MAQUI: Interweaving Queries and Pattern Mining for Recursive Event Sequence Exploration

Po-Ming Law, Zhicheng Liu, Sana Malik, and Rahul C. Basole

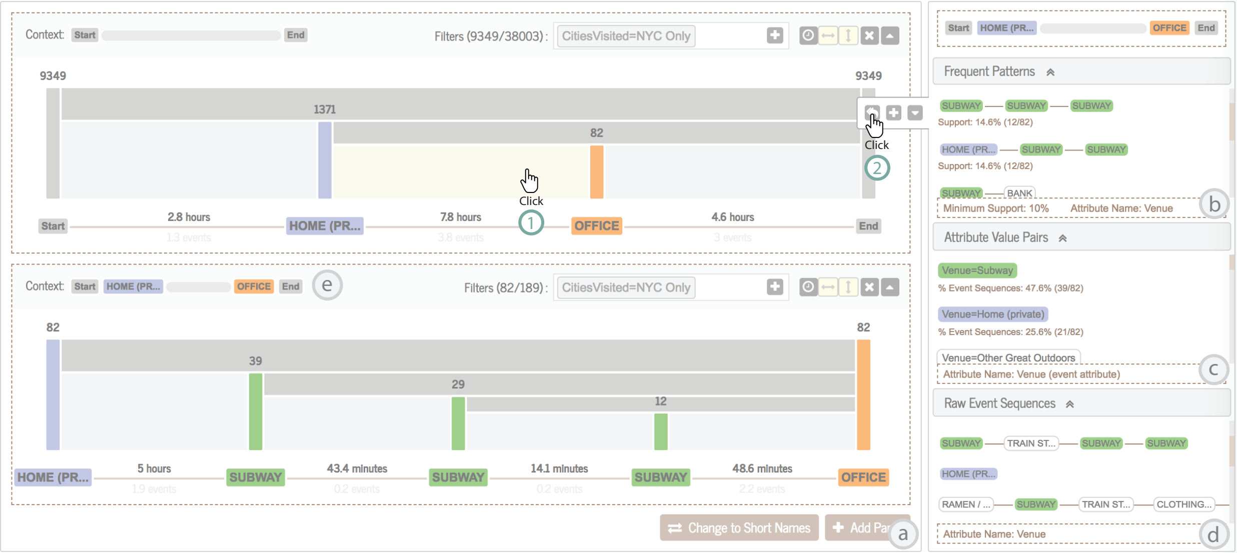


Fig. 1. There are four major views in MAQUI: (a) the workspace, (b) the frequent pattern view (c) the attribute-value pair view and (d) the raw sequence view. (1) As the analyst clicks on the rectangular region between *HOME* and *OFFICE*, it becomes the current focus and is highlighted in yellow. The frequent pattern view, the attribute-value pair view and the raw sequence view are subsequently updated. (2) The analyst is selecting the *SUBWAY*→*SUBWAY*→*SUBWAY* pattern and using it to split up the segments between *HOME* and *OFFICE*. The bottom panel is in turn created.

Abstract—Exploring event sequences by defining queries alone or by using mining algorithms alone is often not sufficient to support analysis. Analysts often interweave querying and mining in a recursive manner during event sequence analysis: sequences extracted as query results are used for mining patterns, patterns generated are incorporated into a new query for segmenting the sequences, and the resulting segments are mined or queried again. To support flexible analysis, we propose a framework that describes the process of interwoven querying and mining. Based on this framework, we developed MAQUI, a **M**ining **A**nd **Q**uerying **U**ser **I**nterface that enables recursive event sequence exploration. To understand the efficacy of MAQUI, we conducted two case studies with domain experts. The findings suggest that the capability of interweaving querying and mining helps the participants articulate their questions and gain novel insights from their data.

Index Terms—Sequential pattern mining, temporal query, event sequence exploration

1 Introduction

The collection and analysis of event sequence data occurs in many domains. For instance, e-commerce companies seek to understand customer behaviors from clickstream data and inform marketing decisions [36]. In the healthcare domain, electronic health records are sources of information that can provide insights into whether recommended guidelines are followed [43].

The sheer volume and complexity of event sequences present many challenges in the visual analysis of such data. Visualization techniques alone are often not scalable to provide an overview of the data [28]. To summarize large scale event sequence data, sequential *pattern mining* algorithms extract linear patterns using metrics such as frequency and profit [12]. However, using only mining algorithms has its limitations. Without any human input, the mined patterns are not always useful or interesting [25]. In addition, contextual information, such as where the patterns happen in the original sequences, is lost [18]. An interface that only supports user-defined *queries*, on the other hand, enables dynamic formulation of questions analysts have in mind, but risks missing unexpected patterns in the data.

To overcome these limitations, recent works propose to combine *querying* with *mining* for event sequence exploration. (s|qu)eries [45] is primarily a querying tool based on regular expressions, but it also provides a ranked list of events for inspection. DecisionFlow [16, 18] best demonstrates the idea of combining mining and querying. It allows analysts to query for sequences that match specified constraints, perform pattern mining on the retrieved sequences and segments, and then explore the results with interactive visualizations.

While these approaches have been successful in supporting flexible exploratory analysis of event sequence data, the *recursive* nature of event sequence exploration has not received adequate attention. Our conversations with event sequence analysts reveal that the analytic

- *Po-Ming Law, and Rahul C. Basole are with Georgia Institute of Technology. E-mail: {pmlaw, basole}@gatech.edu*
- *Zhicheng Liu, and Sana Malik are with Adobe Research. E-mail: {leoli, sanmalik}@adobe.com*

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

workflows do not always follow a querying→mining→visualization pipeline. Instead, Querying and mining are often interwoven in a recursive manner: sequences retrieved from a query are used as the input for a mining algorithm, the mined patterns may then be used to segment the sequences, and some of the resultant segments in turn serve as the input for follow-up querying or mining. During such processes, it is difficult for analysts to flexibly articulate queries, specify which part of the dataset should be mined, and keep track of the context in which the current exploration happens.

To address these complexities, we propose a framework that describes the process of interwoven querying and mining in recursive event sequence exploration. The framework introduces the concepts of analytic *focus* and *context*. Its novelty lies in articulating how a set of atomic user actions can be combined to modify the analytic focus and context in continuous loops. Grounded on the framework, we designed MAQUI, a **M**ining **A**nd **Q**uerying **U**ser **I**nterface for recursive event sequence exploration. MAQUI employs a panel-based interaction design that is tightly coupled with the concepts in the framework. To demonstrate the efficacy of MAQUI, we conducted two case studies with marketing analysts and a health informatics professional. The capability of interweaving querying and mining was well-received by the participants and was able to help them find novel insights from their data. In particular, our work makes the following contributions:

1. A framework that depicts how a set of atomic user actions can be combined to support interwoven querying and mining in a novel recursive manner. This framework was grounded in an investigation of the analytic questions of clickstream analysts.

2. MAQUI, a visual analytics system that employs novel interaction designs to support interwoven querying and mining in a recursive manner.

2 Related Work

Combining mining with querying for event sequence exploration is not a new idea. However, there has been no systematic investigation into how they should be combined to scaffold analysts' exploration. Existing work that investigates querying and mining techniques for event sequence mainly falls into two categories: mining-centric interfaces and query-centric interfaces. Mining-centric interfaces aid in discovering interesting patterns in event sequence data by utilizing advanced pattern mining algorithms; they offer limited or no query capabilities. On the other hand, query-centric interfaces empower analysts to create complex queries to extract event sequences of interest; they often provide limited support for mining patterns from data. Our work lies at the intersection of both lines of research.

2.1 Mining-Centric Interfaces

Mining-centric interfaces can range from fully automatic pattern mining to semi-automatic pattern mining.

