Session Viewer: Visual Exploratory Analysis of Web Session Logs

Heidi Lam*
University of British Columbia
Google, Inc.

Daniel Russell†
Google, Inc.

Diane Tang†
Google, Inc.

Tamara Munzner*
University of British Columbia

Abstract

Large-scale session log analysis typically includes statistical methods and detailed log examinations. While both methods have merits, statistical methods can miss previously unknown subpopulations in the data and detailed analyses may have selection biases. We therefore built Session Viewer, a visualization tool to facilitate and bridge between statistical and detailed analyses. Taking a multiple-coordinated view approach, Session Viewer shows multiple session populations at the Aggregate, Multiple, and Detail data levels to support different analysis styles. To bridge between the statistical and the detailed analysis levels, Session Viewer provides fluid traversal between data levels and side-by-side comparison at all data levels. We describe an analysis of a large-scale web usage study to demonstrate the use of Session Viewer, where we quantified the importance of grouping sessions based on task type.

Keywords: Web session log analysis, visual exploratory data analysis, information visualization

Index Terms: H.5.m [Information Interfaces and presentation (e.g., HCI)]: Miscellaneous—

1 Introduction

Providing better information-seeking support on the Internet requires understanding web search usage behaviors. Researchers have used methods ranging from field studies to web session log analyses to achieve this goal. While field studies can reveal rich and detailed information in situ and in context of users' goals, the approach is too labor intensive for large population analyses.

Session log analysis is a more scalable alternative. *Sessions* are time-stamped sequences of user actions. Session logs capture realistic search behaviors, as users perform real information searches in their own environments uninterrupted by the data collection mechanism. However, session logs are difficult to analyze due to the large data size and complex composition.

One analysis option is a detailed study of individual sessions. While this approach can lead to interesting insights, the limited sample size and potentially biased sampling may render general conclusions inaccurate, or even misleading. Moreover, this approach is very labor intensive. A more scalable and commonly used alternative is to compute overall population statistics at multiple levels, such as unique term frequency at the query level and event type frequency at the session level [16], or more complex web usage mining methods to model and predict user behaviors [8]. While these statistical approaches are scalable and effective, they tend to be hypothesis-driven and confirmatory rather than data-driven and exploratory, and may not uncover unexpected trends or may obscure subpopulation differences in the data. In addition, without exploring the data, hypothesis formation can be difficult.

A key challenge with session log analysis is to bridge between detailed and aggregate analysis. For example, the mean of a certain session attribute, such as the average number of events, is not representative if the data is bi-modally distributed. However, if the analysts can better define subpopulation boundaries by examining the data distribution and comparing and contrasting potential subpopulations, then the subsequent calculated aggregates would yield more useful information.

These analysis problems are not unique to web session log analysis. Tukey and others advocated exploratory data analysis using graphical plots to ensure adequate data exploration and understanding before applying statistical methods, and data analysis is considered as a continuum from exploratory to confirmatory analysis [32]. Visual exploratory analysis (VEA) is an attractive approach given our visual capabilities to spot trends, patterns, and anomalies. Ideally, for aggregate analysis, analysts should be able to visually estimate population boundaries based on session or event attribute distributions, or relative prevalence of certain event sequences. At the detailed analysis level, being able to visualize individual sessions in context of a larger population can mitigate sample selection bias, as the selection can be guided by session attribute distributions.

In practice, effective VEA requires a sophisticated visualization tool. In this paper, we share our experience in building Session Viewer, a VEA tool to support web session log analysis. We discuss our design goals and choices in building the tool, and illustrate its use with log data from a large-scale user study.

2 Background: Session Logs

The basic unit of session log analysis is a **session**, or a time-stamped sequence of events. An **event** corresponds to a user action, such as submitting a query to the search engine or clicking on a web result. Each event has attributes, such as a time-stamp or URL. A **session population** is a group of sessions with shared characteristics, such as usage patterns or task types.

It is important to respect users' privacy when collecting sessions or any other type of data, and ensure that users agree to having their data collected and understand what is being collected and how it will be used. One way to capture session logs is for the user to voluntarily install a piece of software on their machine that will send information back to a server. Fig 1 shows a sample log from one such study [26], analyzed using Session Viewer in the use-case scenario (Section 6).

In general, a session is a multi-dimensional data object. Most dimensions are single values (e.g., event counts per session), but one dimension is a time-ordered sequence of event objects. Each event is itself a multidimensional datum. In short, a session log has structure at three levels: session population, session, and event.

3 Related Work

We review existing literature about visualization systems and techniques that support analysis of web session logs, computer-based logs, and data exploration in general. We also review existing tools used for non-visual analysis of of web session logs.

3.1 Visualizations for web session logs

While interactive systems designed for session log analysis exist, they generally focus on website design evaluations based on traffic and user paths, rather than on search usage behaviors. Exam-

*e-mail: {hllam,tmm}@cs.ubc.ca
†e-mail:{drussell,diane}@google.com

Pane	Paneli	Que	Navigation_Event	Event_	URL	Title
1	15238	44	Begin_Section	1		
2	15238	45	Begin_Objective	1		
3	15238	45	Go to Start URL of Objecti	31	http://www.msn.com	
4	15238	45	Browser Navigate	59	http://search.msn.com/results.asp	MSN Search: london and tours
5	15238	45	Browser Navigate	118	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
6	15238	45	Browser Add Favorite	175	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
7	15238	45	Browser Navigate	204	http://www.london-tours.info/londo	Discovering London (Full Day Tour No 4) Ticke
8	15238	45	Browser Back	244	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
9	15238	45	Browser Navigate	265	http://www.london-tours.info/londo	Oxford, Stratford & Warwick Castle (Tour 23) Ti
10	15238	45	Browser Back	316	http://www.london-tours.info/	London Tours Info - over 70 London, UK & Pari
11	15238	45	Browser Back	320	http://search.msn.com/results.asp	MSN Search: london and tours
12	15238	45	End_Objective	360		

Figure 1: A sample web session log from the user study. Each row is an event, with sequence number, participant ID, question number, navigation type, event time, URL, and webpage title.

ples of website traffic visualizations include disk-tree and time-tube visualizations [6] and a 3D structure [34]. User paths are often displayed as node-link graphs, as in VISVIP [7], WebViz [22], and WebQuilt [28]. Lee *et al.* take a different approach, displaying web traffic statistics with starfields and user paths with parallel coordinates [15]. Hochheiser and Shneiderman use a multiple-coordinated visualization to show web visitation data [13]. Other visualizations, such as 3D WebPath [9] and History tree [20] display personal web-navigation histories, and are designed to help users navigate rather than to analyze their usage behaviors. Generally in these analyses, analysts tend to look for different paths through a fixed set of web pages, while in web search, analysts are essentially going through an infinite number of pages across many more domains. Consequently, existing systems do not adequately address the needs of web search analysis.

