Requirements for Visual Interaction Analysis Systems

Yi Han*
Georgia Institute of Technology

Gregory D. Abowd†
Georgia Institute of Technology

John Stasko‡
Georgia Institute of Technology

Abstract

Designers who deploy visualization applications usually want to assess how those applications are being used in the field. A promising and scalable method for understanding such use is to collect event logs of people's interactions with the applications. The challenge is how to then analyze the interactions in the logs in order to discover insights. Researchers have used visual analytics to support this analyst-driven process with some success. However, we found that existing visual interaction analysis systems are limited in their flexibility, scalability, and generalizability to fully support this challenging task. In this article we identify the primary tasks of interaction log analysis, discuss the main units of analysis, and derive a set of system requirements to inform the design of future visual interaction analysis systems.

Index Terms: H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces

1 Introduction

Designers who deploy visualization applications often seek to assess how those applications are being used in the field. By examining application usage, its designer can begin to understand the usability and utility of the application, learn about its users, and understand usage patterns/analysis methods. One promising source of such information is an interaction log. Modern visualization applications routinely incorporate a multitude of interactions to support the flexible, exploratory analysis processes of their users. While it has become easier to collect interaction logs, it is still challenging to effectively analyze them.

We consider the analysis of visualization application interaction logs to be highly exploratory, and thus closely aligned to other types of sense-making activities that involve repeated cycles of foraging and gathering information, and then reflecting on those findings to generate new schemas and hypotheses about what is actually occurring [20]. An analyst, who typically is the designer of the visualization application, frequently might explore interaction logs without explicit questions or goals in mind, or the goals may change over time. The detailed, subjective, and open-ended nature of the analysis process may be overwhelming, especially to those who rarely conduct this type of analysis.

We believe that these analysis needs suggest a visual analytics solution. Visual analytics is particularly suitable for interaction log analysis because it effectively combines automated computational analysis with human exploration and guidance, especially when applied to large collections of data [15]. Furthermore, a visual analytics approach is helpful when analysis goals are dynamic and possibly imprecise. Unfortunately, existing visual interaction analysis systems are not flexible, scalable, and generalizable enough to support this need. We speculate that this is a reason why interaction data from design studies are seldom extensively analyzed in the visualization literature, despite such data's prominent role in the analysis process.

The objective of this work is to establish a set of requirements for visual interaction analysis systems. We make the following contributions. (1) We identify an analyst's tasks and discuss how these tasks can be accomplished by analyzing interaction logs. (2) We identify and present three analysis units that are essential for the process. (3) For these tasks and analysis units, we derive a set of requirements for visual analytics on interaction logs that emphasize flexibility, scalability, and generalizability.

2 Related Work

Researchers have been using visualizations for interaction log analysis on systems other than visualization applications in a variety of domains for about two decades. The visualizations typically have been static representations of users' interaction patterns using heat maps for general UI interactions [10], trees for web navigations [5, 19, 27, 31], graphs for social network interactions [1, 23], or line and bar charts for online video interactions [6, 16]. While useful, at times such visualizations can be limiting because the view cannot be modified or transformed. Therefore, researchers recently have turned to interactive visualization systems to more dynamically explore interaction data. These interactive visualizations include interconnected bar charts and graphs for web search behaviors [17], timelines for web interactions [18], stacked area charts for online video interactions [25], icicle trees for social network and web interactions [24, 29], connected matrices [32], and even visual clusters of sequences [28]. Although these visualization techniques have been applied to interactions on *non-visualization applications*, we can learn from those experiences and examine which techniques may apply well to interactions on *visualization applications*.

With respect to the use of visualization to analyze interactions on visualization applications, researchers have used static visualizations such as state transition graphs [22], colored bars [14], and graphs and scatterplots [12]. Some used interactive visualizations as well. For example, Jeong et al. [13] created two interactive visualization systems, one for exploring interaction data on a timeline and one in treemaps. Blascheck et al. [2] also designed an interactive visualization system to analyze interactions on a text visualization application. They used a line chart as the primary visualization to show interactions with think aloud and eye movement data. A particular strength of their work is in the computational analysis of the data – The system can automatically identify similarities between users and help analysts find usage patterns. Such computational methods are particularly useful when the amount and variety of interactions are large.

