

Exploring Point and Interval Event Patterns: Display Methods and Interactive Visual Query

Megan Monroe, Krist Wongsuphasawat, Catherine Plaisant, Ben Shneiderman, Jeff Millstein, Sigfried Gold

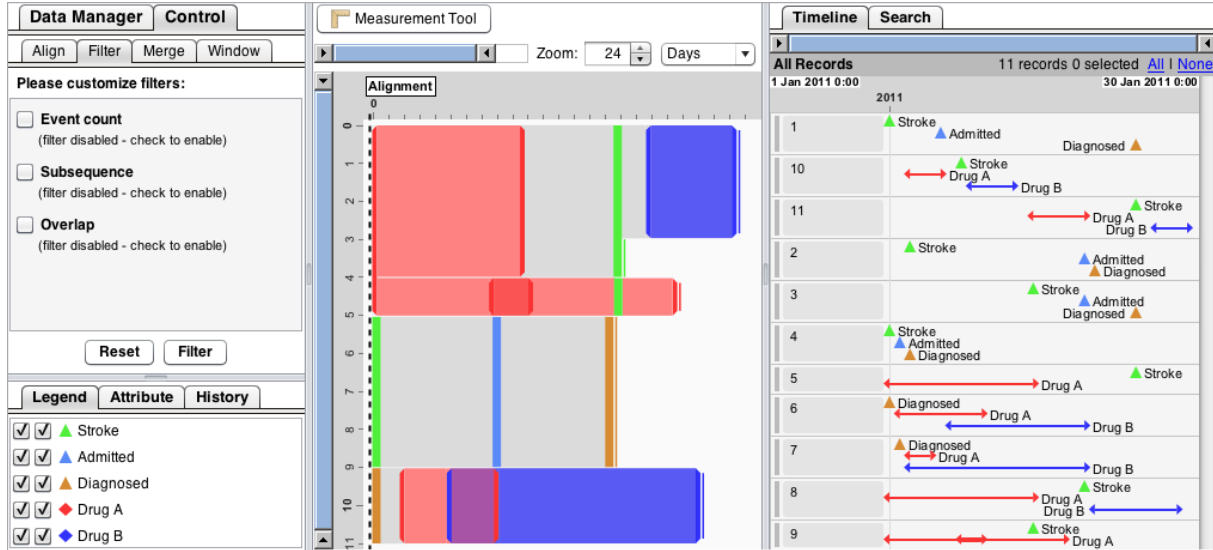


Fig. 1: The EventFlow display including the control panel and legend (left), the aggregated record display (middle), and the individual record display (right).

Abstract— Our work on searching point-based event sequences with the Align, Rank, Filter, and Summary concepts produced a powerful tool (Lifelines2) that is being applied in an increasing set of medical and other applications. The LifeFlow aggregation tool expanded the capabilities of Lifelines2 to show common patterns of events on a single-screen display, resulting in the ability to summarize millions of individual patient records. However, users found that point-based event sequences limited their capacity to solve problems that had inherently interval attributes, for example, the 3-month interval during which patients took a medication. This paper reports on our development of EventFlow, an application that integrates interval-based events into the original LifeFlow mechanisms. Interval events represent a fundamental increase in complexity at every level of the application, from the input and data structure to the eventual questions that a user might ask of the data. Our goal was to accomplish this integration in a way that appeared to users as a simple and intuitive extension of the original LifeFlow tool. In this paper, we present novel solutions for displaying interval events, simplifying their visual impact, and incorporating them into meaningful queries.

Index Terms—EventFlow, temporal event sequence, temporal event querying.

1 INTRODUCTION

The task of finding interesting and important event sequences has been explored across many different fields and applications. Our work, while relevant across many domains, has focused primarily on medical data, and the exploration of patient records. These records are comprised of categorical, event-based data. For example, a given patient record might consist of the following sequential events:

Stroke → Admitted into Hospital → Diagnosed

To assist in the exploration of these records, we initially developed a powerful visualization tool, Lifelines2 [21], which allows these

- Megan Monroe, Krist Wongsuphasawat, Catherine Plaisant, and Ben Shneiderman are with the Department of Computer Science & Human-Computer Interaction Lab, University of Maryland.
- Jeff Millstein is with Oracle.
- Sigfried Gold is with Social & Scientific Systems.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

Tool	Event Types	Records	Display
LifeLines [15]	points, intervals	one	indiv
LifeLines2 [21]	points	many	indiv
Similan [23]	points	many	indiv
LifeFlow [22]	points	many	indiv, aggr
EventFlow	points, intervals	many	indiv, aggr

Table 1: A summary of our evolving work with point and interval event types, one and many patient records, and individual and aggregated displays.

records to be searched for meaningful sequences using the Align, Rank, Filter, and Summary concepts. Medical researchers and practitioners used this tool to support hypothesis generation, and find cause-and-effect relationships in a population.

This work evolved further into the LifeFlow aggregation tool [22], which consolidates records with similar event patterns into a summarizing display. LifeFlow allows for an entire set of patient records to be represented, manipulated, and evaluated on a single screen. Both LifeLines2 and LifeFlow were designed to support point-based events. That is, discrete events that occur at a single point in time. A complete summary of our evolving work with event-based temporal records can be found in Table 1.

However, point-based events do not adequately cover the complete range of both medical and real world phenomena. Our current partners, researchers at the US Army Pharmacovigilance Center (PVC),

are primarily responsible for conducting drug related studies involving prescription administration and medication interaction. Their data, and the questions they need to ask of it, are inherently interval based. For example, they might need to know when two medication are being taken at the same time.

Unlike point events, which subscribe only to the sequential notions of “before” and “after,” interval events introduce the concept of “during.” Take, for example, the following two event sequences:

Stroke → Admitted into Hospital → Diagnosed

Started Medication A → Stroke → Ended Medication A

While both of these sequences contain three points in time, the second sequence conveys two additional pieces of information. First, the start of the medication necessarily implies, and is fundamentally linked to the end of that medication. Second, a stroke (a point event) occurs *during* the time that the patient was taking that medication (an interval event). This additional information changes the types of questions that the user can ask of the data. It changes the way that the data must be visually displayed in order to remain intuitive, and it changes the requirements on the underlying structure in which the data is stored.

