ABSTRACT

Title of Document: INTEGRATING STATISTICS AND VISUALIZATION TO IMPROVE EXPLORATORY SOCIAL NETWORK ANALYSIS.

Adam Perer, Doctor of Philosophy, 2008

Directed By: Professor Ben Shneiderman, Department of Computer Science

Social network analysis is emerging as a key technique to understanding social, cultural and economic phenomena. However, social network analysis is inherently complex since analysts must understand every individual's attributes as well as relationships between individuals. There are many statistical algorithms which reveal nodes that occupy key social positions and form cohesive social groups. However, it is difficult to find outliers and patterns in strictly quantitative output. In these situations, information visualizations can enable users to make sense of their data, but typical network visualizations are often hard to interpret because of overlapping nodes and tangled edges.

My first contribution improves the process of exploratory social network analysis. I have designed and implemented a novel social network analysis tool, *SocialAction* (http://www.cs.umd.edu/hcil/socialaction), that integrates both statistics and visualizations to enable users to quickly derive the benefits of both. Statistics are used to detect important individuals, relationships, and clusters. Instead of tabular display of numbers, the results are integrated with a network visualization in which users can easily and dynamically filter nodes and edges. The visualizations simplify the statistical results, facilitating sensemaking and discovery of features such as distributions, patterns, trends, gaps and outliers. The statistics simplify the comprehension of a sometimes chaotic visualization, allowing users to focus on statistically significant nodes and edges. SocialAction was also designed to help analysts explore non-social networks, such as citation, communication, financial and biological networks.

My second contribution extends lessons learned from *SocialAction* and provides designs guidelines for interactive techniques to improve exploratory data analysis. A taxonomy of seven interactive techniques are augmented with computed attributes from statistics and data mining to improve information visualization exploration. Furthermore, systematic yet flexible design goals are provided to help guide domain experts through complex analysis over days, weeks and months.

My third contribution demonstrates the effectiveness of long term case studies with domain experts to measure creative activities of information visualization users. Evaluating information visualization tools is problematic because controlled studies may not effectively represent the workflow of analysts. Discoveries occur over

weeks and months, and exploratory tasks may be poorly defined. To capture authentic insights, I designed an evaluation methodology that used structured and replicated long-term case studies. The methodology was implemented on unique domain experts that demonstrated the effectiveness of integrating statistics and visualization.

INTEGRATING STATISTICS AND VISUALIZATION TO IMPROVE EXPLORATORY SOCIAL NETWORK ANALYSIS.

By

Adam Nathaniel Perer

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
Professor Ben Bederson
Professor Lise Getoor
Professor Jen Golbeck
Professor Alan Neustadtl

© Copyright by

Adam Nathaniel Perer

2008

Dedication

Dedicated to my Mom and Dad,

for always letting me push buttons.

Acknowledgements

I first thank my advisor, Ben Shneiderman. He has been a vast source of wisdom, inspiration, and motivation throughout my graduate career. It is hard to imagine a more supportive, kind, and dedicated mentor than Dr. Shneiderman. I'd also like to thank my all-star committee for shaping and refining my dissertation: Ben Bederson, Lise Getoor, Jennifer Golbeck and Alan Neustadtl.

The Human-Computer Interaction Lab at the University of Maryland has been an amazing place to spend during my doctoral years. I thank the leadership of Allison Druin, the thoughtful Catherine Plaisant, the insightful François Guimbretière, the problem solver Anne Rose, and the helpful Kiki Schneider. The students of the HCIL have also played a pivotal role throughout my graduate career. In particular, Amy Karlson, Georg Apitz, Jinwook Seo, Dave Wang, Hyunyoung Song, Aleks Aris, Alex Quinn, Bill Kules, Haixia Zhao, Gene Chipman, Jerry Fails, Sherri Massey, Nick Chen, Bongshin Lee, Chang Hu and Chunyuan Liao deserve much praise for making the lab such an enlightening atmosphere. Other computer science students, Nikhil Swamy and Dave Levin, have also been critical to my success.

Outside of the University of Maryland, I've also had several wonderful research opportunities. I thank Marc Smith for continual support and inviting me to spend two wonderful summers at Microsoft Research. I thank Eric Bier for hosting me for a summer at Xerox PARC and allowing me to innovate in the place I admired most growing up. I also thank Christine Youngblut (Institute for Defense Analyses), Takeo Kanade (Carnegie Mellon University), Michael Widom (Carnegie Mellon

University), and Randy Beer (Case Western Reserve University), without whom I may have never learned the joy of research in computer science.

My friends and family deserve much of the credit for my graduate school success. I couldn't imagine more amazing parents than the ones I have, who have always given me the confidence and support to pursue what I love. My sister, Jennifer Slattery, is a constant source of love and pride. My significant other, Bita Azhdam, has made the last 18 months of my life among the best. The undying support of my friends can also not go unstated: Gregg Tarter, Sara Tadikamalla, Eric Dash, Ben Kelley, Michael Kalnicki, Matt Ittigson, Michael Rosenthal, Josh Baugher, Bethia Cullis, Ankita Singh, Jeremy Baugher, Andy Gordon, Atheir Abbas, Tim Fasel, Marisa Pauly, Julie Chung, Jaime Setton, Melissa Bowman, Ezi Yomtovian, Margo Serlin and Aaron Cohen. Finally, I thank Grammy, Nana, and Colin for looking out for me from above.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Insights are the measure of success for analysts. Analysts may be seeking to confirm their intuitions, detect anomalies or outliers, or uncover underlying patterns. Insights, as characterized by North in [62], are complex, deep, qualitative, unexpected and relevant findings. With the proliferation of data in digital form, analysts can analyze their data by using tools to computationally detect such patterns, gaps, and outliers. In support of such tasks, user interfaces for exploratory data analysis should be fluid and efficient. The most powerful sensory input for analysts, our human eyes, have more bandwidth and processing power than smell, sound, taste or touch. Presenting data through information visualizations is therefore an effective way to utilize the strong capabilities of human perceptual systems. However, choosing an effective presentation is challenging, as not all information visualizations are created equally. Not all information visualizations highlight the patterns, gaps and outliers important to analysts' tasks and, furthermore, not all information visualizations "force us to notice what we never expected to see" [89]. Well-designed interactive techniques are an effective strategy for making information visualizations more comprehensible for sensemaking. The goal of this dissertation is to provide interaction techniques that lead analysts to the discovery of greater insights during exploratory data analysis, particularly social network analysts.

This dissertation focuses on social network analysis because it is topical, emerging and an inherently challenging process that requires creative approaches to problem solving. It's difficult to visualize, navigate, and analyze networks, and most problematic, its difficult to find task-relevant patterns in networks. Despite all of

these challenges, the network perspective remains appealing to sociologists, intelligence analysts, communication theorists, bibliometricians, food-web ecologists, and numerous other professionals. The growing popularity of social network analysis (SNA) can be seen in , and inspired by, popular best-selling books such as Malcolm Gladwell's, "The Tipping Point", Albert-László Barabási's, "Linked", and Duncan Watt's "Six Degrees." Countless analysts wish to analyze their network data, but there are few mature or widely used tools and techniques.

Network analysts focus on relationships instead of just the individual elements which can explain social, cultural, or economic phenomena; how the elements are connected is just as important as the elements themselves. Prior to the social network analysis perspective, many analysts focused largely on inherent individual attributes and neglected the social facet of behavior (how individuals interact and the influence they have on each other) [23]. Using newer techniques from the social network community, analysts can find patterns in the structure, witness the flow of resources or messages through a network, and learn how individuals are influenced by their surroundings.

In practice, social network visualizations can be chaotic, particularly when the network is large. Visualizations are useful in leveraging the powerful perceptual abilities of humans, but cluttered presentations, overlapping edges and illegible node labels often undermine the benefits of visual exploration. In these situations, interactive techniques are necessary to make sense of such complex static visualizations. Interactions such as zooming, panning or filtering by inherent attributes can simplify complex visualizations. However, even such techniques only

get users so far with complex networks, particularly in small-world networks where dense connections will rarely untangle [91]. For situations like this, dynamic queries and filtering on *inherent* attributes of nodes and edges remain important strategies. Inherent attributes are the attributes that exist in the data set, such as gender, race, salary, or education level. However, *inherent* attributes may not tell the whole story. Inherent attributes lack the structural, topological information critical to social network analysts. A major contribution of my dissertation is to augment information visualizations with *computed* attributes that reflect the tasks of users. *Computed* attributes can be calculated from relevant statistical importance metrics (e.g. degree or betweenness centrality), clustering algorithms, or data mining strategies.

This approach is particularly valuable for social network analysis, as they have also come to believe that inherent attributes do not tell the whole story. In fact, a philosophy shared by many social network analysts is to ignore inherent attributes during exploration to avoid bias, and only focus on the data's structural properties. For social network analysts, computed attributes can be calculated with a rich set of statistical techniques - from sociology to graph theory - that allow analysts to numerically uncover interesting features within their networks. Analysts might seek a tight-knit community of individuals, or the gatekeepers between them, or the most centrally powerful entities; there are a variety of sophisticated algorithms for finding these traits.

1.1 Contribution (C1:Integration)

Provides an integration of statistics and visualization to improve exploratory social network analysis.

Most visualization tools aim to project complex data into comprehensible views. But few tools assist users by providing computed attributes that highlight important properties of their data. Users can switch back and forth between statistical and visualization packages, but this can result in an inefficient flow in the analysis process, which inhibits discovery.

SocialAction is the software tool created to explore these issues (http://www.cs.umd.edu/hcil/socialaction). It provides meaningful, computed attributes on the fly by integrating both statistics and visualizations to enable users to quickly derive the benefits of both. *SocialAction* embeds statistical algorithms to detect important individuals, relationships, and clusters. Instead of presenting statistical results in typical tabular fashion, the results are integrated with a network visualization while providing meaningful computed attributes of the nodes and edges. With computed attributes, users can easily and dynamically filter nodes and edges to find discover interesting data points. The visualizations simplify the statistical results, facilitating sensemaking and discovery of features such as distributions, patterns, trends, gaps and outliers. The statistics simplify the comprehension of a sometimes chaotic visualization, allowing users to focus on statistically significant nodes and edges. The presence of these rich interactions within one consistent interface provide a fluid, efficient, visual analytic system that allows users to focus on insights and generating hypotheses rather than managing a medley of software packages. Furthermore, although *SocialAction* is designed to support social network analysis, it also allows users to explore and interpret non-social networks, such as citation, communication, financial and biological networks.

1.2 Contribution (C2:Guidelines)

Provides design guidelines for interactive techniques to improve exploratory data analysis with computed attributes and systematic yet flexible guides.

As data sets increase in size and complexity, static information visualizations decrease in comprehensibility. Interactive techniques are often necessary to yield valuable discoveries, but current data analysis tools typically support only opportunistic exploration - which may be inefficient and incomplete. That said, interactive techniques do not get as much attention in the information visualization community despite their growing need to produce comprehensible visualizations [106].

Providing computed attributes is not enough to guarantee a successful information visualization tool, but they should be integrated into rich interaction capabilities usually reserved for inherent attributes. The interaction taxonomy of Yi et al. [59], I demonstrate how computed attributes can be incorporated into the design for tools beyond social network analysis.

Furthermore, although interactive techniques are often necessary to yield valuable insights, interaction techniques typically only support opportunistic exploration that may be inefficient and incomplete. To resolve this, I present a refined architecture that uses *systematic yet flexible* (SYF) design goals to guide domain expert users through complex exploration of data over days, weeks and months. The SYF system aims to support exploratory data analysis with some of the simplicity of an e-commerce check-out while providing added flexibility to pursue novel insights. The SYF system provides an overview of the analysis process,

suggests unexplored states, allows users to annotate useful states, supports collaboration, and enables reuse of successful strategies. The affordances of the SYF system are demonstrated by integrating it into *SocialAction*.

The methods are complete and repeatable, so if two analysts are presented with the same data, they should reach the same conclusion. However, in the field of social networks, different networks need to be analyzed differently. The spread of an epidemic among villages is not necessarily the same as a spread of a financial crisis on world markets [102]. Since there is no systematic way to interpret networks, users also need to be able to flexibly explore features to discover patterns. The SYF design goals support this philosophy.

1.3 Contribution (C3:Evaluation)

Demonstrates the effectiveness of long term case studies with domain experts to measure creative activities of information visualization users.

Although computer applications, such as *SocialAction*, shift from productivity support to creativity support [83], research evaluation methods are still predominantly based on older strategies. Controlled experiments with dependent variables such as time for performance of benchmark tasks are still valuable, but they may be inadequate to study tools that support creative exploration [70, 75]. These new tools may require substantial learning, changes to problem-solving strategies, and exploratory use of tactics that defy controlled experimentation.

New research evaluation methods based on ethnographic observation and longitudinal study are being refined to meet the needs of these type of tools [84]. I designed a novel procedure to conduct Multi-dimensional In-depth Long-term Case

studies (MILCs) with academic and professional social network practitioners. These long-term case studies shift the strategy to working with small numbers of domain experts over longer time periods. MILCs with a political analyst, bibliometrician, healthcare consultant, counter-terrorism researcher and an online news magazine help demonstrate how my design of integrating statistics and visualization improves the exploratory data analysis process of social network analysis.

These contributions are a constant theme throughout the dissertation. Chapter 2 describes the work most closely related to my dissertation, including social network analysis, network and graph visualization, guided exploration, and methodologies for evaluation. Chapter 3 describes the contribution of integrating statistics and visualization, and demonstrates how the tasks of social network analysis are integrated into *SocialAction* (C1:Integration). Chapter 4 describes general design goals for information visualization systems that integrate statistics and visualization with computed attributes (C2:Guidelines). Most notably, there are *systematic yet flexible* (SYF) design goals to guide domain expert users through complex exploration of data over days, weeks and months. Chapter 5 describes a novel evaluation methodology for information visualization systems with long-term case studies and its implementation on studying the use of *SocialAction* (C3:Evaluation). The dissertation culminates in demonstrating that there are effective ways to improve exploratory data analysis by using statistical techniques and information visualizations in an integrated manner.

Chapter 2: Related Work

This dissertation builds on work from social network analysis as practiced in sociology and related disciplines. This work aims to provide powerful interactive computing tools for analysts in a growing set of disciplines to make significant discoveries concerning relationships in their data. This chapter focuses on tools used for network and graph visualization, as this dissertation has a substantial visual and statistical components. A second key topic for this chapter is systems that help guide users through a discovery process that might take weeks and months. The third component of this chapter is a review of methodologies for evaluating interactive visualization tools, especially those used for discovery.

2.1 Social Network Analysis (SNA)

Freeman suggests that social network analysts seek to uncover two types of patterns in networks: (1) those that reveal subsets of nodes that are organized into cohesive social groups, and (2) those that reveal subsets of nodes that occupy equivalent social positions, or roles [24]. There is a large body of work that presents techniques and methods developed over the past 60 years to uncover such patterns. *Social Network Analysis: Methods and Applications*, by Wasserman and Faust, is perhaps the most widely used reference book for structural analysts [99]. The book presents a review of network analysis methods and an overview of the field.

Using visualizations to assist in SNA is not a new concept for sociologists. Visual images can be used to examine the patterning of network data [24]. A history of the use of visual images in social networks is described in [25], including one of the earliest known examples of a social network visualization by Jacob Moreno in

1934. In Figure 1, the triangle nodes are boys and the circle nodes are girls. Without knowing any details about who the individuals in this classroom on, one quickly learns from the visualization that 1) boys are friends with boys, 2) girls are friends with girls, 3) one brave boy chose a girl as his friend which wasn't reciprocated and 4) there are an isolated group of two girls. This visualization shows how a legible and well positioned network can explain the social structure of individuals.

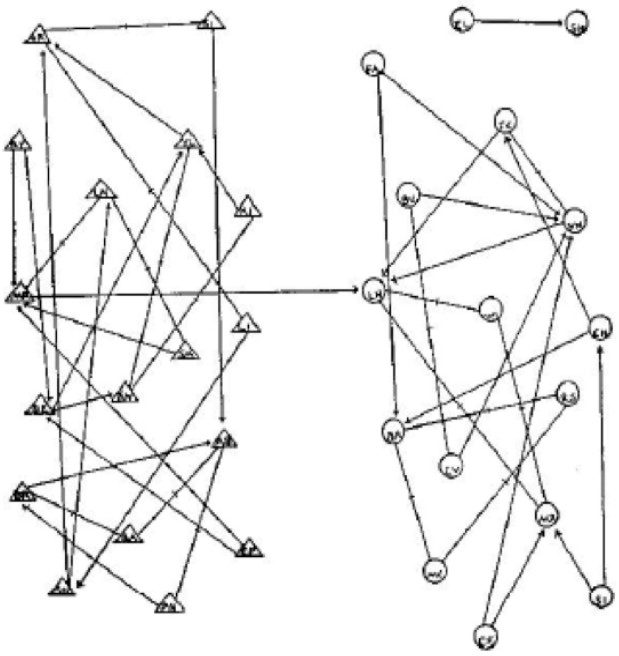


Figure 1. One of the earliest known examples of a social network visualization by Jacob Moreno in 1934. This network illustrates the friendship choices among fourth graders. The triangular nodes are male, and the circular nodes are female.

In order to provide social scientists with the statistical and visualization techniques proven to be effective in analysing networks, many software tools have been developed. I provide a brief review of the two most popular SNA packages, *UCINET* [9] and *Pajek* [17]. Huisman and van Dujin provide an overview of six software tools available for social network analysis [47]. De Nooy et al. also wrote a textbook that integrates theory, applications and professional software [17].

2.2 SNA Tools

According to interviews, two of the most popular tools used by sociologists for social network analysis are *UCINET* [9] and *Pajek* [17]. Each of these tools features an impressive number of feature to measure social networks, grounded in the theory and techniques of sociologists. However, these numerous features are not necessarily organized to support the tasks of the users. In general, the programs are designed as shown in Figure 2.

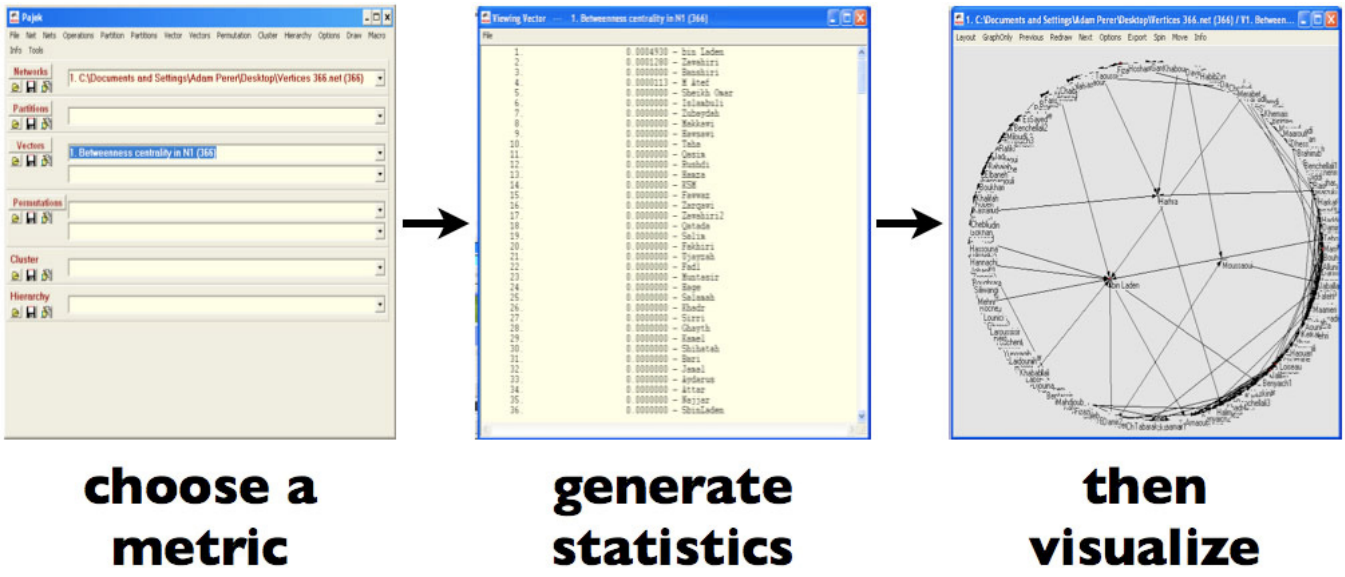


Figure 2. The Methodology of Popular SNA Tools. In this particular example, I demonstrate how exploratory data analysis is performed in Pajek. When the users wish to measure a network,they select a statistical measure from a menu-driven interface. After selecting the algorithm, the results of the analysis are displayed in a text-based view. If the users wish to see a visualization of the results, they can then generate a network view. If users wish to perform a different statistical measure, the user needs to start the process all over again.

In the existing methodology, as practiced in *UCINET*, users begin by performing a statistical analysis of the network. Then they output the results to a text window and a file, which is exported to a drawing tool, such as *NetDraw* which is typically bundled with *UCINET*. If the users wish to run another analysis, they must close i, open the original network file, run the new method, and the re-export to *Netdraw*. *UCINET* can handle networks with up to 32,000 nodes and *Netdraw* can support 10,000 nodes. However, the author admits that networks of these sizes run extremely slow in both packages.

Pajek was built to overcome *UCINET*'s limitation on network size, and it describes itself as a "program for large network analysis". Its interface resembles *UCINET*, inasmuch as it organizes its analysis methods in deep, hierarchical menus and outputs all analysis to a textual report screen. Unlike *UCINET*, *Pajek* has a built-in graph visualizer. However, this tool is not interlinked either. After an analysis procedure has been run, one can call the 'draw' command to the see the network in a draw screen. However, it is not possible to use the visualization as a point of departure for future analysis, e.g. selecting a portion of the graph to run additional analysis.

These tools have been a breakthrough for the sociologist community because they no longer require structural analysts to implement their own algorithms. However, social network analysis is inherently a deductive task and these interfaces do not support exploratory data analysis well due to the strong disconnect between the statistical and visual analysis. I believe these are reasons for analysts rarely using

visualizations during the exploratory phase. Instead, they mostly use visualizations for communication purposes at the end of the analysis.

2.3 Network and Graph Visualization

The visualization of networks is important because it is a natural way to communicate connectivity and allows for fast pattern recognition by humans. However, there are great challenges when visualizing networks [19, 43]. There are many layout algorithms that attempt to calculate the position of each node and the curve of each link to minimize link crossings and adhere to aesthetic principles. These algorithms fall short, however, when the number of nodes is larger than several hundred and the large number of overlapping links makes it hard to judge connectivity [90]. Herman et al. provide a survey of layout and interaction techniques for information visualizations of graph [43]. They compile an impression of the limitations of various graph layout algorithms, approaches to navigation of large graphs, and methods of reducing visual complexity.

In Viegas and Donath's critique of the illegibility usually associated with social networks, they suggest two principles adapted from cartography: (1) adaptive zooming and (2) multiple viewing modes [94]. By using principles of good layout and visual perception, graphs can more effectively capture social network phenomena.

Mark Lombardi used social networks in his artwork to present financial and political scandals [57]. The drawings are accessible to his audience because actors never overlap, edges rarely cross, and the connections are smooth and curvy (Figure

3). Studies of visual perception also confirm that it is easier to follow connecting lines that are curvilinear smooth [97].

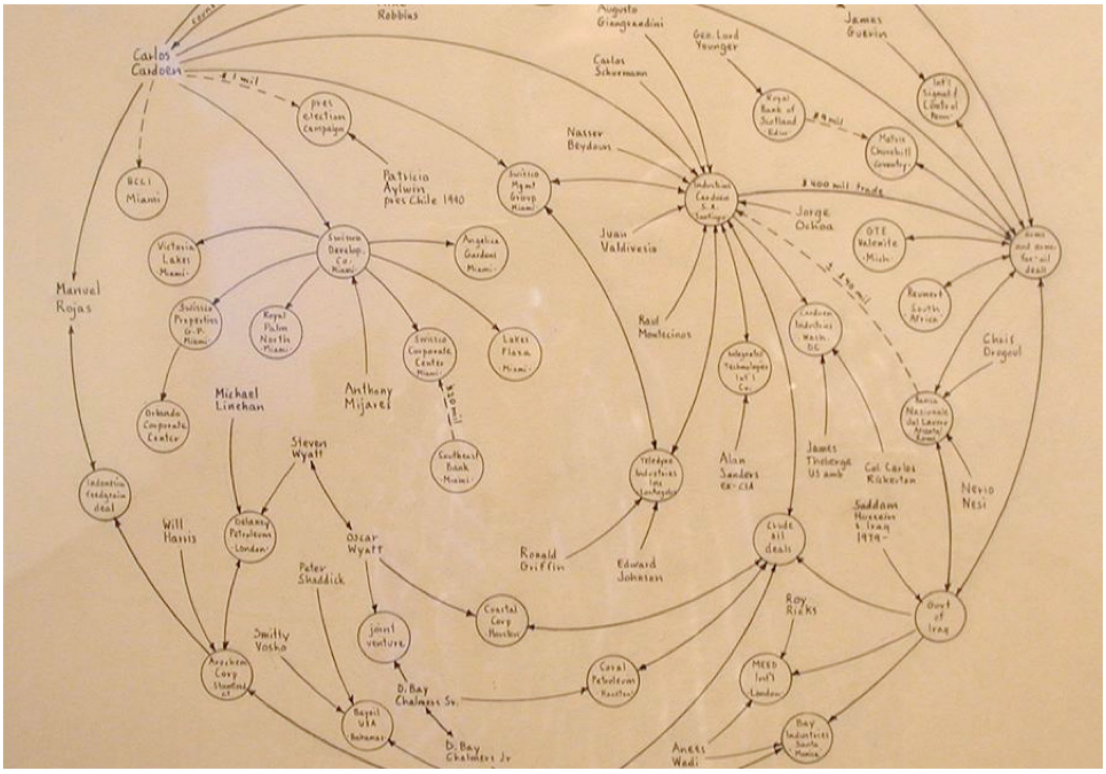


Figure 3. Example's of Mark Lombardi's social network artwork. The top graphic depicts relationships pertinent to Oliver North and the Iran-Contra affair.

2.4 Layout Distortion Techniques

Several approaches also attempt to more efficiently use available display space by distorting the graph. One such distortion technique is the use of fisheyes (Figure). Fisheye techniques allow users to examine a focus area in great detail, but also tend to obscure the global structure of networks, e.g. [54, 59].

Pirolli et al. studied the effects of focus+context techniques in a hyperbolic tree behavior [69]. They concluded that strong information scents, which are cues to guide browsing, improves visual search. However, crowding of nodes in a compressed region degrades visual search, especially when there is a weak information scent.

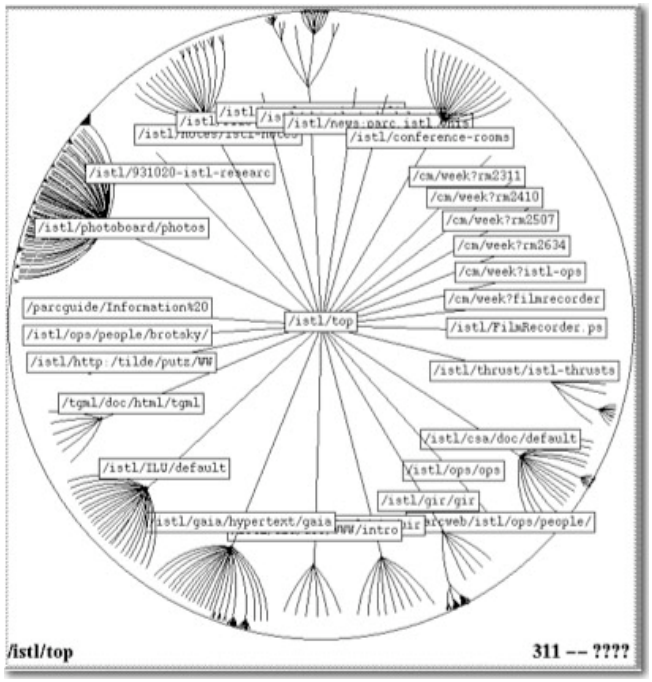


Figure 4. Fisheye techniques, such Lamping and Rao's Hyperbolic Tree Browser [54] allows users to focus on a specific area but obscures the global structure.

Multiscale graph abstraction is another technique that preserves global structure by showing small-scale and large-scale structure simultaneously (Figure). However, navigation is often difficult because clusters are explicitly contracted and expanded, e.g. [5, 64].

Recent work combines these two approaches with topological fisheye views to reduce the number of displayed nodes while preserving the network structure [27]. Van Ham and van Wijk also combine distortion strategies for highly connected, small-world networks [91], as shown in Figure.

Even with these layout and distortion techniques, it still seems ineffective to show the entire network at once. There have been a variety of systems that allow users to interactively explore networks, which I review in the next section.

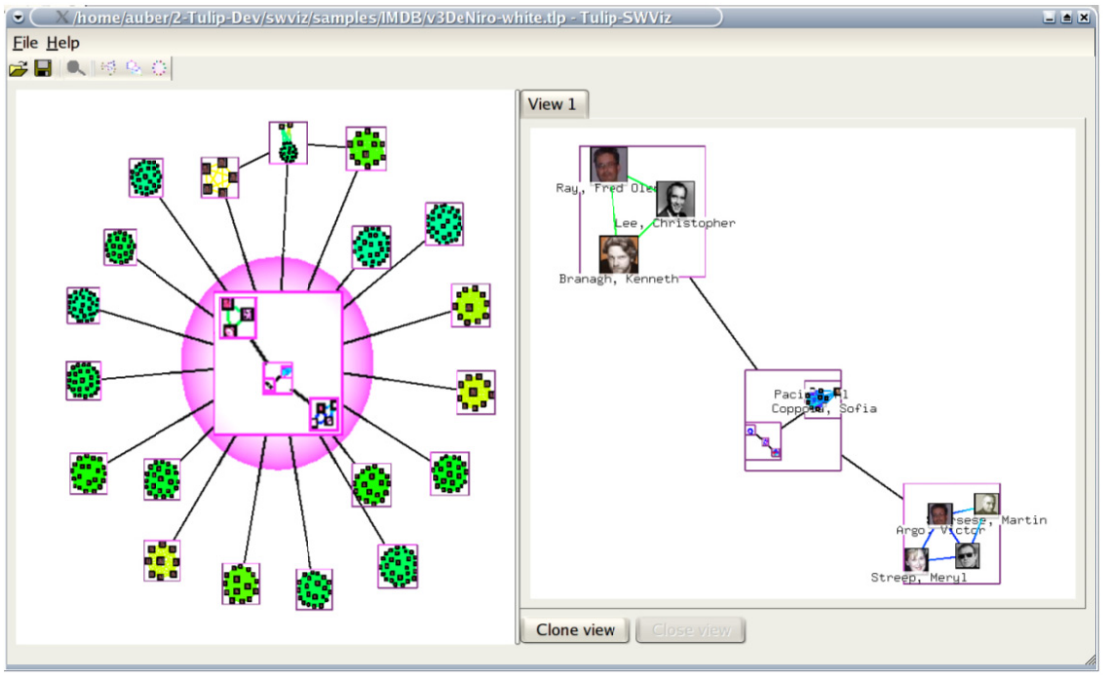


Figure 5. Auber et al's SWViz [5], which shows an overview of the network on the left along with a selected component (pick background) in greater detail on the right.

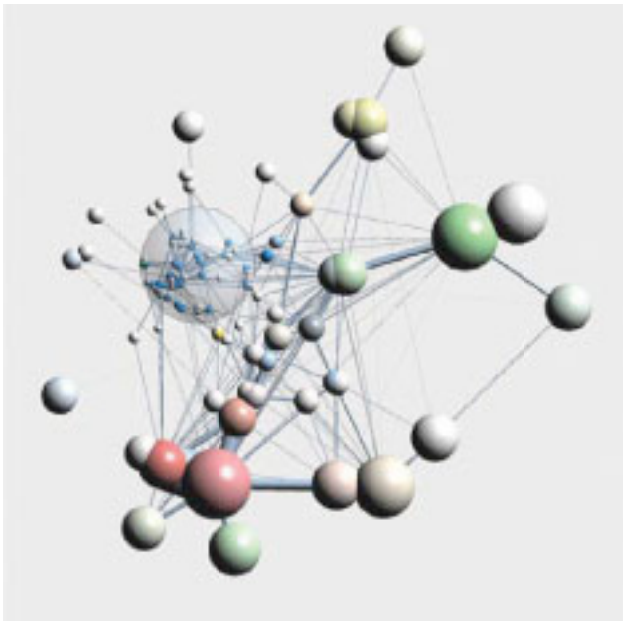


Figure 6. Van Ham et al's small-world network of IEEE Visualization conference proceedings [91]. The focus lens is on the blue portion of the network, which is the 'volume visualization' cluster of work, while the visualization preserves the global network structure.

2.5 Interactive Techniques

GUESS is a novel graph exploration system that combines an interpreted language with a graphical front end [1]. GUESS attempts to combine analysis and visualization of graphs into one package to improve interactive exploration. The system requires learning a domain-specific embedded language, Gython. However, once learned, users can type commands and control the visualization of the graph to show attributes of interest. Programmers can also use GUESS to rapidly deploy visualizations to analysts, and limit the menus to what is useful for the task at hand. However, this requires a customized solution for each scenario.

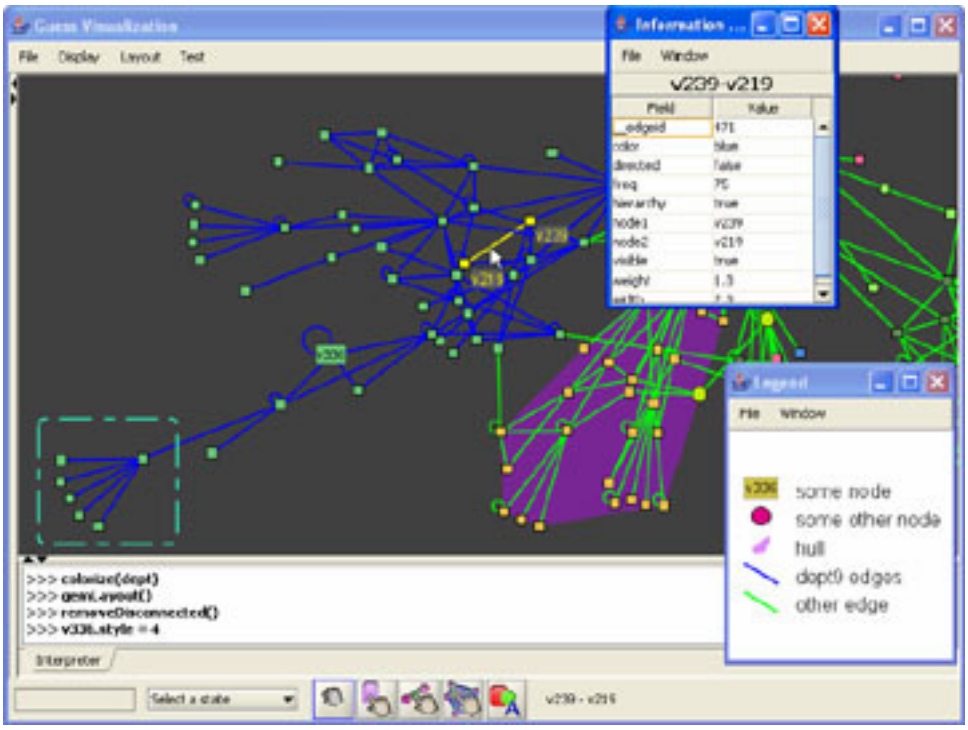


Figure 7. GUESS [1] is an exploratory data analysis system for networks that requires users to learn a domain-specific embedded language.

TreePlus allows users to explore graphs using more comprehensible enhanced tree layouts [55]. Lee et al. suggest using trees extracted from networks to aid users' ability to navigate. Their implementation, *TreePlus*, increases the readability and stability of graphs using a tree representation. The technique works best when users are interested in a local portion of a network – the overall structure or existence of clusters or bridges cannot be easily revealed. *TreePlus* also outperforms force-directed layouts for several tasks.

NetLens allows users to explore an actor-event network yet avoids using graphs and trees as the driving visualization [50]. Instead, *NetLens* uses histograms and iterative queries to help users understand the network. This approach seems to be especially effective when nodes have rich textual attributes.

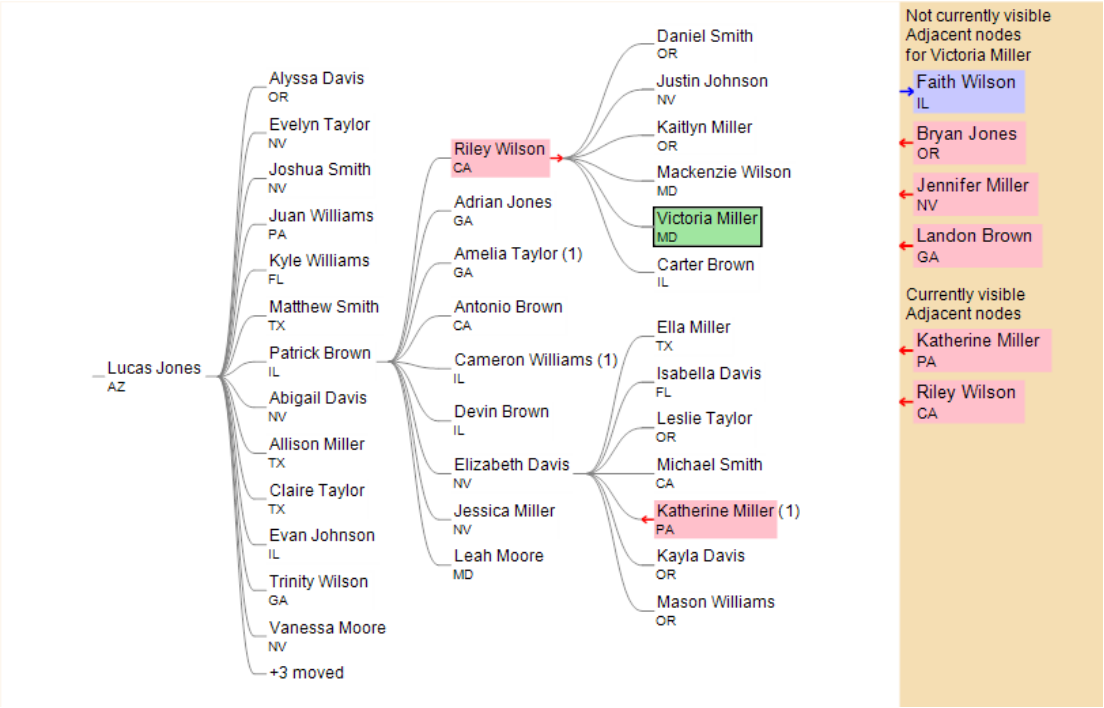


Figure 8. TreePlus is an interactive graph visualization system based on a tree-style layout. TreePlus transforms graphs into trees and shows the missing graph structure with visualization and interaction techniques. Here, a social network is visualized in TreePlus.

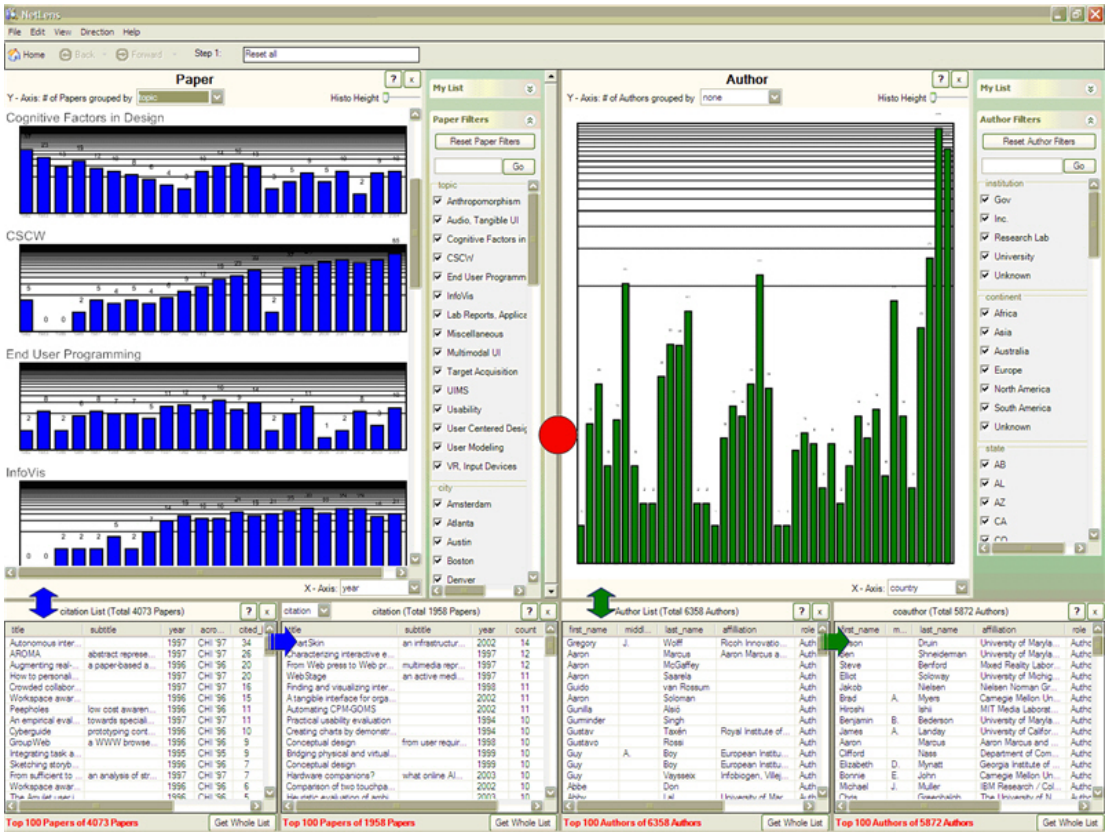


Figure 9. The NetLens system visualizing the citation network of the ACM Digital Library.

