Recovering Reasoning Process From User Interactions

Wenwen Dou[*]
UNC Charlotte
Viscenter

Dong Hyun Jeong[†]
UNC Charlotte
Viscenter

Felesia Stukes[‡]
UNC Charlotte
HCI Lab

William Ribarsky[§]
UNC Charlotte
Viscenter

Heather Richter Lipford[¶]
UNC Charlotte
HCI Lab

Remco Chang[||]
UNC Charlotte
Viscenter

Abstract

With visual analytical tools becoming more sophisticated and prevalent in the analysis communities, it is now apparent that understanding how analysts utilize these tools is more important than ever. Such understanding can lead to improving the tools, but a more subtle and equally important aspect lies in the discovery of the analysts' reasoning process for solving complex problems through the use of these visual analytical tools. In this paper we demonstrate that we were able to identify several of the strategies, methods, and findings of an analysis process using a financial visual analytical tool through the examination of an analyst's interaction log. In our study, we recorded the interactions and think-alouds of 10 financial analysts in a fraud detection task. By examining their interaction logs, we are able to quantitatively show that 60% of strategies, 60% of methods, and 79% of findings could be recovered through the use of two visual analytic log analysis tools.

1 Introduction

In the short number of years since the establishment of the visual analytics research agenda, visual analytical tools have already made an impact in the intelligence and analysis communities. However, until recently, most of the research in visual analytics has focused on the techniques and methods for refining these tools, with the emphasis on empowering the analysts to make discoveries faster and more accurately. While this emphasis is relevant and necessary, we propose that it is not always the end product that matters. Instead, we argue that the process in which an analyst takes to arrive at the conclusion is just as important as the discoveries themselves. By understanding *how* an analyst performs a successful investigation, we will finally be able to start bridging the gap between the art of analysis and the science of analytics.

Unfortunately, understanding an analyst's reasoning process is not a trivial task, especially since most researchers rarely have access to analysts performing analytical tasks using classified or highly confidential material. While there has been a recent increase of activity in the visual analytics community to help analysts document and communicate their reasoning process during an investigation, there is still no clear method for capturing the reasoning processes with minimal cognitive effort from the analyst. This raises the question we look to address in this paper: how much can an analyst's strategy, methods, and findings using a visual analytical tool be recovered?

It is our hypothesis that when interacting with a well-designed visual analytical tool, a large amount of an analyst's reasoning process is embedded within his interactions with the tool itself. Therefore, through careful examination of the analyst's interaction logs, we propose that we should be able to retrieve a great deal of the analyst's reasoning process. To validate our hypothesis, we designed a study to quantitatively measure if an analyst's strategies, methods, and findings can be recovered through human examination of his interaction logs. Our study consists of four stages: user observation, transcribing, coding, and grading. In the user observation stage, we invited 10 financial analysts to use a financial visual analytical tool called WireVis [1] to identify potentially fraudulent wire transactions within a synthetic dataset in think-aloud sessions. The analysts' interactions were logged into file and at the same time their think-alouds captured on video and audio. These information were transcribed by the authors later into files that collectively were considered to be representative of the analysts' reasoning processes and used as the "ground truth" for the study.

Four coders who are students familiar with the WireVis tool examined each analyst's interaction log using two log analysis tools (Operation and Strategic Analysis tools) that we developed [8]. Through visual inspection and analysis of each analyst's interaction log, the four coders were asked to annotate what they believed the analysts' strategies, methods, and findings were. We then compared the coders' inferences with the ground truth, and the result became the basis of our claim on the types and amount of an analyst's reasoning process that were recoverable through the examination of interaction logs.

The result of our study has been most encouraging. Aside from a few specific, low-level types of findings, the four coders (who are not trained in financial fraud detection) were able to correctly retrieve 60% of the analysts' strategies, 60% of the methods, and 79% of the findings. This result indicates that an analyst's strategies, methods, and findings in using a visual analytical tool are indeed recoverable through human examination of interaction log. It is relevant to note that the extracted reasoning process is solely based on the analyst's activities within a visual analytical tool and does not include the overall intelligence analysis that often involves multiple tasks and tools such as searching through websites, phone discussions, the use of additional software, etc. However, our findings represent an important aspect of the intelligence analysis, and provides an example for visual analytics as a community to uncover a new path towards better understanding and capturing of an analyst's reasoning processes.

2 Related Work

We roughly categorize the current research in visualization and visual analytics for capturing the reasoning process of an analyst into two groups: capturing the user's interactions and interactive construction of the reasoning process using a visual tool.

2.1 Capturing User Interactions

Capturing user interactions for the purpose of understanding the user's behavior is very common both in academics and industry. Commercially, there are many off-the-shelf applications that range from capturing a user's desktop activities such as usability software

[*] e-mail: wdou1@uncc.edu
[†] e-mail: dhjeong@uncc.edu
[‡] e-mail: fcartis@uncc.edu
[§] e-mail: ribarsky@uncc.edu
[¶] e-mail: richter@uncc.edu
[||] e-mail: rchang@uncc.edu

to interactions on a website (which is a common feature in most web servers).

In the field of visualization, one of the most notable systems for capturing and analyzing user activities is the GlassBox system by Greitzer at the Pacific Northwest National Laboratory [5]. The primary goal of the GlassBox is to capture, archive, and retrieve user interactions [2]. However, it has also been shown to be an effective tool for capturing specific types of interactions for the purpose of intelligence analysis [3]. While GlassBox and most usability software are effective tools for capturing user activities, they focus primarily on low level events (such as copy, paste, a mouse click, window activation, etc), whereas the events captured in our system are at a higher level that corresponds directly to the data (such as what transaction the user clicked on). For more information on the differences in these two approaches, see the work by Jeong et al. [8] or work by Heer et al. [6].

