Submitted to INTERACT 2005—please do not cite or circulate without authors permission

Social network visualizations of streaming data: Design and use considerations

Daniel M. Russell and Malcolm Slaney

IBM Almaden Research Center
650 Harry Rd., San Jose, CA, 95120
daniel2@us.ibm.com, malcolm@ieee.org

Abstract. Understanding networks of people linked by some common factor is an important task in many domains. Most commonly, a user creates a visualization of social interactions to see the patterns of interactions between individuals, and then used to find and identify important groups. Networks of individuals and links between them form graphs that vary with time and importance. Visualizing the changes in social networks over time is a non-trivial design task, imposing interesting demands on the visualization and interaction model. In this paper we briefly analyze the user requirements for interactive visualizations of streaming social network data. We find that these continuously updated, dynamic displays need: (1) controls that permit time-based control of the visualization, including pausing, restarting and variable speed playback of the data, (2) the ability to continue importing and processing streamed information even the display is paused, (3) a visually represented method to track changes in the displays over time, (4) interaction methods to allow drill-down from the visualization to original source data, and (5) information extraction from the displayed social network. We describe our visualization tool, SSNV, showing how it embodies these interaction requirements.

1 Introduction

Streaming data that suggests relationships between people and organizations comes in many forms—e-purchase data streams, click-throughs, instant messaging, email—and is characterized by never-ending, continual updates to the data set. In many cases, end-users need the ability to understand both what's happening at the moment and what has led up to the current state of affairs.

Standard social network analysis (SNA) combines evidence about relationships among people as seen in the collected data, producing between-person relationship measures, potentially identifying groups and clusters of people as portrayed by the data analytics. [2, 4]

Social network data is intrinsically time-varying: emails, chat sessions, discussions and meetings all take place at a specific time, although the data may be collected or analyzed at a much different time. Streaming social network analysis collects real-time data (such as an email or IM chat stream [17]) and creates an on-the-fly analysis of social relationships among people referred to in the stream data.

A streaming data visualization has properties that are fundamentally different than those of static data sets. The goal of streaming social network analysis is to continuously handle incoming social networking information, and rapidly identify social groups as they form, reform and dissolve. The addition of streaming data adds a constantly changing time-base for the analytics and visual presentation.

The effective use of streaming data requires that the user-interaction tasks and interaction models be aligned with a streaming model. Issues that never arise in unchanging data sets suddenly become pivotal: the need to analyze incoming data in real-time and the need to provide an understandable user interaction for the ever-increasing data collection. Such streaming live data is common in real-time data and telemetry applications of all kinds, yet the tradition of data visualization is to work on an established data set, doing manipulations of the data set.

The goal of social network visualization is to rapidly explore the space of possible social network data interpretations, looking for patterns and structure in the data as seen through the layouts and structures that emerge. [5] With such data, the dynamic nature of shifting relationships and patterns of connections are important—and with streamed information this time basis becomes immediately apparent and important.

2 Characteristics of Streaming Social Network Data

The most salient characteristic of streaming data is that it never stops arriving. Since one of the goals of analyzing streaming social network data is to have an ever-current picture of the networks of interest, the demands of streaming data puts an interesting twist on the role of the visualization system as it provides both current and historical views onto the increasing data set.

While many dashboard visualizations [16, 14] provide results analysis on streaming data that summarizes the current state of the world, social network data is especially useful when the time course of events can be seen and examined in detail. Social networks often change with time, fluctuating in membership, relationships and overall network stability and structure. This implies a real-time data processing system design needs to be used to allow user interactions to continue while data flows into the system. At a high level of description, this is just the Tivo™ effect—allowing the user to pause real-time data updates while the information continues to be collected (and analyzed) in the background, with an interface to back up and replay previously recorded data.

Unlike Tivo™, which essentially copies incoming data to a backing store unaltered, a streaming social network visualization system needs to perform its analysis in real-time to keep up with the current display. Similarly, when the user browses earlier times in the recorded history of events, the system needs to continue to provide both current stream data updates, continued analysis of the incoming stream, as well as provide visualizations of previously stored data. This implies that the data ingestion process be kept separate from the view of earlier data. Viewing data then requires a way to move forward and backward in time (as well as the ability to see "now," which is just the special case of seeing the historical data at the most recently acquired moment.