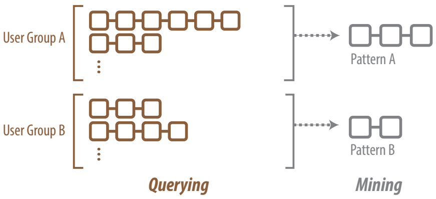
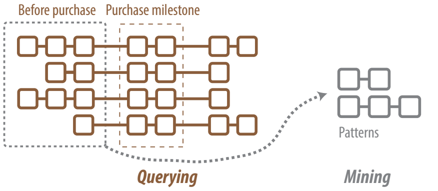
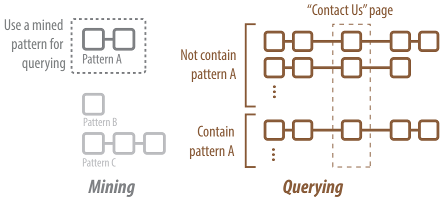
Fully automatic pattern mining is a completely linear process during which analysts apply a mining algorithm and browse a long list of patterns generated by the algorithm [38]. Fournier-Viger et al. [12] offer a comprehensive survey on sequential pattern mining. Two sequential pattern mining algorithms have been widely adopted by the visualization community to extract patterns from event sequences: the SPAM algorithm (SPAM) [3] and the VMSP algorithm (VSMP) [13]. SPAM [3] uses a smart bitmap representation to efficiently generate frequent patterns. Albeit efficient, SPAM may produce a large number of patterns, creating difficulty in browsing through them. VMSP [13] was subsequently developed to generate more compact patterns to reduce the number of patterns presented to analysts. Much research has also been devoted to developing visualization techniques for frequent patterns, shielding users from the tedious process of browsing long lists of them. FP-Viz [21] is an early work that visualizes frequent patterns using the Sunburst visualization. Frequence [32] visualizes the patterns produced by a modified SPAM algorithm using a Sankey-based visualization. Peekquence [23] focuses on visualizing the co-occurrence relationship between event types and patterns. Coreflow [25] extracts branching patterns and visualizes them as icicle plots. Liu et al. [26] identified four levels of granularity for visualizing and analyzing clickstream data. Recently, Chen et al. [7] proposed a mining algorithm based on the minimal description length principle to construct an overview of event sequences while reducing information loss.

In semi-automatic pattern mining, a system provides mechanisms for interacting with a pattern mining algorithm. For instance, Xin et al. [44] developed a technique to allow analysts to rank a small set of patterns generated by a mining algorithm. To help analysts rapidly explore interesting patterns, the patterns are re-ranked based on user interests inferred from the interactions. Vrotsou et al. [38] developed an interactive technique that lets analysts mine interesting patterns in a stepwise manner. To tackle the long running time required to generate sequential patterns, Progressive Insights [37] allows analysts to interact with the partial results generated by SPAM to abort the process or prioritize subspaces of interest. The most relevant mining-centric interfaces only offer limited query capabilities for event sequence exploration. Parthasarathy et al. [31] proposed computational methods for efficiently mining and querying an event sequence database when the database is updated on a regular basis but they did not consider a user interface. Chronodes [35] and TimeStitch [34] allow analysts to use patterns generated by SPAM as focal points. For example, analysts can explore the sequences that occur between a pattern A and a pattern B. However, complex queries such as setting time gap constraints between two patterns and segmenting the dataset by record attributes are not supported. DecisionFlow [16] allows analysts to specify an ordered list of event types as a query. A variant of DecisionFlow [18] empowers analysts to mine frequent patterns from a segment between event types. Its primitive query capability, however, offers limited expressiveness for retrieving event sequences, restricting the variety of questions analysts can ask. As illustrated by research in query-centric interfaces, analysts have diverse questions that require expressive querying techniques to answer. Our work attempts to complement mining by endowing analysts with the capability of creating expressive queries.

2.2 Query-Centric Interfaces

Contrary to mining-centric interfaces, query-centric interfaces offer advanced capabilities for defining event sequences of interests. The event sequences extracted are often visualized to help analysts gain insights into the data. The earliest work in this area includes Pattern Finder [11, 33], LifeLines2 [39] and LifeFlow [40]. With Pattern Finder [11,33], users can specify queries using events, event sets, event attributes, and time spans. Both LifeLines2 [39] and LifeFlow [40] allow users to align event sequences by choosing an event type to be the alignment point. The aligned sequences are then visualized using a simple horizontal timeline or aggregated into a visualization similar to icicle plots. Building on LifeLine2 and LifeFlow, Monroe et al. [28,29] developed EventFlow that provides advanced capability for searching event sequences. They demonstrated through real-life use cases (e.g., [4,6,27]) that complex queries are of importance for answering real-life questions by real users. Driven by this line of research, advanced query capabilities are introduced into other applications [8,9,17,19,20,22,42, 46]. For instance, Tempo [17], COQUITO [22] and CAVA [46] enable analysts to express complex queries for iterative cohort construction. PeerFinder [9], Similan [42] and Similan2 [41] allow users to search for event sequences that are similar to a target record. Finally, there is a recent trend in developing even more expressive techniques for querying event sequences based on regular expression [5, 45]. An example is EventPad [5], which compresses multivariate event sequences by using user-defined regular expression rules. The most relevant query-centric interface to our work is (s|qu)eries [45]. With (s|qu)eries, users can search for sub-event sequences using complicated regular expression-based queries. Similar to our work, it enables users to mine frequent events in selected sub-event sequences and incorporate these frequent events into subsequent queries. However, (s|qu)eries does not support pattern mining algorithms, limiting users to inspecting frequent events but not frequent patterns. With query-centric interfaces, analysts risk missing unexpected patterns. We strive to complement querying by utilizing sequential pattern mining algorithms that can generate patterns analysts would have missed if they have to find patterns manually.

Table 1. Sample tasks collected from the clickstream data analysts. The action column illustrates how the analysts can accomplish the tasks by combining querying and mining.

Action	Task
Querying→Mining	**Task:** Compare behaviors between groups of users. (T1) **Example question:** Is the common path of user group A different from that of user group B? (Q1)  User Group A User Group B Pattern A Pattern B Querying Mining
Querying→Mining	**Task:** Study user paths before/after a milestone. (T2) **Example question:** For those who purchased product X, what is the common path that leads to the purchase? (Q2)  Before purchase Purchase milestone Patterns Querying Mining
Mining→Querying	**Task:** Identify the characteristics of event sequences that do not contain a known pattern. (T3) **Example Question:** How else are they getting to the "Contact Us" page? At what rate? (Q3)  "Contact Us" page Use a mined pattern for querying Pattern A Pattern B Pattern C Not contain pattern A Contain pattern A Mining Querying

3 Identifying Usage Patterns from Analysts' Questions

The motivation for interwoven querying and mining comes from our long-term collaborations with analysts working in a large software company. The company collects massive amount of clickstream data of visitor behaviors on its websites. The analysts would like to draw insights from this data to understand potential website usability issues and relate customers' behaviors to their purchase decisions. Table 1 lists some of the recurring tasks the analysts want to perform and the example questions related to these tasks.

These questions suggest some of the common analytic needs and workflows found in a real-world problem domain. To understand whether the findings can be generalized to domains other than clickstream data, we surveyed the literature to verify these findings. Two major observations were identified:

Querying and mining are often interwoven recursively. The analysts often want to create queries to retrieve sequences of interest and apply a mining algorithm to extract the common paths (querying→mining). For instance, to address Q1, the analysts can divide the dataset by user groups into different sets and mine frequent patterns from each set. The analysis often does not stop here. After mining some patterns, the analysts may use them to create a subsequent query (mining→querying). Prior work also shows that event sequence analysts often use the answer of a previous question to build a new question [45] and it verifies our observation. As another example of mining→querying, answering Q3 requires the analysts to first mine the common paths before the visitors reach the "Contact Us" page and use a common path to retrieve the sequences that eventually reach the "Contact Us" page but do not get there via that common path. The common path mined becomes part of the new query. The analysts may then recursively perform the querying→mining or mining→querying operations on the event sequences retrieved.

Advanced query capabilities are essential. The querying→mining questions in Table 1 hints on the analysts' diverse needs to query event sequences. In Q1, the analysts segment the dataset by user groups. To answer Q2, the analysts need to define the sequence of events involved in purchasing a product. This sequence (the purchase milestone) may contain a single event or multiple events. After defining the purchase milestone, the analysts extract the event sequences that contain the milestone and inspect only the segments that occur before the milestone. Articulating these queries can be challenging without advanced query capabilities. Research in visual temporal queries (e.g., [22, 28]) and the recent attempt to adopt regular expression to enhance the expressiveness of event sequence query language (e.g., [5, 45]) corroborate this observation. Hence, the capabilities to create expressive queries is not only desirable but also essential for answering analysts' diverse questions.

Many tools are available to the analysts, ranging from simple next event/previous event visualization dashboards to script-based mining tools. The lack of integration between these tools, however, impedes recursive exploration. The analysts may issue a query through a graphical user interface or writing in SQL, but the results will then need to be exported and saved as files for further mining operations. To use the mined patterns as input for further analysis, additional custom scripts must be written to transform the saved data files. The process is cumbersome and it is easy to lose context of the analysis. It is our goal to design an integrated exploration environment that interweaves querying with mining and provides expressive querying capabilities. To do so, we need a conceptual tool to help us think about the recursive process involved in such explorations.

4 A Framework of Interwoven Querying and Mining

Based on the analysis of user tasks in Section 3, we propose a framework that describes the dynamics of the atomic user actions in recursive exploration of event sequences. While the atomic user actions are grounded in prior research in event sequence exploration, we contribute to prior art by proposing the new concepts of analytic focus and context (Sec. 4.2), and how querying and mining are interleaved as analysts refine the context and focus using the atomic user actions (Sec. 4.3).