There have been several matrix-based approaches to visualizing networks, e.g. Figure 10 Ghoneim et al. presented the promise of using matrix-based visualizations instead of node-link diagrams [29]. They demonstrated that matrices outperform node-link diagrams for several tasks.

Ziegler et al. designed Matrix Browser, which uses an interactive matrix display to show relationships in a network [108]. The simplest matrix has all nodes on both axis, but the system also allows the axis to be flexibly filtered based on node attributes or properties of the relations in the grid. This system was further designed to support networks with hierarchical information structures, so the axis had tree widgets that users could collapse to reduce that amount of visually displayed material. In two small studies of small networks (N=27), the matrix visualization was 2.5 times faster than standard network visualizations for visual search tasks.
MatrixExplorer is a recent system designed for exploring social networks using a matrix visualization as the primary view [41]. After participatory design sessions with social scientists, they devised a list of requirements including matrix views, filtering, clustering, and interactive parameter tuning. Their tool attempts to empower users for several of these tasks, such as reordering matrices, interactive clustering and comparing clusters.

NodeTrix uses a hybrid approach of node-link diagrams, which show the structure of a network, and adjacency matrices, which highlight communities [42].

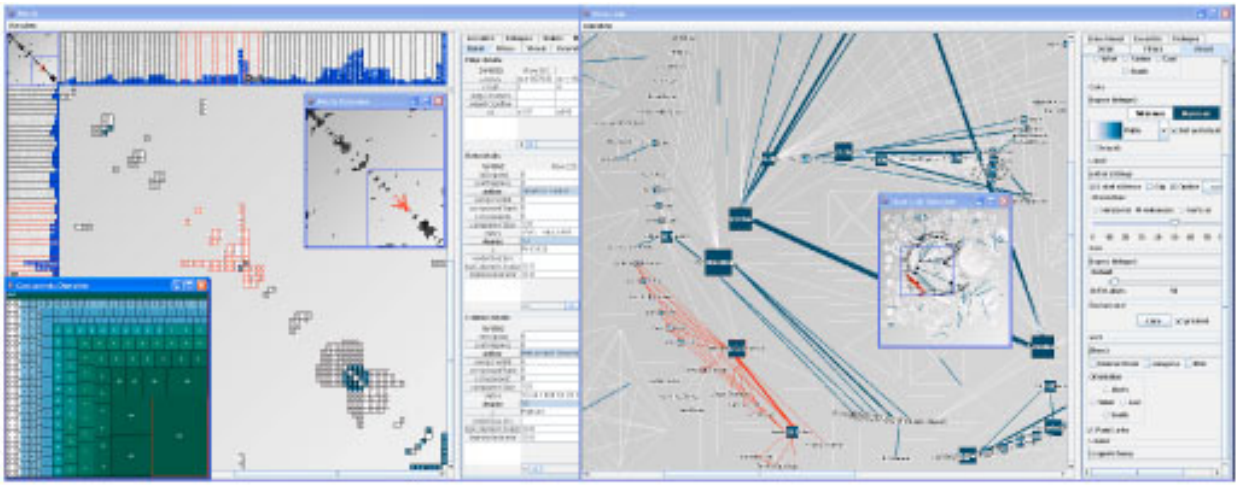


Figure 10. The MatrixExplorer system shows two synchronized views of a network. The matrix view is show on the left, while the node-link diagram is on the right.

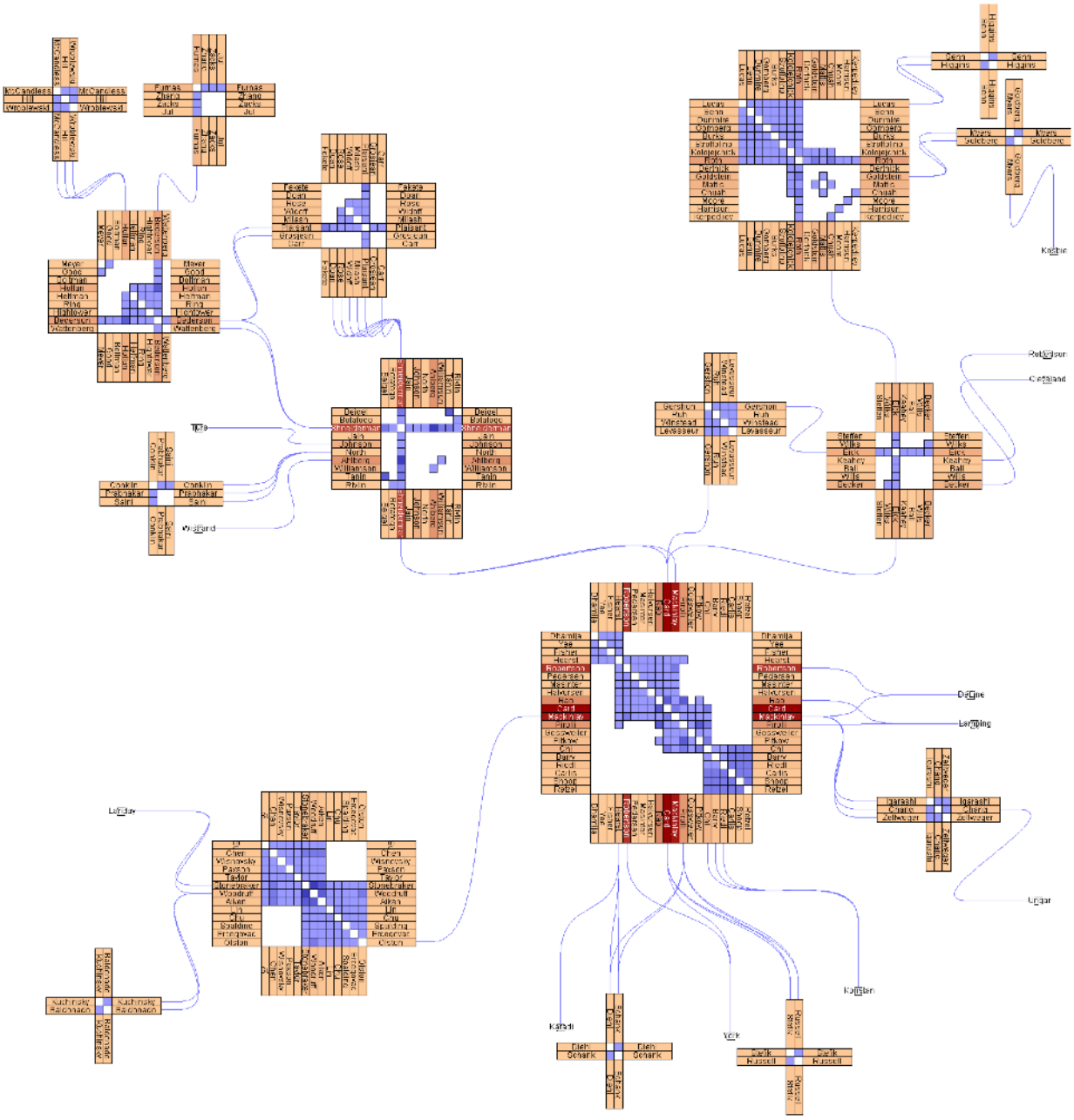


Figure 11. NodeTrix visualization of the information visualization field. This is the largest connected component extracted from the dataset used in the Infovis'04 Contest. Some remaining duplicated authors were manually removed. Colors on axes of matrices represent the number of citations of each author. Color intensity within the matrices represents the strength of each collaboration.

The PivotGraph system is a tool for visualizing and exploring multivariate graphs [101]. The system compresses the network to a small number of meta-nodes and aligns them on a grid to present users with a simplified view of a network. Preliminary tests with experts provide evidence that these visualizations provide a different perspective from traditional graph visualization layouts.

NVSS also addresses the challenge of node and edge layout by using attributes of nodes [81]. User-defined *semantic substrates* act as regions for nodes that share similar attributes. This allows users to examine patterns within and across attributes among nodes. To further reduce cluster, links between types of nodes can also be filtered to achieve a less cluttered network.

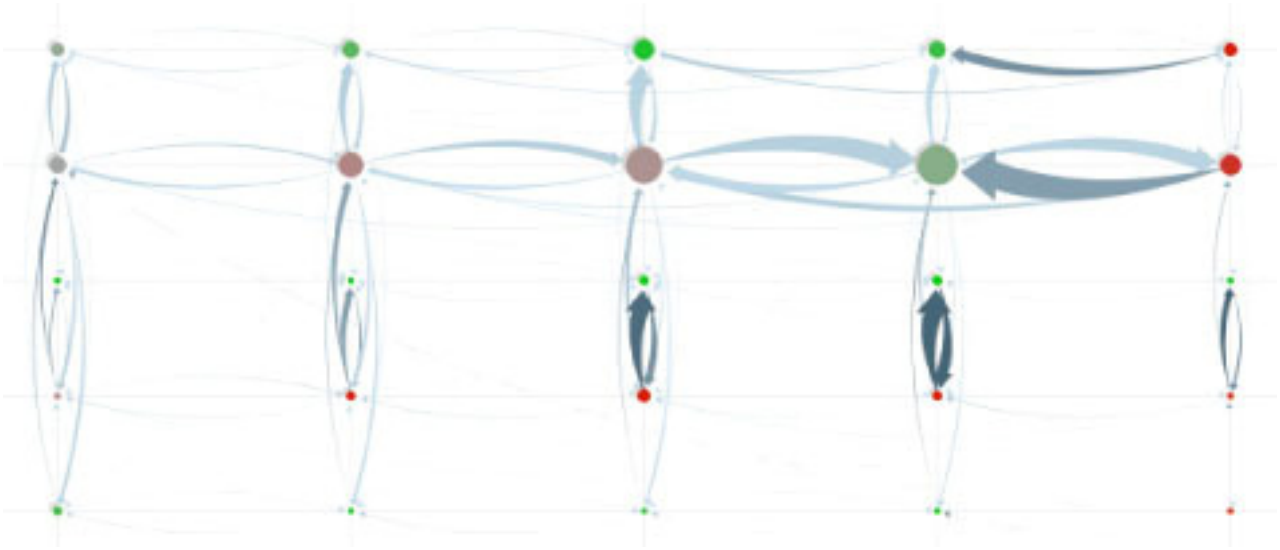


Figure 12. A PivotGraph [101] visualization of a large network rolled up onto two categorical dimensions.

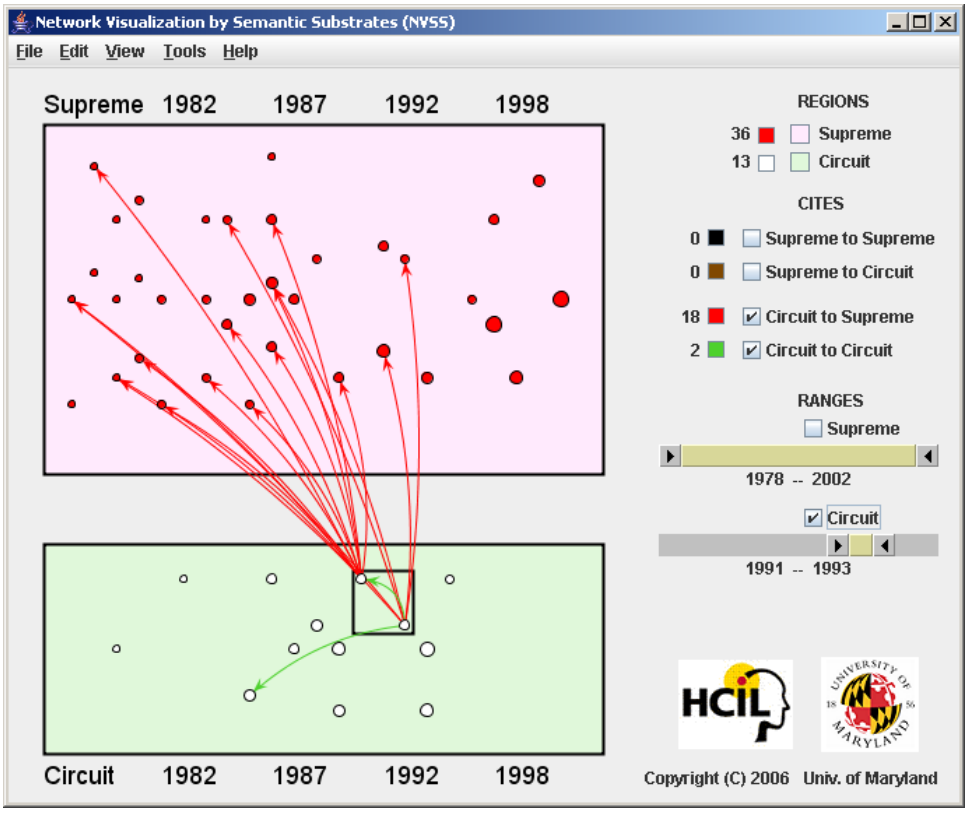


Figure 13. NVSS, a system that visualizes networks by assigning node positions based on a semantic structure.

In the computer science community, many domain-specific visualization and analysis tools have been built. Online social networks [38], criminal networks [105], scientific literature networks [14], language networks [60], and personal communication networks [93] are among a sample of previous work.

There has also been a recent push to make use of network structure for guiding data mining techniques [28]. Social network analysis has also been used to aid knowledge discovery. Xu and Chen use structural analysis techniques to extract criminal network knowledge with CrimeNet Explorer [105]. Chen uses social network analysis to detect emerging trends and transient patterns in scientific literature [14]. More recent work includes Greenland, which augments a node-link diagram with a MDS scatterplot of statistical graph signatures [104].

2.6 Guides for Discovery

When exploring large networks of information, maintaining a path history and providing guides can improve navigation.

The World Wide Web is a one such vast repository of information, which users navigate with hyperlinks and view pages with browsers. Most browsers feature history mechanisms, including a visual cue of changing the hyperlink's color once it has been visited. This technique is effective at alerting users to pages they have already visited, so they need not bother visiting them again [88]. Google's Notebook [31] and Grokker's Working List [33] enable easy recording of web pages that can be saved or sent to others.

However, as a task's complexity increases, more sophisticated guides can alleviate the inevitable struggles of users. "Wizards" are a common type of interface that, instead of informing users how to perform a task, break the task into a linear series of steps. This interface strategy is most successful for tasks that have standard solutions; that is, when a simple step-by-step procedure leads to success [21]. Users often wish to turn off wizards after they have learned a task, and research suggests that users have trouble transferring knowledge gained from wizards to a non-wizard environment [12]. Furthermore, secondary navigation is often preferred to allow users to complete the steps in their own order, and is featured in some commercial software (e.g. Intuit's TurboTax) [12].

Another type of guide is an adaptive interface that reduces the complexity of tasks by "understanding" the user's needs and simplifying the interface [58]. In practice, the algorithms supporting adaptive interfaces are often simple, such as Microsoft Office's Adaptive Menus, which hide the least recently used items. COACH provides pre-coded, in-context guidance, captured from demonstrations that were based on observing user behavior [77]. DocWizards allows users to more easily create *follow me documentation wizards* by learning from demonstrations using a task model [7].

For complex document assembly tasks, some systems will provide an overview of what is needed, so users can see their progress and make informed choices about what their next steps should be. For example, the U.S. National Science Foundation FastLane provides such guidance for the 20+ components that

research teams must submit in grant proposals, with feedback about the last update for each component.

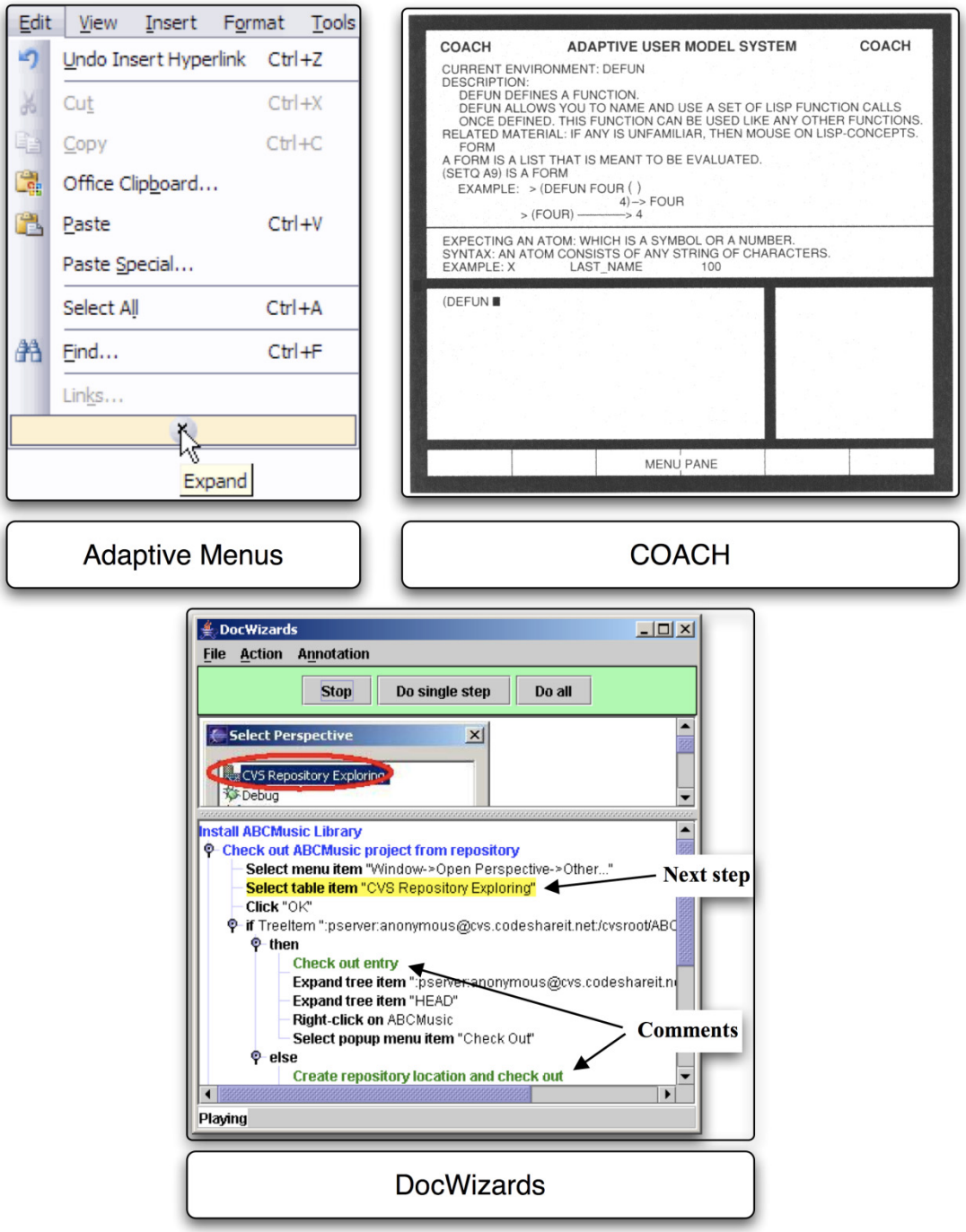


Figure 13. Three adaptive user interfaces. A version of Microsoft Office featured adaptive menus, which hid the least recently used items. COACH provided in-context guidance for programming tasks by observing user's behaviors. DocWizards allows users to create documentation wizards more easily by learning from a task model.

However, there have been few approaches specifically designed for data analysis. Spotfire, a commercial information visualization software package, allows end-users to create guides (Figure 14) [85]. After the process of analysis has been understood, end-users can compose Guides to help automate repetitive procedures and ensure consistency among analysts [86]. Spotfire Guides are presented as a series of hyperlinks that assist users in preparing data, opening standard visualizations, sorting data and even removing outliers. However, the guides do not monitor the actions of users and thus do not provide a measurement of progress. Another approach is by Groth and Streefkerk who describe a prototype system without guides that records the history of user explorations in a visualization tool, as well as the capability for users to annotate their exploration [34].

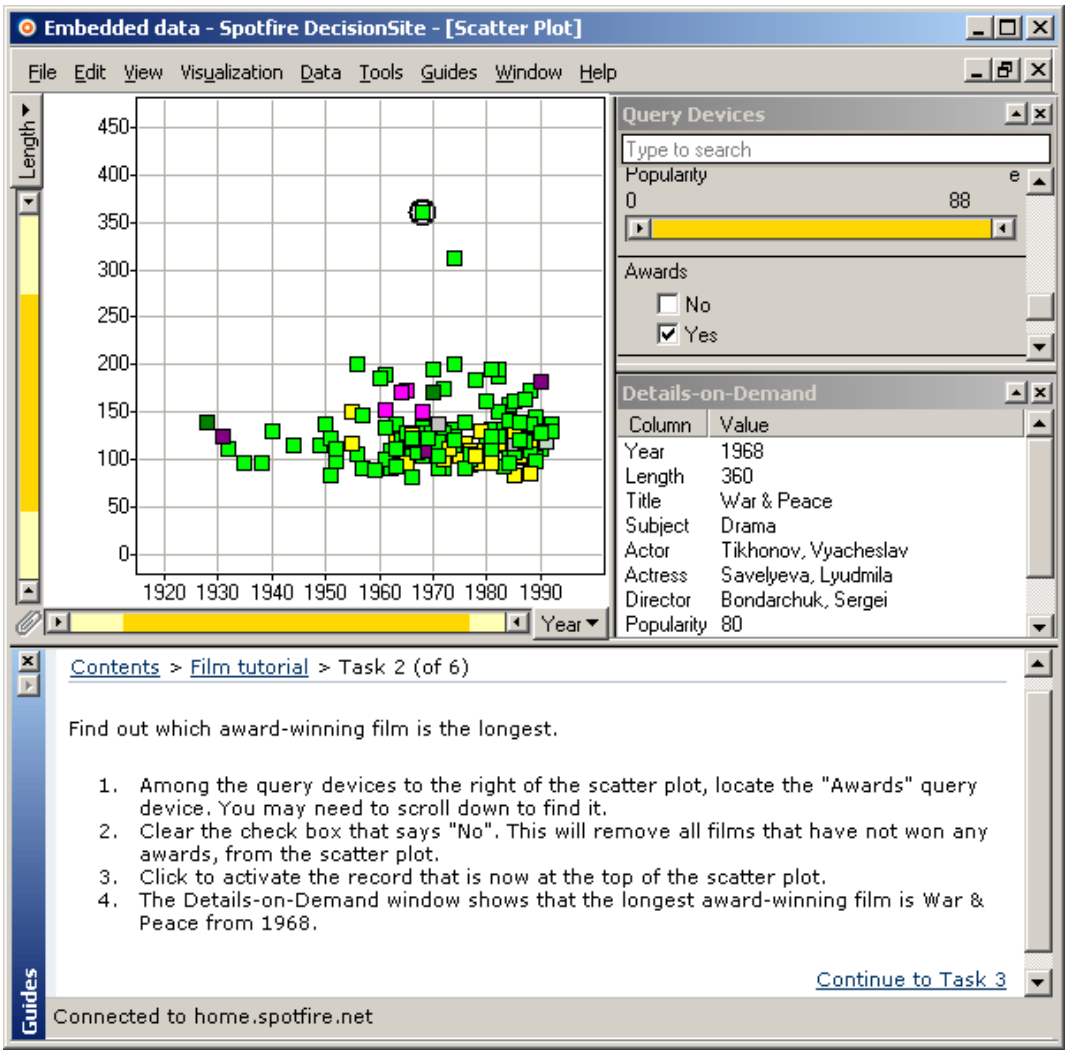


Figure 14. Spotfire, a commercial information visualization software package, allows end-users to create guides for exploring data. In this example, the guide is located on the bottom of the interface. The guide describes the current task, provides instructions on how to manipulate the data, and offers a hyperlink to the next task after users believe they have finished.

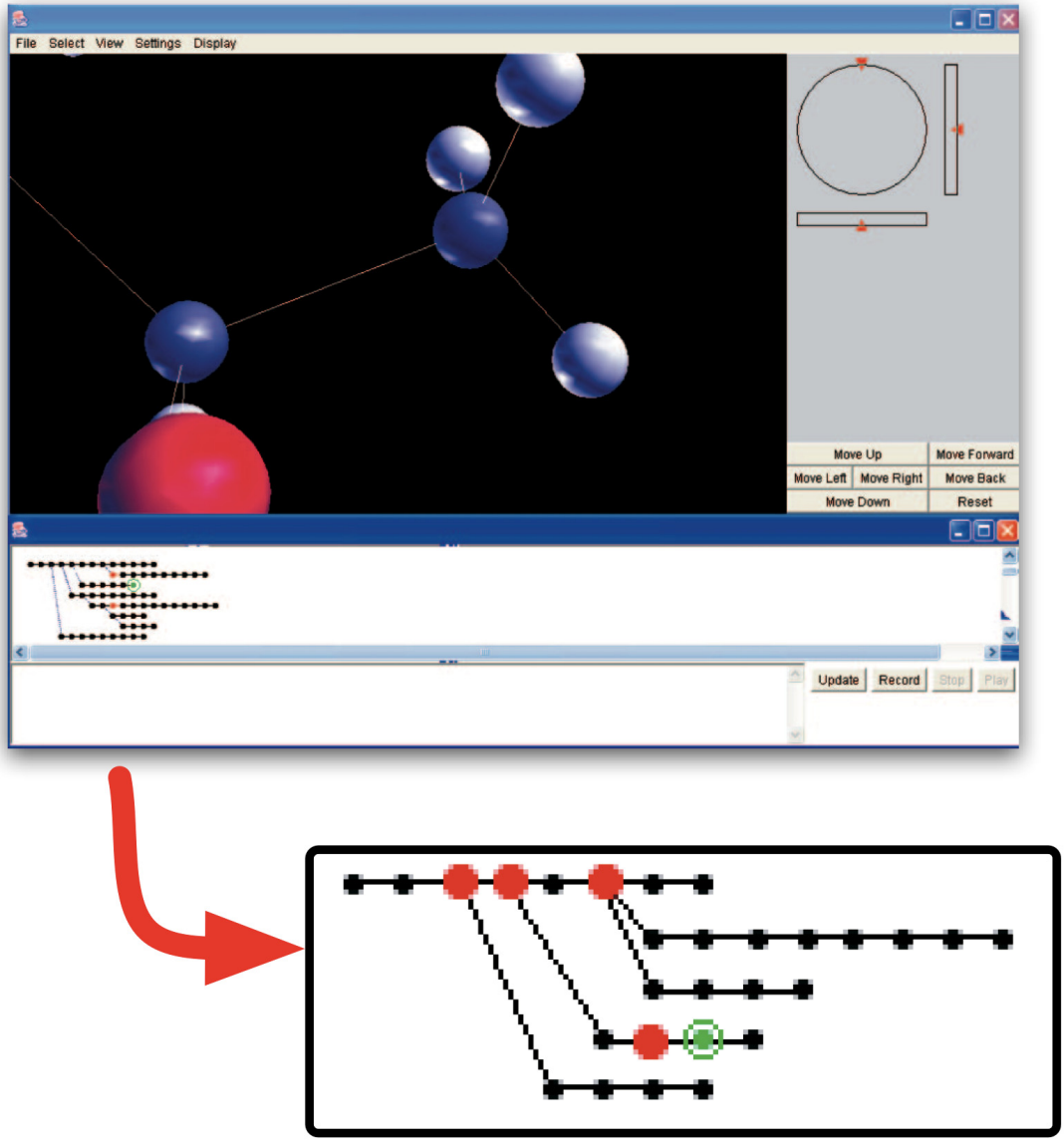


Figure 15. Groth and Streefkerks's Provenance and Annotation system. This prototype system keeps track of the history of a user's exploration through a scientific visualization. The history is visualized as a branching tree, as shown in the enlargement above. Users also have the capability to annotate.

2.7 Evaluation of Information Visualization Systems

There have been many studies evaluating information visualization systems using controlled experiments [15]. Controlled experiments can be effective when trying to decide between multiple designs, such as choosing between widgets [2] or visual mappings [49]. Similarly, controlled experiments can be effective when comparing multiple versions of tools [72].

However, Plaisant has recently initiated a challenge to information visualization researchers to rethink their evaluation strategies and choose approaches that consider the nature of exploratory tasks [70]. In this spirit, Shneiderman and Plaisant propose Multi-dimensional In-depth Long-term Case studies (MILCs) to study the tasks of information visualization system users [84]. Their methodology suggests working closely with expert users and performing in-depth observations to capture users' creative activities during exploration. Of course, there is a long history of qualitative analysis and case studies [107], but I focus on work that takes into account the unique demands of information visualization users.

Saraiya et al. identified characteristics of insight, arguably the primary purpose of visualization tools. By pairing tools with experts and measuring the number of insights reached, they empirically evaluated five visualization tools [75]. However, these evaluations did not capture long-term insights as the evaluation sessions lasted only a few hours. Saraiya et al. followed up this work by performing long-term case studies with experts to address two key characteristics missing from their previous approach: motivation and significance [76]. Their work provides insight into the practices of actual data analysts which have implications for both

design and evaluation of information visualization systems. Seo and Shneiderman also conducted 3 long-term case studies with domain experts, aswell as a survey, which helped show efficacy and suggested improvements to Hierarchical Clustering Explorer [78]. Gonzales and Kobsa used a six-week case study with weekly interviews that illustrated that information visualization tools are most powerful when they are complementary to the workflow of analysts [30].

In order to garner more support for information visualization evaluation, several recent initiatives have taken place The InfoVis Contest also allows long-term analysis but the evaluation is informal [71]. The VAST Challenge improves upon this by making ground truth available [32]. Both of these initiatives provide information visualization designers with standard data sets, which they can use as benchmarks between multiple systems or designs.

2.8 Summary

This chapter has provided a detailed summary of previous work relevant to this dissertation. The chapter begins with the current practice of many social network analysts and the tools they use. This discussion is followed by a variety of advanced network visualization techniques, including layout distortion methods and interaction techniques. Contribution C1:Integration builds on these techniques. Next, a review of guides for discovery are reviewed, which Contribution C2:Guidelines builds on. Finally, a review of information visualization evaluation techniques are described, which Contribution C3:Evaluation builds on.

Chapter 3: Integrating Statistics and Visualization to Improve Exploratory Data Analysis of Social Networks

This chapter focuses on Contribution C1:Integration:

> *Provides an integration of statistics and visualization to improve exploratory social network analysis.*

The integration of statistics and visualization is demonstrated in the implementation of *SocialAction*, a novel social network analysis tool. *SocialAction* uses interactive information visualizations augmented with computed attributes to support the tasks of social network analysts. Lessons from the design of *SocialAction* are described in Chapter 4, where I describe design guidelines for systems to integrate computed attributes. The evaluation of these design goals and *SocialAction* are described in Chapter 5.

3.1 Introduction

My field work with social network analysts, both in academia and industry, suggest that statistical analysis is the most commonly used technique when attempting to interpret social networks. Although visualizations are common in their research publications and reports, they are typically created after the analysis is complete for communicative purposes. However, the most effective visualizations are those that are meticulously hand-crafted.

These exploratory practices might seem surprising, as there is evidence that humans are better at analyzing complex data with images rather than with numbers

[13]. Social network data is extremely complex, as the dimensionality of the data increases with each relationship. However, those familiar with network visualizations might sympathize with these statistically attuned practitioners. Network visualizations are typically a tangled set of nodes and edges, and rarely achieve "NetViz Nirvana" (the ability to see each node and follow its edges to all other nodes). Network visualizations may offer evidence of clusters and outliers, but in general it is hard to gather deeper insights from static visualizations.

My first argument is that it is hard to find patterns and trends using purely statistical methods. My second argument is that network visualizations usually offer little utility beyond a small set of insights. So what should a social network researcher do? Use both – in a tightly integrated way. The design of *SocialAction* centers on this goal.

3.2 Designing for Social Network Analysis: Integrating Statistics with Visualization

Numerous measures have been proposed by structural analysts to statistically assess social networks [99]. However, there is no systematic way to interpret networks, as measures can have different meaning in different networks. This is problematic, as analysts want to be certain they are not overlooking critical facets of the network. In order to make exploration easier, I interviewed social network analysts and reviewed social network journals to tabulate the most commonly used measurements. I then implemented and organized these measures into 6 user-centered tasks: Overview, Rank Nodes, Rank Edges, Plot Nodes, Find Communities, Edge Types. In the sections below, I describe each of these tasks and their associated

features in detail. However, I first begin with an illustration of the main goals of the process.

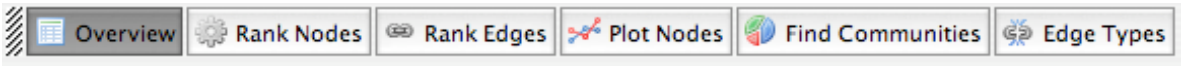


Figure 16. The main toolbar of *SocialAction*. Each of the features are organized into six tasks relevant to social network analysts.

The Visual Information Seeking Mantra ("Overview first, zoom and filter, then details on demand") [80] serves as guidance for organizing the complex tasks of a social network analyst. At the first step, analysts begin with an overview of the network both statistically and visually (Figure 17a). Networks are sometimes to referred to as graphs in other communities. Measurements of the entire network, such as density, diameter and number of components, are computed and presented alongside a force-directed layout of the network. The visualization gives users a sense of the structure, clusters and depth of a network, while the statistics provide a way to both confirm and quantify the visual findings. If the network is small, or the analysts are interested purely in the topology of the network, this step may be enough.

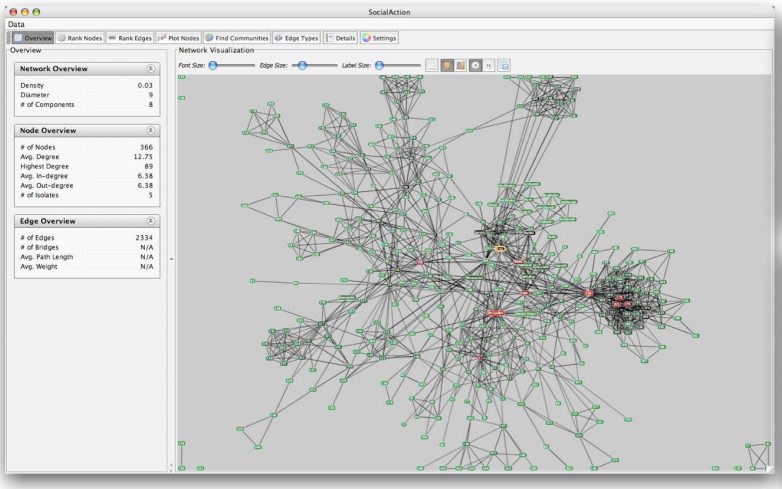
A more capable analyst will wish to gain a deeper understanding of the individual elements of the network. Users can use statistical importance metrics common in social network analysis to measure the nodes (also known as vertices) and edges (also known as links). For instance, analysts can rank the nodes by degree (the most connected nodes), betweenness (the gatekeepers), closeness (well-positioned nodes to receive information) or other metrics. After users select a metric, a table lists the nodes in rank order. *SocialAction* assigns each node a color, ranging from green (low ranking) to black (average ranking) to red (high ranking). This helps illustrate each node's position among all ranked entities. The network visualization is

updated simultaneously, as well, and paints each node with the corresponding color. Users now can scan the entire network to see where the important nodes reside (Figure 1a).

To gain further insights, *SocialAction* allows users to continue on to step 2 of the Visual Information Seeking Mantra ("filter and zoom"), where most other social network analysis packages strand users. Panning and zooming naively is not enough to empower users. Zooming into sections of the network force users to lose the global structure, and dense networks may never untangle. *SocialAction* allows user-controlled statistics to drive the navigation. Users can dismiss portions of the network that do not meet their criteria by using range sliders. Filtering by attributes or importance metrics allows users to focus on the types of nodes they care about – while simultaneously simplifying the visualization (Figure 17b).

After analysts make sense of global trends through statistical measurements and visual presentations, their analyses often are incomplete without understanding what the individual nodes represent. Contrary to most other network visualizations, labels in *SocialAction* are always present. The controls for font size and length allow the analyst to decide their emphasis. In line with step 3 of the Visual Information Seeking Mantra's "Details on Demand", users can select a node to see all of its attributes. Hovering over a node also highlights each node's edges and neighbors, achieving "NetViz Nirvana" for the node of interest (Figure 17c).

Figure 17. Exploring a social network in *SocialAction*. This figure features the "Global Jihad" terrorist network from a case study (366 nodes, 2334 edges). In order to protect sensitive information, node labels have been anonymized except for those individuals publicly identified in the Zacarias Moussaoui trial.

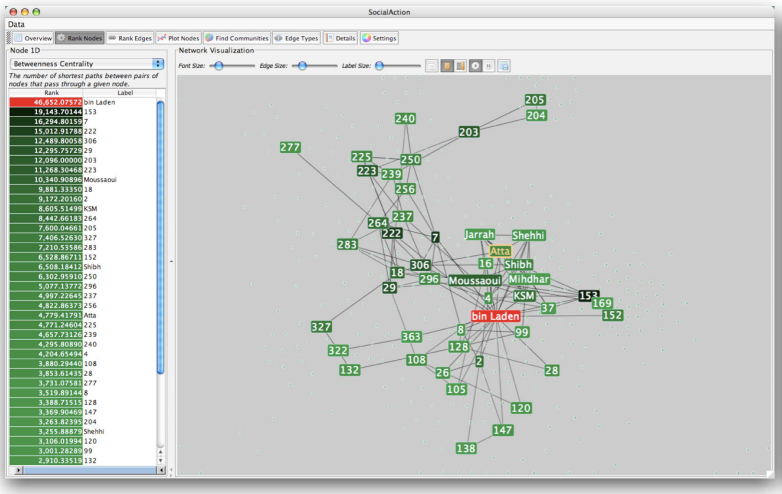


Statistics

Users choose from statistical algorithms to find important nodes, detect clusters and more.

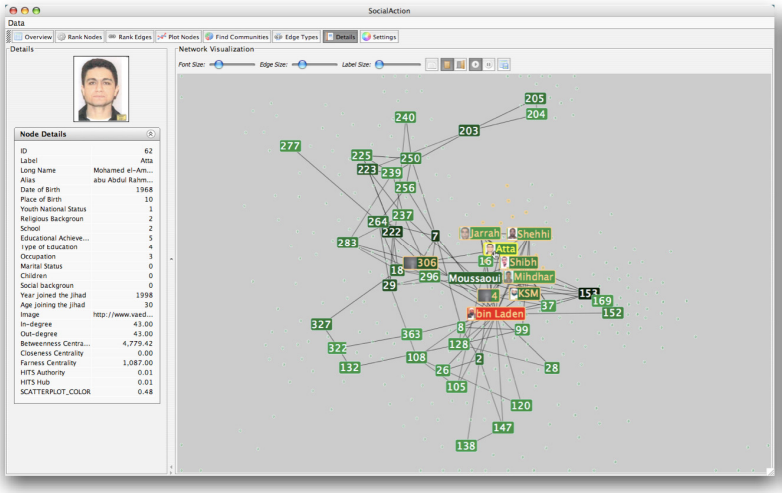
Network Visualization

The visualization is integrated with the statistics. Nodes are colored according to their ranking, with red nodes being the most statistically important.



Gaining Clarity

The gatekeepers are found using a statistical algorithm. Users filter out the unimportant nodes using a dynamic slider which simplifies the visualization while maintaining the node positions and structure of the network.



Understanding the Details

Labels are always given priority so users can understand what the data represents. When user selects a node, neighbors are highlighted and details appear on the left.

In summary, bringing together statistics and visualization is an elegant solution for exploratory data analysis. The visualizations simplify the statistical results, improving the comprehension of patterns and global trends. The statistics, in turn, simplify the comprehension of a sometimes chaotic visualization, allowing users to focus on statistically significant nodes and edges.

3.3 Overview: Gaining Insights from the Network Structure

After users load their data into *SocialAction*, an overview of the network is shown both visually and statistically, as shown in Figure 18.

3.3.1 Visual overview

The network is visually presented with a force-directed layout. This technique is common layout technique for networks [20]. Their goal of the layout is to position nodes so there are few edge crossings and edges are close to equal length. A full explanation of the algorithm and its implementation is in Chapter 5, but I quickly summarize the technique here. Each node and edge is assigned a force. Nodes are assigned forces similar to an electrically charged particle, whereas edges are assigned a force similar to a spring. A simulation is then run on a network, which results in connected nodes to be pulled closer together and disconnected nodes to be pushed further away.

