Flexible Organization, Exploration, and Analysis of Visualization Application Interaction Events using Visual Analytics

Yi Han, Gregory D. Abowd, and John Stasko

Fig. 1. Explore view of the visual interaction analysis system, IntiVisor, includes the following views: (B) graph view, (C) usage distribution view, (D) frequent orderings view, and (E) bar view. At the left side is the control panel (A) that supports switching between views.

Abstract—People's interactions with a visualization application can reveal information about the visual analysis methods and reasoning processes they employ. By instrumenting an application with logging code, one can capture an event trace of all the interactions that occur during its use. This type of temporal event log data is typically reorganized into more semantically meangingful units during analysis to support the variety of goals of an analyst. Unfortunately, current visual analytics systems designed for this purpose are limited in this aspect and typically only support one analysis perspective. We present IntiVisor, a visual interaction analysis system that is specifically designed to address the flexible event organization and pattern discovery needs of the interaction analysis process.

Index Terms—Temporal event data, Visual analytics, Interaction

1 Introduction

Interaction events from people's use of a visualization application can be a valuable resource for understanding their analysis methods and reasoning processes [4, 8]. Existing systems for this purpose are superb at helping analysts extract usage patterns, such as frequent sequences, but most fall short at providing a beneficial capability in the analysis process—reorganizing events under a variety of usage perspectives.

Reorganizing a log of events can be useful for many reasons. For example, some events may need to be removed from the analysis because they are irrelevant to the current line of inquiry. Including such events invites unnecessary noise into the data that might overshadow otherwise obvious usage patterns. Further, oftentimes it can be useful to categorize events for analysis. For instance, Pohl et al. [10] and Guo

- *Yi Han is with Georgia Institute of Technology. E-mail: yihan@gatech.edu.*
- *Gregory D. Abowd is with Georgia Institute of Technology. E-mail: abowd@gatech.edu.*
- *John Stasko is with Georgia Institute of Technology. E-mail: stasko@cc.gatech.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

et al. [6] both used Yi et al.'s [12] interaction categories, such as select and filter, to organize their interaction events. This process places the interaction events into a smaller and more semantically meaningful set of categories for further analysis. Conversely, an analyst may seek more details about each event. For example, if an analyst seeks to examine when users zoom a graph to a certain level, this level will need to be extracted from the zoom event parameters. Events organized in different ways can generate vastly different usage patterns. These types of event organizations are typically laboriously conducted as a preprocessing step outside of the analysis system.

We present IntiVisor, a visual interaction analysis system that integrates both pattern extraction tasks and event organization tasks. The system provides a suite of visualizations that can be flexibly employed to iteratively support these tasks.

2 Related Work

Numerous temporal event visualizations have been developed over the years but they typically focused on the visualization aspect of the data analysis process, not the event organization aspect [1]. Event organization has largely been regarded as a data preprocessing step that is out of scope of visualization research. Recently, researchers of visual analytics systems such as EventFlow [9] are beginning to incorporate this vital capability into their visual analytics systems because they have realized the importance of supporting the event organization process on the fly. In one of the EventFlow papers, Monroe et al. specifically discussed the variety of event organization methods in EventFlow for their field study applications. In our work, we focus on the needs of an interaction log analyst and derive a different set of organization methods and visualization techniques to more seamlessly integrate this important aspect of temporal event analysis into an analyst's workflow.

We also found that visual analytics systems specifically designed for interaction log analysis are lacking in this important analysis aspect. Many systems only used one method to organize their events [2,4,7]. For example, Blascheck et al. organized events based on abstract visualization tasks defined by Brehmer and Munzner [3]. As a result, their systems seem to be more for supporting a one-off research analysis, than an iterative, flexible analysis process. This limitation is significant because these systems would not be as useful when an analyst determined that even just one of the categories is not appropriate for a new analysis task.

3 IntiVisor: Visual Interaction Analysis System

Our design goal is to provide *flexibility* to the iterative interaction analysis process through visual analytics. The need for flexibility permeates throughout the process. In this section, we first discuss where this property is required step by step. Next, we present the key features of our visual interaction analysis system, IntiVisor, that implement these flexibility requirements. In this paper, we use the term "analyst" to refer to a user of a visual interaction analysis system, such as IntiVisor, and "user" to someone who interacted with the visualization application whose usage data is being analyzed.