What does one do with streaming data? As with all streaming data problems, the issues are determining what merits attention, and how to work with the salient information. The key problem becomes one of figuring out what to look at, and how to keep the display simple**.**

3 A Brief Task Analysis of Social Network Visualizations (SNV)

While social network visualizations (SNV) have been very popular demonstrations of graph layout algorithms and animation techniques, there are a set of deeper issues about how an end-user would actually interact with and use a visualized SNA[1].

We conducted interviews with three professional SNV end-users on the topic of current SNV use and extensions to streaming visualizations. Our analysis was based on multiple face-to-face interviews, demonstrations, heuristic evaluations of prototypes we created and telephone interviews over a ten month period. The goal was to determine the most important aspects of SNA use and problems as manifest in their practice, and to envision similar issues in streaming social network analysis. We focused on understanding both the net-value of the SNV display, as well as details of their interactions with current SNA visual tools. In particular we asked: (1) What kinds of tasks does one do with an SVN? (2) How do current tools support or impede those tasks? (3) What problems arise in current SNV tool use?

Their answers surprised us: many of the attractive and glitzy features of current SNVs turned out to be detrimental or just uninteresting. In particular, social networks that were "always moving" (either with jittering node locations, or with nodes flying around unpredictably when new data was added) were found to be annoying. In addition, extremely large networks (typically, more than 50 nodes in the display) were difficult to use. From these discussions we identified the following common tasks that are not always well-supported in current SNV tools.

A. Identifying groups and patterns: Key to an SNV's success is the ability to visually present groupings of individuals that correspond to clearly identifiable groups as seen in the underlying data. Ideally, patterns of relationships (e.g., a bridge person connecting two otherwise disconnected groups [4]) should be easily discernable.

B. Tracking individuals and groups: In addition to showing group patterns, users want to be able to follow the members of a group over both short and long periods of time (where they come from, where they go). The SNV should provide a way to track individuals, as well as making their entry and exit from the visualization somewhat more graceful than random point appearances (which the users find difficult to visually understand).

C. Additional information about individuals and groups: Underlying every social network analysis is a set of base data that gives rise to the visualization. In many

[1] We distinguish between the practice of analyzing social networks (SNA) and computer-based visualizations (SNV).

cases, analysts want to be able to drill down to the underlying original data and inspect it. Although there is often a large quantity of data beneath the visualization, being able to get access to the source content is a key for validating the visualization, as well as getting to a deep understanding of the relationships. While an SNV can show some selected aspects of social relationships, end-users frequently want to look at the data in ways suggested by the network relationships, but not limited by it.

D. Working with the data: Although it seems obvious in retrospect, the ability to import and export data to/from the visualization tool was seen as very important. Some visualizations exist solely to highlight relationships, making visible the previously unseen, with results that are immediately apparent and usable. While SNVs do this, users need to also create lists of individuals and groups identified on the visualization. They need to be able to selective identify groups and individuals, then export the associated identifying data for further analysis by other tools and systems. SNVs don't stand alone, but are part of a larger repertory of analytic tools.

4 A Streaming Social Network Visualization Tool (SSNV)

SSNV is the tool we co-developed with our subject matter experts in response to their heuristic evaluations of our prototypes and their commentaries on current SNA systems. SSNV is a system built to analyze and visualize a large email stream, analyzing it for relationships between email senders and receivers. First we describe how SSNV analyzes an input email stream for social affinities, then discuss how the interface observations of Section 3 are implemented in SSNV.

How SSNV Works: Affinity Analysis

Our social network analysis is based on a measure of affinity between individuals and a visualization that uses a standard spring-force model between nodes. The mechanics of our SNA visualization differs from previous work in three important ways:

1) Calculating the affinity matrix over time,
2) Changing the mass of each object, and
3) Initializing the object's position in the visualization

There are many ways to measure the basic information needed to drive a social network analysis. We call this an affinity measurement, and each type of communication gives some evidence that the two parties are linked. Mutton's work [11] describes several kinds of measurements that are possible in a chat room environment, including directly addressing another party and temporal proximity. Here we simply infer an affinity when two people exchange an email message. More precisely, an email message contributes a unit of affinity if the message is sent to less than 10 names, and 0 units of affinity for more than 10 names.

