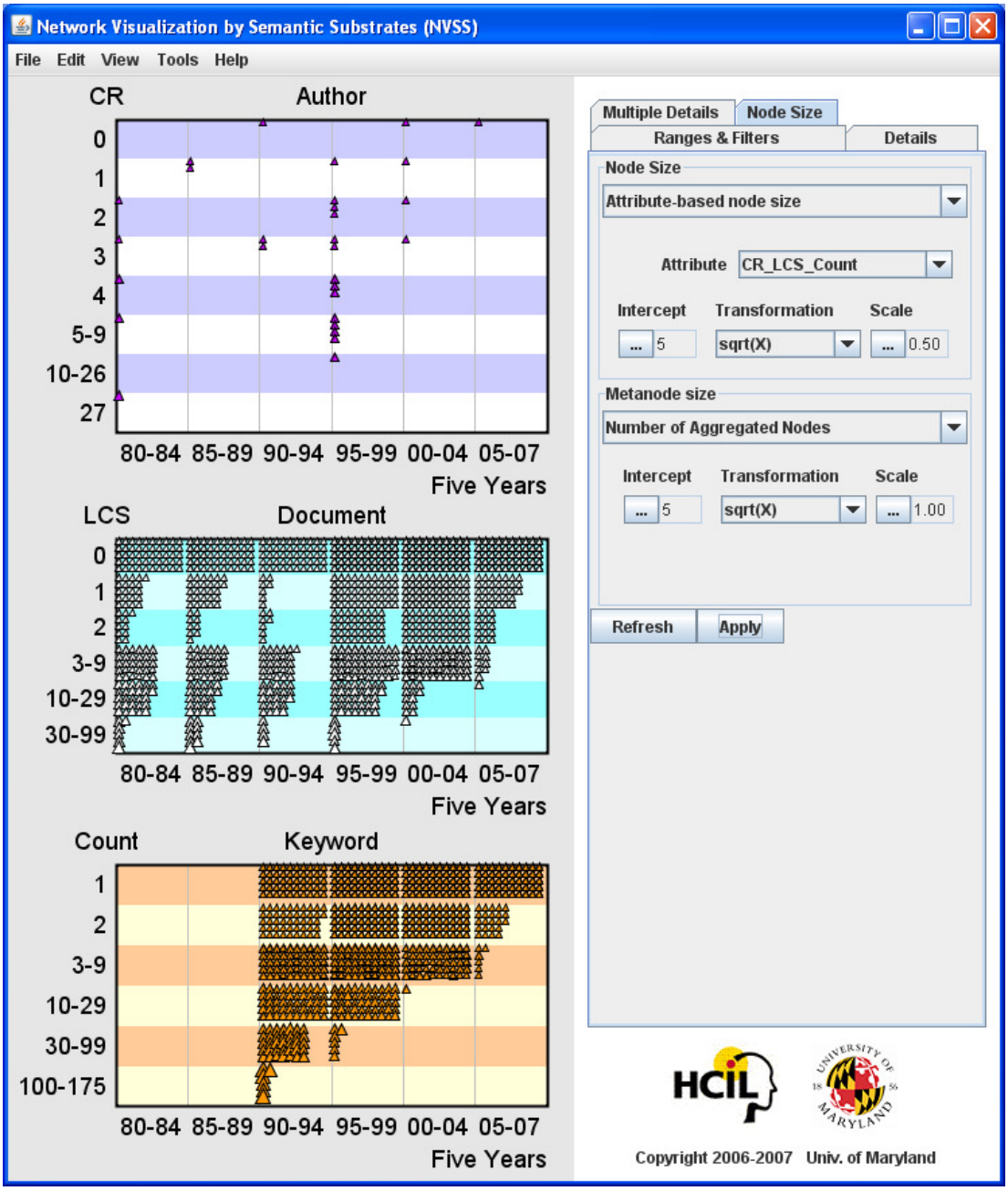


Figure 157 Setting node size using scale 0.5.

Outcome

This case study provided a worthwhile exploration process for a scholarly paper citation dataset that promises a technique for this kind and other kinds of dataset. Our participant is eager to explore other datasets, provide feedback to improve the tool to be used by other users under funded agreements. Through those collaborations results of exploration on real various data has a high chance to contribute to scholarly publications.

Bibliography

Agrafiotis, D. K. (1997). "A new method for analyzing protein sequence relationships based on Sammon maps." Protein Science **6**: 287-293.

Ahlberg (CEO of Spotfire), C. (2008). Influence of NVSS concepts in Spotfire**:** Personal communication.

Ahlberg, C. (1996). "Spotfire: an information exploration environment." ACM SIGMOD Record **25**(4): 25-29.

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

Archambault, D., Tamara Munzner, and David Auber (2007). Grouse: Feature-Based, Steerable Graph Hierarchy Exploration. Eurographics/IEEE-VGTC Symposium on Visualization**:** 67-74.

Aris, A., Ben Shneiderman (2007). "Designing Semantic Substrates for Visual Network Exploration." Information Visualization Journal **6**(4): 281-300.

