



# Scalable and interactive visual analysis of financial wire transactions for fraud detection

Remco Chang<sup>1</sup>  
Alvin Lee<sup>2</sup>  
Mohammad Ghoniem<sup>1</sup>  
Robert Kosara<sup>1</sup>  
William Ribarsky<sup>1</sup>  
Jing Yang<sup>1</sup>  
Evan Suma<sup>1</sup>  
Caroline Ziemkiewicz<sup>1</sup>  
Daniel Kern<sup>2</sup>  
Agus Sudjianto<sup>2</sup>

<sup>1</sup>Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC, U.S.A.; <sup>2</sup>Bank of America, Charlotte, NC, U.S.A.

Correspondence:  
Remco Chang, Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223, U.S.A.  
E-mail: rchang@uncc.edu

## Abstract

Large financial institutions such as Bank of America handle hundreds of thousands of wire transactions per day. Although most transactions are legitimate, these institutions have legal and financial obligations to discover those that are suspicious. With the methods of fraudulent activities ever changing, searching on predefined patterns is often insufficient in detecting previously undiscovered methods. In this paper, we present a set of coordinated visualizations based on identifying specific keywords within the wire transactions. The different views used in our system depict relationships among keywords and accounts over time. Furthermore, we introduce a search-by-example technique, which extracts accounts that show similar transaction patterns. Our system can be connected to a database to handle millions of transactions and still preserve high interactivity. In collaboration with the Anti-Money Laundering division at Bank of America, we demonstrate that using our tool, investigators are able to detect accounts and transactions that exhibit suspicious behaviors.

*Information Visualization* (2008) 7, 63–76. doi:10.1057/palgrave.ivs.9500172

**Keywords:** Fraud detection; financial data visualization; categorical and time-varying data; visual analytics

## Introduction

Large American banks such as Bank of America handle hundreds of thousands to millions of wire transfers per day. While most of these transactions are perfectly legal, a small number are performed as part of criminal endeavors such as money laundering. The enormous amount of generated activity and the unconstrained nature of the data makes it very difficult to find these few instances among all the legitimate ones. At the same time, strict regulations require banks to spend considerable effort to find and report these activities, or face significant fines or even being shut down. The problems faced by risk managers and fraud analysts are exacerbated by the fact that an increasing number of transactions are purely digital and often involve a web of financial institutions around the world. Thus a bank's wire transfers may come from and go to individuals or businesses who are not the bank's customers. Often the bank is just a middle man for transactions that originate in different countries. In these circumstances, banks may know little about the individuals or businesses involved other than what is in the transaction record. Yet they must still exercise due diligence in discovering and reporting suspicious activity. For a large financial institution, this means monitoring hundreds of thousands of transactions per day, then investigating possibly suspicious ones in depth at considerable expense (and risk, if the monitoring is not effective). The problem is overwhelming and growing worse. Hierarchical interactive visual analysis with multiple-linked views can effectively attack this problem because it is geared toward the visualization

and interactive exploration of massive data sets, integrating multiple methods from various disciplines such as information visualization, human computer interaction, and statistics.

In this paper, we present *WireVis*,<sup>1</sup> a multiview approach that assists analysts in exploring large numbers of categorical, time-varying data containing wire transactions. Our method is highly interactive, and combines a keyword network view, a heatmap, a search-by-example tool, and a new visualization called *Strings and Beads*. These four views together fully depict the relationships among accounts, time, and keywords within the transactions, and present the user with a global overview of the data, providing the ability to aggregate and organize groups of transactions for better investigation and analysis and the ability to drill-down into and compare individual records. Although the examples and results in this paper concentrate on wire transaction data, the approach is general and applicable to any type of financial transaction data. This method should be effective for any keyword-based data, semi-structured or not, with varying but substantial levels of activity over time.

The large number of transactions at Bank of America poses a challenge for interactive visualization. With millions of records to visualize, the data would not fit into the memory of a regular desktop all at the same time. Data storage and organization thus become important aspects of the overall system. Furthermore, maintaining a global overview and the ability to interactively drill down to any record using such a large amount of data is not trivial. An efficient communication between data storage and the visualization component is essential for the user to explore the data without long delays.

Our solution to the data storage challenge is to connect the visualization tool to a commercial relational database with multiple indexes into the data for fast data retrieval. To maintain fast interaction with the data, we further create temporary tables in the database such that the visualization tool only receives enough information to create the visual elements but never the raw transaction data itself. By separating the data storage and organization from the visualization, the amount of data can be scaled up by increasing the performance of the database (via better hardware or tuning of the database) independently of the visualization component. Furthermore, with the data storage centralized to a single commercial relational database, we receive additional benefits such as security, data consistency, etc.

This work presents substantial qualitative advances over current practice in investigating financial transactions, which involves blind queries followed by painstaking analysis of spreadsheets.

- It provides an overview that scales to hundreds of thousands to millions of transactions over any desired length of time by using a commercial relational database.

- It provides tightly integrated views that look at patterns of activity over time and over keywords for clusters, sub-clusters, and so on.
- It replaces blind queries with contextual exploration, clustering, reclustering, and in-place analysis.
- It introduces powerful search-by-example techniques.

## Monitoring wire transactions

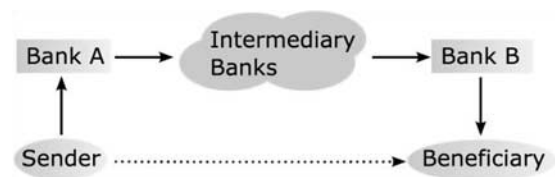
In collaboration with Bank of America, we have tackled the problem of monitoring wire transactions. As we describe the nature of the data and the current practice in monitoring wire transactions, we will shed light on the requirements of this problem and how interactive visual analysis can bring about a drastic improvement in financial investigation.

Normalizing data in the various fields of a wire transaction is difficult as these fields are frequently open to interpretation at the point of origination. Roughly speaking, a wire transaction corresponds to a certain amount of money sent by a payer to a payee via a chain of intermediaries, with or without additional comments or instructions (Figure 1). The payee could be the real beneficiary (e.g., a person or business) or his account holder, with additional information as to what to do with the money. The transferred amount could come from the sender's account directly or via a third party. It could also be sent to the receiver's account directly or handled through a third party. Other information such as the address of the sender and receiver, additional comments, and instructions may be appended. A wire transaction could be best seen as a semi-structured data record with numerous optional free text fields.

Currently, financial analysts probe wire transactions based on various considerations:

- Official rules governing which transactions must be reported, for example, when the amount of money exceeds a certain threshold.
- A potential risk inherent to the transaction, for example, when the destination corresponds to a high-risk country or organization or when the transaction relates to a high-risk activity.

While a set of filters can easily be set up to catch transactions matching a limited set of rules and automate a report generation process, risky or suspicious transactions



**Figure 1** The flow of money and associated information in a wire transaction.

are more evasive and uncertain. They are governed by an ever changing context where, for instance, geopolitical, economic and strategic motivations, and various actors are in play and where methods to hide illegal activity are constantly evolving as older methods are discovered and stymied. Hence, in the current practice, analysts query the transactional data over a time period, looking for certain keywords that may be indicative of high risk. Based on intelligence reports and previous analyses, investigators create a large list of keywords that best fit the international state of affairs. All transactions are filtered through this list of keywords, and the transactions that contain one or more keywords are displayed using a spreadsheet for investigation with transactions raising multiple red flags to be scrutinized more thoroughly. Additional information channels could then contribute to the investigation. For instance, home-grown expertise, the bank's own records, publicly available databases, and search engines could provide evidence for or against further action. An increase in false negatives would cast serious doubts on the financial institutions who would appear as purposely harboring fraudulent activities, not to mention having to pay large fines, whereas too many false positives could harm their relationships with their clients or irritate official agencies who would be wasting their time and resources on overly paranoid reports.