To facilitate discussion, we use a dataset on Foursquare check-ins as a running example. Foursquare is a location sharing service that allows users to "check-in" at different places to share their locations with friends. Our dataset contains check-ins in New York City (NYC) and Tokyo (TKY) in May 2012 [1].

4.1 Data Model

We define an **event** as a set of attribute-value pairs: $E_i = \{\mathsf{A}_1 = v_1, \mathsf{A}_2 = v_2, \ \dots \ , \mathsf{A}_j = v_j\}$. In the Foursquare dataset, there are 141,220 events (i.e. check-ins). Each event consists of four **event attributes**: Venue (with a set of possible values such as *Coffee shop*, *Home*, and *Bus stop*), City (*NYC* or *TKY*), UserID and Timestamp (exact time when the check-in occurred). An example event would be {Venue=*Bar*, City=*NYC*, UserID=*JohnSmith*, Timestamp= *05/12/2012 20:30:05*}.

An **event sequence** is an ordered list of events: $S_i = [E_1, E_2, ..., E_k]$. Given all the check-ins in the dataset, we can group the events by UserID and order them by Timestamp to form event sequences. Each event sequence has a number of **record attributes** that describe the properties of the sequence. Record attributes can be domain-independent such as PathLength (number of events) and TimeSpan, or domain dependent such as CitiesVisited (with possible values being *NYC*, *TKY* and *Both NYC and TKY*)

A **segment** $S_i' = [E_m, E_{m+1}, ..., E_n]$ of an event sequence $S_i = [E_1, E_2, ..., E_k]$ is an event sequence contained in S_i, where $m \geqslant 1$ and $n \leqslant k$. A **pattern** is an ordered list of event attribute-value pairs found in a set of event sequences, where the attribute-value pairs do not have to be contiguous in the original sequences. For example, the pattern

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

Bus stop→*Work*→*Bus stop*→*Home* for the attribute Venue may be found in many sequences, but in certain sequences, there may be other venues in between Venue=*Work* and Venue=*Bus stop*. Our model does not consider patterns with a mixture of different attributes.

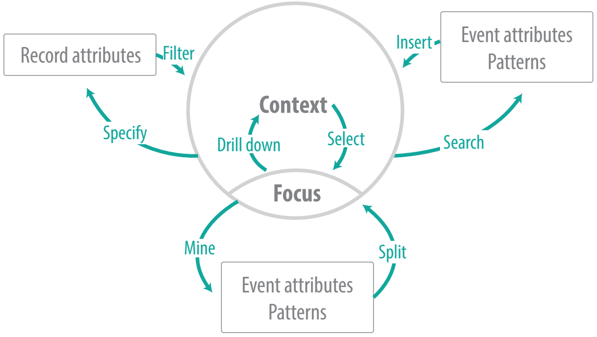


Fig. 2. A Framework of Interwoven Querying and Mining.

4.2 Analytic Focus and Context

Figure 2 provides a visual summary of our framework. Data components are colored in gray, and user actions are highlighted in blue. The blue arrows indicate the transformation between data components through user actions.

A **focus** is a set of segments that is the current target of analysis. A focus situates in a **context**, which is a larger set of event sequences that the focus is part of. Recursive exploration keeps redefining the focus set and the context so that analysts can ask questions and gain insights on the part of the dataset they are interested in.

For example, we may want to analyze the Foursquare check-in segments before an event with the attribute-value pair Venue=*Train Station*. This set of segments constitutes the current focus, highlighted in yellow in Figure 3. This focus is shown in a larger context, which also includes a set of segments after the Venue=*Train Station* event, and a set of sequences that do not contain a Venue=*Train Station* event.

Simply put, the focus is a subset of the context. At any time, analysts can **select** a set of segments in the context to make it the focus of analysis. Moreover, the notions of a focus and a context are relative: a focus may be **drilled down** to become the context for a subsequent focus, and thus contexts can be multi-level or nested.

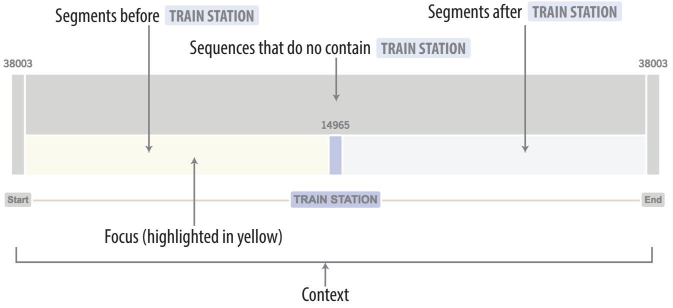

Segments before TRAIN STATION
Segments after TRAIN STATION
Sequences that do no contain TRAIN STATION
38003
38003
14965
Start
TRAIN STATION
End
Focus (highlighted in yellow)
Context

Fig. 3. A focus (in yellow) is a set of segments currently being analyzed. It resides in a larger context which may include additional sets of segments.

4.3 Atomic Actions for Refining Focus and Context

In our framework, users can take the following actions to refine and transform the analytic focus and context. For each action, we specify whether the action is *focus-only* (i.e. applicable to the set of sequences in focus) or *context-wide* (i.e. applicable to one or more sets of sequences).

Search for event attributes or patterns (*context-wide*). Analysts often need to look for a particular event or pattern they have in mind during the analysis. For example, in Google Analytics [2], analysts search for events to create a conversion funnel which describes the steps for a visitor to become a customer on an e-commerce website. In EventPad [5], a primary task is to search for events to specify patterns in regular expressions.

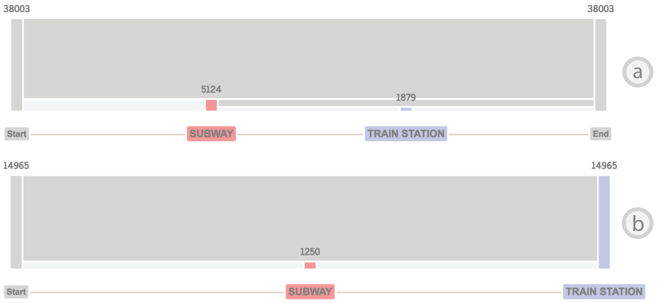


Fig. 4. (a) *Inserting* Venue=*Subway* before Venue=*Train Station* in Figure 3. (b) *Splitting* the focus in Figure 3 by Venue=*Subway*.

Insert/Remove event attributes or patterns (*context-wide*). Analysts often insert an event into an existing pattern, which will update the entire context. For example, we may insert a Venue=*Subway* event before the Venue=*Train Station* event in Figure 3. This action causes the entire context to be recomputed, resulting in Figure 4a. The insert action is often used in exploratory specification of a pattern. An example use case is conversion funnel analysis in e-commerce using tools such as Google Analytics [2] and Adobe Analytics [1]. The event can be removed to restore to the previous state in Figure 3.

Mine event attributes or patterns (*focus-only*). When analysts do not have a firm idea of what to expect in the focus, they can use a mining algorithm to extract frequent event attributes and frequent patterns. At any point in time during their analysis, analysts can only perform mining on one set of segments, which is the focus. The mine action thus does not apply to contexts.

Split/Merge event sequences (*focus-only*). Analysts can modify the focus by splitting or merging the sequences using the events or patterns obtained from the search or mining actions. For example, we may want to split the focus in Figure 3 by a Venue=*Subway* event. The outcome of the split is shown in Figure 4b. The original focus is divided into a set of segments before and after the Venue=*Subway* event, and a set of segments that do not contain the Venue=*Subway* event. The split action is often used for drill-down analysis of a set of segments. The resulting segments can be merged by removing the Venue=*Subway* event.

Specify record attributes (*context-wide*). To make the analysis more manageable, analysts often want to examine a subset of the sequence data by specifying the properties of the sequences for analysis. For example, they may not care about short sequences and want to focus on sequences with a relatively large value for the record attribute PathLength. To be able to precisely specify what kind of sequences they want to analyze, they need to see the distributions of different record attributes.

Filter sequences (*context-wide*). Once the analysts have specified the properties of sequences to explore, they can use that specification to filter the sequences. The filtering action can impact both the context and the focus as the focus is part of the context.

4.4 Recursive Loops

The atomic actions in the framework (*search-insert/remove*, *mine-split/merge*, *specify-filter*) recursively transform the focus and the context. The output of the transformation becomes the input of the next action, thereby forming a recursive loop. Recursive event sequence exploration is supported by flexibly combining these atomic actions. Users could start by *searching* and *splitting*, then perform *mining* on the selected focus — such usage pattern is seen in systems like DecisionFlow [16]. They could also start by *mining*, asking the system to produce frequent events, then use a frequent event to *split* the focus.

