Temporal Search and Replace: An Interactive Tool for the Analysis of Temporal Event Sequences

Rongjian Lan[1,2], Hanseung Lee[1,2], Megan Monroe[1,2], Allan Fong[1,2], Catherine Plaisant[2], Ben Shneiderman[1,2]
Department of Computer Science[1] & Human-Computer Interaction Lab[2]
University of Maryland, College Park, MD 20742
{rjlan, hanseung, madeyjay, afong12, plaisant, ben}@cs.umd.edu

***Abstract*—Visualization of temporal event data is increasingly important for the analysis of a broad range of data including electronic health records, web logs, and financial data. In many analytic tasks, users need the capability to manipulate the data to reveal patterns and make insights. To support this analytic need, we introduce a novel temporal search and replace tool (TSR) implemented in our existing EventFlow visual analytic system to facilitate visual-language-based search and replacement of temporal event sequences. We also introduce two types of search constraints: repetition and permutation that integrate regular expression concepts into temporal event sequence searching. We present the replacement strategy for event sequences under these constraints. Example use cases are discussed where TSR solves problems both on temporal event data analysis and simplification. Finally we report on a usability study with 10 participants.**

I. Introduction

Visualization of temporal event data is increasingly important for the analysis of a broad range of data including electronic health records, web logs, and financial data. Analysts using visual analytic sytems usually explore the data in the following steps: 1) seeing the overview; 2) narrowing down the data to focus on a subset; 3) applying special operations (e.g., aligning by a focal event); 4) visual scanning of the data. This process is sufficient to discover most of the explicit knowledge embedded in the occurrence of events and their temporal relationship. However, the way that the raw data is recorded is generally not perfect in terms of generating a concise and meaningful representation of what's happening, which renders such analysis process limited to exploring only what the raw data explicitly shows. To dig into the hidden knowledge, analysts need the capability to manipulate the data in the way that generates a better visual representation that reveals the hidden knowledge.

The way the data is logged is usually a function of logging convenience, lacking the consideration of future data analysis benefits. Complex events (e.g., pregnancy) could be represented as an interval event with a start and an end time, but are usually not logged as such, and the information has to be gleaned from a series of point events with single timestamps (e.g., doctor visits, sonograms, vitamin prescriptions). The consequence of this workaround is a less helpful representation of the real world events, making the analysis of the data difficult. Suppose users need to analyze a medical history dataset with point events indicating the prescription of different drugs, admission to hospitals, tests, and discharge from hospitals. In this dataset, users may be interested in what drugs were prescribed while the patient was hospitalized. Previous visualization techniques—such as LifeLines2 [1]—can help answer this problem by presenting the event data on timelines. However, Users still have to find the occurrence of the "Admission" event, tracing forward to find the corresponding "Discharge" event while looking for the drug prescription events. Analysing the raw data in this manner is not only tedious, but also error-prone.

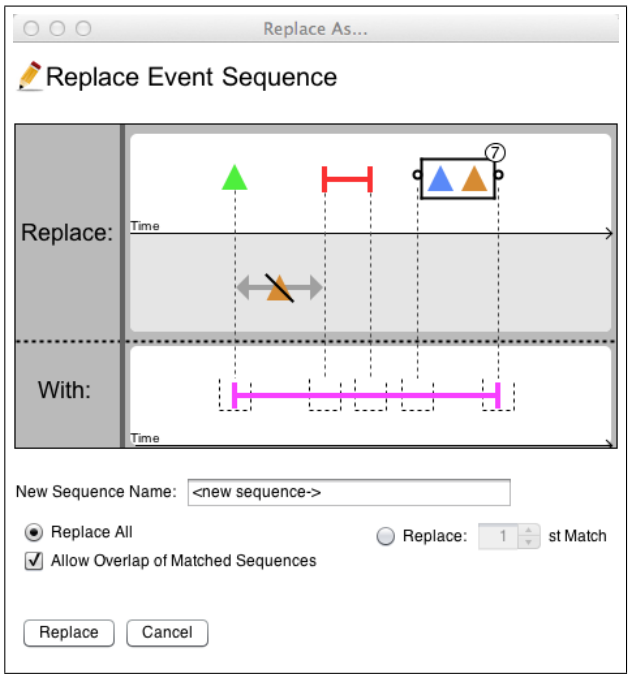

Fig. 1: The temporal search and replace interface, including a search panel where users specify a search event sequence, a replace panel where replacement events are added and linked to the search sequence, and the replacement control options. (Colored triangle - point event; Colored line - interval event)

This paper introduces a novel visual-language-based temporal search and replace tool (TSR) (Fig. 1)—implemented in our existing EventFlow visual analytic system [2]—that allows users to transform temporal data with user-defined search and replacement of temporal event sequences. Two types of search constraints, repetition and permutation, are introduced to bring regular expression concepts into temporal visual query

language. We believe that proper replacement of temporal event sequences is helpful to discover hidden knowledge in the data. We also demonstrate that TSR is intuitive and helpful at doing temporal event data manipulation.

