

Progressive Multiples for Communication-Minded Visualization

Doantam Phan, Andreas Paepcke, and Terry Winograd

dphan@cs.stanford.edu, paepcke@cs.stanford.edu, winograd@cs.stanford.edu

Stanford University

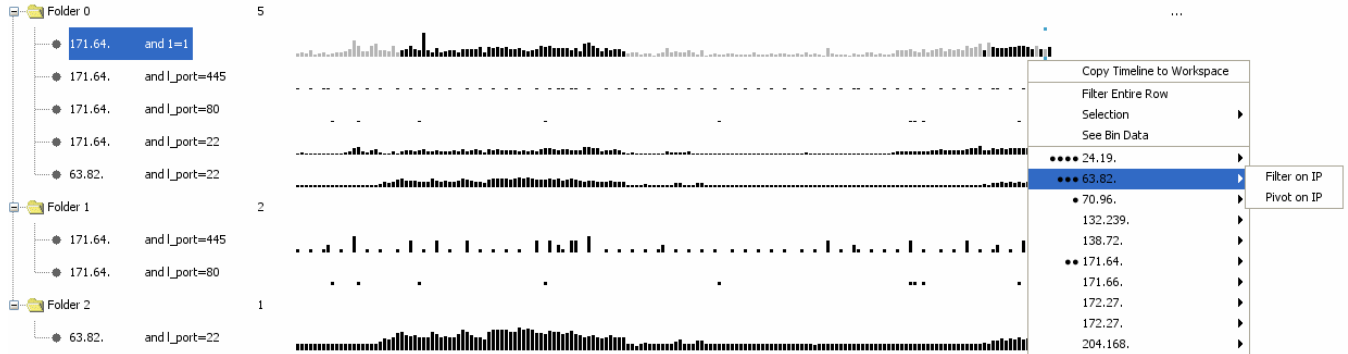


Figure 1. Exploring event data using progressive multiples allows users to interact with a timeline on one row to produce timelines on new rows. Rows are added to the display in sequence and serve as a history of a user’s exploration. Since the timelines are aligned, users can create a different row for each filter and juxtapose the rows to compare the filters. Users can place timelines in folders to group related data. Relationships are defined among events in the time-series with similar metadata. Pivoting on an event creates a new timeline containing the related events. Users can create a visual narrative that illustrates a sequence of interesting events by hiding rows and reordering other rows. In this example we show an exploration of time-series network data where the user has filtered the 171.64 address by port 445, 80, and 22, and then pivoted to the 63.82 address. For privacy purposes, the IP addresses have been anonymized.

ABSTRACT

This paper describes a communication-minded visualization called *progressive multiples* that supports both the forensic analysis and presentation of multidimensional event data. We combine ideas from *progressive disclosure*, which reveals data to the user on demand, and *small multiples* [21], which allows users to compare many images at once. Sets of events are visualized as timelines. Events are placed in temporal order on the x-axis, and a scalar dimension of the data is mapped to the y-axis. To support forensic analysis, users can pivot from an event in an existing timeline to create a new timeline of related events. The timelines serve as an exploration history, which has two benefits. First, this exploration history allows users to backtrack and explore multiple paths. Second, once a user has concluded an analysis, these timelines serve as the raw visual material for composing a story about the analysis. A narrative that conveys the analytical result can be created for a third party by copying and reordering timelines from the history. Our work is motivated by working with network security administrators and researchers in political communication. We describe a prototype that we are deploying with administrators and the results of a user study where we applied our technique to the visualization of a simulated epidemic.

CR Categories: H.5.2 [Information interfaces and presentation]: User interfaces. – Graphics user interfaces.

Keywords: Small multiples, time-series data, permutation matrices, communication-minded visualization

1 INTRODUCTION

Information visualization often focuses either on exploratory data analysis [22], where the goal is to interactively discover underlying structure in the data by mapping different dimensions of the data to visual patterns, or on methods of data presentation. We are researching visualizations that bridge the task of exploratory data analysis and the task of presenting analysis results. In the process of developing forensic visualizations for network security administrators, we have observed that their workflow does not end with the moment the analyst gains an insight. Often the result of their analysis must be conveyed to an interested third party. A visualization that conveys this analysis has been termed a communication-minded visualization (CMV) by Viégas and Wattenberg [23].

We are working with network administrators who conduct forensic analyses after-the-fact on how a computer on their network was compromised, based on event logs of the connections made between computers. Administrators need to inspect the traffic of multiple machines, as any other machine that connected to the compromised machine might be suspect. Since a machine in our lab has thousands of connections in one day, there can be a large amount of data. Once the analysis is completed, they need to use their analysis of the break-in to convince other groups to take action towards shutting down the offending system or to prepare a report that will serve as evidence for an investigation by law enforcement.

The administrators we are working with have access to a database that records information about the connections among computers on the network. Our logging system records events at the flow level. At that level, information is tracked about which systems communicate with one another, rather than at the packet level, which would record the contents of those conversations.