5 MAQUI: A Mining and Querying User Interface for Recursive Event Sequence Exploration

Even with a clear idea about how mining and querying can be interwoven to support recursive event sequence exploration, there are numerous ways to design a system to realize the framework. Based on the literature and our conversations with the analysts, we identified the following design considerations:

C1. Reducing visual clutter. Visualizations of event sequences often suffer from severe visual clutter [26, 28]. As noted by Chen et al. [7], there is often a trade-off between visual clutter and information content in a visualization. In the visual representation, low-level details of the event sequences should be abstracted away to reduce visual clutter while enough information about various aspects of the event sequences should be provided to analysts.

C2. Offering expressive yet intuitive query capabilities. Expressiveness is a desirable characteristic of a query language. Yet, intuitiveness is often compromised while striving for expressiveness. For instance, regular expression is a highly expressive language for querying event sequences [5, 45] but regular users may not have a good sense of how the underlying logic works. The system's query capabilities should be designed in way so that it is expressive enough yet intuitive.

C3. Providing context for the current focus. During recursive exploration, analysts mine event attributes or patterns from the current focus. These event attributes or patterns may subsequently be used for splitting the current focus. Hence, event sequences are repeatedly split and merged to produce different segments during recursive analysis. Conceivably, analysts can easily get lost when many segments are produced. The interface should help analysts understand the context in which the current analysis focus resides.

5.1 System Overview

MAQUI has four major views (Fig. 1): (a) the workspace, (b) the frequent pattern view, (c) the attribute-value pair view, and (d) the raw sequence view. The workspace consists of a collection of panels. Each panel (Fig. 5) corresponds to a context, and contains a top bar, a flow visualization and a timeline. At the beginning of the analysis, the workspace has only one panel, showing the entire dataset as the context (Fig. 5). The start and end nodes in the flow visualization represent the first and last events in all event sequences respectively. The number above the start and end nodes indicates the total number of event sequences in the dataset. The timeline displays the average duration and average number of events between the starting and ending points of all the event sequences.

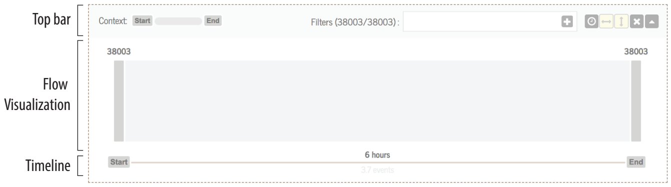


Fig. 5. Each panel in the workspace consists of the top bar, the flow visualization and the timeline.

During exploratory analysis, analysts can add multiple events to the flow visualization. For example, a Venue=*Home* event and a Venue=*Office* event are added to the top panel in Figure 1a. Adding too many events to the flow visualization will reduce its legibility. To avoid visual clutter, MAQUI employs a **multi-panel design** (C1). Analysts can create multiple panels in the workspace by clicking on + Add Panel.

The multi-panel design further facilitates cohort comparison. For instance, in Figure 12a-b, the analyst is comparing the routines between Americans and Japanese using two panels (a detailed usage scenario is described in Sec. 6). She creates the CitiesVisited=NYC Only filter to extract the check-ins in NYC (Fig. 12a) and the CitiesVisited=TKY Only filter to extract the check-ins in Tokyo (Fig. 12b).

To support comparison of cohorts, MAQUI allows users to adjust the positions and heights of the colored nodes so that they reflect duration and number of sequences: by clicking on ⟷, the distance between two nodes encodes the duration between two events (Fig. 6b); by clicking on ↕, the height of a node encodes the number of event sequences that contain the event (Fig. 6c). Analysts can also click on ⊙ to encode the distance between two nodes as the average number of events between the two events (Fig. 6d). By default, the positions and heights of the colored nodes in the flow visualization are distorted: distance between two nodes is equal and height of a node equals a fixed fraction of the height of the adjacent node on the left (Fig. 6a). The distortion aims to help analysts select a rectangular region between two nodes by reducing overlapping between them when the duration between the two events is too short.

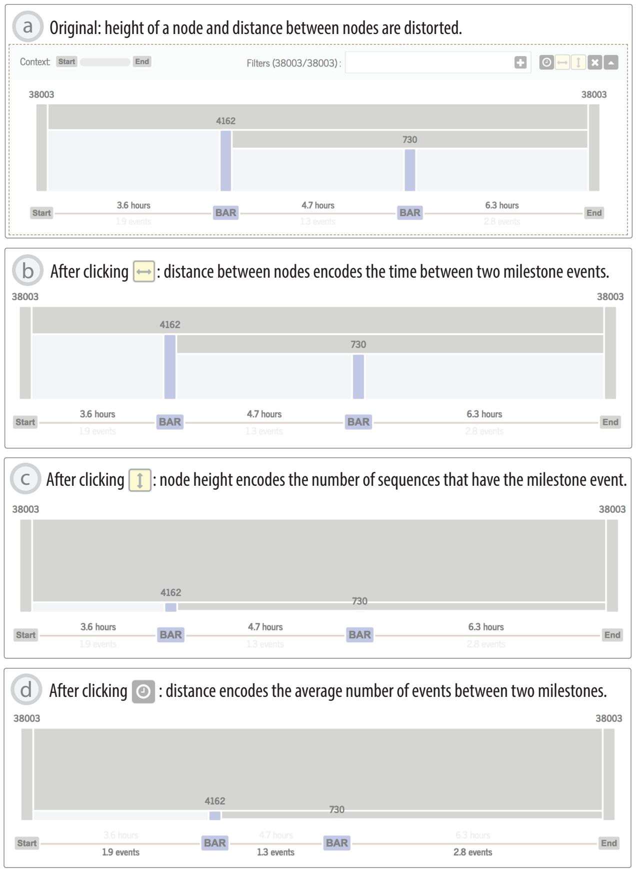


Fig. 6. (a) Node height and horizontal distance between nodes are originally distorted. Analysts can encode distance between nodes as (b) average duration or (d) average number of events in between. They can also encode node height as (c) the number of sequences that contain the milestone event (the number above a node).

As event names can potentially be very long, they are initially represented using two to three characters across different views. Analysts need to hover over the short event names to see the full name. Alternatively, analysts can click on ⇄ Change to Long Names to change all the event names in the interface to a longer version.

5.2 Interactions in Recursive Event Sequence Exploration

In MAQUI, recursive event sequence exploration is supported by allowing analysts to flexibly redefining the analytic focus and context. Analysts can **select** a set of segments in a flow visualization to make it the focus and **drill down** on the focus to make it a context.

To select a focus, analysts click on a rectangular region that corresponds to a set of segments. The selected region is highlighted in yellow to indicate that it becomes the focus (Fig. 1①). This triggers the three views on the right to display information about the focus. The frequent pattern view (Fig. 1b) shows a ranked list of frequent patterns mined from the focus. The attribute-value pair view (Fig. 1c) shows a ranked list of event/record attributes in the current focus. The raw event sequence view (Fig. 1d) visualizes the raw sequences in the focus. The raw sequences help analysts verify the mining results by seeing whether the generated patterns appear in these sequences.

To drill down on the focus to make it a context, analysts double-click on a set of segment to create a new panel. This enables analysts to apply the context-wide user actions on the focus. The context visualization in the top bar of the new panel provides the bigger context in which the drill-downed focus situates (C3).

As discussed in Section 4.3, the atomic actions in our framework can be classified into two types: *focus-only* (i.e. work only on a set of sequences only), and *context-wide* (i.e. work on one or more sets of sequences). *Context-wide* actions are accomplished within the workspace while *focus-only* actions are accomplished through the coordination between the workspace, and the frequent pattern and attribute-value pair views. The following sections illustrate how these atomic operations can be done in MAQUI.

5.2.1 The Search-Insert Loop (*context-wide*)

Searching and inserting an event attribute. To search within a context in a panel, analysts hover over the timeline, and a ✚ icon is shown. Analysts can click on ✚ to search for an event attribute from a menu (Fig. 7a).

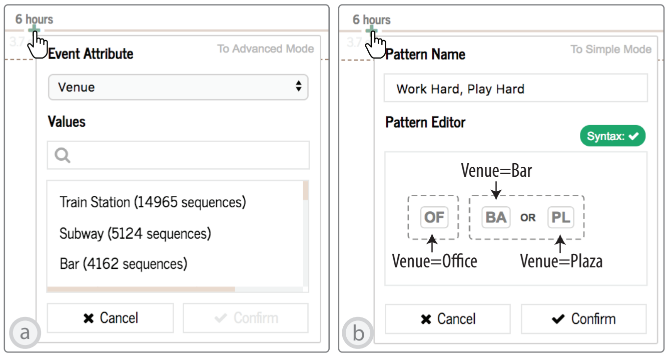


Fig. 7. MAQUI allows analysts to search for (a) event attributes and (b) freeform patterns in a context.