Adding to the complexity of the problem, the data include one-time transactions in great numbers as well as repeated transactions. To hide from scrutiny, customers engaged in fraudulent activities could follow common temporal patterns of legal activity or try to break away from any fixed pattern. Fraudulent activity could also be distributed over a variable number of senders and receivers. At present, such distributed frauds are mostly beyond the reach of financial investigators. In fact, at present they do not have the capability to investigate all the patterns and activities they probably should. Effective analysis tools need to take into account this wide variety of scenarios and allow the user to see the transactions and patterns that match them.

In summary, financial analysts are faced with hundreds of thousands of transactions bearing predefined keywords over periods of time. This set of data is categorical due to the classification using keywords, and is by nature time-varying. However, the data are also semi-structured and certain records are unstructured. Furthermore, since most transactions contain a limited number of keywords, the data are therefore sparse when viewed as a relationship matrix between transactions and keywords.

## Related work

Currently, analysts use spreadsheets to look at large data tables of transactions. Spreadsheets support various operations on rows and columns and give a detailed account of the data. However, they are not effective at providing a clear overview of trends and correlations. To fill this gap, several works in the information visualization field

have been proposed such as TableLens<sup>2</sup> and data sheets<sup>3</sup> and are now part of commercial information visualization suites.

Similarly, heatmap visualizations or correlation matrices have been used successfully in various application domains<sup>4</sup> and are now shipped in several commercial tools such as Spotfire Decision Site<sup>5</sup>. In genomics, this metaphor has been used for the visualization of massive gene arrays<sup>6</sup> and provides a compact overview of the data as well as a drill-down capability for detailed information. It has also been used to visualize social network data<sup>7</sup> as well as co-activity graphs in the arena of software visualization<sup>8</sup>. In the latter, structural and temporal patterns could be exhibited on correlation matrices. Although the orderability of matrices can enhance user performance with regards to certain tasks,<sup>9</sup> finding the appropriate order on a correlation matrix is a task-dependent question that will be receiving increasing interest.<sup>7,10</sup>

The need to follow temporal patterns in transactional data suggests the use of time-series data visualizations. Time-series data capture measurable quantities that change over time, such as stock values, climate data, etc. The users need to find trends and identify patterns in these data sets. Several works have focused on the display of this type of data, such as DiskTrees and TimeTubes<sup>11</sup>, and provided periodic views<sup>12,13</sup> supporting pattern identification. Recently more efforts<sup>14</sup> have been put into querying such data directly through user interaction. However, transactional data sets such as wire transfers could be more challenging to these kinds of tools due to the sparsity of the data points, as most accounts would fire a transaction only once in a while.

Monitoring keyword-tagged transactions over time also suggests the use of corpora exploration and visualization tools such as ThemeRiver.<sup>15</sup> However, this work can only handle a small set of topics or keywords simultaneously, whereas wire data bear on tens or hundreds of corporate-defined keywords. Moreover, ThemeRiver is geared towards the display of overall trends and temporal changes in topics, rather than outliers and isolated behaviors, or, in other words, rare anomalous transactions.

Systems offering multiple coordinated views have arisen as a suitable solution for complex multi-faceted data sets. Recent works such as Snap-Together visualization have studied the feasibility of such systems,<sup>16</sup> provided a taxonomy<sup>17</sup> and model<sup>18</sup> of multiple-view systems, and studied the benefits and trade-offs of such systems from a user perspective.<sup>19</sup> Other authors provided guidelines<sup>20</sup> for using multiple-views systems for information visualization. In the current work, we set out building a multiple coordinated views system in view of capturing as many aspects of transactional data sets as desired.

Owing to the size of the data involved, we have to store it in a commercial relational database management system. While many visualizations are reading data from databases, we are not aware of work that is taking advantage of a specifically designed database scheme for

interactive visualization of large data sets directly from a database. Polaris<sup>21</sup> builds queries into databases based on visual specifications of the attributes to be shown. Tioga<sup>22</sup> allows the rapid development of visualizations that can be used to explore and drill down into databases. DBVis<sup>23</sup> uses pixel-oriented techniques to show as much data as possible from a database on a computer screen.

## Visual analytics tools for monitoring wire transactions

WireVis uses four tightly coordinated views of transaction activity. The keyword network view is used to represent the relationships between keywords, the heatmap view shows relationships between accounts and keywords (see the section Monitoring wire transactions), the search-by-example tool helps discover accounts of similar activities, and lastly, *Strings and Beads* depicts the transactions over time. All views rely on high interactivity along with the ability to see global trends and capabilities to drill-down into specific transaction records (Figure 2).

### User-centric design

The design of WireVis was based on an analysis of the current work of fraud analysts with their existing tools. To manage the enormous amount of information, analysts first filter the data by geographic region using a set of specific keywords and other criteria (like amounts). These data are then inspected by hand, with additional tools like search engines to find out if businesses are legitimate, etc.

For WireVis, we wanted to keep as many of the working aspects of the existing system, while simultaneously enhancing the shortcomings. We therefore defined the following list of requirements for the system:

**Interactivity** Despite the large amounts of data, WireVis must be highly interactive and respond to user input immediately.

**Filtering** The current work method of filtering data using predefined keywords and other criteria must be kept to increase acceptance of the system. At the same time, we cannot rely on the data being filtered down, but must be able to show and work with all the data at once.

**Overview and detail** It must be possible to see aggregated views of all transactions of a day, week, or month in an overview, and then to drill down to the level of individual transactions when needed for the investigation.

**Coordinated multiple views** No single view could fulfill all the requirements and show all the necessary data, so a system of coordinated views was designed that would allow the user to see different data, while being able to understand the connections between the views easily.

During the design phase of the project, we interviewed and communicated with members of the Risk Management, Compliance, and WireWatch (analysis) divisions of Bank of America on their current practices as well as their needs for monitoring fraudulent wire transfers. Furthermore, throughout the development phase, we maintained close communication with these groups and routinely showed them our progress and received feedback.

