Recovering Reasoning Processes from User Interactions

Wenwen Dou, Dong Hyun Jeong, Felesia Stukes, William Ribarsky, Heather Richter Lipford, and Remco Chang ▪ *University of North Carolina, Charlotte*

Understanding how analysts use visual-analytics (VA) tools can help reveal their reasoning processes when using these tools. By examining analysts' interaction logs, the authors identified the analysts' strategies, methods, and findings when using a financial VA tool.

In the few years since the establishment of the visual-analytics (VA) research agenda, VA tools have already made an impact in the intelligence and analysis communities. However, until recently, most of the research in VA has focused on the techniques and methods for refining these tools, with the emphasis on empowering analysts to make discoveries faster and more accurately. Although this emphasis is relevant and necessary, we argue that the process through which an analyst arrives at the conclusion is just as important as the discoveries themselves. Understanding how an analyst performs a successful investigation will finally let us 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 because few researchers have access to analysts performing their tasks using classified or highly confidential material. Although there has been a recent increase in activity in the VA community to help analysts document and communicate their reasoning processes during investigations (see the "Related Work in Visualization and Visual Analytics" sidebar, page 54), there is still no clear method for capturing the reasoning processes with minimal cognitive effort from the analyst. This article seeks to determine how much an analyst's strategy, methods, and findings using a VA tool can be recovered.

We hypothesize that when an analyst interacts with a well-designed VA tool, much of that analyst's reasoning process is embedded within his or her interactions with the tool. Therefore, through careful examination of the analyst's interaction logs, 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 whether an analyst's strategies, methods, and findings can be recovered through human examination of these interaction logs.

WireVis Interactions

We conducted our study using WireVis, a hierarchical, interactive VA tool that logs all user interactions.[1] WireVis 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 revealed aspects of wire activities that analysts were not previously able to analyze. Through a multiview approach, the tool depicts the relationships among accounts, time, and transaction keywords within wire transactions (see Figure 1).

We also developed two tools for visualizing user interactions within WireVis to help us explore analysts' activities and reasoning processes.[2] The *operational-analysis tool* (see Figure 2) shows participants' interactions with the view in WireVis over time. The rows in Figure 2b correspond to the 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 visible transactions, as well as the areas the user is exploring.[2] For example, Figure 2b shows that this analyst never used the search-by-example tool but instead used the time-series view extensively. Similarly, Figure 2c shows that the user drilled down into specific accounts approximately six minutes into the analysis.

 Published by the IEEE Computer Society

The *strategic-analysis tool* (see Figure 3, next page) shows the set of actions taken to achieve a particular goal, without regard to the path taken. The visualization uses a tree map 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.

Evaluation

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

The evaluation consisted of four stages: user observation, transcribing, coding, and grading.

User Observation

To understand the user's reasoning process through his or her interactions, we 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. All of the participants were either working as financial analysts or had professional financial-analyst experience. Eight of the users were professionally trained to analyze data or to detect fraud. Of the 10 analysts, six were male and four were female.

To preserve the privacy of Bank of America and its individual account holders, we created a synthetic data set for the purpose of this study. Although none of the transactions in the data set are real, we captured as many characteristics and

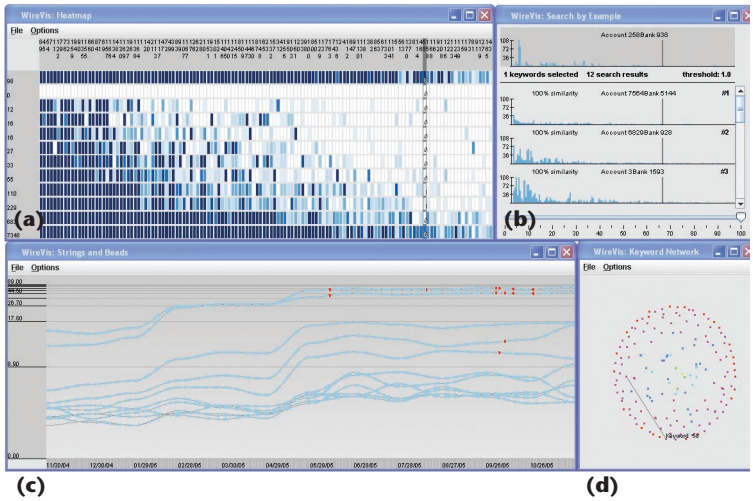


Figure 1. An overview of WireVis. The visual-analytics tool consists of four views: (a) a heatmap view, (b) a search-by-example view, (c) a time series view, and (d) a keyword relation view.