More recently, Jankun-Kelly et al. [7] proposed a comprehensive model for capturing user interactions within a visualization tool. Their work is unique in that they focus on capturing the effects of the interactions on the parameters of a visualization. Although it is unclear how this framework supports higher level event capturing, the direction is interesting and could lead to a more uniform way of capturing user interactions.

The systems and approaches above are all proven to be innovative and effective. However, their objectives differ from our goal in that none of these systems fully addressed our question of how much reasoning process can be recovered through the examination of interaction logs. It is with this question in mind that we expand on this area of research to capturing user interactions and look to extract reasoning processes embedded in them.

2.2 Interactive Construction of the Reasoning Process

An alternative approach to retrieving reasoning through interactions is for the analyst to create a representation of the reasoning process (usually in the form of a node-link diagram) while solving a complex task. There are a few recent systems in this domain, most notably the Aruvi framework by Shrinivasan and van Wijk [11], which contains three main views, data view, navigation view, and knowledge view. Data view is the visual analytical tool itself, navigation view is a panel for visually tracking the user's history, and lastly the knowledge view allows the user to interactively record his reasoning process through the creation of a node-link diagram.

Similar to the Aruvi framework, the Scalable Reasoning System (SRS) by Pike et al. [10] allows its users to record their reasoning processes through the creation of node-link diagrams. However, unlike the Aruvi framework, the SRS focuses on the collaborative aspects of organizing the reasoning processes among multiple users and sharing their results across the web.

Most recently, Heer et al. [6] created a tool for visualizing users' histories within the commercial visualization tool Tableau [9]. Although the emphasis of this work is not on constructing or visualizing the reasoning process, the functionalities within the tool that allows for a user to edit and modify his interaction history could be used towards communicating his reasoning process effectively.

While there has not been a formal comparison between interactively constructing the reasoning process as mentioned above and our method of analyzing interaction logs, we hypothesize that the cognitive load of having to perform analytical tasks while maintaining and updating a representation of the reasoning process could be tiring [4]. We believe that the systems mentioned above will have better representations of the user's reasoning process. However, we argue that a transparent, post-analysis approach offers an alternative that can achieve comparable results without the efforts from the analysts. Most likely the best solution is somewhere in between, and we look forward to analyzing the pros and cons of the two approaches.

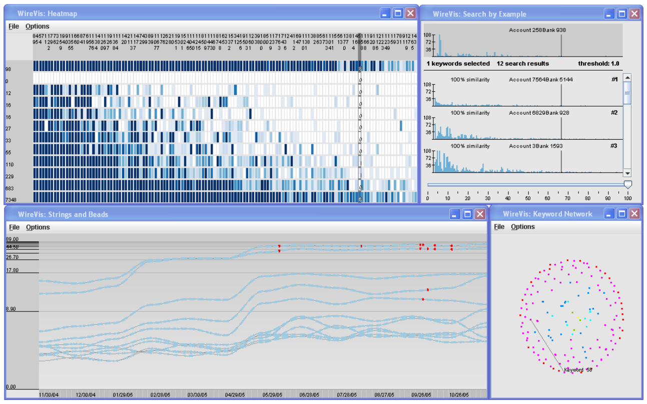

WireVis: Heatmap
File Options
WireVis: Search by Example
Account 258Bank 938
1 keywords selected 12 search results threshold: 1.0
100% similarity Account 7564Bank 5144 #1
100% similarity Account 6829Bank 928 #2
100% similarity Account 3Bank 1593 #3
WireVis: Strings and Beads
File Options
11/30/04 12/30/04 01/29/05 02/28/05 03/30/05 04/29/05 05/29/05 06/28/05 07/28/05 08/27/05 09/26/05 10/26/05
WireVis: Keyword Network
File Options
Keyword 58

Figure 1: Overview of WireVis. It consists of four views including a heatmap view (top left), a time-series view (bottom left), a search by example view (top right), and a keyword relation view (bottom right).

3 WireVis Interactions

We conducted our study with a particular visual analytical tool for investigating financial fraud called WireVis that logged all user interactions. We also developed two additional tools for visualizing user interactions within WireVis to help us explore the analyst's activities and reasoning process [8]. We first describe all of these tools before presenting the details of the user study in the next section.

WireVis is a hierarchical, interactive visual analytical tool with multiple coordinated views [1]. This visual analytical tool was developed jointly with wire analysts at Bank of America for discovering suspicious wire transactions. It is currently installed at Bank of America's wire monitoring group, WireWatch, for beta testing. Although it has not been officially deployed, WireVis has already shown capabilities in revealing aspects of wire activities that analysts were not previously capable of analyzing. Through a multiview approach, WireVis depicts the relationships among accounts, time and transaction keywords within wire transactions (see Figure 1).