### Data aggregation

Since one cannot usually detect suspicious activity from single wire transactions, we need to visualize the activities of the corresponding accounts in order to detect suspicious behaviors. The transaction data are therefore first grouped according to the sending and receiving accounts, and the heatmap visualization tools show the



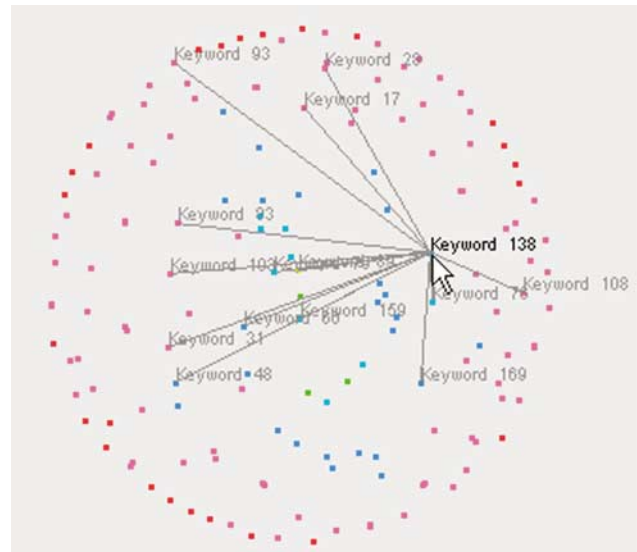
**Figure 2** A view of the entire system showing the heatmap (top left), search by example (top right), keyword graph (lower right), and strings and beads (lower left).

accounts instead of each individual transaction. Even so, the number of accounts still range in the tens of thousands or more; thus, we hierarchically cluster the accounts. This provides the scalability needed, the high-level overview that is imperative, and useful levels of abstraction. The complexity of the clustering algorithm is crucial because the analysts often need to perform reclustering as patterns in the transactions are discovered. Also, the clusters are often not optimal for the analyst's current purpose, so interactive reclustering has proved highly useful in exploratory visualization since it will provide much better clusters for further exploratory analysis<sup>24</sup>. Most existing clustering techniques such as  $k$ -means  $O(kn)$  or single-link clustering  $O(n^2)$  require minutes to hours to compute as  $n$  becomes large, which is unacceptable in our case.

Instead, we use a simple 'binning' technique to find groupings of accounts based on frequency of keywords that occur in the transactions of the accounts. We treat each account as a point in  $k$ -dimensional space (where  $k$  is the number of keywords), and group the accounts based on their distances to the average point of all accounts. This method has the complexity of  $O(3n)$  and can cluster tens of thousands of accounts in seconds. The wire analysts have approved this as a crude but effective way to explore the transaction space. The effectiveness is significantly enhanced because the method provides fast, dynamic overviews of the data while relying on the high interactivity and multiple views to assist in exploring the data space. Our procedure is to apply the hierarchical binning method as a preprocessing step, thus organizing all the data for a selected time period before launching interactive exploration. Then the user can use a strategy of selecting reasonably sized subsets for reclustering, thus maintaining interactivity. We have found that this strategy works well in practice. An alternative could be to provide the user with a more exact but more time-consuming reclustering approach to be used at any point in the exploration. However, if this approach was not hierarchical, its value would be limited.

### Keyword network view

Depicting relationships between keywords is important for identifying questionable transactions. If a transaction contains two keywords that should not be related in the context of a wire transfer, it should be quickly identified and further inspected by an investigator. To show the relationships between keywords, we use a simple network graph as shown in Figure 3. A keyword is said to be related to another if both of them appear in the same transaction. The appearance of keywords in the same transaction forms the basis of the underlying relationship matrix in which the distances between the keywords are calculated based on the number of times that they appear together in transactions. In Figure 3, keywords closer to the center of the keyword network view are the most frequently appearing keywords, whereas keywords on the outskirts of the circle



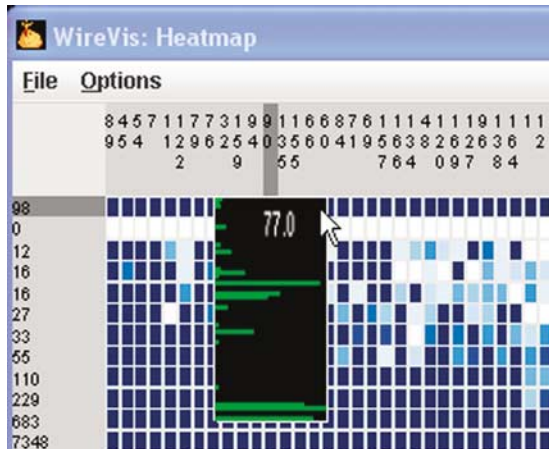
**Figure 3** The keyword network view shows the relationship between keywords. The most frequent keywords appear in the middle of the view, while the less frequent ones appear on the outskirts of the circle. When a user highlights a specific keyword, lines are drawn from that keyword to all relating keywords.

appear less frequently. When a keyword is highlighted, lines are drawn between the highlighted keyword and all relating keywords.

### Overview of keyword-to-account relationships

We use a heatmap to display statistical measurements relating keywords (see the section Monitoring wire transactions) and bank accounts, as the former occur in transactions involving the latter. Our heatmap uses a grid whose columns are the keywords of interest and whose rows are clusters of bank accounts being scrutinized. At the intersection of a given row and column, we color-code a value such as the number of hits for that keyword/column with regards to that account/row in the time-period encompassed in the data. Depending on the nature of the measurement displayed in the grid (e.g., sequential or diverging), various color schemes<sup>25</sup> can be applied to the visualization. We use a simple scheme where the saturation of the cell is proportional to the number of times a keyword appears for that set of transactions, since it is intuitively related to the heatmap concept. Then, the user can spot at a glance the accounts that are more frequently related to a given keyword or set of keywords (e.g., a high-risk country heavily involved in money laundering). He can also see common keywords (full columns in the grid) that are likely to be filtered out in an investigative process. Likewise, accounts hitting all keywords (full rows in the grid) usually correspond to financial institutions rather than individual accounts and would also be deemed irrelevant to investigative work. Moreover, analysts can detect keywords displaying





**Figure 4** Heatmap with close-up view on a cell histogram displaying the number of keywords for each account. Each horizontal (green) line in the histogram represents an account in the cluster. The x-axis shows the number of 'hits' of the keyword for that account. The purpose of showing the histogram is to combat the effect of using a cumulative sum in the Heatmap. By looking at the histogram, the user can quickly identify if the accounts in the same cluster contribute evenly to a specific keyword, or if there are abnormal distributions within the cluster.

a similar activity, that is, hitting the same accounts. In many cases, such coupling of keywords can be accounted for easily (e.g., Paris and France). In other situations, the coupling of two remotely related keywords will trigger further investigation. Therefore, the heatmap view makes it possible to visually compare patterns of behavior across different accounts. Our tools are enhanced with a user-configurable search-by-example capability (see the section search by example) that helps the analysts find accounts, respective to keywords, that are similar to a reference account. Search-by-example is a powerful tool for exploratory analysis, since it permits the user to quickly identify and search for behaviors of interest without having to specify those behaviors in detail.

Since analysts must try to grasp hundreds of thousands of transactions involving as many accounts, it is necessary that the heatmap be scalable in this dimension. Our tools perform clustering as described in the section Data aggregation, providing a high-level abstraction of the data as a first overview. Then, the user can drill-down any set of clusters or keywords through direct interaction by selecting the desired subgrid or by expanding a cluster of his choice. The user can also hover on the heatmap and overlay the value associated to each cell. In case of aggregate data, the user can also overlay a glyph, for example, a histogram, displaying the real distribution of low-level values that add up to the aggregate value as in Figure 4. By doing so, we strive to quickly provide the user with more detail at key points to enhance both exploration and enlightened decision-making.

### The strings and beads visualization

The ability to look for suspicious activities over a period of time is crucial to the analysts. To illustrate, when a terrorist attack took place somewhere in the Middle East, an intelligence agency had reason to believe that the attack was supported and funded by individuals in the U.S. It requested bank wire analysts to search for wire transfers between the U.S. and the location of the attack. By looking for wire activities over a narrow range of time prior to the attack and on selected keywords, the analysts were eventually able to identify the culprit and report the incident. In addition to this event-driven scenario, it is also crucial to look at patterns of activity over periods of time (typically months), as these can bring up unusual behavior. Our coupled interface allows the user to quickly find these temporal patterns and activities.