The one exception is Card *et al.*'s Web Behavior Graphs, which show search structures of individual users as modified state diagrams to help researchers locate problem spaces within the web site under analysis, and identifies usage behavior patterns [27]. Despite the richness of the information and insights obtained from the analysis, Card *et al.*'s approach is difficult to scale.

3.2 Visualizations for computer-based logs

A highly related area is usability log visualizations. Gray *et al.* used the Color Bar Visualization to show usability sessions, with each bar consisting of a stack of color-coded boxes encoding user activities types [10]. However, the system does not allow comparison between populations or displays at multiple levels of detail. Many visualizations associated with usability log analysis use 2D and 3D graphs. Examples include population counts and summary statistics of events [17] and mouse activities [11], sequence patterns displayed as state transition diagrams to support Markov-based analysis [12], and a spreadsheet-like display for sequence alignment [29]. These graphs are better suited for presenting analysis results than for data exploration, especially when generated manually.

There are also systems designed for computer log analysis, such as MieLog, which displays textual log entries as color-coded bars based on their type [30], AuthorLines, which shows emails participation and initiation counts based on author [33], and SnortView, which shows network-based intrusion detection system logs as a 2D time diagram [19]. However, since the analysis goals are typically detecting anomalies rather than identifying and characterizing populations, these systems are not well-suited for session log analysis.

3.3 Visualizations for general VEA

Visualization techniques and systems have been developed to support general VEA [18], including commercial ventures such as Spotfire (www.spotfire.com), Tableau (www.tableausoftware.com), and Inxight (www.inxight.com). While these systems support VEA, their visualizations are typically standard graphical displays that are not tailored for showing multi-level data.

Various visualization techniques have been developed to facilitate aspects of visual data exploration and analysis. For example, dynamic query filtering, where the user progressively refines filter criteria aided by visual feedback of the results [2], facilitates data filtering to isolate interesting subpopulations in the data for more detailed analysis. The reorderable matrix, an idea first introduced by Bertin [4] and implemented in Table Lens [23], allows reordering of samples to reveal outliers, correlated features, and trends in sample populations. Pattern-matching allows the analysts to highlight sequences, and one implementation is TimeSearcher [14]. Providing different views of the same data allows analysts to observe different aspect and projections to further enhance discovery [24]. Small multiples, first proposed by Bertin as *collections* [4], and further advocated by Tufte [31], provides a means to compare sample populations. Different forms of small multiples have been arranged in rows and columns to create univariate and bivariate matrices (e.g., [1]). All of these visualization techniques inspired the design of Session Viewer.

3.4 Non-visual web log analysis tools

Commercial statistics packages are frequently used for web log analysis, for example, Microsoft Excel (www.microsoft.com) and SPSS (www.spss.com). Analysts also build custom programs, from simple scripts to calculate summary statistics, to elaborate algorithms to find usage patterns and population clusters (e.g., [8]).

4 Web Session Log Analysis

We designed Session Viewer for experienced analysts. To ground our design, we interviewed five analysts to understand their analysis goals, tasks, and workflows. Each analyst had at least five years of analytical experience with at least one in session log analysis. The interviews were semi-structured with an initial list of questions, but were largely driven by the participants' descriptions of their analysis processes illustrated with their tools. The interviews were an hour long and were recorded.

Based on the interviews, we identified two analysis levels: detailed-session and aggregate-metrics. Of the five analysts we interviewed, two were detailed-session analysts, two were aggregate-metrics analysts, and one used both methods. Both types of analyses illustrate the different needs and goals of session log analyses.

Detailed-session analysis aims to answer specific but open-ended questions about usage behavior, such as the use of Boolean OR in queries, and to develop standard metrics to measure task nature and user satisfaction. Typically, our analysts look at less than 500 sessions per analysis. The moderate number is partly due to time and effort constraints, but more importantly, most analysts can form satisfactory hypotheses based on 100 to 200 interesting sessions. Statistical methods are sometimes used for further analyses.

Statistical-aggregate analysis also aims to understand usage behavior, but at the aggregate level based on established metrics. Typically, the analysis compares different session populations. Section 6 shows a use-case scenario where we compared populations based on labeled task types.

5 Session Viewer

In this section, we describe Session Viewer in detail. To establish common terminology, we first discuss the main data objects and visualization panes in Session Viewer. We then explain our design goals and highlight tool features that realize these goals, followed by individual component descriptions.