A single email message does not suggest that two people are connected for *only* the time it which it takes to read the email. People form teams over longer period of time and their affinity rises with each message exchanged, and falls over time. We model this rising and falling affinity as a function of time, *t*, with an exponential curve:

$$A(t) = e^{-(t-TO)/T} \qquad \text{for } t \geq 0 \qquad (\mathbf{1})$$

where *TO* is the time of the message and *T* is the decay time for the affinity function. In our explorations, a time constant of 2–3 days was useful and serves to bridge the time over weekends. After a person's maximum affinity (with all other people) drops below a fraction of one message, that person is marked inactive, and its forces are ignored, its position is unchanged, and it is no longer displayed in the visualization.

Our simulation is based on the evolution of a network of springs that apply forces to the people in the graph. Given the vector sum of all forces acting on an object we can compute its new velocity. This velocity is a function of the object's mass. Objects that have been on the screen for a longer period of time have a larger mass, and thus move more slowly. New objects are "lighter" and move around the existing objects. (Another common spring-based layout algorithm is the Kamada-Kawai [6].)

The spring-based simulation is important for human perception. The nodes move slowly based on easy-to-understand forces. The one remaining problem centers on the initial position of objects. At first we placed the nodes randomly on the plane, and let the spring simulation move them over time to the right position. But this approach complicates the user's cognitive task—each randomly placed, new node pulls the existing objects in a random direction that makes it harder for the user to understand. Instead, a better approach is to first place the objects at a position near the center of mass of the objects that are connected to the new object. Then we lock the position of the old objects and refine the position of the new objects by iteratively calculating forces acting on the new objects and adjusting their position relative to the existing constellation. The goal is to find the best possible initial position before the results are shown to the user, and before the existing objects start moving again.

Animating Changes in SSNV

An important aspect of the SSNV model is the ability to move forward and backwards in time. This is easy to do with a recorded movie, but then it is hard to interact with the data. The network and the data are frozen and they can only be viewed as images. Instead, we found it is necessary to interact with the history, perhaps marking interesting nodes and see how they change groups over time.

The SSNV keeps all the data produced by the simulation and uses this data to replay the animation. Now at any time the nodes can be inspected, their properties checked, and their color changed so that it is easier to track the person through time. The recorded data includes each the record of the person-object's position, as well as whether the object was active or not (and thus displayed).

In the SSNV the history mechanism is facilitated by keeping track of two different times. Data enters the system with a natural time based on the data's time stamp. At the same time, the records for the simulation are stored in a database, so the user can get more information about the messages that caused a person to show in the visualization.

The display time is decoupled from real time. In normal operation, the social network in the SSNV is updating in real time based on data that is currently entering the system. But at some point, either because the user sees something that is interesting or because he returns to the system after a break, the user will want to review the history and evolution of the social network. By grabbing the pointer at the bottom of the screen, the current display time can be moved to an arbitrary point in history.

We would like to keep the affinity matrix over time, so we can show the history of links, but this represents a matrix of N^2 points for every point in history. Instead we compute an approximation for the affinity matrix by querying the real-time database for past email and rebuild the affinity data.

User Interface Aspects of SSNV

From our discussions with the SNA experts it became clear that the interaction model for SNVs needed to incorporate several additions beyond current standard SNV practices. Of the four major areas of concern (A-D, above), the first is a direct reflection of the analytic algorithms used for visualization (and outside the scope of this paper), while the latter three suggested immediate changes to SSNV interactions.

Tracking individuals and groups: Tracking groups and individuals can be done by interactively changing the display features of selected nodes. Any node within the visualization is selectable and has editable visual properties (such as shape, color, size, label-type, etc.) These visual properties persist over the time-extent of the visualization, allowing the user to find an individual and then track them forward and backward in time by dragging a time slider to examine other time points of the analysis. In the same way, the entire display supports filtering (to reduce the total number of nodes to those that are within specified criteria) and other viewing modifications over the entire history of the captured data. Thus, by changing the view criteria, the user can move the time slider to another point in the social network history, and see the data from that time filtered, labeled or re-viewed according to the current set of viewing properties.

Additional information about groups and individuals: Finding out additional information can be done simply by right-clicking on a node to drill-down for more data. The nodes displayed can be filtered by changing an interaction control to set the level higher (to show more heavily connected people) or lower (to show more people, including those that are less well connected to other in the network).