These prior systems depicting visualization application interactions have limitations, however. First, most of these projects were research prototypes that visualize a relatively small-scale interaction dataset [2, 7, 13, 14, 22]. For example, Blascheck et al. [2] only studied 16 participants, each using the visual analysis application once, in a lab. In a real-world deployment, the amount of log data can be significantly larger than what these systems can support. Heer et al. [12] seemed to have visualized a realistic real-world dataset but one of the applied visualization techniques, the behavior graph, did not seem to scale well. A second limitation of these projects was that, during analysis, they each organized interactions into a single, subjectively-determined set of cat-

*e-mail:yihan@gatech.edu
†e-mail:abowd@gatech.edu
‡e-mail:stasko@cc.gatech.edu

egories [7, 12, 13, 14, 22]. These categorizations were often necessary to reduce the complexity of the data for meaningful patterns to emerge. However, most of these projects only selected and used one type of categorization. For example, Reda et al. [22] categorized interactions by whether the interactions significantly changed the visualization layout. In realistic analysis tasks, the categorizations should depend upon and iteratively change with an analyst's goal. Methods for flexibly supporting this goal are largely missing in past work. A third limitation is that many of these visualizations were designed specifically for analyzing interactions from one particular visualization application in one specific study [7, 12, 13]. Some data dimensions were hard-coded into the analysis systems, making results difficult to generalize. For example, Jeong et al.'s systems [13] specifically defined and laid out visualizations based on their data dimensions, such as the views of the analyzed visualization application. This design limits the generalizability of their systems. We believe that to fully address these limitations, one needs to re-examine interaction analysis tasks fundamentally.

3 Interaction Analysis Tasks

Based on our review of past research and our own experiences, we have identified the following key tasks or goals of an analyst when seeking to understand how his/her visualization application is being used.

Assess Usability
An analyst seeks to understand how easy it is to use and learn a visualization application. This information could be gleaned from interactions in many ways. For example, features that are easy to use are likely to be performed more frequently, assuming they are important to the application. If a feature is not employed much but was expected to be frequently used, the feature may not be designed well.

Diving deeper, the learning curve of an application could be assessed from interactions. For example, one hypothesis is that when a user is less familiar with an application, his/her interactions would be more diverse and random, showing an experimental usage of features. But as the user discovers which features are more useful from experience, his/her interactions would become more focused on those features and use them in a consistent manner. By assessing the change trajectory, an analyst could infer the learning curve of his/her visualization application. If an analyst seeks to assess which features are more challenging to learn, he/she may inspect how long on average it takes for a user to go from first encountering a feature to using it efficiently.

An easy-to-use application is often efficient to use. In non-visualization applications, fewer mouse clicks and shorter mouse movement distances in usage sessions are indicators of efficient UI design. However, in visualization applications, when visual exploration is the task and broader understanding of data is the goal, these traditional usability indicators may need to be interpreted differently. A visualization application that encourages extensive interactions may help an application user more easily explore the data.

Assess Utility
Card et al. claimed that "the purpose visualization is insight, not pictures" [4]. A good metric to assess the utility of a visualization application is to examine whether its users are able to find insights. A typical method for determining this answer is to interview users. But can this information be acquired from interaction logs instead as interview data are more difficult to collect? Gotz and Zhou defined a set of interaction types as "Insight Actions" in their work because some interactions could be connected to insights [9]. For example, they called Bookmark and Annotate events "Visual Insight Actions". These interactions could indicate an insight was discovered because finding an insight is one reason for bookmarking and annotating views in visualizations. But not all bookmarks and annotations are indications of insights. For example, an annotation could be used to add missing information to the data. Therefore, to differentiate insight-indicating bookmarks and annotations, an analyst would need to manually examine the bookmarked or annotated content to determine which ones are actually insights. The downside is that this analysis would require a significant effort from the analyst and sometimes the bookmarked or annotated contents are difficult to interpret as they are generated by someone else. As a result, visualization designers who plan to use visualization interactions to infer insights might want to explicitly prompt their users to tag insights in bookmarks and annotations to allow automatic classification of insights from interaction logs. After insights are determined, a simple count of them might be sufficient to approximate the utility of a visualization application.