5.1 Data objects

Session Viewer manages two base data types: session and event, each with its own list of attributes. To accommodate diverse analysis goals, Session Viewer allows users to define event *states* based on any number of attributes in a dialog box. For example, a long Image result click is defined as `[(Event.duration >=`

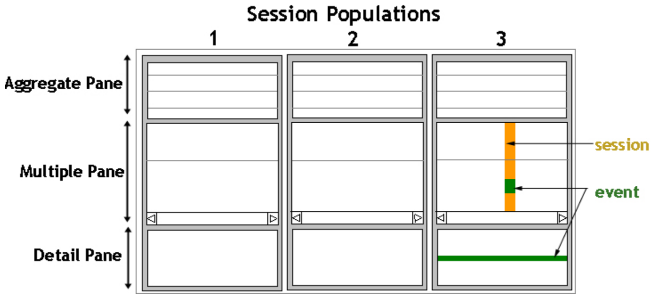


Figure 2: A schematic diagram showing the multiple coordinated views in Session Viewer. Each vertical view shows a population with three panes. Each pane corresponds to a data level and contains one or more panels.

```
300 seconds) AND (Event.property == IMAGE)
AND (Event.action == RESULT_CLICK)].
```

5.2 Main visualization panels

Fig 2 shows a schematic diagram of the interface, and Fig 3 shows a screen capture. Session Viewer uses multiple coordinated views with linked interactions [21] to display several session populations side-by-side for visual comparison. Session Viewer displays each population in a vertical *view* and shows the data at three levels:

1. The *Aggregate Pane* corresponds to the session-population data level and contains panels that display population statistics such as counts, distributions, and annotations.

2. The *Multiple Pane* corresponds to the sessions data level and contains two linked panels of session attributes and sessions as collections of events.

3. The *Detail Pane* corresponds to the events data level and shows the logs for a selected session in a table, with one row per event.

5.3 From design goals to tool features

Based on our analysis experience (e.g., [26]) and the analyst interviews, we set the main design goal for Session Viewer to support session log analysis at both the statistical-aggregate and the detailed-session analysis levels and to bridge between the two. Here we highlight the features designed to achieve this goal.

1. *Support analysis at the statistical-aggregate level*
Even though statistical analysis leads to highly scalable, succinct, and comparable numeric descriptions of populations, proper statistical analysis requires matching data with methodological assumptions, which in turn requires understanding the data distributions. Also, rapid hypothesis testing is often difficult in practice, as most statistics packages require non-trivial data regrouping and formatting for different analyses. Session Viewer:

- *Provides statistical summaries*. As shown in the Aggregate Pane in Fig 3, Session Viewer provides descriptive statistics as graphical plots commonly used in log analysis, such as a state transition diagram to show event state sequences and a state counts plot to show the relative counts of each event state over the course of the sessions. Session Viewer also shows session attribute distributions, either as vertical lines within the double-slider bars for continuous attributes or as text on the toggle buttons of categorical attributes, as shown in Fig 6. Providing such distributions helps analysts identify subpopulations for exploration within Session Viewer or make informed choices of statistical methods for further analysis.

- *Detects event sequences*. Session Viewer provides a sequence-matching feature that is similar to regular-expression matching in strings. In our case, the "alphabets" are user-defined event states. The user can highlight sessions with specific event state sequences in the Multiple Pane to visually and rapidly estimate relative pattern prevalence for hypothesis testing. An example is shown in Fig 12 and detailed in the use-case scenario in Section 6.

- *Enables session population comparisons*. The visual equivalent of comparative statistics is visual comparison between populations. Session Viewer provides side-by-side comparison of populations at all data levels, enabled by shared scales in graphical plots across the vertical views. For example, the state count bar charts in all views share the same x- and y-scales, as shown in Fig 3.

2. *Support analysis at the detailed-session level*
Detailed session analysis can yield insights unavailable from aggregate metrics. However, analysts need to examine individual sessions, track events both within a single session as well as between sessions, and coordinate between the event webpages, the session logs, and their own annotations. Moreover, session selection is difficult, since the nature of sessions is difficult to discern from logs. One common way to select sessions is to use the "Find" function offered in applications to view the sessions. Session Viewer:

- *Displays events in the context of sessions*. Individual events are only meaningful when interpreted within the context of adjacent events within a session, and within a larger session population, as in the Sessions Panel.

- *Integrates analysis resources*. The Aggregate Pane supports annotations for each population and the Detail Pane provides direct links to event webpages, as shown in Fig 3.

- *Guides session selection for detailed analysis*. Users can also choose sessions using the event sequence detection, as described above. Also, Session Viewer displays session attribute distributions that are reorderable. The Multiple Pane in Fig 3 shows sessions reordered by total event counts, and the analyst selected sessions with high event counts for detailed study. The action is based on Bertin's reorderable matrix [4], extended to multiple views to show multi-level data objects.

3. *Bridge between the statistical-aggregate and the detailed-session analysis levels*
While most analysts realize the limitations of specific analysis methods, extending their practices to include multiple data levels is difficult as most tools do not adequately support cross-level analysis. For example, to better understand a particular aggregate metric in a statistical-aggregate analysis, analysts would need to associate and examine individual sessions with the selected metric values. Similarly, to guide session selection in a detailed analysis, the analysts would need to calculate and plot distributions of relevant session attributes. Such data-processing steps are non-trivial and distracting to the main analysis.

Session Viewer encourages multi-level analysis by displaying session logs at three levels of detail. Using standard linked navigation and highlighting techniques, the users can quickly move up and down the data hierarchy. For example, clicking on the State Transitions Panel in the Aggregate Pane would highlight sessions with the specified event state transitions in the Multiple Pane. The users can then select an individual session to display event details.

4. *Connect to, rather than replicate, existing analysis tools*
Instead of covering all aspects of analysis, we envision Session

