DOI: 10.1111/cgf.14820
Eurographics Conference on Visualization (EuroVis) 2023
D. Archambault, R. Bujack, and T. Schreck
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 3

FlexEvent: going beyond Case-Centric Exploration and Analysis of Multivariate Event Sequences

S. van der Linden[1], B.M. Wulterkens[1], M.M. van Gilst[1,2], S. Overeem[1,2], C. van Pul[1,3], A. Vilanova[1], S. van den Elzen[1]

[1]Eindhoven University of Technology, the Netherlands
[2] Sleep Medicine Center Kempenhaeghe, the Netherlands
[3] Maxima Medisch Centrum Hospital, the Netherlands

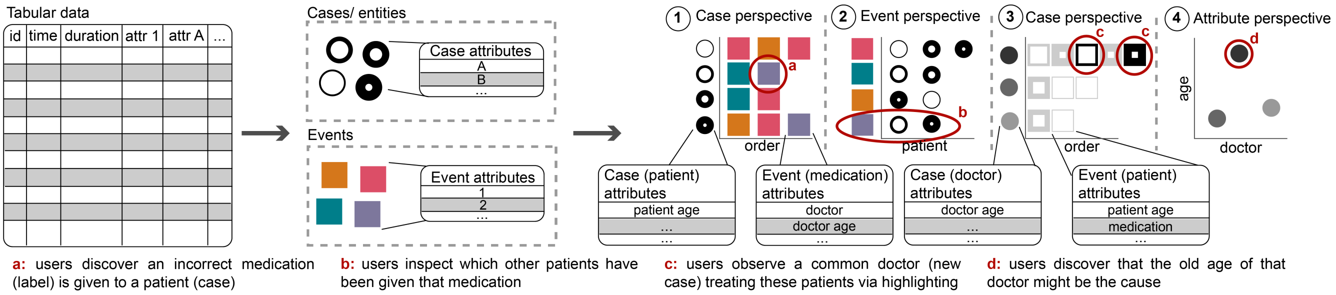

Figure 1: *FlexEvent's tabular data input contains an id, timestamp, duration and at least two attributes. One attribute, the case identifier, is chosen to form event sequences and another attribute represents the multivariate events (label). Both of them have attributes. Current case-centric exploration hinders in answering hypotheses requiring different perspectives. Therefore, we enable users to effortlessly switch between perspectives that can be inspected simultaneously using generalized scatterplots that we call FlexiPlot tiles and visualization multiples.*

Abstract

In many domains, multivariate event sequence data is collected focused around an entity (the case). Typically, each event has multiple attributes, for example, in healthcare a patient has events such as hospitalization, medication, and surgery. In addition to the multivariate events, also the case (a specific attribute, e.g., *patient) has associated multivariate data (*e.g., *age, gender, weight). Current work typically only visualizes one attribute per event (label) in the event sequences. As a consequence, events can only be explored from a predefined case-centric perspective. However, to find complex relations from multiple perspectives (*e.g., *from different case definitions, such as doctor), users also need an event- and attribute-centric perspective. In addition, support is needed to effortlessly switch between and within perspectives. To support such a rich exploration, we present FlexEvent: an exploration and analysis method that enables investigation beyond a fixed case-centric perspective. Based on an adaptation of existing visualization techniques, such as scatterplots and juxtaposed small multiples, we enable flexible switching between different perspectives to explore the multivariate event sequence data needed to answer multi-perspective hypotheses. We evaluated FlexEvent with three domain experts in two use cases with sleep disorder and neonatal ICU data that show our method facilitates experts in exploring and analyzing real-world multivariate sequence data from different perspectives.*

CCS Concepts

• ***Human-centered computing*** $\rightarrow$ ***Visual analytics; Visualization;***