This is done in order to keep the size of the data manageable, and allows our system to maintain records over longer time periods. An event consists of a connection between a local and remote IP, on a given local and a remote port, at a start time, a given duration, and the number of bytes transferred. Currently administrators in our lab query the database contents directly using a textual SQL interface or use Tableau (formerly Polaris) [20] to visualize the connections among computers on the network.

Tableau allows users to visualize databases using a Pivot Table interface. Users can map different dimensions to the x and y axes, refine a query graphically, or drill down upon data. However, it is difficult to perform forensic analysis in Tableau. If an administrator is viewing computer A's communications, and finds a suspicious connection between computer A and B, they must issue a new query where B is the focus. Tableau does not have a facility to pivot on the data in this way. Furthermore, Tableau does not allow users to easily compare multiple visualizations without opening multiple windows, because changing the parameters of Tableau results in a new visualization that replaces the old one on the screen.

The need to navigate among related events is not unique to the domain of network security. We are also working with researchers who are studying political communication and mass media effects. They want to understand how the language of news coverage evolves over time. This can be measured by tracking the propagation of articles across news sources to understand the rise and fall of concepts over time. The researchers also need the ability to present the results of their analysis. Currently they are studying the web-based coverage of Hurricane Katrina and the California special election of 2005. The California special election data set spans a month-long period leading up to the election with approximately 2,000 articles over 15 websites. The Hurricane Katrina dataset covers a two-month period over 400 web sites. Although this paper will focus on the prototype we've created for network security administrators, we will continue to use examples from political communications to motivate the design of our technique.

This paper describes a communication-minded visualization for the forensic analysis and presentation of multidimensional event data which we call *progressive multiples*. We combine ideas from *progressive disclosure*, which reveals data to the user on demand, and *small multiples* [21], which allows users to compare many images at once. Sets of events are visualized as timelines, which are a natural choice for presenting temporal patterns. Events are placed in temporal order on the x-axis, and a scalar dimension of the data is mapped to the y-axis. In support of forensic analysis, we allow users to pivot from an event in an existing timeline to a set of related events, which appears as a new timeline.

The key idea is that a user controls which timelines are added to the display by interacting with existing timelines and selecting the event to pivot on. This provides the user with an exploration history, which has two benefits. First, this exploration history allows users to backtrack and explore multiple paths. Second, once a user has concluded an analysis, these timelines serve as the raw visual material for composing a story about the analysis. A narrative that conveys the analytical result can be created for a third party by copying and reordering timelines from the history.

Allowing users to interact with small multiples poses many research challenges. Relationships among events need to be evident. Although timelines make temporal relationships easier to see, they do not lend themselves to revealing other types of structure, such as geospatial relationships or graph connectivity. Users also have a limited working memory and need to manage

large numbers of rows. Finally, users need to recall meaningful points in their exploration so that they can construct a narrative that supports their analysis.

We describe the tasks that are common to forensic analysis for network security administrators and political communication researchers. We present a prototype for visualizing network data that was developed in consultation with network administrators. To evaluate the specific interface features of our prototype, we created a variant of our prototype that visualizes the spread of a simulated epidemic. We describe the results of a user study where we asked users to locate the source of the epidemic using our tool. Based on this study and our observation of the prototype's usage by network administrators, we describe lessons for the ongoing design of a system that addresses the shortcomings of the prototype.

2 FORENSIC ANALYSIS TASKS

In forensic analysis, a user is looking for temporal patterns in multi-dimensional event data. The following kinds of tasks must be supported:

Selection and categorization. Users must be able to select subsets of their datasets and understand their composition. Network security users want to see the traffic for a given time period and filter connections by the ports used or classify machines into groups with traffic above or below normal activity levels. Communications researchers want to see articles for a given time period and then classify articles by the appearance or absence of phrases that were used in the articles.

Revealing connections among events. Once events are categorized, users want to look for temporal trends in the data by finding events that are related to the event they are currently inspecting. Events with similar metadata are considered to be related. For example, in the network security data set, events may be related if they involve the same IP address, use the same port, or occur near the same time. For news stories, events may be related if they share similar phrases or were published on the same day or by the same newspaper.

Comparison of temporal patterns. Once users discover an interesting sequence of events, they want to compare it to other sequences. They might want to see if one sequence precedes or follows another, or they might want to compare it to another time period to see whether the behavior is periodic or unique. For example, network administrators might want to know if the network usage they see this Friday at 10pm is similar to last Friday at 10pm. Is the activity a backup or something malicious? Communications researchers might want to know if two papers that used the same phrase for one story did so for a different story.

Narrative Construction. Once users find a set of interesting patterns, they need to present this information to third parties. They often start with snapshots of visualizations they have found in the course of their exploration. Users need to be able to edit and summarize these visualizations to communicate effectively. Network security administrators might want to show the sequence of steps an attacker took to compromise a machine, perhaps by cracking a weak password, and then going back and controlling the machine via Internet Relay Chat (IRC). Communications researchers might want to show the leader/laggard phenomenon, in which newspapers such as the New York Times or Washington Post "break" a story, which is then picked up by less prominent newspapers.