3.1 Event Organization Process

As shown in Figure 2, interaction event data are processed through three steps. The products of these steps are *actions*, *operations*, and *tactics/orderings*. The process is iterative as an analyst can return to any earlier steps to flexibly redefine the products for his/her analysis needs. We will walk though this process step by step from the perspective of generating these products.

Events → Actions

The basic interaction unit logged from a visualization application is an event. The simplest event includes a timestamp, an activity, and a set of parameters. For example, adding a data point to a view includes a timestamp, the "add data point" activity, and what the data point is as the parameter. The first step (Figure 2A) in flexibly organizing events is to determine: (1) Should an event be included in the analysis? Not all events are relevant. (2) Should an event be combined with another event? Some events frequently occur in succession, such as

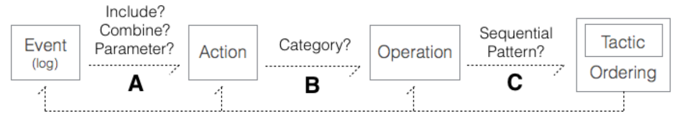

Fig. 2. The flexible and iterative interaction event organization process. (A) Should an event be included, combined, or supplemented with parameters? Selected events are actions. (B) How should an event be categorized? Categorized actions are operations. (C) How should sequential patterns be extracted? Sequences with consecutive operations are tactics and sequences that also include non-consecutive operations are orderings.

mouseover and mouseout a data item, and together represent a more meaningful analysis unit. (3) Should the activity be supplemented with any specific parameters? Only a subset of parameters may be useful for a given analysis. Typical interaction analysis does not even consider parameters [6, 10]. An analyst needs to be able to flexibly make and change these decisions for every round of analysis. After selecting, merging, and adding parameters to events, they become *actions*.

Actions → Operations

Actions are a set of relevant events or event groups (merged) with certain parameters. For example, "add data point (item 1)", where "item 1" is the data point reference, could be an action. Actions are typically organized into a smaller, more meaningful set of categories for further analysis [6, 8, 10]. This step requires another flexible organization decision—how should actions be categorized (Figure 2B)? An analyst needs to select a set of categories and assign actions to them. Frequently used categories include those based on user intent [6, 10] or application feature (view, [8]). The decision needs to be made at the time of the analysis and may frequently change as analysis goals evolve over time. Categorized actions become *operations*.

Operations → Tactics/Orderings

Operations are a set of categories that are semantically meaningful to an analyst for an analysis goal at hand. But individual operations are not sufficient for extracting broader usage patterns. Therefore, an analyst needs to identify sequential patterns of multiple operations. But how should sequential patterns be extracted (Figure 2C)? Two types of sequences are useful for interaction analysis: sequences with and without consecutive operations. Typically, only the first type of sequence is analyzed to discover how users operated a visualization application step by step [2, 5–7, 10]. However, non-consecutive operations could show higher-level analysis methods. For example, the Visual Information-seeking Mantra [11], "Overview first, zoom and filter, then details on demand," is a high-level analysis method that includes a sequence of four operations that do not necessarily need to occur back to back. We call sequences with consecutive operations *tactics* and sequences that also include non-consecutive operations *orderings*. These sequential patterns represent the highest-level products from the event organization process. When events are properly organized, the patterns they form are more meaningful and useful.

3.2 Visualization System Design

We designed IntiVisor to support the event organization process described above (Figure 2). In this section, we present a high-level overview of the views and features in IntiVisor for supporting interaction analysis. The layout of IntiVisor includes a control panel to the left (Figure 1A) and a visualization view to the right. The control panel includes general information about the dataset (e.g., 102 data sessions) and buttons to easily navigate between the views, which facilitate the iterative analysis process. The example interaction data used in the figures of this paper are from the visualization application, CiteVis[1].

