22

Graph Drawing for Data Analytics

Stephen G. Eick
VisTracks and U. Illinois at Chicago

22.1 Introduction

Over the last decade graph drawing and network visualization has emerged as an exciting research area that is addressing a significant problem: how to make sense of the ever increasing amounts of relational information that has become widely available. With the growth of networking and decreasing cost of storage it has become technically feasible and cost effective to store and access vast sets of information. The academic, business, and government challenge is how to make sense of this information and translate the insights into value-producing activities.

As a new emerging field, there will certainly be opportunities for network visualization and graph drawing technology. There have already been some early successes and also many prototypes that have been research successes but have not led to successful deployments. Unfortunately, not all network visualizations create enough value so that users will switch over from conventional user interfaces to adopt new visual interfaces. The goal for this chapter is to present a simple framework that predicts problem areas where network visualization will achieve utilization and result in successful business applications—that is, be useful enough so that users will adapt new visual interfaces. For our academic colleagues our framework is not intended to identify what network problems are interesting for research nor is it intended to identify high-quality results. Rather it attempts to predict which application areas might lead to commercially successful applications.

Network visualizations are exciting and the demos inevitably generate interest among potential users. Unfortunately, however, visualization, as exciting as it is, only involves the

user interface or presentation layer in a technology stack. Useful network applications solve problems that involve collecting relational data, manipulating it, organizing it, performing calculations, and finally presenting the results to users. The value of the application is captured by the complete system. It is often the case that each system component individually is not particularly useful. For example, tires are not useful without a car, but better tires improve a car's performance. The presentation layer, like beauty, is only "skin deep" and the usefulness of the application comes from the whole solution and not just the "lipstick."

Thus, by itself, network visualization is naturally a feature of system and rarely is a complete application by itself. This, unfortunately, makes commercial utilization of a new technique or novel method difficult. With a few exceptions, the technology must be part of an application to capture sustainable value. Network visualization "makes it better" but, except in rare situations, does not make it. The network visualization value stack challenge is to find applications where network visualization creates enough value, either by itself or as part of an applications, to support utilization where it is a key part of the value proposition.

Throughout this chapter, the term *network visualization* refers to methods for visualizing graphs and networks that make use of graph drawing techniques.

22.2 Where Network Visualization Creates High Value

At its most basic level, network visualization is a technique for helping analysts understand structure and relationships. This section describes seven broad classes of information problems, illustrated by examples, where network visualizations create significant value.

22.2.1 User Interface

In certain cases, the *user interface* is essentially a complete application. The canonical example of this is *computer games* which are innovative and sophisticated user interfaces that involve, relatively speaking, little computation and no data integration. Successful games must have a great user interface that challenges and engages prospective players within the first few seconds.

Perhaps the closest network visualization application where the interface is the application involves graph drawing and layout. Arguably, the most successful application in this space is Microsoft's Visio™. Visio is perhaps the most widely used graph drawing package and is distributed as part of Microsoft Office.

22.2.2 Visual Presentation and Branding

Visual presentation and *branding* involves creating custom 3D displays of networks for presentations that are visually exciting. It frequently incorporates aspects of branding and has a high glitz and wow factor. Typical presentation and branding techniques include animation, colorful 3D networks, and visualization that have a high "wow" factors.

Figures 22.1 and 22.2 show two visually exciting examples of network visualizations for presentation and branding. The visualization on the left shows worldwide Internet traffic and the image on the right shows Internet traffic betwen countries. These images have been used on the covers of multiple books, magazines, and as raw material for art work. Their use is really for branding. See, for example, Praba Pilar's network visualization art gallery [Pil05].