1. Introduction

Event sequences are present in many domains, examples range from healthcare [GS14] to vehicle maintenance [CXR17] to security [CvW17]. The event sequences are defined by an entity, we call this a case (*e.g.,* patient or vehicle). Often, the event sequence data is multivariate, each event has a set of attribute values, a start time and a duration. Also, the cases have associated multivariate data. It remains a challenge to visualize and analyze multivariate event sequences taking into account the full context [BSKR18, GGJ*21]. Users need to answer questions and hypotheses about the relations between the different attributes of the cases and the events within one, or between multiple, sequences. For example, "Which doctors prescribe uncommon medications at specific moments in a patient's treatment plan, and, is this influenced by the age of the doctor?". Answering such questions, is currently difficult as event sequences are defined by a fixed case and only visualized from this case-centric perspective. To answer the question, users first need to explore the medication given to each patient (case, Figure 1(1)) to find uncommon medication given the order of previous medication. Users need to see which patients were given this uncommon medication in relation to other medication types and patients (2), which doctor(s) (now the doctor becomes the case) prescribed this (3), and finally, inspect their age (4). Users need the ability to analyze which cases have an event with a certain value for its medication attribute in relation to other medication values, *e.g.,* (2), change the case and, therefore, the sequences from patient to doctor, *e.g.* (1) and (3), and relate events or cases to other attributes (4). The choice, for example, which attribute defines the case, in which context, depends on the user's questions. Also, additional derived attributes are needed. For example, to discover how often these doctors prescribed this uncommon medication. This simple example illustrates the need to interactively explore the data from multiple case-, such as (1) and (3), event-, such as (2), and attribute-, such as (4), centric perspectives. Exploration should be supported with the ability to effortlessly switch between and within perspectives and discover relations through simultaneous inspection.

State-of-the-art visualization techniques do not address the need to explore multivariate event sequences from multiple perspectives, see Figure 2A. First, most techniques visualize the event sequences only from a univariate perspective with a row-based visualization, such as hierarchical visualizations [GGJ*21, GS14] or Sankey flow diagrams [GGJ*21, WMH*21]. Univariate entails that all events show the same, single attribute (label) next to the case it belongs to, time and duration. This results in, for example, only using the medication attribute and ignoring the dosage, doctor, and hospital department attributes. Given the current case-centric approach, it is difficult to visualize the data from an event- or attribute-centric perspective (S1). Second, the visualizations only present the data from a fixed case-centric perspective (S2). It is typically not possible to change within one perspective, *e.g.,* switch the case from patient, see (1), to doctor, see (3). Further, it is not possible to switch between perspectives, *e.g.,* from a case-centric perspective of visualizing the medications per patient, see (1), to an event-centric perspective visualizing which patients received a medication, see (2). Third, to circumvent this, most visualizations provide additional visualizations to display case metadata [KPS15, JGC*20] or other event attributes [CvW17]. However, the connection with the events and cases in the sequences is lost. It is, therefore, difficult to explore multivariate relationships in context (S3).

We present FlexEvent; a domain-independent interactive visualization method to analyze multivariate event sequences. With FlexEvent, users can change the case and event label attribute, or switch between a case, event and attribute perspective. This enables users to answer hypotheses that include relations between any two or more multivariate attributes, events, cases or a combination of these. Also, users can explore the data without needing queries upfront. The presented methods are generic and domain independent. However, in this paper we use the health-care domain as a running example due to our collaboration with domain experts that provide real-world event sequences. Figure 2B depicts our contributions versus current state-of-the-art, see Figure 2A:

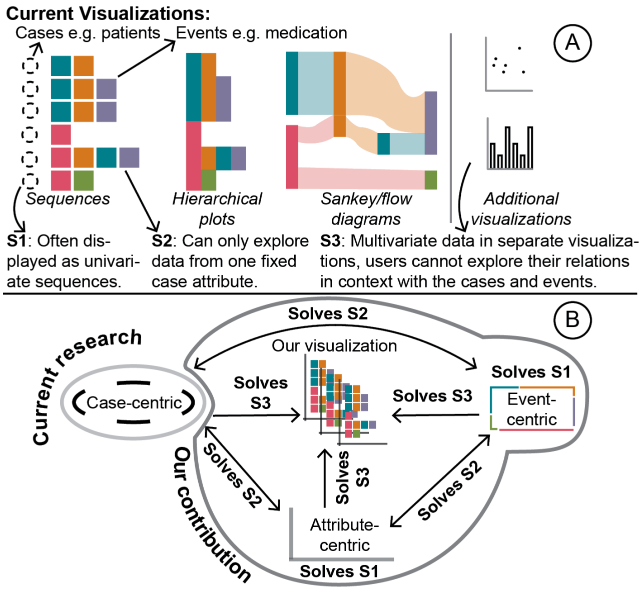

Figure 2: *(A) current visualizations of event sequences are displayed together with their shortcomings (indicated as S1-3). (B) describes how the case-, event- and attribute-centric perspectives give users the ability to switch between them and explore the data in context, which solve the shortcomings (our contribution).*