Learn About Users
An analyst seeks to learn more about a visualization application's users from usage behaviors. For example, which people are "expert" users and which ones need some extra help? For a specific feature, which users are able to properly employ it? Any one feature could be implemented with multiple UI interactions. For example, zooming into a view could be implemented by clicking a Zoom-In button or selecting a Zoom-In menu item. Which method is preferred by users for their day-to-day tasks? When using an application for different types of analysis tasks or occasions, how do the users' behaviors differ? At an abstract level, an analyst seeks to find groups of users or sessions that exhibit certain/varying behaviors (e.g., different keyword search [2]) or examine the behaviors of users or sessions under certain circumstances (e.g., different display sizes [22]). The key to this analysis is to map the user information to the varying interaction patterns in the data. This information is helpful to an analyst for understanding user differences in visualization usages.

Understand Usage Patterns/Analysis Methods
An analyst seeks to explore the variety of ways a visualization application was used. Specifically, frequent ways of using the application, which form usage patterns, are of particular interest. Some of these patterns are expected by the analyst. For example, an analyst of interaction logs from a visual text analysis application would expect its users to extensively read the text documents. But what other usage patterns might there be? Kang et al. found several more specific usage patterns after studying the usages from Jigsaw [14]. For example, some users start from scanning all the documents first to filter out irrelevant ones, then read those remaining documents. Some users repeatedly search the document set with different keywords and read the documents in the search result. These usage patterns might not have been expected by the analyst. Therefore, finding the relative portions of these usage patterns helps the analyst gain a deeper understanding of the varying ways his/her application was operated in actual usage scenarios.

Some usage patterns of a visualization application may indicate that a visual analysis method (VAM) is taken. A VAM, which is sometimes called a visual analysis strategy [7, 14, 22], is a methodological and semantically meaningful way of operating a visualization application. Many researchers look for VAMs in their applications to learn about such semantically meaningful usage patterns. One example of a VAM is the Visual Information-seeking Mantra [26]. This VAM is indicated by a set of usage patterns that start from an "overview", followed by a mixture of "zooming" and "filtering", and then show "details" on demand. Shneiderman identified this VAM as being widely used in a variety of visualization applications. Other VAMs may only occur when certain types of data or visualization techniques are employed. For example, Kang et al. identified a set of VAMs (strategies) for analyzing text documents with a specific set of text visualization techniques [14]. These VAMs are less generalizable but are more contextually relevant to the application. Finding the set of VAMs employed in a visualization application is useful for understanding users' reasoning processes behind the usage patterns [7].

4 Analysis Units

Within the analysis tasks above, we found that an analyst needs to find frequency distributions of not just individual interaction events but also groups of events in categories or sequences. Therefore, identifying these analysis units are vital to the analysis. But how should these events, categories, and sequences be defined and identified? Once they are determined, identifying their frequency distribution over time or any other contextual information should be relatively easy. In this section, we discuss these analysis units.

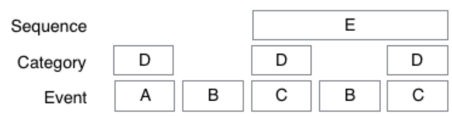

Figure 1: Interaction events. Individual events (ABC) could be categorized (A→D, C→D) or grouped by their sequences (DBD→E).

Event
We assume that individual interaction events are the basic unit logged. As shown in Figure 1, suppose a string of five events (ABCBC) of three types (A, B, and C) are logged in a usage session. Example events may include "clicking the Zoom-In button" or "scrolling the view." Events sometimes include the visualized data as a parameter. Such events are generally defined in the code that produces the log. Individual events can be analyzed directly but typically they are first filtered, categorized, and grouped into more semantically meaningful units to the analyst.