To insert an event, analysts select an event attribute (e.g., Venue) and a value (e.g., *Bar*) from the menu. After clicking on the ✓ Confirm button, MAQUI updates the entire flow visualization in the panel as shown in Figure 8a. A blue node is added to the flow visualization. We call the event attribute added by analysts to the flow visualization a milestone. A milestone is color-coded using the same color across different views. We do not color-code all the event attributes by default because the number of event attributes can outrun the number of distinguishable color channels. The rectangular region before the blue node represents the segments that occur before the milestone while the region after the blue node represents the segments occurring after the milestone. The dark gray region above the blue node represents the event sequences that do not contain the milestone. We decided to use the flow visualization design because the it is simple and easy to understand (C1). MAQUI also inserts the same milestone to the timeline to show the average duration and average number of events for the segments before and after the milestone.

Event attributes can be repeatedly inserted into a context. For example, analysts can insert Venue=*Train Station* before the blue milestone in Figure 8a. The updated context is shown in Figure 10a. Clicking on the red milestone event on the timeline in Figure 10a removes the event attribute from the context and restores the context to the state in Figure 8a.

Searching and inserting a user-defined pattern. Expert users might ask questions that involve user-defined patterns. Indeed, many query interfaces (e.g, EventFlow [28]) support searching for event sequences that contain a complex pattern specified by users. To cater to expert users, we designed an advanced mode for defining patterns. To enter the advanced mode, analysts clicks on ✚ on the timeline and click on the To Advanced Mode button in the menu (Fig. 7a top right). The advanced mode contains a pattern editor for creating freeform patterns by combining logical operators (i.e. **AND**, **OR**, **NOT**) and event attributes. In Figure 7b, the analyst is creating the pattern *Office*→*Bar* **OR** *Plaza*.

To insert a pattern into a context after defining it in the advanced mode, analysts click on ✓ Confirm. Inserting a user-defined pattern behaves the same as inserting an event attribute In Figure 8b, the flow visualization is divided into three rectangular regions that represents the segments before and after the user-defined pattern, and the sequences that do not contain the pattern. Using the visualization, analysts can answer questions including "what are the common behaviors in the event sequences that do not exhibit the defined pattern".

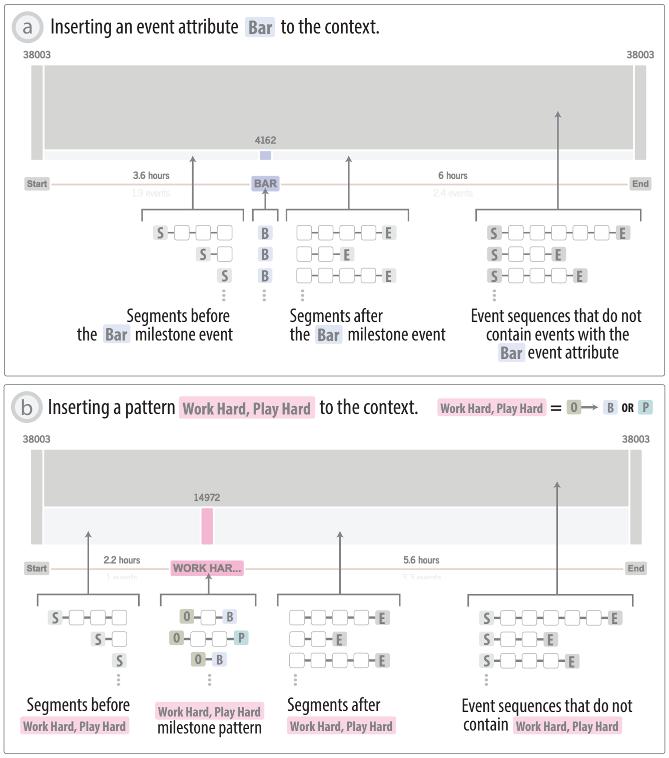

a Inserting an event attribute Bar to the context.
38003
38003
4162
Start
3.6 hours
BAR
6 hours
End
Segments before the Bar milestone event
Segments after the Bar milestone event
Event sequences that do not contain events with the Bar event attribute
b Inserting a pattern Work Hard, Play Hard to the context.
Work Hard, Play Hard = O → B OR P
38003
38003
14972
Start
2.2 hours
WORK HAR...
5.6 hours
End
Segments before Work Hard, Play Hard
Work Hard, Play Hard milestone pattern
Segments after Work Hard, Play Hard
Event sequences that do not contain Work Hard, Play Hard

Fig. 8. Inserting an event attribute or a pattern to the context.

5.2.2 The Specify-Filter Loop (*context-wide*)

Filtering by record attribute. One of the features demanded by the analysts is the capability to extract event sequences using record attributes (e.g., Q1 in Table 1). MAQUI enables analysts to filter out some sequences in a panel by specifying record attributes. When analysts click on ⊞ in the filter bar (Fig. 9a), MAQUI shows a menu to let analysts select a record attribute. If the record attribute is a categorical attribute, analysts simply select a value from the list (Fig. 9a); if the record attribute is numerical, analysts select a range of values (Fig. 9b). A record attribute filter is created by clicking on ✓ Confirm. Analysts can create multiple record attribute filters and can remove a record attribute filter by dragging it out of the filter bar.

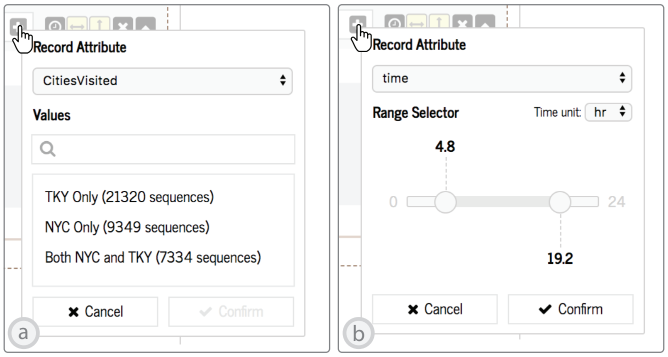


Fig. 9. Analysts can filter out some sequences in a context using (a) record attribute-value pairs or (b) time constraints.

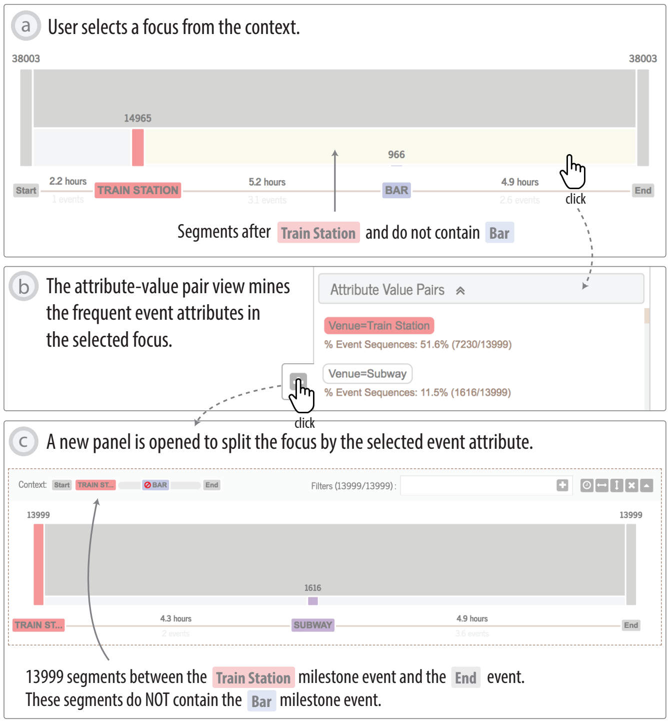
Setting time and number of events constraints. Albeit expressive, regular expression queries [5, 45] consider an event sequence as a pure sequence of events without considering duration between consecutive events. Yet, duration between events is important for answering questions in many domains [22, 32]. To enhance the expressiveness of MAQUI's query interface (C2), we allow analysts to set time gap constraints between events. Analysts can set time gap constraints by clicking on ⊞ in the filter bar and choose time from the Record Attribute pull down menu (Fig. 9b). After creating a time constraint, some sequences in a panel that do not satisfy the constraint will be filtered out. Analysts can also filter out the sequences in which the number of events falls outside a defined range. To do so, analysts select eventCount from Record Attribute list in the menu.

5.2.3 The Mine-Split Loop (*focus-only*)

As analysts select a set of segments in a panel, it becomes the focus and is highlighted in yellow. This selection action also triggers the mining algorithms to compute frequent patterns and frequent event attributes for the focus.

Mining and splitting by frequent event attribute. The attribute-value pair view (Fig. 1c) shows the mined values as a list, sorted in descending order of the percentage of sequences in the focus that contain an event. At a given time, the list comprises attribute-value pairs that belongs to a single attribute. Analysts can change the attribute by clicking on Attribute Name:.