In order to support visualization of wire activity over time, we create the *Strings and Beads* view in which the *strings* refer to the accounts or cluster of accounts over time, and the *beads* refer to specific transactions on a given day. Together, the strings and the beads show the overall trends of the activities as well as the individual transactions. The x-axis of the view shows the progression of time, and the y-axis shows the 'value' of the transaction, where value can be the amounts of the transactions, the frequency of activities, etc. Figure 5 shows that the Strings and Beads view is quite effective in giving an overview on top of showing specific detail. The strings show the overall activities for selected accounts or clusters for an entire year, and the beads depict the details of a handful of transactions for that day.

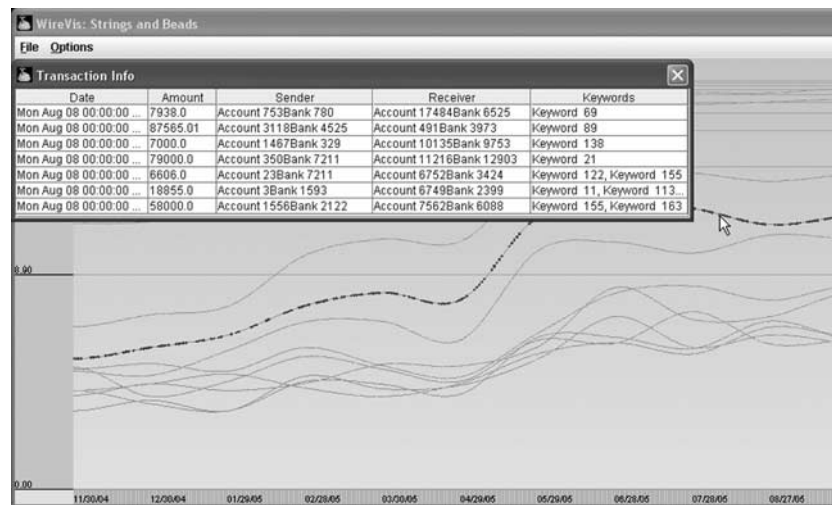
Due to the fluctuation of the data, we choose to represent strings as splines instead of disjointed line segments. Since transactions do not take place over weekends or holidays and often vary drastically in amounts or frequency, representing the strings as line segments creates jagged lines, making it difficult to distinguish between different strings. Instead, we smooth out the strings as splines with the option to change the number of control points, which gives a good overview of trends over time. At any point, the user can display the original transaction values for detailed analysis, as shown in Figure 6.

To facilitate fast interactions with the Strings and Beads view, the analysts can quickly zoom in to the time period in question by brushing a range of time. To further examine the details of a specific wire transaction, the analysts can double-click on a bead to bring up the original wire information in a separate window as shown in Figure 5.

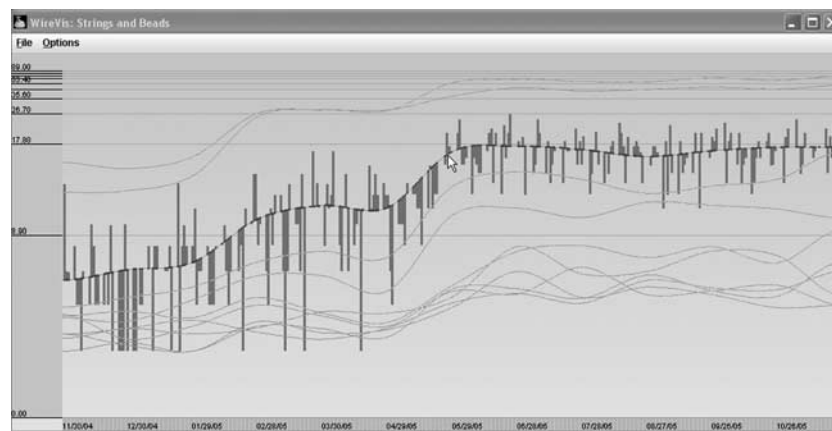
### Coordination between views

The resulting clusters from binning provide a foundation for the coordination between the four views. Since the underlying data structure is the same, message passing between the views becomes trivial.

With the four views coordinated together so that an action performed in one view affects all other windows,



**Figure 5** Double-clicking on a specific bead brings up the related wire transactions in a separate window. The user can interactively drill-down into individual transactions.



**Figure 6** Turning on the option to show the original data, the user is able to see the real data points (shown as bars) on top of the smoothed string.

the analysts can now interact with accounts, keywords, time, account values, etc. and see how the selections correlate in all dimensions. This is significantly more powerful than using the views separately.

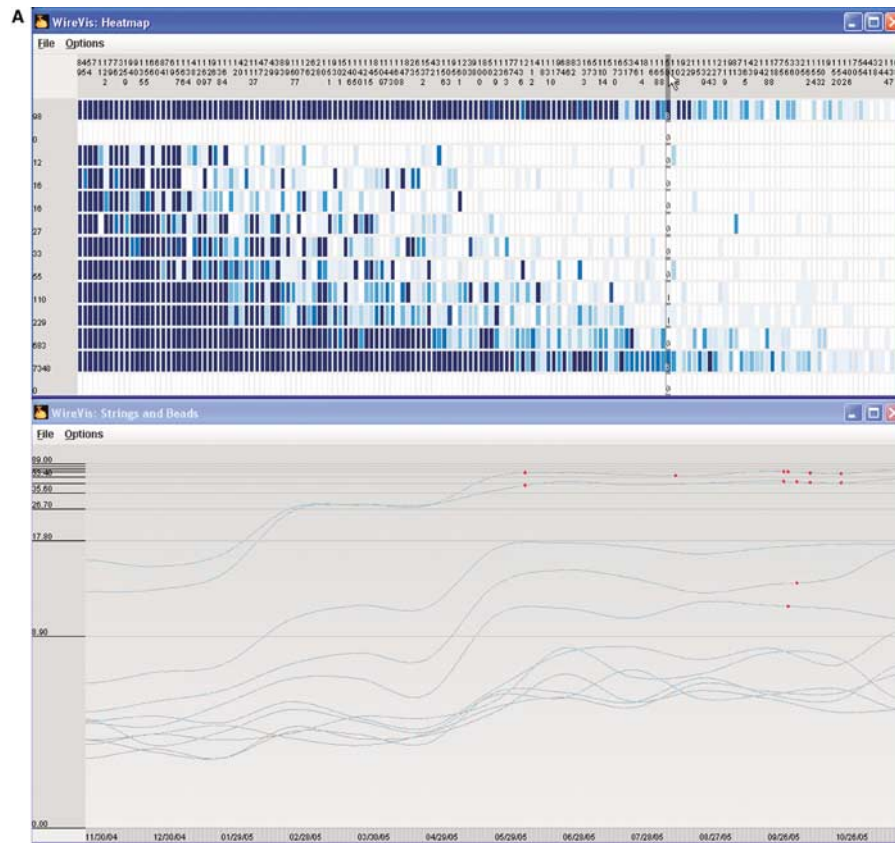
For example, when an analyst hovers the mouse over the keyword names in the heatmap view, all cells in the heatmap are highlighted along with the number of occurrences of the keyword in each account cluster. At the same time, all the beads in Strings and Beads are highlighted to show when the transactions with such keyword occur (see Figure 7A). This highlighting technique allows the analyst to search for suspicious keywords and see when these keywords occur over time.