Most social network analysis packages run a force-directed algorithm until a state of equilibrium is reached. However, in complex networks, the running-time can be quite long for a layout to converge, so they often resort to less-than-optimal

layouts. *SocialAction* takes a different approach that allows users to decide when the layout reaches an acceptable state.

For each iteration of the force-directed layout in *SocialAction*, the layout animates into the next optimal state. Users also have the capability to adjust the automatic layout of the network by dragging nodes around. This can be useful during the force-directed layout to speed up convergence, or after the force-directed layout to untangle portions of the network that never reached a state of clarity to users. When users are happy with the layout, users can stop the layout and begin to develop a mental model of the network. These node positions in the network layout will remain stable throughout all of the other tasks and features of *SocialAction*, so analysts will not lose their orientation, unless they opt to redo the layout.

Users are also in control for the number of pixels that can be used for the network layout. By default, the layout forces all nodes and edges to be inside the viewport of of *SocialAction*. Users can zoom out to enlarge the viewport, or zoom-in to narrow it. If users wish for the layout to not be constrained by the viewport, the users can allow nodes and edges to float off-screen. Users can reach these nodes in the future by panning in the visualization.

Network visualizations with force-directed layouts are often criticized by information visualization researchers (e.g. [94]) for usually resulting in a tangled set of nodes and edges. However, they are appropriate to use in *SocialAction* for a variety of reasons. One reason is that social network analysts are often familiar with these layouts. A second reason is that a force-directed layout does not require any specific attributes of nodes or edges, allowing analysts to visualize any type of

network they wish. A third reason is that despite having international symposiums devoted to this topic (the Graph Drawing Conference is in its 16th iteration), the state of the art doesn't seem to be much better than this traditional approach. Rather than devote energy to devising new layout algorithms for networks, I focused my dissertation on developing new interaction techniques. By allowing users to interact with the force-directed layout by deciding its convergence and manually altering the layout, the users' experience with the layout is improved.

The network visualization of *SocialAction* always draws nodes with labels. The ability for social network analysts to quickly identify which data point a node represents has been shown to be quite useful, a feature that is surprisingly absent from many typical network visualizations. *SocialAction* renders nodes with the labeling inside the nodes which is helpful, compared to labels outside of the node. This seems to reduce visual complexity by allowing one eye fixation per node rather than two. It allows improved layouts with fewer edge crossings over or under a secondary text label. At any time, users can increase the font size of the nodes to make the labels more readable with a dynamic slider. Conversely, they can truncate the length of the label to limit the width of a node with another dynamic slider.

By default, edges are rendered as straight lines intersecting the target and source nodes in their respective centers. However, users can also optionally select to draw the line as a cubic Bezier curve which can sometimes produce more aesthetically pleasing visualizations as well as allowing users to more easily follow the paths between nodes. The thickness of these lines can be controlled at any time with a dynamic slider to further emphasize their role in the network.

3.3.2 Statistical Overviews

Three tables of descriptive statistics are calculated automatically when a network is loaded. The first table describes the whole network, measuring its density, diameter, and the number of components. (The implementation of these and other statistical measurements are described in Chapter 5). These statistics can be used by the analyst to ensure the visualization they are seeing accurately describes the network. This is particularly useful when the visualization technique (e.g. force-directed layout) can sometimes be less than optimal. Statistics can confirm the visual evidence, allowing users to be more confident about their interpretations.

In addition to the overall properties, summary statistics on the nodes and edges are also provided. In the nodes table, a count of the number of nodes is presented, along with the maximum degree and average degree, in-degree, and out-degree of all of the nodes. Finally, a count of the isolates is also provided. The edge table provides an edge count and average weight (if such an attribute exists).The number of bridges (removing this edge causes a connected component to become disconnected) and the average path length between nodes can also be displayed. By default, these latter statistics are not calculated until users articulate their interest, as their run-times are considerable.

3.4 Ranking Nodes: Gaining Insights from the Network's Individuals

When users are ready to begin ranking nodes, they click the 'Rank Nodes' button from *SocialAction*'s toolbar. The social network visualization is kept stable on the right, and a corresponding list of nodes is presented on the left. Users can select

nodes in either the ordered list or node-link diagram, and they will be highlighted in both views. In some cases, such as very small networks, this display alone may allow the users to make discoveries. Many networks tend to be larger, such as the example, and this is where computed attribute rankings come into play.

According to my interviews, the methodology of social network analysts suggests to ignore the individual attributes of nodes and instead look at their structural attributes for meaning. Nodes can have meaning derived from their position in the network, as nodes can be isolated or connected to many other nodes. *SocialAction* allows users to rank nodes by their structural position by choosing a ranking of interest from a drop-down menu. Sample choices are:

- *bary center*: the total shortest path of a node to all other connected nodes [63]
- *betweenness centrality*: how often a node appears on the shortest path between all other nodes [10]
- *closeness centrality*: how close a node is compared to all other nodes [22]
- *degree*: the number of nodes a node is connected to. (For directed graphs, rankings based on in- and out-degree also exist)
- *HITs*: a "hubs and authorities" importance measure [52]
- *clustering coefficient*: how close the node and its neighbors are from being a clique [103]

More details about each of these algorithms and their implementation can be found in Chapter 6. When users select a ranking, all of the nodes are ranked according to this criterion in the ordered list.

One of my early partners is working with a dataset consisting of terrorist groups committing over 70,000 terrorist attacks across the world spanning 27 years. This network is being assembled by the Center of Excellence for Behavioral and Social Research on Terrorism and Counter-Terrorism, with the goal of developing strategies for disrupting the formation of terror networks and minimizing the impact of future attacks. The technique is illustrated using this network and others from case studies in Chapter 5, to suggest how the approach applies to real data.

3.4.1 Visual Coding

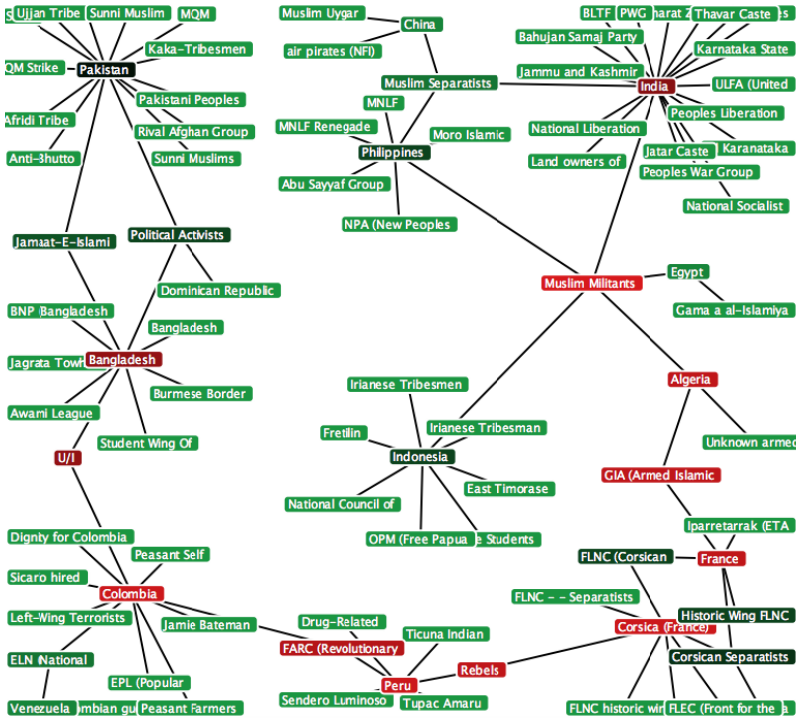
Each ranking is encoded with a corresponding color, ranging from green to black to red, based on its value. This helps illustrate each node's position among all ranked entities. The network visualization also paints each node with this color. Figure 17 illustrates *SocialAction*'s technique on a subgraph from the global terrorism network. This network is two-mode, which mean it has two different types of nodes: terrorist groups and countries. In this network, an edge exists if a terrorist group committed an attack in that country. In this example, betweenness centrality was selected as the ranking criterion. This feature is often used to detect "gatekeepers" between highly connected subgraphs. The nodes are ordered by ranking in Figure 18a. In this example, the "Muslim Militants" group has the highest score. The network visualization, colored according to these rankings, is shown in Figure 18b. Groups with high betweenness appear red in this network.

Rankings

Betweenness Centrality

The number of shortest paths between pairs of nodes that pass through a given node.

Rank	Node	Type
2,516.00	Muslim Militants	Terrorist Group
2,436.50	Corsica (France)	Country
2,413.00	Colombia	Country
2,368.00	Peru	Country
2,280.50	France	Country
2,239.00	Algeria	Country
2,226.00	Rebels	Terrorist Group
2,214.00	GIA (Armed Islamic Group)	Terrorist Group
2,124.00	FARC (Revolutionary Armed For...	Terrorist Group
1,718.00	Bangladesh	Country
1,656.00	U/I	Terrorist Group
1,598.00	India	Country
1,063.00	Pakistan	Country
798.00	Corsican Separatists	Terrorist Group
704.00	FLNC (Corsican National Libera...	Terrorist Group
704.00	Historic Wing FLNC	Terrorist Group
637.00	Indonesia	Country
614.00	Political Activists	Terrorist Group
596.00	Philippines	Country
520.00	Jamaat-E-Islami	Terrorist Group
330.00	Muslim Separatists	Terrorist Group
276.00	ELN (National Liberation Army)	Terrorist Group
187.00	Venezuela	Country
187.00	China	Country
94.00	Egypt	Country
0.00	Dignity for Colombia	Terrorist Group
0.00	Jamie Bateman Canon Front	Terrorist Group
0.00	Sendero Luminoso	Terrorist Group
0.00	Jamiat-ul-Mujahideen	Terrorist Group
0.00	Timorese Students	Terrorist Group



(a) Ordered list of 97 nodes in the largest connected component of the terrorism network in 1996. The nodes are ranked according to their betweenness centrality.

(b) Network visualization of the same 97 nodes, colored according to their ranking. The nodes with highest betweenness rankings, sometimes referred to as "gatekeepers", are painted red.

Figure 18.

3.4.2 Filtering by Rankings

In line with the Visual Information Seeking Mantra [80], *SocialAction* allows users to zoom and filter, since users' perceptual clarity improves when the number of visualized elements is limited [43]. Users can freely zoom into sections of the network to improve clarity by dragging the right-mouse button. *SocialAction* also allows users to filter the nodes in both the ordered list and the network view based on their rankings.

Users can dismiss portions of the network that do not meet their criteria using a double range slider. Users are also able to use the filter to fade the nodes to keep the networks full structure intact (Figure 19). In this example, the betweenness centrality measure was selected and the left bar of the range slider was dragged to the right until it reached the value of 1000. All nodes that do not have a betweenness centrality measure of at least 1000 are faded and are no longer labeled. The nodes that meet the criteria are now prominently displayed with larger labels and allow users to focus on them. If users believe the remaining nodes are still a distraction, users can have them removed entirely.

By allowing users to filter based on rankings that are important to them, the network becomes more manageable in terms of legibility, as the number of nodes and link crossings will be reduced. It also allows users to spot the phenomena of interest across an entire network.

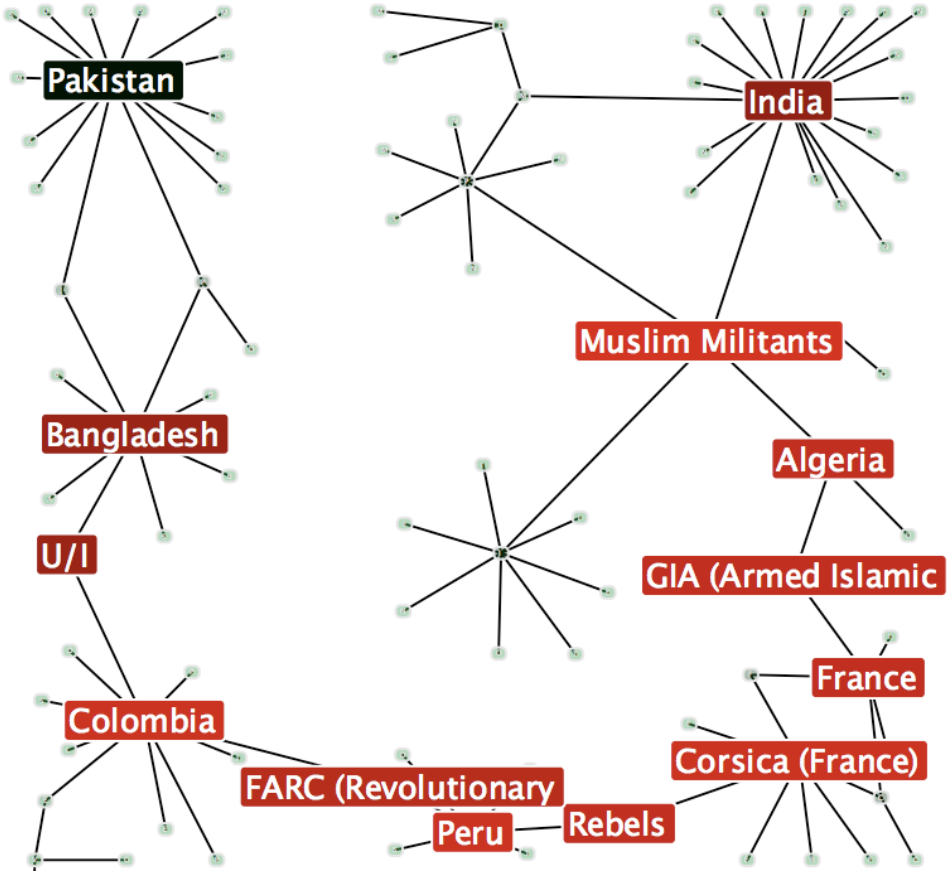


Figure 19. Users can adjust the double range slider to filter nodes that are not of interest. In this graphic, the nodes of Figure 18 that do not have a "betweenness centrality" ranking score of at least 1000 become faded and their labels are removed (all but 13 of 97 nodes). The labels of nodes that meet the ranking criteria can be increased by the user. This allows users to focus on the type of nodes they are interested without ignoring the overall structure.

3.5 Ranking Edges: Gaining Insights from the Network's Relations

Edges can also be ranked to inherent and computed attributes. Similar to nodes, ranked edges are presented in a ordered list and optionally visually encoded with color. In Figure 20, the edges are colored according to their edge weight. Darker edges represent high edge weights, and lighter edges represent smaller edge weights. In this image, color clearly communicates that the left cluster of blue nodes (U.S. Democratic Senators) have much higher edge weights than the right cluster of red nodes (U.S. Republican Senators). This data set is described in more detail later on in a case study in Chapter 4.

Edges can also be filtered. Users have the option of having the edges removed simply from the visual display, or also removing their forces from the force-directed layout. This type of filtering can be useful when users wish to deemphasize weak links in the layout. This has a great effect on the layout, as illustrated in Figure 21.

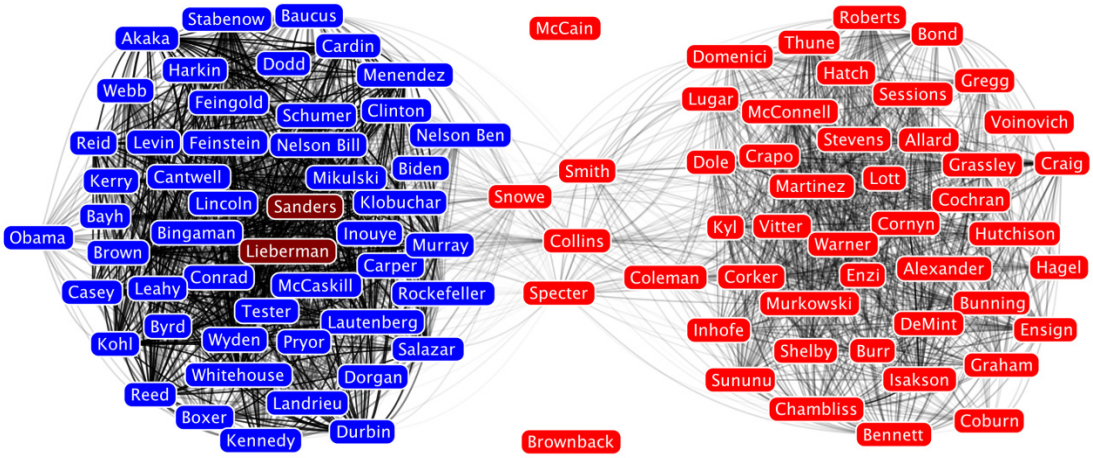


Figure 20. A social network of collobaration among US senators. The color of the edges represent the edge weights, with darker edges implying a higher edge weight. In this social network, a weight of an edge is based upon how often a senator co-voted with another senator. There are 98 nodes and 4753 edges.

Figure 21. The same network as above, but with unfiltered edge rankings. Since every node is connected in this network, the layout results in a strongly connected, tangled sphere of nodes and edges. By filtering the network so that only higher edges were present, the structure reshaped itself as shown in the figure above.

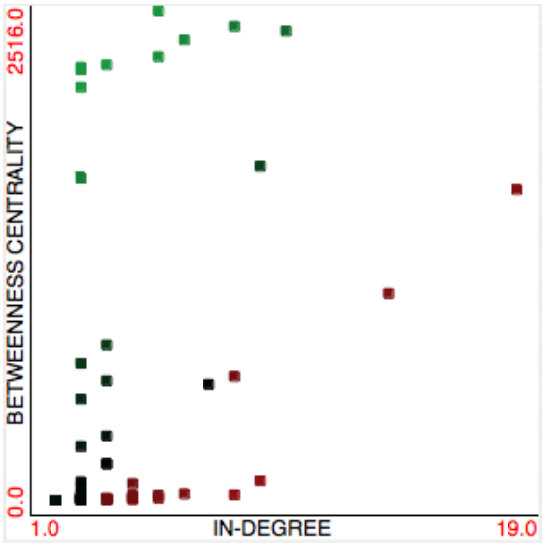
3.6 Plotting Node Rankings: Detecting Patterns of Individuals

Structural analysts may be interested in the nodes that meet criteria across two rankings. *SocialAction* presents this two-dimensional projection as a scatterplot. A scatterplot reveals the form, direction and strength of a relationship between two features, in addition to identifying outliers easily. Users can select two features that form the axes for a scatterplot.

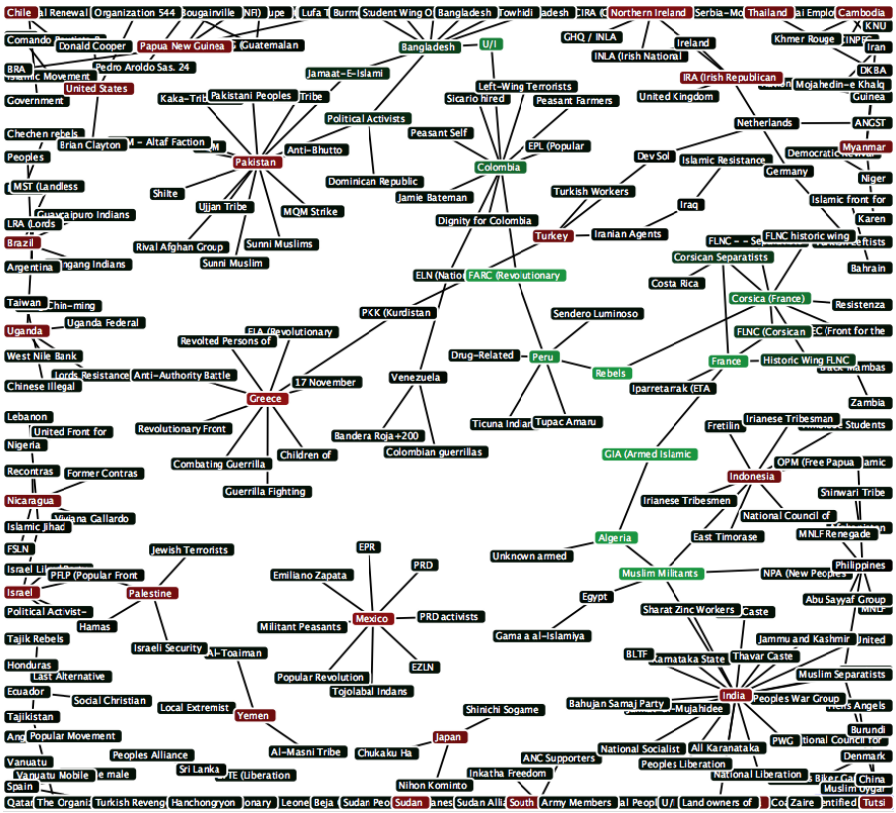
A scatterplot of 276 nodes is shown alongside the network it represents (Figure 22). Users can select any of the ranking features to be the horizontal and vertical axes. In this example, the horizontal axis is in-degree and the vertical axis is betweenness centrality. For the visual coding, *SocialAction* fits the scatterplot to a linear function. All nodes that appear above this linear cross-section are shaded from black to green, and those below are shaded from black to red. The nodes in the network visualization are painted using the same palette. The views are coordinated, so when users select a node in the scatterplot, the corresponding node also becomes highlighted in the network visualization.

Using this scatterplot, users can quickly spot nodes of interest. For instance, suppose an analyst was seeking nodes with low degree (committed attacks in few countries) but high betweennness centrality. The nodes would appear in the upper-left of the scatterplot (Figure 22a). These nodes are also easily detected in the network visualization according to their bright green color (Figure 22b).

Figure 22. *SocialAction* allows users to rank nodes by two different features in a scatterplot. The colors of nodes in the network visualization are determined by the scatterplot position. This allows users to find nodes exhibiting characteristics they seek, as well as outliers. For instance, nodes with low-degree but high betweenness centrality are colored bright green. These nodes can be quickly spotted even in the otherwise unkempt network visualization.



(a) Scatterplot plot of 276 nodes



(b) Network visualization of 276 nodes colored by scatterplot position

3.7 Finding Communities

When networks become large, presenting rankings for every node and edge may not be ideal. Ordered lists get quite long and scatterplots become crowded. Filtering by ranking is one solution to this problem but it is not perfect because it ignores nodes that do not meet a certain criteria. *SocialAction* offers subgroup detection to isolate groups of nodes based on their structural properties. In fact, one of the main goals of sociologists studying social networks is to find cohesive subgroups of nodes [24].

There are a variety of techniques to detect subgroups. For disconnected graphs, a subgroup could be defined as each connected component. However, in practice, components are often large and need to partition themselves into local "communities" of tightly-knit nodes. *SocialAction* includes choices, including a feature to automatically determine communities based on link structure. Newman's community identification algorithm [61] was chosen because it is fast enough to support interactive real-time adjustments.

Like *Vizster* [38], *SocialAction* visually presents the communities by surrounding all members with a translucent convex hull. In this example, the color of the convex hull represents the maximum ranking of any of its entities, so communities containing nodes with high betweenness are red. Users can optionally color the communities by minimum and average ranking values, or opt for each community to be assigned an arbitrary, unique color. By default, communities are labeled with a unique integer but users can rename the labels to have more semantic meaning.

This algorithm was also demonstrated effective in isolating subgroups of personal online social networks in *Vizster* when combined with a slider [38]. Since the algorithm may identify communities at an undesirable granularity, users can move the slider to adjust the state of clustering. This capability is demonstrated in Figure 23. In the upper left image, the entire network is grouped into one group. The next three images demonstrate the network divided into 2, 3 and 4 clusters.

The force-directed layout is also modified with the presence of communities. Additional forces are adding into the algorithm's simulation, which aim to separate the communities into non-overlapping portions of the viewport. This separation is effective at making the edges than span multiple communities more noticeable. The effects of this modified algorithm are also demonstrated in Figure 23, the nodes move into a region with only their fellow community members. Users can also keep the network layout to remain stable during the community analysis, if they've already developed a meaningful mental model.

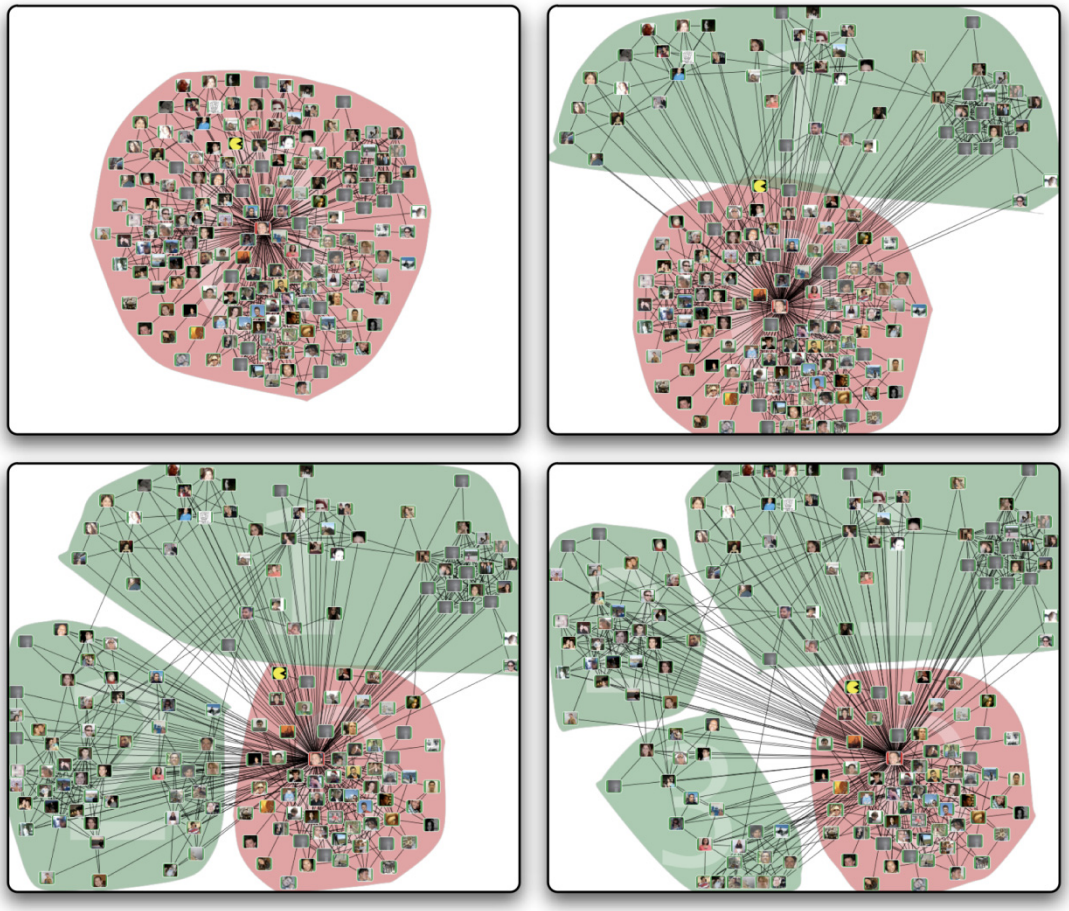


Figure 23. Users have the ability to adjust the granularity of the clusters. By dragging the slider left, users can divide the clusters into smaller groups. In the upper left image, the entire network is grouped into one group. The next three images demonstrate the network divided into 2, 3 and 4 clusters. Users have control over the algorithm, empowering them to choose clusters that make sense for their analysis. In this example, the layout also updates with each new community, as additional forces are created to make a clear separation between clusters. Users can control if they wish for the new layout to be in effect or not.

In Figure 24a, community detection is enabled on a 97-node network. The algorithm finds nine communities. In addition to detecting subgroups, *SocialAction* allows users to use the subgroup information to improve clarity in two ways.

The first technique is presented in Figure 24b. If users are interested in an overview of the structure, users can collapse a subgroup into a single meta-node (linked with meta-edges).

This meta-node, representing the entire subgroup, will be positioned in the center of where the subgroup previously existed. The node's size will be in proportion to the number of nodes it contains. Similarly, the size of the meta-link between nodes will be proportional to the number of links between the groups. The ranking panels (ordered list, scatterplots) treat each subgroup as one entity, and users can search for patterns using the compressed subgroups.

If users are interested in local structure, subgroups can be analyzed in isolation. The system can treat the subgroup as if it is the entire graph, and all ranking panels will be updated accordingly. Further aggregation can be performed on this subgroup, as well. Figure 24c is the result of users choosing the third community in the upper center of Figure 24a, and then further dividing the community into two subgroups.

After users finish exploring subgroups using either of these techniques, the users can return to the original graph, and all nodes that reappear will keep the position they held when they initially disappeared.

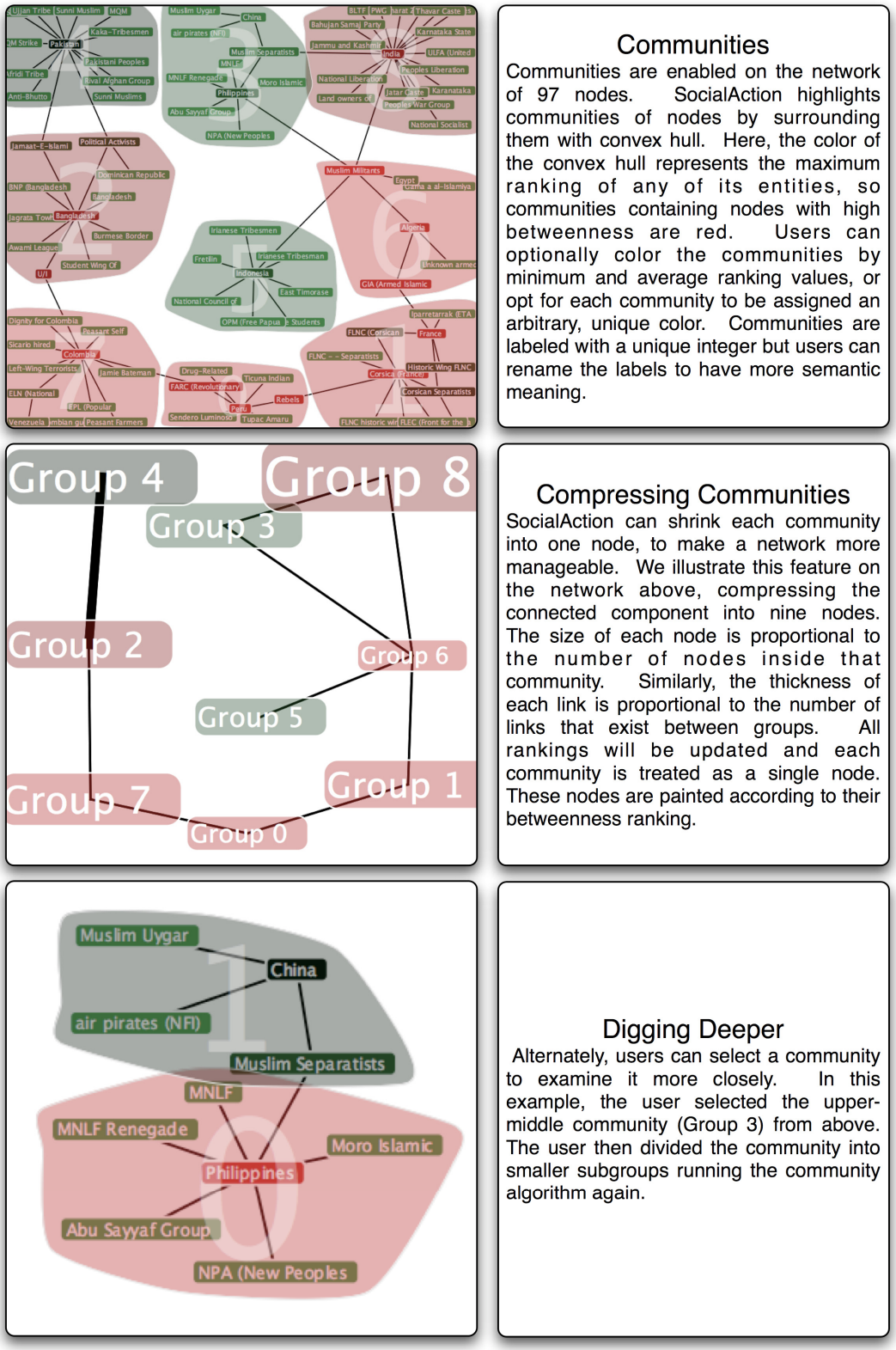


Figure 24. A demonstration of the community features available in SocialAction on a social network of 97 nodes.

3.8 Multiplex Rankings

When social networks have multiple edge types, they are often referred to as multiplex networks. For instance, in a terrorism network, nodes can be connected based on if they committed a terrorist attack in the same area, or used the same weapons, or if they come from the same region. Edges can also have temporal characteristics; a edge could represent an attack in a certain year. The types of edge used depend on what types of questions the analyst is trying to answer.

Often, a network will look drastically different based on which types of edge are shown (Figure 25). The top image shows all edges. The middle shows edges between terrorist groups and countries based on if they attacked in the year 1988. The bottom shows edges if they attacked in 1989. *SocialAction* allows users to quickly iterate between networks of different edge types while keeping the layout stable. In this example, a force-directed layout was used based on the network structure with all edges present. Since the layout was not optimized for the individual years, users have the choice to leave them in this position, or have the layout update with smooth animation to reduce the number of edge crossings.

In these examples, the nodes are colored based on their degree ranking. Nodes that do not have any edges of the selected type are faded. Users can increase the legibility of nodes with edges by increasing their font size with a slider. For instance, 'LTTE' is an active terrorist group in 1988 (attacking India and Spain and painted red), but fades in 1989 due to a lack of activity.

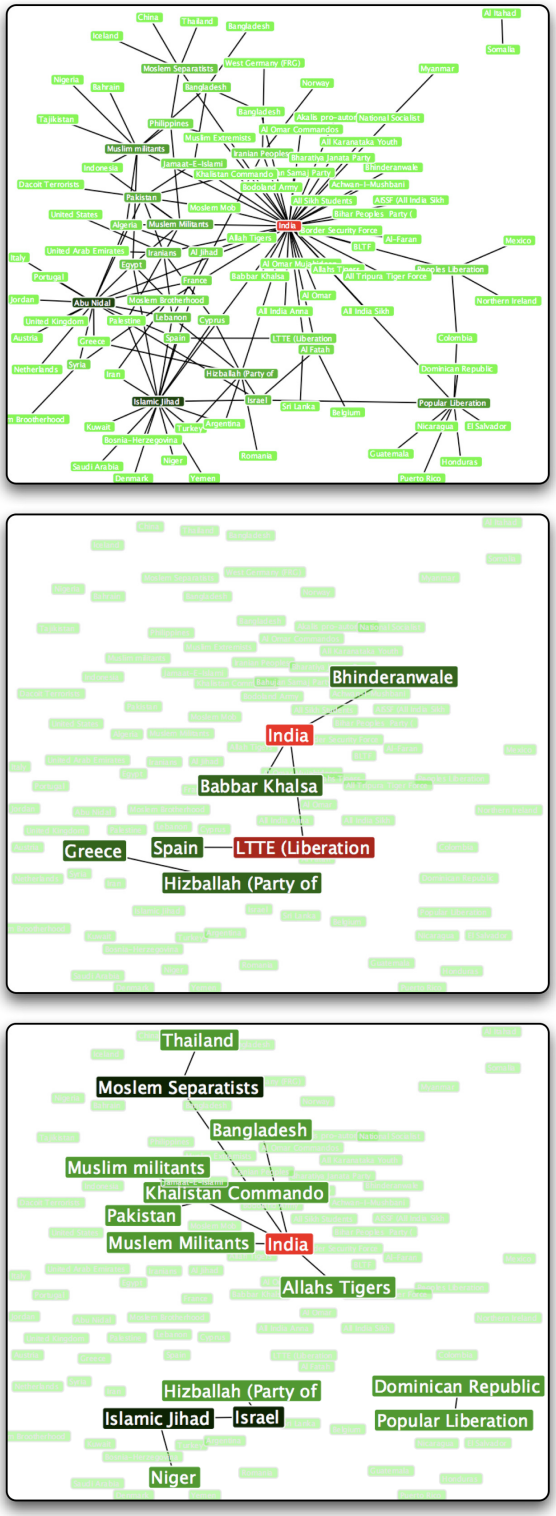


Figure 25.A demonstration of the multiplex features of *SocialAction*.

SocialAction also offers a matrix summary so users can spot patterns across many different edge types at once (Figure 26). Each node occupies a row, and each column represents a different edge type. Each cell is colored based on a node's ranking when only that edge type is present. In Figure 26, degree was the selected ranking criterion and the rows are sorted in descending order by degree when all edge types are present. For this subgraph, India has the highest degree (most terrorist attacks) when all edges are present as well as most years from 1980-1997, as those cells are colored bright red. Countries such as Lebanon, Egypt and Pakistan are dominant in years that India is not. *SocialAction* allows users to flexibly explore multiplex networks. Users can iterate through different edge types separately and apply the ranking and aggregation techniques as well. Users can also spot patterns across edge types using the matrix overview.

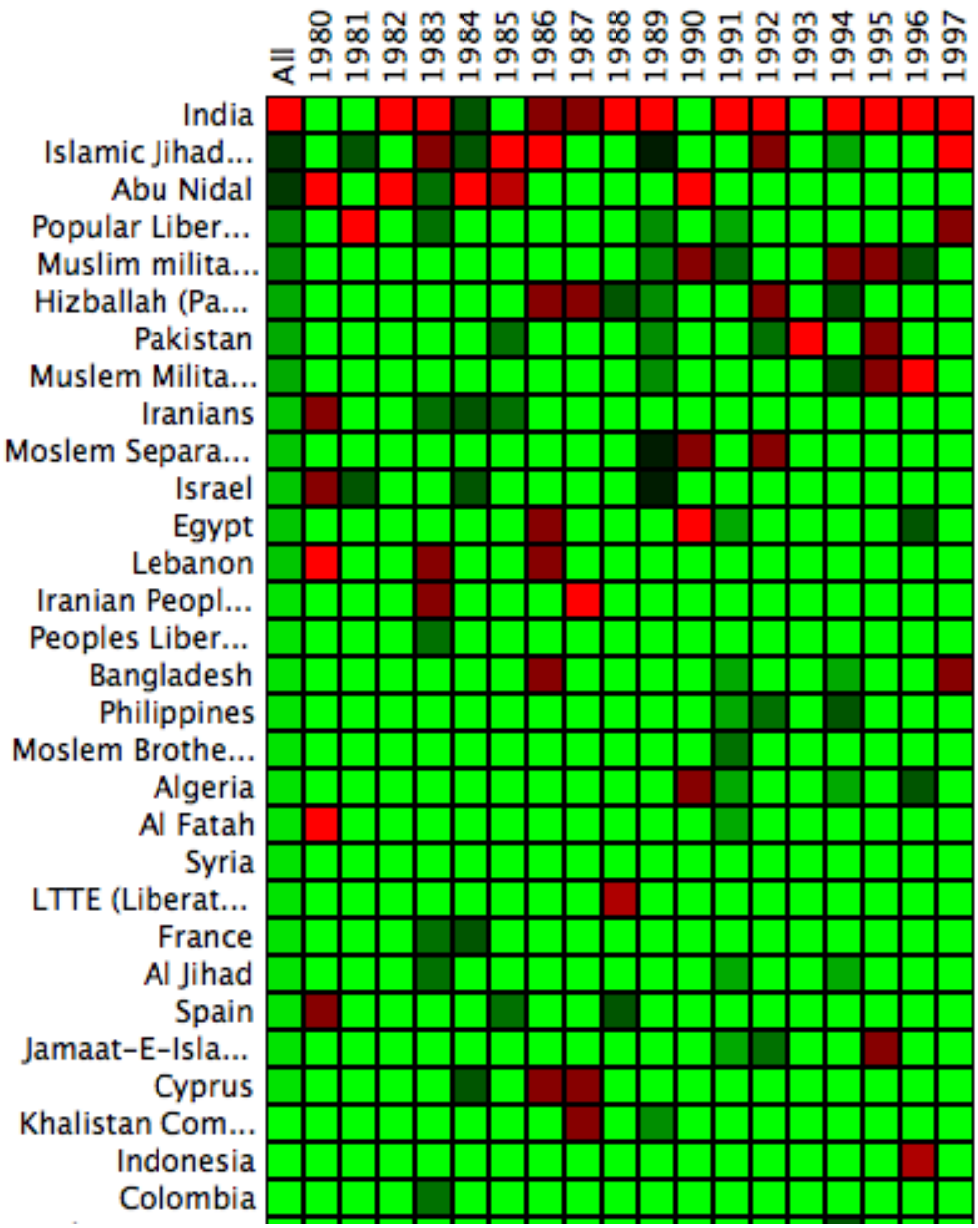


Figure 26. The overview matrix's rows are the top 30 nodes with the highest degree (when all ties are present), sorted in descending order. Each column represents a different type of edge present. For each cell, a greater intensity of red implies relative higher degree, whereas greater intensity of green implies lower degree.