3 NETWORK SECURITY PROTOTYPE

Our prototype was designed in consultation with network security administrators, as part of a broader project on security

visualizations. Our initial designs made use of various node-link representations for networks [5, 7, 15, 16]. Although these graphs effectively represented topology, they did not provide a good way of conducting forensic analysis because it was difficult to understand temporal relations among nodes. We mapped computers to nodes, and communications between computers to edges. However, it was difficult to understand when events occurred because the edge only indicated the presence of communication but not when it occurred.

We have adapted the small multiples technique, which allows users to compare objects by juxtaposing multiple related images. Small multiples are most effective when the designer of the visualization understands the structure of the data and chooses the correct dimensions to present to the viewer. In this case, however, there is no way for the system to automatically discern the structure of the data, given the exploratory nature of the forensic analysis task. Instead of presenting all the data at once, our system allows the user to control which small multiples are made visible. The user is not overloaded by too many rows, and can perform comparisons across the small multiples that are present.

Figure 1 shows an example of our network exploration prototype. Timelines are added to the interface in the order that the user asks for them, either by *querying* to create a new timeline, by *filtering* an existing timeline to produce a new one, or by *pivoting* on an event in an existing timeline to produce a timeline containing a set of related events.

3.1 Querying

When an administrator starts the prototype, our assumption is that the user is investigating the traffic of a specific computer. The administrator is first presented with a dialog box, as seen in Figure 2, which is used to specify the IP address and the time period of interest. The user also specifies the aggregation query and the filter string. Once the user fills in all the options, the system translates the input into a SQL query which returns a set of events, and adds a timeline to the current folder.

For the network security domain, an event is considered to be a unique conversation between a pair of computers. Ideally, we would like to map the timestamp of each event to a unique bar onscreen. However, this is usually not possible to do, since a computer may communicate with many other computers over an arbitrary time period. As a result, our prototype must bin multiple events into the same bar. Thus, the height of each bar represents the number of events contained in a given bin. The default aggregation query simply counts the existence of each event, which can be thought of as giving an event a value of 1. This is accomplished by using the SQL statement `count(sequence_id)` as seen in Figure 2.

However, the administrator could change the aggregation query to count some other attribute of the event. The aggregation string is passed directly to the database and can be any valid SQL expression that returns a scalar value in the SELECT clause. This allows the administrator to take advantage of SQL’s expressivity. For example, although overall traffic volume using the query `count(sequence_id)` is an obvious choice for the analyst, it can be useful to use other aggregations. If the analyst is concerned that one of his machines may be infected and attacking others, he could use `count(distinct remote_ip)` to identify local machines with unusually large numbers of remote partners which would indicate that a machine may be scanning the network.

We note that allowing the user to interact with the database directly at such a low-level is only suitable for very SQL-sophisticated users. For domains where the users may be unfamiliar with query languages, we would need to preselect a set of commonly aggregated measures.

Figure 2. The initial query window presented to the network administrator. For a given time period, the user specifies which ip address to look at, how to aggregate the traffic (in this case by counting the sequence id, or the number of flows), and which filter to use (in this case by port 22, or SSH traffic)

The filter string is simply appended to the SQL query as part of the WHERE clause. In Figure 2 we are limiting the events we see to have a `local_port=22`, which simply means the visualization is only showing communication between computers that took place over SSH.

3.2 Filtering

If a query returns too many events, a user can filter a row to reduce the number of events under consideration. This is done by selecting the “Filter Entire Row” menu item. The user is presented with a dialog box similar to Figure 2, where the user can modify the existing filter string. For example, they could break the row down into different components to compare the components with one another and to the original row. In Figure 1, the user has created a series of rows where 171.64 has been filtered on port 445, 22, and 80 in Folder 0. Since the rows are all from the same time period, their vertical alignment allows the user to compare the rows and judge which filter is the most promising avenue for further investigation.

3.3 Pivoting

Whereas filtering reduces the number of events under consideration, pivoting on an event creates a new timeline that contains events that have similar metadata. For the network prototype, this means a new timeline is created where that IP address is the focus. In particular, a similar dialog box to Figure 2 is presented, where the Focus IP field has been changed to be the IP address that was clicked upon. The time, aggregation query, and filter strings are filled in by default to be the same as that of the timeline that was clicked upon, but can be modified by the administrator if desired. In Figure 1, the row containing 63.82 is a pivot from the 171.64 row. The user discovered that 171.64 had communicated with 63.82 in two bursts. To see if any of those connections were suspicious, the user pivoted on each of those addresses. The user can interact with these new rows to further filter and pivot on interesting events.