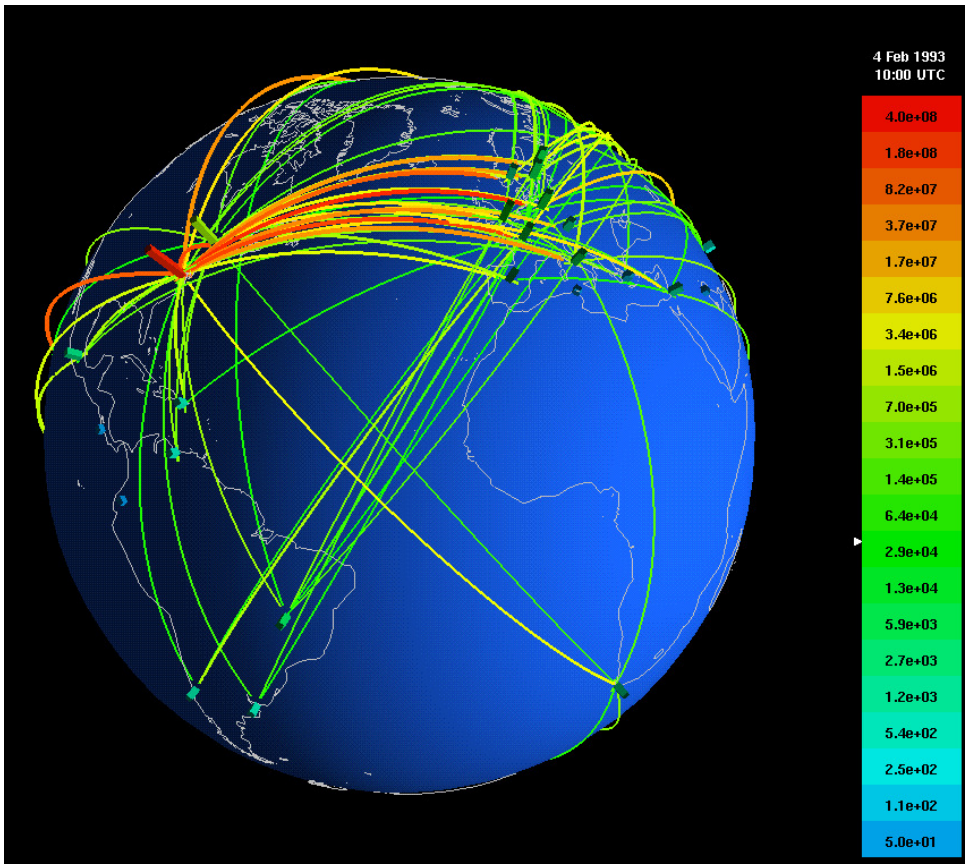

Figure 22.1 3D Internet Network visualizations for presentation and branding.

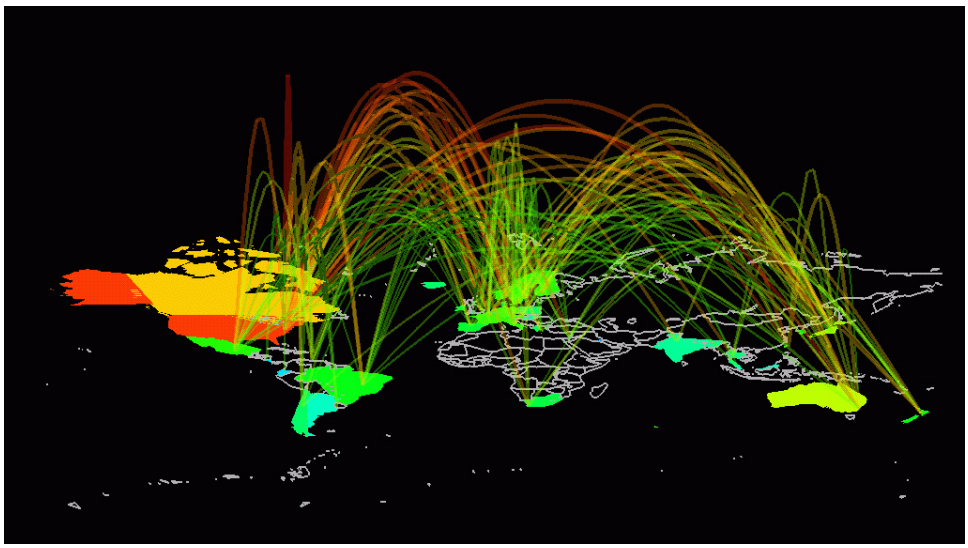

Figure 22.2 Internet traffic flows between countries used for presentation and branding.

22.2.3 Executive Dashboards

Executive dashboards provide decision-makers with instant access to key metrics that are relevant for particular tasks. Much of the intellectual content in dashboards is in the choices of metrics, organization of information on the screen, and access to supporting, more detailed information. Network visualization techniques can improve this presentation, as shown in Figure 22.3. Executive dashboards may include the ability to export result-sets to other tools for deeper analysis.

State-of-the-art implementations of active executive dashboards are web-based, interactive, dynamic, involve no client-side software to install, and often include *action alerts* that fire when pre-defined events occur. End user customizations include *sorting*, *subsetting*, *rearranging layouts* on the screen, and the ability to *include* or *exclude* various metrics. It is common for visual reports to be distributed via email, published on a corporate intranet, or distributed through the internet.