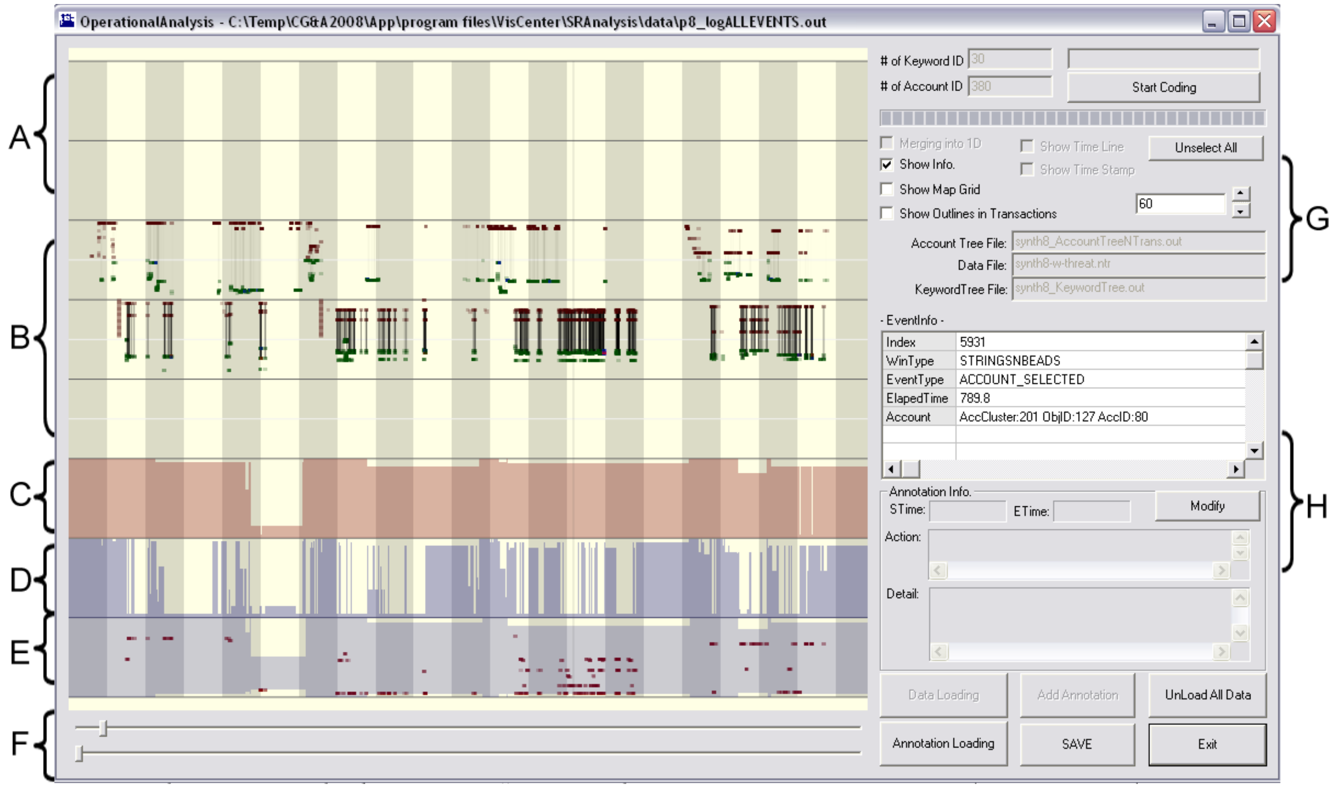
The Operation Analysis Tool (Figure 2) shows the participants' interactions with the view in Wirevis over time. The rows in Figure 2(B) correspond to heatmap view, time-series view and search by example view separately in the WireVis tool. In addition, the depth of the analysis is shown via the number of transactions that are visible, as well as the areas the user is exploring [8]. For example, Figure 2(B) shows that this analyst never used the search by example tool, but instead utilized the time-series view extensively. Similarly Figure 2(C) shows that the user drilled down into specific accounts approximately six minutes into the analysis.

The Strategic Analysis Tool (Figure 3) shows the set of actions taken in achieving a particular goal, without regard to the particular path taken. The visualization uses a treemap to show the transactions grouped by time, then by keyword, and finally by accounts. A cell on the visualization represents a transaction, and the size of the colored circle indicates the time the participant's investigation included that transaction. For example, Figure 3 shows an analysis that focuses on two particular accounts.

4 Evaluation

We conducted a user study to determine how much of an analyst's reasoning process can be recovered using just the captured user interactions. We evaluated this recovery in a quantitative fashion by comparing the process that was inferred by a set of coders against the ground truth determined from videos of the exploration process.

Four stages are designed as user observation, transcribing, coding, and grading. The comprehensive information of each stage is



OperationalAnalysis - C:\Temp\CG&A2008\App\program files\VisCenter\SRAnalysis\data\p8_logALLEVENTS.out

# of Keyword ID 30

# of Account ID 380

□ Merging into 1D
☑ Show Info.
□ Show Map Grid
□ Show Outlines in Transactions
□ Show Time Line
□ Show Time Stamp

60

Account Tree File: synth8_AccountTreeNTrans.out

Data File: synth8-w-threat.ntr

KeywordTree File: synth8_KeywordTree.out

- EventInfo -

Index	5931
WinType	STRINGSNBEADS
EventType	ACCOUNT_SELECTED
ElapedTime	789.8
Account	AccCluster:201 ObjID:127 AccID:80

Annotation Info.

STime:

ETime:

Action:

Detail:

Figure 2: Overview of the Operation Analysis Tool. (A) shows the potential area for adding annotations. (B) shows the participant's interactions with the three views in WireVis (the three rows from top row to bottom correspond to heatmap, time-series, and search by example views respectively). (C) represents the depths of a participant's investigation. (D) shows the areas of the participant's investigation, and (E) the time range. Sliders in (F) control the time scale, while checkboxes in (G) change various visualization parameters. (H) shows the detail information of a participant's selected interaction element.

provided in the following subsections.