The matrix overview was limited and required scrolling, as not every node could fit on the display. This was further improved to the form of a stacked histogram, similar to ThemeRiver [36]. Each node is represented as a line, and each column represents an edge type. The node's thickness in each column represents the node's ranking in the network of that edge type. The color is based on their overall ranking over all edge types. In Figure 27, two stacked histograms are shown, which demonstrate the evolution of the terrorist bipartite network over time. The country nodes are alphabetized and stacked in the top visualization, whereas all the terrorist groups are in the bottom visualization. The thickness of the node at each year is based on the node's degree in the network. Nodes are colored based on their degree (red implies high degree, green implies low degree). Nodes receive labels in their peak year, if their degree is at least 6. There is a clear peak of attacks in 1992. Various trends can be interpreted from this image, such as Italy has many different groups attacking in the earlier years, whereas India has peak activity in the later years. Since there are many more terrorist groups than countries, the bottom image is harder to interpret. However, these visualizations are interactive and users can filter the visualization according to name. So if an analyst typed the word "Armenia", only the nodes with terrorist groups with the word Armenia (such as the Armenian Secret Army for the Liberation of Armenia, and Justice Commandos for the Armenian Genocide) would be shown. This type of interaction is similar to searching for names in NameVoyager [100].

The visualizations in Figure 27 were featured in the 2007 Competition on Visualizing Network Dynamics (http://vw.indiana.edu/07netsci/). One of the reviewers sent a particular rewarding quote that emphasizes some of the goals of *SocialAction*:

"Networks are best read if they are not only 'technically accurate' and visually attractive but when they employ a type of rendering that creates a landscape. That creates a bridge for the uninitiated audience to cross into the field of expertise. Dataland travels have now become so enjoyable; they may soon appear as special fare destinations at a travel agency near you. Perer's visuals make that trip into the land of terror networks absurdly attractive. Having intellectual entertainment and visual pleasure with terrorism analysis is perhaps one way to diffuse the very essence of terror - by analyzing it without being terrified. And in the end it leads to a hopefully more rational dealing with it, which is the opposite of what terrorism is trying to instill."

Ingo Günther (Journalism & Art)

Tokyo National University for Fine Arts & Music, Japan

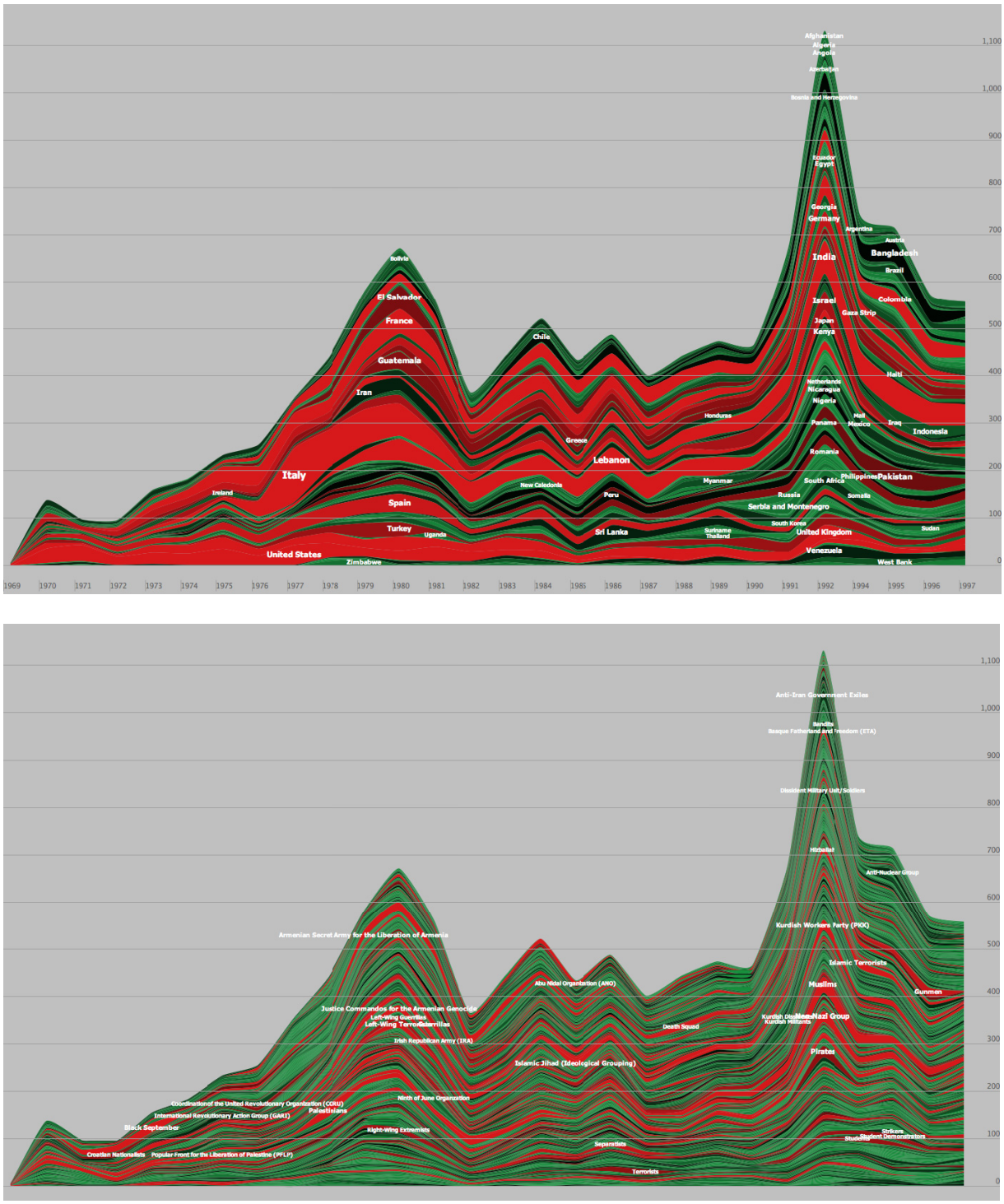


Figure 27. A multiplex overview of a bipartite graph. The top image shows the nodes representing countries, while the bottom image shows nodes representing terrorist groups.

3.9 Supporting Effective Exploratory Data Analysis

So far, I have described techniques of importance to the SNA community: ranking nodes and edges, plotting nodes, finding cohesive subgroups and exploring multiple edge types. In addition to providing these features, I also designed the interface to support orderly exploratory data analysis. Users can iterate through the network measures available to examine the range of structural properties. The spatial layout of the node-link diagram remains unchanged during this process to preserve users' mental model of the network. If the network is too large to effectively deal with the entire network, users can iterate through each subgroup and apply the network measures to these smaller groups in order. Finally, if a network is multiplex, users can iterate through each edge type while being provided a matrix overview. Users have the freedom to apply specific techniques to support their hypotheses. However, if they are interested in exploratory data analysis and want to examine the full range of measures, the interface provides an orderly, systematic method for doing so.

3.10 Summary

This chapter demonstrates how *SocialAction* provides computed attributes by integrating statistics and visualization to improve exploratory social network analysis. The design organizes the features for social network analysts into six distinct tasks (Overview, Rank Nodes, Rank Edges, Plot Nodes, Find Communities, and Multiplex). For each task, relevant statistical and visual information are presented.

The Overview task provides statistics that measure the overall topology of the network alongside a force-directed network visualization. The Ranking tasks allow users to find important nodes and edges by using statistical algorithms to rank, color, and filter interactively in statistical and visualization views. The Plotting task allows users to compare and filter multiple rankings using network visualizations and a more comprehensible scatterplot to locate patterns and outliers. The Communities task allows users to find cohesive subgroups of nodes, presenting the results of the clustering algorithms in the visualization. The multiplex task allows users to identify different edge types, statistical and visually measure them in isolation, or see a visual overview of them all. The organization of these statistical and visual features into six tasks supports effective exploratory data analysis.

Chapter 4: Design Guidelines for Information Visualizations with Computed Attributes

This chapter focuses on Contribution C2:Guidelines:

> *Provides design guidelines for interactive techniques to improve exploratory data analysis with computed attributes and systematic yet flexible guides.*

Humans can be quite good at scanning data, recognizing patterns, and remembering images. However, as data grows larger and more complex, it is clear that interaction is necessary to present data interpretable by humans. Yi et al. provided a taxonomy of seven types of interaction techniques of existing approaches of interaction in information visualization systems [106]. Yi et al.'s categories (Select, Explore, Reconfigure, Encode, Abstract/Elaborate, Filter, and Connect) suggest ways that users can navigate through complex information landscapes. However, each of these techniques described usually rely on inherent attributes of the data.

For instance, in a social network of male and female students, users could select students of interest, explore by performing a direct-walk through the network, or filter based on gender. However, the inherent attribute-based interactions may not support the needs of certain users. If the tasks are to find the gatekeepers, communities, and most popular students, an algorithmic approach may be faster and more precise. Thus, it seems to make sense that the visualizations should be augmented with these computed attributes if they are relevant to users' tasks.

Using the taxonomy of Yi et al., I describe each of the seven interaction categories and demonstrate how they can be augmented with computed attributes from statistics and data mining. As data becomes larger and more complex,

leveraging the benefits of statistical analysis seems both rational and necessary to answer many analytic tasks. For example, Amar, Eagen and Stasko categorized all information visualization analytic tasks into 10 components (Retrieve Value, Filter, Compute Derived Value, Find Extremum, Sort, Determine Range, Characterize Distribution, Find Anomalies, Cluster, Correlate) [3]. For each of these low-level visual analytic tasks, the use of statistics and data mining algorithms seems obvious to help show information that makes it easier to discern from visualizations.

I present seven interaction techniques that can assist users when augmented with computed attributes. In each of these interactions, users will still be in complete control not having to rely interpreting a black box of automatic algorithms.
Of course, computed attributes from statistics and data mining requires information visualization tools to feature a more sophisticated design. Users much navigate both the visualization and the statistical algorithms. In order to aid such a design, I present guidelines for navigating the statistical algorithms as well.

However, many different types of interaction can lead to complex paths of analysis. In order to assist discovery of insights, I also provide design goals for *systematic yet flexible* (SYF) discovery. The goal of these design goals is to help guide domain experts through analysis over days, weeks and months. SYF offers *systematic* guides that provide users the ability explore relevant analytical features. However, SYF also supports *flexible* diversions to pursue insights while still maintaining overall progress. To assist analysis, SYF provides annotation, collaboration and reuse capabilities. This results in seven additional design goals for assisting analysts throughout the analytical process.

4.1 Exploratory Data Analysis & Computed Attributes

The visual information seeking mantra of "Overview first, then details-on-demand" has been defined by Shneiderman in [80] to support interactive exploration of information visualizations. Similarly, successful statistical analysis techniques have been defined in Tukey's mantra of "Exploratory Data Analysis" [89]. Tukey's techniques and inspired techniques allow analysts to answer questions like, "What is a typical value?", "What is the distribution?", "What is the percentile?", and "What are the outliers?".

Given these two widely cited techniques, one might question the need for additional design goals. However, there is a growing need to argue that *both* interactive techniques and statistical techniques are necessary for successful information visualization systems. There are few information visualization tools that integrate this dual-front approach to solving data analysis problems. However, my research suggests that long-term cases studies with information visualization analysts reveal integrating visualization and statistics lead to insights [66]. In fact, novel insights were reached in all case studies. However, the lack of specific design guidelines perhaps limits its reach of direct usefulness to future researchers, designers, and practitioners.

I note that despite this approach supporting algorithms from statistics, mathematics, and data mining, this approach is different than Keim et al.'s proposed Visual Analytics Mantra which emphasizes automatic detection of salient data to reduce complexity before and after human analysis [51]. I believe that users should be present in each step of the analytical process. Instead of automatically preparing

and summarizing data, information visualization systems designers should design their interfaces to allow users to choose the most relevant algorithms, get feedback about the effect of the algorithms, and then filter the data according to pertinent statistical recommendations of the algorithms.

This is similar in spirit to the GRID principles proposed by Seo and Shneiderman [79], which are summarized as:

(1) Study 1D projections, study 2D projections, and then find features.

(2) Ranking guides insight, statistics confirm.

However, the design principles described here go beyond ranking and support any types of computed attributes from statistical techniques or data mining knowledge. They also focus on specific interactions and emphasize how to show the data relevant.

Here, each category of information visualization interactions are augmented and improved with computed attributes. For each interaction technique, I also provide an example that demonstrates how *SocialAction*'s integrated computed attributes can improve user's ability to find patterns, gaps, and outliers in the visualization.

4.1.1 Reconfigure: Augmenting Visualizations with Computed Attribute Views

The *reconfigure* interaction is common in information visualizations to provide users with a different perspective on the data. For instance, Spotfire provides 8 different perspectives to visualize tabular data, such as scatterplots, bar charts and heat maps [85], allowing users to easily switch between representations that best suits their task at hand. In similar spirit, TableLens allows users to sort and rearrange columns of tabular data to highlight different patterns in the data [73]. Most of these techniques

focus on reconfiguration based on inherent attributes. However, this dissertation demonstrates a valuable design goal of supporting *reconfiguration* based on computed attributes from statistical techniques and data mining algorithms, particularly when the visualization is complex.

The complexity of network visualizations is an example when a reconfiguration can be useful. While others have tried reconfiguring network visualizations into trees [55] or matrices [41], a reconfiguration based on computed attributes can provide an even simpler view on the data while also highlighting statistically interesting properties. *SocialAction* reduces the network visualization into tables and scatterplots. A table view, ranked by a computed attribute, allows users to focus attention on important nodes. A scatterplot view, presenting 2 computed attributes, allow users to compare multiple importance rankings in a visualization that has been shown effective at highlighting patterns, gaps and outliers. The reconfigurations in Figure 28 illustrate how the reconfigure interaction technique can be made even more powerful by computed attributes.

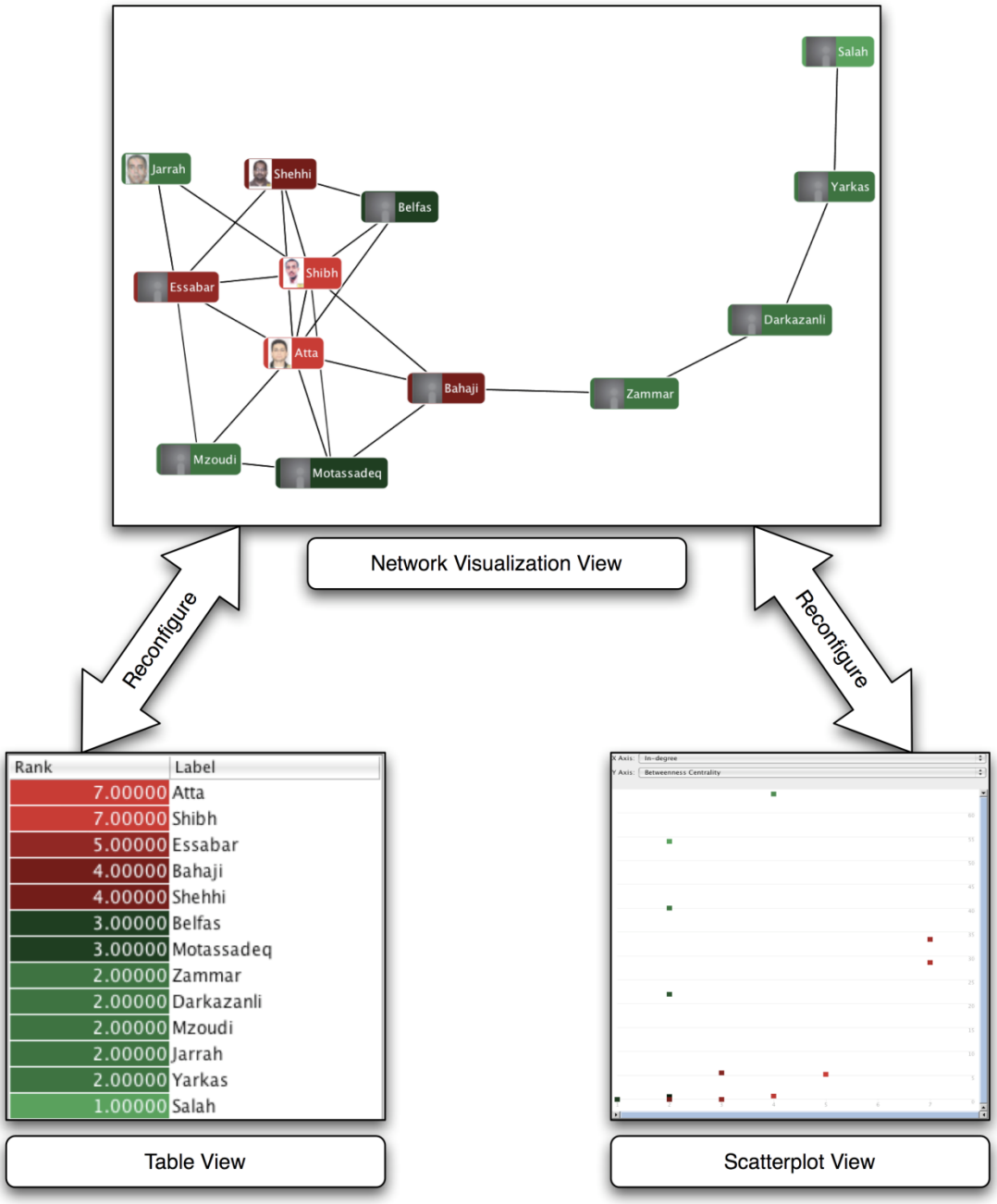


Figure 28. The *reconfigure* interaction technique with computed attributes, as implemented in *SocialAction*.

4.1.2 Connect: Coordinating Statistical and Visualization

The *connect* interaction is in synergy with the *reconfigure* technique. This technique aims to highlight associations and relationships between data items. As the *reconfigure* technique suggests, it can be advantageous to users to see different displays of the same data. *Connect* suggests to show these multiple displays at the same time. A common interaction technique in this category is brushing, which allows users to select a data item in one view and see the item in multiple views. Brushing is most often used in coordination with different projections of inherent attributes of the data. For instance, if a Spotfire user selects a data point in a scatterplot, the corresponding data point will also highlight in an associated bar chart.

Providing reconfigured views of computed attributes a good first step, but connecting these views is essential. Users should be able to browse the data visualization and the computed attribute views in a coordinated manner. At time, the inherent attributes may reveal discoveries, and at other times, the computed attributes will help reveal clues during analysis. *SocialAction* connects these two views together using side-by-side displays that are coordinated. If a data-point is in one view, the same point is represented in another view. Users can brush from one view to the other. If users wish to find nodes with certain structural properties, they can choose an algorithm that detects it in a sorted table instead of being forced to visually scan a complex visualization. If the users care about multiple structural properties, a scatterplot can saliently show the intersection between them. However, computed attributes may not always measure what users are seeking. By connecting both views, users can judge the utility of the algorithms and reflect on its impact to users

tasks. Thus, the connect interaction with computed attributes allows users to both learn about the data and the quality of the algorithms. *SocialAction* supports *connect* by providing side-by-side views, brushing operations, and coordinated visual encoding (described in the next section).

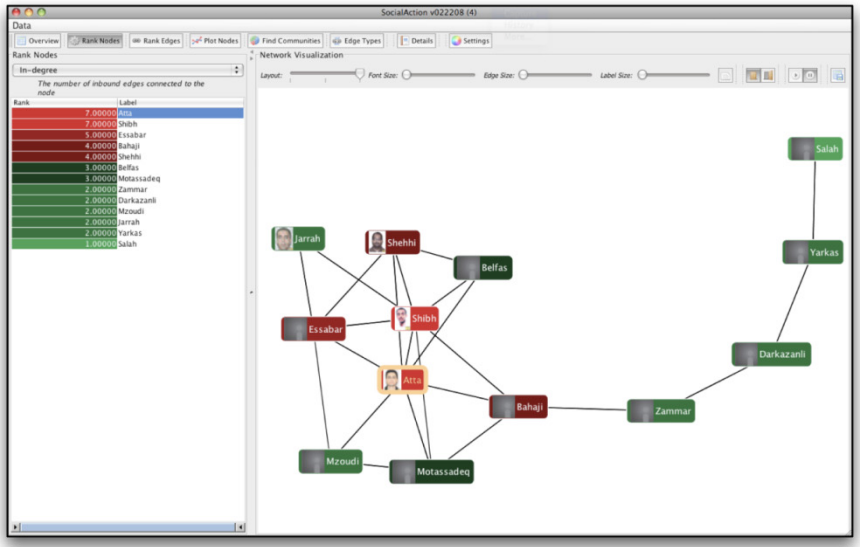
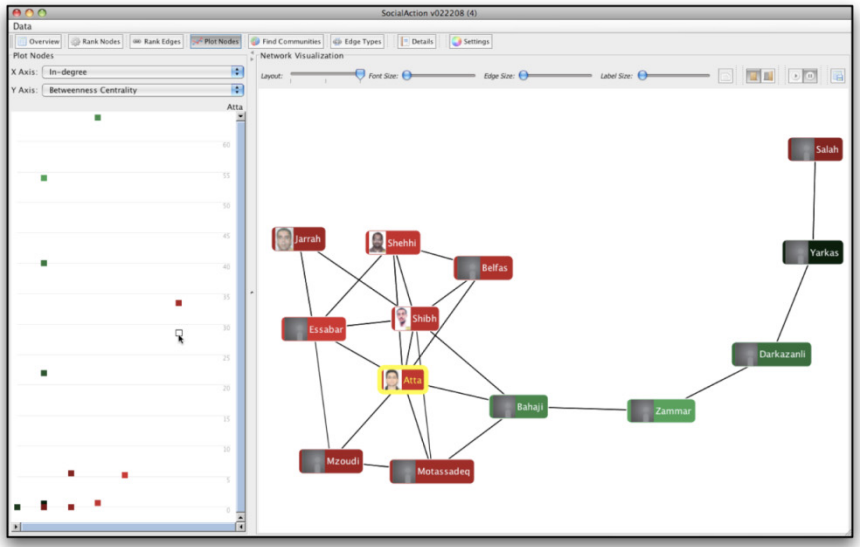


Table View *connects* with Network VIsualization



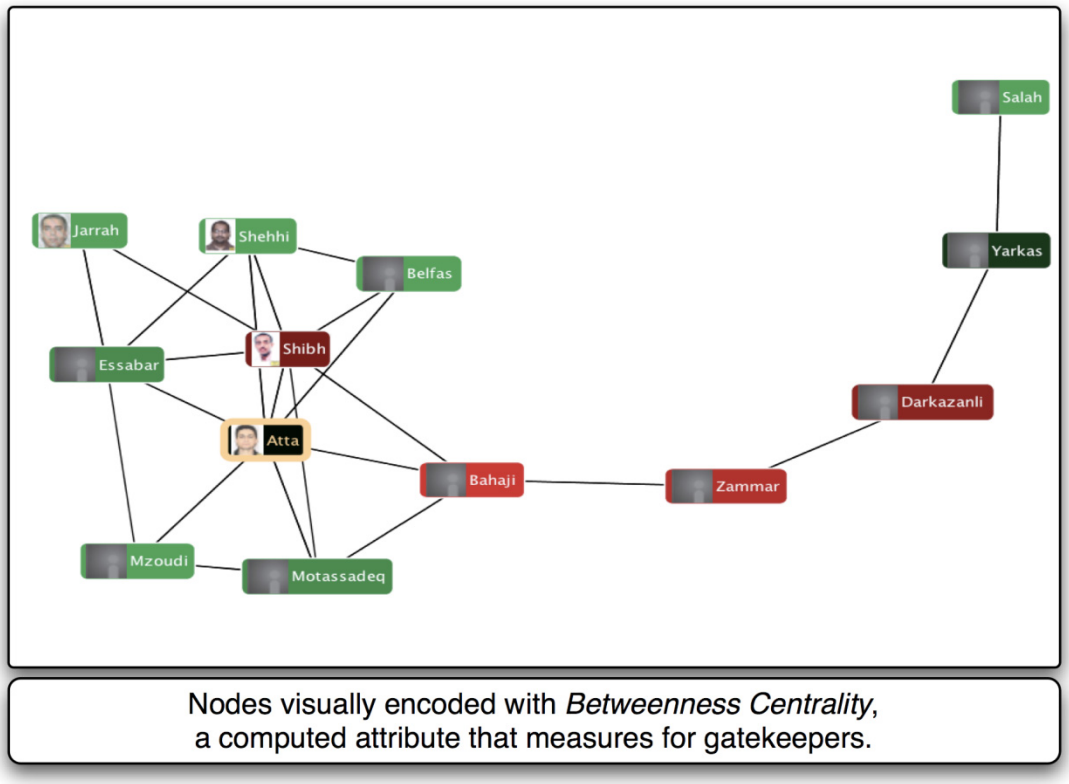
Scatterplot View *connects* with Network VIsualization

Figure 29. The *connect* interaction technique with computed attributes, as implemented in *SocialAction*.

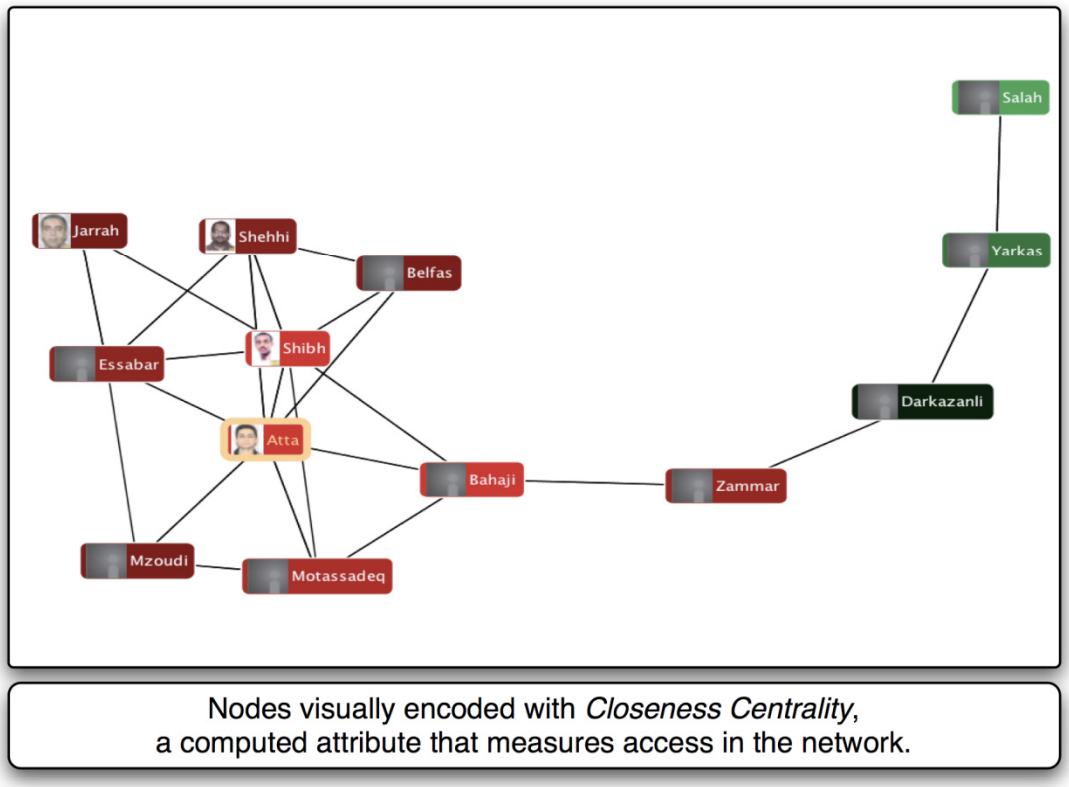
4.1.3 Encode: Representing Computed Attributes.

A popular use of *encode* is to allow users to use color, size, fonts, shapes, and orientation to visual code data point with attributes of interest. This allows visualizations to keep their spatial arrangement constant while visually presenting additional data about each of the nodes. This is a widely used technique in many information visualization systems; however, most encoding focuses on inherent attributes. Visually encoding with computed attributes is a natural extension and a convenient way to augment visualizations with results from statistics or data mining algorithms.

SocialAction follows this design goal by using color to encode results from algorithms. By encoding nodes and edges with computed attributes, entities with certain statistical features can be easily found (e.g. the most popular individuals or the gatekeepers). By default, color is defined along a red-black-green spectrum. Values with the highest ranking are colored red, those in the middle are colored black, and those with the lowest ranking are colored green. *SocialAction* assigns these values along this gradient based on the value of a computed attribute. When users select a different computed attribute, the colors update appropriately. Non-colorblind human eyes can easily distinguish between red and green, suggesting it as an effective color spectrum. These computed attribute encodings can provide clues about the topology even when the topology is too cluttered and dense to make sense of, as demonstrated in Figure 30.



Nodes visually encoded with *Betweenness Centrality*,
a computed attribute that measures for gatekeepers.



Nodes visually encoded with *Closeness Centrality*,
a computed attribute that measures access in the network.

Figure 30. The *encode* interaction technique for computed attributes, as implemented in *SocialAction*.

4.1.4 Select: Marking interesting computed attributes

The *select* interaction technique allows users the ability to mark a data item as interesting. As already discussed, the *connect* interaction technique allows users to select items in each of the views available to users. However, after users find and mark a data point as interesting, the selection should be persistent. This is a particularly important design goal when many different computed attributes are available to users. During exploratory data analysis, users may not know which computed attributes will lead to insights. The ability to efficiently switch between various computed attributes should be supported. But ultimately, users may care about the effect of each available algorithm on particular data points. When both computed attribute values and layouts can change, this selection technique can be important for analysts.

This design goal is demonstrated in *SocialAction* where uses have the ability to select a node at anytime. They can choose a node in either the network visualization or the statistical views. No matter if they change the layout of the network, or compute a new statistical measure, both views will keep track of the previously selected node. Keeping selection information persistent is important for allow users to be more adventurous when trying out additional statistical algorithms. Users shouldn't have their exploration feel constrained by the system, but instead give them the freedom to creatively select and analyze particular data points of interest.

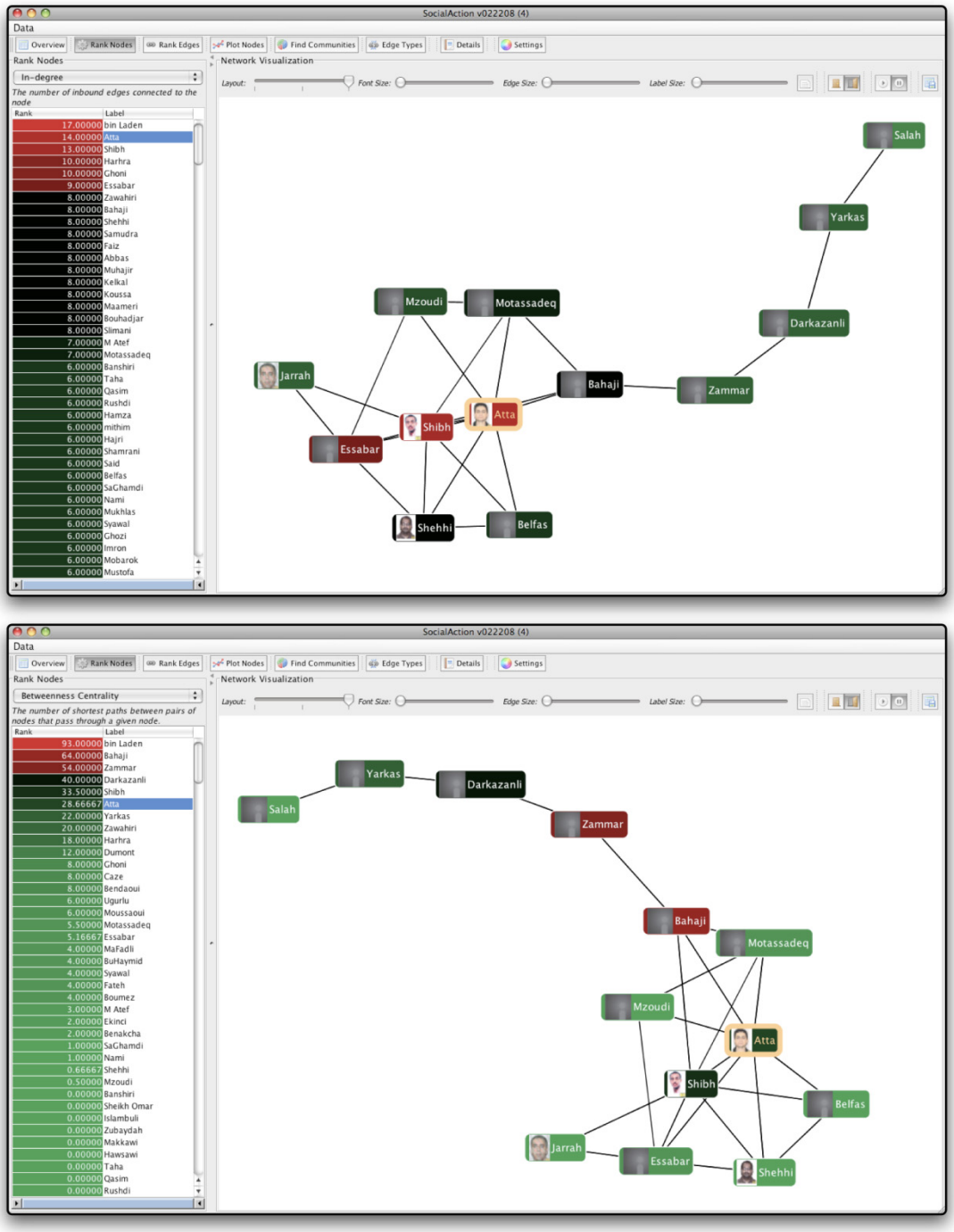


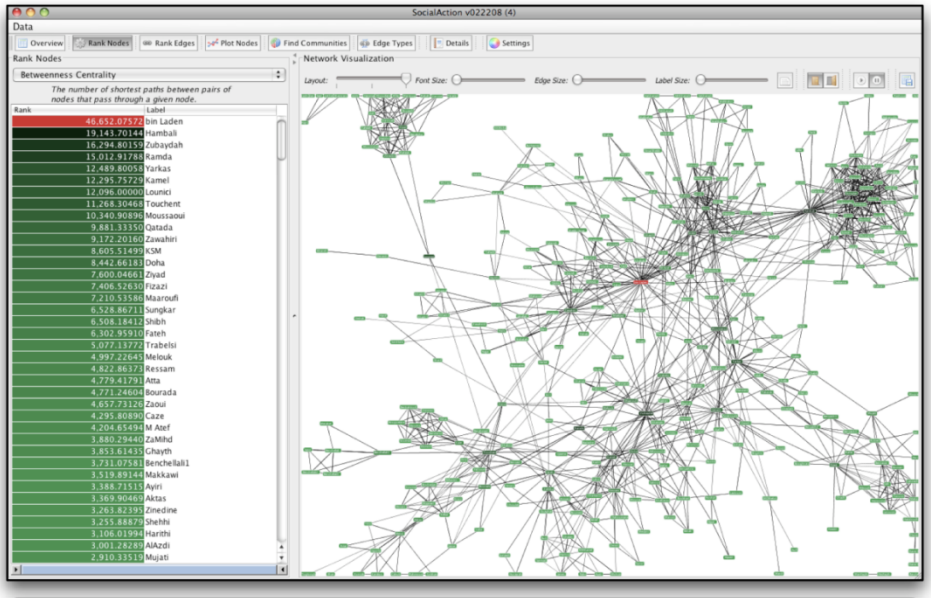
Figure 31. The *select* interaction technique for computed attributes, as implemented in *SocialAction*.

4.1.5 Filter: Reducing Complexity by Focusing on Important Data

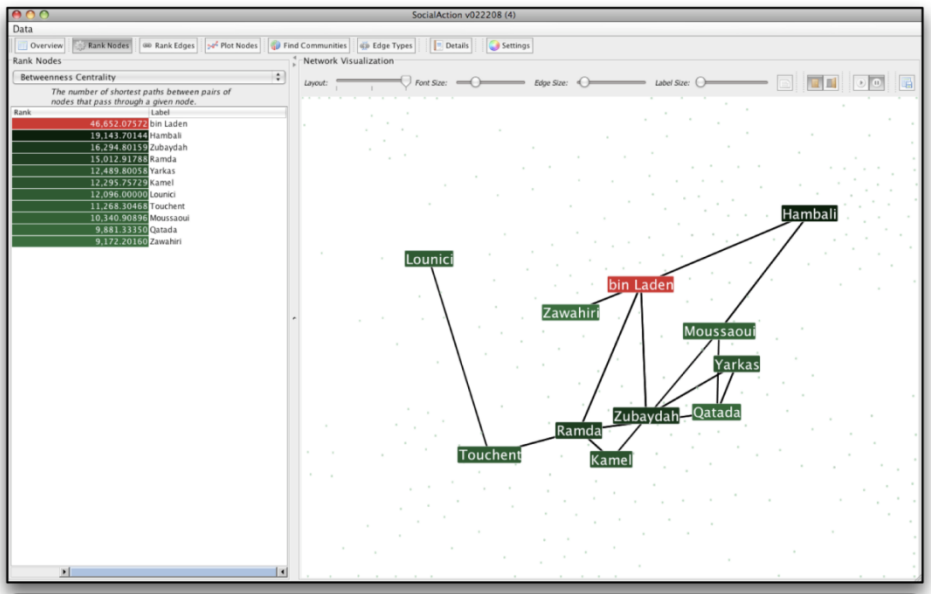
Filtering can decrease the complexity of the visualization by removing data points that aren't immediately relevant to the task of users. Many information visualization systems allow users to filter out certain data points based upon inherent attributes, and are most effective with dynamic queries and range sliders (e.g. [85]). However, few systems allow users to remove data points that are deemed statistically less important. Allowing users to filter by computed attributes is one way to achieve this goal. Although computed attributes can be displayed with visual encoding and coordinated with statistical views, resulting visualizations may still be too complex to comprehend. However, the ability to filter out data that is relevant according to task-related computed attributes is an effective way of reducing complexity.

The design goal of filtering by computed attributes should be integrated into all systems wishing to give users more control over the visualization display. Similar to inherent attributes, dynamic queries and range sliders should be used to give users freedom to see how the filtering process affects their data. *SocialAction* follows this design goal, which is extremely important in social network visualizations that are typically incomprehensible with more than 50 nodes and edges.

In this example, a social network visualization is made more comprehensible by filtering according to statistical rankings. The selected ranking is betweenness centrality, a social network analysis statistic that attempts to quantify the gatekeepers. The top image shows the network unfiltered, whereas the bottom shows the filtering enabled. In the right example, the important nodes can be read and the edges between them are apparent.



The complete social network before filtering.



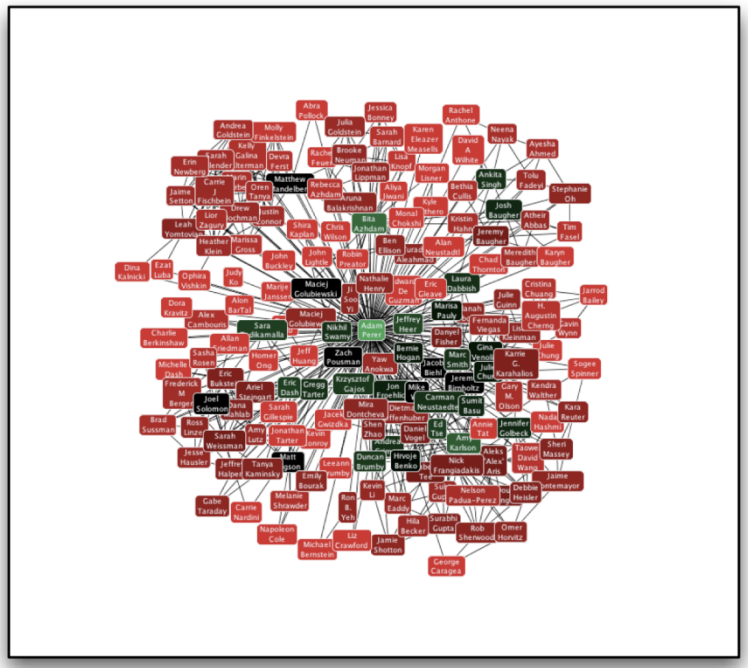
Nodes filtered except those with high *Betweenness Centrality*, a computed attribute that measures for gatekeepers.

Figure 32. The *filter* interaction technique for computed attributes, as implemented in *SocialAction*.