II. Related Work

A. Search of Temporal Event Sequences

Search of temporal event sequences to find all the occurrences of a temporal pattern is important in temporal event data analysis. Many temporal event sequence querying tools have been developed [3, 4, 5, 6, 7]. Database researchers have produced numerous works on designing temporal algebras and languages to support access to temporal relationships. Snodgrass [4] proposed extensions to the well-known database query language, SQL, to encompass temporal relationships. However, this work centers around command-based query languages, which have a notoriously steep learning curve, making it difficult for most users to specify desired temporal patterns, thus limiting the use of these techniques.

Just as the graphical user interface revolutionized the way people interact with computers, visual query languages reduce user confusion by conveying information with meaningful graphical representations. Visual Query Languages (VQL) were adopted as the search interface in previous temporal event sequence querying systems. For example, Jin et al. [8, 9] built on comic strip metaphors and proposed a novel VQL that is easily understood and specified by users. However, their tool doesn't allow users to manipulate the data based on search results. Monroe et al. [10] developed a powerful VQL that incorporates both interval events and absences of events. Its design is based on the rule of "query-by-example", and the search algorithm expands on the Temporal Pattern Search algorithm [11]. Although these visual query languages can simplify the specification of complex temporal pattern queries, they don't allow users to manipulate the data based on the search result. We base our search interface on [10] because it's suitable for integration with the replacement interface. We also extend its search capability by introducing two search constraints: repetition and permutation.

B. Search and Replace

"Search and Replace" (a.k.a "Find and Replace") features are widely utilized in text editing softwares, such as Microsoft Word and Notepad++. The concept of "Search and Replace" also led to tools that help users rapidly process different kinds of data to find information or propagate changes. Kurlander and Feiner [12] demonstrated the feasibility to search for graphical objects and replace the search results with new illustrations. Haag [13], Cody [14], and Mustafa [15] showed how replacement helps in improving users' understanding of graphs and maps. TSR for the first time brings "Search and Replace" concept into temporal event data analysis.

C. Domain-Knowledge Based Temporal Abstraction

Researchers in clinical event data analysis developed software facilities to support temporal event data abstraction (i.e., representation of complex event sequences with simpler ones). Goren-Bar et al. [16] introduced KNAVE II, a system using domain knowledge to inform temporal abstraction of event patterns. In KNAVE II, a domain-expert-supported knowledge base is provided for users to search abstraction rules and apply them to the raw data to generate context-specific abstractions. While this system is powerful at abstraction based on built-in domain knowledge, it doesn't provide users with the flexibility in defining their own specific abstraction rules. Post and Harrison [7] created PROTEMPA, which allows for abstraction of clinical event sequences and retrieval of patient population with characteristics determined by the temporal relationships between event sequences. Although PROTEMPA gives users flexibility to define their own abstraction rules, the command-based language poses a big challenge for general clinical data analysts without a solid database background. By contrast, TSR provides users with an intuitive visual interface that helps users easily search for event sequences and replace them with different user-defined event sequences.

III. Temporal Search and Replace

Our TSR helps users manipulate temporal event sequences by supporting visual-language-based search and replacement functionalities. It contains four components: A) Search Panel, where users specify the event sequence they want to manipulate using the visual query language [10]; B) Replace Panel, where users specify the replacement event sequence linked to the search event sequence; C) Control Panel, which controls how the replacement operation is conducted and defines the name of the replacement sequence (defaulted to the concatenation of names of all the replacement events); D) Replacement History List, which allows users to review and control the replacements if needed.

A. Search Panel

We built our search interface on the existing EventFlow visual query interface, which is powerful at specifying complex event sequence queries involving both point and interval events, absences of events and related time constraints. The upper half of the search panel is allocated for the occuring event while the lower half is for the absence of event. Queries are specified by adding query elements to a designated canvas. Users click on the canvas to bring up an editor panel where they specify the element type, whether or not it occurs, and any additional time constraints [10].

The existing search capability in EventFlow only supports one-to-one match of events, which means the users need to specify an occuring query event for each event in the searched result. This process can be cumbersome or impossible when it comes to cases where repeated events or any permutations of events is the target. These search constraints of repetition and permutation of events are analogies to the regular expression's concepts, quantifier and alternation, which are widely used and provide great searching power in string matching. To further empower TSR's event sequence search capability as well as introduce the regular expression based concepts into

temporal events visual query language, we extend the original visual query language [10] to support search of repetition and permutations of temporal events.