- A method to analyze multivariate event data from multiple case-, event- and attribute-centric perspectives (S1, Figure 2B). Users have the flexibility to switch within or between the three perspectives (S2, Figure 2B), without losing the context compared to the traditional fixed case-centric perspective (S3, Figure 2B).
- The ability to derive additional attributes to enrich the data. Users can compute additional user specified information to discover non-trivial patterns between events or cases, *e.g.,* frequency of uncommon medication, or discover multivariate relations by computing combined information of multiple attributes.
- The ability to display the different perspectives side-by-side and link them together to present the information in context.
- An implementation of the proposed visualization method tested on two data sets with tens of thousands of events and tens of attributes. We evaluated two use cases with three domain experts to test if users can explore the data as intended with our method.

The paper is structured as follows, we identify related work (Section 2), user tasks (Section 3) and our input data (Section 4). Next, we present FlexEvent (Section 5), an evaluation (Section 6) and our use cases (Section 7). Finally, we provide a discussion, address limitations and present the conclusions (Sections 8 and 9).

2. Related Work

First, we discuss event sequence visualizations in general and then focus on multivariate event sequence visualizations.

2.1. Event Sequence Visualization

Event sequences are typically visualized using timelines [GGJ*21, CvW17, LWD*16, WGW*20, CXR17, YM22, HPK*21], hierarchical visualizations [GGJ*21, WGGP*11, WG12, GS14, LWD*16, JGC*20, YM22], matrix visualizations [GGJ*21, YM22], bar chart visualizations [GGJ*21, MSD*16], and Sankey flow diagrams [DBZSD20, KPS15, HLI*15, GGJ*21, MXC*19, WMH*21, YM22, WLS*21]. Often, case metadata is displayed next to the main visualization serving as a filter [KPS15, GS14, JGC*20, MXC*19, WLS*21]. Two well-known examples of hierarchical visualization are DecisionFlow [GS14] and LifeFlow [WGGP*11], one of the first to use flow visualizations for sequential event data. These methods aim to provide event sequence overviews and comparisons. To provide overviews, the event sequences can be summarized [LWD*16, CXR17, KPS15] using aggregation and displaying different levels of granularity in multiple linked views [LWD*16, CXR17, MSM*21] to, for example, display cohorts [KPS15] or reduce the information loss [CXR17]. Methods also show progression patterns for different stages in the event sequences [GJG*18, WMH*21, GXZ*17, WLG*21] to provide overviews.

High-dimensional data leads to visual and computational scalability challenges. Visual scalability concerns a large number of sequences [DSP*16, WGGP*11, MSM*21], long sequences or a large number of events [GS14]. Computational scalability is addressed from computation time [LWD*16, CPYQ18]. To deal with a large volume of events and sequences, users have to query a subset of the data upfront [JGC*20, GS14, GJC*19]. This, however, limits the overview, hides relationships and hinders explorative capabilities in general. Others use different levels of granularity [CPYQ18, MSM*21] potentially combined with aggregation/simplification (*e.g.,* frequent pattern mining [CXR17, LWD*16, MXC*19, PW14, WGW*20, WLG*21, WLS*21]). Similarly, in this work, we also provide users with two levels of granularity in combination with frequent sequential pattern mining to deal with long sequences.