[1] CiteVis: http://www.cc.gatech.edu/gvu/ii/citevis/

Events → Actions

The first step in any log analysis process is to examine the logged events to assess the data quality and to determine analysis goals. IntiVisor provides the *Inspect view* to give an analyst a way to become familiar with the dataset in an intuitive, sequential manner. It provides a session by session view of the data in a set of line charts. For example, Figure 3 shows a session that lasted about 15 minutes with 9 types of interaction events (e.g., mouseover paper). If user identity is collected, it can be used to further segment the data. When this information is not collected, such as in the example dataset, we assume the visualization application was used by one unknown user, represented as a “star (*)” in IntiVisor.

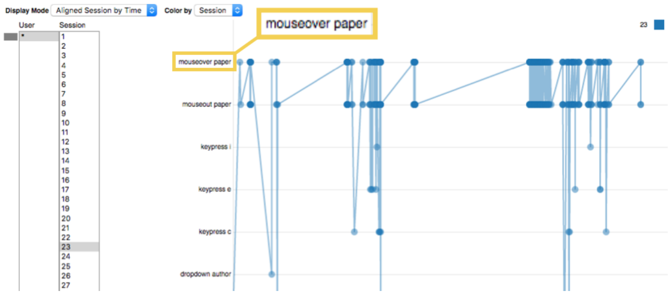

Fig. 3. Inspect view showing one interaction session that lasted about 15 minutes with 9 interaction types (e.g., mouseover paper).

IntiVisor provides the *Select view* to give an analyst a way to select, combine, and include parameters to events, in order to identify actions. This conversion process helps clean the event data for the analysis. The view includes lists of single, pairs (bigrams), and triples (trigrams) of events, ordered by frequency (Figure 4). An analyst can select any to be included in the analysis. Selected events and event groups (bigram/trigram) are added to the left-most list, as shown in yellow in Figure 4. After each selection, the selected event or event group is removed from the n-grams lists. As a result, the n-gram lists become shorter as the selection continues. The analyst can stop selections when all relevant events and event groups are chosen.

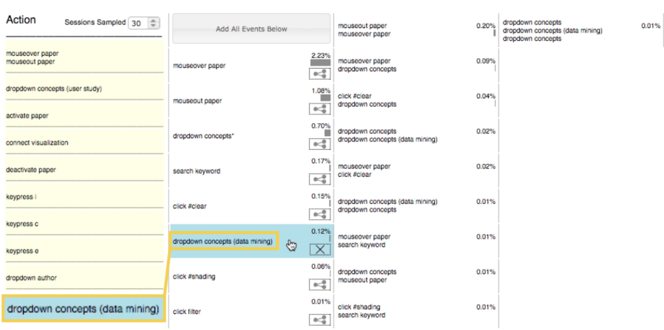

Fig. 4. Select view showing a set of events and event groups selected (yellow).

To add parameters to an event, an analyst selects the button next to the event, as shown in Figure 4. A dialog will be displayed that shows an example of an event’s parameter list (Figure 5). In Figure 5A, one example of the event “dropdown concepts” includes a parameter, “user study.” If the analyst selects the parameter, other values of this parameter will be listed in the view below (Figure 5B). The analyst can now select all the specific values of this parameter that he/she is interested in. The parameter values selected are used to extract a subset of events that have these values. This subset of events is extracted as new, selectable events in the Select view with the attached parameter values in parenthesis (e.g., “drop down concepts (data mining)”, Figure 4). The n-gram lists will be updated accordingly with these newly separated events that include parameter values.

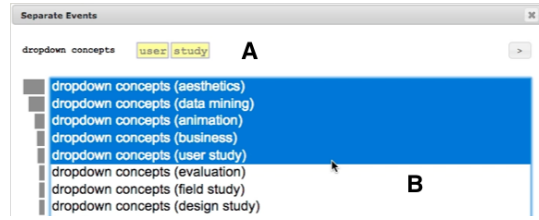

Fig. 5. Dialog for adding parameters to events. (A) Example parameter list, user study, from event “dropdown concepts.” (B) Frequency distribution of parameter values from the selected parameter indices in the example parameter list.

The n-gram extraction and update process can be slow when a large amount of interaction data is available. Since the n-gram lists are only for guiding an analyst’s event selection, we by default use a sample of the interaction data to calculate the lists. The assumption is that a sample will be sufficient to approximate the frequency distribution of events and event groups. If an analyst would prefer to include a larger amount of sample sessions, he/she can configure it manually in the upper-left corner of the Select view.