1) Repetition Constraint: The repetition constraint allows users to search for event sequences consisting of multiple occurrences of an event or a sequence of events without explicitly specifying each constituent event. To visually present the repetition constraint, we use a rectangle with a repetition number indicator on its upper right corner (Fig. 2 left). To add such a search constraint on the query events, users can lasso the target events to bring up the constraint editor (Fig. 3), where they specify the constraint type and the options.

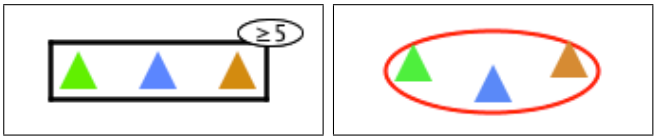

Fig. 2: The visual metaphor for repetition constraint (left) and permutation constraint (right).

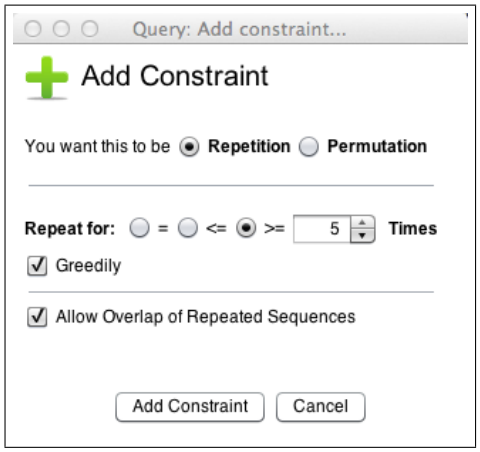

Fig. 3: The pop up panel for specifying constraints

In the repetition editor panel, users can specify the number of repetitions, the repetition types (greater/equal, less/equal and equal), whether matching is greedily done and whether the matched sequences can overlap with each other (Fig. 3). The concept of greediness is borrowed from regular expression, which defines that a greedy match will try to do as many as matches until failure. Temporal event sequences are generally treated as non-continual, implying event sequences can overlap in terms of temporal duration. For example, in sequence "A→A→B→B" (→indicates a certain elapsed time), the sub-sequence "A(first)→B(first)" and "A(second)→B(second)" overlaps. Therefore, we provide a checkbox "Allow Overlap of Repeated Sequences" for users to decide whether the repeatedly matched temporal event sequences are allowed to overlap with each other.

2) Permutation Constraint: In many cases, users are interested in the occurrence of multiple events without requirements on their temporal relationship. One example is "Drug A followed by both Symptom 1 and 2 without order." To fulfill this need, we introduce the permutation constraint that discard the temporal relationship requirement among the searched events. The visual metaphor we use for permutation constraint is a red circle (Fig. 2 right), which is inspired by the common use of circle to illustrate the concept of set in computer science. Similarly, users can add permutation constraints by simply lassoing the target events and choose the "permutation" radio button (Fig. 3). There are no options for this constraint.

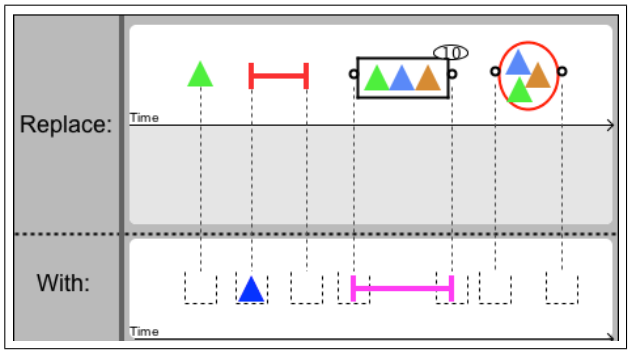

Fig. 4: The correspondance between the search event sequence and replacement event sequence are shown with dotted lines. The search sequence will be replaced with a point and an interval event with timestamps of their replacement slots

B. Replace Panel

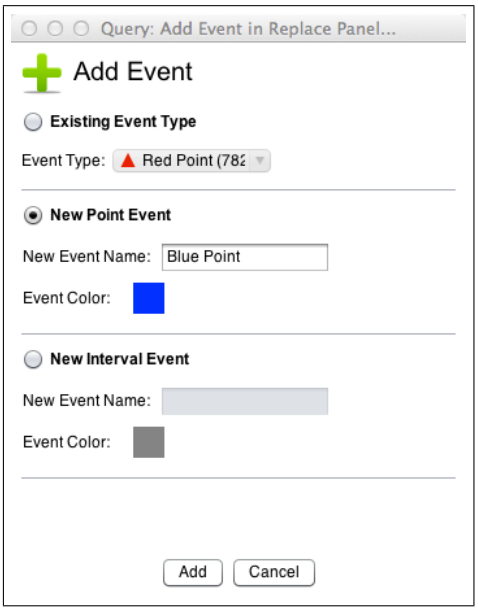

Fig. 5: The pop up panel for specifying the replacement events. Users can choose between existing event, new point or interval event.