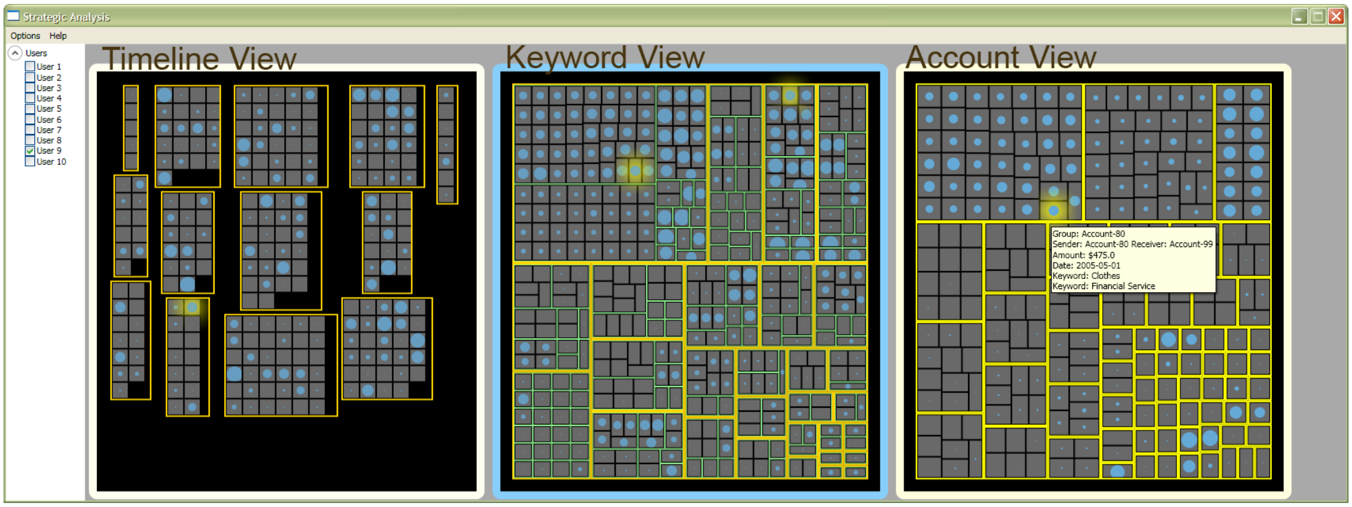
4.1 User Observation

In order to understand the user's reasoning process through his interactions, we first conducted a qualitative, observational study of users analyzing data with WireVis. We recruited 10 financial analysts with an average of 9.9 years (and a median of 8 years) of financial analysis experience who all worked in large financial firms in our area. All of the participants were either currently working as a financial analyst or had professional financial analyst experience. Eight of the users were professionally trained to analyze data for the purpose of fraud detection. Of the 10 analysts, six analysts were male and four were female.

To preserve the privacy of Bank of America and their individual account holders, we created a synthetic dataset for the purpose of this study. Although none of the transactions in the dataset are real, we captured as many characteristics and statistics from real financial transactions as we could and modeled the synthetic data as closely to the real one as possible. The dataset was designed to be simple enough that users were able to look for suspicious transactions within the time frame of a study, but was complex enough that interesting and complicated patterns could be found. This dataset contained 300 financial transactions, with 29 keywords. Some keywords were the names of countries, such as Mexico, and others were goods or services, such as Software or Raw Minerals. We also developed four threat scenarios and injected a total of nine cases we deemed suspicious into the dataset. The threat scenarios included transactions in which keywords should not appear together, accounts with dual roles, keywords with unusually high transaction amounts, and accounts with suspicious transactional patterns appearing over time. More details of the synthetic dataset and sample threat scenarios can be found in [8].

At the beginning of the study session, each participant was asked to fill out a demographic form and was then trained on the use of WireVis for approximately 10 minutes. The participant was also provided a one-page overview of the functionality of WireVis and encouraged to ask questions. Following the training, the user was asked to spend 20 minutes using WireVis to look through the dataset to find suspicious activities. We asked the participant to think-aloud to reveal his strategies. We specifically encouraged the participant to describe the steps he was taking, as well as the information used to locate the suspicious activities. Once the user drilled down to a specific transaction, he was asked to write it down on a Discovery Sheet for the purpose of recording and reporting his findings. Once the user documented a specific transaction, he was encouraged to continue looking for others until the time limit was reached. After the exploration process, a post-session interview was conducted for the participant to describe his strategies and additional findings.

Several methods were used to capture each participant's session as thoroughly as possible. Commercial usability software was used to capture the screen. A separate microphone was used to record the user's audio during the session. Lastly, functions built into the WireVis system captured the user's interaction with the tool itself as information relevant only to the WireVis system. Instead of recording every mouse movement or keystroke, WireVis captures events that generate a visual change in the system. For example, a mouse movement that results in highlighting a keyword in the Heatmap view will generate a time-stamped event noting that the user has



☐ User 1
☐ User 2
☐ User 3
☐ User 4
☐ User 5
☐ User 6
☐ User 7
☐ User 8
☑ User 9
☐ User 10

Group: Account-80
Sender: Account-80 Receiver: Account-99
Amount: $475.0
Date: 2005-05-01
Keyword: Clothes
Keyword: Financial Service

Figure 3: The left view shows transactions grouped by time, middle view shows grouping by keywords, and the right view shows grouping by accounts. The patterns in the account view indicate that the primary strategy employed by this participant was to examine two specific accounts (located on the top of the Account View).

highlighted a specific keyword.