Similarly, the analyst can hover over the dates in the Strings and Beads view, highlighting all the beads of a particular day. The heatmap then reacts by highlighting all the cells that contain the transactions of these beads

and displaying the number of occurrences of the keyword for that day (see Figure 7B). This allows the analyst to focus on specific dates and observe which accounts are transacting over what keywords over that period of time.

Such tight integrations occur throughout the keyword network view, the heatmap view, and the Strings and Beads views: selecting a string highlights a row in the heatmap, selecting a cell shows all the beads that contain such keyword, etc. All these quick actions permit rapid exploration over many accounts, keywords, and time ranges in terms of animated patterns, as shown in the video, at an almost subliminal level. The user can then pause, slow down, or go back to observe a pattern more closely.

Finally, both the heatmap and strings and beads view react to zooming (e.g., the cells in the heatmap view will only contain values over a certain time range when a user



**Figure 7** (A) Highlighting keywords in the heatmap view shows the corresponding beads in the Strings and Beads view. The user can interactively see keyword occurrences over time.

zooms into a time period in the Strings and Beads view), the analysts can see a global trend of the account activities over time by simple highlighting, but has the ability to further investigate specific incidences, events, and time ranges via zooming if necessary.

### Search by example

Because of the complex structure of the data and the observed patterns, it is difficult to define transaction patterns that one is looking for *a priori*. Once an interesting pattern is found, it is usually necessary to find not just transactions from and to the involved accounts, but also accounts that show similar activities.

An important feature of WireVis is therefore to search by example (see Figure 8). In a separate view, the user is shown the currently selected cluster or account to use as the prototype for a new search. Bars represent the number of hits for all the defined keywords, which the user can select as relevant criteria by clicking them. A slider is used to define the maximum difference for identifying a cluster as similar to the prototype. As the user moves the slider, the number of similar accounts grows or shrinks, giving the user a feeling for the space to explore. There is no separate search button, the search is performed whenever

the user changes the criteria or the threshold, and results are shown immediately.

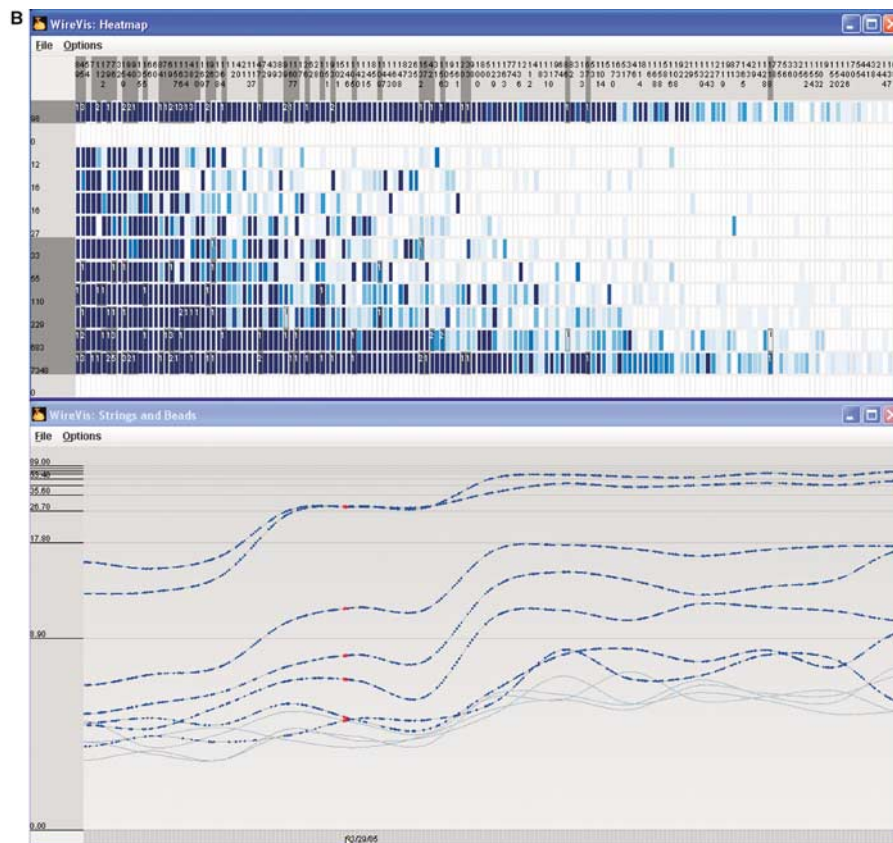
The user can then select particular results, which are shown in other views for further investigation.

### Connecting to a database

The number of wire transactions processed by Bank of America ranges in the hundreds of thousands per day. Filtering the transactions using keywords reduces the number of transactions that require further examination by up to 90%, but the remaining 10% can still amount to tens of millions of transactions over the course of a year. Since this amount of data would not fit into the memory of a regular desktop computer, a strategy for storing and organizing the data to support interactive visual analysis becomes a critical aspect of the design process.

One method for organizing the data is to store pre-computed results of specific views in a hierarchical flat file structure. For example, the cells of the heatmap view for every possible cluster and keyword pair can be pre-computed and stored on file. However, it soon becomes evident that such a strategy is difficult to implement when both sub-selection of time and reclustering change the views and the cluster hierarchy. To create and





**Figure 7** (B) Hovering in the time axis in Strings and Beads highlights the beads of that particular day, as well as the related cells in the heatmap. The user can quickly browse through time and see keyword and account activity.

maintain pre-computed results for all time ranges while retaining flexibility for reclustering in a flat file system would increase the storage requirement exponentially.

Instead, we store the data in a commercial relational database management system (RDBMS). The main advantage of using an RDBMS is the ability to scale the database as the amount of data increases. This can be done by fine tuning the RDBMS, adding more or better hardware, or distributing the database across multiple computers. On the flip side, connecting to an RDBMS over ODBC or JDBC using SQL queries has significant performance penalties when real-time interaction is the end goal. By storing and organizing wire transactions in an RDBMS, our design problem becomes minimizing the number of SQL queries sent to the RDBMS and the amount of data transferred for each query.

‘Go to where the data is’ is the motto that we follow when connecting the visualization component to the RDBMS. Conceptually, our goal is to perform all computations in the database, and only return enough results to the visualization component for it to render itself. Under this scheme, the visualization component is never aware of the wire transaction data (unless specifically requested by the user), but only receives enough information to create the visual elements.