This paper describes EventFlow, which implements novel display, control, and query mechanisms in order to accommodate interval events. Our contributions are as follows:

- A visual representation of interval events for both individual and aggregated displays.
- A set of controls for simplifying and exploring records containing interval events.
- A simple visual query language for professionals in non-technical fields that allows users to specify the presence *or* absence of both point and interval events.

This paper is organized as follows: we present related work in Section 2, followed by an overview of the EventFlow display in Section 3. Our methods of data and display simplification are described in Section 4. The scope and implementation of the query interface are presented in Sections 5 and 6. We discuss our preliminary results in Section 7, and conclude in Section 8.

2 RELATED WORK

The development of EventFlow was rooted in three major spheres of prior work: temporal logic, temporal querying, and temporal visualization.

Temporal Logic

EventFlow draws first on the methods to logically represent interval events, most notably the work of Allen et al. [2]. Allen identified 13 unique relationships between two intervals and developed a logic capable of representing each of them. This work has been substantially reinforced by work in the data mining community [17, 24, 14].

While Allen’s 13 relationships encompass the full spectrum of possible interval relationships, our work addresses only a subset of them. The time-based datasets used by EventFlow, specifically from medical data sources, are not precise enough to make all 13 of Allen’s distinctions valuable. For example, Allen defines many relationships in which event points are expected to occur concurrently, such as when the end of one interval exactly meets the beginning of another. EventFlow, however, would consider only that the two intervals occur disjointly, and that one interval occurs before the other. By narrowing this focus, we are able to present a more complete set of querying controls without compromising the simplicity of the interface.

Furthermore, Allen’s work does not address the relationships created by point *and* interval events, or the absence of interval events. The absence of an interval is a complex occurrence that can assume point-like characteristics. A discussion of absence interval events, as

well as the full spectrum of event relationships supported by EventFlow, is described in Section 5.

Temporal Querying

Temporal querying techniques have been substantially explored in the database community. Snodgrass [18] and Jensen et al. [10] extended the well-known database querying language, SQL, to encompass temporal relationships. While this work encompasses a wide range of point and interval relationships, including constructs for specifying the absence of an event [6], it focuses on command-based querying. As we will demonstrate in the following sections, command-based querying becomes exceedingly complex for users to quickly learn and execute.

We also surveyed efforts in the visualization domain, where there have been considerable efforts made towards constructing a more usable, visual query language [20, 7, 4, 3, 9]. Jin and Szekely use a comic strip metaphor to facilitate the process of querying [11]. Wongsuphasawat et al. developed Similan to query temporal event sequences for both exact matches and similar sequences [23]. EventFlow extends this work specifically by allowing users to specify the absence of an interval event.

Temporal Visualization

Finally, EventFlow draws on work in the field of visualization, and the varied attempts to visualize different kinds of temporal data. Aigner et al. took a comprehensive look at such attempts, covering the differences in both data types and visualization techniques [1]. Rind et al. narrows this focus considerably in a review of visualization for medical data, specifically electronic health records [16]. None of the implementations surveyed in either of these works, however, incorporate both point and interval events into an aggregated display.

In the following section, we discuss our approach to creating a meaningful representation of interval-based events that remains consistent with the original LifeFlow representation of the categorical, event-based data that comprises electronic health records.

3 DISPLAY METHODS

The EventFlow display consists of the individual-record lists that comprised the original LifeLines2 implementation, the aggregated record display, and the controls for manipulating them. The complete view of the EventFlow interface can be found in Figure 1. Point events are represented in EventFlow as colored triangles, both in the legend (bottom left of display) and within the individual records (right pane of display). For individual records, events of the same category are drawn in the same color across a unique horizontal axis. Thus, our original sequence of three point events is drawn as follows:

Stroke → Admitted into Hospital → Diagnosed



In the aggregated record display, sequential point events are displayed as a series of vertical, colored bars (middle pane of display). Thus, the aggregated display of the above sequence is drawn as follows:



The height of the bars is used to convey the number of records that consist of this identical sequence. For example, in Figure 1, four of the nine patient records (listed in the right pane) consist of this sequence. Accordingly, the sequence accounts for just under half of the vertical space in the aggregated display. The space between each pair of bars

is computed from the average lapse of time between those two events. A complete description of the original LifeFlow display can be found in [22].

To include intervals in the display, EventFlow needed to indicate not only the interval's beginning and end, but also that the event was still happening during the time in between. To do this, interval categories are represented in the EventFlow legend as a colored diamond (bottom left of display). In the individual record display, this diamond is split into a double-sided arrow to represent the duration of each interval event. Our original interval sequence then, is drawn as follows in the individual record display:

Started Medication A → Stroke → Ended Medication A



In the aggregated display of interval events, the interval start and end points are drawn as thin, beveled bars. The area between these bars is then filled with a less saturated version of the appropriate category color. This technique of representing intervals using filled bars has been widely used in the visualization field [19, 13, 12]. The aggregated interval sequence is drawn as follows:



The EventFlow display must also take into account the fact that multiple interval events can occur concurrently. In all of our surveyed works, overlapping intervals are displayed by tiling them vertically [5, 13, 8]. For individual records, this technique is used as a natural consequence of the LineLines2 display scheme. However, in the aggregated display, vertical delineation is already used to convey diverging event sequences.

As a result, EventFlow renders overlapping intervals by filling the overlap region with the combined color of the two overlapping categories. The colors selected for interval categories default to primary colors, resulting in intuitive overlap colors. For example, when a red interval intersects a blue interval, the resulting overlap is purple. When two intervals of the same category intersect, the color saturation in the overlapping region is increased. The overlapping display is demonstrated below:



Fig. 2: Overlapping of two interval events of different categories (left). Overlapping of two interval events of the same category (right).

This overlap display technique is effective for up to three interval categories, where an overlap between any pair of categories results in an intuitive, composite color, and an overlap of all three categories

produces a less attractive, but still unique color. Our motivation in allowing for this limitation was based primarily on feedback from users, particularly in the medical domain. Their research, while potentially involving many interval categories overall, is typically focused on at most three categories at any single point in time.