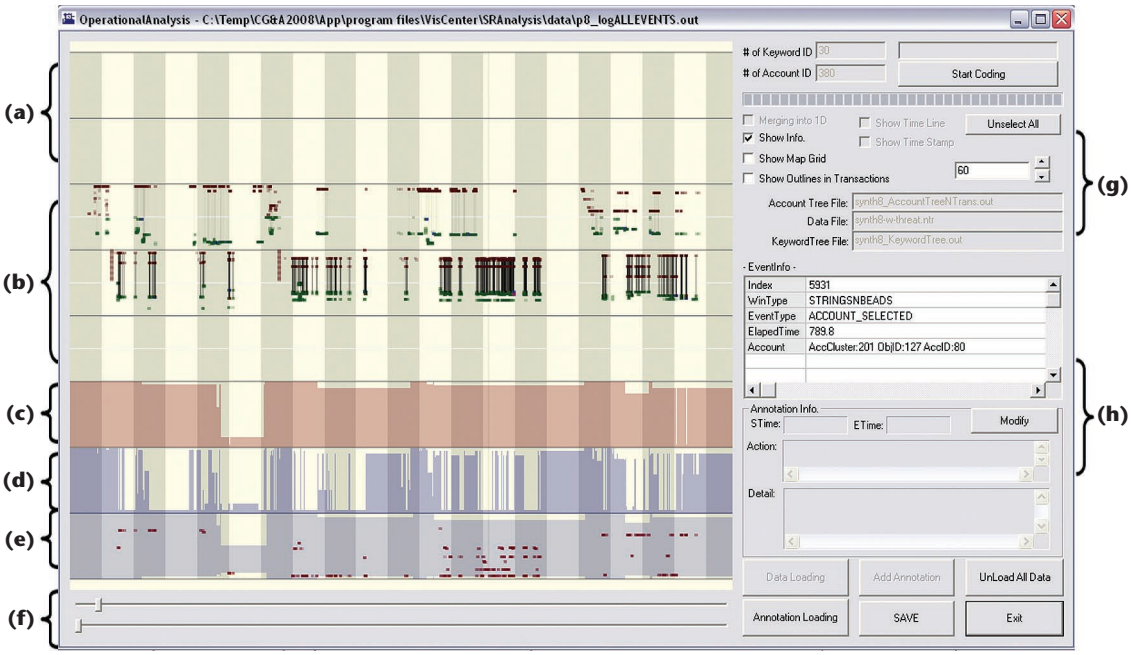


Figure 2. Overview of the operational-analysis tool: (a) a potential area for adding annotations; (b) the participant's interactions with the three views in WireVis (the three rows, from top to bottom, correspond to the heatmap, time series, and search-by-example views). Other highlighted areas include the (c) depths of a participant's investigation, (d) areas of the participant's investigation, and (e) time range. In addition, (f) sliders control the time scale, while (g) checkboxes change various visualization parameters. Another view (h) shows detailed information about the participant's selected interaction element.

Related Work in Visualization and Visual Analytics

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

Capturing User Interactions

Capturing user interactions for the purpose of understanding the user's behavior is common in both academics and industry. Some commercial-off-the-shelf applications capture a user's desktop activities (such as usability software), whereas others capture interactions on a Web site (a common feature in most Web servers).

In the visualization field, Frank Greitzer's Glass Box system is one of the most notable systems for capturing and analyzing user activities.[1] Its primary goal is to capture, archive, and retrieve user interactions.[2] However, it is also an effective tool for capturing specific types of interactions for the purpose of intelligence analysis.[3] Although Glass Box and most usability software are effective tools for capturing user activities, they focus primarily on low-level events (such as copy, paste, mouse clicks, and window activation). The events captured in our system are at a higher level that corresponds directly to the data (such as the transaction the user clicked on). More information on the differences between these two approaches is available elsewhere.[4]

More recently, T.J. Jankun-Kelly and his colleagues proposed a comprehensive model for capturing user interactions within a visualization tool.[5] 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.

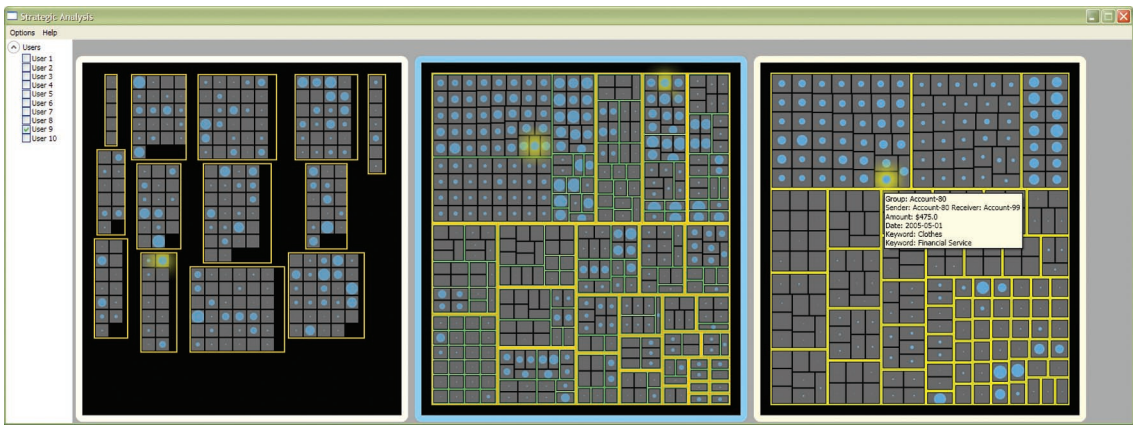
These systems and approaches are all innovative and effective. However, their objectives differ from our goal in that none of them fully address how much of a reasoning process can be recovered through the examination of interaction logs.