In most text editors, the "Search and Replace" functionality supports a simple sequence-to-sequence replacement strategy, that is, replacing the whole string with another string. We allow users to replace event sequences on an event-to-event basis, by specifying each replacement event in relationship to a specific

time in the search event sequence. We call that specific time, to which the replacement events can correspond, a replacement slot. The replacement slot can be the timestamp of a occurring search event, or the timestamps of the first and last events in a search sequence under the constraint. There is no replacement slot allocated for the events inside the repetition or permutation constraints, because their exact time is less meaningful than the duration of the whole sequence. There are no replacement slots for the absence of events because they are a semantic requirement on the search result and not matched to a specific event or time. The replacement slots are visually represented as a box with dotted line connected to a specific position in the search event sequence (Fig. 4).

After users completely specify the search event sequence in the search panel, all the replacement slots are displayed in the replace panel. Users can add replacement event with existing point and interval events, or new point and interval events (not existing in the original data), by clicking on the replace panel and specifying the replacement event in the pop up panel (Fig. 5). For new events, users need to specify the name and the color of it before adding in the replace panel. After the addition, the new event is added to the existing event list for later reuse. After users specify the replacement event and click "Add", that event will be placed at the replacement slot closest to the click position. The timestamp of the replacement event is defined to be the same as its corresponding replacement slot. To change the location of the replacement events, users can simply drag the replacement events to somewhere close to the target replacement slot and it automatically snaps to it.

C. Control Panel

The control panel is provided for users to define the replacement sequence's name and control how the replacement should be done. The new sequence name is by default the concatenation of all the replacement events' names, which can be changed if needed. Users can choose to replace all the matched event sequences or only a specific occurrence in the record by selecting "Replace All" or "Replace Nth Match" option. The "Allow Overlap of Matched Sequences" checkbox (only available when replacing all) is for users to decide whether overlaps are allowed in matched event sequences.

D. Repacement History List

To help users memorize and control the event sequence replacement they have done, we provide a replacement history list for users to view and do retrospective control of the applied replacement operations. For each successfully applied replacement operation, a new item with the replacement name, its visual representation, the number of the replaced sequences and other control options is added to the top of the replacement history list (Fig. 6). By default, the replacement operation removes the matched sequences and adds the replacement sequences in the data. In case users want to bring back the original event sequence, they can check the checkbox to the right of the search sequence in the replacement history list. Similarly, if users need to temporarily hide the replacement sequence, they can uncheck the checkbox to the right of it. Sometimes users make mistakes and need to modify the replacement they specified. We support replacement reformulation by simply clicking on the replacement item and modifying it in the search and replace interface.

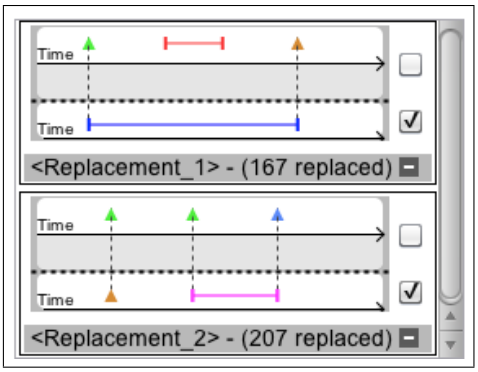

Fig. 6: The replacement history list shows the replacement name and the number of replaced event sequences. Users can control the visibility of the original and the replacement event sequences by checking the checkbox to the right, and can cancel the replacement by clicking the minus sign.

IV. Potential Use Cases

The introduction of the Temporal Search and Replace feature in the EventFlow visualization system greatly extends its analytic capability. Here we describe two potential use cases to demonstrate the usefulness of TSR in EventFlow.

Case 1: Analysis of Events in Context

In the data analysis of temporal event sequences, many interesting patterns are implied not by the occurrence of a single event, but rather by the complex temporal relationship between events or event sequences. However, most of the current visual analytic systems are focused on the event-level analysis, rather than sequence-level analysis that involves event-sequence relations. One potential use case of TSR is to replace complex event sequences with one or a few events, so that event sequences are brought into the analytic scope of event-level analysis. Consider again the case of medical history analysis described in Section 1. In this dataset, a general analytic task such as "What drugs are prescribed during hospitalization" can be better answered by replacing the event sequence "Admission→Discharge" with a single interval event called "Hospitalization", and then visually scanning or by issuing an overlap query (supported in EventFlow).