Working with the data*:* A set of selected nodes and their associated data (e.g., name, affinity value, email messages, etc.) can be exported to a file for use outside of the SSNV. As with the other interaction techniques, nodes can be selected for export or additional information at any time while displayed.

Time-based display: In addition to these interaction requirements, the streaming nature of the data suggested several additional features needed for use. Since the SNV for streaming data is inherently paced by incoming data, the SNV needs controls that permit time-based control of the visualization, including pausing, rewinding, fast-forward, restarting and variable speed playback of the data. We also implemented a kind of "comet trail" (selectable on/off) that shows the motions of nodes over the past few moments of time. The trails are especially useful when monitoring an individual's movement over different parts of the display—as happens when a person moves from one group to another.

Another issue that arose was the ambient motion of nodes on the graph. With many SNA algorithms, nodes can appear, disappear and move nearly randomly on the display. While the SSNV layout and update algorithm attempts to minimize the overall jumpiness of node placement through adding inertial effects, by making node mass a user selectable property of a node, individuals can be effectively "pinned" in place, making them very resistant to arbitrary motion.

5 Using SSNV: An Example

In this example SSNV accepts the Enron email corpus [7] replayed back to it as an email stream. (To compress time, an email server sends it to the SSNV client at 10,000 times real-time of the real timestamp: a year then takes less than an hour, a time compression that makes simulations of year-long streams possible.). The affinity analytic used is the one describe above in section 4. Other analytics could be used, but this is simple and performs well within the time performance requirements of the system.

To limit the complexity of our graphs, and allow us to visualize the results in the confines of these columns, we only looked at the social network between the primary people included in the Enron database. By automatically searching the "From" address in the all the sent-mail folder, we extracted 250 distinct names that we deemed to be primary. Thus almost 60k messages were used in this study, or about 100 messages a day.

With this email data streaming into the SSNV client, we illustrate two key features of SSNV: (1) the ability to move back and forth in time while also manipulating display properties of the underlying visual display, and (2) the ability to drill down to the data that defines the affinity between nodes.

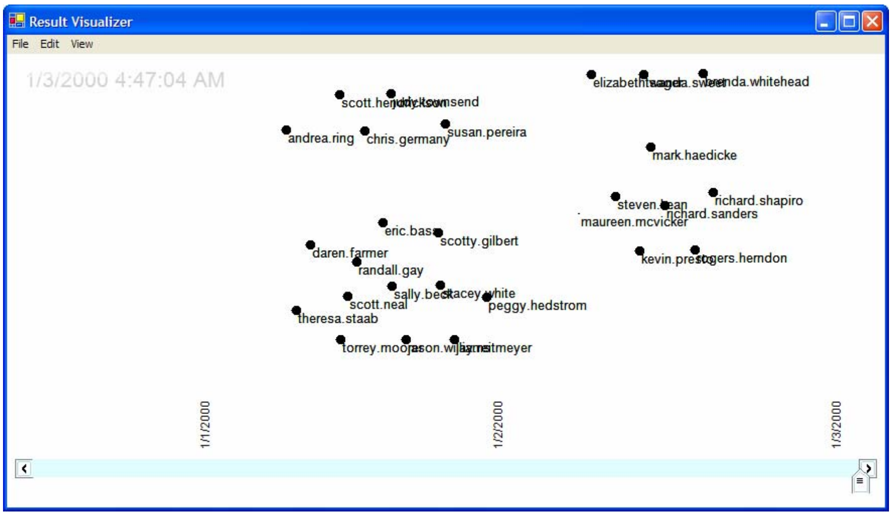

Fig. 1. The SSNV display showing a still frame from the Enron email stream. This is the set of people on 1/3/2000 (just after the new year) who have been exchanging email and have the affinity clusters as shown here.

With the time cursor positioned all the way to the right (as in Fig.1), the user sees the social network as of "now," the leading edge of the stream in time. Nodes representing people appear on the display, move around as affinities change with traffic between people; email patterns suggest shifting patterns of work and communication.

Figure 1 shows three, possibly four, fairly obvious self-organized groups of people. But by showing the affinity link strengths, the pattern of interaction becomes evident.