2.2. Multivariate Event Sequence Visualization

Traditional event sequence visualization techniques present the data from a fixed case-centric perspective and consider the events as univariate. Traditional generic business intelligence frameworks, such as Tableau [Tab23] or Power BI [Mic23], support multivariate analysis. However, they do not have a notion of a case or an event and do not consider the properties of event sequences. All reduce the ability to answer hypotheses that involve different perspectives. In general, it is possible to pre-process the data differently to switch the case and event representation attribute. However, this requires programming skills, is time and labor intensive, loses the relation to other perspectives, and does not enable users to explore complex multivariate relations. Several authors [CvW17, ZDFD15, FKSS06, KPS15] use the multivariate data as a visual filter and form sequences based on users' filter selections. Events can be filtered using regular expressions [ZDFD15, CvW17] or node-link diagrams [KPS15]. To support additional attribute exploration and analysis, data distributions of the attributes are shown as small charts on the side [CvW17, XSZX22]. For initial data insight, we also provide users with attribute distribution visualizations using scented widgets [WHA07].

Methods that integrate the multivariate data, are the works of Wu et al. [WGW*20, WLG*21], Wang et al. [WMH*21], Loorak et al. [LPK*15], Xu et al. [XSZX22] and Bernard et al. [BSKR18]. For example, they used glyphs in a table [LPK*15] or summary statistics in aggregated blocks [BSKR18]. The examples closest to our contribution are from Wu et al. [WGW*20, WLG*21] in tactical patterns in the racquet sports domain. They display frequent patterns of event sequences as glyphs, where each part of the glyph represents a different attribute [WGW*20] or in a flow diagram where the nodes are the frequent patterns represented by the glyph [WLG*21]. A dimensionality reduction plot helps to steer users to interesting sequences. Wang et al. [WMH*21] also use dimensionality reduction with clustering to represent states and highlight transitions. In our work we also use dimensionality reduction but instead link it back to the events and sequences. All reviewed work that integrate the multivariate data focuses on specific use cases with no more than fifteen attributes. Except for Xu et al. [XSZX22] but their focus is on distance measures.

Overall, it remains a challenge to visualize multivariate event sequences. If multivariate events are displayed, their focus is often on a small number of attributes. It is difficult to explore the data from different perspectives (S1, Figure 2B), switch within or between perspectives (S2), or analyze a large number of attributes in context (S3). In this work, we address these challenges to enable users to explore and analyze complex multivariate relationships in context between different cases, event attributes and sequences from different perspectives.

3. Data Type, Definitions and User Tasks

In this section, we discuss definitions and identify tasks from literature and extend this with our own, based on interviews with experts.

Multivariate Event Data: event sequence data often comes in a tabular form, consisting of categorical, numerical and ordinal data. This tabular data typically consists of two linked tables, one with the metadata, data related to the cases, *e.g.,* the patient's age and one table with all the event data. To realize the general applicability of our method, we define an event as an instance of a discrete state containing multiple attributes with a timestamp and a duration. Typically, not all event attributes can be shown simultaneously. Also, users do not need all attributes all the time. Therefore, an event is represented by a type (label), which is one of its attributes. The attribute determining the event type is the same for all events, *e.g.,* all the events are represented by their medication value (the medication attribute is the event type). We define an event as $e_i = \{a_{i0}, \ldots, a_{ij}, \ldots, a_{im}, t_{is}, t_{ie}\}$ where a_{im} is attribute m of event e_i with an attribute value $a_{im} \in \Theta$. Θ is the alphabet of all possible event types of a_m and t_{is}, t_{ie} are the start and end time of e_i. Events are grouped around one attribute (the case attribute) and ordered on another attribute (often chronological) to determine the sequences. The definition of a case is $c_u = \{e_0, \ldots, e_i, \ldots, e_n\}$ where $a_{0m} = \ldots = a_{im} = \ldots = a_{nm}$. Moreover, the sequence belonging to case c_u has the following definition, $s_u = [e_0, \ldots, e_i, \ldots, e_n]$ where $t_{si} \leq t_{s(i+1)}$. In this paper, we assume that the data is homogeneous; all events have the same set of attributes. Note, it can occur that attributes have missing values.