### Database design

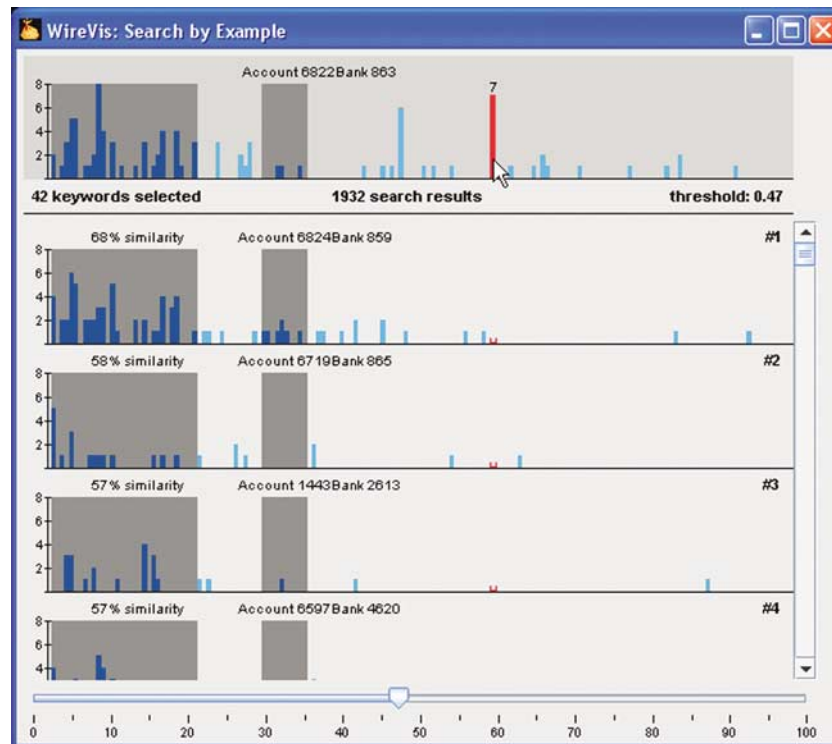
We create temporary tables for each visualization view (e.g., heatmap, Strings and Beads, etc). The tables contain information specifically for their corresponding views. For example, the temp table relating to the heatmap view has three main fields: cluster ID, keyword ID, and count, which is the information necessary for rendering the heatmap view.

During runtime, user interactions with the visualization components invoke stored procedures to perform data-related tasks such as binning/clustering and hierarchy traversal (drill downs). These procedures store the results of their computations in the temp tables, which are then fetched from the database into the visualization components to create the visual elements (Figure 9).

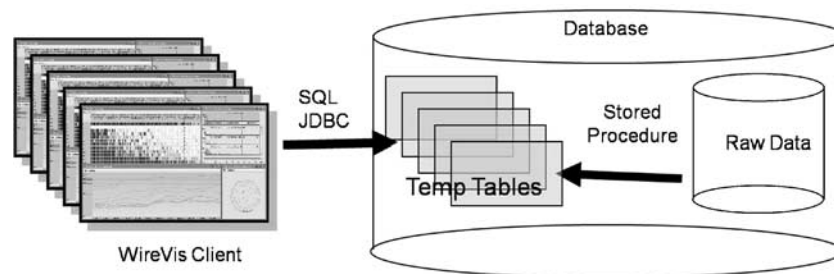
### Implementation

For our implementation, we use Microsoft SQL Server 2005 Express Edition as our RDBMS. Three tables are created to store the raw data, and three tables are used to organize it (Table 1).

Five temp tables are created to correspond to the three main visualization views (Table 2). Heatmap\_Table contains information on rendering the Heatmap. Strings



**Figure 8** Search by Example. The user can select a prototype cluster/account as well as keywords of interest. The program shows all similar clusters according to a user-controlled similarity threshold.



**Figure 9** Connecting WireVis to a database. Instead of directly connecting to the raw transaction data via SQL queries, WireVis only receives enough information to render the visual elements. Stored procedures perform computations on the raw data and place the results in temp tables from which WireVis retrieves the information it needs.

and Beads uses two temp tables, one for storing amount information and the other for information regarding keywords and their frequencies. Transaction\_Table is used to store results for querying specific transaction information (via double-clicking on a bead, see Figure 5). Lastly, Keyword\_Network\_Table stores the positions of the keywords to create the network view.

A set of stored procedures is used to process the data in the data tables and insert them into the temp tables (Table 3). These stored procedures are invoked by the visualization front end when the user interacts with the

system, and they store the results to the temp tables that are then retrieved by the visualization components.

Data in the temp tables are not easily updatable in our current design. When new data need to be inserted into the database on a daily basis, the data would need to be processed overnight to be available the following day.

### Performance results

We tested our database design on a desktop Windows computer with an Intel dual core 2.0GHz processor,

**Table 1** Tables in the database

Table name	Attributes	# Records
Accounts	Account_ID	500k
	Account_Number	
	Bank_Name	
Keywords	Keyword_ID	250
	Keyword_Text	
Transaction	Transaction_ID	7000k
	Sender Account_ID	
	Receiver Account_ID	
	Transaction_Date	
	Transaction_Amount	
Account_Date_Keyword	Account_ID	7500k
	Keyword_ID	
	Transaction_Date	
Trans_Account_Keyword	Transaction_ID	21,000k
	Account_ID	
	Keyword_ID	
	Transaction_Date	
Trans_Keyword	Transaction_ID	10,000k
	Keyword_ID	
	Sender_Account_ID	
	Receiver_Account_ID	
	Transaction_Date	
	Transaction_Amount	

The first three tables are used to store the data, and the last three tables are used to organize the data for fast access. Note that the number of records shown in the last column are approximate for the reason of anonymity.

1 Gigabyte of memory, and 80 Gb SATA hard drive. The database contains approximately 7 million records relating to one or more keywords over 13 months. In our experiments, both the database and the visualization components are run on the same computer.

Table 4 shows the amount of time required to perform each operation. The total number of records is approximately 7 million. However, as the user navigates lower into the hierarchy, the number of records used to perform each operation decreases. To demonstrate the scalability of the system, we show the performance of each operation using the full 7 million records, 3 million records (which could be the amount after one drill down), and 300 thousand records.

The user maintains real-time interaction with the visualization front end during exploration. It is only when a cluster/recluster, a drill down, or a transaction request occurs that the user has to wait for the database to respond. The experiment was conducted on a mid-range computer with limited memory. With better hardware, we should be able to further decrease the response time of the database.

**Table 2** Temp tables that correspond to the views in WireVis

Table name	Attributes
Heatmap_Table	Cluster_ID
	Keyword_ID
	Keyword_Count
SB_Table	Cluster_ID
	Date
	Total_Amount
	Num_Transactions
SB_Data_Table	Num_Keywords
	Cluster_ID
	Date
	Keyword_ID
Transaction_Table	Keyword_Count
	Cluster_ID
	Transaction_ID
	Date
	Amount
	Sender Account_ID
Keyword_Network_Table	Receiver Account_ID
	Keyword_String
	Keyword_ID
	X_Position
	Y_Position

**Table 3** Stored procedures that can be invoked by the visualization components via JDBC

Stored procedure	Description
Recluster	Clusters accounts at the top level
Cluster_Drilldown	Drills down into a cluster
Gen_Heatmap_View	Fills in the Heatmap_Table temp table
Gen_SB_View	Fills in the SB_Table and SB_Data_Table temp tables
Gen_Transaction_View	Fills in the Transaction_Table temp table

**Table 4** Time required to execute each stored procedure (in seconds)

Stored procedure	7 million	3 million	300 thousand
Recluster	57	30	8
Cluster_Drilldown	33	14	0
Gen_Heatmap_View	12	4	0
Gen_SB_View	12	4	0
Gen_Transaction_View	0	0	0

In addition, however, our performance testing revealed an unexpected result. The WireVis interface was found to be a quite efficient way to access the database and retrieve

useful information in general. This is a potentially important result because our colleagues at Bank of America tell us that the database is notoriously hard to use with up to 80% of a user's time spent just making queries (and finding the appropriate information for use). One reason for this is that the user is 'flying blind' and must plan and make many queries to probe the database for a given task. On the other hand, WireVis provides a highly interactive, exploratory capability for seeing information in context and getting further detail whenever needed. One uses this exploratory, probing capability *before* making specific queries, which can be launched from within WireVis. Since this process can be performed so quickly, the user gets relevant information right away, not only information that she may specifically ask for but also related information that is discovered to be relevant for the task at hand. This is a general insight that will apply to other uses of databases such as this one, not just the uses described here. We plan to pursue this research path further in our ongoing work.