For example, the PVC researchers were investigating the interplay between one particular asthma medication and all other asthma medications. While their dataset consisted of far more than three interval categories, for their specific purposes, all but one of them could be lumped into a larger meta-category. Additionally, EventFlow allows for categories to be easily added and removed from the display, and for category colors to be quickly changed. Thus, the primary color scheme can be quickly shifted to whichever three categories are currently at the forefront of interest.

4 DISPLAY SIMPLIFICATION

Interval events necessarily result in an increase in the amount of information that must be displayed on the screen and understood by users. In addition to understanding the sequence of event points (“event points” referring to either a point event, or an endpoint of an interval event), users must also be aware of what is occurring between these event points, and their relation to each other. Furthermore, medical datasets frequently consist of many thousands of records, each with hundreds of events. Figure 3 depicts only a small sample of a real world dataset to illustrate the potential visual complexity.

To account for this, EventFlow is equipped with a number of mechanisms that allow users to reduce the amount of information on the screen, and focus in on the questions that are important to them. The goal of these controls is to serve as an information zoom, achieving an effect similar to the traditional spatial zoom by reducing the amount of visual noise.

Interval Opacity

In certain instances, the sequence of event points may be more important than the duration of the interval events. In these instances, the bright color of the interval background may serve as a distraction. To reduce this complexity, users are able to control the opacity of the interval area (Figure 3). This allows the display to be quickly reduced to that of a purely point-event display when necessary.



Fig. 3: Varying levels of opacity for an interval event.

Highlighting

As datasets become more complex, it becomes increasingly difficult to identify, and thus analyze single features, particularly the interplay between intervals. To account for this, we provide users with the ability to highlight areas of the aggregated display where events of different categories overlap. This can be an overlap between multiple interval categories (including intervals of same category), or a point event that occurs during a given interval category. Users can select the highlight color of their choice, and can strengthen the effect by reducing the overall interval opacity. A demonstration of the highlighting feature can be found in Figure 4.

Interval Event Merging

In research involving drugs, data is generally available in the form of dispensing dates and prescription durations, so we will know the date that a drug was dispensed (the start date) and can infer the end date based on intended length of the prescription. These dates then, are dictated substantially by the behavior of the patient. For example, it might be convenient for the patient to refill a prescription a few days

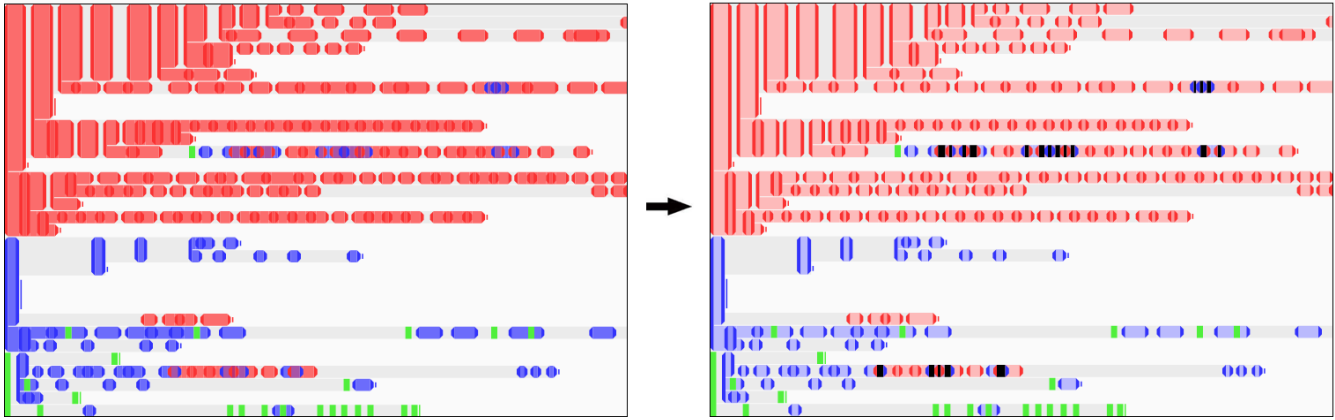


Fig. 4: To quickly identify overlaps between the red and blue intervals in a complex dataset (left), black highlighting and reduced interval opacity can be specified (right).

before the previous one runs out. In EventFlow, this can lead to a series of overlapping and disjoint intervals.

Medical research questions, however, are typically concerned with exposure, that is, the time during which the patient was actually taking a drug and exposed to its effects. Exposure can only be roughly inferred from a series of prescriptions, but we can generally assume that if a patient receives a series of thirty-day prescriptions, with each prescription starting a few days before or after the previous one ends, that this indicates a continuous, steady exposure. Short gaps and overlaps between successive prescriptions do not necessarily imply that the patient stopped taking the medication or took a double dose during those few days.

It is important to distinguish the above scenario, which results from an obvious and common behavior, from situations where a combined dose of a single medication is a critical and interesting phenomenon. For example, taking concurrent pain medication prescriptions may have serious health implications.

In order for EventFlow to better represent exposure, and not to clutter the display with clinically uninteresting information about the exact dates of prescription refills, EventFlow provides a simple interface that allows for multiple interval events of the same category to be merged into a single interval event. This can be done in two ways: eliminating gaps of a certain duration, and eliminating overlaps of a certain duration. This allows users to decide which interval interactions are important, and eliminate those that are not to produce a cleaner display. A demonstration of this feature can be found in Figure 5. Merging can be easily undone, or saved permanently as a new dataset.

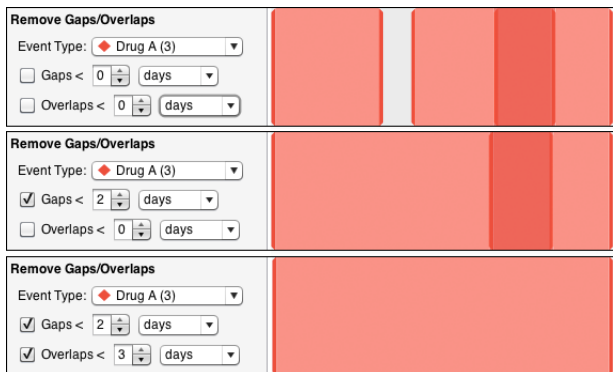


Fig. 5: A series of disjoint and overlapping intervals are merged into a single interval.