3.4 Tooltips and Popup Menus

Once a timeline is created, the user can interact with it in several ways. Figure 1 is a screenshot of the prototype we have built for network administrators. The first row represents a query that asked the system to display all the connections that were made to and from the 171.64 machine (addresses have been anonymized for privacy purposes) for a given time period. The height of the bars is a count of the number of events that appear in each bin. When a user moves the cursor over a bar, a tooltip summarizes the

number of events in each bin and the time period represented by the bin.

To obtain more detail, the user can click on a bar, which presents a popup menu listing all of the IP addresses that contacted 171.64 in the time period spanned by that bin. An example of this can be seen in Figure 1. The number of the dots next to each IP address in the popup menu represents the $\log(\text{number of times})$ that connection appeared in the bin. As the user moves the cursor over the elements of the popup menu for 171.64, we use brushing [4] to alter the appearance of the bars to show which other bars contain an event with the same IP address. In Figure 1, the cursor is over the menu item representing 63.82. As a result, only the bins containing 63.82 have been highlighted, and the bins that do not contain it are faded out. This reveals two bursts of activity between 171.64 and 63.82 which the user may want to investigate further.

Finally, since we are working with network administrators who like to be able to see the actual details of the flows, we also allow them to drill down and see the data as a simple table with all the associated information.

3.5 Folders and Workspaces

Folders can be used to organize timelines into different lines of inquiry. By default, all timelines are placed sequentially in the initial folder. A user can always create a new folder and drag timelines to this folder. To support the exploration history, dragging a timeline only copies it, instead of moving it. Users do have the option of deleting a timeline, but we wanted to encourage administrators to keep around history through this interaction.

To facilitate comparisons, the height of a bar is linearly scaled to the height of the tallest bar in the folder. This may cause some problems because when data has a high dynamic range, as in Folder 0, any detail for the timelines filtered on *171.64* and *l_port=445* and *171.64* and *l_port=80* is obscured. This is because they are in the same folder as the timeline for *171.64* and *l=1* (which returns all communication with 171.64). As a result, the user created Folder 1 to compare *171.64* and *l_port=445* and *171.64* and *l_port=80* by themselves. Similarly, Folder 2 was created to place *63.82* and *l_port=22* in a folder by itself so that its pattern of communication could be seen more clearly.

We also allow users to create multiple windows, which can be used to further organize investigation. This allows us to take advantage of the operating system's window management, which can be customized by each network administrator. To support this, the user has the ability to copy timelines to other workspaces. Both folders and workspaces can be renamed by the user.

4 EPIDEMIC VISUALIZATION EXPERIMENT

Currently we are conducting a long-term deployment with the network administrators in our lab to evaluate the ecological validity of our tool. We wanted to get more feedback about specific interface elements of progressive multiples. We also

wanted to extend our technique to the visualization of other kinds of event data. We looked for a different domain that also required the forensic analysis of event data and decided on a simple disease/infection domain. For this domain, we constructed a simulation, which is not intended as a realistic disease model but was designed to create temporal data that could be analyzed by a broad range of users. An advantage of choosing this domain was that we were able to have more people give feedback on interface elements, which is helping us judge the generality of our technique.

The scenario presented to subjects was that they were analysts for the CDC during an epidemic. Subjects were told there was a single carrier and that the length of the incubation period of the disease was fixed, but unknown. The subject's goal was to determine the identity of the carrier by using the interface. Subjects had the ability to "test" a person, which represented sending a CDC team out into the field to verify if the person carried the virus or not. We associated a cost with the test to encourage subjects to use the interface to accumulate enough visual evidence to decide if someone was a carrier, instead of "solving" the problem by testing everyone.

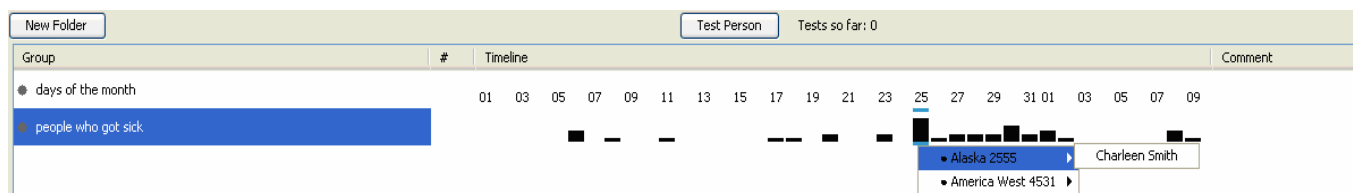
In our study, eight subjects used our prototype on two data sets. Seven of our subjects were engineering graduate students and the eighth was a researcher in computer science. Subjects were asked to follow a talk-aloud protocol as they worked. Each study took between 60 and 90 minutes. We reserved the last 30 minutes for questions to ask about their reactions to different interface components, such as the pivoting interface and the use of folders.