Aris, A., Ben Shneiderman (2008). "A Node Aggregation Strategy to Reduce Complexity of Network Visualizaton using Semantic Substrates." HCIL Technical Report, University of Maryland **HCIL-2008-10**: 1-8.

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), Rome, Italy, Springer Berlin / Heidelberg.

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

Auber, D., Y. Chiricota, F. Jourdan, G. Melancon (2003). Multiscale visualization of small world networks. IEEE Symposium on Information Visualization (InfoVis 2003)**:** 75-81.

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.

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

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. Jünger, 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.

Brose, U., Lara Cushing, Eric L. Berlow, Tomas Jonsson, Carolin Banasek-Richter, Louis-Felix Bersier, Julia L. Blanchard, Thomas Brey, Stephen R. Carpenter, Marie-France Cattin Blandenier, Joel E. Cohen, Hassan Ali Dawah, Tony Dell, Francois Edwards, Sarah Harper-Smith, Ute Jacob, Roland A. Knapp, Mark E. Ledger, Jane Memmott, Katja Mintenbeck, John K. Pinnegar, Bjoern C. Rall, Tom Rayner, Liliane Ruess, Werner Ulrich, Philip Warren, Rich J. Williams, Guy Woodward, Peter Yodzis, and Neo D. Martinez (2005). "Body sizes of consumers and their resources." Ecology **86**(2545): Ecological Archives E086-135.

Buja, A., Deborah F. Swayne, Michael L. Littman, Nathaniel Dean, Heike Hofmann, Lisha Chen (2008). "Data Visualization With Multidimensional Scaling." Journal of Computational and Graphical Statistics **17**(2): 444-472.

Buono, P., Catherine Plaisant, Adalberto Simeone, Aleks Aris, Ben Shneiderman, Galit Shmueli, Wolfgang Jank (2007). Similarity-Based Forecasting with Simultaneous Previews: A River Plot Interface for Time Series Forecasting. Proc. of the 11th International Conf. on Information Visualisation (IV'07)**:** 191-196.

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.

Dengler, E. F., J. M. Marks (1993). Constraint-driven diagram layout. Proceedings of IEEE Symposium on Visual Languages, Bergen, Norway, IEEE Computer Society Press.

Di Battista, G., P. Eades, R. Tamassia, and I.G. Tollis (1999). Graph Drawing: Algorithms for visualization of graphs, Prentice Hall.

Dwyer, T., and Yehuda Koren (2005). Dig-CoLa: Directed Graph Layout through Constrained Energy Minimization. Proceedings of the 2005 IEEE Symposium on Information Visualization. Minneapolis, MN, IEEE Computer Society Washington, DC, USA 65-72.

Dwyer, T., Yehuda Koren, and Kim Marriott (2006). "IPSep-CoLa: An Incremental Procedure for Separation Constraint Layout of Graphs." IEEE Transactions on Visualization and Computer Graphics **12**(5): 821-828.

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.

Fekete, J.-D., Catherine Plaisant (1999). Excentric labeling: dynamic neighborhood labeling for data visualization. Conference on Human Factors in Computing Systems 512-519.

Fowler, M., Kent Beck, John Brant, William Opdyke, Don Roberts (1999). Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional.

Fruchterman, T. M. J., and E.M.Reingold (1991). "Graph Drawing by Force-directed Placement." Software-Practice and Experience **21**(11): 1129-1164.

Gamma, E., Richard Helm, Ralph Johnson, and John Vlissides (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional.

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.

Gersh, J., Bessie Lewis, Jaime Montemayor, Christine Piatko, Russell Turner (2006). "Supporting Insight-Based Information Exploration in Intelligence Analysis " Communications of the ACM **49**(4): 63-68.

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). Austin, Texas, IEEE Computer Society Washington, DC, USA 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 (AVI'2000)**:** 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.

He, W., and Kim Marriott (1998). "Constrained Graph Layout." Constraints **3**(4): 289-314.

Heer, J., and D. Boyd (2005). Vizster: Visualizing Online Social Networks. IEEE Symposium on Information Visualization.

Henry, N., and Jean-Daniel Fekete (2006). "MatrixExplorer: a Dual-Representation System to Explore Social Networks." IEEE Transactions on Visualization and Computer Graphics **12**(5): 677-684.

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.

Hirsch, J. E. (2005). "An index to quantify an individual's scientific research output." Proceedings of the National Academy of Sciences of the United States of America **102**(46): 16569-16572.

Holten, D. (2006). "Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data." IEEE Transactions on Visualization and Computer Graphics **12**(5): 741-748.

Huffaker, B., Evi Nemeth, and K. Claffy (1999). Otter: A general-purpose network visualization tool. Internet Society INET'99 Conference.

Johnson, B., Ben Shneiderman (1991). "Tree-Maps: a space-filling approach to the visualization of hierarchical information structures." Proceedings of the 2nd conference on Visualization '91: 284-291.

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.