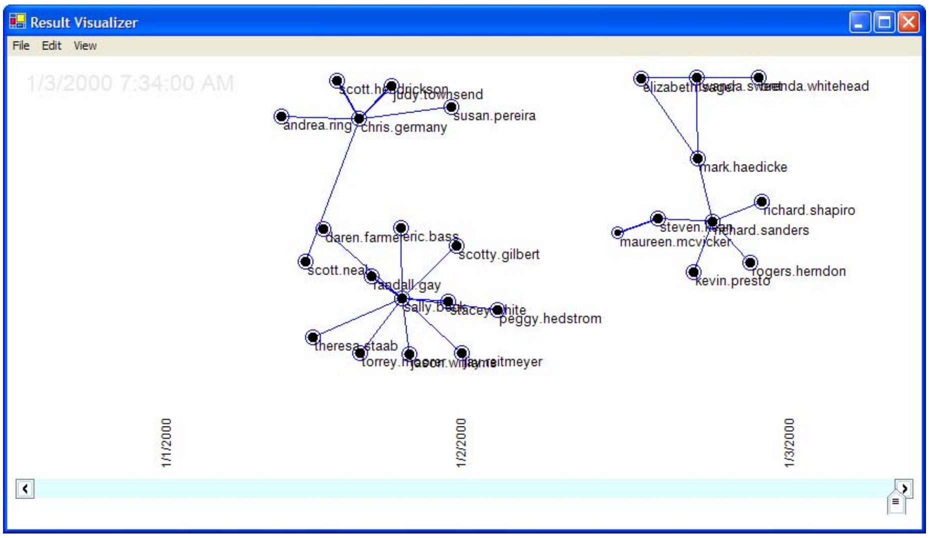

Fig. 2. Just a bit after Fig 1 was taken, the link affinities were turned on to show that two bridge people link each of four obvious groups.

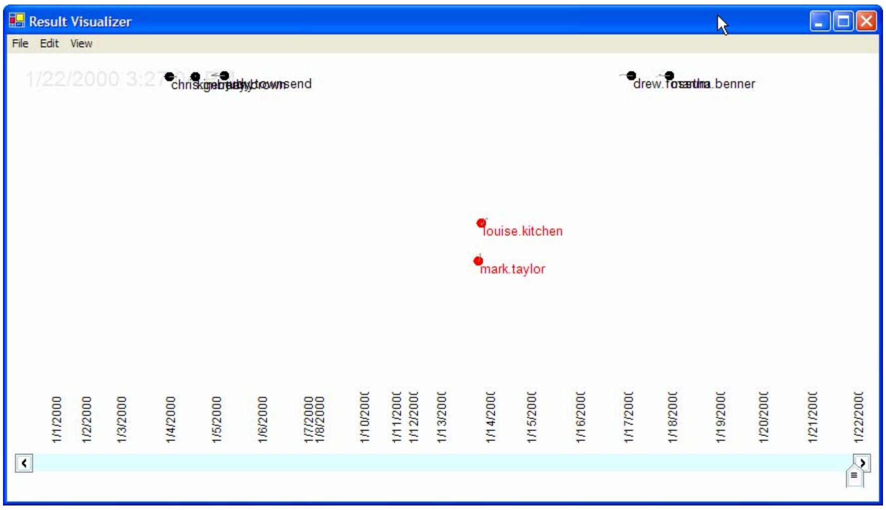

Fig. 3. Nodes can have their visual properties changed. These two nodes are changed to red and now have that property from the beginning of the timeline until the end.

In later investigation of the email stream, the user noticed that Kitchen and Taylor are frequently seen with a high degree of affinity (usually a value > 5.0, which is extraordinary for this peer group). On this date (1/22/2000) they are seen in close collaboration. The user selects the two for more careful inspection, and colors them red for easy visibility in subsequent investigation.