During the study we took notes and asked questions of the subjects about their analysis process. By having subjects talk aloud as they worked, we were able to judge which features of the interface were useful to their analysis. We asked them to justify why they were pivoting on certain people and to explain why they wanted to “test” a particular person as the carrier. We were also able to see if subjects referred to the visualization history to explain their reasoning.

Before we present the results of the study, we discuss the specifics of the simulation and the prototype.

4.1 Simulation Data

The simulation has 14 cities. Each day, some of the people in each city travel to another city on a flight. A single carrier is chosen at random from the population. If the carrier shares a flight with a person, that person is infected with a fixed probability. After becoming infected, people continue to travel and spread the virus with the same probability as the carrier. Once a fixed incubation period passes, infected people manifest symptoms, become sick, and stop traveling. The carrier never gets sick and continues taking flights. We simulated the first 40 days of an epidemic and generated two data sets with different incubation periods and initial populations. The first set had 200 people per city and the second had 300 people per city.



4.2 Visualization Prototype

Each event in the system is a record of each person's apparent health and location for every day. This information is presented in two ways. First, there is an overview timeline which shows when people got sick, as seen in Figure 3. Users may also create the timeline of a person who got sick on a given day, which can be seen in Figure 4. Users can interact with existing timelines to produce new ones.

4.2.1 Exposing Structure in the Data

We provide brushing to reveal underlying structure in the data. For the epidemic task, events are linked if their location is the same. Moving a cursor over a bar will highlight that bar in blue-green. If the cursor is over a person's timeline, the system highlights other bars in other rows that describe the same location as the current bar. For example, in Figure 3, if any other bars contained American Eagle 5992, they would be highlighted.

4.2.2 Managing Rows

We use a hierarchical folder representation similar to that of Windows Explorer. We chose not to allow folders within other folders as we wanted to keep the grouping mechanism simple. Timelines created by the same interaction appear in the same folder so that they are meaningfully grouped. Within folders, users can reorder timelines to facilitate comparisons. Users manage space by deleting rows or expanding and collapsing folders.

4.2.3 History and Narrative

Folders are added sequentially to the display in the order of their creation. Their position is fixed to provide users with a history of exploration. Comments can be added to any row. Users may create empty folders and populate them by dragging timelines from other folders.

4.3 Results of User Observation

Seven of our 8 subjects were able to find the carrier with one or two tests. We ran each subject on the same two data sets. Of the subjects who found the carrier, the first time it often took 30 to 50 minutes to understand the strategy necessary to find the carrier. However, once they understood the strategy, they were able to use the visualization to quickly find the carrier, often on the order of 5 minutes. In and of itself, this does not provide substantive support for the utility of our interface. However, our observation of users' interactions with a set of timelines allowed us to identify issues for improving the design of future systems.

4.3.1 Narrative and History Management

We observed that the user's ability to construct a narrative often depended on their search strategy. Some users were disciplined about their exploration and used a depth-first approach. They would pivot to create a new set of timelines, and then recursively explore those timelines before coming back to the first level.

In the post-task interview, these users explicitly referred to the row history to recall how they had analyzed the epidemic. Figure 5 is an example of a visualization created by one of our users after they had successfully found the carrier. The folder labeled "Story" shows the carrier, Bill Bilodeau's pattern of movement, followed by the first several people he infected. In particular, 5 days before Donny Jacobsen became sick, he took a plane flight, JetBlue 2181 with Bilodeau. Through analysis, the user determined that the infection period was 5 days and that Bill Bilodeau was the carrier.

However, other users employed a breadth-first approach. These users would pivot on multiple events on a single timeline before looking at the results of their pivots. This would add multiple folders of results to the interface. When the user encountered that pivot result much later, they were unable to remember why they had created it. This was exacerbated because users did not usually add comments to the row. Users stated that they had forgotten which object they had interacted with to generate a row. We describe this problem as being unable to determine a row's lineage. To support communication-minded visualization for both search strategies, it is important to develop interaction techniques for making lineage more visible.

When there were a large number of rows, users did not remember having already seen a timeline for a specific person's movements, which led them to repeat pivots and create the same row twice. Users wanted to be able to mark a person with an icon, such as a star, and then have the system "star" other occurrences of that person.

However, users generally liked the row history and the organization imposed by collecting timelines in the same folder. Most users were careful about deleting unnecessary rows and did not run into as many problems as users who expanded many rows. Users found brushing of similar rows to be helpful, although sometimes there were so many rows that the highlighted ones were off-screen.