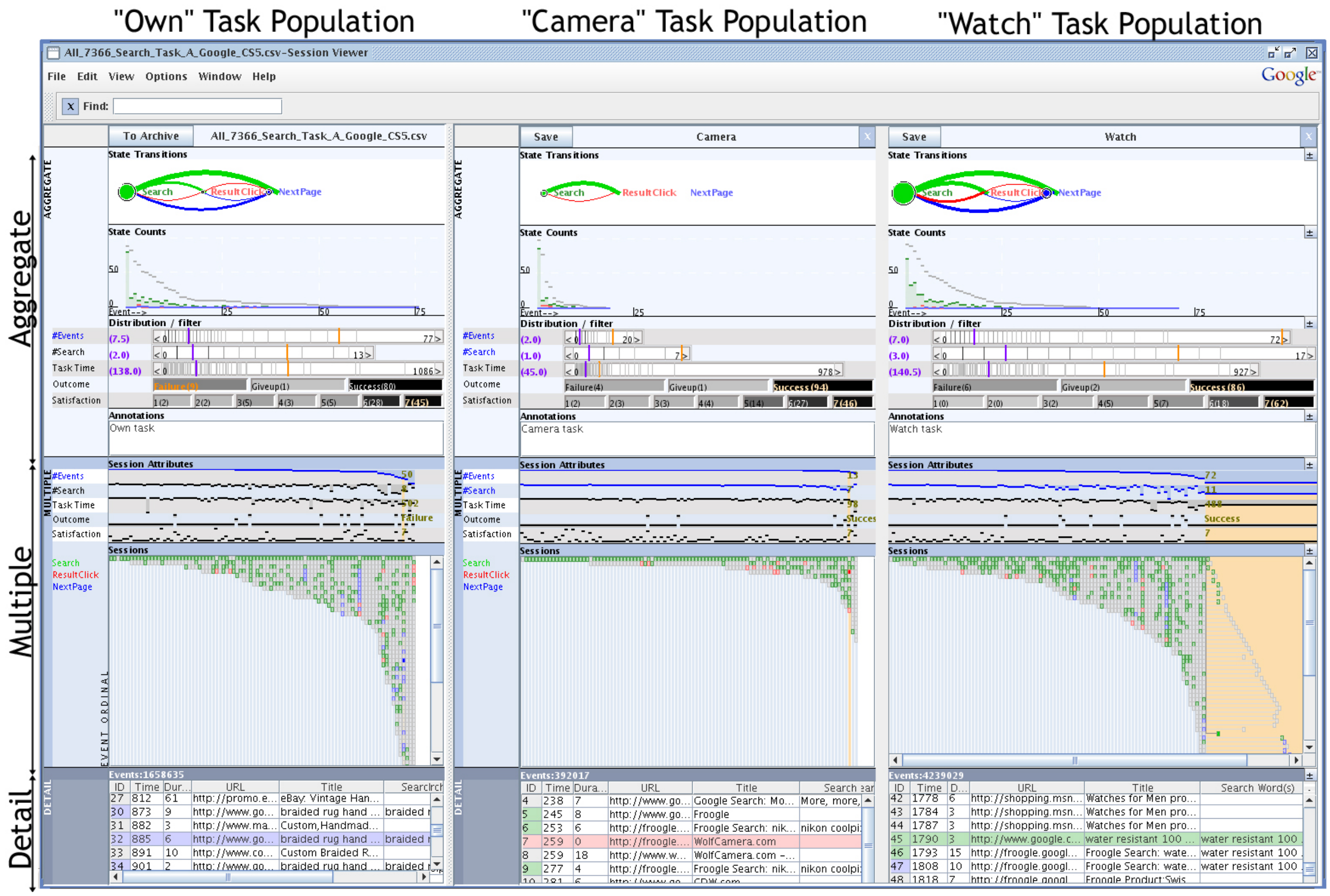


Figure 3: The main screen of Session Viewer showing user study data. Session Viewer displays multiple session populations in vertical views. From left to right: the "Own" Task population where participants performed their own self-created tasks; the Camera Task population with searches for a camera feature given the brand and model; and the Watch Task population with searches to locate a watch based on given criteria. For each population, the session logs are shown in the Aggregate, the Multiple, and the Detail Panes.

Figure 4: State Transitions Panel.

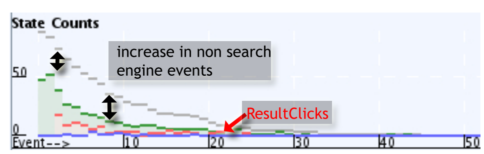


Figure 5: The State Counts Panel is a bar chart with event ordinality on the x-axis and event counts on the y-axis for all events (gray bars) and individual event states (stacked and color-coded). Here, the later events were predominantly non-search engine events, as indicated by the increasingly large gap between the total event counts and coded event state counts.

Viewer as part of a toolkit and focus on supporting hypothesis generation. To connect with log sources and commercial statistics packages, Session Viewer imports data from various log sources and exports them in standardized formats.

5.4 Panel details and coordinations

We now describe the individual panels organized as three panes: Aggregate, Multiple, and Detail, as shown in Fig 3.

5.4.1 Aggregate Pane

This top pane has four panels, showing population metrics and distributions: the State Transitions Panel, the State Counts Panel, the Distribution/Filter Panel, and the Annotations Panel.

The **State Transitions Panel** displays event state transitions flowing clockwise (Fig 4). Arcs sharing the same originating and destination states are bundled to avoid overlapping. The panel can be used to detect unexpected event sequences and states, such as Search events after a long delay that may indicate user goal change.

The **State Counts Panel** is a stacked bar chart with the x-axis being session event ordinal and the y-axis as event state counts (Fig 5). The panel can be used to monitor relative event state prevalence over the course of the sessions. For example, Fig 5 shows that while some of the initial decline in the green Search events is due to an increase in red ResultClick events, later events are mostly uncoded non-search-engine events, suggesting exploration.

The **Distribution/Filter Panel** displays ranges or categories of session attributes for data filtering (Fig 6). Users can filter continuous data using the double-slider bars. Filtering is guided by the stripe graphs, which shows distributions with median values as purple vertical stripes and text on the right. Toggle buttons provide

Figure 6: Distribution/Filter Panel. The continuous attribute Task Time is represented by a double-slider bar for filtering with the distribution displayed as a stripe graph on the slider bar. The categorical attribute Outcome is represented by a series of toggle buttons for filtering, with the category name and count as labels, and the counts encoded with luminance in the button background.