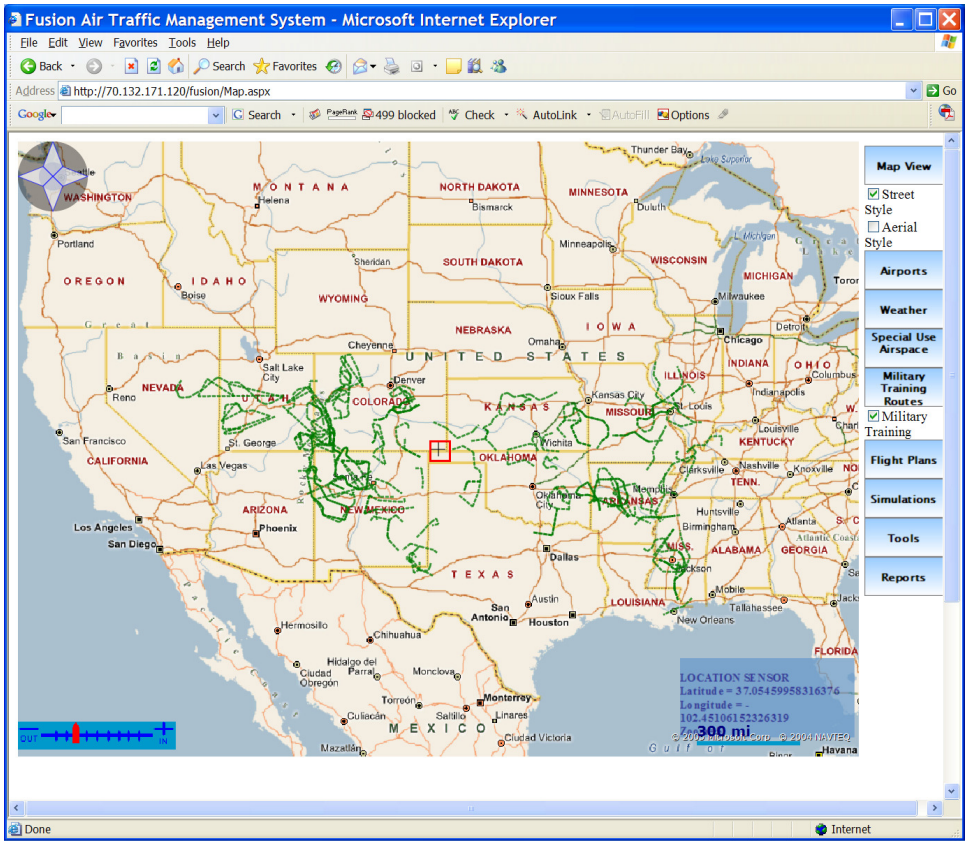

Figure 22.3 Executive dashboard showing a network superimposed on a geospatial map.

22.2.4 Real-Time Visual Reports

Real-time visual reports are related to executive dashboards but provide an active presentation of an information set consumable at a glance. Although the distinction is subtle, visual reports usually involve fat client-side software and thus can provide richer presentations of the information. Visual reports exploit the idea that a picture is worth a thousand words and, in particular, for many tasks a picture is more useful than a large table of numbers.

Visual reporting systems are:

1. Easy to use for both sophisticated and non-sophisticated user communities;
2. Suitable for broad deployments; and
3. Provide capabilities for flexible customization;

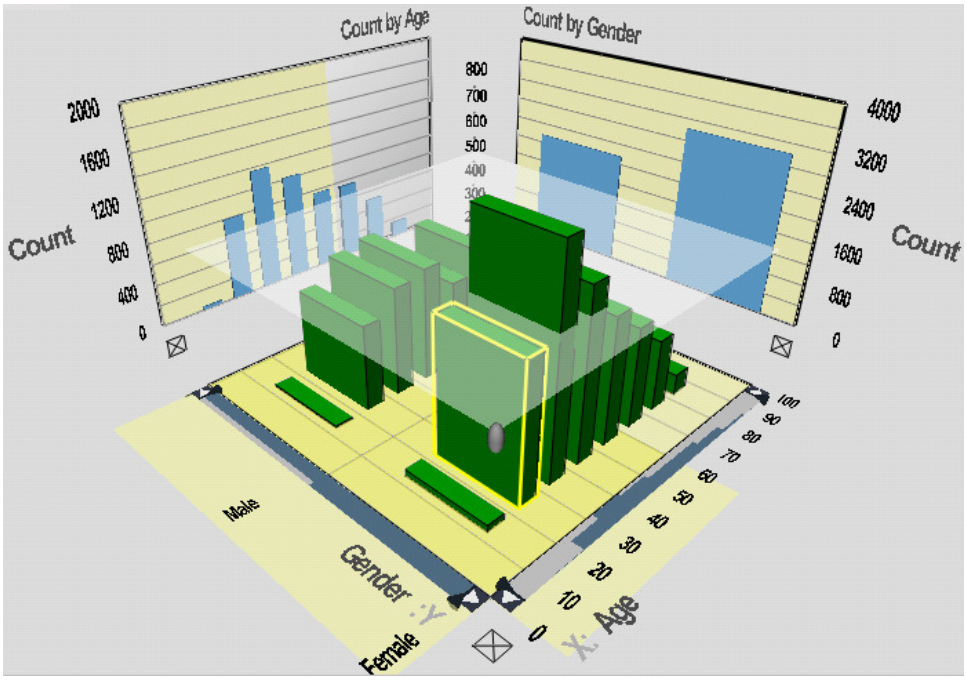

Figure 22.4 Real-time 3D visual report.

Visual reports, as with all reports, are a tool for *assumptive-based* analysis. Reports answer "point questions": How much of a particular item is in stock? Where is it? How long will it take to get more? Reports are ideal for operational tasks, but do not provide full analytics, or enable an analyst to automatically discover new information that a user has not thought to ask about.

This is a well-known characteristic of all report-based analytical solutions. The reports pre-assume relationships that are reported upon. The difficulty with this approach is that most environments are too complex for a pre-defined report or query to be exactly right. The important issues will undoubtedly be slightly, but significantly different. This is particularly true for complex, turbulent, environments where the future is uncertain. There are two common solutions to this problem. The first is to create literally hundreds of reports

that are distributed out to an organization, either using a push distribution mechanism such as email or a pull mechanism involving a web-based interface. The second involves adding a rich customization capability to the reporting interface that increase UI complexity. Unfortunately, neither works particularly well. Although a report containing novel information might exist, finding it is like finding a needle in a haystack. Adding UI features makes the reporting system difficult to use for non specialists.