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. Recent systems in this domain include the Aruvi framework,[6] which contains three main views:

- The *data* view is the VA tool itself.
- The *navigation* view is a panel for visually tracking the user's history.
- The *knowledge* view lets the user interactively record his or her reasoning process through the creation of a node-link diagram.

The Scalable Reasoning System (SRS) also lets users record their reasoning processes through the creation of node-link diagrams.[7] However, unlike the Aruvi framework, SRS focuses on the collaborative aspects of organiz-



(a) (b) (c)

Figure 3. Strategic-analysis tool views: transactions grouped by (a) time, (b) keywords, and (c) accounts. The patterns in the account view indicate that the primary strategy used by this participant was to examine two specific accounts (located on the top of the view).

statistics from real financial transactions as we could and modeled the synthetic data as closely to the real one as possible. The data set was designed to be simple enough that users could look for suspicious transactions within the study's time frame, but complex enough that interesting and complicated patterns could be found. This data set 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 data set. The threat scenarios included transactions in which keywords should not appear together, accounts with dual roles, keywords with unusually

ing the reasoning processes among multiple users and sharing their results across the Web.

Most recently, Jeffrey Heer and his colleagues[4] created a tool for visualizing users' histories within the commercial visualization tool Tableau.[8] Although the work's emphasis is not on constructing or visualizing the reasoning process, the functionalities within the tool that let users edit and modify their interaction history could be used to effectively communicate their reasoning processes.

Although there has not been a formal comparison between interactively constructing the reasoning process 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.[9] The systems we describe here will have better representations of the user's reasoning process. However, a transparent, postanalysis approach offers an alternative that can achieve comparable results without the analysts' efforts. The best solution is most likely somewhere in between, and we look forward to analyzing the pros and cons of these approaches.

References

1. F. Greitzer, *Methodology, Metrics and Measures for Testing and Evaluation of Intelligence Analysis Tools*, tech report PNWD-3550, Battelle-Pacific Northwest Division, 2005.
2. P. Cowley et al., "Glass Box: Capturing, Archiving, and Retrieving Workstation Activities," *Proc. 3rd ACM Workshop Continuous Archival and Retrieval of Personal Experiences* (CARPE 06), ACM Press, 2006, pp. 13–18.
3. P. Cowley, L. Nowell, and J. Scholtz, "Glass Box: An Instrumented Infrastructure for Supporting Human Interaction with Information," *Proc. 38th Ann. Hawaii Int'l Conf. System Sciences* (HICSS 05), IEEE CS Press, 2005, article 296c.
4. J. Heer et al., "Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 6, 2008, pp. 1189–1196.
5. T. Jankun-Kelly, K.-L. Ma, and M. Gertz, "A Model and Framework for Visualization Exploration," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 2, 2007, pp. 357–369.
6. Y.B. Shrinivasan and J.J. van Wijk, "Supporting the Analytical Reasoning Process in Information Visualization," *Proc. 26th Ann. SIGCHI Conf. Human Factors in Computing Systems* (CHI 08), ACM Press, 2008, pp. 1237–1246.
7. W. Pike, R. May, and A. Turner, "Supporting Knowledge Transfer through Decomposable Reasoning Artifacts," *Proc. 40th Ann. Hawaii Int'l Conf. System Sciences* (HICSS 2007), IEEE CS Press, 2007, article 204c.
8. J. Mackinlay, P. Hanrahan, and C. Stolte, "Show Me: Automatic Presentation for Visual Analysis," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 6, 2007, pp. 1137–1144.
9. T.M. Green, W. Ribarsky, and B. Fisher, "Visual Analytics for Complex Concepts Using a Human Cognition Model," *Proc. IEEE Symp. Visual Analytics Science and Technology* (VAST 08), IEEE CS Press, 2008, pp. 91–98.