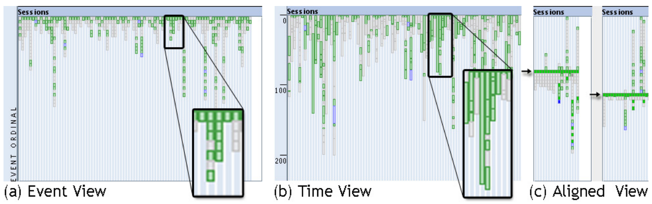


Figure 7: Sessions Panel. (a) Event view is the default, where each event rectangle has the same height to better show transitions. (b) In the Time view, the height of each rectangle encodes the event duration. (c) In the Aligned view, sessions are aligned vertically at some chosen common event (indicated with arrow annotations).

categorical data filtering, guided by the button labels that show the categories' session counts.

5.4.2 Multiple Pane

The middle pane has two panels that function as a unit, showing individual sessions and attributes. Each session occupies a unique vertical lane spanning both panels, as seen in the orange highlight in the schematic diagram in Fig 2and in the screen capture in Fig 3.

The **Session Attributes Panel** shows user-selected attributes for each session displayed in a Table Lens-like chart (Fig 9). These attributes can be continuous, such as the search counts, or categorical, such as task outcome. Since the panel is used for displaying trends instead of directly reading off individual attribute values, heights of as few as 10 pixels are acceptable for the bar charts.

The **Sessions Panel** shows each session as a stack of colored rectangles (Fig 7). Each rectangle corresponds to an event, color-coded by event state. Time flows from top to bottom. The rectangle height is either uniform to better display event sequences (Fig 7a), or encodes event duration to highlight long events (Fig 7b). Sessions are aligned at the start of the sessions by default or aligned at a user-chosen common event. Fig 7c show two examples of aligning by the first occurrence of a search event. The right-hand side sessions in Fig 7c had fewer events after the common query than those in the left, suggesting a more effective query string.

Users can click to expand individual sessions into 2-dimensional displays to show usage behavior, with the vertical dimension still encoding event ordinal or time while the horizontal dimension encodes unique event URLs. For example, Fig 8 shows methodical exploration of results listed on the page returned by the search engine as columns of events of the same result page (SS) punctuated by webpages launched from the result page.

Users can drag and drop attribute names in the Session Attributes Panel to reorder the sessions. The vertical display order of attribute names determines the horizontal sort order of the sessions of both the Session Attributes and the Sessions Panels. Fig 9 shows two populations: top (a), and bottom (b1, b2). In Fig 9a and b1, sessions with low satisfaction scores are highlighted in orange and reordered by the task outcome, with Failure and Given-up sessions in the far left. In Fig 9a, the orange highlighting is concentrated on the left side of the display, indicating a strong correlation between task

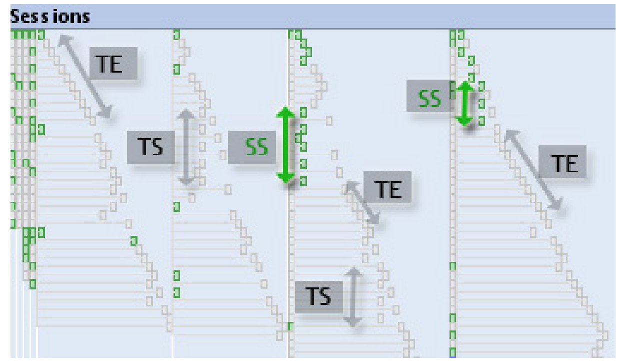


Figure 8: Expanded sessions in the Sessions Panel showing three search patterns: SS, Search-engine Searches; TS, Third-party Searches using third-party online sites as search engines; and TE, True Explorations of search results.

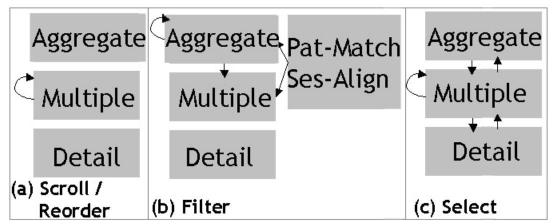


Figure 10: Interaction coordination scheme between the three data levels: (a) Session reordering and scrolling is limited to the Multiple Pane; (b) Filtering is initiated at the Aggregate level, or by the pattern-matching or session-alignment feature, affecting the Aggregate and the Multiple Panes; (c) Selection and highlighting can be initiated at all data levels and are transitive.

outcome and satisfaction score in the top population. However, in Fig 9b1, we do not see that same correlation in the bottom population. Instead, we notice that the ResultClick event counts may be correlated with task outcome: there are more red ResultClick events in the Sessions Panel on the right side (Success Task Outcome) of the display. To explore that correlation, we reorder the sessions by the #ResultClicks session attribute: sessions with low ResultClick counts are on the left in Fig 9b2. Our hypothesis is confirmed since unsuccessful sessions (highlighted in orange) are clustered on the left with low ResultClick counts.

5.4.3 Detail Pane

The low-level pane has a single panel, the **Events Panel**, that shows an individual session as a table (Fig 3). Each row shows an event, with columns displaying attributes such as timestamp and URL.

5.4.4 Interactions and view coordinations

Session Viewer uses standard linking and navigation techniques for view coordinations [21], as shown in Fig 10. For example, highlighting a session in the Sessions Panel will display its events in the Events Panel, highlight the associated session in the Session Attributes Panel, and highlight the distribution plots in the Distribution/Filter Panel, as shown in Fig 3.

5.5 Implementation details

Session Viewer was written in Java using the JRE 1.5.0_06 library and the Java2D graphics library.