22.2.5 Visual Discovery for Deep Analysis

Visual discovery-based analysis addresses the shortcomings of assumptive-based analytics by providing a rich environment to support novel discovery. Systems supporting visual discovery are used by analysts and frequently combine data mining, aspects of statistics, and also predictive analytics. Visual discovery is domain specific and iterative. Network visualization improves visual discovery by enabling discoveries to often "jump" out and may lead to "why" questions. For example, in a supply chain management analysis, visual discovery might identify an unusual inventory condition that would lead to a subsequent investigation into why it occurred and how to fix it.

NicheWorks and its successor StarGraph are examples of general purpose information visualization system for visual discovery [Eic00] (see Figure 22.5).

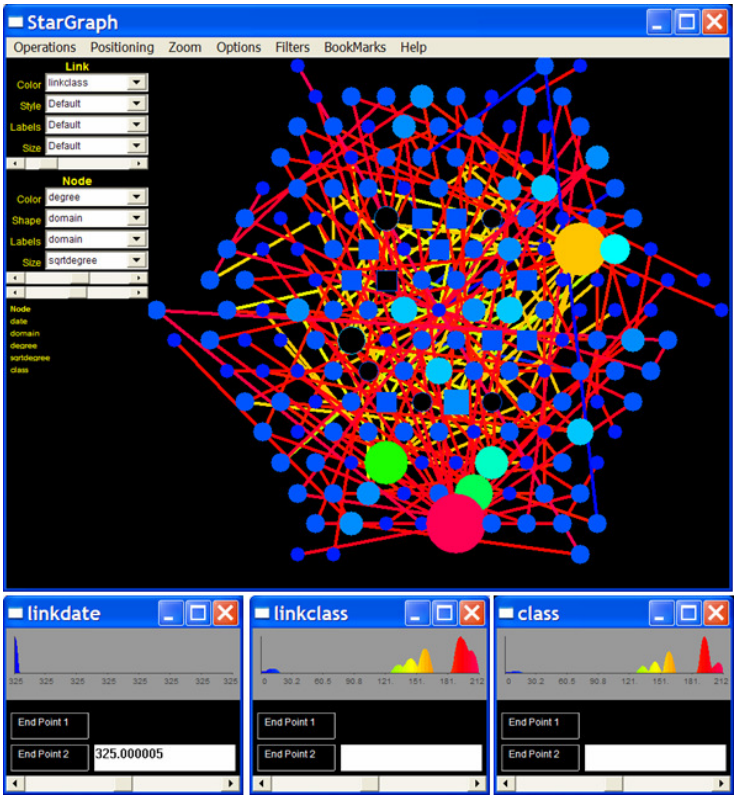

Figure 22.5 Network visual discovery and analysis tool. The linked histograms function as interactive filters to control display complexity.

It consisted of a workspace with standard data acquisition capabilities, and a set of visual metaphors, e.g., views, each of which showed data in a particular way. Some of the views were conventional, (e.g., geographical networks, abstract networks, barcharts, linecharts, piecharts) and some were novel (Data Constellations, Multiscape, Data Sheet). For visual analysis, the views could be combined into fixed arrangements called perspectives. Within any perspective the views could be linked in four ways: by *color*, *focus*, *selection* and *exclusion*. Components linked by color used common color scales and those linked by *focus*, *selection* and *exclusion* were tied by data table row state using a *case-based* model [EW95].

There are three important ideas in this general class of visual discovery and analysis tools. First, *perspectives* extend general linked view analysis systems by reducing complexity for non-expert users. Perspectives are "authored" by "power users" who are experts. Analysts who are domain experts, but not power users, use the perspectives as a starting point for analysis and as a guiding framework. The output from their analysis, visual reports, may be published and distributed for use by casual users, executives, and decision-makers. The user model is similar to that employed by spreadsheets where there are spreadsheet authors, users, and consumers.

Second, *visual design patterns* are recurring patterns within perspectives that are broadly useful and apply to many similar problems. Following the object-oriented programming community [GHJV95], recognizing, cataloging, and reusing design patterns have the potential for significantly improving network visualizations.

Examples of design patterns are Shneiderman's *information-seeking mantra*:

> *overview first, zoom and filter, then details on demand* [CMS99]

The *overview* shows the entire dataset, e.g., all movies in the dataset, and supports the ability to *zoom* in on interesting movies and query the display with the mouse to extract additional details. This design pattern incorporates interactive filters, frequently bar and pie charts, that enable you to *filter* out uninteresting folders so that you display only the data that is interesting. Filtering might be by category, numeric range, or even selected value.

Another design pattern, called *linked bar charts*, is particularly strong for data tables containing categorical data. Categorical data, sometimes called contingency tables, involves counts of the number of data items organized in various bins or subcategories. This design pattern employs one bar plot for each categorical column with the height of the bar tied to the number of rows having that particular value. In statistical terms each of the bar charts shows a marginal distribution. As the user selects an individual bar, the display recalculates to show one-way interactions. Using exclusion and selection shows two-way interactions.

Third, *details on demand* is a feature set where the system provides tooltips and other details when the user mouses over any particular item on the screen. The idea is to provide immediate access to fine-grain information when it is needed without unnecessarily cluttering the interface.

22.2.6 Searching and Exploration

Network visualizations focused on *visual searching* involves undirected *knowledge discovery* against massive quantities of uncategorized, heterogeneous relationship data with varying complexity. This scenario is typical of web searching where users recognize information when they find it. Searches are iterative, intuitive, and involve successive refinements.