Perspective Definitions: previously, we described the users' need to inspect the data from a case-, event- and attribute-centric perspective. We define the *case-centric* perspective as reasoning about the data where there is one sequence of events for each possible value of the case attribute (*e.g.,* one event sequence per patient). We define the *event-centric* perspective as reasoning about the data where for each possible event type attribute value all possible case values are known in context (*e.g.,* which medication was prescribed to which patients, disregarding when or how often). If users are interested in when, how often or in what order patients are prescribed certain medications, users can change the medication attribute to the case, and the patient attribute to the event type. Finally, we define the *attribute-centric* perspective as reasoning about the event type and case values based on other attribute values that are related to either the events or cases instead of reasoning from a sequence level. For example, hypotheses about the dosage in relation to the doctor prescribing it for each medication event.

User Tasks: first, we examined literature, papers mentioned in the related work, and three event-related (VAST) challenges [Vis17, Vis18, vDBCDW18], to identify general tasks. Second, we held interviews with domain experts, five sleep researchers and one clinical researcher, to confirm the tasks, see Section 7. During the interviews, we asked the experts to describe their current analysis process, their current analysis tools and example hypotheses they need to answer. We used a thematic analysis [BC06] to code the findings in non-domain-specific themes. For example, "What is the relation between different attributes?" instead of "What is the relation between different arousal attributes and the sleep stages?". We noticed that these generalizations from the different domains in literature and the interviews were similar. Users need a combination of a case-, event- and attribute-centric perspective to perform each task. Each theme led to the identification of one user task:

- T1 **Explore and analyze relations within the data from different perspectives:** experts and literature, the generalized tasks, *e.g.*, from the BPI challenge [vDBCDW18], mentioned the need to explore and analyze relations from the three different perspectives. However, literature focused on presenting the data from one fixed case-centric perspective. These relations are based on order, duration, time, frequency and multivariate attributes between events, cases and sequences. Additionally, all three perspectives need to be analyzed together to discover relationships in context.
- T2 **Exploration of (frequent) patterns:** users need to know which events and/or cases (do not) follow a particular pattern based on values for a combination of attributes. Users often need multiple perspectives to discover these patterns, see Figure 1. These patterns are based on order, duration, time and frequency of events based on the event type or case attribute possibly in relation to other attributes, the three perspectives.
- T3 **Enrich the data by computing derived attributes:** during the interviews, users indicated the need to interactively compute extra information for a (sub)set of the events, cases, and attributes to discover non-trivial patterns, *e.g.,* computing the relationships between tens of attributes.
- T4 **Grouping of multivariate event sequences:** users need to group and inspect events, event-centric perspective, and/or cases, case-centric perspective, based on a combination of attribute values, attribute-centric perspective.
- T5 **Comparing multivariate event sequences:** users need to compare which cases belong to which (groups of) events, which event types belong to which (groups of) sequences and what the difference in a set of attribute values (the three perspectives) is between individuals or groups. Also, the interviewees and challenge descriptions mentioned that comparison between more than two attributes or patterns from different perspectives is required to explore complex relations, see T1 and T2.
- T6 **Detailed information:** users need to see the details of multiple, specific attribute values of one sequence, event and case.
- T7 **Detailed context of pre-, co- and post-occurring events:** users need to see which events happened before, after or at the same time as a certain (set of) events in sequences of interest, case-centric perspective, and which cases occurred per event type, event-centric perspective, based on the event types and possibly a set of other attributes, attribute-centric perspective.

4. Input Data Structure

Our method aims to support domain experts in all identified tasks for the analysis and exploration of multivariate event sequences. Therefore, we require a flexible data structure. As mentioned in Section 3, event data often consists of a metadata and event data table. However, it is assumed that users do not change the case attribute, as with each change the linked metadata also changes. When a different case attribute is selected, the sequences are formed based on those attribute categories. When the selected case attribute is numerical, its values are binned into categories [FD81]. In line with the example from the introduction, instead of displaying medication sequences for each *patient*, users might want to display a sequence *per doctor* (new case). The metadata, such as age of the patient, are no longer metadata of the new case, doctor. Since we aim to offer users the flexibility to switch the case and event type attributes (T1) in a generic manner, the input to our visualization method is one big table constructed as a join of the original metadata and event data tables. The metadata adds one column per metadata attribute to the joined table and does not influence the number of rows (events). Each row represents one event and the columns represent all attributes. FlexEvent needs at least a timestamp, duration, id and attribute(s) as data input. This method is generic and supports dynamically selecting the case and event attributes.