Category
Because a visualization application may have a large number of interactions, an analyst typically organizes them into a smaller set of categories that are more semantically meaningful and suitable to his/her analysis goal. For example, some researchers [11, 21] organized events into the intent-based interaction categories defined by Yi et al. [30]. Other researchers used other classification criteria, such as the view an event occurs in [14] or the significance of a layout change [22]. A categorization is illustrated in the second row of Figure 1 where events A and C are both classified into category D.

Two analyst-driven steps are required in the categorization process. An analyst not only needs to determine which categories to use for the analysis but also how the events should be mapped to the categories. First, the categories to be used should largely depend on the analysis goal. For example, if an analyst wishes to study the uses of different views in a visualization application, the analyst could categorize each interaction event by its view [14]. Second, determining the category that each event should be mapped to requires the analyst's subjective determination. Some determinations are easy, such as whether the interaction is within a specific view. Other determinations are more difficult, such as determining the interaction intent category from Yi et al.'s interaction taxonomy [30]. The more difficult categorizations typically require a semantic interpretation of the events and thus a potentially significant effort from the analyst. This labor requirement can become quite significant as reclassifications may frequently be needed when goals change over the course of analysis. As a result, it is important for an interaction analysis system to be able to flexibly and efficiently support this analyst-driven categorization process.

Sequence
An analyst typically examines sequences of interactions in order to identify longer and higher-level usage patterns (e.g., DBD→E in Figure 1). For example, to determine whether a specific system feature that requires multiple interactions is being used as expected, an analyst may need to look for a specific set of interaction sequences. Frequently occurring interaction sequences are considered particularly informative because they often represent useful or conventional ways of using an application. An interaction sequence may include both consecutive and non-consecutive interactions. For example, a sequence of "Inspect" actions is considered a "Scan pattern" in Gotz and Wen's work [8]. Conversely, an interesting sequence may be made up of non-consecutive interactions. For example, suppose an analyst wishes to determine whether people are following the Visual Information-Seeking Mantra [26] when they use an application. To do so, the analyst does not need to find overview, zoom and filter, and details on demand interactions necessarily occurring back-to-back. The interactions simply need to occur non-consecutively in the proper sequence for the analyst to determine if the mantra has been followed. Therefore, being able to identify sequences that may not only include events occurring back to back increases the flexibility in identifying longer and higher-level usage patterns.

5 System Requirements

Using the tasks and analysis units as a basis, we identify a set of requirements for a visual interaction analysis system to assist with interaction log analysis, specifically for supporting flexibility, scalability, and generalizability.

Support Event Organization (Flexibility)
An interaction analysis system needs to be able to support the flexibile oganization of events. Different analysts and situations may call for different organizations of interaction events because of their varying needs. For example, one analyst may be looking for underused application functions, another may be optimizing the interface, and a third may simply be seeking to understand the visual analysis methods people employ using the application. The same analyst may even change analysis focus "on the fly" as new discoveries about application use are made. Below we describe three specific requirements for flexibily organizating the logged interaction events that we believe an ideal interaction analysis system should provide.

a. Select events of interest
A system may log all interaction events that occur, but only a subset of the logged interaction events may be relevant for a given analysis goal. An analyst needs to be able to flexibly select these relevant events for further exploration. Otherwise, when many irrelevant events are kept in the analysis, the "noise-level" of the data may hide otherwise meaningful patterns. How to determine which events are relevant during an analysis session is very likely a subjective judgment of the analyst based on knowledge of the analysis goal and the interaction data.

b. Define analysis perspective
Analysts may approach log events with a wide variety of goals. Thus, it should be possible for analysts to flexibly define different analysis perspectives for classifying the events based on these different goals. An analysis perspective provides the means to differentiate events that can be based on different criteria. It essentially defines a set of categories for organizing events.