4.2 Transcribing

The video and the think-aloud of each participant were used to create a detailed textual timeline of what each participant did during their session, along with the participant's self-reported reasoning and thinking process. While the created textual timeline is an interpretation and might not perfectly reflect the (internal) reasoning process of the participant, it was created based on the facts recovered from video and audio with conscious efforts in minimizing human bias. We therefore consider the resulting transcript to represent the "ground truth" of what each participant did during their analysis with WireVis.

During the transcribing stage, different strategies, methods, and findings in investigating fraudulent activities were identified to serve the grading process later. Specifically, we identified the following in the transcript:

- A *"Finding"* represents a decision that an analyst made after a discovery.
- *"Strategy"* is used to describe the means that the analyst employed in order to arrive at the finding.
- Also, the link between "finding" and "strategy" is captured by *"method"* which focuses on what steps the analyst adopted to implement the strategy for discovering the finding.

In a typical investigation, an analyst's *strategy* might be to search for a specific suspicious keyword combination based on his domain knowledge. For example, the analyst might determine accounts and transactions involving both the keywords Mexico and Pharmaceutical to be potentially suspicious. Using this strategy, the *methods* employed by this analyst could then be comprised of a series of actions such as highlighting or filtering those keywords, and drilling down to specific accounts and transactions. At the end of the investigation, the analyst would record his *findings* based on the encountered account numbers and transaction IDs along with their decision about whether the particular finding is suspicious or not.

4.3 Coding of the interaction logs through visual examination

We asked several people familiar with WireVis to view each participants' interactions and determine their reasoning. Specifically, we recruited four "coders" from our university, all of whom were familiar with WireVis (three male, one female). They then used the two interaction log analysis tools (Operation and Strategic Analysis tools) to view participant interactions, and created an outline of what occurred.

We first gave all coders comprehensive training on how to use the Operation Analysis Tool and Strategic Analysis Tool to examine the interaction logs of each analyst's investigations. We also provided a guideline of hierarchical coding procedures, asking coders to, in free-text format, provide hierarchical annotations within the visual analytical tools. The hierarchies are reflected as different levels of decision points and strategies extracted by the coders. We asked coders to identify and label findings, strategies, and methods for each analyst. In addition, coders were encouraged to annotate on the transitions if they could discover relationships between each decision point such as one strategy leads to multiple findings or one finding transforms to a new strategy.

All findings from the coders were recorded as annotations and linked to corresponding interaction events and time range. Each coder went through the 10 analysts'interaction logs one by one using the visual analytical tools, spending an average of 13.15 minutes reconstructing each analyst's reasoning process. Thus, at the end of the coding phase, we collected 10 sets of annotations from each coder, resulting in 40 sets of annotations overall.

4.4 Grading

We then compared the annotations the coders produced to the "ground truth" to determine how much of the reasoning process was able to be reconstructed by the coders. The comparisons are graded according to a set of pre-determined criteria by one of the authors, which we describe below.

The categories we used in the grading were in accordance with both transcribing and coding: finding, strategy and method. Generally speaking, "strategy" and "finding" do not necessarily have a one-to-one mapping relationship since some strategies may lead to multiple or null findings. But one "finding" always comes with a "method" in the sense that a method is always needed to make a decision.

For each finding, strategy, and method, we graded according to the following criteria: "Correctly Identified", "Incorrectly Identified", "False Detections" and "Never Identified". This combination was chosen because the four measurements covered all possible scenarios and yet were explicitly distinguishable. "Incorrectly Identified" indicated that a coder noticed some meaningful interactions but incorrectly interpreted them, while "False Detections" captured the scenarios in which a coder thought that certain action took place but in fact there was none. "Never Identified" involved actions that took place, but were not noticed or annotated by the coders.

	Ground Truth			C1	C2	C3	C4
P1	Finding	6	Correctly Identified	5	3	4	5
			Incorrectly Identified	0	1	0	1
			False Detections	0	0	2	0
			Never Identified	1	2	2	0
	Strategy	3	Correctly Identified	3	3	1	0
			Incorrectly Identified	0	0	1	3
			False Detections	0	0	1	1
			Never Identified	0	0	1	0
	Method	6	Correctly Identified	6	4	3	3
			Incorrectly Identified	0	0	1	2
			False Detections	0	0	0	0
			Never Identified	0	2	2	1
	Time Spent (min)			14.7	33.39	10.27	35.9

Figure 4: Grading results of participant 1. A participant's analysis process is separated into findings, strategies, and methods. This figure shows the results of four coders' annotations and how they match the participant's analysis according to the four grading criteria: correctly identified, incorrectly identified, false detections, and never identified.

Figure 4 illustrates the overall criteria used for grading. We determined that a "finding" was correct as long as the coders correctly identified there was a decision made during the analyst's investigation. But we did not ask them to determine what the outcome of that decision was (whether the certain transaction is suspicious, not suspicious or inconclusive). Additionally, if only a part of the coder's annotation was correct, for example if he determined a "strategy" was looking for five incompatible keywords but only identified four keywords correctly, we graded that annotation as "Incorrectly Identified". This purpose for such a strict grading criteria is to minimize potential bias in the grading process.

5 Results

Both the quantitative and the observational results we obtained from grading are rich and informative. In this section, we first demonstrate quantitatively the amount of reasoning that can be extracted from analyzing interaction logs. We then describe some of the trends and limitations of the coding process using our interaction log analysis tools.