To split the focus by an event attribute, analysts hover over the attribute-value pair, and a ⊞ button appears (Fig. 10b). Clicking on this button will automatically open a new panel, where the focus will be split and visualized. Figure 10c shows the result of splitting the selected segments in Figure 10a. The context visualization in the top bar (Fig. 10c) provides the context in which a focus situates (C3).


a User selects a focus from the context.

Segments after Train Station and do not contain Bar

b The attribute-value pair view mines the frequent event attributes in the selected focus.

Attribute Value Pairs

Venue=Train Station

% Event Sequences: 51.6% (7230/13999)

Venue=Subway

% Event Sequences: 11.5% (1616/13999)

c A new panel is opened to split the focus by the selected event attribute.

Context: Start TRAIN ST... BAR End

Filters (13999/13999) :

13999 segments between the Train Station milestone event and the End event. These segments do NOT contain the Bar milestone event.

Fig. 10. Splitting the selected focus using a frequent event attribute.

We choose not to split the focus in the original panel because the flow visualization may cause misinterpretation after several splitting operations if analysts were allowed to directly split the segments on it. Figure 11 shows the resulting visualization when analysts split the segments before Venue=*Train station* directly on the flow visualization using Venue=*Bar*. It is not straightforward to tell whether 2,900 means 2,900 event sequences out of the 14,965 event sequences that contain Venue=*Train station* or 2,900 out of all 38,003 event sequences. To keep the flow visualization intuitive, a new panel is created for splitting the focus (Fig. 10c).

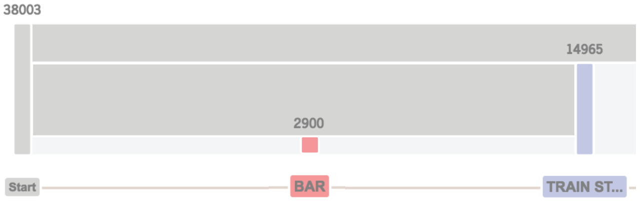


Fig. 11. Potential misinterpretation if analysts were allowed to split segments in the flow visualization directly. It is difficult to tell whether 2,900 means 2,900 event sequences out of 14,965 event sequences that contain the Venue=*Train Station* milestone event or 2,900 out of all 38,003 event sequences.

Mining and splitting by frequent pattern. As analysts choose a focus, sequential patterns are generated for the focus. We use VMSP [13] for mining frequent patterns since prior work has demonstrated that VMSP produces more compact patterns and effectively reduces the number of frequent patterns for inspection [26]. The frequent pattern view shows the mined patterns (Fig. 1b). When analysts hover over a frequent pattern, a menu is shown (Fig. 1②). By clicking on ⌄, a frequent pattern is expanded. This is useful when analysts want to see the full names of the events in the frequent pattern. Analysts can change the minimum support and the event attribute on which the algorithm is mined by selecting Minimum Support: and Attribute Name: at the bottom of the frequent pattern view respectively.

MAQUI offers two ways to split the focus by a mined pattern. Analysts can click on ⊞. This operation opens a new panel and the selected segments are split into three parts: the segments that occur before and after the pattern, and the sequences that do not contain the pattern. This operation enables analysts to incorporate a frequent pattern into an existing query to create new segments.

Alternatively, analysts can click on ↶. This button adds all the event attributes in a pattern and splits the focus by the event attributes in order. This allows analysts to see the average duration between consecutive event attributes in the pattern and to select segments between consecutive milestone events as the focus for further investigation. For instance, the bottom panel in Figure 1a is created after the analyst clicks on the ↶ button in Figure 1②.

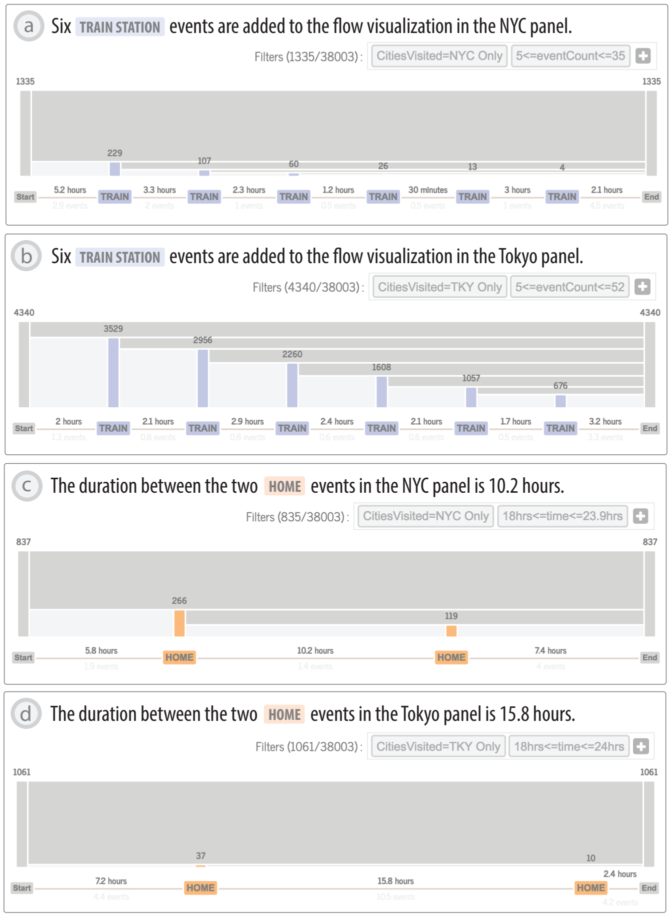
6 Usage Scenario

To elucidate how analysts might interweave queries and pattern mining during recursive event sequence exploration, let us consider how Jane, an event sequence analyst, explores the Foursquare dataset using MAQUI. We refer to the supplemental video for a demonstration of the usage scenario[2].

Jane is interested in comparing the daily routines of the people in NYC and the people in Tokyo. To begin with, she creates two panels in the query view. For the first panel (hereafter, the NYC panel), she specifies a record attribute filter CitiesVisited=*NYC only* using the filter bar (Fig. 12a) while for the second panel (hereafter, the Tokyo panel) she adds the CitiesVisited=*TKY only* filter (Fig. 12b). She also filters out the event sequences with less than five events for both panels. Clicking on the sequences in the Tokyo panel, she observes that there is only one frequent pattern with a support greater than 30%. She lowers the minimum support to 15% so that more frequent patterns are mined. Browsing through the list of frequent patterns, she is amused to see that 15.6% people in the Tokyo panel check-in six times in a train station. By clicking ↶, she adds all six Venue=*Train station* event attributes in this pattern to the Tokyo panel. Figure 12b shows the resulting flow visualization after it is split by the six Venue=*Train station* event attributes. As she wonders whether this pattern is common for the people in NYC, she adds six Venue=*Train station* events to the NYC panel by searching Venue=*Train station* from the context menu. She also clicks on ↕ to encode the height of nodes with the number above it. Figure 12a shows the result. In contrast to people in Tokyo, people in NYC do not check-in frequently at train stations.

Jane would like to know whether focusing only on the event sequences that span longer than 18 hours will give a more realistic depiction of the daily routines of the two groups of people. To keep a

[2] https://youtu.be/UhlBhDrejK0


a Six TRAIN STATION events are added to the flow visualization in the NYC panel.

Filters (1335/38003) : CitiesVisited=NYC Only 5<=eventCount<=35

b Six TRAIN STATION events are added to the flow visualization in the Tokyo panel.

Filters (4340/38003) : CitiesVisited=TKY Only 5<=eventCount<=52

c The duration between the two HOME events in the NYC panel is 10.2 hours.

Filters (835/38003) : CitiesVisited=NYC Only 18hrs<=time<=23.9hrs

d The duration between the two HOME events in the Tokyo panel is 15.8 hours.

Filters (1061/38003) : CitiesVisited=TKY Only 18hrs<=time<=24hrs

Fig. 12. Analyzing the Foursquare dataset using MAQUI. (a-b) Jane observes that compared with people in Tokyo, people in NYC check-in less frequently at a train station. (c-d) The average durations between two Venue=*Home* milestone events are 10.2 hours and 15.8 hours respectively for the NYC panel and the Tokyo panel.

history of what she did, she opens two new panels and specifies the record attribute filters CitiesVisited=*NYC only* and CitiesVisited=*TKY only* respectively. She filters out the event sequences that span less than 18 hours for both new panels. When she clicks on the event sequences in the new NYC panel, the attribute-value pair view is updated to show a ranked list of event attribute-value pairs. Jane notices that 31.9% of the people in NYC check-in at their home. She adds Venue=*Home* from the attribute-value pair view to the NYC panel to split the event sequences in the NYC panel. To continue her recursive exploration, she selects as the focus the segments to the right of the *Venue=Home* milestone event in the NYC panel. This set of segments contains the events that occur after the people in NYC check-in at home. The attribute-value pair view shows that most people (44.7%) check-in at their home again. Jane thinks that people probably check-in again at the end of the day when they go back from work. She adds another Venue=*Home* event to the right of the first Venue=*Home* event to split the segments after the first Venue=*Home*. The timeline tells Jane that the average duration between the two Venue=*Home* milestone events for the people in NYC is 10.2 hours (Fig. 12c), which probably indicates the average working hours for the people in NYC. Curious about the time between two Venue=*Home* events in the Tokyo panel, Jane adds them to the new Tokyo panel and observes that the average time between the two events is 15.8 hours (Fig. 12d). The long duration between the two Venue=*Home* piques Jane's interest. She wonders whether it is related to the long working hours in Japan, which is a well-known issue [30].