4.3.2 Feature Requests

Users wanted to sort the timelines in a folder by one of their attributes. For example, some users wanted to sort the rows by location to see who had traveled together, or by the height of the black bars, which counted how many people got sick after being

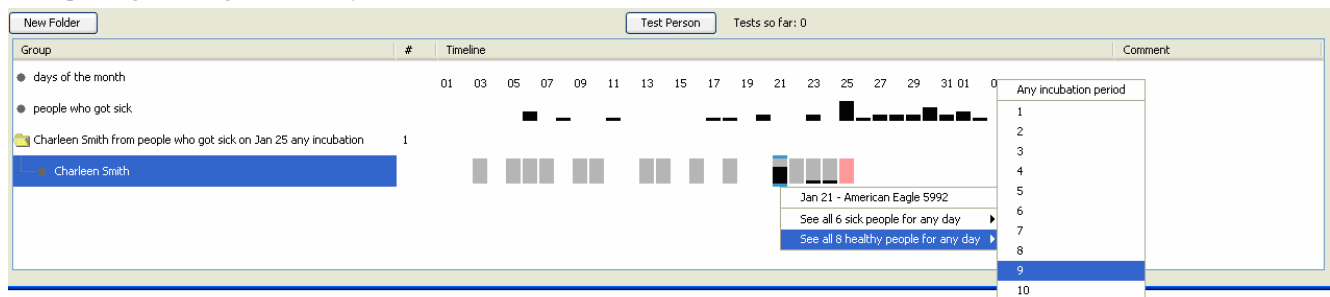


Figure 4. **Timeline for Charleen Smith's Travel on Airplanes.** Gray boxes indicate the days she took a flight and the red box indicates when she got sick. The height of the black bar counts the number of people who traveled with Charleen who later got sick at *any time*. Clicking on a gray box allows users to see the timelines for the other people on the flight who are sick or healthy at the end of the simulation. The lack of black bars until January 21st means that until then, Charleen never traveled with anyone who later got sick. On the 21st, 6 other people got sick after being on a flight with her. Here the user is creating a folder containing timelines for healthy people who traveled with Charleen on American Eagle 5992. Selecting #9 means the black bars of the new timelines will only count the other people who got sick *exactly 9 days* after traveling with the healthy people from American Eagle 5992.

in a particular location.

Users also wanted to operate on sets of folders. For example, they wanted to intersect two folders and store the result as a new folder. A common strategy was to find two flights the carrier must have taken. Users would expand the set of healthy people for each flight, and then manually intersect the contents of the two folders. For folders with 10-20 people, this was tractable, but for larger numbers of people this was impossible.

5 DIRECTIONS FOR FURTHER RESEARCH

5.1 Row (and Screen Space) Management

In our prototype, managing screen space is the means of managing the amount of available external cognition. Although both variants of our prototype allow users to manage rows by deleting (and undeleting) them, subjects from the epidemic experiment indicated a need for better interactive ways to hide rows and unhide rows instead of permanently deleting them.

One solution is allowing users to segment their work into multiple windows. However, this does not address the problem of when too many timelines are created in one window. One solution may be to allow users to archive and label timelines, similar to the metaphor used by gmail to manage email. If we allowed users to assign labels to a timeline or a folder, they could then manage their visibility by setting certain labels to be visible or invisible.

5.2 Revealing Non-Temporal Structure in Event Data

As we have mentioned, although timelines are useful when looking for temporal patterns, it is difficult to see non-temporal structure in event data. By implementing different kinds of brushing, it is possible to reveal non-temporal information. For example, the system could keep track of where users clicked to create a row. In this way, we might show row lineage by drawing an arc between the current row and where it originated.

The advantage of using brushing is that it is transient in nature and does not clutter the display. On the other hand, sometimes



Figure 5. **An example of an analysis performed by a user.** Donny Jacobson was the first person to become sick. The user concluded that the incubation period was 5 days by intersecting the possible set of incubation periods for Pedro, Della, and Harriet. Five days before Donny became sick, he traveled on JetBlue 2181, on January 3rd. The second folder shows the healthy people from that flight. The height of the black bars only counts people who got sick exactly 5 days after traveling with this group. By looking at the black bars and gray boxes for Bill Bilodeau, the user observed that many people became sick 5 days after traveling with him. Other passengers, such as Greg Baron, never flew with anyone who got sick 5 days later. The user verified that Donny had infected Chadwick, but Bill had infected Donny and the other passengers. During the interview, the user created a new folder labeled “Story” to explain his reasoning to the interviewer.

brushing may reveal an interesting pattern, but there is no way to record that in the history. We plan to combine the advantage of transient and permanent display by including brushing-to-folders, which allows users to save highlighted rows into a folder.

5.3 Folder Algebra

For event data, we can formalize the operations that users want to perform on rows, folders and sets of folders into a relational algebra. With folder algebra, it would be easy to sort timelines in a folder by different attributes. Similarly, the union and intersection of folders would be a special case of a join. We are exploring how to expose this functionality to the user in a way that is easy to understand.

5.4 Generalized Pivoting

The network security administrators that we are working with have found the pivoting to be useful for forensic analysis as well as for gaining a better understanding of network behavior. Our tool allows administrators to track a suspicious connection between computer A and B by pivoting to create a new timeline for B. However, they have expressed a desire for a more generic form of pivoting. If computer A is communicating with computer B, they don't want to just pivot to a timeline for computer B. They would like to be able to see all computers from the same autonomous system (ASN) as B. An ASN can be thought of as all the IP addresses under control of a single group. It is usually, but not always, the same as an Internet Service Provider.