The key measures for the performance visual searching systems revolve around the amount of information per unit of search effort expended. The search effort may be measured in user time, number searches, personal energy, etc. The results, or information found, may

be measured in articles, references, relevance, novelty, ease of understanding, etc. Different systems exploit various design points trading off these factors.

22.2.7 Domain Task-Specific Visualizations

Task-specific visualizations help users solve critical, high-value tasks. Examples include visualizations to:

1. Design and layout complex circuits;
2. Identify relationships in product purchases;
3. Trace calling patterns among subscribers (Figures 22.6 and 22.7);
4. Manage huge communications networks (Figure 22.8); and
5. Study relationships in a complex social network.

These visualizations are tuned to particular problems often delivered as part of a complex system. They are highly valuable, frequently involve fusing of a large number of information streams, and serve both as an output presentation for information display and also control panel and input interface for user operations.

22.3 Network Visualization Sweet Spot

One very simply way to characterize network visualization problems uses three dimensions:

1. *Dataset size* is a measure of the total amount of data to be analyzed. Although some might disagree, sophisticated network visualization techniques are not needed for small datasets containing tens of observations. In these cases reports, spreadsheet graphics, and standard techniques work fine. More powerful techniques are unneeded.
   Conversely, network visualization techniques do not scale to analyze massive datasets containing gigabytes of information. The basic problem is that network visualization is a technique that makes human analysts more efficient and human scalability is quite limited. The exact scalability limits of network visualization are subject to debate and are an active research area [EK02, Eic04]. Most researchers would agree, however, that massive datasets containing hundreds of thousands to millions of observations are too big and need to be subdivided, aggregated, or in some way reduced before the information can be presented visually. Network visualization, it would seem, cannot be applied to analyze massive image databases containing millions of images, but might be applied to meta data associated with the images.
2. *Dataset complexity* can be measured by the number of dimensions, structure, or richness of the data. Network visualizations are not needed for (even large) simple datasets with low-dimensional complexity. Statistical reduction tools such as regression work fine and are sufficient in this situation.
   Conversely, datasets of massive complexity containing thousands of dimensions are too complex for humans and thus for network visualizations. Some have argued that network visualizations can cope with as many as fifty dimensions, although a more practical upper limit is say half to a dozen dimensions.
3. *Dataset change rate* is a measure of how frequently the underlying problem changes. Static problems, even for very complex problems, can eventually be solved by developing an algorithmic solution. The algorithmic solution has a

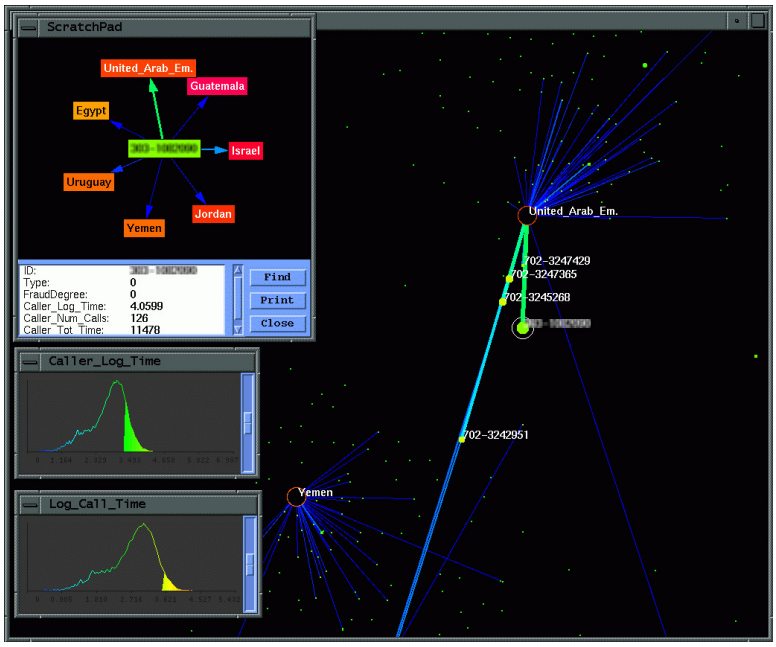

Figure 22.6 Calling patterns among subscribers in a massive international network.