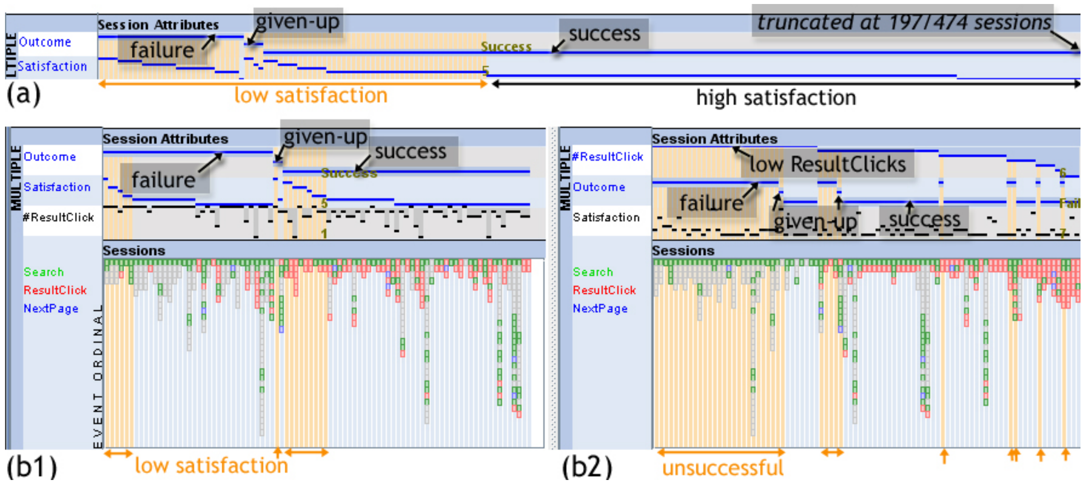


Figure 9: Examples to show how the Session Attributes Panel reveals trends and correlations. Top population: (a) Low satisfaction scores are highlighted in orange. Sessions are reordered by Outcome and by Satisfaction. Since the orange-highlighted sessions with low satisfaction scores cluster with the Failure and Given-up outcomes, the panel shows a high correlation between task outcome and satisfaction score in this population. Bottom population: (b1) Satisfaction score and task outcome are not correlated, as seen by the lack of clustering of low satisfaction scores in the Failure and Given-up sessions. (b2) Instead, we see the ResultClick event count correlates with task outcome, as shown by the cluster of highlighted unsuccessful sessions with low ResultClick counts on the left.

6 Use-Case Scenario: Exploring the Relationships between Task Type and Search Behavior

We now describe an analysis of logs from a user study to understand search usage behavior [26]. Even though Session Viewer can be used on any session logs, we showcase study data for the rich labels, such as task instructions and user satisfaction.

The study recruited close to 400 participants and generated about 6,000 sessions grouped by three experimental factors: search engine type, search domain (e.g., Image, News), and question variant. Question variant includes three defined search tasks, plus one where participants were asked to create their own tasks. In a previous analysis, we grouped the sessions along the three experimental factors and identified two main populations: sessions where participants were given explicit instructions, and those when they performed their own tasks [26]. We revisit the data and find that task types play an important role in characterizing session populations.

We first defined a set of event states based on actions: Search, ResultClick, and NextPage, and did not code events unrelated to the search engine. We then loaded two session populations with the same search engine and domain, but different question variants. The center view in Fig 3 shows the Camera question variant:

> *Assume you are looking for a digital camera and a friend suggested the Nikon Coolpix 4600. Use <site> to search for information about the Nikon Coolpix 4600. How many megapixels is the image resolution of a Nikon Coolpix 4600 digital camera?*

and the right-hand view is the Watch question variant:

> *Assume you are looking for a man's watch as a gift for a friend or family member. Use <site> to search for a man's watch that is water resistant to 100 meters and under \$100. What brand of watch did you choose?*

We expected the populations to look similar as the tasks were supposedly isomorphic, but they were different: the Watch sessions were longer and more involved than the Camera sessions. This difference is readily apparent in the Aggregate Pane (Fig 3): the Watch population had more event transitions in the State Transitions Panel and more high ordinality events in the State Counts Panel plot.

To better understand the event count distribution, we reordered the sessions by dragging the #Events header in the Session Attributes Panel to the top, and the Sessions Panel showed that most of the sessions in the Camera population had less than three events (Fig 3). A quick visual scan of the Session Panels showed that short sessions had proportionally more Search events (green rectangles). The Distribution/Filter Panel in Fig 3 provided more quantitative information: the Watch population had a wider range of both total event and search event counts, as seen in the wider #Events and #Searches slider-bars. Moreover, despite having a similar absolute range of total task time, the Watch population had a wider distribution, as seen in the more dispersed stripe graph on the Task time slider-bar when compared to the Camera population, where the stripes clustered around the shorter end of the task time range.

These visual differences led us to re-examine the task instructions to understand the task goals, as that was the main difference between the two populations. Even though both tasks were aimed to find a commercial product, they differed in nature: the Camera task directly looked for a property of a specific object, while the Watch task required exploration as only the properties of the object, rather than a specific identifier, were given. Using the framework for search goals proposed by Rose and Levinson [25], we classified the Camera task as a directed closed informational search, whereas the Watch task is an informational locate task.

Side-by-side visual comparison of the event state sequences in the Sessions Panels led to another hypothesis: different search patterns would be prevalent in session populations of different task types. We visually tested this hypothesis using the event sequence pattern-matching feature. Using S to denote a Search event and X to denote a non-search engine event, we defined four usage patterns identified in earlier detailed session analyses, which we further refined based on event state sequences in the Sessions Panels:

1. *Short Navigation*: S[Start]→X[End], with the S event limits to the first session events and the X event to the last events.
2. *Topic Exploration*: S→X→X→X→X
3. *Methodical Results Exploration*: S→X→S→X→S
4. *Query Refinement*: S→S→S→S

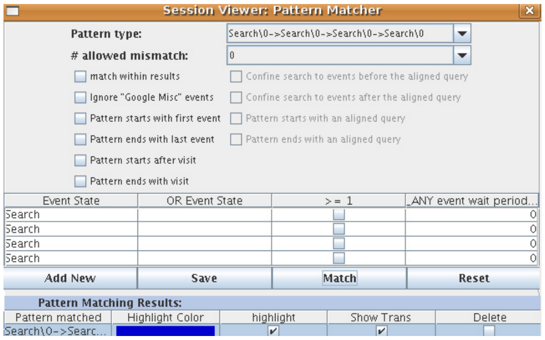
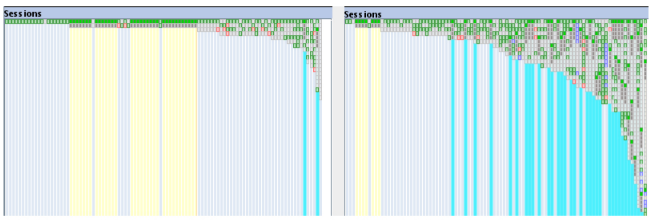


Figure 11: Dialog for users to specify event sequences in a pattern.