For example, a perspective for understanding visualization application commands and operations that are performed would simply be a set of categories of those commands and operations. One could consider this a relatively low-level perspective. An alternative perspective for understanding which and when application views are used would include a category for each view in the application. Kang et al. [14] employed this perspective when performing an analysis of usage of the Jigsaw visual analytics system. They identified Jigsaw interactions by the views in which they occurred in order to analyze overall strategies taken by each user. Yet another perspective might include a set of categories defined by user intent, that is, what was the intent of the person using the visualization application when performing an interface operation. Pohl et al. [21] used the interaction intent framework introduced by Yi et al. [30] to categorize user interactions with two visualization systems, VisuExplore and Gravi++, in order to explore and compare user strategies when using those systems. Guo et al. [11] also used those intent-based categories plus a new category called "retrieve" as "high-level actions" in their interaction analysis of a visualization application. As a final example analysis perspective, an analyst may be interested in the degree that application usage follows a well-known analysis method, such as Shneiderman's "Overview first, zoom and filter, then details on demand [26]." For this perspective, the set of categories would be those four activities defined in the mantra.

c. Categorize events
Once an analysis perspective is defined, an analyst next assigns the events to categories. For some perspectives, multiple interaction events can be considered "similar" and placed into the same category. For example, interface events such as clicking on buttons, spinning the mouse wheel, or choosing menu commands, when they are relevant to zooming in or out of the visualization, can all be placed into a "zoom" category. This ability to flexibly categorize interaction events is important for log analysis.

Provide Configurable Visualizations (Flexibility)
As discussed in the related work, many researchers have adopted different styles of interactive visualizations for use in their analysis processes. From these efforts, we learned that effective visualizations not only need to provide visual overviews of the interaction data, but also interactive features to support functions such as querying and filtering patterns on demand. Providing an overview of event sequences may be challenging due to size of the interaction logs, however. Moreover, which interactive features to include and how they should be designed in order to give analysts investigative flexibility and strength is still an important research question.

Include Automated Computational Assistance (Scalability)
Usage patterns at a large scale may be difficult to manually identify. Automated computational data analysis is essential to help analysts discover patterns from large interaction logs. For example, it is relatively easy and significantly faster to algorithmically identify and quantify frequent/infrequent events, categories, and sequences. After the computation, the output of these algorithms can be visualized to provide an analyst a summarized view of the information in a large scale interaction dataset. For example, frequent interaction sequences are commonly automatically extracted and visualized to identify higher-level usage patterns [2, 3, 11, 21]. This use of computational analysis significantly increases a person's ability to analyze the interaction data.

Apply to Any Visualization Application (Generalizability)
As mentioned earlier, a number of interaction analysis systems have been built for analyzing *particular* visualization applications. When focusing on only one visualization application, it is easier to custom-design an analysis approach and system with a very specific set of tuned views. However, when designing an analysis system that could be applied to any visualization application's interaction logs, the analysis system needs to be highly configurable to support varying log formats, interaction types, and analysis needs.

6 Conclusion

We presented a set of tasks, analysis units, and system requirements for designing future visual interaction analysis systems. We hope these requirements can bring about discussions on this challenging research topic.

References

[1] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user navigation and interactions in online social networks. *Information Sciences*, 195:1–24, July 2012.

[2] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl. VA2: A Visual Analytics Approach for Evaluating Visual Analytics Applications. *IEEE TVCG*, 22(1):61–70, Jan. 2016.

[3] E. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding Waldo: Learning about Users from their Interactions. *IEEE TVCG*, 20(12):1663–1672, Dec. 2014.

[4] S. K. Card, J. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[5] S. K. Card, P. Pirolli, M. Van Der Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, and J. Boshart. Information Scent As a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability. In *Proc. of ACM CHI '01*, pages 498–505, 2001.

[6] K. Chorianopoulos. Collective intelligence within web video. *Human-centric Computing and Information Sciences*, 3(1):1–16, June 2013.

[7] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. Lipford, and R. Chang. Recovering Reasoning Processes from User Interactions. *IEEE Computer Graphics & Applications*, 29(3):52–61, May 2009.

[8] D. Gotz and Z. Wen. Behavior-driven Visualization Recommendation. In *Proc. of ACM IUI '09*, pages 315–324, 2009.

[9] D. Gotz and M. Zhou. Characterizing user's visual analytic activity for insight provenance. In *Proc. of IEEE VAST '08*, pages 123–130, Oct. 2008.