5.1 How much reasoning can we infer?

Figure 5 shows the average accuracy of each coder's reconstructed reasoning processes of all participants. The results are separated into three categories as described in section 4.2: findings, strategies and methods. The results indicate that it is indeed possible to infer reasoning from user interaction logs. In fact, on average, 79% of the findings made during the original investigation process could be recovered by analyzing the captured user interactions. Similarly, 60% of the methods and 60% of the strategies could be extracted as well with reasonable deviation between the coders.

An interesting observation is that all coders performed better in extracting findings than strategies or methods. We will discuss a possible explanation for this phenomenon in section 6.

Across Participants A different perspective from which to examine the results is to look for variations in accuracy across the 10 participants. Figure 6 shows the average accuracy of the coders in recovering the reasoning processes behind the 10 participants. This result indicates that there is a noticeable difference between accuracies in extracting reasoning processes for different participants. This finding leads to the conclusion that there are some analysis processes that are more difficult to follow than others. Although there is no definitive answer to why this is, our own investigation suggests that there are two plausible contributors. The first is the difference in experience in financial fraud detection between our participants and our coders. Since our coders have no training in fraud detection, it is natural that some of the strategies and methods in investigative processes are lost to them.

Another cause of this variation is manifested in the acute drop in the accuracy when extracting "methods" from P2 and P4's analysis as shown in Figure 6. As the figure suggests, the coders were baffled by the methods of these two participants. Upon investigation in the video of the participant's analysis process, we discovered that participants 2 and 4 focused their analysis on the irregularities in the time-series view in WireVis. Specifically, they closely examined "spikes" in the view (Figure 7) which indicate sudden increases in amounts or frequencies of wire transactions. Our coders had no way of seeing these visual patterns, so they were not able to identify the methods behind the participants' analyses.

Considering False Detections Since the purpose of this study is to figure out *how much* of the reasoning process can be extracted from interaction logs, we have reported the accuracy based purely on the number of "correctly identified" elements. However, it is relevant to make note of the number of times that our coders made detections that turn out to be inaccurate. Under our grading scheme, the number of annotations made by a coder often exceeds the number of elements in the transcription due to the false detections. For example, the grading result of participant 1 in Figure 4 shows that the number of "findings" in the ground truth is 6, however, coder 3 made a total of 8 annotations. He correctly identified 4 of the 6 elements, missed on identifying 2 of the 6 elements, and falsely detected 2 times when there were no corresponding elements in the ground truth.

With the "false detections" in mind, we re-examine the accuracy of the coders based not on how much of the reasoning process can be recovered, but on the accuracy of their annotations. Figure 8 shows the result of the coders' accuracies that include the coders' false detections. Not surprisingly, the accuracy of the coders all decrease slightly. The accuracy in extracting findings drop by 3% from 79% to 76%, strategies by 5% from 60% to 55%, and finally methods by 2% from 60% to 58%.

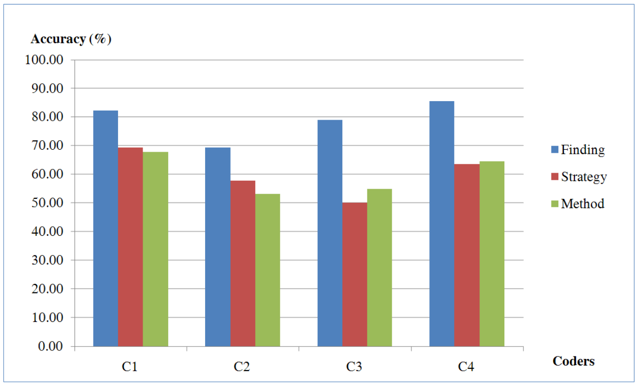


Figure 5: The average accuracy of the four coders correctly identifying "findings", "strategies" and "methods" of all ten participants.

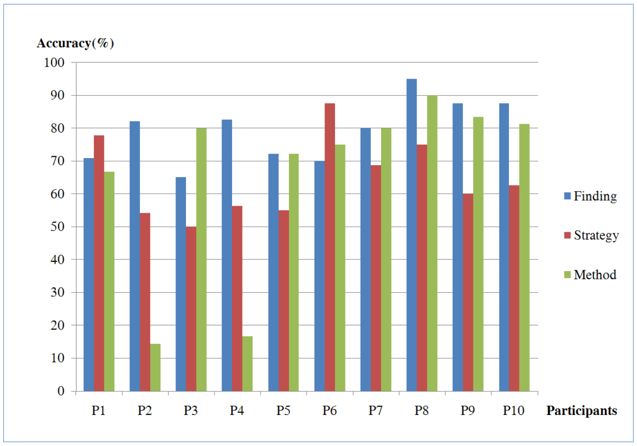


Figure 6: The average accuracy of correctly identifying “findings”, “strategies”, and “methods” based on the 10 participants.

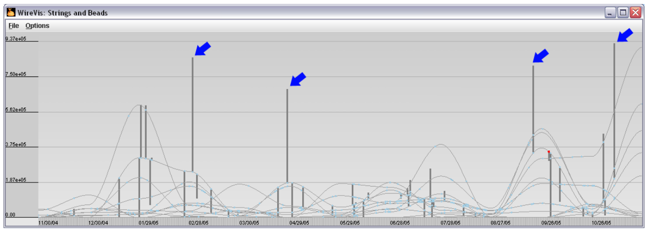


Figure 7: The time-series view in WireVis showing spikes that indicate sudden increases in the amounts or frequencies of wire transactions.