high transaction amounts, and accounts with suspicious transactional patterns appearing over time. More details about the synthetic data set and sample threat scenarios are available elsewhere.[2]

At the beginning of the study session, participants were asked to fill out a demographic form and were then trained on the use of WireVis for approximately 10 minutes. Participants were also provided with a one-page overview of the functionality of WireVis and encouraged to ask questions. After the training, participants were asked to spend 20 minutes using WireVis to look through the data set for suspicious activities. We asked participants to think aloud to reveal their strategies. We specifically encouraged them to describe the steps they were taking, as well as the information used to locate the suspicious activities. Once the users drilled down to a specific transaction, they were asked to record and report their findings on a discovery sheet. Once they documented a specific transaction, they were encouraged to continue looking for others until they reached the time limit. After the exploration process, participants were asked to describe their strategies and additional findings in a postsession interview.

Several methods were used to capture each session as thoroughly as possible. Commercial usability software captured the screen. A separate microphone recorded the user's audio during the session. Finally, functions built into the WireVis system captured the user's interaction with the tool as information relevant only to the WireVis system. Instead of recording every mouse movement or keystroke, WireVis captures events that generate visual changes in the system. For example, a mouse movement that highlights a keyword in the heatmap view will generate a time-stamped event.

Transcribing

Participants' videos and "think-alouds" were used to create a detailed textual timeline of what participants did during their sessions, along with their self-reported reasoning and thinking process. Although the created textual timeline is an interpretation and might not perfectly reflect the participant's (internal) reasoning process, it was created on the

basis of the facts recovered from video and audio with conscious effort to minimize human bias. We therefore consider the resulting transcript to represent the ground truth of what each participant did during the WireVis analysis.

During the transcribing stage, we identified different *findings*, *strategies*, and *methods* analysts used when investigating fraudulent activities:

- A finding represents a decision that an analyst made after a discovery.
- A strategy is the means the analyst employed to arrive at the finding.
- A method captures the link between finding and strategy. It focuses on the 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 on the basis of his or her domain knowledge. For example, the analyst might determine accounts and transactions involving two keywords, "Mexico" and "pharmaceutical," to be potentially suspicious. Using this strategy, the methods employed by this analyst could consist 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 or her findings based on the encountered account numbers and transaction IDs along with the decision about whether the particular finding was suspicious.

Coding

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 used the operation and strategic-analysis tools to view participant interactions and created an outline of what occurred.

We gave all coders comprehensive training on how to use the operation and strategic-analysis tools to examine analysts' interaction logs. We also provided a guideline of hierarchical coding procedures, asking coders to provide hierarchical annotations in free-text format within the VA 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 the transitions if they could discover relationships between each decision point, such as one strategy leads to multiple findings or one finding transforms into a new strategy.

All findings from the coders were recorded as annotations and linked to corresponding interaction events and time ranges. Each coder went through the 10 analysts' interaction logs one by one using the VA 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.

Grading

Our next step was to compare the annotations that the coders produced to the ground truth to determine how much of the reasoning process the coders could reconstruct. The comparisons are graded according to a set of predetermined criteria.

The categories we used in 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 because some strategies might 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*—that is, a coder both noticed and correctly identified a meaningful finding, strategy, or method;
- *incorrectly identified*—that is, a coder noticed some meaningful interactions but incorrectly interpreted them;
- *false detections*—that is, a coder thought that a certain action took place when in fact none had; and
- *never identified*—that is, actions took place but were not noticed or annotated by the coders.

We chose this combination because the four measurements cover all possible scenarios and yet are explicitly distinguishable.

Table 1 illustrates the overall grading criteria. We determined that a finding was correct as long as the coders correctly identified that there was a decision made during the analyst's investigation. But we did not ask them to determine the outcome of that decision (whether the transaction is suspicious, not suspicious, or inconclusive). Additionally, if only a part of the coder's annotation was correct—for example, if the coder determined that a strategy was to look for five incompatible keywords but only identified four keywords cor-

rectly—we graded that annotation as incorrectly identified. Such strict grading criteria minimizes potential bias in the grading process.

Results

The quantitative and observational results obtained from grading are rich and informative. We demonstrate quantitatively the amount of reasoning that can be extracted from analyzing interaction logs and describe some of the coding process's trends and limitations.

How Much Reasoning Can We Infer?

Figure 4 shows the average accuracy of each coder's reconstructed reasoning processes for all participants. The results indicate that it is possible to infer reasoning from user interaction logs. In fact, on average, 79 percent of the findings made during the original investigation process could be recovered by analyzing the captured user interactions. Similarly, 60 percent of the methods and 60 percent of the strategies could be extracted, with reasonable deviation between the coders.