Actions → Operations

IntiVisor provides the *Categorize view* to give an analyst a way to organize actions onto a spatial layout as operations. This organization can help an analyst gain a better understanding of the relations between operations and recall the selected operations better in the next two views. An analyst can use a drawable canvas in this view to create a context for categorizing actions. For example, in Figure 6A, an analyst categorized actions by the features of the visualization application. As a result, the context was drawn to mimic the layout of the original visualization application. To categorize actions, the analyst drags and drops each action from the list to the left onto a relevant position on the canvas as a circle (operation). Alternatively, if an analyst wishes to categorize the actions into Yi et al.’s interaction categories [12], the context could be constructed differently (Figure 6B). When multiple actions need to be categorized into the same category, an analyst simply drags them on top of each other from the list. The actions within the same category will have their corresponding circles tightly connected with lines (Figure 6B).

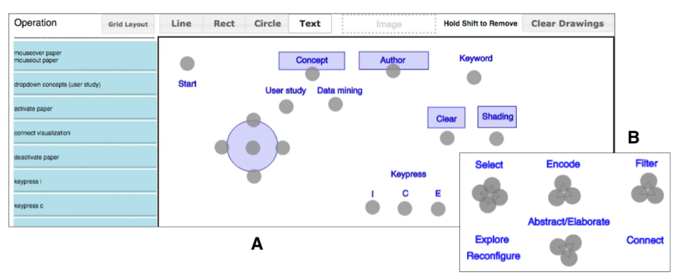

Fig. 6. Categorize view showing actions being categorized by feature using (A) a UI layout of the visualization application or (B) Yi et al’s [12] interaction categories as context.

Operations → Tactics/Orderings

IntiVisor extracts tactics and orderings in two views. The system provides the *Sequence view* to give an analyst a way to extract tactics, which are sequential patterns of *consecutive* operations (Figure 7). Sequences are shown as circles (operations) with tapered edges. Similar to the Select view, the Sequence view also by default uses a sample of the interaction data to extract sequential patterns to increase the extraction speed. Several parameters can be configured to extract patterns, such as the minimal (sequence occurrence) frequency. Sometimes, only sequential patterns including a certain operation is of interest to an analyst. For example, an analyst may be specifically interested in sequences that occur at the beginning of a session, such as strategies observed in Kang et al.'s study [8]. To find these sequences, an analyst selects the "connect visualization" operation, labeled "Start" on the canvas, that occurs at the beginning of each session (Figure 7). This selection by default will filter out all sequences that do not include this operation. Next, the analyst selects the "Start" button at the top of the view to indicate that the "connect visualization" operation needs to be at the beginning of a sequential pattern. These sequences are shown in Figure 7. Similarly, an analyst could specify if a sequence ends with a specific operation.

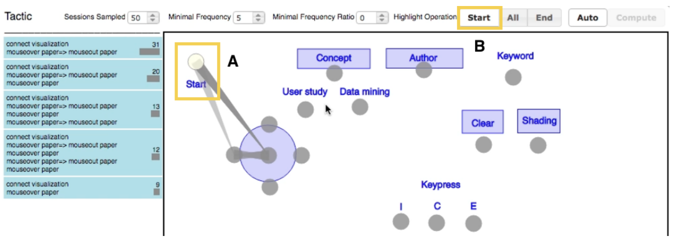

Fig. 7. Sequence view showing sequential patterns of consecutive operations (tactics) that start with "connect visualization" (labeled "Start" on the canvas).

Sequential patterns of both consecutive and *non-consecutive* operations are extracted as orderings in the last view of IntiVisor, the *E*xplore view, to help an analyst find higher-level analysis methods. The most frequent orderings of two and three operations from a subset of data (sample) are listed in two columns at the bottom of the view (Figure 1D). Orderings are visualized as a list of colored circles (operations) with dots in between that indicate the operations do not need to occur back to back. These orderings can be selected to explore their occurrences in other visualizations of the Explore view, which will be described next.