Temporal Range Limits

Like all temporal visualizations, EventFlow allows for spatial zooming into any region of the display. This feature, while effective for a wide range of tasks, only provides further insight into the aggregated

display, which is plotted according to the average lapse in time between events. This averaging can mask the temporal features of the individual records. To retain access to these individual features, temporal range limits allow users to specify temporal constraints on the individual records before they are processed into the aggregated display.

For example, researchers might be interested in the potential side effects of a medication. In EventFlow, it would be intuitive then, to align the aggregated display by the endpoint of this medication interval. Using the spatial zoom feature, the researcher could narrow the display to the period of time directly following this alignment point. However, because the aggregated event markers are placed according to the averaged time lapse, it is difficult to determine the appropriate zooming position with any precision. The display is likely to contain events that are not of interest, and exclude potentially important events.

Alternatively, the researcher could use a temporal range limit, which allows a range limit to be specified on the individual records based on either a time period or an event count. In this case, the researcher might specify a limit of two months such that only events within two months of the alignment point would be processed into the aggregated display. The result is a cleaner, more streamlined aggregated display. This feature is particularly useful for long event records, which are common when dealing with medical records. A demonstration of temporal range constraints can be found in Figures 6 and 7.

5 QUERY SCOPE

As discussed previously, the timestamps in many of the real world datasets that EventFlow utilizes are not precise enough to define relationships where event points are expected to occur simultaneously. This allows us to focus our querying efforts on only a subset of Allen's 13 interval relationships. However, from this subset, EventFlow extends Allen's work to account for point events, their relationship to interval events, and queries for the absence of a point or interval event.

In the section we describe the full set of event relationships supported by the EventFlow query interface. To demonstrate the added complexity of interval event relationships, we first contrast it with the relative simplicity of point event relationships. For the sake of further simplicity, we limit this analysis to the relationship between two unique events.

Querying for Presence Events

For our users, the absence of an event can be just as important as the presence of an event. For example, a medical researcher may want to identify the patients who did *not* have a heart attack after taking a certain medication. As such, EventFlow supports querying for both presence and absence events. We begin our discussion with the more straightforward of the two: querying for presence events.

Point/Point Event Querying

As discussed previously, the relationship between two point events is encompassed by the notions of before and after. For example, researchers might be interested in the following sequence of events:



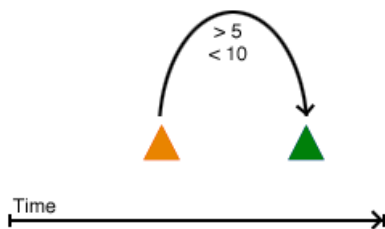
This relationship, expressed graphically above, can also be described in plain English.

(1) English: Show all records in which an Orange event is followed by a Green event.

Or in a standard querying language such as SQL: `SELECT RecordID FROM EventList AS List1 INNER JOIN EventList AS List2 ON List1.RecordID = List2.RecordID WHERE List1.EventType = Red AND List2.EventType = Blue AND List1.Time < List2.Time`

We can see already, even for the simplest query, that standard querying languages are too complex for most users (in our case physicians, epidemiologists, medical and pharmaceutical researchers) to quickly master and execute. Moving forward, we will limit our discussion to only graphical and plain English expressions. We will also assume throughout that the ordering of the events can be reversed to produce a conceptually identical scenario.

Given two point events, we can tighten the constraints on their temporal relationship by specifying either an amount of time that must elapse between the two events, an amount of time in which both events must occur, or both. For example, researchers might be interested in patients who reported back pain with 5 and 10 days of an injury. We will refer to these specifications as range constraints:

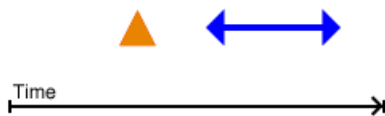


(2) English: Show all records in which an Orange event is followed by a Green event that occurs within 5 and 10 days.

Where “days” can be substituted with any time unit. We will assume from this point on that such range constraints can be added to any pair of event points.

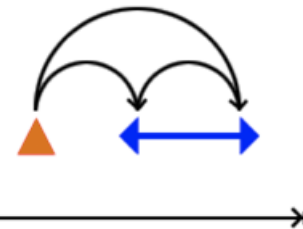
Point/Interval Event Querying

When one of the two events in question is an interval event, we can still employ the same notions of before and after:

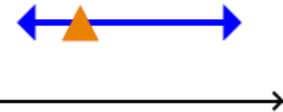


(3) English: Show all records in which an Orange event is followed by a Blue interval.

The application of range constraints remains the same in this setting, however the number of possible constraints increases due to the fact that an interval event has both a start and an end point. There are now three different pairs of event points between which range constraints can be specified:



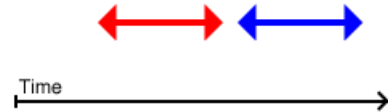
Interval events also introduce the concept of during. For example, researchers might be interested in patients who experienced a headache while taking a certain medication. We can now specify the following sequence:



(4) English: Show all records in which an Orange event occurs during a Blue interval.

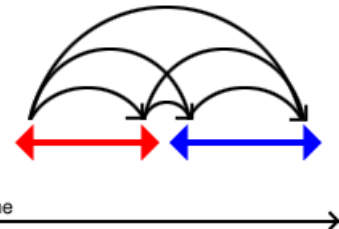
Interval/Interval Event Querying

Similar to the previous two conditions, two interval events can be queried according to their sequential ordering. For example, researchers might be interested in patients who were prescribed a new medication after their previous medication ran out:

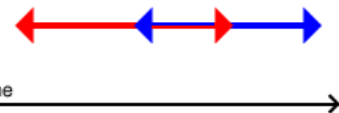


(5) English: Show all records in which a Red interval is followed by a Blue interval.