Across participants. A different perspective from which to examine the results is to look for variations in accuracy across the 10 participants. Figure 5 (next page) shows the average accuracy of the coders in recovering the 10 participants' reasoning processes. This result indicates a noticeable difference between accuracies in extracting reasoning processes for different participants. This finding leads us to conclude that some analysis processes are more difficult to follow than others. Although there is no definitive explanation for this difference, our investigation suggests two plausible contributors.

The first factor is the difference in experience with financial fraud detection between the participants and coders. Because our coders have no fraud detection training, it is natural that some of the strategies and methods in investigative processes are lost to them.

The second reason for this variation is manifested in the acute drop in accuracy when extracting methods from the analyses of participants 2

Table 1. Grading results for participant 1.

Analyst activity	Ground truth	Grading criteria	Coder 1	Coder 2	Coder 3	Coder 4
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 (minutes)			14.7	33.39	10.27	35.9

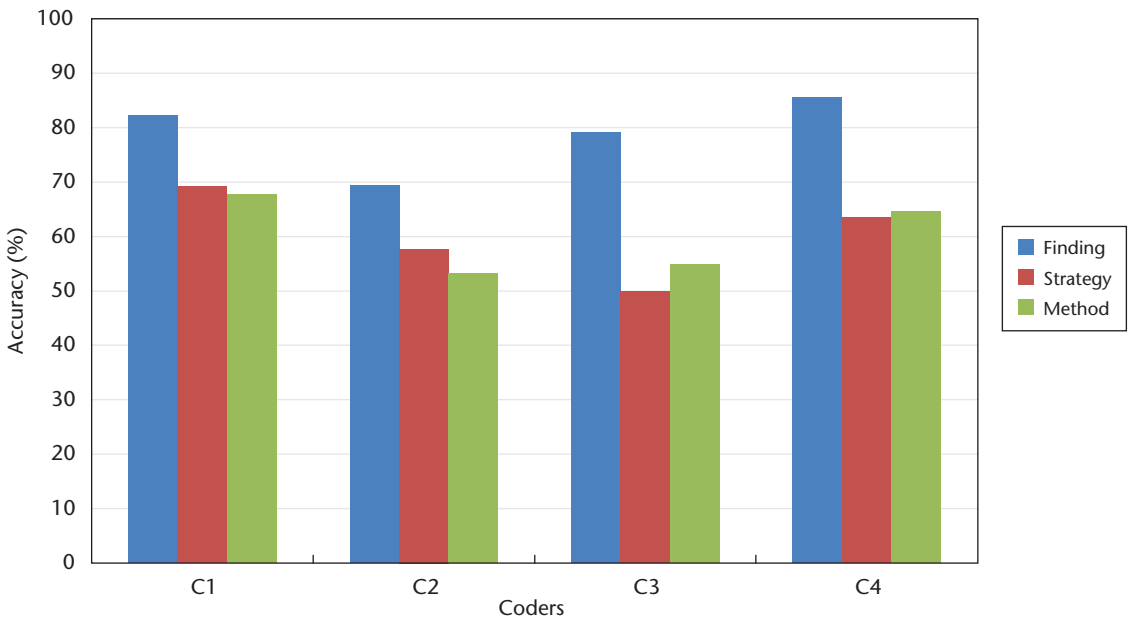


Figure 4. The average accuracy of the four coders in correctly identifying findings, strategies, and methods of the 10 participants. Analyzing the captured user interactions recovers an average of 79 percent of the findings, 60 percent of the methods, and 60 percent of the strategies, with reasonable deviation between coders.