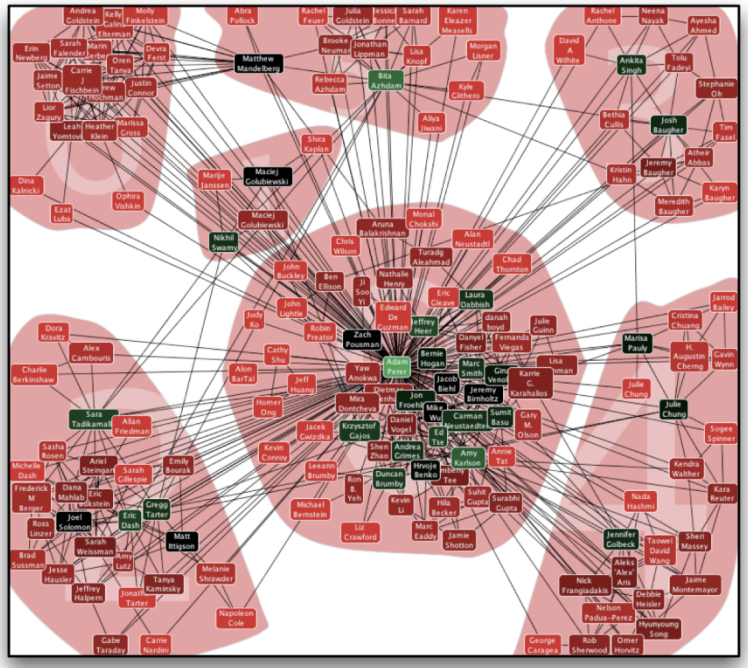
4.1.6 Abstract/Elaborate: Focusing on more or less detail

When information visualizations are too dense, abstracting the data into higher-level components can be useful. Inversely, when the information visualizations are too sparse, elaborating details can be effective. By abstracting or elaborating in statistically significant ways, users may understand the data more effectively. Clustering is one statistical technique that adds or reduces detail. For example, in a social network visualization, it can be difficult to discern where various communities of tightly-connected individuals exist. Abstracting and elaborating can both be used to display the statistical findings from the clustering.

In this example, the community information is elaborated into the visualization by surrounding each community with a surrounding polygon. This new information allows users to see which nodes belong to which community, and which relationships span multiple communities. Compared to the non-elaborated image on the left, rich information previously hidden from the human's eyes is now presented thanks to statistical algorithms. This community information could also be used to simplify the visualization by turning each community in a meta-node, as shown in [65].



A social network visualization using only a force-directed layout.



The social network visualization after elaboration with computed attributes from a community detection algorithm.

Figure 33. The *elaboration* interaction technique for computed attributes, as implemented in *SocialAction*.

4.1.7 Explore: Reaching insights through exploration.

Computed attributes can help guide users to *explore*. Algorithms from statistics and data mining are often created to find interesting properties of data. These results can act as suggestions for exploration by users. However, there are often a variety of algorithms on which to measure a data set so I focus this discussion on the exploration of computed attribute choices. For instance, SPSS, a leading statistical analysis tool, includes over 80 sophisticated statistical procedures (http://www.spss.com/spss/alpha.htm). Having quick access to being able to assess the usefulness of the algorithm is important.

4.1.8 Grouping of Statistical Algorithms by Task

Rather than present a lengthy list of statistical features to users, they should be organized according to tasks the users are trying to accomplish. By focusing on tasks rather than features, users can concentrate on their analytical goals: understanding their data.

In *SocialAction*, I took the complicated, opportunistic nature of social network analysis and organized the numerous techniques into 6 tasks. These tasks were based upon knowledge gained through interviews with analysts who described their methodologies. The 6 tasks, as described in [67], involve finding important individuals, relationships, and communities. This results in an interface that is task-based, allowing users to focus on discoveries and insights, rather than focusing on what techniques to use.

Obviously, different types of users may require different tasks. User interfaces should be robust enough to support multiple user types. Again, this robustness may add an additional layer of complexity to user interface design, but focusing on tasks rather than features will allow users to focus on discoveries rather than navigating menus.

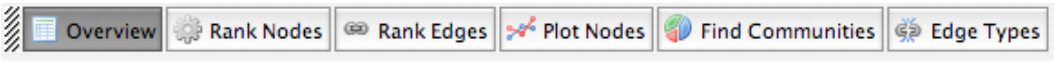


Figure 34. The Toolbar of *SocialAction* illustrates its organization of statistical features into tasks.

4.1.9 Choosing and ordering Statistical Algorithms by Usefulness

As mentioned above, there is no shortage of statistical algorithms to measure data. For instance, in social network analysis, nearly every author seems to invent a centrality measurement to suit their needs. Analysts generally have a finite amount of time to analyze their data, so bringing their attention to the most useful algorithms should be considered.

In *SocialAction*, I reviewed and tabulated the use of ranking algorithms in popular social networks journals, such as "Connections". Based upon their popularity of use, this provided guidelines for which algorithms were implemented. However, their order in the interface is also based upon their popularity. The effect of this is two-fold: explorers will be more often to select algorithms to analyze their data that is respected by their peers, and also give users quick access to the algorithms they most likely care about.

4.1.10 Optimize Statistical Algorithms

Algorithms should be optimized to run in real-time. If the algorithms are too slow, algorithmic results should be pre-computed when possible. Another strategy is to log which algorithms are most often run by the users and run them on a background thread when the user is performing less CPU-intensive actions.

4.1.11 Guiding Users through Algorithms

Statistical techniques can yield valuable discoveries, but typical data analysis tools typically support only opportunistic exploration that may be inefficient and incomplete. When the number of tasks is large and the algorithms are complex, guides can help domain expert users through complex exploration of data over days, weeks and months. In fact, I believe this idea is so important I dedicate the next section to defining this explicitly for both inherent and computed attributes.

4.1.12 Guiding Users with Systematic Yet Flexible Discovery

The integration of statistics and visualization emphasizes interaction which can lead to complex paths of exploration for discovery. For this reason, I present a refined architecture that uses *systematic yet flexible* (SYF) design goals to guide domain expert users through complex exploration of data over days, weeks and months [67]. The SYF system aims to support exploratory data analysis with some of the simplicity of an e-commerce check-out while providing added flexibility to pursue insights. The SYF system provides an overview of the analysis process, suggests unexplored states, allows users to annotate useful states, supports collaboration, and enables reuse of successful strategies. The affordances of the SYF system are

demonstrated by integrating it into a social network analysis tool employed by social scientists and intelligence analysts. The SYF system is a tool-independent component and can be incorporated into other data analysis tools.

4.2 Systematic Yet Flexible Guides

The increasing availability of digitized information encourages users to conduct more frequent and complex exploratory data analyses. The basic string search or SQL query are no longer adequate for advanced users who seek to understand patterns, discern relationships, identify outliers, and discover gaps. Data mining strategies, cluster analysis, and search engine results are helpful tools for such exploration, which typically takes days, weeks, or months. Domain experts may be trying to sift through gigabytes of genomic data to understand the causes of inherited disease, to filter legal cases in search of all relevant precedents, or to discover behavioral patterns in social networks with billions of people. For these challenging tasks, users must conduct repeated searches, combine results, and consult with colleagues. As they grow familiar with the data, they move from divergent conjectures to more careful hypothesis testing so as to collect evidence supporting their emerging insights.

Current tools can produce useful nuggets of information, but domain experts are increasingly aware of the need to shift from opportunistic discoveries to more systematic approaches. A *systematic* approach guarantees that all measures, dimensions and features of a data set are studied. Such an approach guides new users, ensures analysts of completeness, and facilitates cooperation during analyses that may take weeks or months. However, a wholly strict guide would undermine the

flexible needs of an analyst, as they will inevitably wish to pursue insights based on past successes, new information, fresh hypotheses, or unproductive directions. Legal searchers, who need to find every relevant case to avoid surprises, have developed paper-based and sometimes electronic tools to guide their work. Their goals are to ensure complete coverage, allow measurement of their progress, and enable team members to combine their partial results. Another expectation of systematic approaches is that they allow different users working independently to come up with largely similar results. Other professional examples demonstrating a similar need for systematic analysis include physicians completing diagnostic examinations, field biologists surveying forest grounds, and forensic scientists investigating murder scenes. Such professions have developed orderly strategies to assist investigators with challenging, non-trivial, multi-faceted exploration. Systematic-only approaches may suit some users' needs, but complex problems rarely yield to clean algorithmic strategies. If real problems were that simple, their solutions could be automated. Thus, systematic yet flexible strategies are emerging as a key

Figure 35. The 4 systematic steps for checkout at Amazon.com [4]. Users must step through all four stages in order, while the progress is this process is updated at each step.

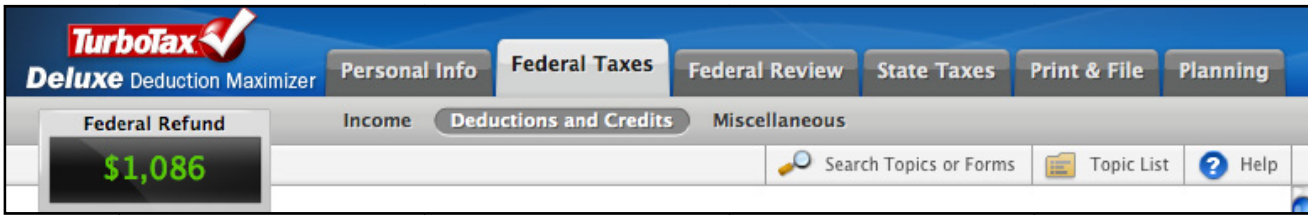


Figure 36. Intuit's TurboTax [48] guides users through the complex process of preparing tax returns in the United States. The top of the interface features tabs that allow users to complete steps in their own order, in case want to make changes or review. The interface also presents an overview of the user's expected refund or debt owed, and updates after each step.

topic in areas such as survey completion, job applications, and business process modeling. Such strategies are all the more central in the e-science community, where scientific workflow management and record keeping are issues of vital importance. E-science researchers must also address long duration projects, collaboration complexities, and guarantees of completeness [45, 87].

Most computer users have some form of experience with systematic interfaces, as they are pervasive in many common activities. The checkout process at Amazon.com [4], shown in Figure 35, provides an overview of the four steps users are required to complete before making a purchase. The process is simple and systematic, but inflexible in that it requires users to complete their purchase following a strict order of operations, as part of a one-time process which does not allow them to return to or revise entries weeks later.

A more sophisticated interface is Intuit's TurboTax [48], which guides users safely through the complex U.S. Internal Revenue Service tax filing procedures. TurboTax steps users through the process of entering required information. The top of the interface, shown in Figure 36, features secondary navigation tabs that allow users to complete steps in any order, in case they should wish to make changes or review previously entered information. The top of the interface presents an overview of users' expected tax refunds or debts owed, and updates after each question is answered. TurboTax then verifies that all appropriate forms are filled out before allowing users to print or file their taxes. While flexible to user preferences, the TurboTax system still does not explicitly track user progress for presentation in the header overview.

Inspired by these approaches, my goal is to enhance the tools available for data analysis with *systematic yet flexible* (SYF) support. Data analysis is not as simple as a purchase on a website or filling out tax forms, so I present seven design goals to handle these more challenging tasks. I integrate these design goals into my tool-independent SYF infrastructure. This infrastructure supports discovery through *systematic* and *flexible* exploration, as well as annotation, collaboration, and process reuse (Figure 37). This integration supports orderly exploration over weeks, record-keeping to support discovery claims, and collaboration with colleagues. This also supports the iterative process of returning to review earlier work and bold initiatives that break from the formulaic approach.

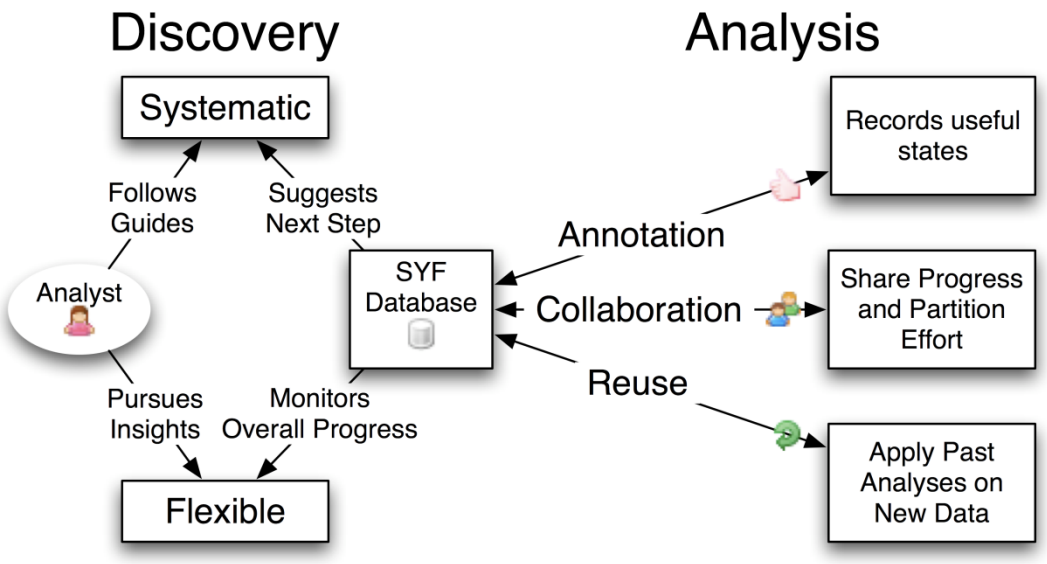


Figure 37. The SYF infrastructure facilitates discovery by providing systematic guides while also allowing users to flexibly pursue insights. SYF also facilitates analysis by allowing users to easily annotate during exploration, share exploration results with colleagues and partition effort, and reapply past exploration paths on new data sets.

I demonstrate the benefits of SYF by integrating it into *SocialAction* [65]. The maturity of social network analysis tools has not advanced as fast as the popularity of social network analysis. Numerous measures have been proposed by structural analysts to statistically assess social networks [99]. With a wealth of metrics, analysts want to be certain they are not overlooking critical facets of the networks in question. A design that allows social network analysts to quickly iterate and keep track of computed metrics is critical for exploring these vast statistical measures. The ability to share results, annotate key findings and reapply past measurements on new networks allow past efforts to not be wasted. The SYF system provides such benefits critical to social network analysts.

Systematic Yet Flexible Design Goals Enable users to:
1. See an overview of the sequential process of actions
2. Step through actions
3. Select actions in any order.
4. See completed and remaining actions
5. Annotate their actions.
6. Share progress with other users.
7. Reapply past paths of exploration on new data.

Table 1: Seven design goals for systematic yet flexible interface support

4.2.1 The SYF Infrastructure

I propose a set of seven design goals shown in Table 1. The first four goals provide *systematic yet flexible* discovery support by ensuring analysts of completeness and guiding novices. The last three goals support analysis by enabling annotations, collaboration and reuse. Each of these goals supports analysts who work over many days, weeks, or months. Furthermore, these design goals emphasize maintaining concentration to achieve task completion [82]. By showing users their prior, current and future steps, users are assisted when returning after inevitable distractions.

In order to facilitate the integration of SYF principles into data analysis tools, I provide an open-source infrastructure to tool developers. First, the tool developers register the systematic steps of exploration via SYF's application programming interface (API). Then, they register GUI events from their tool using the API and specify which steps the events belong to. SYF keeps track of user progress by maintaining a history of GUI events invoked. After developers augment their application with the SYF user interface, they can easily provide users with an overview, progress feedback, history navigation, annotation support, and the additional features listed in Table 1.

4.2.2 Supporting Discovery with Systematic Yet Flexible Guides

When users are exploring data, there are many paths and permutations to examine and users can easily get lost. The SYF system provides feedback to users about their current state, the actions they have already completed, and which actions

remain. This information gives confidence to users that they have made progress through the rich landscape of data analysis.

The SYF system, which augments a data analysis tool's interface, provides an overview of each of the systematic steps for completeness (Design Goal 1). The left-hand side of Figure 39 presents *SocialAction 3.0*'s seven systematic steps for social network analysis derived from practitioner interviews.

Users who wish to explore the data via SYF's *systematic* guiding can use the navigation buttons, also found on the left-hand side of Figure 5. When users are ready to continue analysis, they can click the 'Next' button to bring them to the next unvisited state (Design Goal 2), or return to a previous state using the 'Back' button. If users wish to explore the data in a *flexible* way, each step button acts as a secondary navigation button, much like a tab. Users can click this button to navigate to the actions required to complete the step (Design Goal 3).

Each step button features a progress bar. These meters give users a sense of how far away they are from completing the current step, as well as the entire data analysis (Design Goal 4). If users wish to view their path of exploration so far, they can launch the history panel. In Figure 40, a user's history is shown as a tabular list that is sortable by step number, state type, user action or annotation rating.

Users can also hide the SYF panel if they wish to focus on their work. By dragging the divider panel that separates SYF and the data analysis tool, they can shrink or minimize the guide. Even when the SYF interface is hidden, the user's actions are monitored so the benefits of SYF can be leveraged later.

4.2.3 SYF In Action: *SocialAction* 3.0's Node Rankings

One step in the defined systematic social network analysis path is ranking all nodes according to importance metrics. In Figure 38, an analyst has completed 40% of the current step. In order for users to finish this step, they must examine the rest of the node importance rankings. Information about completed rankings is not isolated to the SYF interface, but can also be integrated into the main UI of *SocialAction 3.0*. For instance, the combo box in which users select importance rankings are augmented with icons highlighting previously visited options (Figure 39). If users have already examined a ranking, a checkmark appears beside it. Similarly, if users have already made an annotation about this ranking, an annotation icon appears. *SocialAction* can look up this information about each ranking state by using the SYF system's API. Informing users in a consistent manner is important, as many users prefer to use secondary navigation instead of following all steps in order, depending on their hypotheses or experience [12].

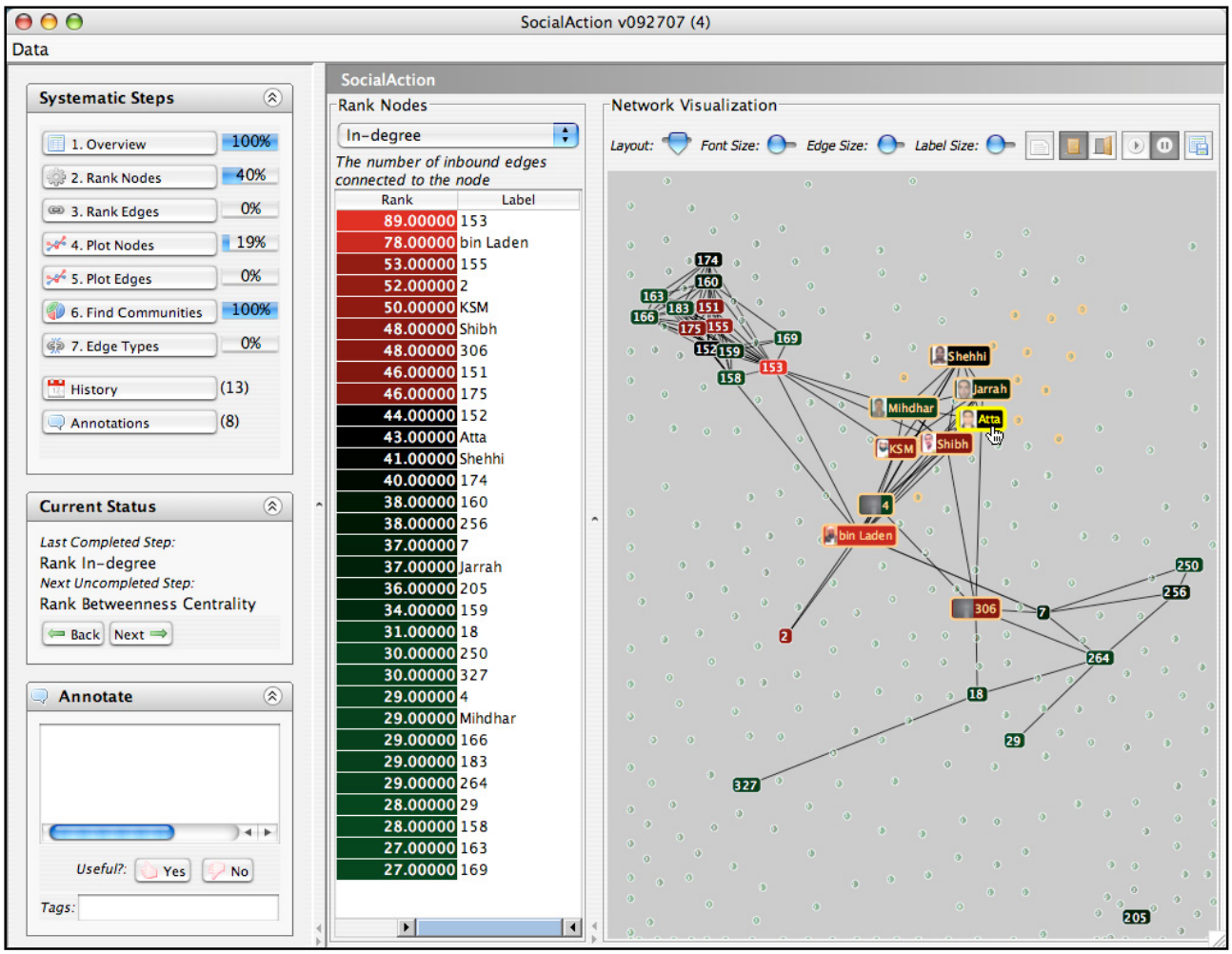


Figure 38. The SYF system integrated into *SocialAction 3.0*. The interface to SYF is presented on the left-hand side, whereas the main UI for *SocialAction 3.0* is on the right. This figure features a "Global Jihad" terrorist network that researchers are studying using *SocialAction*. In order to protect sensitive information, node labels have been anonymized except for those individuals publicly identified in the Zacarias Moussaoui trial.

4.2.4 Supporting Analysis with Annotation

Throughout the process of exploring data, users may come across important discoveries. The SYF system features a light-weight solution for users to annotate these insights quickly (Design Goal 5). Annotations are textual comments such as indications of insights, notes about informing partners about progress, or questions to be asked to collaborators. Often, annotations will deal with schedules, deadlines, reminders of tasks to be done, or the need to prepare for presentations. Useful annotations might be attached to objects being studied, such as indications of relative value of legal precedents or chemical structures. I augment these textual annotations with ratings and tags so they can be easily found later.

During any stage of data exploration, analysts have access to the annotation functionality shown in the panel on the left of Figures 38 and 41. This persistent panel allows users to quickly comment, rate and tag any state of analysis. Users can write their insights in the enlargeable text editor.

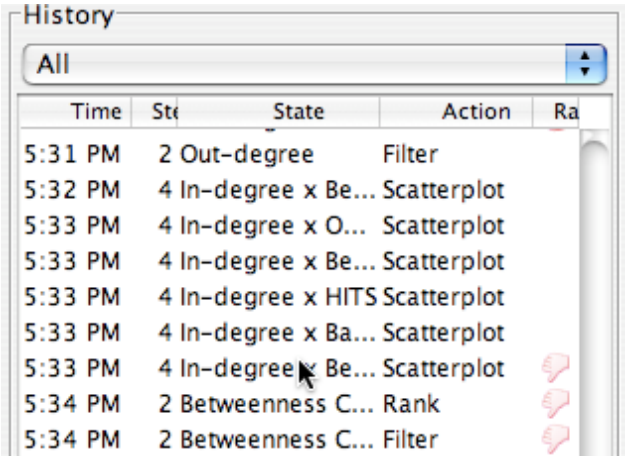


Figure 39. SYF's History panel shows users' past actions in tabular form. Users can navigate by sorting by step number, state type, user action or annotation rating. A 'Date' column also appears when analysis takes place over multiple days. Furthermore, users can filter based on "important" or "unimportant" annotations using the combo box at the top. Users can jump back to a previous state by clicking the 'Go' button.

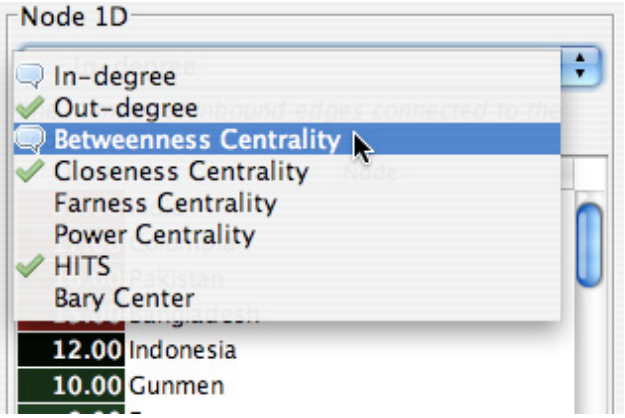


Figure 40. The Combo Box in the *SocialAction* 3.0 GUI provides feedback in the form of a checkmark icon to show which measures have been computed previously.

Users can also mark a state as interesting via the 'thumbs up' button, uninteresting via the 'thumbs down' button, or tag the state with meaningful words or phrases. Users can also choose to mark this state and comment with a tag in the 'Tag' text field. Whenever users return to an annotated state of analysis, the annotations reappear automatically in this space.

Users can review all past annotations by clicking the annotation button located below the systematic steps. The number next to the annotation link informs users how many annotations have been composed. In the annotation panel, shown in Figure 41, users can browse all annotations, keyword search for specific annotations, navigate using the tag cloud for tagged comments [35], or filter based on rated interestingness. Users can select individual annotations from a sortable, tabular list

where they can read the comment or jump back to the state where the annotation was written.

In addition to allowing users to return to interesting states for further exploration, annotations are useful when users wish to create reports about their findings. Since useful discoveries have been recorded, users can export the images, tables and descriptions associated with interesting states into word processors or web pages.

4.2.5 SYF In Action: *SocialAction* Communities

I illustrate the annotation functionality in another step of social network analysis: community detection. One of the main goals of sociologists studying social networks is to find cohesive subgroups of nodes [24]. *SocialAction*'s algorithms automatically determine communities based on their link structure, to help users find groups of nodes that are closely connected in the network. Communities are visually represented by surrounding all members with a translucent convex hull as shown in the right side of Figure 41.

In this example, the user is browsing all annotations created with SYF. The tag cloud shows the user's tags for all annotated states, and the tabular list shows all annotations. The last annotation is selected and displayed below. The user activated this state by clicking the "Go" button and can review and continue analysis.

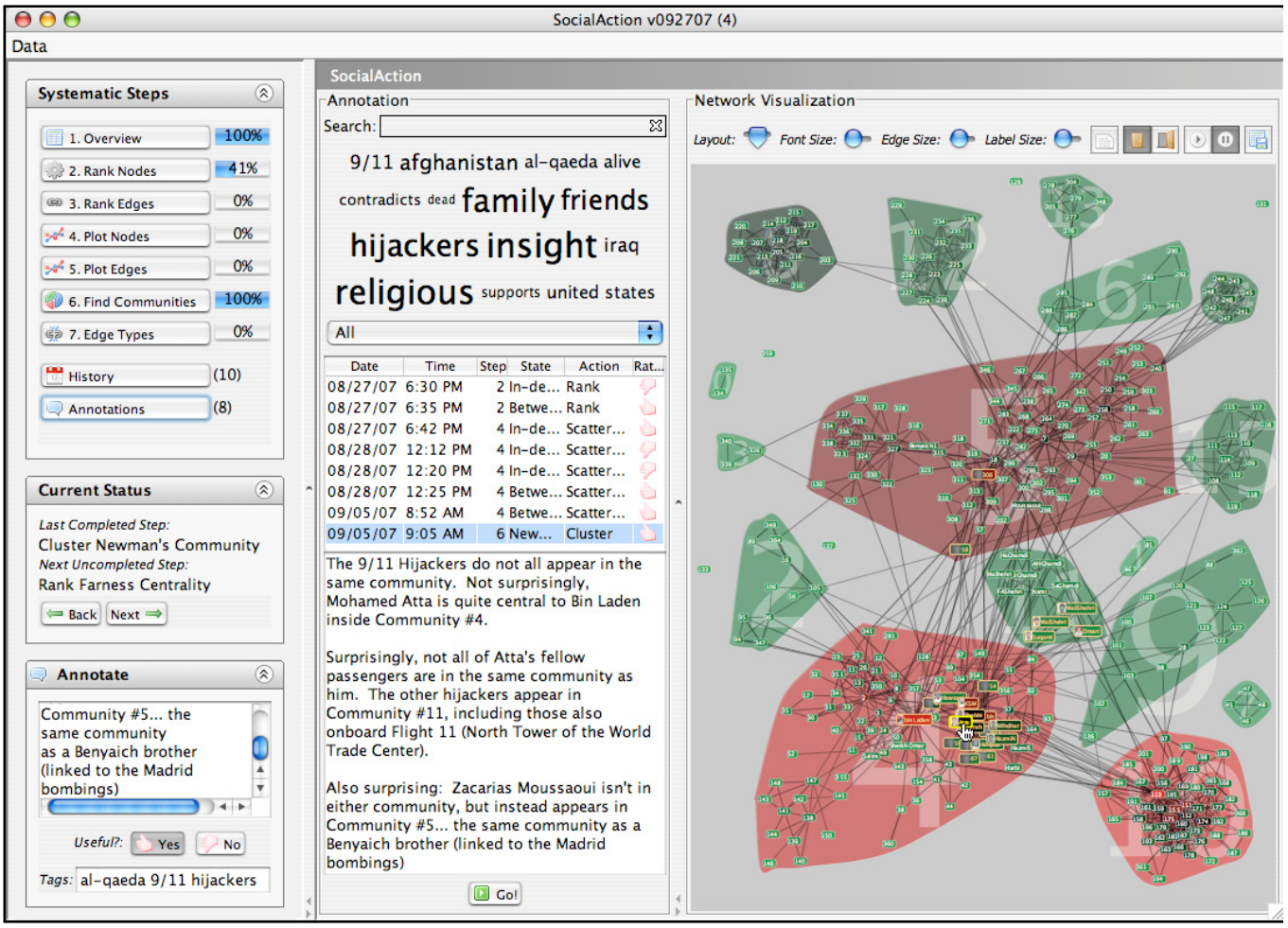


Figure 41. This figure is shows the annotation features of SYF. Users can browse their annotations by selecting the annotation button located at the bottom of the systematic steps panel. Users can keyword search, navigate using the tag cloud, or filter based on the rating to find specific comments. When a user selects an annotation from the resulting tabular list, it is displayed below. Users can jump to the state where the annotation occurred by selecting the 'Go' button.

4.2.6 Supporting Analysis with Collaboration

New evidence has emerged suggesting that communication and collaborations are necessary components of successful visualization systems [95]. User studies also suggest that supporting collaboration with visual data analysis can help people explore a data set both broadly and deeply [40].

The SYF system supports collaboration by allowing users to easily share their exploratory paths and insights that were annotated during their data analysis (Design Goal 6). Since SYF monitors each interaction and allows users to specify useful states, analysts can easily export interesting states to colleagues. Furthermore, users can partition effort during analysis. After users finish a segment of analysis, they can share their completed results. Recipients will know which analyses have been performed and annotated and will be empowered to not duplicate past efforts.

4.2.7 Supporting Analysis with Reusable Exploration

In addition to user-to-user collaboration, SYF also supports data-to-data collaboration. Users can repeat analyses conducted on previous data to new data sets (Design Goal 7). For instance, if a user already found several useful states during exploration and marked them as useful in the annotation panel, they could reuse these "best practices" on new data, as if it was a macro. Analysts can quickly see if the same patterns, gaps or outliers are present in the new data set.

4.2.8 SYF In Action: Comparing Networks in *SocialAction*

I illustrate an example of reusing past exploration in *SocialAction 3.0*. This example comes from colleagues studying social networks that span thirty years. In order to grasp the dynamics of a network, they often study a year's data independently and then make comparisons to other years. Instead of repeating calculations on every year manually, SYF allows these analysts to automatically compute and present analysis after the first exploratory path has been defined. Social scientists often collect and input data manually and sometimes the visualizations present coding mistakes in the data. In this situation, users need to fix the mistakes in the original dataset. Instead of starting over from the beginning, analysts can use SYF to reapply all past analyses and continue to make progress.

4.2.9 Defining a Systematic Path to Completeness

Understanding the domain experts' tasks is necessary to defining the systematic steps for guided discovery. Although some professions such as physicians, field biologists, and forensic scientists have specific methodologies defined for accomplishing tasks, this is rarer in data analysis. Interviewing analysts, reviewing current software approaches, and tabulating techniques common in research publications are important ways to deduce these steps.

For instance, even though there are many importance rankings, clustering algorithms, and statistical techniques for assessing social networks, there is no well-defined methodology for performing these operations [99]. During the design of *SocialAction* I conducted in-depth interviews with six social network practitioners to understand their current work habits. Since most social network practitioners were not using

visualizations during their exploratory analysis, these findings were augmented with several key principles from the information visualization community. The tenets of the Visual Information Seeking Mantra [80] ("Overview first, zoom and filter, details-on-demand") were kept in mind when ordering the tasks of social network analysts. Furthermore, the Graphics, Ranking, and Interaction for Discovery (GRID) principles [79] ("Study 1D, study 2D, then find features. Ranking guides insight, statistics confirm") also shaped the systematic method for analyzing social networks. The resulting 7-step methodology for social network analysis, integrated into *SocialAction 3.0*, is:

1. Overall network metrics (e.g. number of nodes, number of edges, density, diameter)
2. Node rankings (e.g. degree, betweenness, closeness centrality)
3. Edge rankings (e.g. weight, betweenness centrality)
4. Node rankings in pairs (e.g. degree vs. betweenness, plotted on a scattergram)
5. Edge rankings in pairs
6. Cohesive subgroups (e.g. finding communities in networks)
7. Multiplexity (e.g. analyzing comparisons between different edge types, such as friends vs. enemies)

This is not the only systematic method for social network analysis, but one that will assure analysts they have explored relevant features in *SocialAction 3.0*. This methodology is evident in the SYF user interface that augments *SocialAction 3.0* (left side of Figures 38 and 41).

4.2.10 Summary

This chapter demonstrates that in order to design for exploratory data analysis, rich interactions need to be available to support the creative tasks of analysts. In information visualization systems, interaction controls typically focus on *inherent* attributes, rather than *computed* attributes. Using an established taxonomy of interaction techniques (Select, Explore, Reconfigure, Encode, Abstract/Elaborate, Filter, and Connect), design guidelines are presented to enable system designers to employ *computed* attributes with statistics and visualization. This allows practitioners to yield generalized lessons learned from the design of *SocialAction*.

Although these interaction techniques are powerful, they can lead to complex paths of exploration. In order to support analysts, I provide *systematic yet flexible* (SYF) techniques that guide users through the analytical process. Users are able to see an overview of a sequential process to complete analysis. They can step through actions or choose them in any order. Regardless of the path of exploration, users can see the completed and remaining actions left. In addition, support for annotation, collaboration, and reusable exploration is beneficial for aiding the exploration of users. These design goals improve the process of exploratory data analysis.

Chapter 5: Evaluation

In this chapter, I describe Contribution C3:Evaluation:

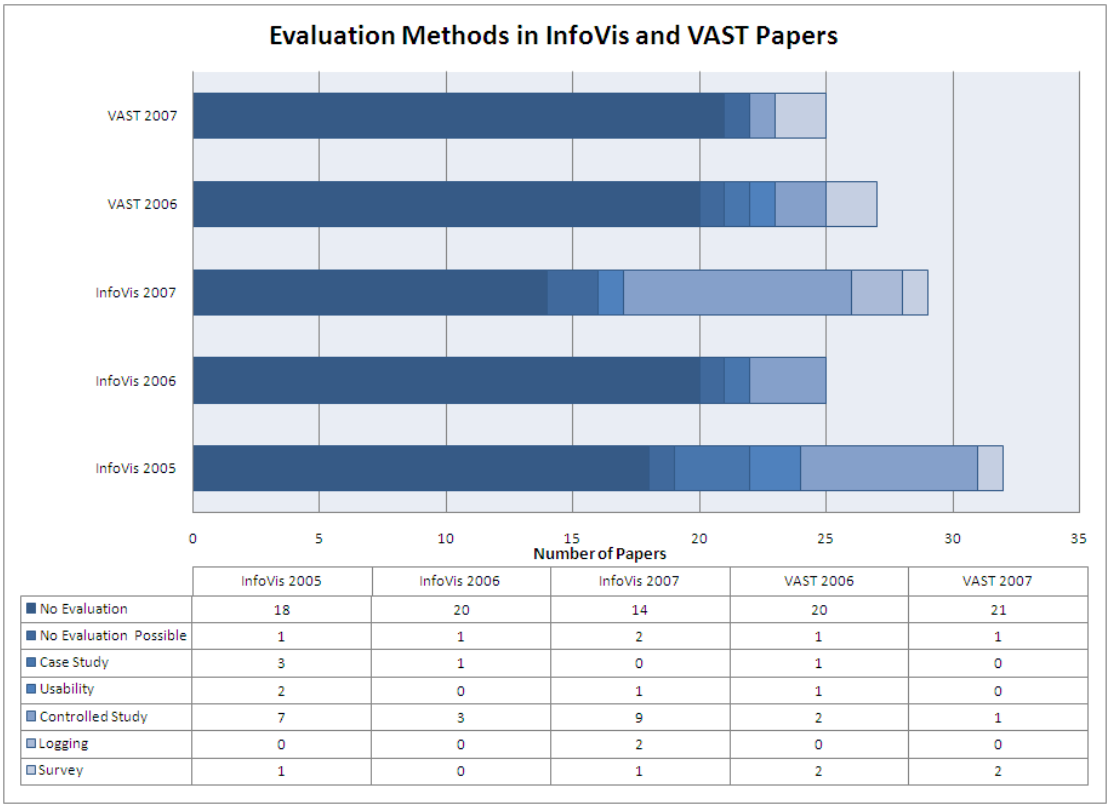
> *Demonstrates the effectiveness of long term case studies with domain experts to measure creative activities of information visualization users.*

Traditional laboratory-based controlled experiments have proven to be effective in many user interface research projects. When new widgets, displays, interaction methods, or input devices are being developed, controlled experiments can compare two or more treatments by measuring learning times, task performance times, or error rates. Typical experiments would have 20-60 participants, who are given 10-30 minutes of training, followed by all participants doing the same 2-20 tasks during a 1-3 hour session. Statistical methods such as t-tests and ANOVA are applied to show significant differences in mean values. These summary statistics are effective, especially if there is small variance across users.

However, because domain experts work for days and weeks to carry out exploratory data analysis on substantial problems, their work processes are nearly impossible to reconstruct in a laboratory-based controlled experiment, even if large numbers of professionals could be obtained for the requisite time periods. A second difficulty is that exploratory tasks are poorly defined, so telling the users which tasks to carry out is incompatible with discovery. Third, each user has unique skills and experience, leading to wide variations in performance which undermine the utility of summary statistics. In controlled studies, exceptional performance is seen as an unfortunate outlier, but in case studies, these special events are fruitful critical incidents that provide insight into how discovery happens. Fourth, I wanted more than

quantitative analyses of the tool. I also wished to hear about the problems and frustrations users encountered as well as their thrilling tales of success [46]. For such reasons, I turned to structured and replicated case study research methods to collect supporting evidence for the conjecture that integrating statistics with visualization would facilitate discovery for social network analysts.

The novelty of structured and replicated case studies is apparent from a review of the 132 papers in the 2005-2007 IEEE Information Visualization and the 2006-2007 Visual Analytics Science & Technology Conferences. Only 39 papers had any



	InfoVis 2005	InfoVis 2006	InfoVis 2007	VAST 2006	VAST 2007
No Evaluation	18	20	14	20	21
No Evaluation Possible	1	1	2	1	1
Case Study	3	1	0	1	0
Usability	2	0	1	1	0
Controlled Study	7	3	9	2	1
Logging	0	0	2	0	0
Survey	1	0	1	2	2

Figure 42. An analysis of InfoVis (2005-2007) and VAST (2006-2007) Papers. This analysis shows how a majority of papers feature no analysis, and most evaluations are controlled studies. No papers in these conferences have had structured, replicated case studies.

user evaluation and each tested users for less than 2 hours of tool usage. Furthermore, all but 9 of these tests used domain novices who were given standard tasks.

5.1 Evaluation Methodology

Evaluating systems for information visualization tools is problematic because controlled studies may not effectively represent research strategies. Information visualization can differ from other fields of HCI since systems are designed to be exploratory: the set of tasks users may want to perform is not known. For these reasons, I designed a methodology to evaluate *SocialAction* with four case studies involving researchers who worked on their own data with their own problems. Inspired by the goals of MILCs [84], I developed a methodology for studying the effectiveness of *SocialAction*. Of course, there is a long history of long-term qualitative case studies (e.g. [107]), but this methodology takes into account the unique demands of information visualization users. The methodology also takes into account the lessons learned from Seo and Shneiderman [78], but provides a structured and replicated approach with diverse domain experts.

1. Interview (1 hour): This initial phase involves an interview to understand the intentions of the domain experts. The achievement of the intentions acts as one benchmark of success at the end of the study. Furthermore, this phase acts as an opportunity for observers to decide if the domain experts are appropriate candidates for the study. This evaluation was limited to knowledgeable domain experts conducting serious research with well-defined goals.

2. Training (2 hours): Users participate in a training session with the software developers. The domain experts are expected to use *SocialAction* to find insights during this practice analysis session. After the training session, users have access to a brief instruction manual.

3. Early use (2-4 weeks): Domain experts install *SocialAction* in their workplace where they load their own data relevant to their research goals. Each week, observers visit the domain experts' workplaces to interview them regarding their progress. For case studies involving remote locations, interviews occur over the phone. In the tradition of action research [98], the developers try to accommodate domain experts needs by modifying and adding features to the software to meet critical needs.