The steps taken to create the hospitalization interval event are as follows. First, in the search panel, users add two query events of "Admission" and "Discharge" in order. Second, in the replace panel, users add a new interval event called "Hospitalization" colored with blue (Fig. 7 Left). Then, they specify to replace all the matched sequences by checking the "Replace All" radio button, after which they click the "Replace" button to issue the replacement operation. Note that

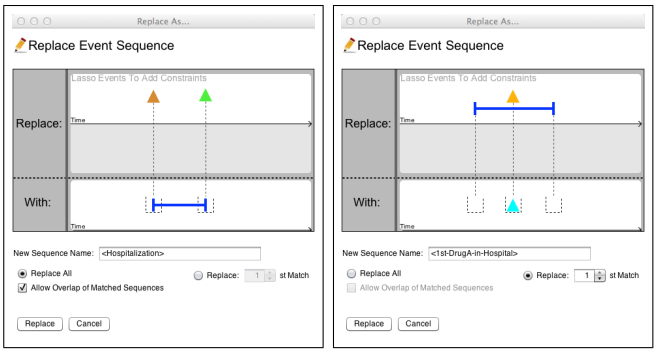

Fig. 7: Replace "Admission" and "Discharge" point events with a new interval event "Hospitalization" (left); Replace the first occurrence of "Drug A" during "Hospitalization" with a single point event "1st-DrugA-in-Hospital" (right).

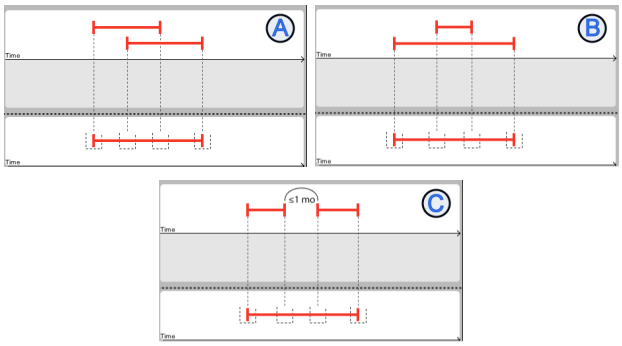

Fig. 8: Remove overlaps (A); Remove "during" overlaps (B); Merge gaps less than 1 months (C).

checking "Allow Overlap of Matched Sequences" shouldn't matter in this case because one patient's hospitalization periods are not supposed to overlap. However, comparing the number of replacements with and without this option may reveal errors in the logging of admissions and discharges.

Furthermore, another interesting question about the dataset is "what happens before and after the first time a patient was prescribed Drug A during their hospitalizations?" The question about what happended before and after a specific event is well solved by the "alignment" functionality [1], which shifts the timeline of each record and brings the focal event in alignment. However, EventFlow doesn't not support alignment by a special event in context of event sequences. The TSR component can bridge this analytic gap by replacing the special event in context with a new event, which then can be aligned. The required steps are as follows.

First, in the search panel, users add a query event "Drug A" and a previous obtained interval event "Hospitalization", letting them overlap with each other. Second, in the replace panel, users add a new point event called "1st-DrugA-in-Hospital" colored with orange and replace it in the replacement slot for event "Drug A". Then users specify to replace the first matched event sequence by checking the "Replace Nth Match" radio box and setting the number to 1 (Fig. 7 Right), After issuing this replacement operation, all the first occurrence of the targeted event sequence are replaced with a single new point event. Finally, users align by the new event "1st-DrugA-in-Hospital" to see what happens before and after it.

Case 2: Event Simplification

In many cases, the original dataset contains an large number of events, resulting in a cluttered visual representation and preventing users from obtaining insights. This is especially true when using EventFlow's aggregation visualization techniques [17]. For example, the clinical drug trial dataset contains all the medication taken by the patients, with every single use of medication represented by a interval event. Since the medication is usually taken regularly by the patients, there can be a large amount of events logged for each patient. In general, the pharmacologists are more interested in the time period when the patient is under the effect of the medication, rather than the exact details about when each medication is taken. This requires the functionality to simplify the original data in a user-defined way, which can be done by TSR.

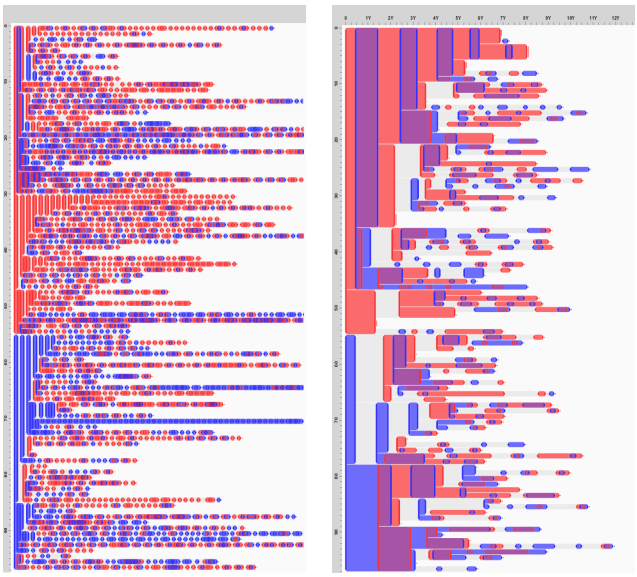

Fig. 9: The original visualization (left); The visualization after simplification that reveals treatment episodes (right).