### Case studies

In order to assess the usefulness of our tools, we employed a sanitized data set containing transactions sampled over twelve months. For privacy and proprietary reasons, account numbers, keywords' and personal information have all been stripped.

While certain members of the Risk Management and Compliance groups have used the system, our expert evaluators at WireWatch were not able to do so. This is due to the fact that the WireWatch group is located in California and has specific security-related hardware and software restrictions that make the installation of our tool impossible in the scope of the first phase of the project. In order to receive feedback from these key collaborators, we asked James Price, Senior Vice President in Bank of America's Global Anti-Money Laundering organization, to observe video of interaction with our system during a teleconferencing session and provide his interpretation of the visualizations. We categorize his observations into two groups: seeing normal, unsuspicious behavior and detecting activity that may indicate fraud.

### Seeing normal behavior

The clustering provides an obvious separation between large corporations or financial institutions from small businesses and individuals. The first row of the heatmap contains accounts that transact in large amounts (as can be seen in the Strings and Beads view) in high frequency over a large range of keywords. These typically represent large institutions and can often be filtered out from consideration. The last row of the heatmap contains individuals that only have exactly one transaction over the course of the year (which can be verified by drilling-down into the sub-clusters), and can sometimes be filtered out because they might not contain sufficient indications of suspicious behaviors.

The keywords in the heatmap view are sorted based on their frequency in all transactions over all accounts. Thus, the most frequently occurring keyword appears furthest left in the view. From this view, one strategy could be, for example, to filter out the keywords that occur with expected frequencies, then recluster all accounts using the remaining keywords to provide a more focused overview of the activities.

### Detecting suspicious activity

First, we identify a keyword that shows abnormal temporal patterns by hovering over the list of keywords in the heatmap view. During the hovering, we see in the Strings and Beads view that keyword 58 occurs only in the second half of the year as can be seen in Figure 10A (which is a detail of Figure 7A). This peculiar time-based behavior prompts further investigation into the transactions involving that keyword.

By rubber-banding in the heatmap view and zooming into the column of keyword 58, we notice that not many transactions involve this word. Switching to the Strings and Beads view, we change the  $y$ -axis to show the amount of the transactions instead of the number (Figure 10B).

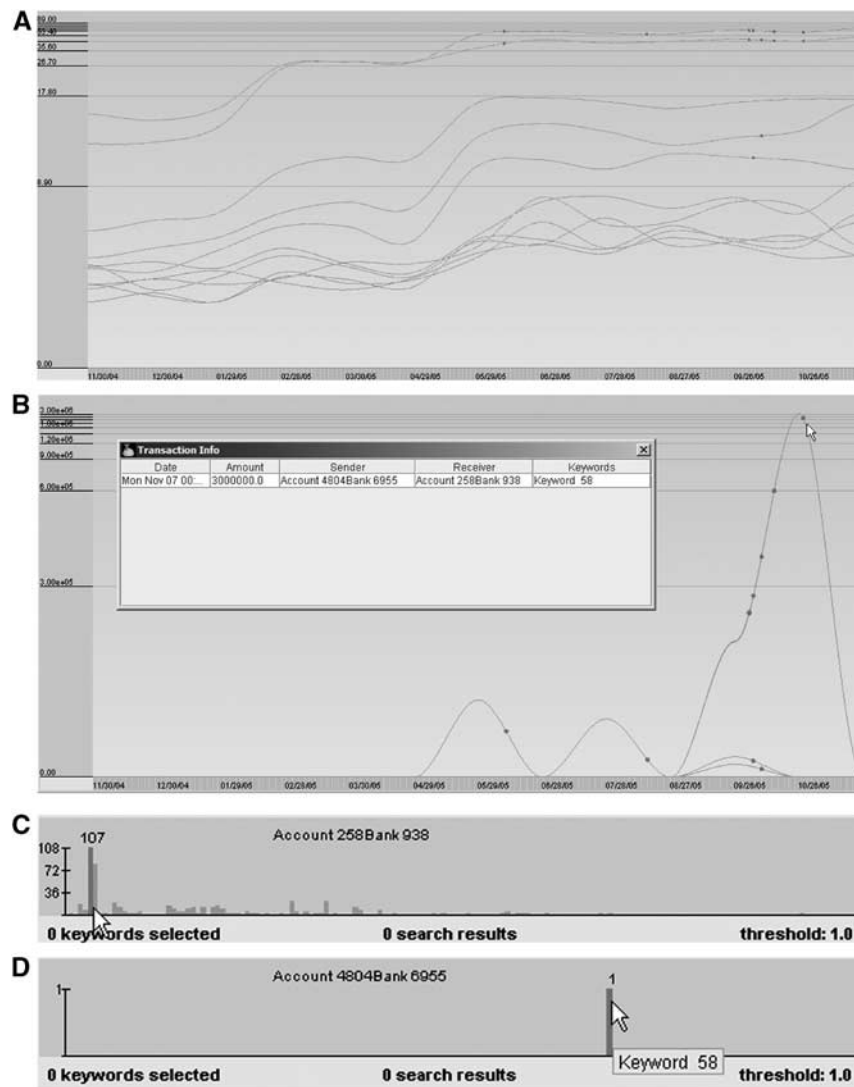
There is a transaction near the end of the year of approximately 3 million dollars. This transaction is peculiar because all other transactions involving this keyword have transaction amounts in the range of tens to hundreds of thousands of dollars. The fact that this particular transaction is of an amount much larger than others makes it stand out as an outlier.

Double-clicking on the bead representing that transaction reveals details about the transaction (Figure 10C). Clicking on the receiver's name brings up the Search By Example window where we see that the receiving account most likely belongs to a bank or a large institution because of the number of transactions and the range of keywords it is involved in. More than likely, this large institutions is an intermediary handler of the transaction, and therefore is not of interest to the investigator. However, clicking on the sender shows that that this account has had only one transaction over the past year (Figure 10D). Not only is this transaction of a very large amount, but it also involves a keyword showing an abnormal temporal pattern. Although there is no single attribute of this transaction that would warrant an investigation into the origination account, the combination of all facets of this transaction lead to further investigation.

### Discussion and future work

By providing exploratory tools for the very specific data of wire transfers, we enable the experts to take a very analytical but still much less constrained approach than using other tools. While WireVis does not provide many of the other methods currently used in fraud detection, its tools are complementary and useful.