4. Mature use (2-4 weeks): This phase features more hands-off, "ethnographic"-style observation. No further improvements are made to the software despite requests from domain experts. Similar to phase 3, researchers visit each domain expert's workplace or conduct phone interviews. The software developers continue to provide technical support as needed.

5. Outcome (1 hour): This exit interview provides domain experts a formal chance to explain how the software impacted their research. The domain experts revisit their original intentions from Phase 1 and rate each intention based on the level of achievement.

5.2 Data Collection

In each of the five phases, the primary data collection method was interviews. The interviews were conducted on-site in the domain expert's workplace. One domain expert was based in the Boston-area, so interviews for him were conducted over the phone. Notes from each interview were recorded as field notes, and they were later transcribed digitally. An example of field notes is located in Appendix A.

The on-site interviews in the Early use and Mature use phases (Phase 3 and 4) occurred weekly. The domain experts would spend at least an hour recalling their insights from that week's efforts using *SocialAction*. Generally, screenshots from *SocialAction* would be shown in conjunction with each insight. Furthermore, the domain experts would often recreate the path of discovery in *SocialAction* to demonstrate how they got there. *SocialAction* did not automatically log the actions of each domain expert, so the interviews required the domain expert to manually step through the steps discoveries.

In the Early use phase, domain experts were able to request additional features in the spirit of action research [98]. In the interviews, the domain experts would highlight ways that they thought certain insights were being impeded by certain missing features. A master list of all feature requests was documented. I would then estimate to the domain experts how long each feature would take to implement. The domain experts would then prioritize the features they would prefer to have before the next interview session. In the Mature use phase, domain experts could still make suggestions but no new features were added to *SocialAction*.

After the Early use and Mature use phases, a closing interview was conducted on-site. I would share the highlights of my field notes and make sure everything was documented correctly and precisely. We would then discuss how *SocialAction* helped or impeded the generation of hypotheses and insights throughout the case study. Here, they could also discuss their future plans and expected outcomes. In the following sections, I describe in case study in full detail, beginning with the initial expectations from the first interview, to the discoveries made in the early and mature use phases. Finally, I conclude with their outcomes, which led to scientific publications and internal reports. One sample summary of the field notes from one case study is located in the Appendix.

5.3 Case Studies with Domain Experts

In order to validate my claims, I conducted four case studies of domain experts with diverse skill sets, domains of knowledge, and social network expertise. The domain experts were not recruited, but instead sought out *SocialAction* on their own after facing challenges in making sense of social networks. The descriptions of the case studies below only discuss a fraction of the domain expert's insights but are representative of their overall experience. The domain experts were given *SocialAction* as described in Chapter 3. The systematic yet flexible interface enhancements discussed in Chapter 4 were not evaluated.

5.4 Case Study 1: Senatorial Voting Patterns

Congressional analysts are interested in partisan unity in the United States Senate. For instance, *Congressional Quarterly* calculates such unity by identifying

every vote in which a majority of Democrats voted opposite a majority of Republicans, and then counts, for each senator, the percentage of those votes where they voted with the party. This metric can be useful for tracking an individual senator's party loyalty from year to year, but it does not tell much about the overall patterns in the body. Chris Wilson, then an associate editor for the *US News & World Report,* was interested in voting patterns among United States senators.

Wilson was seeking to uncover senatorial patterns, such as strategic, bipartisan, or geographic alliances in the data set. Wilson spent significant effort mining voting data from public databases, but was unable to find such distinct patterns through his normal methods of analysis.

Wilson believed social network analysis could yield the answers he sought. His data included voting results for each senator during the first six months of 2007, beginning when the Democratic Party assumed control of the chamber with a one-seat majority. A social network can be inferred from co-occurrences of votes. Before contacting us, Wilson tried to visualize this data in KrackPlot [53], ManyEyes [96] and NetDraw [8] but did not manage to find any interesting patterns.

5.4.1 Early Use

From the data, Wilson constructed the network such that when a senator votes with another senator on a resolution, an edge connects them. The strength of each edge is based on how often they vote with each other (e.g., Barack Obama and Hillary Clinton voted together 203 times, whereas Obama and Sam Brownback voted together only 59 times). This leads to a very dense network because there are certain uncontroversial resolutions that all senators vote for (e.g. Resolution RC-20, a bill

commending the actions of "the Subway Hero" Wesley Autrey). All senators are connected, which leads to a visualization of a huge, tangled web. *SocialAction*'s interactive statistics empower users to dig deeper, without forcing users to choose an arbitrary cut-off before analysis begins.

SocialAction allows users to rank edges according to importance metrics. Wilson used this feature to compare network visualizations by dynamically filtering out relationships with low importance rankings. For instance, the 180-vote threshold (about 60 percent voting coincidence) is shown in Figure 43a. Partisanship is strong even at this fairly low threshold, and the Republican senators who are most likely to vote with Democrats (Collins, Snowe, Specter, and Smith) are evident. This suggests that, in this particular Senate, although both parties are partisan, Republicans are less so than Democrats.

As the threshold increases, the bipartisan edges diminish (Figure 43b). Another unexpected consequence was that the Democrats stay more tightly unified than the Republicans as the threshold increases. Wilson believed this interaction beautifully illustrated the Democratic caucus's success in keeping members in line, an important fact when reviewing legislative tactics. The integration of statistics and visualization made this discovery possible.

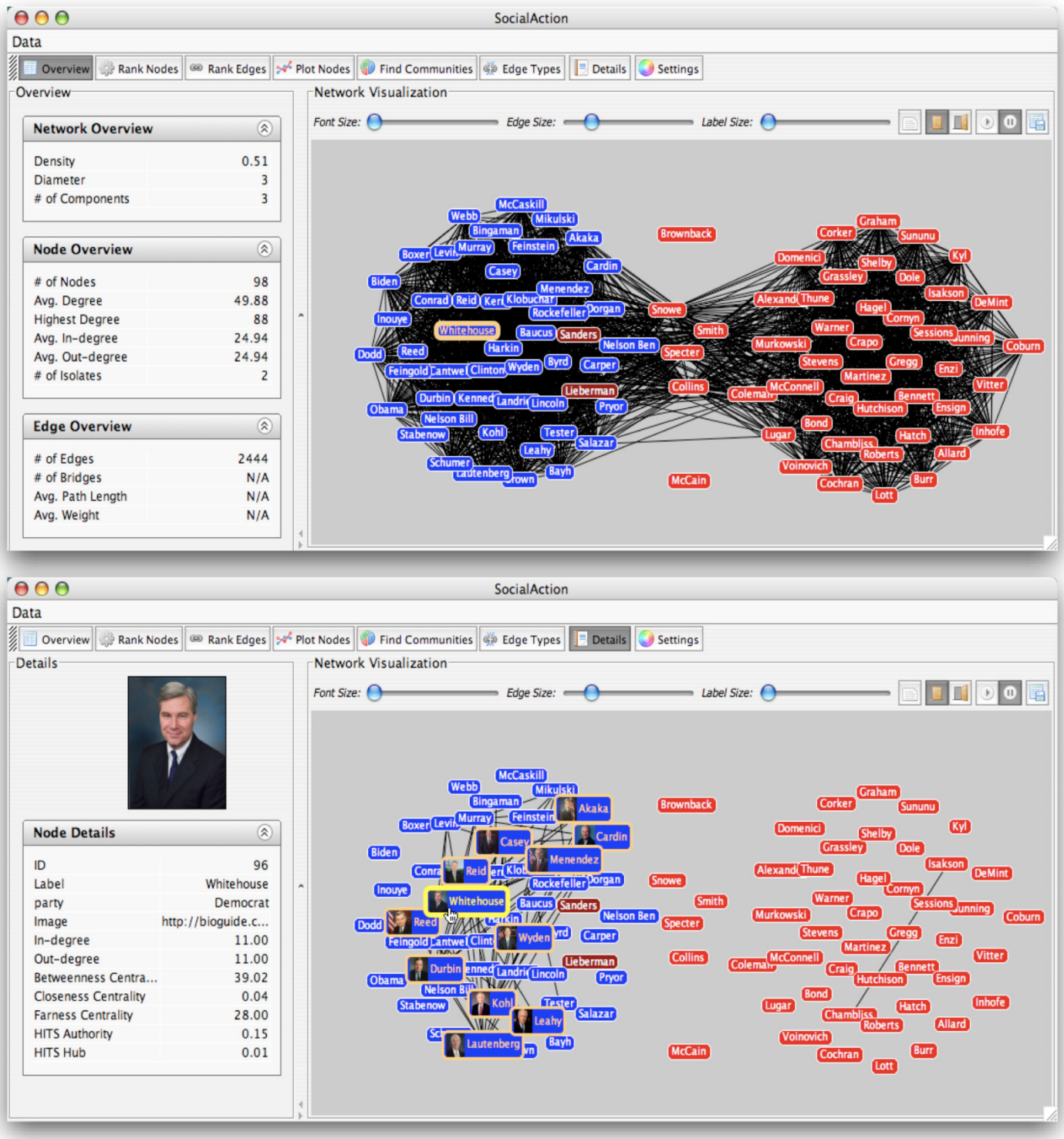


Figure 43. The social network of the U.S. Senators voting patterns (98 nodes, 4753 edges). Republicans are colored red, Democrats blue and Independents maroon. In the top image (a), the partisanship of the parties appeared automatically (180 vote threshold). In the bottom image (b), the threshold is raised to 290 votes. The Democrats' relationships are much more intact than the Republicans. Details-on-demand are provided for Senator Whitehouse, the senator with the highest degree at this threshold.

5.4.2 Mature Use

In order to determine patterns of individual politicians, Wilson used the statistical importance metrics of *SocialAction*. The capability to rank all nodes, visualize the outcome of the ranking, and filter out the unimportant nodes led to many discoveries. Wilson stated, for instance, that the *betweenness centrality* statistic turned out to be "a wonderful way to quantitatively measure the centers of gravity in the Senate". *SocialAction* made it evident that only a few senators centrally link their colleagues to one another. Wilson was also able to use the interactive clustering algorithms of *SocialAction* to "uncover geographic alliances among Democrats". These findings are just a sample of the sorts of insights that eluded Wilson prior to his analysis with *SocialAction*.

5.4.3 Outcome

Wilson was impressed with the discoveries that *SocialAction* helped reveal. The tight integration of statistics and visualization allowed him to uncover findings and communicate them to his peers both at his publication and on Capitol Hill. *SocialAction* received so much attention internally that the magazine hopes to replicate some of its functionality for its online readers. This will provide readers with further data analysis opportunities, in the spirit of [95]. Since the case study, Wilson has moved to Slate Magazine but still uses *SocialAction* for investigative reporting. So far, analysis from *SocialAction* has led to an interactive feature analyzing the social networks of steroids users in Major League Baseball [68], with more stories planned for the future.

5.5 Case study 2: Knowledge discovery for medical research

The National Library of Medicine (NLM) maintains PubMed, a search engine with access to more than 17 million citations in the health sciences. A recently revised feature of PubMed is the related article search. This feature aims to improve knowledge discovery by linking together critical information that may be missed by keyword searching. When users reach a citation of interest, five related articles are suggested on the screen. Sophisticated information retrieval algorithms generate these recommendations automatically. Jimmy Lin, a Ph.D. expert in information retrieval, led the project at NLM.

Lin and his colleagues sought to understand the usefulness of the recommendation algorithm. A successful algorithm would allow users to browse the document collection using the related articles links and reach other relevant documents. A network of documents can be created by linking together each document with its recommendations from the algorithm. The network's structure is important, since isolated documents without links from other relevant documents cannot be reached by browsing. Lin hoped to gain deeper insights about the usefulness of the algorithms by using SNA to explore the recommendation network. The recommendation network is not a "social" network, but demonstrates that although *SocialAction* is designed to support social network analysis, it also allows users to explore and interpret non-social networks, such communication, financial and biological and citation networks.

5.5.1 Early Use

For the experimentation with *SocialAction*, Lin used data from a TREC genomics test set [44]. This set was chosen because there was ground truth on the relevance of documents (such as results for the query "what is the role of the gene GSTM1 in the disease Breast Cancer"). Lin then generated document networks, where for each known relevant document, the top five related documents were linked (e.g., the suggestions from the related article search in PubMed). Upon loading the network for the first time in *SocialAction*, a eureka moment occurred. Lin proclaimed, "This figure is exactly what I wanted to see!"

Two phenomena were immediately noticeable from the visualization. First, relevant documents tend to cluster around each other (notice the dense red cluster in the middle of the network in Figure 44). This supports the cluster hypothesis in information retrieval, which proposes that relevant documents tend to be more similar to each other than to non-relevant documents [92]. However, there were also a number of isolated islands of documents (notice the disconnected, star-shaped clusters in Figure 45). These represent documents that would be unreachable by users when using the related article feature, undermining the goals of that feature.

Lin used a variety of the exploratory features of *SocialAction*. For instance, he used the importance rankings for nodes to find the most suggested articles, or the gatekeeper articles that bridge two clusters together. However, Lin's initial goal was to characterize the effects of the related article search, as opposed to refining the algorithm. Thus, Lin focused mostly on overall network statistics (such as number of disconnected components, density, and diameter) to quantify the output of the

retrieval algorithm. Figuring out which statistics are useful is often an under-surveyed problem of analysis tools. *SocialAction*'s design, which supports users quickly iterating through measurements while maintaining a constant visualization, served a useful role in this exploration.

Lin also requested additional features for *SocialAction*, such as the capability to calculate statistics for nodes with certain attributes (e.g., the number of relevant documents linked from each relevant document). Since Lin also was interested in using the statistical information to inform his retrieval algorithm, an exporter for the statistics was built.

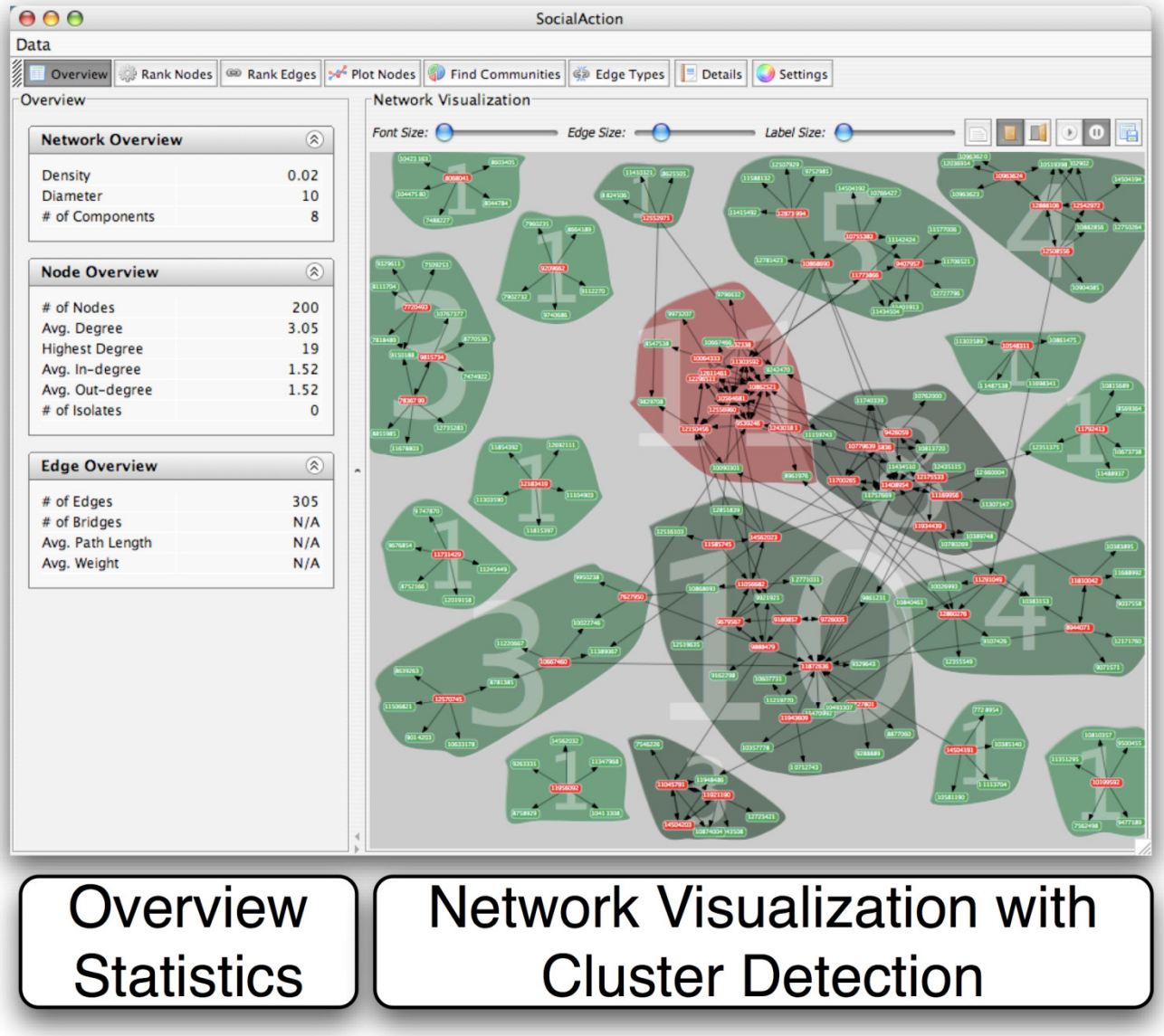


Figure 44. The recommendation network of a query on PubMed documents (200 nodes and 305 edges). Relevant documents are red, non-relevant are green. The community algorithm highlights closely-connected clusters in the network. Communities are color-coded by the percentage of relevant documents and labeled by the number of relevant documents.

5.5.2 Mature Use

With the requested features implemented, Lin used *SocialAction* to study 49 different query networks. Each of the networks had varying properties (number of suggested articles, number of relevant documents, density). The integration of statistics and visualization allowed Lin to quickly explore the networks, spending less than a few minutes on each network after becoming comfortable with *SocialAction*. This exploratory investigation led to the visual insight that networks with more relevant documents (red nodes) clustered together tend to have fewer the disconnected components.

Lin also used the clustering features of *SocialAction* to find tight-knit groups of articles that are highly similar to each other. Figure 44 shows the network components broken down into smaller communities using the hierarchical clustering algorithms available in *SocialAction* [65]. Each community is surrounded by a bubble colored based upon statistical information chosen by users (in this case, the average number of relevant documents). This visual evidence supports the cluster hypothesis Lin sought to confirm. *SocialAction* allows users to control the size of the clusters, digging deeper and deeper into the closest-knit groups. However, while this feature allowed Lin to advance his exploration, he chose to leave these results out of his analysis due to the subjective nature of cluster size.

5.5.3 Outcome

Using *SocialAction*, Lin and his colleagues were able to better understand the performance of their retrieval algorithm. The analysis showed that users can access most of the relevant documents by clicking on the related article links (e.g., without

having to go back to the search results and reformulate a query). However, they also identified isolated clusters, which represented relevant documents that were not reachable by browsing. The results of this analysis led to a publication of a high-quality research article in a prestigious information retrieval journal [56]. The exploratory nature of *SocialAction* allowed the researchers to measure their algorithms even though they had no prior knowledge of which SNA statistics would be useful. They also believe *SocialAction* will be a useful tool for verifying the effectiveness of new recommendation algorithms for PubMed.

5.6 Case Study 3: Engaging Hospital Trustee Networks

A Northeastern healthcare insurer is interested in engaging hospital boards in their region to speak loudly about healthcare quality. They are using social network analysis to help inform and prioritize this initiative. They hired Bruce Hoppe, a professor at Boston University, who also serves as a consultant aiding businesses in optimizing their operational networks. He uses social network analysis to accelerate business results and has experience with many Fortune 500 companies. Despite having a repertoire with over 8 social network analysis software tools (including [8, 9, 11, 53]), he has yet to find a suite that achieves his needs in exploring data effectively. For this reason, he was interested in integrating *SocialAction* into the workflow of his latest project.

5.6.1 Early Use

Hoppe began using *SocialAction* to analyze the board interlock network of over 500 organizations (such as hospitals, businesses, and non-profits) provided by the

healthcare insurer. These 500 organizations had a total of almost 8,000 board members. Hoppe was pleased that *SocialAction* could load all of the data at once and provide an overview of the whole network. In general, he was used to cropping data before analysis.

After seeing the large network, the healthcare insurer asked Hoppe to focus on a subset of the network: the hospitals and their boards of trustees (1740 nodes and 1854 edges). Unlike other SNA tools, *SocialAction* allows users to compare different but related varieties of statistical measures on a scatterplot. When Hoppe noticed this feature, he became interested in the relationship of *degree centrality* and *betweenness centrality*: to what extent were trustees sitting on many boards also the gatekeepers who connected many diverse hospitals. The scatterplot enabled Hoppe to quickly spot patterns in the healthcare network and the important outliers (Figure 45). In particular, a relatively unknown "Trustee 527" (anonymized for confidentiality) emerged as a focus of attention due to her unique position of few hospital connections but nonetheless a key gatekeeper in the network. The integration of statistics and visualization provided Hoppe with inspiration for a report delivered to the healthcare insurer.

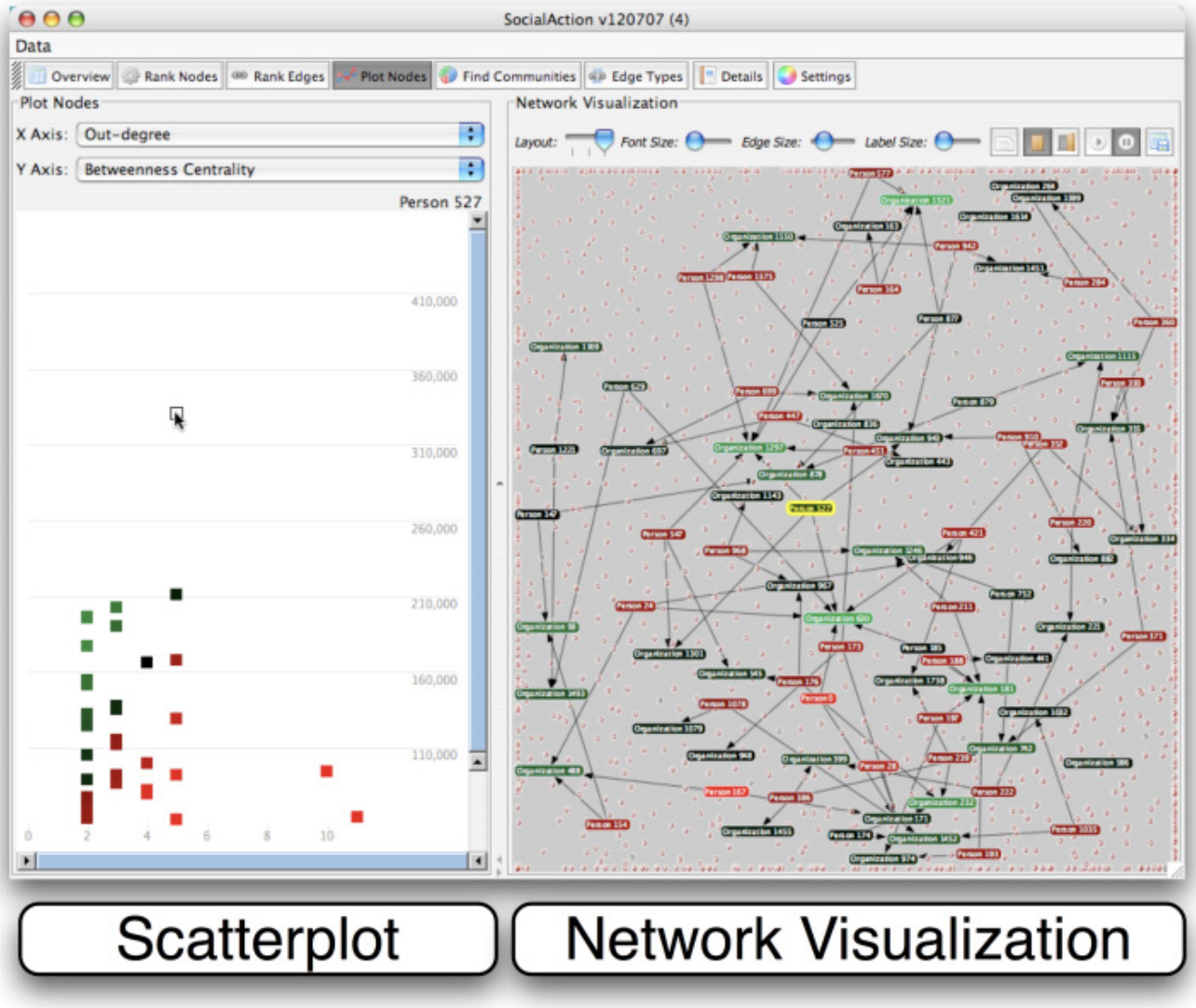


Figure 45. The scatterplot allows users to compare multiple SNA metrics (In this example, out-degree and betweenness) of the healthcare network (1740 nodes, 1854 edges). An obvious outlier is selected in the scatterplot, who was a trustee that had the highest betweenness (key gatekeeper) despite having only 4 connections. The names of trustees and organizations are anonymized to protect this confidentiality.

5.6.2 Mature Use

The healthcare insurer was informed by the report which Hoppe provided after his analysis with *SocialAction*. The network was filtered according to meaningful SNA metrics and had become more comprehensible. Now present were labels which allowed viewers to focus on the connections between hospitals and trustees. This transparency of the underlying data led the healthcare insurer to question the data. In fact, they found gaps in their data. These data discrepancies are being corrected, so Hoppe was forced to temporarily halt his exploration. However, the visualizations and filtering power of *SocialAction* allowed him to interpret these critically important data issues during analysis. Hoppe suspects a purely statistical approach to analysis might have missed these details.

5.6.3 Outcome

Hoppe used *SocialAction* as his main exploratory tool during his consulting work for the Northeastern healthcare insurer. In his monthly reports, he often included insights and visualizations resulting from his use of *SocialAction*. These findings have had significant impact with the healthcare organization. They now have a better understanding of the region's hospital trustee network and are working to make sure it informs their quality initiative. However, Hoppe admitted "I like having a medley of complex and ad-hoc tools. I am much more likely to recommend *SocialAction* to my clients – who need one simple approach to network exploration – than I am to adopt it as my own primary SNA tool." *SocialAction* lacked certain features critical

to his needs, such as additional statistical measures, comprehensive map-editing for nodes (e.g., size, label, and color), the ability to save these edits for future updating, and the ability to export the final results as vector graphics for high resolution presentations.

5.7 Case Study 4: Group Dynamics in Terrorist Networks

The National Consortium for the Study of Terrorism and Responses to Terror (START) is a U.S. Department of Homeland Security Center of Excellence. START has a research team around the world which "aims to provide timely guidance on how to disrupt terrorist networks, reduce the incidence of terrorism, and enhance the resilience of U.S. society in the face of the terrorist threat". One member of this team is James Hendrickson, a criminologist Ph.D. candidate, who is interested in analyzing the social networks of "Global Jihad".

Previous research has pointed to the importance of radicalization informing and sustaining terrorist organizations. While the radicalization process has been well described from a psychological standpoint, he believes theories examining the group dynamics of terrorism have largely failed to properly measure the size, scope and other dynamics of group relations. Hendrickson proposes to systematically compare the density and type of relationships held by members of the "Global Jihad" to evaluate their predictive ability in determining involvement in terrorist attacks. Marc Sageman, a visiting fellow at START, assembled a database of over 350 terrorists involved in jihad when researching his best-selling book, "Understanding Terror Networks" [74]. Hendrickson plans to update and formally apply social network analysis to this data as a part of his Ph.D. dissertation.

5.7.1 Early Use

The Sageman database has over 30 variables for each terrorist. Among these variables are different types of relationships, including friends, family members, and educational ties for religion. Hendrickson proposes that the types of relationships connecting two individuals will hugely affect their participation in terrorist attacks. Hendrickson began this analysis using UCINET [9] and was able to analyze some of his hypotheses. However, he believed UCINET did not facilitate exploration and generating new hypotheses easily. Hendrickson initially was skeptical of using visualizations for his analysis. He prefers being able to prove statistical significance quantitatively rather than relying on a human's judgment of an image. The quick access to the statistical counterparts of *SocialAction*'s visualizations eased his concerns.

In particular, *SocialAction*'s multiplexity feature aided Hendrickson's exploration. *SocialAction* allows users to analyze different relationship types without forcing users to load new data sets. The visualization shows the selected relationship edges but keeps node positions stable in order to aid comprehension. The statistical results are also automatically recomputed based on the newly selected structure. For instance, only the 'Friend' relationships among Jihadists are selected in Figure 46a. (Compare this to the denser Figure 17a, which shows all relationship types.) The nodes here are ranked by degree, so red nodes have the most friends. Jihadists Osama Bin Laden and Mohamed Atta (known for his role in the 9/11 attack) are ranked the highest. However, when the religious ties are invoked, a different set of key jihadists emerge (Figure 46b).

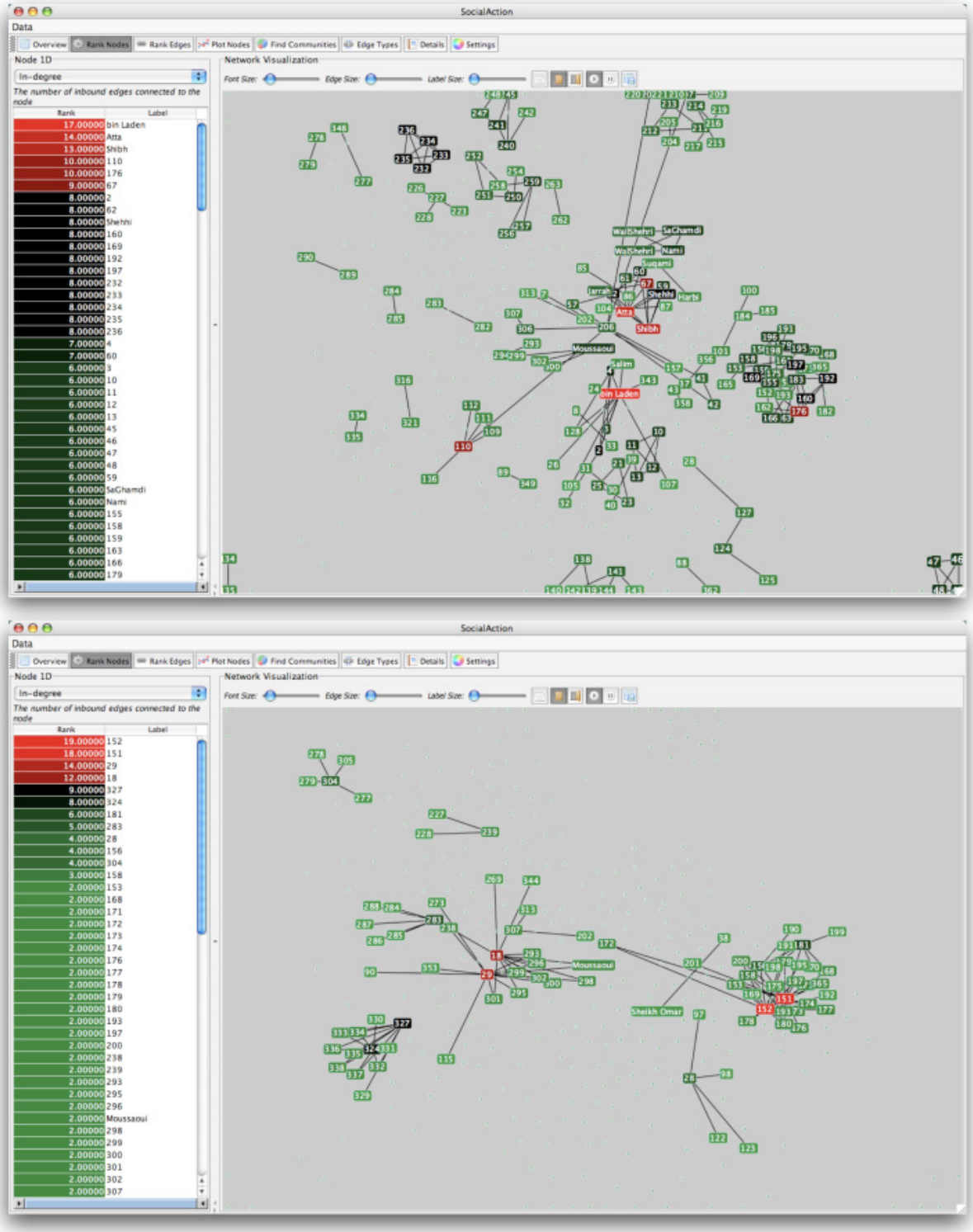


Figure 46. The multiplexity of the "Global Jihad" social network of 366 nodes is demonstrated. The upper visualization (a) shows the Friendship network (338 edges), with bin Laden the most popular individual. The bottom network (b), showing religious ties (106 edges), offers a much different view of the terrorist organization.

5.7.2 Mature Use

After analyzing the statistical attributes of nodes, Hendrickson became interested in understanding the individuals' attributes. As an example, he was interested in answering questions like, "Does an individual's socioeconomic status or education level impact their position in the terrorist network?" Of course, social network data does not allow users to infer causation but instead may show correlation. Like statistical rankings in *SocialAction*, users can rank and filter based upon attributes. Hendrickson filtered out individuals without college degrees, without religious backgrounds, or without engineering expertise and analyzed the results. The combination of nodal attributes with statistical filtering and plotting streamlined his accustomed workflow. He suspects he may not have been as free thinking if it wasn't for the ease of exploration in *SocialAction*. This analysis inspired Hendrickson to think of new, not-yet-coded attributes, to test additional hypotheses. He is currently augmenting Sageman's database with new attributes so he can look for patterns in *SocialAction*, visually and statistically.

5.7.3 Outcome

Hendrickson's experience with *SocialAction* has led to new inspiration for his dissertation thesis. Although he had access to the dataset long before the case study began and conducted analysis with other SNA software, the integration of statistics and visualization allowed exploration in new, interesting ways. As a result, the START center is interested in making *SocialAction* the default network analysis tool for internal and external users who wish to access their databases. They are currently

integrating a specialized version of *SocialAction* into their online global terrorism database.

5.8 From Information Visualization to Journalism: A Case Study with Slate Magazine

Due to the popularity of social networks in the mainstream, as well as the abilities of *SocialAction* to allow users to find and share social network discoveries more easily, there has been demand for *SocialAction* among journalists. Journalists can be quite different than academic researchers. They often have tight deadlines, which can result in limited patience for confusing interfaces, unclear statistical results, chaotic visualizations, or input file formats that are difficult to manually prepare. These problems are further started by Rich Gordon, a journalism professor at Northwestern, who states:

> "One key problem is that many journalists just aren't comfortable with technology. And even if they learn to use technology tools successfully in their work, few want to delve deeply into the process of developing new technology. And most media organizations don't seem to value their programming staffs or involve them in the journalism process. Instead, their work supports back-end systems like payroll and billing."
>
> *http://www.pbs.org/idealab/2008/02/computation-journalism.html*

However, the divide between journalism and computation is narrowing. A course on Computational Journalism has been taught at Georgia Tech, and a recent conference "Journalism 3G: The Future of Technology in the Field" brought together journalists, technology entrepreneurs, and researchers.

Information visualization tools seem to complement the ideals of computational journalism. Unlike data mining, which can sometimes take humans out of the process, Information Visualization tools engage users to analyze and find stories in the data. For this reason, I was curious in investigating the role my information visualization tool, *SocialAction*, among journalists. By taking into consideration the guidelines and findings I provide, a new user base of information visualization users can partake in data analysis. Journalists can be a valuable user base because they can promote our work but also promote the goals of information visualization: visual representations of complex data that highlights patterns, gaps, and outliers. Furthermore, journalists have mastered the art of storytelling, so their comments, feedback, and use of tools can influence information visualization designers to focus on features that lead to insights, discoveries, and compelling stories. In this chapter, I present evidence from a case study with journalists from Slate Magazine.

5.9 A Case Study with Slate Magazine

Slate is a popular, award-winning online news and culture magazine established in 1996. Although Slate does not have an organized, strategic initiative to better support computational journalism, this online magazine seemed to be an ideal home for disseminating information visualization results. For instance, the magazine features a section entitled, "Low concept: Dubious and Far-fetched ideas". Their online medium also facilitates interaction, a prime goal of information visualization tools.

This case studies centers around Chris Wilson, an editorial assistant at Slate, who initially used *SocialAction* while working at US News & World Report. The results from his experience with *SocialAction* were previously described and in [66].

5.9.1 Story #1: The Steroids Social Network

In December 2007, US Senator George Mitchell delivered a 409-page report on performance-enhancing drugs in the sport of baseball. The report described an underground market for steroids and human growth hormone involving many players and trainers throughout the sport. Reading the report, it became clear that drug use spread as players referred their friends and teammates, creating a social network of illicit activity in the majorly league baseball. Slate was interested in providing a visual overview of these connections defined in the Mitchell Report for their readers.

Wilson manually went through the Mitchell Report and tabulated the individuals and relationships mentioned using Microsoft Excel. Wilson then sent this spreadsheet to me, and I imported it into *SocialAction*. Since the network was rather small (58 nodes and 58 relationships), the network's structure became apparent through the force-directed algorithm alone. Kirk Radomski, a former batboy and clubhouse employee for the New York Mets, was at the center of the network. (Radomski cooperated with the federal commission and Mitchell Report and provided much of their evidence). Further discoveries were made when running the community algorithm on the network. Communities centered around trainer Brian McNamee (made famous by testifying against the all-star pitcher Roger Clemens), David Segui (a home-run hitter who played on many teams), trainer Todd Seyler, and

others. I sent a screenshot of these findings to the Slate team, and they immediately saw a story worthy of print (Figure 47).

At the time, *SocialAction* only had the ability to output rasterized graphics. Since they wanted the graphic to become interactive, they paid a graphic designer to trace the screenshot in Adobe Illustrator to make a vector graphic, as shown in Figure 48. This vector graphic was then delivered to a Flash programmer, who implemented the mouse-over features shown in Figure 49. This process took about 48 hours of iterative improvements and continuous contact between the writers, the graphic designer, the programmer, and the Slate editors.

Amidst this collaboration in Slate, I was slightly left out of the loop. This resulted in several complaints I had about the resulting visualization (e.g. there were more tangled edges and nodes than necessary). However, the graphic designer was able to accommodate most of my requests, making the paths between persons in the graphic easier to see.

The end result was a very successful visualization that was timely (released less than 1 week after the report) and popular (favorably talked about on one of the most popular sports blogs, Deadspin) . Even months later, the visualization was still often promoted in Slate Magazine (e.g. Figure 50). The interactive visualization is online at http://www.slate.com/id/2180392/.

The advantage of this joint approach was that I was able to leverage the input from the information visualization tool designer (myself), the reporter (Chris Wilson), the graphic designer (Holly Allen), the programmer (Matt Dodson), and the Slate editorial staff. The downside was that the original reporter and I didn't have as much

control over the final graphic was as one would have liked. It was also rather expensive and inefficient for both a graphic designer and programmer to replicate the output that *SocialAction* had automatically produced.

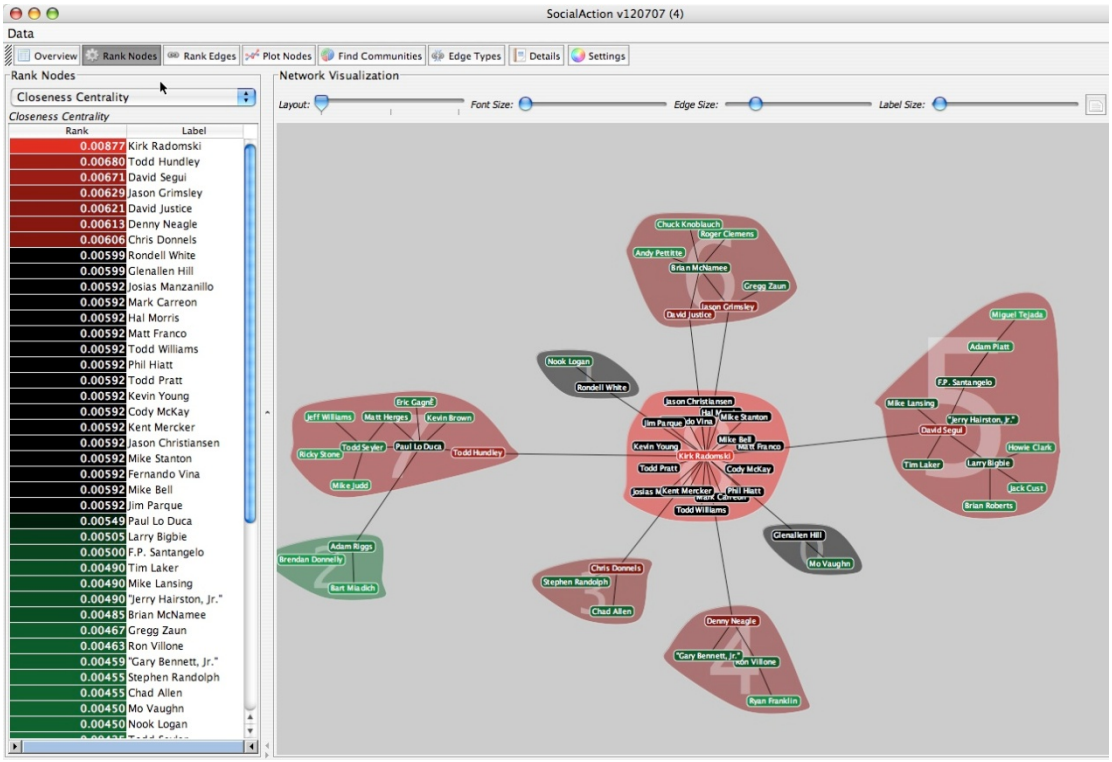


Figure 47. A screenshot of *SocialAction* after analysis of the Mitchell Report social network data. In this screenshot, the clusters were found using the community algorithm.