TSR can help simplify the visualization of medication uses by replacing the uses details with a representation of the period when the patient was taking the medication regularly. Specifically, users may treat the act of taking drugs regularly with time gap less than 1 month as a treatment episode (Note that the size of the gap will vary depending on the drug being studied). To simplify the visualization and reveal these treatment episodes, users can remove the time gaps of less than 1 months in medication interval events and also remove overlaps which may occur when patients refill their prescription before they use all the pills they have at home. There are two kinds of possible interval event overlaps in the data: overlap and "during" overlap, as defined by Allen [18]. For each of these two overlaps, we need to do a replacement

shown in the upper part of Fig. 8 to remove it. After removing all the overlaps, users replace all the medication interval events with a time gap less than 1 month with a continuous medication of the same kind, as shown in the lower part of Fig. 8. We use a synthetic dataset containing medication history of 100 patients with two interval events (Drug A and Drug B) to demonstrate the TSR's capability of simplying event sequences. The aggregation visualization before and after the data simplification is shown in Fig. 9.

V. Evaluation

A preliminary usability study with 9 participants (5 males, 4 females, mean age 26.7, std 2.5) was done after the first implementation of TSR and it helped us identify several issues which were addressed in the current prototype described in this paper. For example, the search panel and replace panel in the original design were separated without any visual linkage, users found it hard to see the correspondance between the search sequence and the replacement sequence, which lead to our design of replacement slots and the adjacency of the two panels. Another design choice suggested by the first study is to have the first interval replacement event automatically span across the whole search sequence. To verify that these improvements were effective we conducted a second round of usability study, which we describe in more detail below.

The second usability study involved 10 participants (7 females, 3 males, mean age 26.9, std 2.38). These participants frequently use computers and have previously used "Search and Replace" features in text editors. The participants are all CS graduate students. The usability study was composed of four main phases: introduction, training, usability testing, and questionnaire. On average the usability study took about 40 minutes to complete. During the entire test, participants were encouraged to think-out-loud, ask questions, and comment on features or concepts, especially those they found confusing. The purpose of the usability study was to see if TSR is easy to learn and use, and elicit users' suggestion and comments. Therefore, we didn't require the participants to answer questions exactly, though most of the participants successfully answered the questions using TSR.

Introduction: We first introduced basic temporal concepts about point and interval events and how the visualization is constructed in EventFlow. Then we described in detail how TSR works. The introduction phase took 5 minutes.

Training: During the training session, we spent 10 minutes guiding participants through four simple tasks covering the following aspects: 1) the visual query language; 2) replacement of simple event sequences; 3) replacement of event sequences under repetition constraint; 4) replacement of event sequences under permutation constraint. A synthetic training dataset with no specific meaning was used in this phase.

Testing: After the training phase, we loaded a new synthetic dataset that describes the academic history of 40 professors. The dataset contains 7 unique point events (the professor's Bachelor, Master and Ph.D. graduation dates, journal and conference publication dates). The dataset also contains 3 unique interval events (the time period the professor are titled an assistant professor, as an associate professor, and as a professor). We choose this synthetic dataset because its meaning is understandable for all the participants and also complex enough to contain hidden knowledge. We asked the participants the following four questions which are specially designed to target various capabilities of TSR.

1) Who had spent the longest time after his/her Bachelor graduation and before Ph.D. graduation?
2) Who had his/her third conference paper publication before Ph.D. graduation?
3) Who has the longest journal publication history (time from the first to the last journal publication)?
4) Who has the most pairs of journal and conference publications?

This phase lasted on average 15 minutes. Our involvement was limited to answering clarification questions and minimum guidance if participants started to struggle. We documented the questions, comments, and problems.

Questionnaire: After completing the usability tasks, participants filled out a survey and answered follow up interview questions about the understandability, usability and usefulness of TSR using a 1-7 point Likert scale where 7 is strongly agreed and 1 is strongly disagreed.

A. Result and Discussion

The first question requires the most basic usage of TSR where an event sequence consisting of two point events is replaced with a single interval event. Nine out of ten participants correctly used TSR to answer the question very quickly. One participant decided to directly scan the raw data and answered the question in slow and unconfident manner.