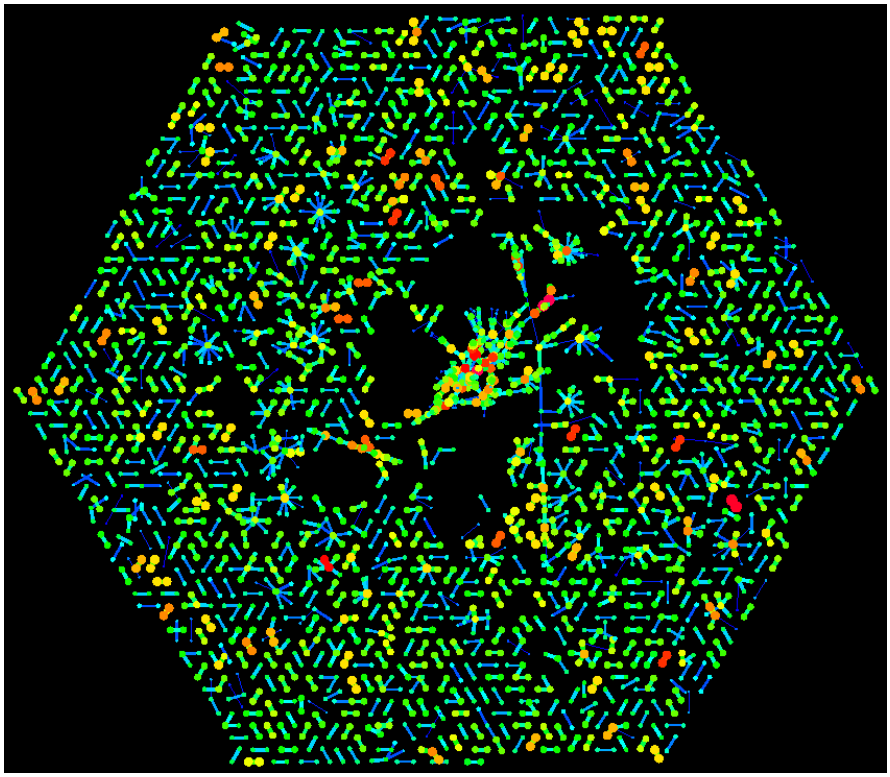

Figure 22.7 Relationship and calling patterns among subscribers in a network.

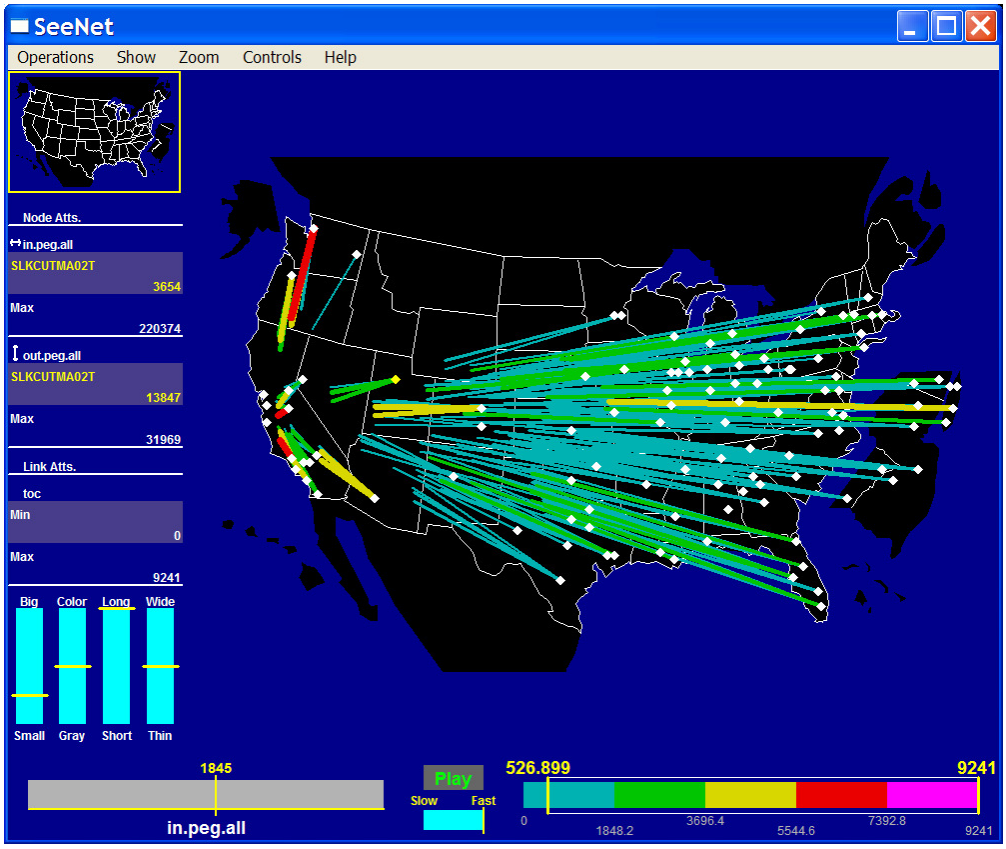

Figure 22.8 Network visualization showing traffic patterns after California earthquake.

huge advantage over an information visualization-based solution since the algorithm can be applied repeatedly without the need for expensive human analysts. Conversely, analysis problems involving change or other dynamic characteristics are extremely difficult to automate because the problem keeps moving. In these cases, human insight is essential. Humans, however, cannot cope with problems that change too quickly. We are incapable of instantaneous responses. Human analytical problem solving occurs on a time scale of minutes to months. We must automate problems needing faster response and partition problem those involving longer time scales.

As shown in Table 22.1 the application sweet spot for network visualization involves analysis problems of moderate data sizes, rich, but not overwhelming, dimensional structure, that change, are not easily automated, or for some reason need human involvement.

Examples of prototypical applications include:

- *Network management*for complex networks where the system is dynamic, constantly changing with new protocols, new devices, and new applications. The systems are instrumented and collect alarms with complex dimensional structure. It is frequently the case that the number of events (alarms) exceeds the capacity of network visualizations and must be algorithmically reduced.

Attribute	Low Value	High Value
Dataset size	10^1 to 10^2	10^4 to 10^6
Dataset complexity	2 or 3 dimensions	50 dimensions
Dataset change rate	minutes	months

Table 22.1 Dimensions and bounding ranges for network visualization sweet spot.