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

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

Kobsa, A. (2004). User Experiments with Tree Visualization Systems. Proceedings of IEEE Symposium on Information Visualization (InfoVis 2004)**:** 9-16.

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.

Lamping, J., Ramana Rao, Peter Pirolli (2005). A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. Proc. ACM Conf. Human Factors in Computing Systems (CHI 2005)**:** 401-408.

Lee, B., Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry (2006). Task taxonomy for graph visualization. Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization. Venice, Italy, ACM New York, NY, USA**:** 1-5.

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.

Lieberman, M. D., Inbal Yahav, Sima Taheri, Huimin Guo, and Fatemeh Mir Rashed (2008). Visual Exploration of Biomedical Databases. CMSC 734 (Information Visualization) Term Project Report, University of Maryland**:** 1-9.

Martin-Merino, M., Alberto Munoz (2004). "A New Sammon Algorithm for Sparse Data Visualization." Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) **1**: 477-481.

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.

Munzner, T., F Guimbretiere, G Robertson (1999). Constellation: A Visualization Tool for Linguistic Queries from MindNet. The 1999 IEEE Symposium on Information Visualization. San Francisco, CA**:** 132-135 + 154.

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.

O'Madadhain, J., Danyel Fisher, Padhraic Smyth, Scott White, Yan-Biao Boey (2008). Analysis and Visualization of Network Data using JUNG**:** 1-35.

Perer, A. (2008). Integrating Statistics and Visualization to Improve Exploratory Social Network Analysis. Dept. of Computer Science. College Park, University of Maryland. **Ph.D.**

Perer, A., and Ben Shneiderman (2006). "Balancing Systematic and Flexible Exploration of Social Networks." (Proceedings of IEEE Visualization/Information Visualization) IEEE Transactions on Visualization and Computer Graphics **12**(5): 693-700.

Perer, A., Ben Shneiderman (2008). Integrating Statistics and Visualization: Case Studies of Gaining Clarity during Exploratory Data Analysis. ACM Conference on Human Factors in Computing Systems (CHI 2008)**:** 265-274.

Phan, D., Ling Xiao, Ron Yeh, Pat Hanrahan, and Terry Winograd (2005). Flow map layout. Proceedings of the 2005 IEEE Symposium on Information Visualization. Minneapolis, MN, IEEE Computer Society Washington, DC, USA 219-224.

Plaisant, C. (2004). The Challenge of Information Visualization Evaluation. Proceedings of Conference on Advanced Visual Interfaces, AVI'04 (2004)**:** 109-116.

Plaisant, C., Jesse Grosjean, Benjamin B. Bederson (2002). SpaceTree: Supporting exploration in large node-link tree: design evolution and empirical evaluation. IEEE Symposium on Information Visualization (InfoVis'2002)**:** 57-64.

Pretorius, A. J., Jarke J. van Wijk (2005). Multidimensional Visualization of Transition Systems. Proceedings of the Ninth International Conference on Information Visualization. London, UK, IEEE Computer Society Washington, DC, USA 323-328.

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.

Ryall, K., Joe Marks, and Stuart M. Shieber (1997). An Interactive Constraint-Based System for Drawing Graphs. ACM Symposium on User Interface Software and Technology. Banff, Alberta, Canada, ACM New York, NY, USA 97-104.

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. (2005). Information Visualization Design for Multidimensional Data: Integrating the Rank-By-Feature Framework with Hierarchical Clustering.

Dept. of Computer Science. College Park, University of Maryland. **Ph.D.:** HCIL-2005-20, CS-TR-4745, UMIACS-TR-2005-48.
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. (1992). "Tree visualization with tree-maps: 2-d space-filling approach." ACM Transactions on Graphics (TOG), v.11 n.1 **11**(1): 92-99.
Shneiderman, B. (1999). "Dynamic queries, starfield displays, and the path to Spotfire." from http://www.cs.umd.edu/hcil/spotfire/.
Shneiderman, B., and Aleks Aris (2006). "Network Visualization by Semantic Substrates." (Proceedings of IEEE Visualization/Information Visualization) IEEE Transactions on Visualization and Computer Graphics **12**(5): 733-740.
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. Venice, Italy, ACM New York, NY, USA 1-7.
Shrinivasan, Y. B., Jarke J. Van Wijk (2007). "VisPad: Integrating Visualization, Navigation and Synthesis " IEEE Symposium on Visual Analytics Science and Technology (VAST 2007): 209-210.
Sindre, G., B. Gulla, and H. Jokstad (1993). Onion Graphs: Aesthetic and Layout. Proceedings of the 1993 IEEE Symposium on Visual Languages. Bergen, Norway, IEEE Computer Society Press**:** 287-291.
Storey, M. A., M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy (2001). Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protege. Workshop on Interactive Tools for Knowledge Capture. Victoria, B.C. Canada.
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.
Swayne, D. F., Andreas Buja, Duncan Temple Lang (2003). Exploratory Visual Analysis of Graphs in GGobi. Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003). Vienna, Austria.
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. Montréal, Québec, Canada, ACM New York, NY, USA 811-819.
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.