[10] M. Gray, A. Badre, and M. Guzdial. Visualizing usability log data. In *Proc. of IEEE Symposium on Information Visualization '96*, pages 93–98, 130, Oct. 1996.

[11] H. Guo, S. Gomez, C. Ziemkiewicz, and D. Laidlaw. A Case Study Using Visualization Interaction Logs and Insight. *IEEE TVCG*, 22(1):51–60, 2015.

[12] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE TVCG*, 14(6):1189–1196, Nov. 2008.

[13] D. H. Jeong, W. Dou, H. Lipford, F. Stukes, R. Chang, and W. Ribarsky. Evaluating the relationship between user interaction and financial visual analysis. In *Proc. of IEEE VAST '08*, pages 83–90, Oct. 2008.

[14] Y.-a. Kang, C. Gorg, and J. Stasko. How Can Visual Analytics Assist Investigative Analysis? Design Implications from an Evaluation. *IEEE TVCG*, 17(5):570–583, May 2011.

[15] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Information visualization. chapter Visual Analytics: Definition, Process, and Challenges, pages 154–175. Springer-Verlag, Berlin, Heidelberg, 2008.

[16] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller. Understanding In-video Dropouts and Interaction Peaks Inonline Lecture Videos. In *Proc. of the ACM Conference on Learning @ Scale*, pages 31–40, 2014.

[17] H. Lam, D. Russell, D. Tang, and T. Munzner. Session Viewer: Visual Exploratory Analysis of Web Session Logs. In *Proc. of IEEE VAST '07*, pages 147–154, Oct. 2007.

[18] P. Nguyen, K. Xu, A. Wheat, B. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process Through Analytic Provenance. *IEEE TVCG*, 22(1):41–50, Jan. 2016.

[19] S. Pachidi, M. Spruit, and I. van de Weerd. Understanding users behavior with software operation data mining. *Computers in Human Behavior*, 30:583–594, Jan. 2014.

[20] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of International Conference on Intelligence Analysis*, volume 5, pages 2–5, May 2005.

[21] M. Pohl, S. Wiltner, S. Miksch, W. Aigner, and A. Rind. Analysing Interactivity in Information Visualisation. *KI - Knstliche Intelligenz*, 26(2):151–159, May 2012.

[22] K. Reda, A. E. Johnson, J. Leigh, and M. E. Papka. Evaluating User Behavior and Strategy During Visual Exploration. In *Proc. of ACM BELIV '14*, pages 41–45, 2014.

[23] F. Schneider, A. Feldmann, B. Krishnamurthy, and W. Willinger. Understanding Online Social Network Usage from a Network Perspective. In *Proc. of ACM SIGCOMM '09*, pages 35–48, 2009.

[24] Z. Shen, J. Wei, N. Sundaresan, and K.-L. Ma. Visual analysis of massive web session data. In *Proc. of IEEE LDAV '12*, pages 65–72, Oct. 2012.

[25] C. Shi, S. Fu, Q. Chen, and H. Qu. VisMOOC: Visualizing video clickstream data from massive open online courses. In *Proc. of IEEE VAST '14*, pages 277–278, Oct. 2014.

[26] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. of IEEE Symposium on Visual Languages '96*, pages 336–343, Sept. 1996.

[27] M. Spiliopoulou and C. Pohle. Data Mining for Measuring and Improving the Success of Web Sites. *Data Mining and Knowledge Discovery*, 5(1-2):85–114, Jan. 2001.

[28] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma. Visual cluster exploration of web clickstream data. In *Proc. of IEEE VAST '12*, pages 3–12, Oct. 2012.

[29] K. Wongsuphasawat and J. Lin. Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter. In *Proc. of IEEE VAST '14*, pages 113–122, Oct. 2014.

[30] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE TVCG*, 13(6):1224–1231, Nov. 2007.

[31] O. Zaïane, M. Xin, and J. Han. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. In *Proc. of IEEE International Forum on Research and Technology Advances in Digital Libraries '98*, pages 19–29, Apr. 1998.

[32] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. MatrixWave: Visual Comparison of Event Sequence Data. In *Proc. of ACM CHI '15*, pages 259–268, 2015.