One way of supporting more generalized pivoting is to treat the data for every event as a n -dimensional tuple. Pivoting imposes requirements on $n-k$ of the elements while allowing the other k to be wildcards. In the example of Figure 1, the user chose to pivot on a tuple that represented a connection between 171.64 and 63.82 in the time period between t_0 and t_1 . The tuples that appeared in the new row were required to have a $\text{StartTime} \geq t_0$, an $\text{EndTime} \leq t_1$, and have an $\text{IP} = 63.82$. Since every event is a tuple of (local IP, remote IP, time, remote ASN, bytes transferred, local port, remote port) showing computers with the same ASN constrains the new set of events to have the same remote ASN in the same time period, but allowing the remote IP can vary.

5.5 Narrative Construction

Figure 5 has a story folder that shows the progression of the epidemic from an initial carrier, in this case Bill Bilodeau. Although the user was able to describe this narrative to us when we were sitting side-by-side and interacting with the prototype, we noted that a static image of the same set of timelines was unable to convey the narrative. The problem is that the visual cues necessary for showing that certain passengers traveled on the same flight together are conveyed through brushing and tooltips, which are transient and not reproduced when printed to paper. Viégas and Wattenberg [23] have also noted that because many visualizations rely on transient cues for conveying meaning, they are often unsuitable for re-presentation of analysis.

Similarly, although the administrator is currently able to convey an analysis to a third party by demonstrating connections using our tool, our experiment indicated that it is important to be able to create a standalone image that conveys the narrative. This can be done by building an authoring tool into our system. If the authoring tool has access to information like row lineage and other relationships among the rows, it should be easy to create an image with insets and arrows that explain a temporal progression.

6 RELATED WORK

SeeNet [5, 7] allowed users to interactively visualize networks with node-link diagrams. More recently, researchers have tried

alternate methods of visualizing network traffic. Goodall et al. describe TNV [9, 10] which maps each IP address to a row and produces a timeline of activity. Connections between IP addresses are drawn as lines among rows, as opposed to our system, which maps the number of connections for a specific IP address to an event in a bin on a single timeline. Abdullah [1] describes a system that visualizes port activity over time using stacked histograms. IDGraphs [19] maps combinations of source and destination IP and port to a plot, where the y-axis is the number of unsuccessful connections and time is on the x-axis. Our technique differs from these others with our emphasis on interacting with progression of views instead of using multiple coordinated views.

Although the focus of our work is on analysis, researchers have also looked at how to improve an administrator's situational awareness when monitoring a network. Lakkaraju et al. [14] describe a tool called NVisionIP, which visualizes flows using three levels of granularity: a galaxy view, which shows the whole network, a small multiples view which shows the information for a selected set of hosts, and a view which shows the behavior of one machine. Similarly, Yin et al. [24] describe VisFlowConnect, which uses parallel coordinates to monitor the state of a network. IDS Rainstorm [2] visualizes alarms from an IDS by mapping them to color variations in a matrix-like visualization.

Researchers have investigated the idea of presenting users with a history of an interaction. Groth and Streefkerk [11] describe a system where users who are exploring a 3D model have access to a tree view of the actions they have taken. Jankun-Kelly et al. [13] describe a model of the visualization exploration process which can be expressed in XML. This can preserve a session for re-use by collaborators. Their work suggests a possible method for preserving information such as row lineage.

The layout for our system was inspired by other tabular visualizations. Table Lens [18] introduced the idea of using focus + context to visualize large amounts of data. Chi et al. allowed users to treat visualizations as elements of a spreadsheet [6]. Lifelines depicted a person's history with interactive timelines [17]. Our dynamic brushing and filtering of objects through our temporary displays is similar to Filmfinder, which allowed users to dynamically filter a 2D set of objects [3].

Hochheiser and Shneiderman have investigated methods of graphically querying a single time series in the domain of molecular biology [12]. One could imagine using their techniques to enhance the interaction with a single timeline, and then using pivoting to create a progression of views. Fails et al. [8] has described an interface for specifying patterns of events in temporal data that could present an alternative mechanism for pivoting to related timelines.

7 CONCLUSION

In this paper we have described a technique called progressive multiples that supports the forensic analysis and presentation of multidimensional event data. This technique was motivated by our collaboration with network security administrators and researchers in political communication. Our technique visualizes events as small multiples of timelines and allows users to pivot from an event to a set of related events. This progression of views provides user with a history which they can interact with or use as the basis for constructing a visual narrative of their analysis.

We presented a prototype for network administrators which we are continuing to refine. Our deployment with administrators indicated a need to support more generalized forms of pivoting. We also explored how our technique can be extended to other domains by creating a variant of our tool that visualizes the spread of a simulated epidemic. By observing how users interacted with a set of timelines, we identified potential areas for improvement.