The second question required participants to replace a specific occurrence of an event, by checking the "Replace Nth Match" radio box and setting the number of occurrences. We expected participants to answer the question by isolating the third occurrence of event "Journal Publication" and then determining whether it occured before event "Ph.D. Graduation". Three participants had difficulty at first, but after a few minutes, they found the right solution by themselves. To our surprise, four participants came up with a different approach that uses the repetition constraint. They specified a search event sequence of event "Journal Publication" repeated for 3 times, followed by event "Ph.D. Graduation", and replaced it with a new interval event across the whole sequence. Thereby, they only needed to see which records have that new event. In fact, using this approach doesn't require doing replacement of the sequence, however, this observation shows that users can quickly learn the concept of repetition constraint and correctly use it in analytic tasks.

The third question was targeted to the repetition constraint and the desired solution was to search for the "Journal Publication" event repeated for more than once (greedily) and replacing the whole sequence with a new interval event, then measure its length. Similarly to the second question, 5 out of 10 participants did not figure out the solution immediately.

Most of the difficultly came from users' unfamiliarity with the detail of repetition constraint feature. However, after some clarification, they successfully figured out how to use it and answered the question quickly. Also, we believe the addition of visual hints (e.g., a lassoing animation) can alleviate new users' trouble in accessing the search constraint functionalities.

The last question required the participants' good understanding of the permutation constraint that the order of the events under this constraint doesn't matter. Specifically, the question could be answered by replacing the event sequence consisting of two events of "Journal Publication" and "Conference Publication" under permutation constraint with a single event and then count the new events. Most participants found this question hard and they needed our hints. However, all of the participants could successfully specify the replacement operation and answer the question very quickly after they knew what constraint to use.

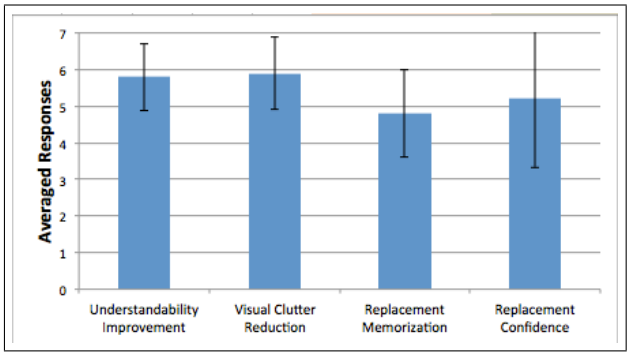

Fig. 10: The ratings about the improvement of the understandability and visual clarity of the data, and how memorable the replacement is and how confident the participants are about it.

The questionnaire result in Fig. 10 shows that the participants think TSR is helpful at improving the understandability of the temporal event data (5.8±0.92). Similarly, participants strongly agree that TSR can reduce the visual clutter of the data (5.9±0.99). The result also shows that participants are generally not able to remember exactly what replacements they did (4.8±1.2), which emphasizes the importance of providing a replacement history list for users' reference. The participants' confidence about what the replacements they specified actually did to the data are not very strong(5.2±1.87). This is mostly because most participants are new to temporal event data and TSR is a new tool of its own kind with a lot conceptual difference to the traditional search and replace feature in text editing softwares. However, this suggests potential improvement that allows users to replace one-by-one until they are confident to replace all.

The result in Fig. 11 suggests that the understandability of both the concept and interface of the two search constraints, repetition and permutation, is high. The understandability of the concept of permutation constraint (6.2±1.03) is slightly higher than that of repetition constraint (5.8±1.03). The

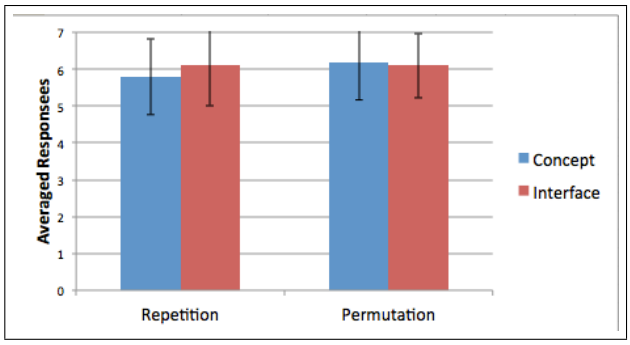

Fig. 11: The ratings about the understandability of the concept and interface of the repetition and permutation constraint.

understandability of the interface of permutation constraint (6.1±0.88) and repetition constraint(6.1±1.1) are basically the same with permutation constraint having a slightly narrower spread. Although the small size of participants makes the slight differences between these scores not interesting, the high rating on both the repetition and permutation constraints shows the feasibility of translating regular expression based concepts into temporal event sequence matching.