(a) Camera Task Population (b) Watch Task Population

Figure 12: Sessions Panels for two task types. Short Navigation sessions are highlighted in yellow, and those with the Topic Exploration pattern are highlighted in aqua.

Using the pattern-matching dialog box (Fig. 11), we defined these patterns and highlighted the Short Navigation sessions in yellow and the Topic Exploration sessions in aqua. Fig 12 clearly shows that Short Navigation searches were more prevalent in the directed, closed Camera population, while the Topic Exploration pattern was more common in the exploratory Watch population. Encouraged by the visual differences, we highlighted the other search-behavior patterns and observed similar results.

To test our visual finding on the entire study data, we manually labeled the other question variants and repeated the analysis with an external statistics package. As shown in Fig 13, our hypothesis was confirmed. In general, only 14% of the exploratory locate-type tasks were Short Navigations compared to 37% in directed, closed-type tasks. List and undirected information searches were more similar in composition to locate than to directed, closed tasks. As in the previous analysis, we also concluded that participants were more exploratory in their Own task, as they were visually more similar to the exploratory Watch tasks than the closed Camera task at all data levels, as shown in Fig 3.

While the Methodical Results Exploration and Query Refinement patterns were understandably present in exploratory sessions, we wondered what the participants were doing in those non-search-engine X events in the Topic Exploration sessions. To answer that question, we selected longer and more involved Topic Exploration sessions for detailed examination in the Events Panel.

To locate such sessions, we sorted the sessions again by the #Events attribute, and then focused on the high end of the distribution in the Sessions Panel. In Fig 3, we expanded the session with the largest event count in the Watch population and found a strange pattern: the first half contained mostly non-search-engine events colored in gray while the second half contained mostly NextPage events colored in blue. In the first half, long sequences of events were punctuated by green Search events in the same horizontal lane, meaning the Search events had the same URL. To better understand this behavior, we examined the individual events in the Events Panel and found two main search strategies. In the first half of the session, the participant used the search engine (Search events in green) to reach third-party websites such as amazon.com, walgreen.com, and shopping.msn.com and searched within those shopping sites (uncoded events in gray). In the second half, the participant used domain-specific searches (Froogle) that involved mostly NextPage events in blue. We were intrigued by the first half of the session, where the participants used the search engine as a launching pad for exploration within third-party websites.

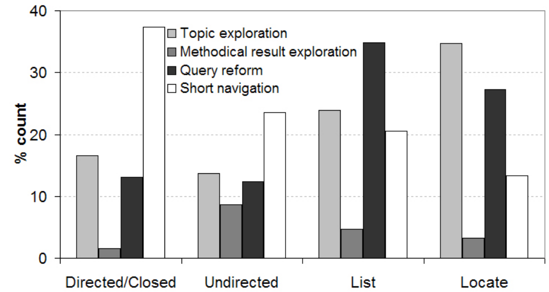


Figure 13: Confirming a hypothesis formed by exploration in Session Viewer: sessions of directed/closed information tasks contain significantly more Short Navigational patterns than sessions of undirected, list, or locate task types.

To determine if the behavior was unique to this participant, we expanded several sessions and saw similar behaviors: search-engine searches (columns of green boxes, annotated with SS) punctuated by third-party sites searches (columns of gray boxes, annotated with TS) and true result explorations within these sites (diagonal gray boxes, annotated with TE), as shown in Fig 8. This finding suggests a need to differentiate between this new Nested Search pattern and true Topic Exploration.

In summary, Session Viewer's side-by-side population comparison at multiple levels allowed us to quickly spot differences between the Watch and the Camera populations, which we had originally assumed to be the same. The pattern-matching feature allowed us to quickly test a hypothesis that the relative prevalence of different event sequences would be an important feature for characterizing different session populations. Session reordering guided our selection of interesting sessions for detail event-by-event examinations, where we discovered the Nested Search usage pattern.

7 Discussion: Design Choices and Lessons Learned

7.1 Spatial consistency over user control

Session Viewer has a relatively rigid spatial layout. While users can selectively place session populations in the vertical views, the same data panels are displayed for all populations in a fixed order.

In an earlier design, we used the sketch-book metaphor and envisioned a free-style workspace where users could drag and drop interesting panels and directly annotate on the workspace. We eventually abandoned that design, as we believe that users can better mentally process the data if the same data panels are displayed for all populations and are arranged to reflect the data hierarchy. In data exploration where the analysis freely moves between different levels of detail, having a consistent display is cognitively less demanding and allows the analyst to focus on the task at hand, rather than mentally organizing the data displays. This design philosophy is in accordance with Baldonado *et al.*'s *rule of consistency* in their guidelines for multiple-view use in visualization [3].

7.2 Data filtering and partitioning over Focus+Context

Providing effective overviews in the Sessions Panel is a challenge due to the large number of sessions and the need to provide enough details for selection and comparison. This challenge has often been addressed with focus+context techniques [5, p.307]. The Sessions Panel was once implemented as a one-dimensional fisheye view with distortion in the vertical time direction. The non-linear scale caused confusion and made time estimation difficult. We now use filtering and reordering to partition the data into a series of overviews. Given the data heterogeneity, comparing and contrasting subgroup overviews proved more effective than viewing the entire data collection at once.

8 Conclusion and Future Work

Session Viewer is a visualization tool designed to support exploratory analysis of web session logs at the statistical-aggregate to the detailed-session analysis level. We take a multiple-coordinated view approach and display multiple session populations at the Aggregate, Multiple, and Detail data levels to support the corresponding analysis methods. To bridge between the levels, Session Viewer supports fluid traversal between data levels within each population and side-by-side comparison at all data levels between populations. Displaying distributions and offering pattern matching helps statistical-aggregate analysts detect and characterize session populations. Displaying a large session population reorderable by attributes helps detailed-session analysts select sessions.