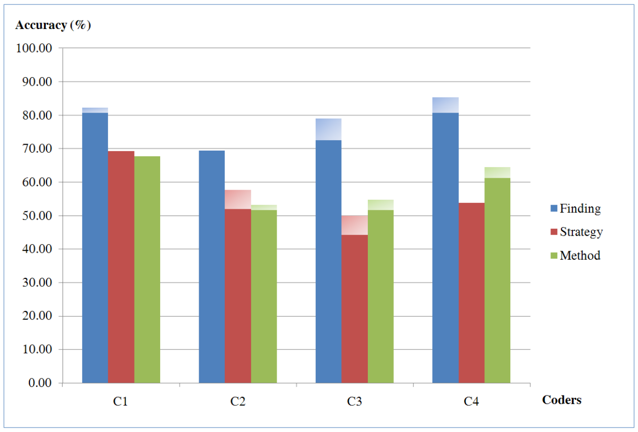


Figure 8: The accuracy of the coder’s annotations in matching up to the ‘findings”, “strategies”, and “methods” of the analyses. The semi-transparent areas indicate the decrease in accuracy compared to Figure 5. The difference between the two figures is that Figure 5 indicates the amount of reasoning that can be recovered, where as this figure shows how accurate the coders’ annotations are.

5.2 Amount of time spent by coders

One important aspect in extracting reasoning process is the amount of time necessary for analyzing the interaction logs. In this section, we discuss the effect of time spent by a coder in analyzing an individual interaction log, as well as the learning effect that the coders exhibit after gaining proficiency in extracting the participants’ reasoning processes.

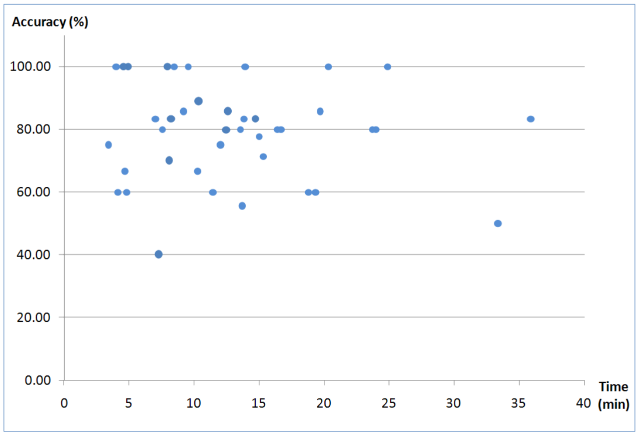


Figure 9: The accuracy of the coders in recovering ”findings” of the participants and the amount of time spent.

Capturing time spent by a coder Built into our Operation and Strategy Analysis tools is the ability to track the amount of time that a coder spends using the tools. The coders were made aware of this feature and were told not to take breaks during an analysis. Since the coders directly annotated their discoveries into the Operation Analysis tool, the system was able to record the amount of time spent by each coder when analyzing an interaction log.

Furthermore, the system tracked when the coder started and stopped the annotations. The purpose of this feature was to separate the time spent in analyzing the interaction log from the time spent in annotating. On average, the coders spend 23.9 minutes analyzing one interaction log, of which 10.75 minutes were spent on annotation and the remaining 13.15 minutes on investigation.

Time spent vs accuracy We examine the relationship between the time spent by a coder and accuracy. Overall, there is no correlation between the two. Figure 9 plots the relationship between the coders’ time spent in analysis (not including time spent for annotation) and their accuracies in extracting “findings”. With the exception of the two outliers in the far right, it appears that the coders are consistently successful when spending anywhere from 5 to 15 minutes. This suggests that spending more time in the analysis does not always yield better results. The two outliers represent the analysis of coders 2 and 4 in their first investigation (participant 1). As we will show in the following section, all coders become more proficient in their analysis as they gain experience.

Increase in accuracy As shown in Figure 6, the accuracy of the coders increase as they gain experience in investigating interaction logs as all four coders began with examining participant 1’s interactions and end with participant 10’s. Based on analyses using Pearson’s correlation coefficient, we find that the number of participants a coder has examined is positively correlated to the coder’s accuracy. This correlation is statistically significant when extracting “findings” $(r(40) = .37, p < .05)$ and “methods” $(r(40) = .52, p < .01)$. Only in extracting “strategies” is the correlation weaker $(r(40) = .21, p = .182(preferred))$. While the sample size is relatively small, these statistics nonetheless imply a subtle but potentially important discovery: with more experience in analyzing interaction logs, a coder could become more proficient in extracting an analyst’s reasoning process.

6 Discussion

The study described in this paper is complex and intricate. On top of involving real financial analysts, the transcription process, the coding, and the grading were all performed with great care and consideration. Although many of the nuances encountered during the study do not affect the results and therefore have not been described in this paper, there are some findings that might be of interest to the community. First of all, during our informal debriefing of the coders, the coders discussed the strategies that they employed in analyzing the analysts' interaction logs. It turned out that our coders often began their investigation by looking for "gaps" in the timeline of the operational view (Figure 2), which are the byproducts of the analysts taking time to write down their findings in the Discovery Sheet (section 4). Based on the gaps, the coders looked for the analysts' findings, and then worked backwards to discover the strategies and methods used to derive the findings.

While this strategy may seem specific to this study and non-generalizable, we argue that in a real life scenario, analysts either directly make annotations in the visualization to make note of a finding, or they write down their finding on a piece of paper for future reference. Either way, there will exist a visible marker that suggests a relevant discovery by the analyst. Therefore, while we did not anticipate this strategy by the coders, we find their quick adoption of this method to identify the analysts' findings to be effective and relevant.