Explore Overall Patterns

Finally, IntiVisor provides the *Explore view* to give an analyst a way to flexibly explore usage patterns from the interaction log. (Figure 1). In addition to showing frequent orderings, this view visualizes the operations and tactics both in a graph (Figure 1B) and in a set of bars (Figure 1E). The graph shows all the operations and tactics the same way as in the Sequence view but with colors. The bars show all the operations as colored blocks in the context of their sessions. Each horizontal bar represents a session. By default, the sessions are ordered by number of operations (length). It can also be ordered by other contextual variables such as time and user. An analyst can choose to highlight any operation or tactic in both views. When a selection is made, it will be displayed at the top of the bar view (Figure 8A) and its usage distribution will be displayed in two bar charts (Figure 1C, Figure 8B): one showing distribution of sessions over time and one showing distribution of users over time.

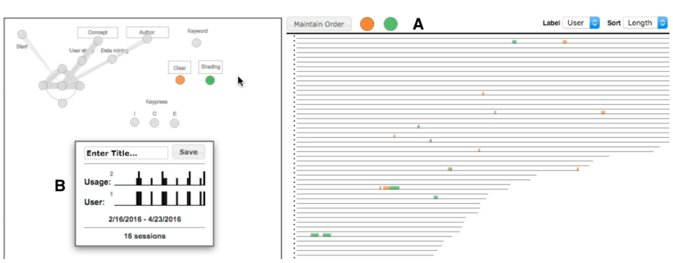

Fig. 8. Explore view with two operations selected and highlighted. (A) Selection area. (B) Usage distributions of selection.

In a selection, operations, orderings, and tactics are visually represented differently. When multiple operations are selected, they are represented as spatially separated circles with spaces in between (Figure 8A, Figure 9A). All operations will be highlighted independently of each other. If an analyst wishes to discover when these operations occur in the selection order, he/she can enable the "Maintain Order" mode, which adds three dots between every operation that indicate other operations could be between the selected operations (Figure 9B). This selection will highlight the operations only when they occur in this specific order. If an analyst wishes to discover when a set of operations is occurring back to back, such as in a tactic, the analyst can click on the three dots between the operations to "merge" them. The merged selection will be shown as a set of partially overlapping circles (Figure 9C) and be highlighted in the bar view.

Fig. 9. Three types of selections: (A) Select two operations independently of each other. (B) Select two operations in this order but not necessarily occur back to back. (C) Select two operations when they occur consecutively.

4 Conclusion

We present a visual interaction analysis system, IntiVisor, that supports the flexible event organization process of visualization interaction analysts. Although we focused our application on interaction analysis, IntiVisor could also be applicable to other applications that analyze temporal events.

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, May 2011.

[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. doi: 10.1109/TVCG.2015.2467871

[3] M. Brehmer and T. Munzner. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE TVCG*, 19(12):2376–2385, Dec. 2013. doi: 10.1109/TVCG.2013.124

[4] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. Lipford, and R. Chang. Recovering Reasoning Processes from User Interactions. *IEEE CGA*, 29(3):52–61, May 2009. doi: 10.1109/MCG.2009.49

[5] D. Gotz and Z. Wen. Behavior-driven Visualization Recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pp. 315–324. ACM, New York, NY, USA, 2009. doi: 10.1145/1502650.1502695

[6] H. Guo, S. Gomez, C. Ziemkiewicz, and D. Laidlaw. A Case Study Using Visualization Interaction Logs and Insight Metrics to Understand How Analysts Arrive at Insights. *IEEE TVCG*, 22(1):51–60, Jan. 2016. doi: 10.1109/TVCG.2015.2467613

[7] 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. doi: 10.1109/TVCG.2008.137

[8] Y.-a. Kang, C. Gorg, and J. Stasko. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *IEEE VAST '09*, pp. 139–146, Oct. 2009. doi: 10.1109/VAST.2009.5333878

[9] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal Event Sequence Simplification. *IEEE TVCG*, 19(12):2227–2236, Dec. 2013. doi: 10.1109/TVCG.2013.200

[10] 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. doi: 10.1007/s13218-012-0167-6

[11] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In , *IEEE Symposium on Visual Languages, 1996. Proceedings*, pp. 336–343, Sept. 1996. doi: 10.1109/VL.1996.545307

[12] 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. doi: 10.1109/TVCG.2007.70515