- *Customer behavior* involving human buying patterns and transaction analysis is an ideal candidate for network visualizations. Human behavior is complex, unpredictable, and dynamic. Furthermore, although aggregate numbers of transactions are large, for any individual or set of individuals the numbers of transactions are not overwhelming and easily suitable for analysis.
- *Intelligence analysis* is an ideal candidate for network visualization. It is difficult to automate, involves complex dimensional data, is dynamic, and necessarily involves human analysts.

22.4 Customers for Network Visualization Software

There are three broad classes of potential network visualization users: *scientists*, *analysts* (including both intelligence and commercial analysts), and *business users*.

- *Scientists* have deep needs for network visualization, are extremely technical, and work on the most significant problems. They want powerful tools for cutting-edge analyses.
- *Analysts*, particularly in commercial companies, also have a strong need for network visualization, but tend to have specialized needs. They are not as sophisticated as scientists and will not tolerate raw software packages.
- *Business users* need simple network visualizations and are easily frustrated by complex software. Business users are numerous, have budget, but need solutions to problems and are not inherently interested in the complexly that excites scientists and analysts.

These three classes of users have different needs and varying tolerances for complex software. As a scientists and analysts want complex rich software that is full featured. Scientists are often willing to use flaky software that is cutting edge and incorporates the latest features. However, there are not many scientists and analysts, and they tend not to have large budgets. Thus the addressable market is not particularly large. Business users, however, have budget, have problems, but do not have patience for leading-edge software that is not robust. The business challenge is to create software that is sophisticated enough to solve scientific problems and yet easy to use for business users. These dynamics shape the market for network visualziation software.

22.5 Business Models for Network Visualization

Successfully deploying network visualizations involves solving a technical problem and creating a business model that supports widespread utilization. Broadly speaking, there are several classes of business models for software companies, as discussed below.

22.5.1 Custom Software

Custom software is written to solve a specific problem, usually for a single customer. The problem being addressed must be significant, valuable, important, and yet specialized enough so that general solutions do not exist. The projects often involve next generation technology and new approaches to problems.

Typical price points for custom software projects usually start at $250K. Custom software is sold directly by the vendor with six months to two year sales cycle. The sales team is highly specialized and the sales process frequently involves company executives.

Organizations involved with customer software include universities, government labs, large commercial organizations, and boutique specialty shops. Although it might seem surprising to some, research universities and government labs act as custom software developers where the funding agencies effectively hire university principal investigators using BAAs and solicitations to solve important custom problems. In this setting, the principal investigators function as both sales professionals and also lead fulfillment efforts with "graduate student" development teams.

In the large organizations that sponsor customer software development there are commonly multiple roles. It is often the case, particularly with government-sponsored projects, that the funding organization is not the organization that will eventually use the software and the users of the software may not receive the value from its use. These separate organizational roles complicate the software sales process. For example, the National Science Foundation funds research to build software for scientists to use. The scientists use the software to solve important national problems. Thus, citizens are the ultimate beneficiary. In the commercial environment, the CFO (Chief Financial Officer) funds a project that is implemented by the CIO (Chief Information Officer) for a business unit. Thus, three organizations are involved.

22.5.2 Enterprise Software

Enterprise software, sold by commercial companies, is essentially a flexible template that is "implemented" on site, either by the vendor or a "business partner." In the implementation phase, the template is customized for a particular customer by means of tasks that include connecting up data sources, defining the specific reports a company needs, and populating tables (e.g., inserting employee names into a payroll file). For an enterprise application, data integration is essential. Since enterprise software is reusable, it can be sold more economically than custom software. Generally price points for enterprise software range from $25K to $250K. The sales model for enterprise software may be direct at the higher price points, e.g., SAP, or through local business partners who are "certified" by the vendor.

22.5.3 Shrink-Wrapped Software

Shrink-wrapped software is highly functional software that solves a specific problem very well. The software usually is customer installed and provides for little or no customization. Customer support, if provided, is usually self-serve via a web site or perhaps with limited help desk support.

Shrink-wrap software is almost always sold by distributors or OEMed to the hardware vendors and sold as part of a bundle. For example, Microsoft, the largest producer of shrink-wrap software, sells essentially all of its software through distributors. As a mass-market item, the price point for shrink-wrap software is less than $25K and more frequently less than $1K.

22.5.4 Open Source Software

Open source network visuazliation software supported by services is one of the newer emerging software business models. In the open source model software is developed by volunteers working on donated time and made available at no charge through the internet. However, support and other customizations are offered as service by companies using the open source model. For operating systems and major applications this model appears viable. It is too soon, however, to predict how well the open source model will do for targeted applications and specialized technologies such as network visualization software. In general, the open source experience for visualization software has been mixed. There have been a few successes but many other projects have not gotten widespread traction.

22.5.5 Cloud Computing

Cloud computing solutions via Web portals and network visualization services are another possible business and distribution model for network visualization software. There is a strong push in corporations away from client software because of its high cost of ownership. As a result an increasing number of applications are moving toward a cloud computing model.

22.5.6 Network Visualization Deployments

Relating the business models back to the visualization deployments, most of the demand for network visualization has been met with custom research software built by universities, government labs, and large communications companies. The customers are the military, intelligence community, biomedical researchers, and other highly specialized users. Demand for network visualization within the research community is healthy.