To go from the raw data to the table format described above, we create a new event, a new row in the table, each time a value change occurs in one of the attributes, see Figure 3. For example, we start with the first event (e1), then a change in attribute two occurs, marking the start of the second event, the red line in Figure 3. If none of the attributes occur, *e.g.,* the time between e4 and e5, no event is created. This general structure makes it possible, to easily add new derived attributes during the analysis process.

5. FlexEvent Visualization Method

This section describes the visualization and interactions of FlexEvent. The graphical user interface consists of three parts; the settings menu, see Figure 5a, scented widgets for filtering, see Fig-

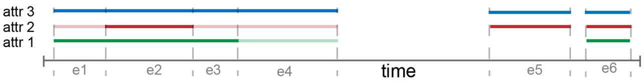

Figure 3: *Creation of multivariate events (e1-6) from raw tabular data. Different color shades represent different attribute values.*

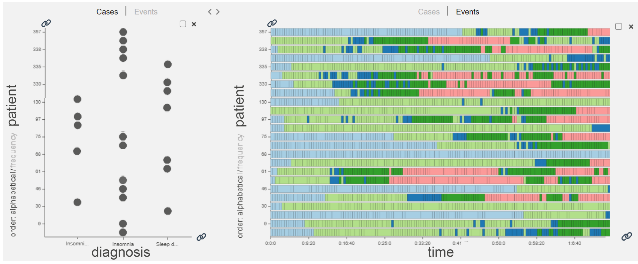

Figure 4: *Two FlexiPlot tiles that display cases as circles and events as rectangles in the sequences. Users changed the attribute perspective (left) to display the diagnosis instead of a constant.*

ure 5b, and visualization multiples, see Figure 5c. The visualization multiples consist of multiple FlexEvent tiles which users can flexibly add, remove and position in a grid.

5.1. Single FlexEvent Tile

The main component of FlexEvent – FlexiPlot tiles– generalizes all perspectives into a single, flexible visualization tile, see Figure 5c. The tiles are built upon the concept of a scatterplot, meaning that the tiles inherit the advantages (and disadvantages) of scatterplots; *e.g.,* it is familiar and easy to understand for novice visualization users. We started from the observation that all existing event visualizations are essentially scatterplots. In general, the events are positioned in a scatterplot-like visualization where they are positioned according to two attributes; each event is part of a sequence defined by the case identifier. Next, the event sequences are ordered based on this case identifier (each sequence has its own row) and each event within a sequence is horizontally ordered on time (chronological). The cases are often implicitly encoded; each event sequence represents a case. Here we make them explicit and introduce a separate (linked) FlexiPlot tile to show these.

For a clear distinction between cases and events, we made the design decision to use circles for cases, and rectangles for events; much like a often used event visualization. However, in contrast to a traditional event visualization, we enable users to change both axes attributes. This opens up the ability to incorporate multivariate data in the exploration. For example, cases (patients) can be ordered horizontally according to an attribute (age). Without changing the vertical position, of one event sequence per case, this integrates the multivariate data in a traditional event visualization. This enables users to explore the data and discover relations.

Color represents the value of a selected attribute. This can be a derived attribute representing information about the relation between multiple attributes. An interactive legend is chosen to adapt known and proven color schemes for numerical (default Viridis [Ent15]) and categorical (default ColorBrewer [BHS*13]) data (see Figure 5a). Overlap between items is visualized using a grey-scale density encoding. Interface controls offer users the flexibility to switch between cases and events, and change within or between perspectives instantaneously. We chose to have each event (row in the data) represented by a case. This guarantees that both the case and event glyphs persists on changing the axes of the FlexiPlot tile. This in turn, enables us to use animation in the form of smooth transition effects (T1) when changing the axes attributes, see Figure 5c and the videos in supplemental material. The animation helps users to track items of interest and preserves the mental map. Further, FlexiPlot tiles support zooming and panning for scalability and to retrieve detailed information (T6). Next, we describe how FlexiPlot tiles support different configurations for the exploration and analysis from all three perspectives.