The number of possible range constraints is now increased to six:

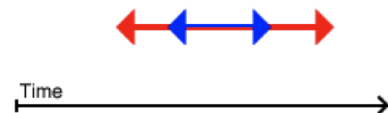


Two interval events also give rise to two different notions of during. For example, researchers might be interested in women who became pregnant before the completion of a hormone treatment:



(6) English: Show all records in which the beginning of a Blue interval overlaps a Red interval.

Alternatively, researchers might be interested in women who took a certain medication throughout their pregnancy:



(7) English: Show all records in which an entire Blue interval overlaps a Red interval.

Given our assumption that event points do not occur concurrently, and the requirement that all event points be specified in sequence,

EventFlow covers a comprehensive set of point and interval presence relationships. From here, we further extend the model to include absence events.

Querying for Absence Events

Querying for the absence of an event increases the complexity of the query because the meaning of an absence event differs between point and interval events. Graphically, we represent the duration of an absence in gray. The category of the absent event is represented with either a slashed triangle (for point events) or a slashed diamond (for interval events).

Point/Point Event Querying

It is critical to observe that the absence of a point event necessarily implies that the absence spans the period of time between the previous presence event (or the beginning of the record) and the subsequent presence event (or the end of the record). We refer to this as a “span absence”. For example, researchers might be interested in patients who did not have a heart attack after experiencing chest pains. By specifying this sequence:

Chest Pains → No Heart Attack

The implication is that a heart attack must not occur for the entire duration of time following the chest pains. That is, the absence of this event must span the time between the heart attack (the previous presence event), and the end of the patient’s record.



(8) English: Show all records in which an Orange event is followed by the absence of a Green event.

Given the less than precise nature of our datasets, it does not make sense to formulate a query that specifies the absence of a point event at a single point in time. Thus, it is only logical to think of absent point events as spanning a duration of time.

Point/Interval Event Querying

The absence of an interval event can take on two separate meanings, the first of which is identical to that of the point event absence. That is, a span absence: the users’ intent is for the absence to span the gap into which it has been inserted.



(9) English: Show all records in which an Orange event is followed by the continued absence of a Blue interval.

For, example, researchers might be interested in patients who were never prescribed a certain medication after a heart attack. However, unlike the absence of a point event, it is also meaningful to construct a query for the absence of an interval event at a single point in time. We refer to this as a “point absence”. In this case, researchers might be interested in patients who stopped taking their medication at any point after having a heart attack.

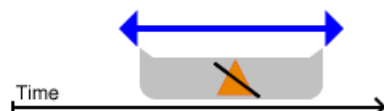


(10) English: “Show all records in which an Orange event is followed by an absence of a Blue interval.

The inclusion of an interval event also gives rise to two additional absence scenarios:



(11) English: Show all records in which an Orange event occurs, but not during a Blue interval.



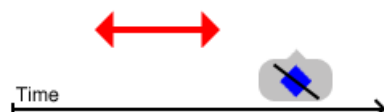
(12) English: Show all records in which an Orange event does not occur during a Blue interval.

Interval/Interval Event Querying

The dual meaning of an absent interval event can be applied in two different scenarios when dealing with two interval events. The first of these addresses the familiar notions of before and after that have been extensively discussed:

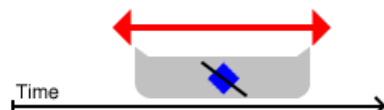


(13) English: Show all records in which a Red interval is followed by the continued absence of a Blue interval.



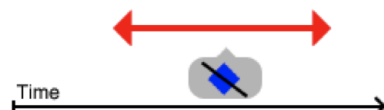
(14) English: Show all records in which a Red interval is followed by an absence of a Blue interval.

We can also apply this dual meaning to the notion of during:



(15) English: Show all records in which an entire Red interval does not overlap a Blue interval.

In this case, researchers might be interested in women were not prescribed a certain medication during their pregnancy. Alternatively, they might be interested in women who stopped taking their vitamins at some point during their pregnancy.



(16) English: Show all records in which a portion of a Red interval does not overlap a Blue interval.

Unlike presence queries, where we can judge the comprehensiveness of EventFlow’s query scope based on previous work in temporal logic, there is not an analogous benchmark by which to compare the scope of EventFlow’s absence queries. The current implementation addresses the needs expressed by users, and an overall goal of maintaining a simple and intuitive interface. We leave a completeness analysis of the absence query scope to future work.

6 QUERY IMPLEMENTATION

EventFlow provides two means of querying the event records: basic and advanced. The basic query interface is designed such that users can quickly access the fundamental features of each record. Advanced querying gives users access to the complete range of event relationships discussed in the previous section. In both cases, queries are formulated using the same visual elements that are used in the display of individual records.

Basic Querying

The original implementation of LifeLines2 provided a simple query module for identifying records with a specific subsequence of events. Users specified this sequence using a set of drop-down menus. While this was an extremely useful feature for identifying patterns of point events, it failed to capture some of most basic interval relationships. For example, consider users who would like to select all records in which the patient has a stroke while they are taking a medication. They might begin by specifying the following familiar subsequence:

Started Medication A → Stroke → Ended Medication A

However, this specification will also select records in which a patient has a stroke *between* two intervals of this medication. The user may then adjust their query to the following:

Started Med. A → No End Med. A → Stroke → Ended Med. A

This specification, as intended, will not select records in which a patient has a stroke between two intervals of the medication. However, it will also not select records in which a stroke occurs after an overlap of two Medication A intervals, even though the stroke indeed occurs while the patient is taking the medication. Below is the original Subsequence query module, in the context of these two situations:

Subsequence Query Module:



Fig. 8: Two examples of an event sequence (right), and the corresponding attempt to select records in which a stroke occurs during a certain medication (left). Based on the sequence specified by the drop-down boxes, the Subsequence query module produces both false positives (top) and false negatives (bottom).