The ability of our tool to support communication-minded visualization often depended upon the user's exploration strategy. Users who employed a breadth-first approach were often unable to recall why they had followed a particular path of analysis. We think that keeping track of a row's lineage, or the set of interactions that created a timeline, would allow users to use their exploration history more effectively. In the post-study interview, we found that although our tool can be used to interactively present a narrative, the use of transient interactive cues limits its effectiveness for static presentations. We plan to develop an interface that uses lineage and information about relationships among timelines that will allow users to author a static narrative of their analysis. We will also continue developing tools to support network administrators and communications researchers who are exploring data about the propagation of news stories across different news sources.

REFERENCES

- 1 Abdullah, K., C. Lee, G. Conti, and J. A. Copeland. Visualizing network data for intrusion detection. *IEEE Information Assurance Workshop*, 2005. pp. 100-08, 2005.
- 2 Abdullah, K., C. P. Lee, G. Conti, J. A. Copeland, and J. Stasko. IDS RainStorm: Visualizing IDS alarms. *IEEE Workshop on Visualization for Computer Security*, 2005. (*VizSEC 05*). pp. 1-10, 2005.
- 3 Ahlberg, C. and B. Shneiderman. Visual information seeking using the FilmFinder. *Conference on Human Factors in Computing Systems*: ACM Press New York, NY, USA. pp. 433-34, 1994.
- 4 Becker, R. A. and W. S. Cleveland. Brushing Scatterplots. *Technometrics* **29**(2): JSTOR. pp. 127-42, 1987.
- 5 Becker, R. A., S. G. Eick, and A. R. Wilks. Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics* **1**(1). pp. 16-28, 1995.
- 6 Chi, E. H. H., P. Barry, J. Riedl, and J. Konstan. A spreadsheet approach to information visualization. *Information Visualization*, 1997. *Proceedings*, *IEEE Symposium on*. pp. 17-24, 1997.
- 7 Cox, K. C. and S. G. Eick. Case study: 3D displays of Internet traffic. *Proceedings of INFOVIS 1995*. pp. 129-31, 1995.
- 8 Fails, J. A., A. Karlson, L. Shahamat, and B. Shneiderman. A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events over Time. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST '06)*, 2006.
- 9 Goodall, J. R., W. G. Lutters, P. Rheingans, and A. Komlodi. Preserving the Big Picture: Visual Network Traffic Analysis with TNV. *IEEE Workshop on Visualization for Computer Security*, 2005. (*VizSEC 05*). 2005.
- 10 Goodall, J. R., A. A. Ozok, W. G. Lutters, and A. Komlodi. A user-centered approach to visualizing network traffic for intrusion detection. *Conference on Human Factors in Computing Systems*: ACM Press New York, NY, USA. pp. 1403-06, 2005.
- 11 Groth, D. P. and K. Streefkerk. Provenance and Annotation for Visual Exploration Systems. *IEEE Transactions on Visualization and Computer Graphics* **12**(6), 2006.
- 12 Hochheiser, H. and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization* **3**. pp. 1-18, 2004.
- 13 Jankun-Kelly, T. J., K. L. Ma, and M. Gertz. A model for the visualization exploration process. *Proceedings of the IEEE Conference on Visualization*. pp. 323-30, 2002.
- 14 Lakkaraju, K., W. Yurcik, and A. J. Lee. NVisionIP: netflow visualizations of system state for security situational awareness. *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*: ACM Press New York, NY, USA. pp. 65-72, 2004.
- 15 Munzner, T., E. Hoffman, K. Claffy, and B. Fenner. Visualizing the global topology of the MBone. *Proceedings of IEEE Symposium on Information Visualization*, San Francisco, California, USA, 1996.
- 16 Phan, D., L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd. Flow Map Layout. *IEEE Symposium on Information Visualization*, 2005. *INFOVIS 2005*. pp. 219-24, 2005.
- 17 Plaisant, C., B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*: ACM Press New York, NY, USA, 1996.
- 18 Rao, R. and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*: ACM Press New York, NY, USA. pp. 318-22, 1994.
- 19 Ren, P., Y. Gao, Z. Li, Y. Chen, and B. Watson. IDGraphs: Intrusion Detection and Analysis Using Histograms. In *Proceedings of Workshop of Visualization for Computer Security*, 2005.
- 20 Stolte, C., D. Tang, and P. Hanrahan. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics* **8**(1). pp. 52-65, 2002.
- 21 Tufte, E. R., *Envisioning Information*: Graphics Press 1990.
- 22 Tukey, J. W., *Exploratory data analysis*: Addison-Wesley Menlo Park, CA 1977.
- 23 Viégas, F. and M. Wattenberg. Communication-Minded Visualization: A Call to Action. *IBM Systems Journal* **45**(4), 2006.
- 24 Yin, X., W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. VisFlowConnect: netflow visualizations of link relationships for security situational awareness. *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*: ACM Press New York, NY, USA. pp. 26-34, 2004.