Considering that the only tools the analysts currently have are lists of text, they find the use of visual metaphors



**Figure 10** Case study using sanitized real-world data. Fraud analysts found the combination of suspicious patterns sufficient to launch a full investigation. (A) When mousing over keyword 58, we see that it only appears a few times, and all those are in the second half of the year. (B) Changing the y-axis to show the amount of the transactions and double-clicking on the highest bead shows the transaction details. (C) The receiving account's information in the search-by-example view shows that this is likely a bank or large company. (D) The sender, on the other hand, has only used this one keyword, and with a very high sum.

to identify questionable behaviors very interesting and promising. However, the WireWatch group has asked to be able to install and use the system on live data for further validation. We understand that making the integration of WireVis into their daily practice as seamless as possible is very important, and is one of our top priority items for the next phase of our project.

Since seamlessly supporting the work of the analysts is such a key aspect of this effort, WireVis is designed to be naturally extensible. Any statistical indicators and metrics can be plugged in depending on the need of the analysts or the latest intelligence information. For example, our clustering technique using binning is not intended to

be the only solution to grouping accounts. It is only an example of how a fast clustering algorithm can enhance the analysts' ability to interact with the data. Depending on the need of the analyst, other clustering techniques can be used.

Although we provided four views that show relationships between accounts, keywords, and time, they are not the only four possible views that could be used. During our discussions with analysts, we have identified other views that can be helpful. Some examples of the views include a matrix or graph view showing the relationship between accounts and a geographical view showing the location of the senders and receivers. We will explore these possibilities in the future.



WireVis can be made significantly more powerful by adding an unstructured text analysis capability, such as that used in IN-SPIRE<sup>24</sup>. This would permit relating any words in the transaction fields to the keywords. Important relations could be uncovered and even new keywords found. We expect to develop WireVis along this line.

Finally, as indicated in the beginning, the WireVis approach described here can be applied to any transactional data or, indeed, to any keyword-based data over time. We will be expanding WireVis by coupling to analysis tools that search for money service businesses (i.e., businesses that deal in money transmission, check cashing, money orders, currency exchange, etc.).

## Conclusion

We demonstrate that using interactive visualization techniques coupled with hierarchical analyses in searching for suspicious financial transactions significantly enhances the analysts' ability to see global trends as well as quickly narrow down to individual activities.

We create our visualization tools based on the principles of high interactivity and coordinated multiviews. With this combination, the analysts can highlight elements in one view and see that element depicted in a different way in the other. This allows the analysts to see a complete relationship between accounts, keywords, time, and patterns of activity.

## Acknowledgements

We thank James Price and his team of investigators in the Global Anti-Money Laundering organization at Bank of America. This work was performed with support from Bank of America and the National Visualization and Analytics Center (NVAC<sup>TM</sup>), a U.S. Department of Homeland Security Program, under the auspices of the SouthEast Regional Visualization and Analytics Center. NVAC is operated by the Pacific Northwest National Laboratory (PNNL), a U.S. Department of Energy Office of Science laboratory.

## References

- Chang R, Ghoniem M, Kosara R, Ribarsky W, Yang J, Suma E, Ziemkiewicz C, Kern D, Sudjianto A. WireVis: Visualization of categorical, time-varying data from financial transactions. *Proceedings Visual Analytical Science and Technology 2007 (VAST 2007)*. IEEE Symposium on, October 30, 2007–November 1, 2007 (Sacramento, CA, USA), IEEE CS Press: Silver Spring, MD, 2007; 155–162.
- Rao R, Card SK. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: Adelson B, Dumais ST, Olson JS (Eds). *CHI*. ACM: New York, 1994; 318–322.
- Eick SG. Visual discovery and analysis. *IEEE Transactions on Visualization and Computer Graphics* 2000; **6**: 44–58.
- Abello J, van Ham F. Matrix zoom: A visual interface to semiexternal graphs. *INFOVIS: Proceedings of the IEEE Symposium on Information Visualization 2004 (INFOVIS '04)* (Washington, DC, USA), IEEE Computer Society: Silver Spring, MD, 2004; 183–190.
- Spotfire. Decision site for functional genomics. <http://www.spotfire.com/>, 2005.
- Seo J, Shneiderman B. Interactively exploring hierarchical clustering results. *Computer* 2002; **35**: 80–86.
- Henry N, Fekete J-D. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department: Piscataway, NJ, USA; 2006; **12**: 677–684.
- Ghoniem M, Cambazard H, Fekete J-D, Jussien N. Peeking in solver strategies using explanations visualization of dynamic graphs for constraint programming. *SoftVis '05: Proceedings of the 2005 ACM Symposium on Software Visualization*, ACM Press: New York, 2005; 27–36.
- Ghoniem M, Fekete J-D, Castagliola P. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization* 2005; **4**: 114–135.
- Siirtola H, Mäkinen E. Constructing and reconstructing the reorderable matrix. *Information Visualization* 2005; **4**: 32–48.
- Chi EH, Pitkow J, Mackinlay J, Piroli P, Gossweiler R, Card SK. Visualizing the evolution of web ecologies. *CHI '98: Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, ACM Press/Addison-Wesley Publishing Co.: New York/Reading, MA, 1998; 400–407.
- Keim DA. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department: Piscataway, NJ, USA; 2000; **6**: 59–78.
- van Wijk JJ, van Selow ER. Cluster and calendar based visualization of time series data. *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization 1999 (Washington, DC, USA)*, IEEE Computer Society: Silver Spring, MD, 1999; 4.
- Hochheiser H, Shneiderman B. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization* 2004; **3**: 1–18.
- Havre S, Hetzler E, Whitney P, Nowell L. Themeriver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics* 2002; **8**: 9–20.
- North C, Shneiderman B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. *AVI '00: Proceedings of the Working Conference on Advanced Visual Interfaces*, ACM Press: New York, 2000; 128–135.
- North C. A user interface for coordinating visualizations based on relational schemata: snap-together visualization. Ph.D. thesis, University of Maryland, 2000.
- Boukhefifa N, Roberts JC, Rodgers P. A coordination model for exploratory multi-view visualization. *Proceedings of the International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2003)*, IEEE: New York, 2003; 76–85.
- North C, Shneiderman B. Snap-together visualization: can users construct and operate coordinated visualizations?. *International Journal of Human-Computer Studies* 2000; **53**: 715–739.
- Baldonado MQW, Woodruff A, Kuchinsky A. Guidelines for using multiple views in information visualization. *AVI '00: Proceedings of the Working Conference on Advanced Visual Interfaces*, ACM Press: New York, 2000; 110–119.
- Stolte C, Tang D, Hanrahan P. Multiscale visualization using data cubes. *Proceedings Information Visualization*, ACM Press: New York, 2002; 7–14.
- Aiken A, Chen J, Stonebraker M, Woodruff A. Tioga-2: a direct manipulation database visualization environment. *Proceedings of the 12th International Conference on Data Engineering 1996* (New Orleans), February 1996; 208–217.
- Keim DA, Kriegel H-P, Seidl T. Visual feedback in querying large databases. *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on 25–29 October 1993* (San Jose, CA), 1993; 158–165.
- Wise JA, Thomas JJ, Pennock K, Lantrip D, Pottier M, Schur A, Crow V. Visualizing the non-visual: spatial analysis and interaction with information from text documents. *INFOVIS '95: Proceedings of the 1995 IEEE Symposium on Information Visualization 1995* (Washington, DC, USA), IEEE Computer Society: Silver Spring, MD, 1995; p 51.
- Brewer C. *Designing Better Maps: A Guide for Gis Users*, Environmental Systems Research, 2004.