Resulting from the shortcomings of the subsequence filter, we developed a second querying module that selects records containing a specific overlapping of events. Both of these modules (in their unexpanded form) can be found in the control panel of the EventFlow display (Figure 1). Using the same drop down controls, users can specify a series of events that must occur concurrently. In the case of absence interval events, this query module selects records based on the less restrictive point absence model. Returning to our previous query, records in which a stroke occurs during Medication A, the specification of one query using the Overlap query module becomes effective in both of our previous, failed cases (Figure 9).

Between the Subsequence query module and the Overlap query module, users have access to both sequential event relationships and overlap relationships limited to point absences. To access these relationships together, or to specify span absences and range

constraints, users must default to the advanced querying interface.

Overlap Query Module:



Fig. 9: Here we see a single overlap query (left) being applied to the same two event sequences (right). The Overlap query module successfully rejects the top sequence and accepts the bottom sequence.

Advanced Querying

The advanced query interface is an update to the Similan interface by Wongsuphasawat [23] to include interval events. It allows users to add events to a blank record template, creating a graphical query that is formatted similar to the display of the individual event records. The benefit of this visual language is that users can model query elements based on the features from these individual records.

Query elements are added by clicking the timeline. Users can then choose whether to specify a point or interval event, whether the event must be present or absent, and in the case of absence interval events, whether it should be a point absence or a span absence. Users can also specify range constraints based on the preceding event. The query interface and the form for specifying a query element can be found in Figures 10 and 11.

The form is titled 'Add Event' and features a green plus icon. It includes a dropdown for 'Event Type' set to 'Stroke (1)'. Below this are radio buttons for 'occur' (selected) and 'not occur'. A section titled 'Does this event have specific time information?' has radio buttons for 'No, just occur in order is enough.' and 'Yes,'. Under 'Yes,' there are checkboxes for 'I want it to occur at a specific time.' and 'I want to set time gap after previous event.' (checked). A text input field contains 'gap <= 1 month' and a button with three dots. At the bottom are 'Add event' and 'Cancel' buttons.

Fig. 10: Form for specifying a new query element.

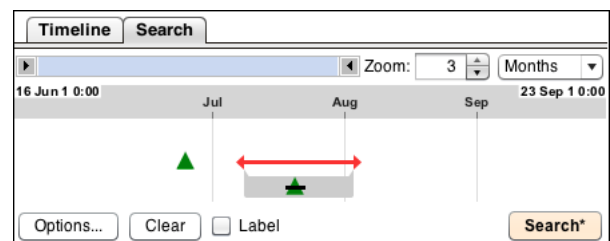


Fig. 11: Query for all records in which the patient has a stroke, followed by a prescription of Medication A during which the patient does not have another stroke.

Once the user has specified an event pattern of interest in the query interface, they are presented with a list of the individual records that

match their query, as well as the option to select those records in the aggregated display. As a result, users have two means of quickly reviewing their query results and modifying their query as needed.

7 EVALUATION

To evaluate the strengths of the updated EventFlow display and controls, we are conducting both individual and group user sessions at the PVC. In the first 2-hour group session, a team of seven PVC researchers (consisting primarily of statisticians and epidemiologists) were introduced to the EventFlow display and guided through its various features using synthetic datasets. We found that, after a brief introduction of both the individual and the aggregated interval display, the PVC researchers were able to quickly grasp the concept and visually process these elements. Without prompting, they began pointing out features and anomalies in our synthetic datasets and relating them back to known features of their own data.

One of the challenges that the PVC researchers face, before they can begin any analysis, is extracting the appropriate subset of patient and event information from the 14 million Electronic Health Records in their database. For example, one on-going study focused on a specific class of asthma medications. To isolate events of this class, a list of all asthma prescriptions (which are listed by medication name) had to be cross listed against a dictionary of medication names and their corresponding classes. EventFlow was used to explore a sample of the resulting dataset during three 1-hour sessions that were conducted with one of the PVC's epidemiologists. In these sessions, a member of our team controlled and manipulated the EventFlow display according to the epidemiologist's evolving questions and interests.

Immediately, the epidemiologist was able notice both unexpected event patterns and potential errors in the data sample. Her initial observation was that a certain asthma medication was being prescribed at an unusual cadence. Her expectation was to see a series of condensed, nearly continuous prescriptions. However, the EventFlow display clearly depicted short, month-long prescriptions occurring approximately once a year. The conclusion drawn from this display was that the sample of records was biased towards patients with seasonal allergies.

In subsequent sessions, the epidemiologist was also interested in the prescribing practices that typically lead up to a prescription for a stronger asthma medication. To assist in this query, EventFlow was used to align the data by the beginning of the strong prescription interval, and then temporal range constraints were applied in order to narrow the focus around this alignment point. A demonstration of this exploration can be found in Figures 6 and 7. The epidemiologist quickly noticed a medication that she had expected to see as a frequent precursor to this strong prescription, but was barely appearing at all. This observation led to a closer look at the dictionary of drug classes, and uncovered inconsistencies in how these classes were being assigned.

Throughout our sessions, the epidemiologist was able to quickly discover new and surprising patterns in data that she had been analyzing for months using standard statistical software. One of our individual sessions quickly turned into an impromptu group session when she began calling in other members of the team. She explained that they had never been able to "see" their data like this before. While this enthusiasm is encouraging, particularly from such experienced researchers, we recognize that much more careful testing and long-term assessments are needed to understand when EventFlow is helpful compared to other tools, and what additional features are needed.

8 CONCLUSIONS AND FUTURE WORK

Based on feedback from users of the original LifeFlow implementation, EventFlow offers a fundamental system redesign, ranging from the underlying data structure to the user interface, in order to handle interval as well as point events. This redesign includes a novel approach for integrating interval events into the individual and aggregated record display. We also presented various means of simplifying both the display and the underlying data records in order to maintain

the most visually streamlined set of information for users to process and analyze.

Additionally, the inclusion of interval events produced a dramatic increase in the number of possible relationships between two events. This increased complexity resulted in new mechanisms for both basic and advanced querying, which employ a visual query language that is intuitive even for non-technical professionals. Queries are constructed using the same elements that comprise the display of the individual records, allowing users to use these records as a template when constructing queries.