Within the enterprise category we might expect network visualization-enabled applications to emerge. In this category the value is provided by the whole application and a network visualization presentation layer could be described as a software feature or add-on product.

In a related field, Business Intelligence, there have been some early successes for visualization-enabled applications. Perhaps the most notable success has been Cognos Visualizer. Cognos sold 300K[1] units of Cognos Visualizer, an add-on for Cognos PowerPlay, at $695 per unit and some of the other "Business Intelligence" software vendors have had similar experiences.

The "Gorilla" analytic application within shrink-wrap category for network visualization is Microsoft Visio. It is generally considered to be good enough for 90% of problems and essentially everybody has it.

22.6 Thin-client Network Visualization

One of the challenges in creating a successful business model for network visualization software involves deployment. The problem is that rich network visualization clients run on desktop machines which means that they are deployed and managed through IT organizations for many institutions. The cost of maintaining and deploying desktop software

[1]For comparison, a software application that sold 2,000 to 5,000 units would generally be considered successful.

restricts their use and potential application to all but the most important problems. One way around this involves web-based deployment.

The advantage of web-based interfaces is that they are simple to deploy, have proliferated rapidly, and are quickly becoming the de facto standard for accessing information. The disadvantage of web-based interfaces is that desktop applications have a richness and responsiveness that has not been possible on the web. Recently, several new web applications have appeared that provide a rich user experience on the web that previously was only available in desktop applications. Examples include Google Maps, Microsoft Virtual Earth, and Google Suggest. The applications are examples of a new approach to web development that combines Asynchronous JavaScript, XML, and DHTML and represents a fundamental shift in what is possible on the web. On Microsoft's Virtual Earth, for example, you use your cursor to grab the map and scroll it around. On Google Suggest the system automatically attempts to complete your search query. This all occurs almost instantly, without waiting for pages to reload. This programming paradigm is often called Web 2.0 or *AJAX* in the popular press.

There are several technologies, each flourishing in its own right, that can be combined in powerful new ways to create the next generation web-based visualization capability. These are:

- Standards-based presentation using XHTML and CSS;
- Web-based 2D graphics using Scalar Vector Graphics (SVG);
- Dynamic display and interaction using JavaScript to manipulate the Document Object Model (DOM);
- Data interchange and manipulation using XML and XSLT;
- Asynchronous data retrieval using JavaScript's XMLHttpRequest;
- DHTML (JavaScript) binding everything together.

Traditional web applications work on a client-server model. The client, a web browser, issues an http request to a server for a new page when the user clicks on a link. The web server, usually Apache or IIS, does some processing, retrieves information from legacy systems, does some crunching, and sends a formatted page of hypertext back to the client for display. This approach is the simplest technically, but does not make much sense from the user perspective. The problem is that the user waits while the server does its thing for the next page to reload.

The new model enabled by these new technologies eliminates start-stop-start-stop nature of web applications. Instead, information is asynchronously downloaded to the client in using XML. JavaScript code in the browser caches this information when it is received from the server and displays it upon user request. Since the information is cache, the system can provide instantaneous responses. JavaScript code in the browser also handles other interactions with the user such as panning, zooming, scaling, and data validation. The advantage of the asynchronous requests for XML data is that users can continue working with the application while data is downloading.

The application shown in Figure 22.3 is an example of a thin-client interactive geospatial network visualization. It is written using SVG and interaction is done via manipulating each page's DOM. Although the programming is quite difficult, the result is stunning. It is able to provide the richness of a desktop application without the hassles of desktop software with the flexibility and rich hyperlinking that is only possible in browsers.

22.7 Discussion and Summary

This chapter attempts to define opportunities where network visualizations create significant business value. Network visualization involves the presentation layer with is naturally a feature of many products. By itself, it usually has insufficient value to support widespread usage and deployment. It is generally a feature of an application and a critical component of a solution.

The chapter identifies various types of network applications and develops a simple model that characterizes an opportunity space for an application. The target for the model is to help identify commercial opportunities network visualization applications.

Although this chapter is not expected to be particularly exciting for researchers or scientists who are interested in pushing the state-of-the-art, it should help practitioners identify opportunities for successful applications. Unfortunately, business and other pragmatic issues often dominate the technical issues when it comes to determining which network visualization applications will achieve commercial success.

References

[CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualiation: Using Vision to Think*. Morgan Kaufman, San Francisco, California, 1999.

[Eic00] Stephen G. Eick. Visual discovery and analysis. *IEEE Transactions on Computer Graphics and Visualization*, 6(1):44–59, January–March 2000.

[Eic04] Stephen G. Eick. Scalable network visualization. In Christopher R. Johnson and Charles D. Hansen, editors, *Visualization Handbook*, pages 819–831. Academic Press, 2004.

[EK02] Stephen G. Eick and Alan F. Karr. Visual scalability. *Journal of Computational Graphics and Statistics*, 11(1):22–43, March 2002.

[EW95] Stephen G. Eick and Graham J. Wills. High interaction graphics. *European Journal of Operational Research*, 81:445–459, 1995.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, 1995.

[Pil05] Monica Praba Pilar. *Cyber.Labia: Gendered Thoughts and Conversations on Cyber Space*. Tela Press, 2005. Available at http://www.prabapilar.com/pages/projects/cyberlabia.htm.