The most confusing part participants reported is about allowing the overlaps between matched event sequences. Seven out of 10 participants ignored the option "Allow Overlap of Matched Sequences" and they assume the "Replace All" option will literally replace all the occurrence of matched sequences in any circumstances. This implies that this option should be checked by default. Similarly, the "Allow Overlap of Repeated Sequences" option in repetition constraint configuration should also be checked by default to reduce user confusion. Two participants mentioned the term "permutation" is not straightforward and suggested using term "No Order" or "Any Order". The term "Greedily" is also unfamiliar especially to non-computer-science majors, and the alternative can be "Match as Many as Possible". One participant suggested to provide recommended colors distinct from the existing event colors when adding new events.

VI. Conclusion and Future Work

Temporal event sequence datasets are difficult to understand and analyze often because of the way the data is recorded is not perfect in terms of including concise and meaningful representations of complex events. In this paper, we introduced a novel temporal search and replace tool (TSR) implemented in the EventFlow visual analytic environment to help users manipulate the event data by searching for event sequences and replacing them with user-defined ones. Also, to enhance the replacement capability, we introduced two useful search constraints of repetition and permutation, allowing users to search and replace complex event sequences. Examples use cases were discussed to demonstrate the usefulness of TSR. The results of the usability study suggests that TSR was generally easy to use and participants agreed that it was helpful at both reducing the visual clutter and improving the understanding of temporal event data.

Although our current design of TSR is shown to be useful in many cases, there is still room for improvements. We have introduced the regular expression's concepts of quantifier and alternation into temporal event search and replace, we will further design and test the other concepts such as boundaries and anchors to complete the TSR's capability. TSR only allows replacement events to occur at the exact same time with the replacement slots. We will provide more fexibility by allowing temporal relationship between replacement events and the replacement slots. Currently, users need to figure out the replacement rules by themselves. However, a lot of the analyses are exploratory and users do not know what questions they want to ask. We would also like to automatically suggest interesting event sequences(e.g., the most frequent sequence).

ACKNOWLEDGMENT

We appreciate the support from the Oracle Corporation and feedbacks provided by Juan Morales del Olmo, Seth Powsner, Sheila Weiss, Gigi Lipori, and the usability study participants.

REFERENCES

[1] 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 26th annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2008, pp. 457–466.

[2] M. Monroe, K. Wongsuphasawat, C. Plaisant, B. Shneiderman, and S. G. Jeff Millstein, "Exploring point and interval event patterns: Display methods and interactive visual query," in *HCIL Tech Report, University of Maryland - College Park*, 2012. [Online]. Available: http://www.cs.umd.edu/localphp/hcil/tech-reports-search.php?number=2012-06

[3] 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*, vol. 24, no. 2, pp. 55–68, 2012.

[4] R. Snodgrass, "The temporal query language tquel," *ACM Transactions on Database Systems*, vol. 12, no. 2, pp. 247–298, Jun 1987.

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

[6] J. H. H. Andrew R Post, Ana N Sovarel, "Abstraction-based temporal data retrieval for a clinical data repository." in *AMIA Annual Symposium Proceedings 2007*, 2007, vol. 603.

[7] A. R. Post and J. H. Harrison, "Protempa: A method for specifying and identifying temporal sequences in retrospective data for patient selection," in *Journal of the American Medical Informatics Association*, vol. 14, no. 5, 2007, pp. 674–683.

[8] J. Jin and P. Szekely, "Querymarvel: A visual query language for temporal patterns using comic strips," in *Proceedings of the 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, ser. VLHCC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 207–214.

[9] ——, "Interactive querying of temporal data using a comic strip metaphor," in *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, Oct 2010, pp. 163–170.

[10] M. Monroe, R. Lan, M. del Olmo J., 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*. New York, NY, USA: ACM, 2013.

[11] T. Wang, A. Deshpande, and B. Shneiderman, "A temporal pattern search algorithm for personal history event visualization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 5, pp. 799–812, May 2012.

[12] D. Kurlander and S. Feiner, "Interactive constraint-based search and replace," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1992, pp. 609–618.

[13] F. Haag, S. Lohmann, and T. Ertl, "Simplifying filter/flow graphs by subgraph substitution," in *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*, Oct 2012, pp. 145 –148.

[14] C. Dunne and B. Shneiderman, "Motif simplification: improving network visualization readability with fan, connector, and clique glyphs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2013.

[15] N. H. Mustafa, S. Krishnan, G. Varadhan, and S. Venkatasubramanian, "Dynamic simplification and visualization of large maps," *International Journal of Geographical Information Science*, vol. 20, no. 3, pp. 273–302, 2006.

[16] D. Goren-Bar, Y. Shahar, M. Galperin-Aizenberg, D. Boaz, and G. Tahan, "Knave ii: the definition and implementation of an intelligent tool for visualization and exploration of time-oriented clinical data," in *Proceedings of the working conference on Advanced visual interfaces*. New York, NY, USA: ACM, 2004, pp. 171–174.

[17] 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*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1747–1756.

[18] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, Nov 1983.