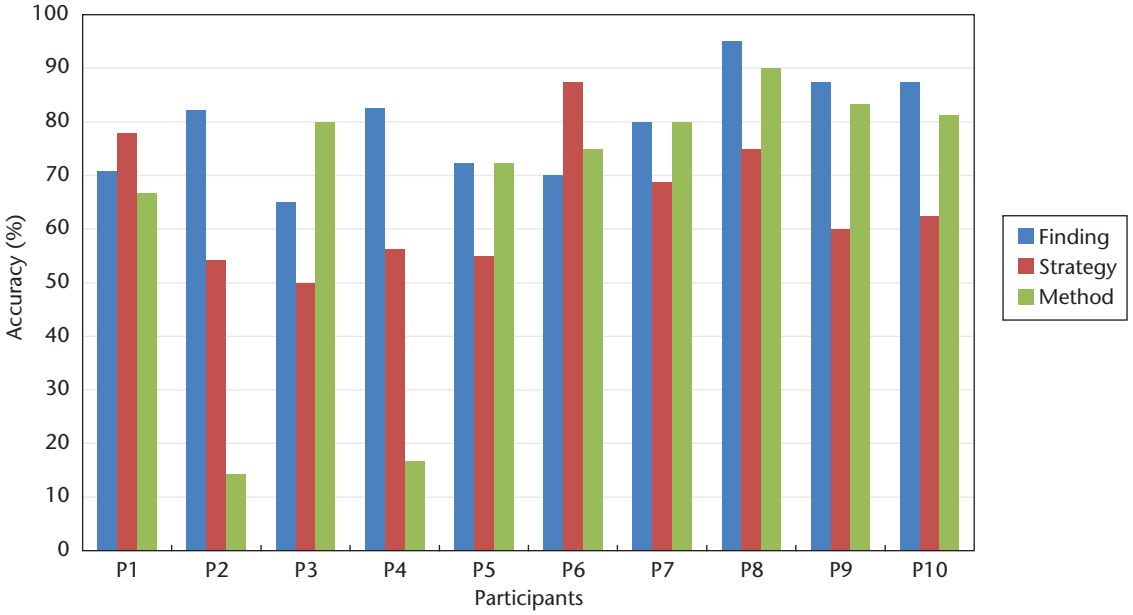


Figure 5. The average accuracy of correctly identifying findings, strategies, and methods based on the 10 participants. As the graph shows, accuracy levels in extracting reasoning process for different participants varied significantly. We hypothesize that this variance is due both to differences in experience between participants and coders and to the difficulty of following some analysis processes.

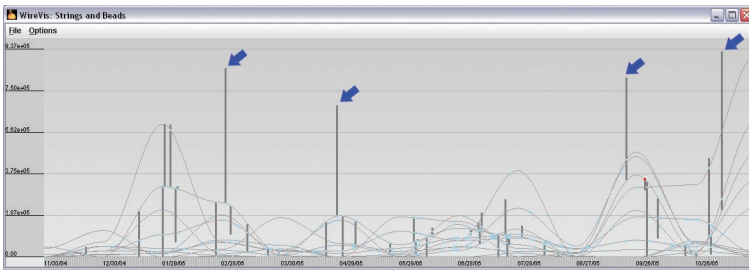


Figure 6. The time series view in WireVis showing spikes that indicate sudden increases in the amounts or frequencies of wire transactions. Because the coders could not see these visual patterns, they could not determine the methods behind the participants' analysis processes.

and 4, as shown in Figure 5. As the figure suggests, the coders were baffled by the methods of these two participants. Upon investigating the video of these participants' analysis process, we discovered that participants 2 and 4 focused on the irregularities in the WireVis time series view. Specifically, they closely examined spikes in the view (see Figure 6), which indicate sudden increases in amounts or frequencies of wire transactions. Our coders had no way of seeing these visual patterns, so they could not identify the methods behind the participants' analyses.

Considering false detections. Because this study aims to determine 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 note how often our coders' detections turned out to be inaccurate. Under our grading scheme, the number of annotations made by a coder often exceeded the number of elements in the transcription because of the false detections. For example, the grading result for Participant 1 in Table 1 shows that the number of findings in the ground truth is 6; however, Coder 3 made a total of eight annotations. This coder correctly identified four of the six elements, missed identifying two of the six elements, and made two false detections.

With the false detections in mind, we reexamine the coders' accuracy not on the basis of how much of the reasoning process can be recovered, but on the basis of the accuracy of their annotations. Figure 7 shows the result of the coders' accuracies that include the coders' false detections. Not surprisingly, the accuracy of all the coders decreases slightly. Their accuracy in extracting findings drops by 3 percent, from 79 to 76 percent; strategies by 5 percent, from 60 to 55 percent; and finally methods, by 2 percent from 60 to 58 percent.

Amount of Time Spent by Coders

One important aspect in extracting reasoning processes is the amount of time needed to analyze the interaction logs.

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. Because the coders directly annotated their discov-