We plan to provide more flexible definitions of session and event attributes, for example, to replace form-like dialog boxes (e.g., for pattern sequences in Fig 11) with a script-like interface given our technical target users. We are planning a study with experienced analysts to study tool use.

References

[1] A. M. MacEachren et al. Exploring high-D spaces with multiform matrices and small multiples. In *Proc. IEEE Symposium on Information Visualization (InfoVis '03)*, pages 31–38, 2003.

[2] C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling for dynamic query filters with starfield displays. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'94)*, pages 313–317, 1994.

[3] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. ACM Advanced Visual Interface (AVI 2000)*, pages 110–119, 2000.

[4] J. Bertin. *Graphics and Graphic Information-Processsing*. Walter de Gruyter, 1981.

[5] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, California, 1999.

[6] E. H. Chi. Improving web usability through visualization. In *Internet Computing*, pages 259–263, 2002.

[7] J. Cugini and J. Scholtz. VISVIP: 3D visualization of paths through web sites. In *Proc. 10th Intl. Workshop on Database & Expert Systems Applications: Web-Based Information Visualization (WebVis)*, pages 259–263, 1999.

[8] D. Pierrakos et al. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction*, 13:311–372, 2003.

[9] E. Frecon and G. Smith. WebPATH: A three dimensional web history. In *Proc. IEEE Symposium on Information Visualization (InfoVis '98)*, pages 3–10, 1998.

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

[11] M. Guzdial, P. Santos, A. Badre, S. Hudson, and M. Gray. Analyzing and visualizing log files: A computational science of usability. Technical Report GIT-GVU-94-08, Georgia Institute of Technology, 1994.

[12] M. Guzdial, C. Walton, M. Konemann, and E. Soloway. Characterizing process change using log file data. Technical Report GIT-GVU-93-44, Georgia Institute of Technology, 1993.

[13] H. Hochheiser and B. Shneiderman. Using interactive visualizations of WWW log data to characterize access patterns and inform site design. *Journal of American Society of Information Sciences*, 52(4):331–343, 2001.

[14] H. Hochheiser and B. Shneiderman. A dynamic query interface for finding patterns in time series data. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'03)*, pages 522–523, 2003.

[15] J. Lee et al. Visualization and analysis of clickstream data of online stores for understanding web merchandising. *Data Mining and Knowledge Discovery*, 5(1/2):59–84, 2001.

[16] B. J. Jansen. Search log analysis: What is it; what's been done; how to do it. *Library and Information Science Research*, 28(3):407–432, 2006.

[17] J. Kay and R. C. Thomas. Studying long-term system use. *Communications of the ACM*, 38(7):61–69, 1995.

[18] D. A. Keim. Information visualization and visual data mining. *IEEE Trans. Visualization and Computer Graphics*, 7(1):100–107, 2002.

[19] H. Koike and K. Ohno. SnortView: Visualization system of Snort logs. In *Proc. ACM Workshop on Visualization and data mining for computer security (VizSec '04)*, pages 143–147, 2004.

[20] J. Kreuseler, T. Nocke, and H. Schumann. A history mechanism for visual data mining. In *Proc. IEEE Symposium on Information Visualization (InfoVis'04)*, pages 49–56, 2004.

[21] C. North and B. Shneiderman. A taxonomy of multiple window coordinations. Technical Report CS-TR-3854, Univ. Maryland Computer Science Dept, 1997.

[22] J. Pitkow and K. Bharat. WebViz: A tool for worldwide web access log visualization. In *Proc. First Intl. World Wide Web Conf. (WWW1)*, pages 271–277. Elsevier, 1994.

[23] R. Rao and S. Card. Table Lens: Merging graphical and symbolic representations in an interactive focus plus context visualization for tabular information. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'94)*, pages 318–322, 1994.

[24] J. C. Roberts. Multiple-view and multiform visualization. In *Visual Data Exploration and Analysis VII. Proc SPIE 3960*, pages 176–185, 2000.

[25] D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proc. 13th Intl. World Wide Web Conf. (WWW13)*, pages 13–19. Elsevier, 2004.

[26] D. M. Russell and C. Grimes. Assigned tasks are not the same as self-chosen web search tasks. In *Proc. IEEE 40th Annual Hawaii Intl. Conf. on System Sciences (HICSS'07)*, page 83, 2007.

[27] S. K. Card et al. Information scent as a driver of web behavior graphs: Results of a protocol analysis method for web usability. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems (CHI'91)*, pages 498–505, 1991.

[28] S. Waterson et al. What did they do? Understanding clickstreams with the WebQuilt visualization systems. In *Proc. ACM Advanced Visual Interface (AVI 02)*, 2002.

[29] P. Sanderson and C. Fisher. Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*, 9:251–317, 1994.

[30] T. Takada and H. Koike. MieLog: A highly interaction visual log browser using information visualization and statistical analysis. In *Proc. USENIX Conf. on System Administration*, pages 133–144, 2002.

[31] E. R. Tufte. *The Visual Display of Quantitative Information.* Cheshire, CT: Graphics Press, 1983.

[32] J. W. Tukey. Data analysis, computation and mathematics. In L. V. Jones, editor, *The collected works of John W. Tukey. Volume IV: Philosophy and principles of data analysis: 1965-1986*, pages 753–775, 1986. (Original work published 1972).

[33] F. Viegas and M. Smith. Newsgroup crowds and authorlines: Visualizing the activity of individuals in conversational cybersapces. In *Proc. 37th Annual Hawaii Intl. Conf. on System Sciences (HICSS'04)*, page 40109.2, 2004.

[34] B. Wong and G. Marden. Effectively exploiting server log information for large scale web sites. In *Proc. South African Institute for Computer Scientists and Information Technologists (SAICSIT)*, pages 223–227, 2001.