Case-centric perspective: we start by presenting a familiar view, a case-centric perspective showing the sequences with the case attribute on the y-axis and time on the x-axis, as is the general solution in related work, see the right plot in Figure 4. The left FlexiPlot tile starts with displaying the cases ordered on case identifier, and for the x-axis we pick a constant value, to align the cases vertically. This provides users easy access to an attribute-centric perspective to start the exploration and analysis, *e.g.*, select the diagnosis per patient (the left plot of Figure 4). In addition, this helps users to familiarize with FlexiPlot tiles and visualization multiples concepts.

Event-centric perspective: users are enabled to inspect an event-centric perspective by changing the FlexiPlot tile axes and select whether events or cases are displayed in the plot. For example, to inspect for each snore value (event type) which BMI bins (cases) were present, see second plot in Figure 6. When users change the event type attribute, see Figure 5a, the axes of the first two plots initially display the event type attribute on the y-axes of both FlexiPlot tiles. For the x-axes, a constant and the case attribute are used.

Attribute-centric perspective: users are enabled to change the attribute variables displayed on the axes by clicking on them, *e.g.,* patient in Figure 5c or change the color, see Figure 5a. This enables users to discover relations between multiple attributes in relation to the event types and cases. Each time users change one of the axes attributes the cases/events are animated to their new positions to give a global impression of the changes, to enable tracking of specific items, and to help with pattern discovery (T2). As mentioned earlier, which attributes are metadata of the cases depends on the case attribute and differs per case attribute. The same holds for the event attributes. Therefore, users can decide which attributes are of interest to inspect the relationship to either cases or events. Also, users are able to discover and analyze relations between multiple attributes via the automated methods, see Section 5.4.

Additional Design Alternatives and Considerations: there are a number of design decisions worth mentioning. In earlier designs, we choose to have a single circle glyph for each case (in contrast to one case glyph per event). This, however, posed problems as a metadata attribute can have multiple values per case. For example, the weight of a patient can change over the time period of one sequence from 80 to 78 kg. Using only one glyph to represent the case poses the following problem: if the weight is displayed on the x-axis and the cases on the y-axis, one case has two corresponding values on the x-axis, which can no longer be represented with