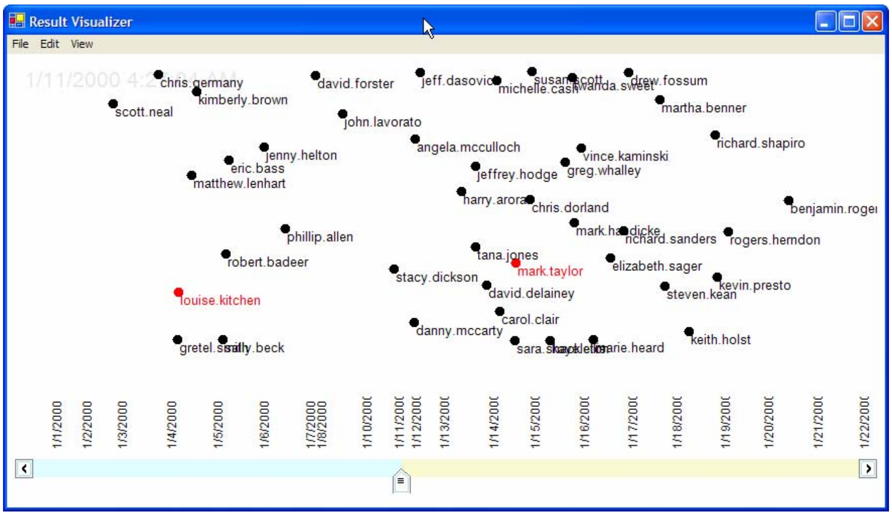

Fig. 4. The time slider moves backward in time, showing the previously colored nodes with their colors prospectively.

By going backward in time to a specific date (1/11/2000) with Kitchen and Taylor still highlighted in red, the user discovers other group affiliations. By selecting additional nearby people and labeling them in a second color, the user can grow the identified set of potential collaborators and easily follow the evolution of the secondary social network.

In Figure 5, with the nodes marking Kitchen and Taylor selected, the user can drill down to find all the messages in the base data set that went into the determination of their affinity value. The blue links in the subject column of the table are links to the body of the email.

Date	From	To	Subject
1/17/2000 2:31:00 AM	mark.taylor@etron.com	louise.kitchen@etron.com	West Power Customer Web SIte
1/17/2000 2:31:00 AM	mark.taylor@etron.com	louise.kitchen@etron.com	West Power Customer Web SIte
1/17/2000 6:50:00 AM	mark.taylor@etron.com	louise.kitchen@etron.com	Re: Eastern - Summary of their Response to PA/ETA
1/17/2000 6:50:00 AM	mark.taylor@etron.com	louise.kitchen@etron.com	Re: Eastern - Summary of their Response to PA/ETA
1/17/2000 12:25:00 PM	louise.kitchen@etron.com	mark.taylor@etron.com	Eastern - Summary of their Response to PA/ETA

Fig. 5. The drill-down table showing details of data that determined the affinity between two individuals.

6 Related Work

While there are a large number of social network visualization systems [1,3,8,9,10,11], the vast majority of available tools do not address streaming social network data analysis. Instead, most are retrospective analysis of captured streams that are then analyzed and rendered into movies with fairly limited interaction.

Of these, *sonia* [10] provides a rich toolkit where relational events are aggregated across a time window. This approach of sliding a window across events is particularly well suited for their educational domain, where data is often rapid and of short duration. The sliding window approach differs from SSNV's affinity cutoff mechanism, but is another approach to limiting the number of visible relationships at any one time.

PieSpy [11], like sonia, also generates SNVs on IRC participant data (using nicknames and messages between individuals for data). It has a nearly-streaming capability to analyze a network on demand to show the current behavior of a channel, and it can also process a datafile of time-stamped events to create a movie.

Nardi, Whittaker, and Schwarz [12] study collaborative behavior through social network analysis. Their qualitative research shows many factors influencing individuals' choices in initiating collaboration. In addition they provide detailed descriptions of how social ties change over time, varying across projects and organizations. While they did not create a visualization of the social networks in their original study, their results lead to Contact Map [13], a system that does visualize ego–centric social networks.

There are streaming data analysis tools in the world, although they are nearly all tools to support system administration and are focused on analyzing real-time streams of data to defend against network attacks, or to characterize the workload of large groups of processors.

7 Future Work

Extending the current work will rest primarily on providing increasingly richer and better ways of analyzing the underlying source data, as well as providing multiple interpretations of it. One approach would be to use topical/semantic clustering in addition to social network in order to see alternative views or analyses of the same data, drawing on some of the "layers" ideas illustrated by Sacks' conjoining of semantic net and social net analysis [15].

In addition, streaming data can come from multiple sources, with event data arriving at different times and potentially out-of-sequence. That is, an event might occur at one time, but not be registered with the system until some other, much later time. Again, the approach of separating data ingestion from data visualization provides a neat separation of data acquisition from data viewing. Late arriving data would be sequenced into the data store and then reanalyzed by the viewer to provide newer, more accurate view of the data, even late arriving data.