Finally, Jane wants to know whether she can observe patterns of a typical day from the data. Many people start their days by commuting to somewhere and end their days by commuting back home. She would like to investigate if people do similar things in between. Jane first creates a new panel and clicks on ✚ to open the context menu. Using the advanced mode, she creates a pattern (*Home*→*Subway* **OR** *Bus Station*) and names it as "Leaving Home". She creates another pattern (*Subway* **OR** *Bus Station*→*Home*) after the "Leaving Home" pattern and names it as "Going Back Home". The flow visualization shows that only 34 sequences have a "Going Back Home" pattern followed by a "Leaving Home" pattern. To inspect the frequent patterns in the segments between the "Leaving Home" and "Going Back Home' patterns, Jane selects the rectangular regions between the two milestone patterns. She did not find frequent patterns with a support above 15%. This illustrates a wide variety of activities people do between "Leaving Home" and "Going Back Home" in a typical day. Jane lowers the support to 5% and observes that 5.9% (2 out of 34) of the sequences have the pattern *Movie Theater*→*Movie Theater* (checked-in twice in a movie theater). She hovers over the pattern and clicks on ✚ to split the segments between the "Leaving Home" and "Going Back Home" patterns further. This allows her to dive into what happens before and after the *Movie Theater*→*Movie Theater* pattern.

7 Case Studies

We conducted two real-world case studies with domain experts, one in healthcare and one in marketing, to evaluate the efficacy of MAQUI.

7.1 Software Usage Logs

Background. The first case study was conducted with two data analysts exploring software usage logs. The analysts were interested in understanding frequent patterns of their users with respect to specific milestone events. For example, they were interested in questions such as "After people print, what are the things they do most?" Both analysts had experience in event sequence analysis, but had not previously used MAQUI. Prior to MAQUI, the data analysts had used software to analyze common next and previous events, but found it difficult to distinguish looping sequences and repeating events. They had also tried Markov Decision Processes (MDP), but found them difficult to iterate over and interpret on-the-fly.

Method. The session was conducted remotely, using screen sharing and video conferencing between the two analysts and three researchers from our team. The analysts were provided access to MAQUI on their own machines and, after a brief overview of the interface, talked aloud as they used the tool to explore their own dataset. The session lasted approximately one hour and was recorded.

The dataset consisted of 140,193 events across 3,638 users and 8,738 sessions sampled from logs in March 2017, with each event corresponding to an application feature (e.g., Category=*Save As*,and Category=*Print*) used at a given point in time. There were 2,339 unique features in the dataset, grouped into 453 subcategories and 268 categories. We grouped the events by user ID and session ID to form sequences.

Analysis Process. The analysts began by inspecting the workspace, which displayed all records by default, and noted that on average, users workflows took 2.1 hours. From here, the analysts **searched** and **inserted** the Category=*Print* event followed by Category=*Print Success*. After noticing the surprising time gap of 50 minutes, the analysts **selected** the segment after Category=*Print Success* to **mine** the most common patterns after their queried pattern. Aside from inspecting the mined patterns more closely, the analysts used the raw sequence panel to verify the patterns matched their expectations of the data.

As is often the case in exploratory data analysis, questions beget more questions, and the analysts were able to iteratively repeat processes of querying, mining, and freeform exploration for additional questions they had.

Feedback and Takeaways. In general, the analysts appreciated the interwoven querying and mining, noting that, "*I really like how you don't have to drill down event by event or guess what the common patterns are. The fact that [MAQUI] suggests them is very nice.*"

Providing focus to the mining through querying and filtering made the patterns more discernible. *"Here I've had a case where they did Text Move→Resize→Copy→Paste. That sounds like a real workflow that somebody would do, and I'm not sure that I've ever been able to get to* *that level before [with the other tools]... [It] is something we always struggled for.*"

Further, the machine-aided pattern mining and visual representation improved the overall interpretability of the results, with one analyst noting, "*It's useful that it looks more like user behavior than just machine output [logs].*"

The analysts also seemed to use MAQUI as verification for their understanding of the dataset. In particular, they commonly used the raw sequence view to confirm that insights were inline with their expectations ("*Oh yeah, this all makes sense*" [more inspection] "*Yeah, this totally makes sense.*") or to understand why they deviated from their expectations ("*Does that seem right?*").

By leveraging the benefits of automatic pattern mining to find relevant, important events combined with human-guided querying to set focus, the analysts were able to arrive at interesting insights more easily using MAQUI than with previous efforts.

7.2 Workflows in a Pediatric Emergency Department

Background. To understand the utility of MAQUI in the healthcare domain, we conducted a case study with a highly experienced health informatics professional with a medical degree. The expert commented that the overarching problem that health practitioners would like to have insights on is what care patterns lead to better results at the lowest costs. For instance, he would like to investigate the patterns of care for each doctor, how much their patients spend, and what the disposition ultimately is. Similar to our previous study, the expert had experience with event sequence analysis, but had not used MAQUI before.

Method. We conducted a one-hour onsite interview with the expert. During the interview, the expert was invited to think aloud while exploring a dataset using MAQUI. The dataset consisted of 295,686 events that occurred in the emergency department (ED) of a major pediatric hospital between Jan 2013 and Jan 2014. Each event sequence corresponds to the process a patient went through in his/her visit to the emergency department. Examples of events include Category=*Arrival*, Category=*ED Exit*, Category=*Triage Start*, and Category=*Triage End*. There are in total 10,020 event types that are grouped into 28 categories. The events are grouped by visit ID to form 3,919 event sequences. Our participant was knowledgeable about the dataset.

Analysis Process. During the interview, the expert began by investigating the average time between when a patient arrived and s/he saw the first attendee. After **splitting** the event sequences by Category=*Arrival* and Category=*First Attendee*, he saw that the average time is 55.3 minutes, which is not surprising. However, as he **selected** the segments before Category=*Arrival*, he found some abnormal patterns from the frequent pattern view: 21.7% of the patients had the pattern *Diagnose→Diagnose* even before they arrived at the hospital. We later found that this was indeed a data quality issue introduced when we cleaned the data — MAQUI helped us to find unexpected data quality issues that were unknowingly introduced.

The expert participant then used the frequent patterns mined by the system to **split** the current focus for recursive exploration. When he was investigating the patterns between Category=*Arrival* and Category=*Disposition*, he saw the pattern *Triage Start→Triage End→First Attendee→Medication Start→Medication Ordered* and used it to **split** the segments between Category=*Arrival* and Category=*Disposition*. He **merged** the segments between Category=*First Attendee* and Category=*Medication Start*, and the segments between Category=*Medication Start* and Category=*Medication Ordered*. He then **selected** the new segments between Category=*First Attendee* and Category=*Medication Ordered* to **mine** frequent events in between.

Feedback and Takeaways. Overall, the expert participant liked the system's capability despite the data quality issues he encountered during the analysis: *"I think this has a lot of promise but I think we need a better dataset"*. He particularly appreciated how he could mine patterns from different segments and with different criteria: *"What is going to be interesting is to take one of these pairs [of milestone events] and analyze different branches to understand where they [different branches] lead to and how they differ"*. Throughout his analysis, he noted the ease at which care patterns could be examined and was particularly impressed with the capability of recursively querying and mining, something he referred to as *"the key to deep understanding, improvement, and potential redesign of healthcare processes"*

8 Discussion

One of the limitations of MAQUI concerns scalability. While MAQUI can effectively handle the Foursquare dataset that contains more than 30,000 event sequences, it is highly sensitive to datasets with a large number of event types. With more than 1,000 event types, the speed of mining frequent events and frequent patterns significantly degrades. This would introduce high latency to user interactions and potentially hampers users' performance during analysis [24]. However, as research in sequential pattern mining algorithms matures, we anticipate that faster algorithms will be developed. By being agnostic to the sequential pattern mining algorithms used, our technique can be widely adopted when faster algorithms are available.