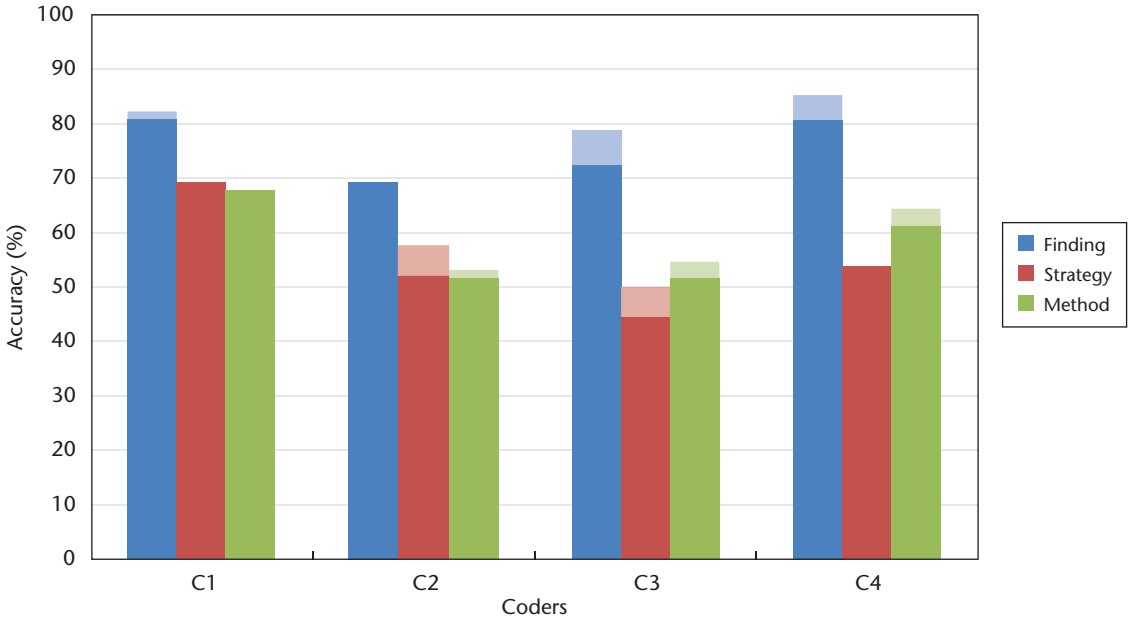


Figure 7. The accuracy of the coder's annotations in matching the analysts' findings, strategies, and methods. The semitransparent areas indicate the decrease in accuracy compared to Figure 4. Whereas Figure 4 indicates the amount of reasoning that can be recovered, this figure shows the accuracy of the coders' annotations.

eries into the operational-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 analyzing the interaction log from the time spent annotating. On average, the coders spent 23.9 minutes analyzing one interaction log. Of this amount, 10.75 minutes were spent on annotation and the remaining 13.15 minutes on investigation.

Time spent versus accuracy. Overall, there is no correlation between the time spent by a coder and accuracy. Figure 8 plots the relationship between the coders' time spent in analysis (not including time spent for annotation) and their accuracy in extracting findings. With the exception of the two outliers on the far right, it appears that the coders are consistently successful when spending anywhere from five 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). All coders became more proficient in their analysis as they gained experience.

Increase in accuracy. As Figure 5 shows, the coders' accuracy increases as they gain experience investigating interaction logs. All four coders began with examining Participant 1's interactions and ended with Participant 10's. Using Pearson's correlation coefficient, we find that the number of participants a coder has examined is positively correlated to the

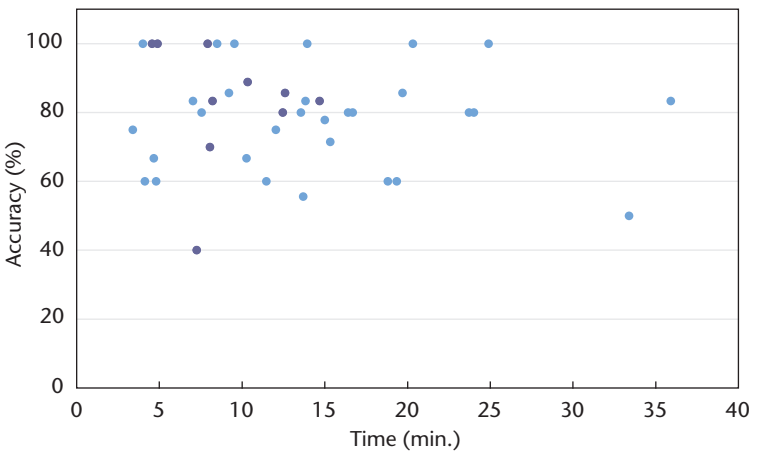


Figure 8. The accuracy of the coders in recovering findings of the participants and the amount of time spent. The results show that coders generally can accurately extract findings when they spend between five and 15 minutes on the analysis.

coder's accuracy. This correlation is statistically significant when extracting findings ($r(40) = 0.37$, $p < 0.05$) and methods ($r(40) = 0.52$, $p < 0.01$). Only in extracting strategies is the correlation weaker ($r(40) = 0.21$, $p = 0.182$). Although the sample size is relatively small, these statistics imply a subtle but potentially important discovery: with more experience analyzing interaction logs, a coder could become more proficient in extracting an analyst's reasoning process.

Findings

The study we've described is complex and intricate. Although many of the nuances encountered during the study do not affect the results and therefore

are not described here, some findings might be of interest to the community. First, during our informal debriefing, the coders discussed the strategies that they used in analyzing participants' interaction logs. The coders often began their investigations by looking for gaps in the operational-view timeline (Figure 2), which are the by-products of the analysts taking time to write their findings on the discovery sheet. The coders used these gaps to look for the analysts' findings and then worked backward to discover the strategies and methods used to derive the findings.