In our continuing efforts to understand temporal event data and adapt the EventFlow tool, we plan next on extending the capabilities of the advanced querying tool. These extensions include better integration of flexibility into the interval query element specification. Currently, users must specify the sequence of both end points for every interval query element. However, users might be interested in the overlap between two medications, but may not necessarily care which medication begins first. This will give rise to queries such as: Show all women who were taking a certain medication *at some point* during their pregnancy.

We are also investigating the scalability of EventFlow, and means of improving both storage and speed to accommodate increasingly larger datasets. Our goal is to support datasets with both large numbers of records, large numbers of events per record, and large numbers of event categories. As we continue our collaboration with the PVC, our hope is that their datasets and research questions will guide the development of the advanced query module, as well as scalability.

ACKNOWLEDGMENTS

We appreciate the partial support of the Oracle Corporation and NIH-National Cancer Institute grant RC1-CA147489, Interactive Exploration of Temporal Patterns in Electronic Health Records. We appreciate the collaboration of researchers at the US Army Pharmacovigilance Center and the Univ. of Maryland Human-Computer Interaction Lab.

REFERENCES

- [1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.
- [2] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. In *Journal of Logic and Computation*, volume 4, pages 531–579, 1994.
- [3] L. Chittaro and C. Combi. Visualizing queries on databases of temporal histories: new metaphors and their evaluation. In *Data & Knowledge Engineering - Special issue: Temporal representation and reasoning*, volume 44, pages 239–264, 2003.
- [4] C. Combi and B. Oliboni. Visually defining and querying consistent multi-granular clinical temporal abstractions. In *Artificial intelligence in medicine*, volume 54, pages 75–101, 2012.
- [5] S. B. Cousins and M. G. Kahn. The visual display of temporal information. In *Artificial Intelligence in Medicine*, volume 3, pages 341–357, 1991.
- [6] J. Dunn, S. Davey, A. Descour, and R. T. Snodgrass. Sequenced subset operators: Definition and implementation. In *IEEE International Conference on Data Engineering (ICDE2002)*, pages 81–92. IEEE, 2002.
- [7] 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 *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 167–174. IEEE, 2006.
- [8] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 115–123. IEEE, 2000.
- [9] S. Hibino and E. A. Rundensteiner. User interface evaluation of a direct manipulation temporal visual query language. In *MULTIMEDIA '97 Proceedings of the fifth ACM international conference on Multimedia*, pages 99–107. ACM, 1997.
- [10] C. S. Jensen, J. Cliord, and S. K. G. et al. The tsq benchmark. In *Proceedings of the International Workshop on an Infrastructure for Temporal Databases*, pages QQ1–QQ28, 1993.
- [11] J. Jin and P. Szekely. Interactive querying of temporal data using a comic strip metaphor. In *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 163–170. IEEE, 2010.

- [12] N. W. Kim, S. K. Card, and J. Heer. Tracing genealogical data with timenets. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, pages 241–248. ACM, 2010.
- [13] S. J. Nordb. Information visualisation and the electronic health record. In *Masters Thesis: Norwegian University of Science and Technology*, 2006.
- [14] D. Patel, W. Hsu, and M. L. Lee. Mining relationships among interval-based events for classification. In *SIGMOD '08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 393–404. ACM, 2008.
- [15] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: Visualizing personal histories. In *ACM CHI '96 Conference Proc.*, pages 221–227. ACM, 1996.
- [16] A. Rind, T. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman. Interactive information visualization for exploring and querying electronic health records: A systematic review. Technical report, University of Maryland, College Park: Human Computer Interaction Lab (HCIL), 2010.
- [17] P. shan Kam and A. W.-C. Fu. Discovering temporal patterns for interval-based events. In *DaWaK 2000 Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, pages 317–326. Springer-Verlag, 2000.
- [18] R. Snodgrass. The temporal query language tquel. In *ACM Transactions on Database Systems (TODS)*, volume 12, pages 247–298. ACM, 1987.
- [19] E. Tufte. *The Visual Display of Quantitative Informatin*. Graphic Press, 1983.
- [20] K. Vrotsou, J. Johansson, and M. Cooper. Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. In *IEEE Transactions on Visualization and Computer Graphics*, volume 15, pages 945–952, 2000.
- [21] T. D. Wang. Interactive visualization techniques for searching temporal categorical data. In *Ph.D. Dissertation from the Department of Computer Science, University of Maryland*, 2010.
- [22] K. Wongsuphasawat, J. A. G. Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: Visualizing an overview of event sequences. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI'11)*, pages 1747–1756. ACM, 2011.
- [23] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, and B. Shneiderman. Querying event sequences by exact match or similarity search: Design and empirical evaluation. In *Interacting with Computers*, volume 24, page 5568, 2012.
- [24] S.-Y. Wu and Y.-L. Chen. Mining nonambiguous temporal patterns for interval-based events. In *IEEE Transactions on Knowledge and Data Engineering*, volume 19, pages 742–758. IEEE, 2007.