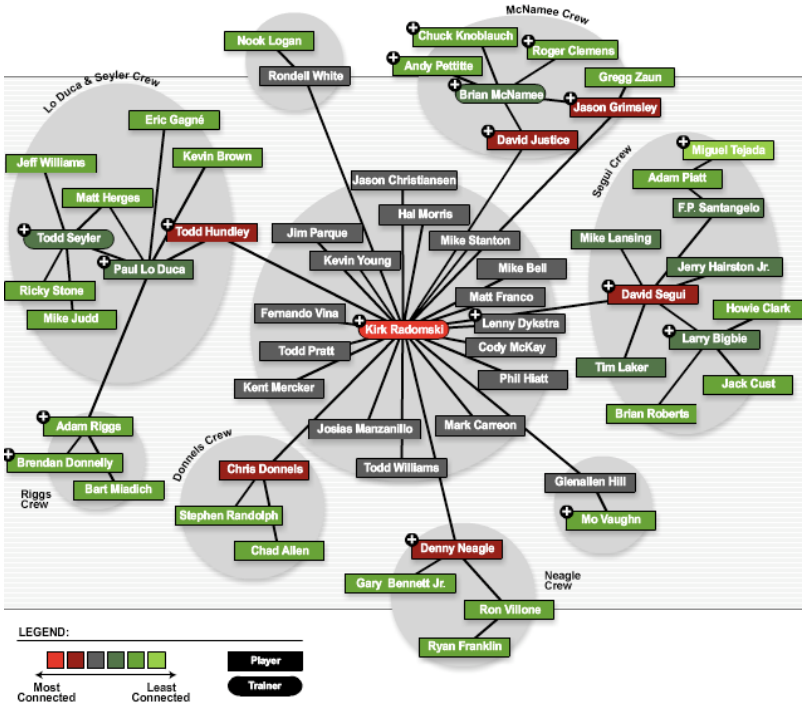


Figure 48. The Steroids Social Network interactive graphic on Slate. This graphic was manually created by a graphic designer and Flash programmer based upon the screenshot of *SocialAction* shown in the previous figure.

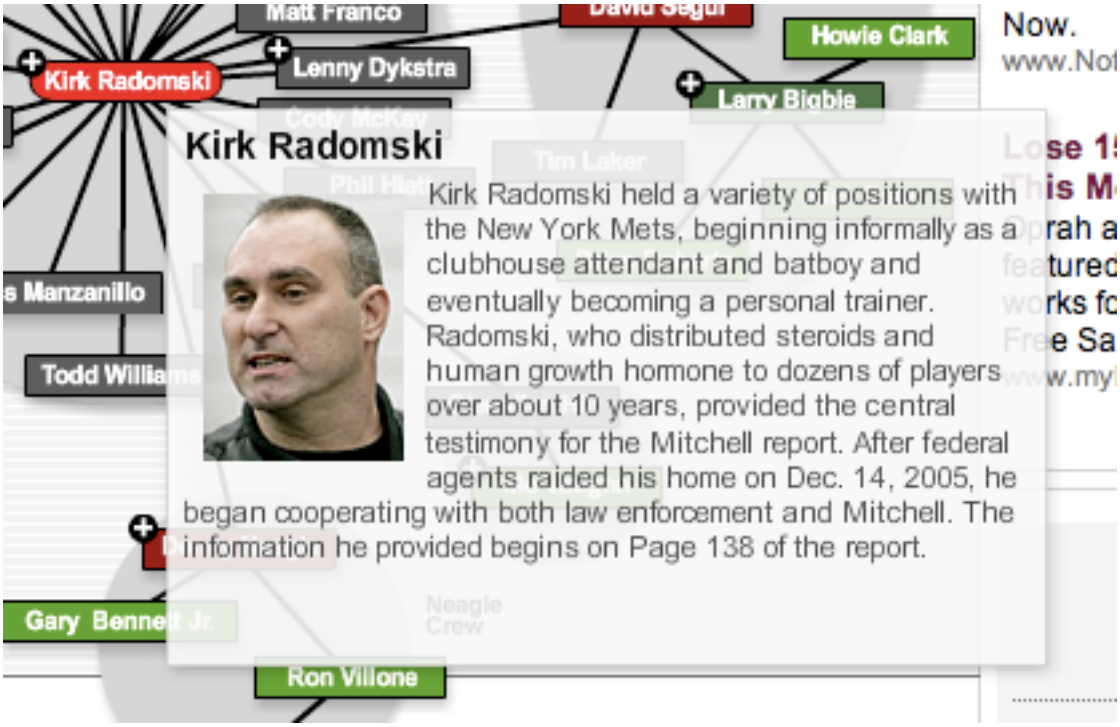


Figure 49. The Steroids Social Network graphics's interactive features. When a user mouse-overs a person or relationship, the applet describes the information in the Mitchell Report about that person or relationship.

Figure 50. Promotional Advertisements on Slate Magazine's website, which were featured prominently December 2007 through February 2008.

5.9.2 Story #2: The Oscars Social Network

The Slate Editorial Staff was interested in producing an interactive visualization to support their Oscar coverage. Chris Wilson and I wondered if a story could be told surrounding the collaboration of the nominated actors. Using public data sources, Wilson created a collaboration network of movie stars based upon the movies they've acted in, and their fellow co-stars. By crawling out from the nominated actors, a social network was constructed of all of the nominated actors and their co-stars. This resulted in a social network of over 4,000 actors and 25,000 relationships. This may seem like a huge, diverse talent pool -- but upon closer inspection, it was a rather small world for these elite actors. Using social network analysis analysis algorithms to find the gatekeepers, it turns out there were four prominent non-nominated actors in the network to connect all of the nominees: Catherine Keener, Keira Knightley, Steve Buschemi, John Turturro.

While the editors were intrigued by this "small world" graphic, Wilson was not able to convince them that it had a clear message that could easily be distilled for readers not steeped in SNA theory. This can be a tension in these collaborations between journalists and academics. Wilson adds that the first question an editor is likely to ask is: "So what? What to readers learn from this? What specific question does it answer that people may have wondered in the past?"

The editors, however, strongly supported the concept of the graphics and are eager to incorporate them in Slate's coverage when the right opportunity arises. Their argument, as Wilson describes, is that since the public is not usually familiar with the

particulars of social networks, the first data-driven implementation should present a clear image of a data set with which people are particularly familiar. One problem with the Oscar data this year, from this perspective, was that it included a lot of unfamiliar names from newcomers and foreign actors. Furthermore, one year's worth of nominees struck some as a bit arbitrary.

The editors searched this graphic to find if a reader could use it to answer a juicy question. For example, could this data identify people who had never won (or never been nominated) but had a close connectivity to many winners/nominees? This question was pegged to a larger data set of winners over a least a decade, and occurred too late in the week for it to be plausible to build this exponentially larger database.

Wilson mentioned this was an interesting experience, as the mere existence of non-intuitive communities within a network doesn't qualify as a story. Rather, it has to reveal something that fits logically into a storyline or (as was the case with the steroids network) provides an insightful way to look at a group that people already fundamentally think of as a social network. While this was true of the Mitchell Report, actors and actresses in movies are equally as logical as candidates for a network.

Wilson adds there was one other distinction between the two stories: "The Mitchell Report was a bounded network of about 60 people, whereas the movie data could theoretically have continued expanded to tens of thousands of names. There was no glaringly obvious group of people who qualified for the network based on some other distinction than connectedness. This is one reason the Senate graphic is

such a good example of the technology: There is a discrete group of 100 nodes." Wilson and his editors are confident that *SocialAction* "will be an engaging editorial tool". However, the artificial construction of the actor's social network didn't quite fit with the editor's preference for their Oscar coverage.

Figure 51. The Oscar Nominee Social Network after filtering out all but the essential gatekeepers.

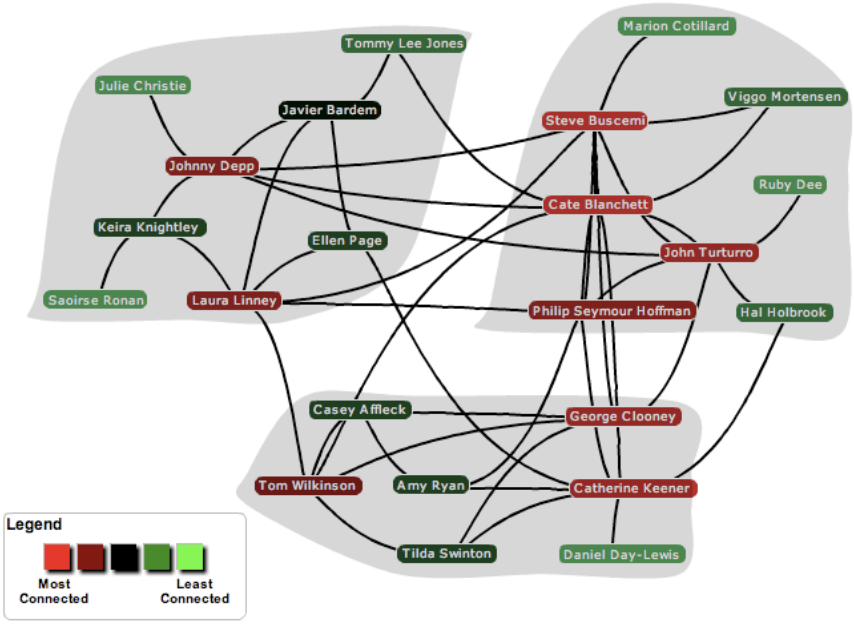


Figure 52. The corresponding Flash applet , automatically generated by *SocialAction*, for distribution in Slate Magazine.

5.9.3 Summary of Slate Case Study

In summary, this case study with Slate Magazine illustrates a journalist's perspective into information visualization, and how a tool can be used for both data analysis and a story of interest to the general public. In addition to these insights, tool designers should also keep in mind the requirements of accessibility for journalists, including:

- Ability to input data from standard programs, such as Microsoft Excel. Complex data formats like XML are difficult for non-programmers to provide.
- Ability to export data formats.
- Print quality graphics.
- Cross-Platform applets.
- No special software to be installed by end-users to experience the visualization.

Journalism is a media outlet that can help gain information visualization more prominence among readers, researchers, analysts and scientists. However, journalists may be different than typical data analysts due to their short deadlines and requirement for thrilling discoveries.

5.10 Summary

This chapter describes how my dissertation provides a novel evaluation methodology that improves the ability to measure the creative activities of information visualization users. Traditional laboratory controlled experiments are often not suited for the tasks of information visualization users, whose tasks are exploratory and not well known, and users have unique domain expertise. I propose the use of long-term case studies with domain experts to validate if a system is successful or not. My 5

step methodology lasts 4-8 weeks with the system deployed to domain expert users. The first step, the interview, gauges the domain experts intentions, where are used as a reference to help validate the tool success. After a subsequent training session, the early use phase begins where domain experts use the tool for 2-4 weeks with the ability to request missing features. This stage is followed by a mature use phase, where domain experts continue to use the tool for 2-4 more weeks. In both the early and mature use phase, interviews are conducted on-site where the domain experts report on their experience and insights. The final stage is an outcome session, where users debrief their experience and revisit their intentions from the first stage.

This methodology was applied four times for case studies with unique users with varying domain expertise. A political analyst, bibliometrician, healthcare consultant, and counter-terrorism researcher were all able to make previously unknown discoveries using *SocialAction*, despite having prior studied the data with other tools. These unique insights help validate that *SocialAction*'s integration of statistics and visualization helps better support the workflow and creative activities of social network analysts. A follow-up case study, with Slate magazine, is also included to further illustrate how information visualization tools deployed over a long time can aid the creative activities of journalists.

Chapter 6: Implementation

I have developed *SocialAction* since 2005. *SocialAction* 1.0 was a prototype demonstrated coordinated views and attribute rankings in my InfoVis 2006 paper [65]. *SocialAction* 2.0 was a significant rewrite of *SocialAction* 1.0, where the statistical and visualization algorithms have been optimized to support real-time interaction with large networks of interest to my case study partners described in the my CHI 2008 paper and Chapter 5 [66]. *SocialAction* 3.0 added the Systematic Yet Flexible guides for exploration described in my IUI 2008 paper and in Chapter 4 [67].

SocialAction is implemented in Java to provide an OS-independent social network analysis tool for social network analysts. *SocialAction* uses the Prefuse [39] visualization toolkit to render the visualizations in Java2D. This chapter describes the implementation details of *SocialAction* 3.0 without the systematic yet flexible guiding to simplify the description.

The resulting *SocialAction* application is over 15,000 lines of code, in addition to contributions to various open-source libraries to improve the rich interaction and visualization capabilities necessary for its design.

6.1 User Interface of SocialAction

Since the design of *SocialAction*'s user interface is a primary contribution of this dissertation, much of the user interface discussion is found in Chapter 3. This chapter only provides details relevant to the implementation. The overall user interface layout of *SocialAction* is shown in Figure 53. The ToolBar features a button for each of the tasks relevant to social network analysts described in Chapter 3. When one of these buttons is clicked, both the Statistical Panel view and the Network Visualization

view are updated. Throughout all user interactions, both the Statistical Panel and the Network Visualization view are coordinated. This achieves my design contribution of integrating statistics with visualization throughout analysis. A common challenge with multiple views of data is they each view competes for screen real estate. Users have the ability to emphasize a particular view by enlarging the panel of interest (which results in the shrinking of the panel of less interest). This action can be done by dragging the divider between the two panels to resize both views. There are also arrows on the divider that can be selected to completely hide one view.

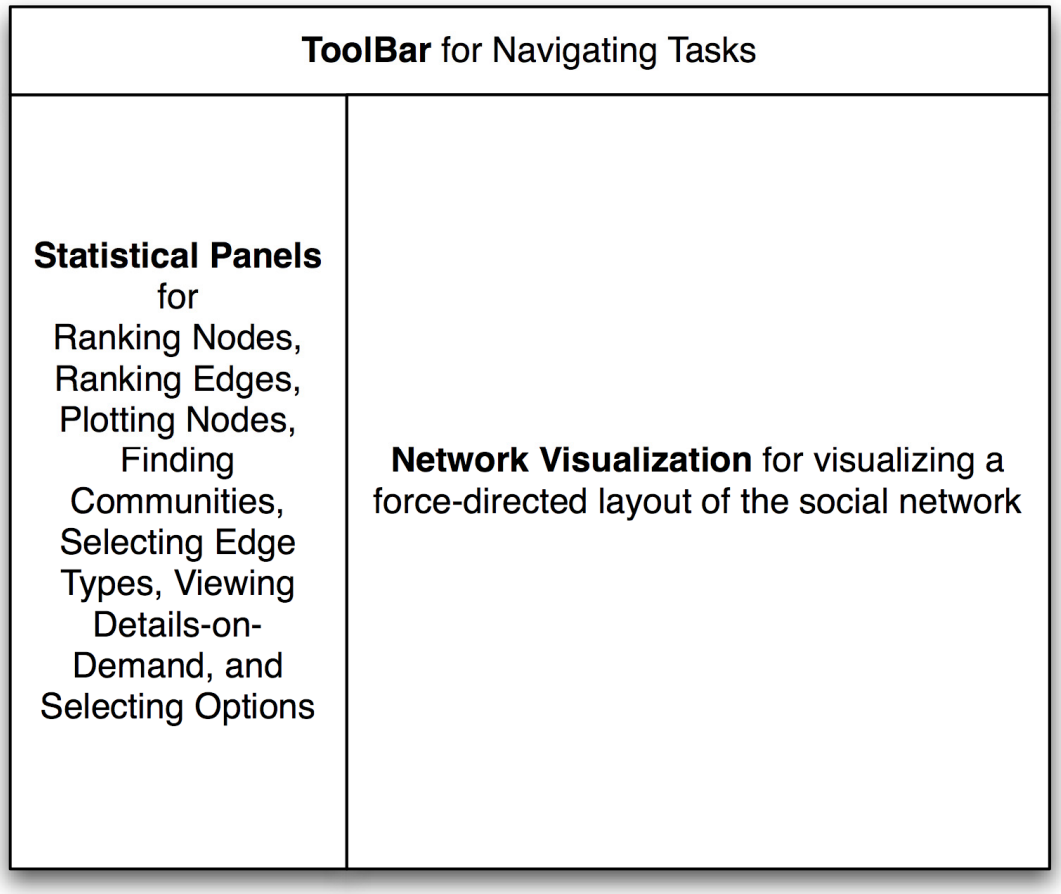


Figure 53. An overview of the user interface of *SocialAction*.

To achieve this coordination, *SocialAction* stores the social network data into the *Prefuse* Reference Model [37]. This architecture is appropriate for an information visualization system because it separates the data and visual models to allow multiple visualizations of the same data source. Multiple views of visualizations are also possible due to separate visual models. And finally, separate controllers allow user input to flexibly update any component of the system. This Reference Model framework has been advocated by both Chi et al.'s *data state model* [16] and Card et al.'s *infovis reference model* [13]. This model allows each of the visualizations in *SocialAction* to be coordinated with each other.

The social network data is stored in four main tables. A NetworkTable is maintained for overall attributes of the social network, such as the number of nodes and edges or if it's directed, as well as the computed statistical attributes, such as density and diameter. Each row in this table in an attribute describing the network. A NodeTable is maintained to store data about each of the nodes. Each row in this table is a node, and each column represents an attribute of the node. Some columns are attributes (e.g. Label and Image information, as well as any attributes inherent to the data). When a new statistical measure on the nodes has been computed, a new column is added to this table to store the newly computed attribute. The EdgeTable is similar to the NodeTable, except each row represents an edge. An edge is required to have two columns, the first referencing the source node and the second reference the target node. Each additional column reflects an inherent or computed attribute. Finally, there is also a CommunityTable that maintains information about each of the communities. Rows represent each of the communities, and the columns represent

their computed attributes. The NodeTable, EdgeTable, and CommunityTable each have a corresponding VisualTable that represents their current visual representation, as shown in Figure 54. Here, size, color, position, and other visual characteristics corresponding to the nodes, edges, and communities are stored. The statistical and network visualizations use this information, and allow both the data and visual attributes to be coordinated.

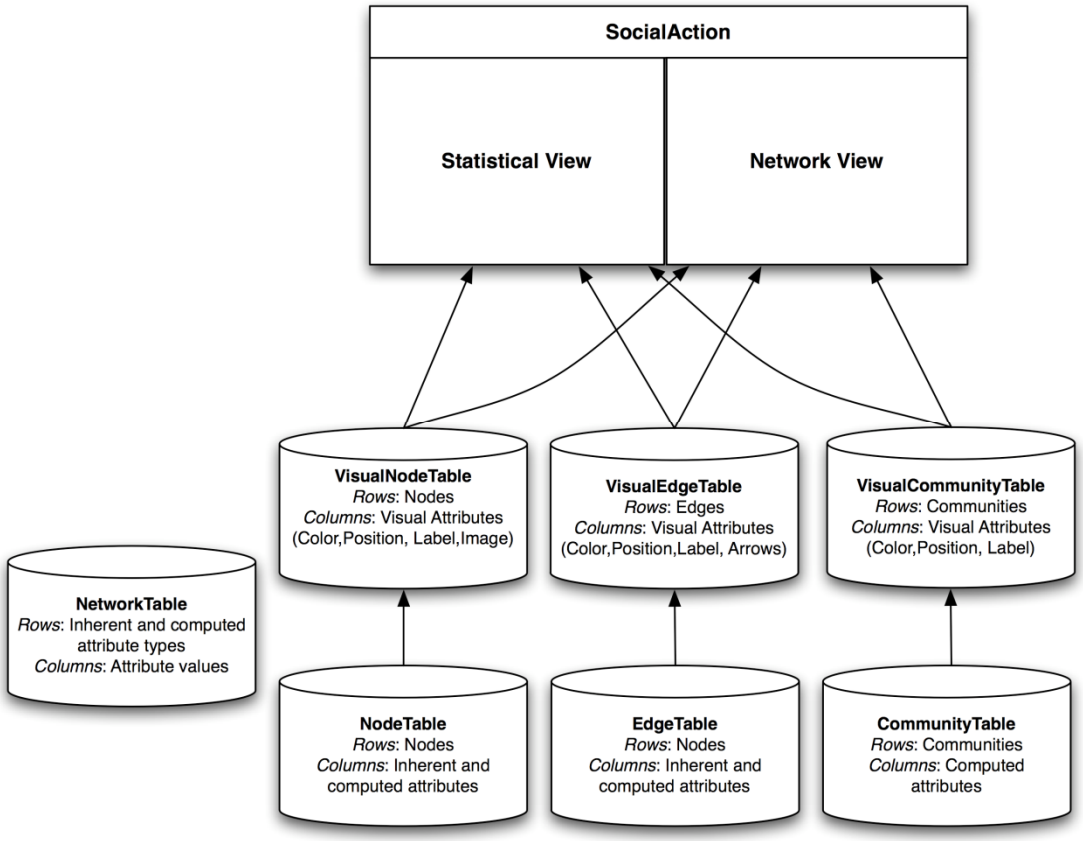


Figure 54. The data reference model of *SocialAction*.

6.2 Input Data Formats

SocialAction supports a variety of data formats. Users can import social networks in a variety of popular formats among social network analysts, such as Pajek [17] and GraphML. However, these formats can be difficult for non-programmers to produce, as they are structured formats that cannot easily be edited in a spreadsheet program like MS Excel. For this reason, SocialAction also supports the HCIL Network Visualization Input Data Format (http://www.cs.umd.edu/hcil/nvss/netFormat.shtml). The format requires two separate files: a node table and an edge table. Similar to the internal representation described above, the node/edge tables consists of a row for each node/edge. Each column represents an inherent attribute of the node/edge that can be accessed from within *SocialAction*. For instance, users can choose to color, label, or rank according to these attributes present during the import process.

After users select the input files, *SocialAction* provides a preview dialog window, as shown in Figure 55. Users can construct the network based upon the attributes of the nodes and edges they've imported. Users can select the column in the node table that represents the ID of each node. The ComboBox is automatically populated with the names of each column in the node table. Similarly, users must select the two edge columns. Users can also select the initial columns they want to reflect the labels of each node, the multiplex attribute for varying edge types, and the edge column that stores in each edge. Furthermore, users can check the appropriate box if the edges are bi-directional or directed. After users click the "Import" button, the network is constructed and loaded into the statistical and visualization views.

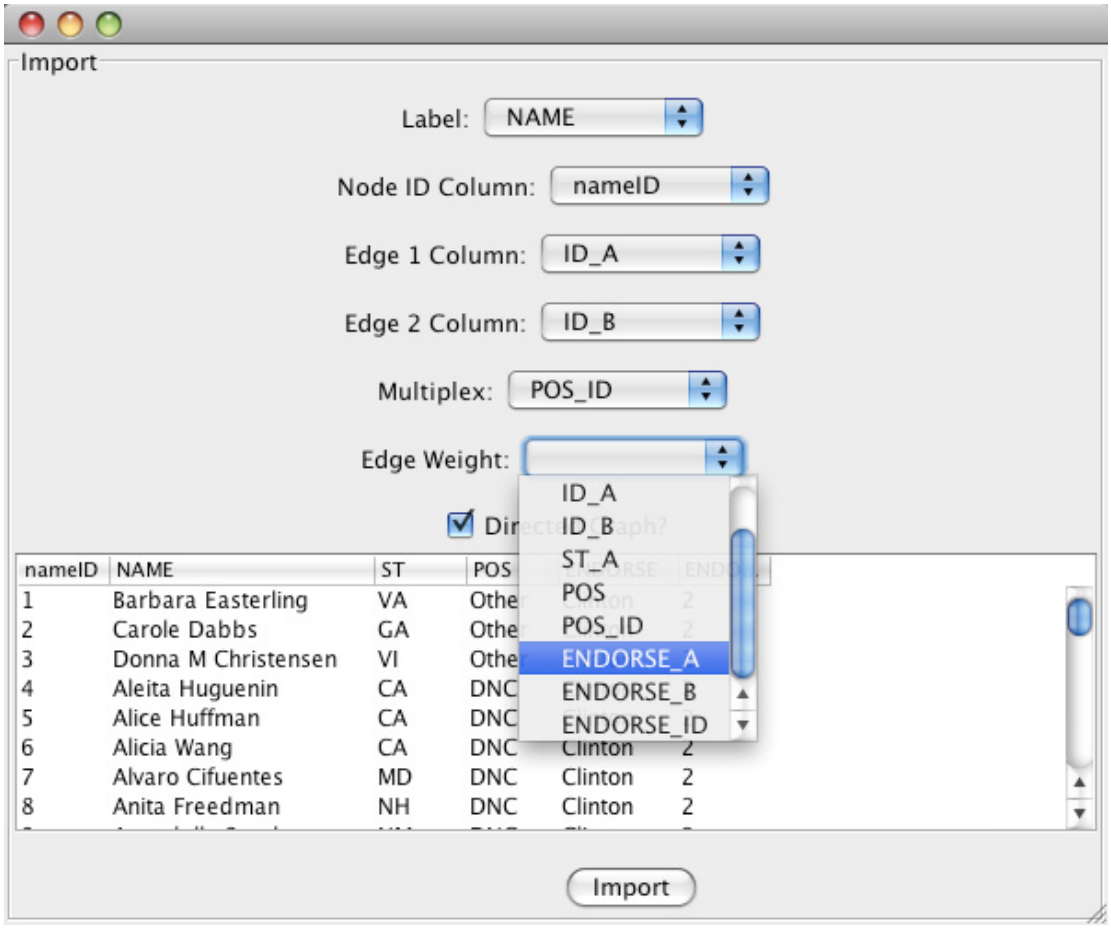


Figure 55. The import Dialog Box for *SocialAction*

6.3 Algorithms and Scalability

Due to the tight integration of statistics and visualization, the speed of the algorithms driving the statistically computed attributes is very important. The algorithms implemented in *SocialAction* were often based upon known fast optimizations (e.g. [10]) or derived from existing fast implementations (e.g. [63]). For certain algorithms not available, they were personally implemented. In all cases, they were optimized to interact with the *SocialAction* data reference model previously described.

To illustrate the runtime of *SocialAction* and the computational complexity of the statistical algorithms, a table of run-time speed is provided for a sample of algorithms. The length and time of each algorithm is dependent on the topology of the network, such as the number of nodes, edges and the diameter. Run-time speeds are shown for each of the networks used in the case studies to illustrate the speeds in public. These speeds with computed on a Microsoft Windows XP PC with an Intel Core2 Duo 3.00 GHz processor, with 3.25 GB RAM.

Network Attributes	Case Study 2	Case Study 1	Case Study 3	Case Study 4
# of Nodes	220	98	1740	366
# of Edges	305	2444	1854	2334
Diameter	11	3	24	9
Algorithm	**Time (Milliseconds)**			
Degree Centrality [99].	16	16	46	16
Betweenness Centrality [99]. *Derived from implementation in [63].*	125	204	2265	627
Closeness Centrality [99].	110	392	5016	895
Farness Centrality [99].	109	189	4594	831
Clustering Coefficient [103]. *Derived from implementation in [63].*	16	439	62	110
HITS Authority [52]. *Derived from implementation in [63].*	109	267	672	345
HITS Hub [52]. *Derived from implementation in [63].*	110	219	641	345
Diameter [99].	125	189	6312	910
Fast Community Algorithm [61]. *Derived from implementation in [39].*	47	78	1031	110

Table 2. Run-times for a set of statistical algorithms in *SocialAction*.

In addition to algorithms for the computed attributes, the visualization also relies on complex algorithms for producing a layout of the nodes. The strategy employed by *SocialAction* is a force-directed algorithm, which acts as a physics simulation. Nodes repel each other and edges act as springs bringing connected nodes together. The following pseudo-code simplifies this behavior:

```
initialize node velocities to 0
initialize node positions to random
loop
    for each node
        node-forces := 0   // sum of total forces on this node
        for each other_node
            node-forces:= node-forces + repulsion( node, other_node )
        for each edge connected to this node
            node-forces:= node-forces + attraction( node, edge )
        node.velocity := node.velocity * node-forces
        node.position := node.position * node.velocity
until user intervention
```

The force directed layout's implementation has a running time of MAX(*O(*N log N*), O(*E*)*), where N is the number of nodes and E is the number of edges. This running time increases when additional forces are added, such as when new forces are introduced after the community algorithm to produce separation between communities. The implementation of this algorithm is derived from [39], which uses

the Barnes-Hut algorithm for efficient force simulations [6]. However, *SocialAction* extends the force simulator with customizations to better support social network analysts, as described in Chapter 3.

Force-directed algorithms generally lack a clear definition of convergence. Consistent with the design to empower users with these critical decisions, the algorithm runs until users decide when the layout is good enough. The algorithm updates the visualization in a fluid and dynamic way by animating the nodes until users. This process is illustrated in Figure 56 for each of the case study data sets. For each case study, the left-most image was rendered 0.1 seconds after the data was loaded. The right-most image is after users decided the network layout was good enough.

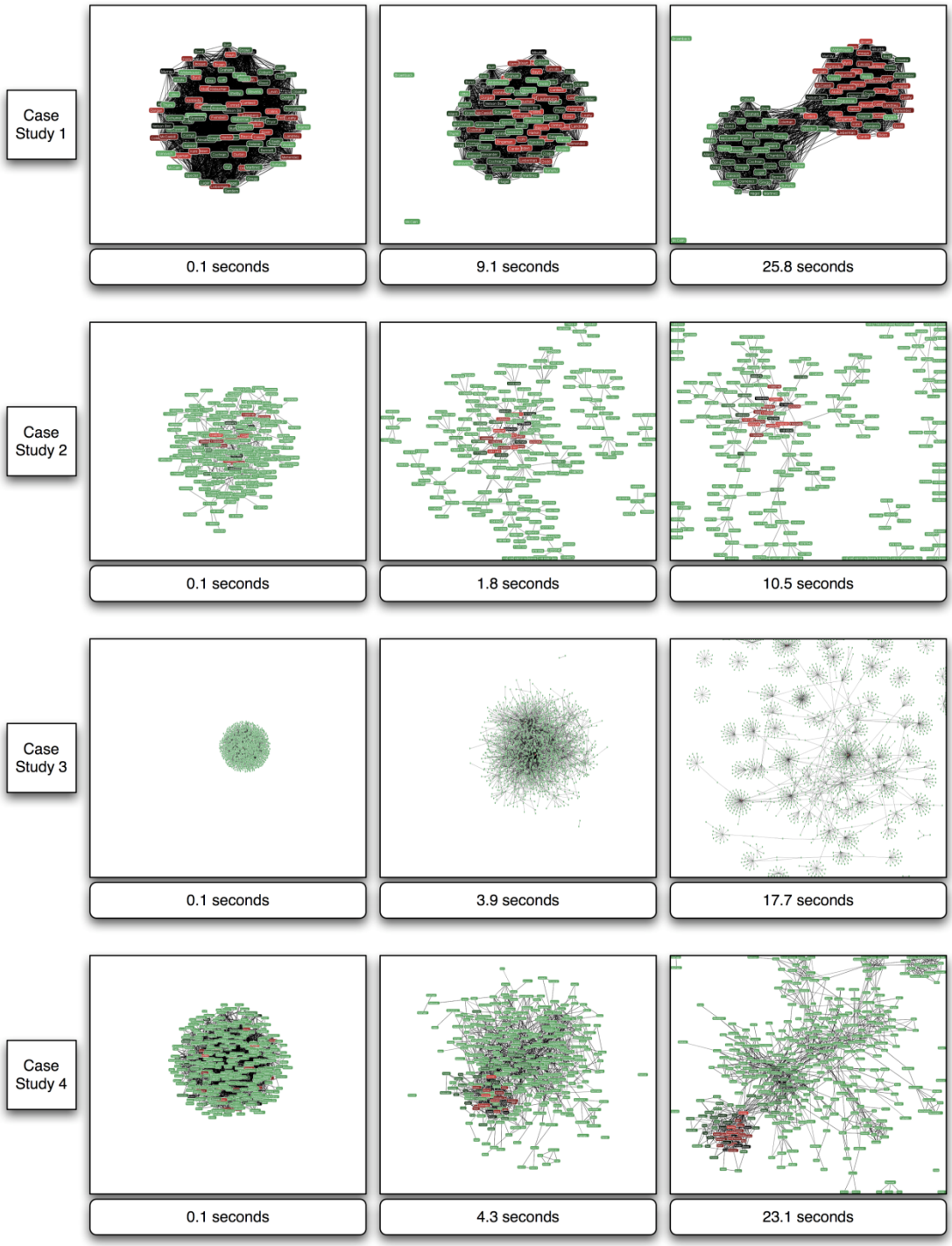
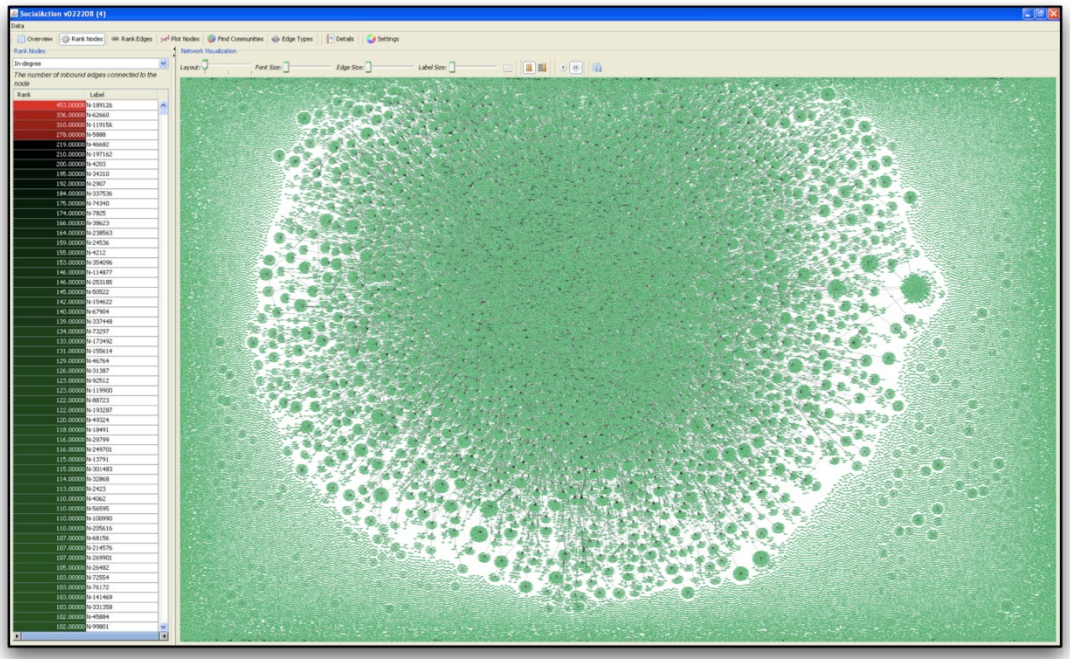


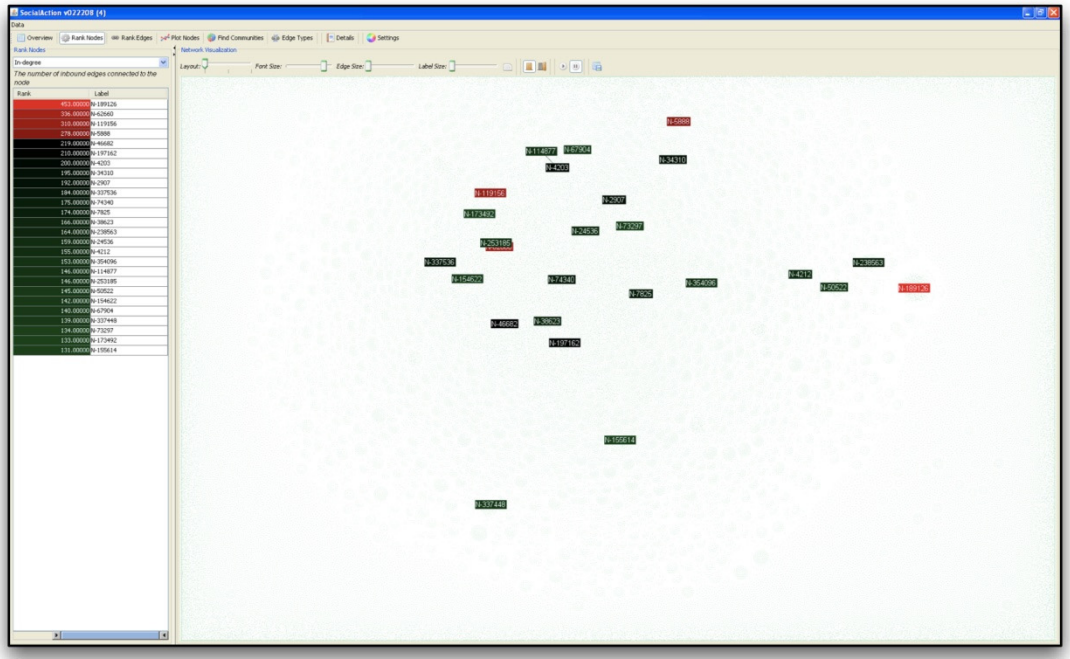
Figure 56. An illustration of the run-time of the force-directed algorithm on 4 case study data sets.

In each of the case studies, the statistical algorithms for computed attributes in *SocialAction* were able to run without noticeable lag. Some of the more complex algorithms, as illustrated in Table 2, induce a small delay. The complexity of networks will certainly always result in certain meaningful algorithms with large runtimes. For this reason, *SocialAction* was designed to be an interactive environment even when algorithms are running. The statistical and visual processes run on separate threads, allowing the user to examine the visualization while statistical attributes are being computed. Conversely, users can browse statistical attributes if the force-directed layout is still untangling. Of course, the greatest discoveries may rely on synergy between the statistical and visual components. By for situations when it is impossible to do so due to algorithmic complexity issues, users are still able to explore the data that is currently available without being forced to stare at a frozen interface.

Although certain algorithms have lengthy run times with large networks, the *SocialAction* infrastructure is robust. The robustness is illustrated in Figure 57, which demonstrates the scalable capabilities of *SocialAction*. In this image, over 152,000 nodes are connected by 148,000 edges, representing a real and complete Customer Relationship Management database of donors to non-profit organizations. The interactions and rendering are slow, but still work, as the filtering interaction demonstrates in the bottom image of Figure 57.



152k nodes and 148k edges loaded in *SocialAction.*



The network above filtered to the top 26 popular nodes.

Figure 57. An illustration of the scalable infrastructure of *SocialAction*.

Since networks of this size and larger have not been a primary focus of *SocialAction*, there are still many improvements to achieve better performance with larger networks. Implementing faster statistical algorithms that approximate importance metrics may provide users with effective enough strategies for exploration. Furthermore, off-loading algorithms to the GPU [26]or MapReduce [18] may also be possible.

6.4 Summary

The implementation of the *SocialAction* to achieve the integration of statistics and visualization, as well as the requirements of social network analysts, was a significant computer science challenge. The architecture and user interface of *SocialAction* required sophisticated data models. Furthermore, the algorithms necessary for statistical analysis had to be optimized to run in real-time, to better support the creative activities of analysts. Scalability challenges also needed to be met to make sure the system worked on real data of value to domain experts. The design and implementation of *SocialAction* was a significant effort, which results in a powerful tool that will live on for social network analysts to help them make sense of their data.

Chapter 7: Conclusions and Future Work

This dissertation focuses on three major contributions of my doctoral research. These contributions were outlined in Chapter 1. I revisit these contributions here and reflect on their impact to interactive, visual analysis of complex data.

7.1 Contribution (C1:Integration)

> *Provides an integration of statistics and visualization to improve exploratory social network analysis.*

Social network analysis is a complex process. Not only do analysts have to understand the inherent attributes of each individual in a network, but also the relationships of between individuals. Both visual and statistical approaches have great value, but an integrated solution aids the exploratory and creative tasks of analysts. By organizing the many features of social network analysis into goal-centric tasks, users may gain an overview of networks, find important nodes and edges by ranking and filtering, find patterns and outliers by plotting nodes in a scattergram, find communities by enabling clustering algorithms, and understand different types of relationships by isolating multiplex edge types. These tasks are all part of the design of *SocialAction*, which coordinates the statistical and visual information necessary to make sense of the social networks. The system is highly interactive, so users can simplify the lengthy tabular output of statistical results or crowded network visualizations. *SocialAction* allows users to dig deeper into their networks, guided by statistical algorithms. This also improves the scientific workflow for complex analysis, as users no longer have to navigate back and forth between statistical and visualization packages. Furthermore, although *SocialAction*

is designed to support social network analysis, it also allows users to explore and interpret non-social networks.