A second interesting trend pointed out by our coders concerns the usefulness of our visual tools for depicting the operational and strategic aspects of the analysis (section 3). According to the coders during the debriefing, all of them used the Operational Analysis tool first to gain an understanding of the overall impression of an analyst's interactions. However, the Strategic Analysis tool is often utilized to examine a specific sequence of interactions when the interactions appear random and jumbled. By presenting the results of the interactions from three perspectives (accounts, keywords, and time) in the Strategic tool, the coder could often identify the focus and intent behind the series of interactions. This finding not only validates our design of the tools, but also reconfirms the importance of visualizing both the strategic and operational aspects of an analysis process. In fact, most of the coders began their investigation by identifying the "findings" through looking for gaps in the interactions, followed by looking for "strategies" through examining the overall visual patterns in both the Strategic and Operational Analysis tools without focusing on individual user interactions. Finally, "methods" were extracted through the use of the Operational Analysis tool where specific interactions were examined in detail.

One last relevant aspect of our study is the measurement of "incorrectly identified" elements in the grading process. In all of our results shown in section 5, we do not take into account elements that have been graded as "incorrectly identified." As mentioned in section 4.4, any annotation by a coder that does not perfectly match the transcription is considered to be incorrectly identified. This includes scenarios in which a coder identifies the analyst's strategy to be examining 4 keywords when in fact the analyst was examining 5, or when a coder determines that the finding of the analyst is a transaction between accounts A and B instead of accounts A and C. If we were to give half a point to these incorrectly identified elements, the overall accuracy of extracting strategies increases drastically from 60% to 71%, methods from 60% to 73%, and findings from 79% to 82%.

7 Future Work

As mentioned in section 5.1, when analysts make their investigation purely based on visual patterns, our coders have a difficult time determining the methods behind the investigation. Although some of the coders' errors in extracting the analysts' reasoning process can be attributed to expected operator errors, the most consistent and common errors stem from the coders not being able to see the same visual representations as the analysts. This observation reveals a potential pitfall of only examining interaction logs without considering the visual representations, which is that for visualization systems that are less interactive, our proposed approach is likely to be ineffective. Understanding how interactivity vs. visual representation affects reasoning extraction remains an open question that we are still investigating.

One practical solution to the problem is to connect the Operational Analysis tool directly to the video of the analysis. With the Operational Analysis tool functioning as an overview, it allows the coder to only review videos of segments of interactions that are ambiguous to them. If an analyst were to use the Operational Analysis tool to aid the recall of his own analysis process, the video could further serve as a record of the details of the original investigation.

By combining video with the Operational Analysis tool, we believe that coders can achieve a higher degree of accuracy and in turn be able to derive winning strategies of different analysts that lead to the same findings. By combining all of these winning strategies, we wish to identify critical decision points that are shared by these strategies and be able to uncover the necessary reasoning process for identifying a particular type of fraudulent activity.

Lastly, we would also like to analyze the difference between groups of participants with diverse backgrounds. Our previous study involved participants who are not trained in financial fraud detection [8]. While they were also able to point out suspicious events and activities, we wish to compare their decisions with the findings of real financial analysts. If we can discover some common pitfalls in novice analysts' reasoning process, we believe that we can create better training tools to help these novices become proficient faster.

8 Conclusion

The path to perfectly capture an analyst's reasoning process is still elusive. However, in this paper, we have demonstrated that it is indeed possible to extract a great deal of the reasoning process through the visual examination of the analyst's interactions with a financial visual analytical tool. Our results indicate that with careful design in capturing user interactions and the use of both operational and strategic tools to visually analyze an analyst's interaction logs, we can understand the strategies, methods, and findings of an analytical process. The implication of this finding could be significant. To name a few, these findings can lead to principles for building better visual analytical tools. Also, we can study the winning strategies recovered using our method to assist other analysts in their investigations, or training novice analysts. While we have not fully considered all potential applications of our discovery, we nonetheless believe that our finding has the potential of uncovering a rewarding path towards deeper and more meaningful understanding of the relationship between the art of analysis and the science of visual analytics.

References

[1] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 155–162, 30 2007-Nov. 1 2007.

[2] P. Cowley, J. Haack, R. Littlefield, and E. Hampson. Glass box: capturing, archiving, and retrieving workstation activities. In *CARPE '06: Proceedings of the 3rd ACM workshop on Continuous archival and retrival of personal experences*, pages 13–18, 2006.

[3] P. Cowley, L. Nowell, and J. Scholtz. Glass box: An instrumented infrastructure for supporting human interaction with information. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05*, pages 296c–296c, January 2005.

[4] T. M. Green, W. Ribarsky, and B. Fisher. Visual analytics for complex concepts using a human cognition model. *Visual Analytics Science and Technology, 2008. VAST 2008. IEEE Symposium on*, pages 91 – 98, November 2008.

[5] F. Greitzer. Methodology, metrics and measures for testing and evaluation of intelligence analysis tools. PNWD-3550, Battelle-Pacific Northwest Division, Richland, WA, 2005.

[6] J. Heer, J. D. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189 – 1196, 2008.

[7] T. Jankun-Kelly, K.-L. Ma, and M. Gertz. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):357–369, March/April 2007.

[8] D. H. Jeong, W. Dou, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Evaluating the relationship between user interaction and financial visual analysis. *Visual Analytics Science and Technology, 2008. VAST 2008. IEEE Symposium on*, pages 83 – 90, November 2008.

[9] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.

[10] W. Pike, R. May, and A. Turner. Supporting knowledge transfer through decomposable reasoning artifacts. In *40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007*, pages 204c–204c, January 2007.

[11] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1237–1246, New York, NY, USA, 2008. ACM.