Data quality is another issue that emerged during both case studies. For small data quality issues, the patterns generated by mining algorithms will often not be affected because these algorithms summarize event sequences by abstracting away low-level details. Large data quality issues such as systematic errors introduced during data wrangling, however, might contaminate the patterns mined. As suggested by the analysts in the software usage logs study, one potential solution is to let analysts wrangle the data on the fly as they explore the patterns. For instance, analysts may want to remove a particular events from all event sequences or combine two events into one during their analysis.

MAQUI takes advantage of the multi-panel design to support recursive event sequence exploration, reduce visual clutter and facilitate cohort comparison. Yet, managing a large number of panels can be difficult because of a limited screen real estate and high cognitive load involved in monitoring. Currently, MAQUI only supports simple panel management operations such as collapsing and closing a panel. Other techniques such as rearranging and grouping panels can potentially reduce cognitive load in recursive event sequence exploration. Furthermore, MAQUI only supports visual comparison of cohorts by juxtaposition. Strategies such as superposition [15], cloning events from one panel to another and rearranging panels can be adopted to help analysts gain insights into the differences between cohorts. In future, we would like to explore the best practices of panel comparison by following Gleicher's guidelines [14].

It has been known that volume and variety of event sequences pose significant challenges to event sequence analysis [10]. Other challenges, including data quality, diverse tasks in different domains, and complicated logic of event sequences, further create barriers to making sense of event sequence data. A single solution (e.g., querying only, and mining only) is insufficient for dealing with all these complexities that frequently appear in real-world event sequence exploration. To address real-world challenges, research has to be done on how various existing techniques (e.g., visualization, data wrangling, querying, and pattern mining) can be combined and interoperate. Our work takes a step in this direction by contributing an understanding of how querying and mining can be interwoven during event sequence exploration.

9 Conclusion

In this paper, we introduced MAQUI, a visual analytics system that enables analysts to interweave queries and pattern mining for recursive event sequence exploration. Based on the analysts' tasks, we identified the need for combining querying and mining to explore event sequences in a recursive manner. Following this observation, we proposed a framework of interwoven querying and mining that describes the atomic user actions for recursively refining the analytic context and focus during analysis. Through two real-world usage scenarios, we demonstrated the utility of our approach in event sequence exploration. As the variety and volume of event sequence data continue to increase, fascinating challenges emerge. MAQUI provides an important foundation to address these challenges. We hope that MAQUI inspires new approaches to event sequence exploration.

References

[1] Adobe analytics - now part of adobe analytics cloud. `https://www.adobe.com/data-analytics-cloud/analytics.html`. [Accessed: 31st March 2018].

[2] Google analytics solutions - marketing analytics & measurement. `https://www.google.com/analytics/`. [Accessed: 31st March 2018].

[3] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 429–435. ACM, 2002.

[4] M. V. Bjarnadóttir, S. Malik, E. Onukwugha, T. Gooden, and C. Plaisant. Understanding adherence and prescription patterns using large-scale claims data. *PharmacoEconomics*, 34(2):169–179, 2016.

[5] B. C. Cappers and J. J. van Wijk. Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):532–541, 2018.

[6] E. Carter, R. Burd, M. Monroe, C. Plaisant, and B. Shneiderman. Using eventflow to analyze task performance during trauma resuscitation. In *Proceedings of the Workshop on Interactive Systems in Healthcare (WISH 2013)*, 2013.

[7] Y. Chen, P. Xu, and L. Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):45–55, 2018.

[8] F. Du, C. Plaisant, N. Spring, and B. Shneiderman. Eventaction: Visual analytics for temporal event sequence recommendation. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 61–70. IEEE, 2016.

[9] F. Du, C. Plaisant, N. Spring, and B. Shneiderman. Finding similar people to guide life choices: Challenge, design, and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 5498–5544. ACM, 2017.

[10] F. Du, B. Shneiderman, C. Plaisant, S. Malik, and A. Perer. Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1636–1649, 2017.

[11] J. A. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *IEEE Symposium On Visual Analytics Science And Technology*, pp. 167–174. IEEE, 2006.

[12] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, and Y. S. Koh. A survey of sequential pattern mining.

[13] P. Fournier-Viger, C.-W. Wu, A. Gomariz, and V. S. Tseng. Vmsp: Efficient vertical mining of maximal sequential patterns. In *Canadian Conference on Artificial Intelligence*, pp. 83–94. Springer, 2014.

[14] M. Gleicher. Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2018.

[15] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[16] D. Gotz and H. Stavropoulos. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1783–1792, 2014.

[17] D. Gotz, S. Sun, and N. Cao. Adaptive contextualization: Combating bias during high-dimensional visualization and data selection. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pp. 85–95. ACM, 2016.

[18] D. Gotz, F. Wang, and A. Perer. A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. *Journal of Biomedical Informatics*, 48:148–159, 2014.

[19] F. Haag, R. Krüger, and T. Ertl. Vespa: A pattern-based visual query language for event sequences. In *VISIGRAPP (2: IVAPP)*, pp. 50–61, 2016.

[20] J. Jin and P. Szekely. Querymarvel: A visual query language for temporal patterns using comic strips. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 207–214. IEEE, 2009.

[21] D. A. Keim, J. Schneidewind, and M. Sips. Fp-viz: Visual frequent pattern mining. In *InfoVis*, 2005.

[22] J. Krause, A. Perer, and H. Stavropoulos. Supporting iterative cohort construction with visual temporal queries. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):91–100, 2016.

[23] B. C. Kwon, J. Verma, and A. Perer. Peekquence: Visual analytics for event sequence data. In *ACM SIGKDD 2016 Workshop on Interactive Data Exploration and Analytics*, vol. 1, 2016.

[24] Z. Liu and J. Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, 2014.

[25] Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. In *Computer Graphics Forum*, vol. 36, pp. 527–538. Wiley Online Library, 2017.

[26] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):321–330, 2017.

[27] T. E. Meyer, M. Monroe, C. Plaisant, R. Lan, K. Wongsuphasawat, T. S. Coster, S. Gold, J. Millstein, and B. Shneiderman. Visualizing patterns of drug prescriptions with eventflow: A pilot study of asthma medications in the military health system. Technical report, Office of the Surgeon General of the Army, 2013.

[28] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, 2013.

[29] M. Monroe, R. Lan, J. Morales del Olmo, B. Shneiderman, C. Plaisant, and J. Millstein. The challenges of specifying intervals and absences in temporal queries: A graphical language approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2349–2358. ACM, 2013.

[30] H. Ono. Why do the japanese work long hours?: Sociological perspectives on long working hours in japan. *Japan labor issues*, 2(5):35–49, 2018.

[31] S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas. Incremental and interactive sequence mining. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pp. 251–258. ACM, 1999.

[32] A. Perer and F. Wang. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, pp. 153–162. ACM, 2014.

[33] C. Plaisant, S. Lam, B. Shneiderman, M. S. Smith, D. Roseman, G. Marchand, M. Gillam, C. Feied, J. Handler, and H. Rappaport. Searching electronic health records for temporal patterns in patient histories: A case study with microsoft amalga. In *AMIA Annual Symposium Proceedings*, vol. 2008, p. 601. American Medical Informatics Association, 2008.

[34] P. J. Polack, S.-T. Chen, M. Kahng, M. Sharmin, and D. H. Chau. Timestitch: Interactive multi-focus cohort discovery and comparison. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 209–210. IEEE, 2015.

[35] P. J. Polack Jr, S.-T. Chen, M. Kahng, K. de Barbaro, M. Sharmin, R. Basole, and D. H. Chau. Chronodes: Interactive multi-focus exploration of event sequences. *arXiv preprint arXiv:1609.08535*, 2016.

[36] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23, 2000.

[37] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, 2014.

[38] K. Vrotsou, J. Johansson, and M. Cooper. Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):945–952, 2009.

[39] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 457–466. ACM, 2008.

[40] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: visualizing an overview of event sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1747–1756. ACM, 2011.

[41] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, and B. Shneiderman. Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interacting with Computers*, 24(2):55–68, 2012.

[42] K. Wongsuphasawat and B. Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pp. 27–34. IEEE, 2009.

[43] A. P. Wright, A. T. Wright, A. B. McCoy, and D. F. Sittig. The use of

sequential pattern mining to predict next prescribed medications. *Journal of Biomedical Informatics*, 53:73–80, 2015.

[44] D. Xin, X. Shen, Q. Mei, and J. Han. Discovering interesting patterns through user's interactive feedback. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 773–778. ACM, 2006.

[45] E. Zgraggen, S. M. Drucker, D. Fisher, and R. DeLine. (s|qu) eries: Visual regular expressions for querying and exploring event sequences. 2015.

[46] Z. Zhang, D. Gotz, and A. Perer. Iterative cohort analysis and exploration. *Information Visualization*, 14(4):289–307, 2015.