7.1.1 Reflections on Contribution C1:Integration

Much of the design of *SocialAction* has focused on visualizing and interacting with *computed* attributes that use statistical algorithms to measure the network. This is due to the need to understand the social position and structure of the nodes and edges. However, less attention has focused on the *inherent* attributes. More dynamic queries and filtering based on inherent attributes is an obvious next step, but an even more challenging approach would be to support both concurrently. As an example, filtering based on inherent attributes of nodes should update the related computed attributes automatically. If an analyst was looking for the gatekeepers in a network, and decided to exclude a gender, the statistical output should update automatically. Similarly, when filtering according to statistically computed attributes, analysts should be given statistical summaries on the inherent properties of the remaining nodes. This may lead to further hypotheses among inherent attributes that may not have been thought to be important.

SocialAction's design has mainly focused on static networks that do not change over time. Although its multiplex features supported binning of time, as demonstrated earlier in this dissertation, dynamic network analysis requires even more sophisticated support. Networks that evolve, grow, and change over time are of growing interest to analysts. Although many dynamic network analysis techniques are still immature, the ability to support these tasks is becoming critical for domains like fraud detection and intelligence analysis.

Furthermore, a complex set of engineering challenges remain. Although scalability has been achieved for the initial target group of *SocialAction* users, there are many social network analysts who wish to study networks with millions or even billions of nodes. The algorithms generating the statistical analysis and network visualizations require optimization in order to support networks of these sizes. Faster algorithms that can approximate the structure of networks may also need to be designed, as the running time of some existing algorithms in their current form do not scale well. Another strategy might be to offload computationally expensive operations to the GPU or cloud computing, if the algorithms can be parallelized. Reducing the visual clutter of large networks is also still a challenge. *SocialAction* already supports node aggregation and filtering to this end, but additional solutions may be necessary. With proper engineering, the design goals presented in the next contribution will continue to aid the exploratory process of analysts studying social networks.

7.2 Contribution (C2:Guidelines)

Provides design guidelines for interactive techniques to improve exploratory data analysis with computed attributes and systematic yet flexible guides.

The implementation of *SocialAction* has demonstrated that the integration of statistics and visualizations can improve exploratory data analysis, as it supports the explorative and creative tasks of analysts. Interactive techniques are a key part of the design to make both the statistical and visual components comprehensible. Without interactions such as ranking and filtering, statistical output and network visualizations may be too complex to interpret. In most information visualization systems, these

interactive techniques often focus on inherent attributes, rather than computed attributes from statistical techniques or data mining. In order to make the lessons learned from *SocialAction* more applicable to a broad range of researchers and designers, I have proposed a set of design guidelines. Using an established taxonomy of seven categories of infovis interaction techniques, I provide a thorough description of how the technique can be enhanced with computed attributes.

Another challenge is that complex and large data sets like social networks typically require numerous interactions in order to find patterns, outliers, and insights. The many resulting interactions can lead to complicated paths of exploration. To assist discovery, I provide *systematic yet flexible* (SYF) design goals to help guide domain experts through analysis over days, weeks and months. SYF offers *systematic* guides that provide users the ability explore relevant analytical features. However, SYF also supports *flexible* diversions to pursue insights while still maintaining overall progress. To assist analysis, SYF provides annotation, collaboration and reuse capabilities. These three tasks offer analysts a way to record, share, and more easily find new insights. After all, data analysis is all about finding the useful nuggets. SYF still relies on human analysts to find these nuggets, but empowers them by maintaining their history, measuring their progress, and most importantly, keeping them informed.

7.2.1 Reflections on C2:Guidelines

These design goals highlight how researchers and practitioners can integrate statistics and visualization to better improve exploratory data analysis of social networks. However, integrating statistics and visualization goes beyond social networks –

statistical algorithms and data mining results can aid in analyzing temporal, hierarchical, and multi-dimensional data. The design goals and examples can extend to such other complex data types as those above - to emphasizing the power of an integrated approach.

Similarly, the *systematic yet flexible* design outline the high level goals of guided discovery, but additional techniques might be required for it to achieve adoption by both system designers and users. Providing a guide through all actions ensures a systematically complex exploration, but as the number of states and interactions grows, a complete exploration may be impractical. One possible strategy is to use the statistical methods and data mining techniques to emphasize certain states that have interesting patterns. A second possibility is to build a recommendation system on top of the systematic yet flexible architecture. If analysts follow similar paths of exploration, recommending future steps may be appropriate. These recommendations can help novices as well as experts who have developed systematic and repeatable strategies for analysis.

Systematic yet flexible support has implications beyond data analysis tools. Wizards and tabs are pervasive in the user interfaces of many applications. SYF combines the *systematic* properties of wizards with the *flexible* properties of tabs, while providing users feedback about progress. For any interface that requires steps to be completed, and where order of completion is not restricted, I believe the SYF interface would improve the user experience.

To date, the SYF design has only been integrated into *SocialAction*. However, since the SYF system is designed as a modular component, it is possible to

integrate the system into other data analysis tools as well. Several tool designers that were given a preview of the SYF system immediately noted the benefits it would offer to their users. In addition to providing guides, developers would obtain critical features that users demand for free, such as history keeping and supporting "undo".

There is also future work to be done in advancing the collaborative functionalities. Although users can take turns and share their exploration, *SocialAction* provides no way to merge them if they are concurrent. The collaborative requirements of small groups (2-10) and larger teams (10-100) of researchers need to be studied further.

Expert users might also wish to rearrange or design their own steps for social network analysis. Currently, step design is left up to the developer using the API. However, since most expert users are end users and not developers, it makes sense to afford them this capability as well. This feature would also be useful in allowing users to compose smaller steps for more specific tasks. If analysts are only interested in a small subset of measurements, having a way to measure progress based on those goals instead of the overall features is important. For these reasons, a systematic customization feature for experts seems necessary.

7.3 Contribution (C3:Evaluation)

Demonstrates the effectiveness of long term case studies with domain experts to measure creative activities of information visualization users.

In order to study the effects of the integration of statistics and visualization on exploratory data analysis, novel evaluation methods are necessary. Traditional

laboratory-based controlled experiments have proven to be effective in many user interface research projects. However, because domain experts work for days and weeks to carry out exploratory data analysis, their typical workflow is nearly impossible to recreate in a laboratory-based controlled experiment. Furthermore, exploratory tasks are poorly defined, so telling the users which tasks to perform is incompatible with discovery. For these and other reasons, I chose to design an evaluation methodology that used structured and replicated case studies. The resulting 5-step case study design was implemented on four unique domain experts with unique data and research questions.

The four case studies conducted provide evidence that exploratory data analysis improves with integrated statistics and visualization. Tools to support the generation of hypotheses are sometimes overlooked. *SocialAction* provides users with the freedom to load all of their data to identify global trends. Instead of removing data blindly, users can filter the data according to statistical principles of social network analysis. This provided the domain experts with a level of comfort they lacked in using other tools.

In addition to providing evidence to support my hypothesis, the case studies also served as a stimulus for pushing the technology's development forward. The implementation was not driven for a controlled study, but rather to handle a wide range of use by inquisitive researchers. It forced the implementation to operate on real, large datasets. Subsequently, *SocialAction* has matured into a tool that can be used by numerous professional researchers to solve a wide range of research problems.

7.3.1 Reflections on C3:Evaluation

Long-term case studies with domain experts clearly show that *SocialAction* led to insights and discoveries previously unknown to their users. These creative discoveries might have been lost or undermined in traditional experiments with summary statistics. However, the case studies do not show quite as clearly the extent to which *SocialAction* was responsible for the discoveries. The evaluation method relied on weekly interviews, in which the domain experts would share their tales of success or frustration and would recreate their discovery process. This would often highlight their great moments of discovery. Unfortunately, moments of limited success were recorded or remembered less frequently, which provided less useful feedback about negative aspects of the design.

Logging user actions is an obvious extension to the methodology. While logs alone will not capture the full story of exploration, they can be used in conjunction with interviews to refresh users' memories as well as figure out a quantitatively accurate version of where users spent most of their time. Logs may highlight users getting lost during exploration, or never using certain features that may have led to insights. Logging will hopefully serve as a tool to improve the accuracy of reporting on insights, and serve as a reference for reporting on failures. Advancing logging that allows users to annotate important states mid-analysis and later replay states, as described in the *systematic yet flexible* architecture, should yield advantages for data collection during evaluation as well.

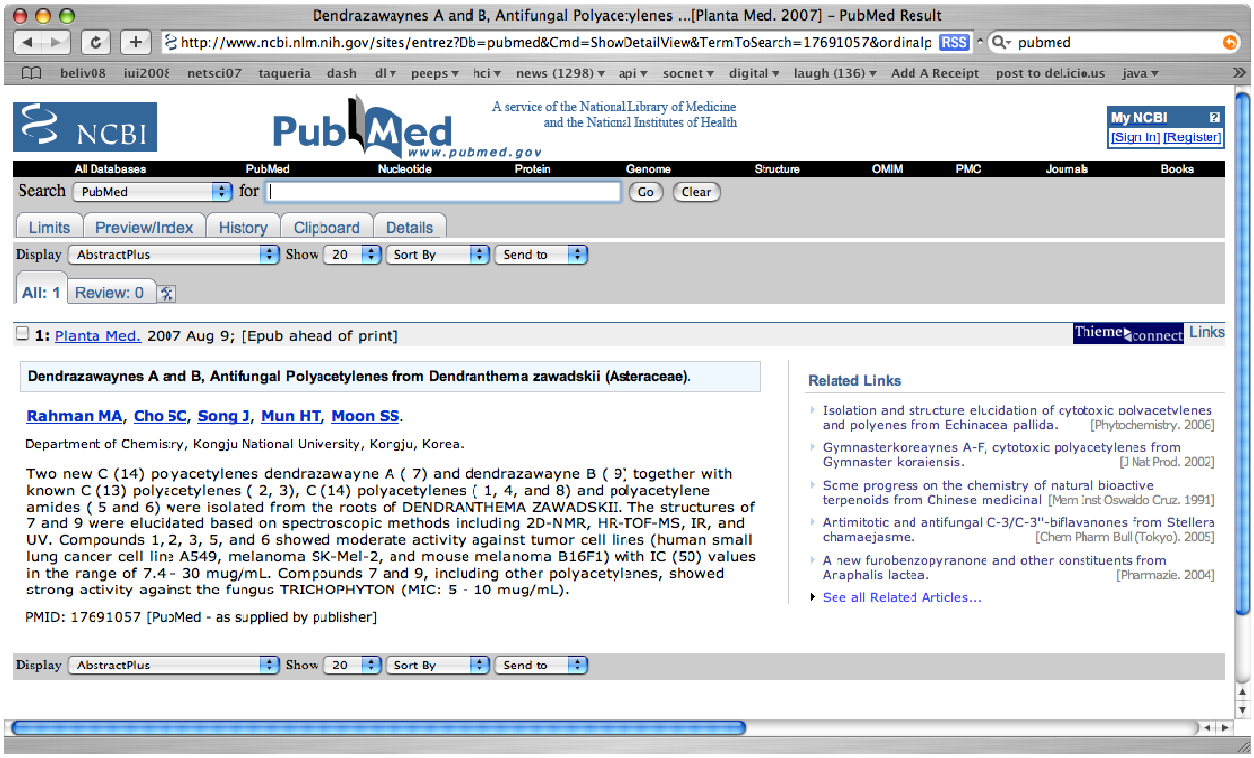
Appendices

A.1 Field Notes from Chapter 5's Case Study #2

The following is an abridged summary of field notes from Chapter 5's Case Study #2. These are included to give readers a sense of the type of information that was recorded during interviews.

A.1.1 Interview Phase

The National Library of Medicine runs a service called PubMed, a service of over 16 million citations in the medical field. When users reach a citation of interest, they are presented with a list of 5 related articles.



These related links are generated from Information Retrieval algorithms. These researchers were interested in understanding and refining these algorithms to make sure they were covering all of the relevant documents – and that if users were to

browse using these links, they'd eventually stumble upon all important documents. Essentially, they wanted to make sure there weren't any isolated or broken paths in the network. The lead researcher, Dr. Jimmy Lin, explained his goals:

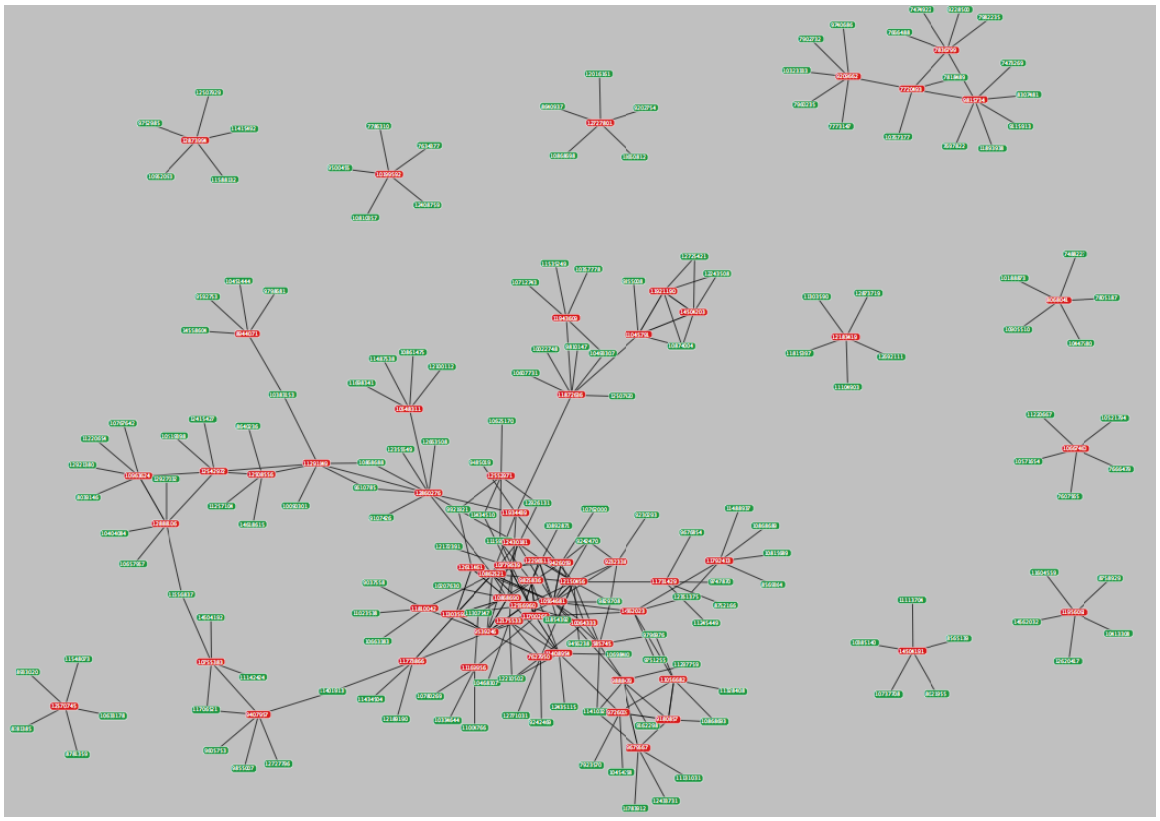
> For a given PubMed article, we are able to find a number of related articles (each associated with a relevance score). So one might imagine a large network of documents connected by these association links (some stronger than others). Users would "traverse" this network by clicking on "related article" links. I'm wondering what this network looks like. For example:
>
> - is the network densely connected? Or are there "islands" you can't get to just by clicking? (We would like to filter by strength of link---i.e., show only related score > 0.8)
>
> - for some networks, we have ground truth (relevance judgments as assessed by users): are these documents clustered? (these relevance judgments can be metadata on each node---i.e., show relevance nodes in red).
>
> - in addition, we were wondering if it was possible to get stats like diameter of network, etc.

Related to this is the Cluster Hypothesis in information retrieval---that relevant articles tend to be similar to each other (hence they cluster together). The researchers were interested in seeing if this hypothesis was true.

These researchers had no experience with network visualizations or social network analysis in the past.

A.1.2 Early Use

The first contact with these researchers was via email. Jimmy first had data from a TREC genomics test set, where they had ground truth on the relevance of documents (in this case, "what is the role of the gene GSTM1 in the disease Breast Cancer"). For every known relevant document, the top five related docs were linked (i.e., what would be displayed on the related links panel in PubMed)---the data file includes both rank and similarity score. This network is shown in the figure below:



The relevant documents are colored in red, whereas the irrelevant are colored in green. This visualization was quite exciting to the researchers, e.g. Jimmy proclaimed, “The new figure is exactly what I wanted to see!”

From this figure alone, the researchers reached a variety of insights. They were able to discern that these results support the hypothesis that relevant documents cluster together. They also noted, however, that there were a number of singletons---relevant documents that are not connected to other relevant documents. But the visualization shows that users can access most of the relevant documents by clicking on these related article links---all without having to go back to the hit list and reformulating a query. This figure also brought them to desire statistical evidence of this phenomenon, including:

1. How many connected/disconnected components are there in the network?

2. What is the average in-link for each relevant document? (from other relevant docs)?

3. What is the average out-link for each relevant document? (to other relevant docs)?

4. We want to characterize the clumpiness of rel docs---any statistic that does this will be good.

SocialAction didn't support features 2-4. I quickly implemented features 2 and 3 (which turned out to be that useful) and then deployed the new version in their offices of NLM on June 20, 2007.

I visited the offices of two NLM researchers (Jimmy Lin and John Wilbur) and gave them a brief training session on using SocialAction. For the training sessions, we used their real data (two networks of two topics), and show them how to rank, filter and visually explore the network. In this training session, it became apparent there were several features of importance to them that were missing from SocialAction. In particular, there were very interested in the direction of links (this was quickly implemented to have links show arrows when there is direction). After the meeting, the researchers formulated their goals and objectives:

Goals:

Explore the typology of related abstract networks in the biomedical domain. Characteristics of interest include:

the connectedness of abstracts (in and out degrees)

the number of disconnected components

the diameter of the graph

Understand the extent to which the Cluster Hypothesis holds in the biomedical domain. In short, the Cluster Hypothesis says that closely associated documents tend to be relevant to the same requests.

Desired outcomes:

Gain insights about the structure of related abstract networks.

Quantify the typology of such networks in a meaningful way: histograms, scatter plots, etc.

Publish high quality research articles with the above data.

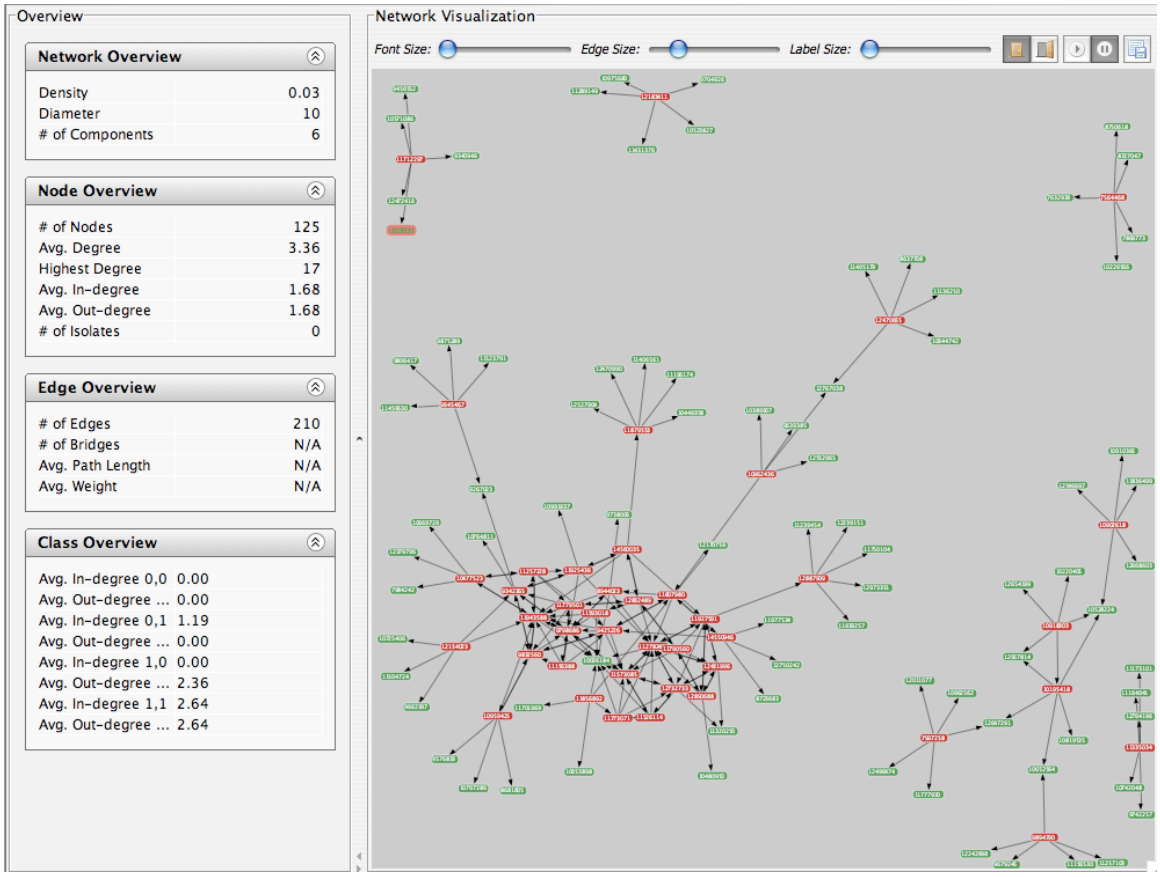
After these initial meetings, I provided regular minor feature requests and bug fixes throughout the early phase.

A.1.3 Mature Phase

Once SocialAction was a stable and robust enough platform, the researchers followed the following methodology:

The researchers had 50 topics from a genomic dataset, which in turn meant 50 networks to explore. Jimmy Lin wrote a perl script to convert their data into a format readable by SocialAction. For each network, the researchers manually:

1. Loaded each network.
2. Adjusted the visualization, zooming and panning where appropriate.
3. Froze the image (paused the force-directed algorithm).
4. Copied and pasted results from the Statistics Panel (e.g. the left side of the figure below). These researchers didn't use more advanced or node-specific statistics, but instead just relied on these "overall" network stats.



5. Used the visualization component (e.g. the right side of the above image) to note patterns. When looking at all 50 networks, patterns became obviously evident. In particular, the researchers paid attention to the following properties:
    a. The number of components.
    b. How dense the network was. This was a way of understanding how easy it was to navigate from one document to the next.
    c. Confirmed their hypothesis about the # of components (that most documents would cluster together) and the # of related documents (that related documents tended to point to each other – the number of red's connected to red's)
6. Pasted the statistics into excel and plotted the results after each network. They noticed patterns emerging after each new network was added to the spreadsheet and replotted.

The researchers spent about 10 minutes on each of the initial networks they explored. After about 5 and patterns were emerging, the researchers spent much less time, and then began simply copying & pasting the stats in Excel.

The researchers didn't use the Community feature, because they could not objectively assess which community parameters defined a good community. Furthermore, the communities found by the algorithms often turned out to be too fine-grained for their tastes.

This above methodology led to an accepted journal paper:

> Jimmy Lin, Michael DiCuccio, Vahan Grigoryan, and W. John Wilbur. Navigating Information Spaces: A Case Study of Related Article Search in PubMed. Information Processing and Management, 2008

In this paper, the researchers used SocialAction solely for their analysis. Visualizations from SocialAction were used to communicate how document networks look in the PubMed database. Furthermore, statistical information from SocialAction

were illustrated in tables and scatterplots to convince their readers of interesting phenomenon.

In summary, the authors got a unique perspective on how related documents connected to each other, and they were able to report these findings thanks to SocialAction.

A.1.4 Outcome

I met again with Jimmy Lin, the lead researcher. He briefed us on the methodology above and provided us feature requests.

> SocialAction Bugs/Feature Requests:
>
> - Disconnected components continue drifting apart forever. This shouldn't happen.
> - In community mode, communities continue drifting apart forever. This shouldn't happen.
> - Different behavior as nodes approach edges---perhaps edges should have repulsion?
> - **Ability to run in batch mode and do dump of statistics
> - Ability to remove (or dim out) nodes based on certain properties, without changing layout. Same with edges.
> - Ability to differentiate between edge types
> - Ability to do raw data dump of computed statistics: e.g., in/out degree of each node, community membership
>
> Important!!
>
> How SocialAction potentially could support my work flow:
>
> I use the tool to visualize the network, find patterns, etc.---however, my end goal may be something different. For example,
>
> develop algorithms for information retrieval based on properties of these networks. Therefore, it would be helpful to dump to statistics wholesale so I can manipluate, e.g., by a retrieval algorithm.

Jimmy summarized his intended workflow quite nicely. He intends (and did) use SocialAction as an exploratory data analysis device. By using the combination of visualization + statistics, he could find patterns that were prominent in the networks he was analyzing. From there, he would dump the useful statistics and use them to implement and inform new retrieval algorithms.

Bibliography

1. Adar, E. GUESS: a language and interface for graph exploration. In *Proc. SIGCHI conf. on Human Factors in computing systems*. ACM Press (2006), 791-800.
2. Ahlberg, C. and Shneiderman, B. The Alphaslider: A Compact and Rapid Selector. In *Proc. Conference on Human Factors in Computing Systems (CHI '94)* (1994), 365-371.
3. Amar, R., Eagan, J. and Stasko, J. Low-Level Components of Analytic Activity in Information Visualization. In *Proc. IEEE Symposium on Information Visualization* (2005), 111-117.
4. Amazon.com *Amazon.com*. http://www.amazon.com, (2007).
5. Auber, D., et al. Multiscale Visualization of Small World Networks. In *Proc. IEEE Symposium on Information Visualization* (2003), 75-81.
6. Barnes, J. and Hut, P. A Hierarchical O(n log n) Force Calculation Algorithm. *Nature*, 324 (1986), 446-449.
7. Bergman, L., et al. DocWizards: a system for authoring follow-me documentation wizards. In *Proc. ACM symposium on User interface software and technology*. ACM Press (2005), 191-200.
8. Borgatti, S. *Netdraw 2*. Analytic Technologies, (2007).
9. Borgatti, S., Everett, M. G. and Freeman, L. C. *UCINET 6*. Analytic Technologies, (2007).
10. Brandes, U. A Faster Algortithm for Betweenness Centrality. *Journal of Mathematical Sociology*, 25, 2 (2001), 163-177.
11. Brandes, U. and Wagner, D. *visone - Analysis and Visualization of Social Networks* In *Graph Drawing Software*, M. Junger and P. Mutzel. Springer-Verlag (2003).
12. Burton, M., et al. Secondary navigation in software wizards. In *Proc. CHI '99 extended abstracts on Human factors in computing systems*. ACM Press (1999), 294-295.
13. Card, S. K., Mackinlay, J. D. and Shneiderman, B. *Readings in Information Visualization: Using Vision To Think*. Morgan-Kaufman (1999).
14. Chen, C. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology* (2005), 359-377.
15. Chen, C. and Czerwinski, M. Empirical Evaluation of Information Visualizations: An Introduction. *Intl. Journal of Human-Computer Studies*, 53 (2000), 631-635.
16. Chi, E. H. and Riedl, J. T. An Operator Interaction Framework for Visualization Systems. In *Proc. IEEE Symposium on Information Visualization* (1998), 63-70.
17. de Nooy, W., Mrvar, A. and Batageli, V. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, Cambridge (2005).
18. Dean, J. and Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51, 1 (2008), 107-113.

19. Di Battista, G., et al. Algorithms for Drawing Graphs: an Annotated Bibliography. *Computational Geometry: Theory and Applications*, 4, 5 (1994), 235-282.
20. Di Battista, G., et al. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New Jersey (1999).
21. Dryer, D. C. Wizards, guides, and beyond: rational and empirical methods for selecting optimal intelligent user interface agents. In *Proc. International conference on Intelligent user interfaces*. ACM Press (1997), 265-268.
22. Freeman, L. C. Centrality in social networks: Conceptual clarification. *Social Networks*, 1, 1 (1979), 215-239.
23. Freeman, L. C. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press (2004).
24. Freeman, L. C. *Graphic Techniques for Exploring Social Network Data* In *Models and Methods in Social Network Analysis*, P. J. Carrington, J. Scott and S. Wasserman. Cambridge University Press, Cambridge (2004).
25. Freeman, L. C. Visualizing Social Networks. *Journal of Social Structure*, 1, 1 (2000).
26. Frishman, Y. and Tal, A. Multi-Level Graph Layout on the GPU. *IEEE Transactions on Visualization and Computer Graphics*, 13, 6 (2007), 1310-1319.
27. Gansner, E. R., Koren, Y. and North, S. Topological Fisheye Views for Visualizing Large Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 11, 4 (2005), 457-468.
28. Getoor, L. and Diehl, C. P. Link Mining: A Survey. *ACM SIGKDD Explorations*, 7, 2 (2005), 3-12.
29. Ghoniem, M., Fekete, J.-D. and Castagliola, P. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In *Proc. IEEE Symp. on Information Visualization* (2004), 17-24.
30. Gonzales, V. and Kobsa, A. A Workplace Study of the Adoption of Information Visualization Systems. In *Proc. Proc. I-KNOW'03: Third International Conf. Knowledge Management* (2003), 92-102.
31. Google *Notebook*. http://www.google.com/notebook/, (2007).
32. Grinstein, G., et al. The VAST 2006 Contest: A tale of Alderwood. In *Proc. IEEE Symp. on Visual Analytics Science and Technology* (2006), 215-216.
33. Grokker *Working List*. http://www.grokker.com, (2007).
34. Groth, D. P. and Streefkerk, K. Provenance and Annotation for Visual Exploration Systems. *IEEE Transactions on Visualization and Computer Graphics*, 12, 6 (2006), 1500-1510.
35. Hassan-Monteroa, Y. and Herrero-Solanaa, V. Improving Tag-Clouds as Visual Information Retrieval Interfaces. In *Proc. International Conference on Multidisciplinary Information Sciences and Technologies* (2006).
36. Havre, S., Hetzler, B. and Nowell, L. ThemeRiver: Visualizing Theme Changes over Time. In *Proc. IEEE Symposium on Information Vizualization*. IEEE Computer Society (2000), 115.

37. Heer, J. and Agrawala, M. Software Design Patterns for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12, 5 (2006), 853-860.
38. Heer, J. and boyd, d. Vizster: Visualizing Online Social Networks. In *Proc. IEEE Symposium on Information Visualization* (2005), 5.
39. Heer, J., Card, S. K. and Landay, J. A. prefuse: A Toolkit for Interactive Information Visualization. In *Proc. ACM Conf. on Human Factors in Computing Systems* (2005), 421-430.
40. Heer, J., Viegas, F. B. and Wattenberg, M. Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization. In *Proc. ACM Conference on Human Factors in Computing Systems*. ACM Press (2007), 1029-1038
41. Henry, N. and Fekete, J.-D. MatrixExplorer: A Dual-Representation System to Explore Social Networks. *IEEE Trans. on Visualization and Computer Graphics*, 26, 5 (2006), 677-684.
42. Henry, N., Fekete, J.-D. and McGuffin, M. J. NodeTrix: a Hybrid Visualization of Social Networks. *IEEE Trans. on Visualization and Computer Graphics*, 13, 6 (2007), 1302-1309.
43. Herman, I., Melancon, G. and Marshall, M. S. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. on Visualization and Computer Graphics*, 6, 1 (2000), 23-43.
44. Hersh, W., et al. TREC 2005 genomics track overview. In *Proc. 14th Text Retrieval Conf.* (2005).
45. Hodgman, C. An information-flow model of the pharmaceutical industry. *Drug Discovery Today: BIOSILOCO*, 1, 6 (2001), 1256-1258.
46. Hughes, J., et al. The role of ethnography in interactive systems design. *interactions*, 2, 2 (1995), 56-65.
47. Huisman, M. and van Dujin, M. A. J. *Software for social network Analysis* In *Models and Methods in Social Network Analysis*, P. J. Carrington, J. Scott and S. Wasserman. Cambridge University Press, Cambridge (2004).
48. Intuit *TurboTax 2007*. http://turbotax.intuit.com/, (2007).
49. Irani, P. and Ware, C. Diagramming information structures using 3D perceptual primitives. *ACM Transactions on Computer-Human Interaction*, 10, 1 (2003), 1-19.
50. Kang, H., et al. NetLens: Iterative Exploration of Content-Actor Network Data. In *Proc. IEEE Symp. on Visual Analytics Science and Technology*. IEEE Press (2006), 91-98.
51. Keim, D. A., et al. Visual Analytics: Scope and Challenges. *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics, Lecture Notes In Computer Science* (2008).
52. Kleinberg, J. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms* (1998), 604-632
53. Krackhardt, D., Blythe, J. and McGrath, C. KrackPlot 3.0: An Improved Network Drawing Program. *Connections*, 17, 2 (1994), 53-55.

54. Lamping, J. and Rao, R. The hyperbolic browser: A Focus+Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing*, 6, 4 (1995), 33-55.
55. Lee, B., et al. TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts. *IEEE Trans. on Visualization and Computer Graphics*, 12, 6 (2006), 1414-1426.
56. Lin, J., et al. Navigating Information Spaces: A Case Study of Related Article Search in PubMed. *Information Processing and Management*, In press (2008).
57. Lombardi, M. *Mark Lombardi: Global Networks*. Independent Curators (2003).
58. McGrenere, J., Baecker, R. M. and Booth, K. S. An evaluation of a multiple interface design solution for bloated software. In *Proc. SIGCHI conference on Human factors in computing systems*. ACM Press (2002), 164-170
59. Munzner, T. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In *Proc. IEEE Symposium on Information Visualization* (1997), 2-10.
60. Munzner, T., Guimbretière, F. and Robertson, G. Constellation: A Visualization Tool for Linguistic Queries from MindNet. In *Proc. IEEE Symposium on Information Visualization* (1999), 132-135.
61. Newman, M. E. J. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69 (2004).
62. North, C. Visualization Viewpoints: Toward Measuring Visualization Insight. *IEEE Computer Graphics & Applications*, 26, 3 (2006), 6-9.
63. O'Madadhain, J., et al. Analysis and Visualization of Network Data using JUNG. *Journal of Statistical Software*, VV, 2 (2005).
64. Parker, G., Franck, G. and Ware, C. Visualization of Large Nested Graphs in 3D: Navigation and Interaction. *Journal of Visual Languages and Computing* (1998), 299-317.
65. Perer, A. and Shneiderman, B. Balancing Systematic and Flexible Exploration of Social Networks. *IEEE Trans. on Visualization and Computer Graphics*, 12, 5 (2006), 693-700.
66. Perer, A. and Shneiderman, B. Integrating Statistics and Visualization: Case Studies of Gaining Clarity during Exploratory Data Analysis. In *Proc. Conference on Human Factors in Computing Systems*. ACM Press (2008), 265-274.
67. Perer, A. and Shneiderman, B. Systematic Yet Flexible Discovery: Guiding Domain Experts through Exploratory Data Analysis. In *Proc. Intelligent User Interfaces (IUI)* (2008), 109-118.
68. Perer, A. and Wilson, C. The Steroids Social Network: An Interactive Feature on the Mitchell Report. *Slate Magazine*, (2007), http://www.slate.com/id/2180392/.
69. Pirolli, P., Card, S. K. and Van der Wage, M. M. The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser. *ACM Transactions on Computer-Human Interaction*, 10, 1 (2003), 20-53.
70. Plaisant, C. The challenge of information visualization evaluation. In *Proc. Advanced visual interfaces*. ACM Press (2004), 109-116

71. Plaisant, C., Fekete, J. D. and Grinstein, G. Promoting Insight Based Evaluation of Visualizations: From Contest to Benchmark Repository. *IEEE Trans. on Visualization and Computer Graphics*, 14, 1 (2008), 120-134.
72. Plaisant, C., Grosjean, J. and Bederson, B. B. SpaceTree: Supporting exploration in large node-link trees: design evolution and empirical evaluation. In *Proc. IEEE Symposium on Information Visualization* (2002), 57-64.
73. Rao, R. and Card, S. K. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proc. Human Factors in Computing Systems (CHI '98)* (1998), 329-399.
74. Sageman, M. *Understanding Terror Networks*. University of Pennsylvania Press, Philadelphia (2004).
75. Saraiya, P., North, C. and Duca, K. An Evaluation of Microarray Visualization Tools for Biological Insight. In *Proc. IEEE Symp. on Information Visualization*. IEEE Press (2004), 1-8.
76. Saraiya, P., et al. An Insight-based Longitudinal Study of Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics*, 12, 6 (2006), 1511-1522.
77. Selker, T. COACH: a teaching agent that learns. *Communications of the ACM*, 37, 7 (1994), 92-99.
78. Seo, J. and Shneiderman, B. 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 (2006), 311-322.
79. Seo, J. and Shneiderman, B. A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization*, 4, 2 (2005), 99-113.
80. Shneiderman, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. In *Proc. Visual Languages* (1996), 336-343.
81. Shneiderman, B. and Aris, A. Network Visualization by Semantic Substrates. *IEEE Trans. on Visualization and Computer Graphics*, 12, 5 (2006), 733-740.
82. Shneiderman, B. and Bederson, B. B. Maintaining concentration to achieve task completion. In *Proc. Conference on Designing for User eXperience*. AIGA: American Institute of Graphic Arts (2005).
83. Shneiderman, B., et al. Creativity Support Tools: Report from a U.S. National Science Foundation Sponsored Workshop. *International Journal of Human–Computer Interaction* 20, 2 (2006), 61-77.
84. Shneiderman, B. and Plaisant, C. Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. In *Proc. Beyond time and errors Workshop (BELIV)*. ACM Press (2006), 1-7.
85. Spotfire *DecisionSite* http://www.spotfire.com, (2007).
86. Spotfire *Introduction to Spotfire DecisionSite Analysis Builder*. http://spotfire.tibco.com/spotfire_downloads/whitepapers/analysis_builder.pdf , (2007).
87. Stevens, R., et al. myGrid and the drug discovery process. *Drug Discovery Today: BIOSILOCO*, 2, 4 (2004), 140-148.

88. Tauscher, L. and Greenberg, S. How people revisit web pages: empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies*, 47, 1 (1997), 97-137.
89. Tukey, J. W. *Exploratory Data Analysis*. Addison-Wesley (1977).
90. van Ham, F. *Interactive Visualization of Large Graphs*. Technische Universiteit Eindhoven, 2005.
91. van Ham, F. and van Wijk, J. J. Interactive Visualization of Small World Graphs. In *Proc. IEEE Symposium on Information Visualization* (2004), 199 - 206.
92. van Rijsbergen, C. J. *Information Retrieval*. Butterworths, London (1979).
93. Viegas, F. B., et al. Digital artifacts for remembering and storytelling: posthistory and social network fragments. In *Proc. HICSS 2004* (2004), 118.
94. Viegas, F. B. and Donath, J. Social Network Visualization: Can We Go Beyond the Graph? In *Proc. CSCW'04 Workshop on Social Networks* (2004).
95. Viegas, F. B. and Wattenberg, M. Communication-Minded Visualization: A Call to Action. *IBM Systems Journal*, 45, 4 (2006).
96. Viégas, F. B., et al. Many Eyes: A Site for Visualization at Internet Scale. In *Proc. IEEE Symp. on Information Visualization* (2007).
97. Ware, C. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers (2000).
98. Warmington, A. Action Research: Its Methods and its Implications. *Journal of Applied Systems Analysis*, 7 (1980), 23-29.
99. Wasserman, S. and Faust, K. *Social Network Analysis: Methods and Applications*. Cambridge University Press (1994).
100. Wattenberg, M. Baby Names, Visualization, and Social Data Analysis. In *Proc. IEEE Symposium on Information Visualization*. IEEE Computer Society (2005), 1.
101. Wattenberg, M. Visual exploration of multivariate graphs. In *Proc. SIGCHI conference on Human Factors in computing systems* (2006), 811-819.
102. Watts, D. J. *Six Degrees: The Science of a Connected Age*. W.W. Norton & Company, New York (2003).
103. Watts, D. J. and Strogatz, S. Collective dynamics of 'small-world' networks. *Nature*, 393 (1998), 440-442.
104. Wong, P. C., et al. Graph Signatures for Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics*, 12, 6 (2006), 1399-1413.
105. Xu, J. J. and Chen, H. CrimeNet Explorer: A Framework for Criminal Network Knowledge Discovery. *ACM Transactions on Information Systems*, 23, 2 (2005), 201-226.
106. Yi, J. S., et al. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13, 6 (2007), 1224-1231.
107. Yin, R. K. *Case Study Research: Design and Methods*. Sage Publications, Inc. (2002).
108. Ziegler, E., et al. Visualizing and exploring large networked information spaces with matrix browser. In *Proc. IEEE Symposium on Information Visualization* (2002), 57 -64.