Although this strategy might seem specific to this study and nongeneralizable, in a real-life scenario, analysts either directly make annotations in the visualization to note a finding, or write 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, although we did not anticipate this strategy by the coders, we found 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 analysis's operational and strategic aspects. All of the coders used the operational-analysis tool first to get an overall impression of an analyst's interactions. However, the strategic-analysis tool is often used 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 looking for gaps in the interactions to identify findings. Next, they looked for strategies by examining the overall visual patterns in the strategic- and operational-analysis tools without focusing on individual user interactions. Finally, methods were extracted using the operational-analysis tool, in which specific interactions were examined in detail.

One last relevant aspect of our study is the measurement of incorrectly identified elements in the grading process. None of our results account for elements that have been graded as incorrectly identified. As we mentioned earlier, 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 four keywords when in fact the analyst was examining five, 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 gave half a point to these incorrectly identified elements, the overall accuracy of extracting strategies would increase drastically from 60 to 71 percent, methods from 60 to 73 percent, and findings from 79 to 82 percent.

We found that when analysts based their investigations purely on visual patterns, our coders had a difficult time determining their methods. Although some of the coders' errors in extracting the analysts' reasoning process can be attributed to 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 examining interaction logs without considering the visual representations. Our current approach is thus probably limited to highly interactive visualization systems. Understanding how interactivity versus visual representation affects reasoning extraction remains an open question.

One practical solution is to connect the operational-analysis tool directly to the video of the analysis. With the operational-analysis tool functioning as an overview, coders can review videos of segments of interactions that are ambiguous to them. If analysts used the operational-analysis tool to aid the recall of their 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 these winning strategies, we hope to identify critical decision points that these strategies share and uncover the necessary reasoning process for identifying a particular type of fraudulent activity.

Finally, we plan to analyze the difference between groups of participants with diverse backgrounds. Our previous study involved participants who are not trained in financial fraud detection.[2] Although 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 processes, we believe

we can create better training tools to help these novices become proficient more quickly.

Acknowledgments

We thank Xiaoyu Wang, Caroline Ziemkiewicz, and Lane Harrison for their help coding the analysts' reasoning processes. We performed this research with support from the National Visualization and Analytics Center (NVAC), a US 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, a US Department of Energy Office of Science laboratory.

References

1. R. Chang et al., "WireVis: Visualization of Categorical, Time-Varying Data from Financial Transactions," *Proc. IEEE Symp. Visual Analytics Science and Technology* (VAST 07), IEEE CS Press, 2007, pp. 155–162.
2. D.H. Jeong et al., "Evaluating the Relationship between User Interaction and Financial Visual Analysis," *Proc. IEEE Symp. Visual Analytics Science and Technology* (VAST 08), IEEE CS Press, 2008, pp. 83–90.

Wenwen Dou *is a PhD student in the Department of Computer Science at the University of North Carolina, Charlotte, and a member of the Charlotte Visualization Center. Her research interests include visual analytics, user interaction, information visualization, and visual reasoning. Dou has a BS in telecommunication engineering from the Beijing University of Posts and Telecommunications. Contact her at wdou1@uncc.edu.*

Dong Hyun Jeong *is a PhD student in the Department of Computer Science at the University of North Carolina, Charlotte. His research interests include virtual reality, information visualization, and visual analytics. Jeong has an MS in computer science from Hallym University. Contact him at dhjeong@uncc.edu.*

Felesia Stukes *is a PhD student in the Department of Software and Information Systems at the University of North Carolina, Charlotte. Her research focus is human-computer interaction. Stukes has an MS in information systems from the University of North Carolina, Charlotte. Contact her at fcartis@uncc.edu.*

William Ribarsky *is the Bank of America Endowed Chair in Information Technology at the University of North Carolina, Charlotte, and the founding director of the Charlotte Visualization Center. His research interests include visual analytics, 3D multimodal interaction, bioinformatics visualization, virtual environments, visual reasoning, and interactive visualization of large-scale information spaces. Ribarsky has a PhD in physics from the University of Cincinnati. Contact him at ribarsky@uncc.edu.*

Heather Richter Lipford *is an assistant professor in the Department of Software and Information Systems at the University of North Carolina, Charlotte. Her research area is human-computer interaction, with specific interests in visual analytics and privacy. Lipford has a PhD from the Georgia Institute of Technology's College of Computing. Contact her at heather.lipford@uncc.edu.*

Remco Chang *is a research associate in the Department of Computer Science at the University of North Carolina, Charlotte, and a member of the Charlotte Visualization Center. His research interests include geospatial and urban visualization and analysis, interactions in visual analytics, and information visualization. Chang has an MS in computer science from Brown University. Contact him at rchang@uncc.edu.*