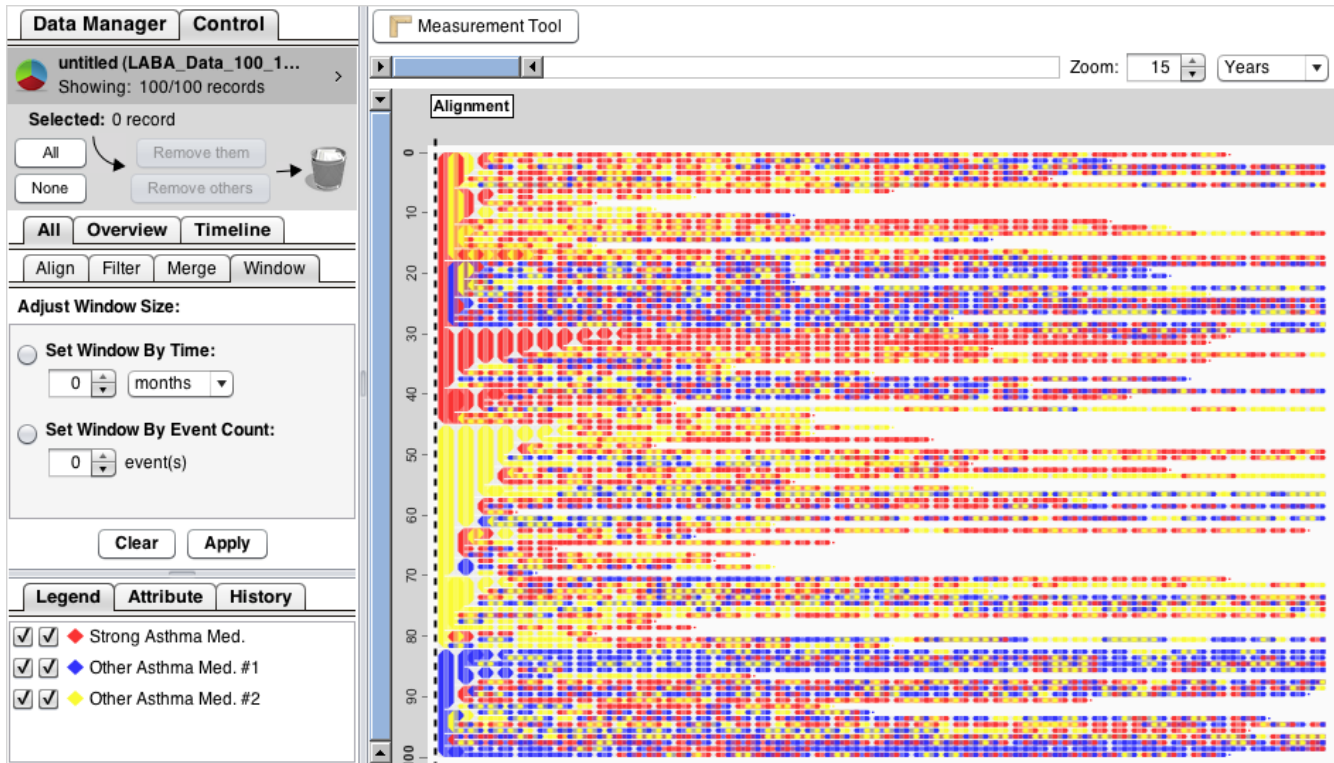


Fig. 6: **Before** - Data imported from Electronic Health Records can be complex to the point of distraction. Here, we see EventFlow's aggregated display for a real set of 100 patient records consisting of over 4000 events across a 6-year span of time (the individual record display is not pictured). In this particular case, researchers were interested in how the prescriptions for weaker asthma medications (in blue and yellow) lead up to the prescription of a strong asthma medication (in red).

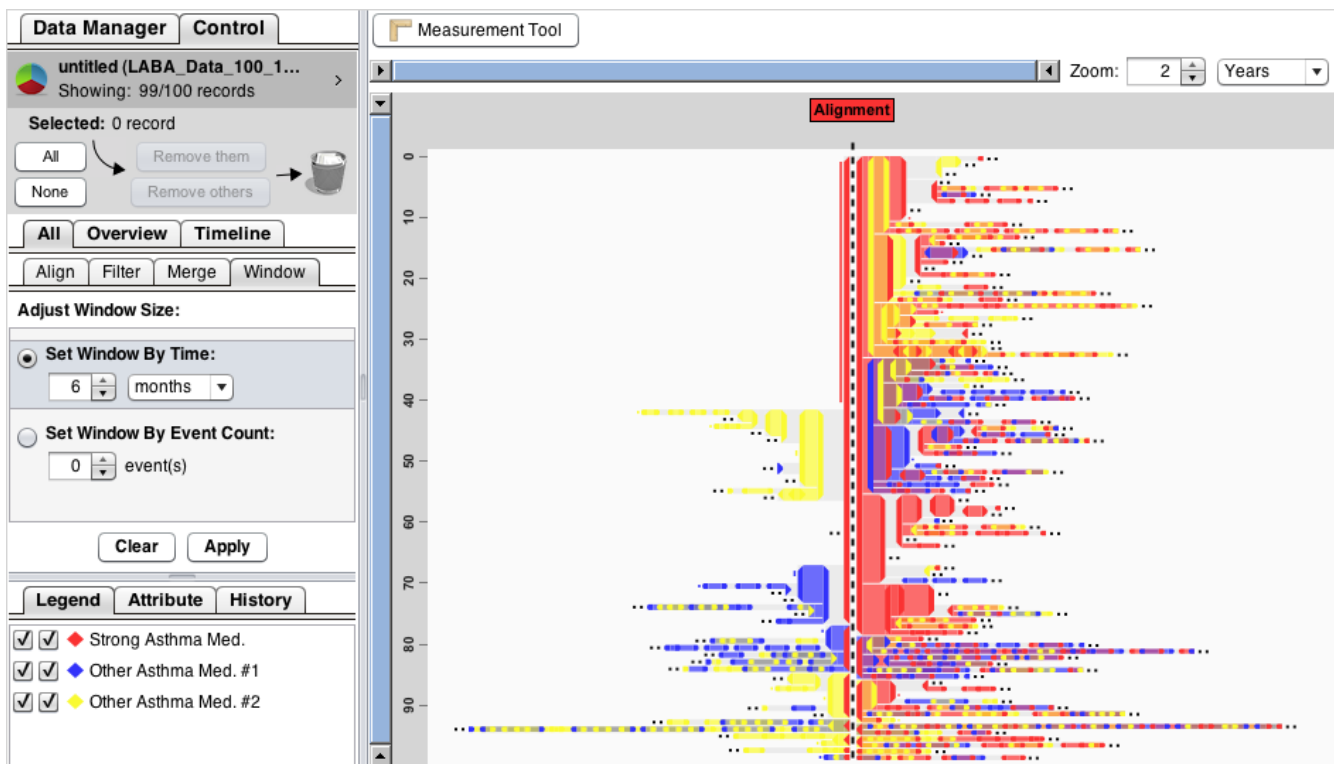


Fig. 7: **After** - The dataset is aligned by the starting point of the strong medication, and a temporal range constraint of 6 months is applied (the black dots indicate that the remainder of the sequence is not displayed). In only three clicks and one key stroke, the display is simplified to focus in on the events of interest.