Since data may arrive out-of-sequence, we realize that analyses may also return results out-of-time as well. That is, in SSNV the analytics are tuned to keep up with the flow of data into the system. However, some analytics may take significantly longer and will need to be added to the results stream well past the time when the data arrives. This suggests that integrating late-arriving data may well be critical to streaming analysis of complex media types such as video or cross-stream correlations.

Finally, we anticipate creating extensions to the current time-based visualization that extend the idea of time in nonlinear ways. For instance, a "blink comparator" is a device used by astronomers to compare two images of a star field that are slightly different. Building comparators that highlight differences between similar social situations, or over a group across a long period of time. The goal of all this work is to provide new tools and ways of seeing into complex data about which it is difficult to form coherent a priori queries. The ultimate goal is to provide exploration tools for streaming data, and this is just our beginning.

9 Acknowledgements

We very much thank our anonymous subject matter experts, whose work with anonymous social network analysis tools and visualizations informed so much of our work. We are also grateful to the Freedom of Information Act and CMU for providing the opportunity to work with the Enron email corpus.

References

1. Baur, M., Benkert, M., Brandes, U., Cornelsen, S., Gaertler, M., Köpf, B., Lerner, J., Wagner, D. visone - Software for Visual Social Network Analysis. Proc. 9th Intl. Symp. Graph Drawing, LNCS 2265, Springer-Verlag (2002) 463–464
2. Berkowitz, S.D. *An Introduction to Structural Analysis: The Network Approach to Social Research*. Butterworths, Toronto. (1982)
3. Franham, S., Portnoy, W., Turski, A., Cheng, L, Vronay, D. Personal map: Automatically modeling the user's online social network. INTERACT '03, 567–574 (2003)
4. Freeman, L. Visualizing Social Networks. Journal of Social Structure (e-journal: http://www.cmu.edu/joss/content/articles/volume1/Freeman.html)
5. Horn, D. B., Finholt, T. A., Birnholtz, J. P., Motwani, D., Jayaraman, S. Six degrees of Jonathan Grudin: A social network analysis of the evolution and impact of CSCW research. Proc. 2004 ACM Conf. on Computer Supported Cooperative Work (CSCW) (2004) 582–591.
6. Kamada, K. and Kawai, S. An Algorithm for Drawing General Undirected Graphs. Information Processing Letters 31, (1989) 7–15.
7. Klimt, B., Yang, Y. The Enron Corpus: A New Dataset for Email Classification Research. European Conference on Machine Learning (2004) 217–226.
8. McGrath, C., Krackhardt, D., Blythe, J. Visualizing Complexity in Networks: Seeing Both the Forest and the Trees. Connections, 25(1) . (2003) 37–47
9. McDonald, D. W. Recommending collaboration with social networks: A comparative evaluation. Proc. Computer-Human Interaction (CHI) Conference, CHI-2003, (2003) 593–600, Ft. Lauderdale, FL
10. Moody, J., McFarland, D. A., Bender-deMoll, S. Visualizing Network Dynamics. American Journal of Sociology, (2005) (in production)
11. Mutton, P. Inferring and Visualizing Social Networks on Internet Relay Chat. Eighth International Conference on Information Visualization (IV04), (2004) 35–43.
12. Nardi, B.A., Whittaker, S. and Schwarz, H. NetWORKers and their Activity in Intensional Networks. Computer Supported Cooperative Work: The Journal of Collaborative Computing, 11 (1-2), (2002) 205–242
13. Nardi, B.A., Whittaker, S., Isaacs, E., Creech, M., Johnson, J. and Hainsworth, J. Integrating Communication and Information Through ContactMap. Communications of the ACM, 45 (4). (2002) 89–95.
14. Negash, S., Gray, P., Business Intelligence. Ninth Americas Conference on Information Systems, (2003) 3190–3200.
15. Sack, W. Discourse diagrams: Interface design for very large-scale conversations. Proc. HICSS-33, (2000) Hawai'i.
16. Smith, M. A., Fiore, A. T. Visualization components for persistent conversations. Proc. of Conf. on Computer-Human Interaction (CHI). . (2001) 136–143
17. Vronay, D., Smith, M., Drucker, S. Streaming Media Interfaces for Chat, Proc. of Conf. on Computer-Human Interaction (CHI) (2002)