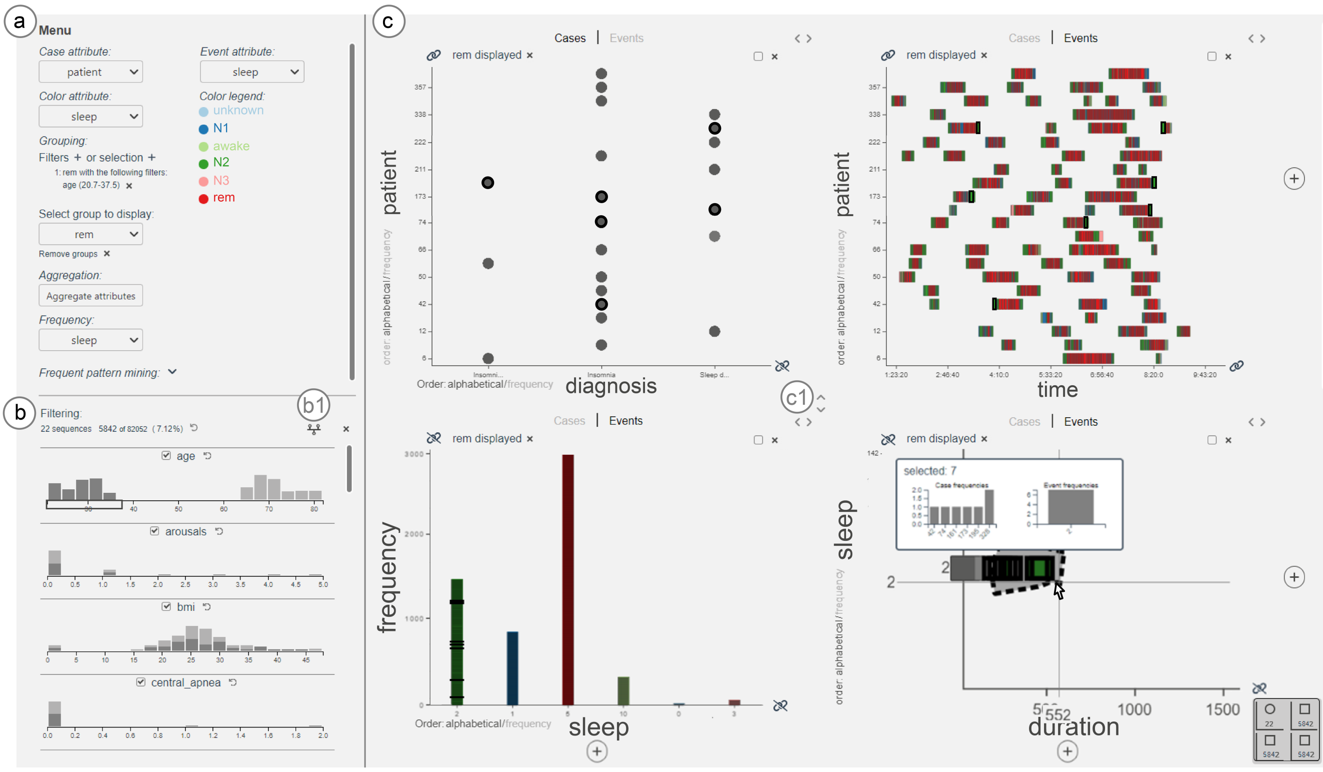

Figure 5: *FlexEvent's graphical user interface has a settings menu to change the case and event perspectives along with controls for the automated methods (a); scented widgets for filtering (b); and, event and case visualizations based on reconfigurable small-multiple plots, FlexiPlot tiles (four are shown in c), to go beyond case centric exploration. Pre- and post-occurrence filters (b1) can be used to discover which sleep stages occur around rem sleep (top-right) and users discovered that especially sleep stage N2 and N1 occur often around rem (bottom-left) with in general a short duration. Only, seven events had a slightly longer duration and are highlighted in the tooltip (bottom-right) and the linked plots. There does not seem to be an obvious relation to the diagnosis (top-left).*

a single glyph in our FlexiPlot tile approach. Therefore, we concluded it is not possible to only use one glyph per case. In line with our FlexiPlot tile approach, we display a circle for each attribute value on the corresponding position on the x-axis. This results in the patient being represented by two circles instead of one, each representing a part of the patient. Their y-axis position is the same because they belong to the same case, but they differ in x-axis position, corresponding to 80 and 78 kg. The inner part of the circle is colored dark gray to indicate that this circle only represents part of the case, for another example see third plot in Figure 6. We tried alternatives, such as showing the average or most frequent category of attributes. However, we concluded that this would mislead users, especially if users need in-depth information (T6). Also, the inner and outer circles of these partial cases have fixed proportions because we already have three dimensions; position, shape and color. This multi-value case problem does not occur for the events because each event has one value for all attributes.

5.2. Multiple FlexiPlot Tiles

Next, we enable users to add/remove additional FlexiPlot tiles for a small-multiple like exploration experience [vdEvW13] having tiles arranged in a grid with flexible axes [CVW11], see Figure 5c for four FlexiPlot tiles. This supports users with the explorative process and helps with pattern discovery (T2) and comparison (T5). The default configuration for the multiples consists of two FlexiPlot tiles, one showing the cases, and another the events. The axes of both FlexiPlot tiles are chosen such that it resembles common event visualization where each row represents one sequence of chronologically ordered events belonging to one case (see right tile Figure 4).

When users hover over an event, the same event and corresponding case are highlighted in the other plots. This is similar for cases. The linking combined with dynamically changing the axes, enables users to easily switch between the case and event perspectives (T1) for pattern discovery (T2). In addition, on hover, a rich tooltip is displayed showing the value of the event and case attribute (T2,6). When multiple points are selected, all their event and case values with corresponding frequency are shown in the tooltip similar to related work (see Figure 7, plot 1).

Moreover, axes attribute selections can be linked to support pattern (T2) and comparison (T5) analysis (*i.e.,* all horizontally aligned FlexiPlot tiles have a synchronized y-axis attribute, and, all vertically aligned FlexiPlot tiles have a synchronized x-axis attribute). Furthermore, the zooming and panning can be linked in

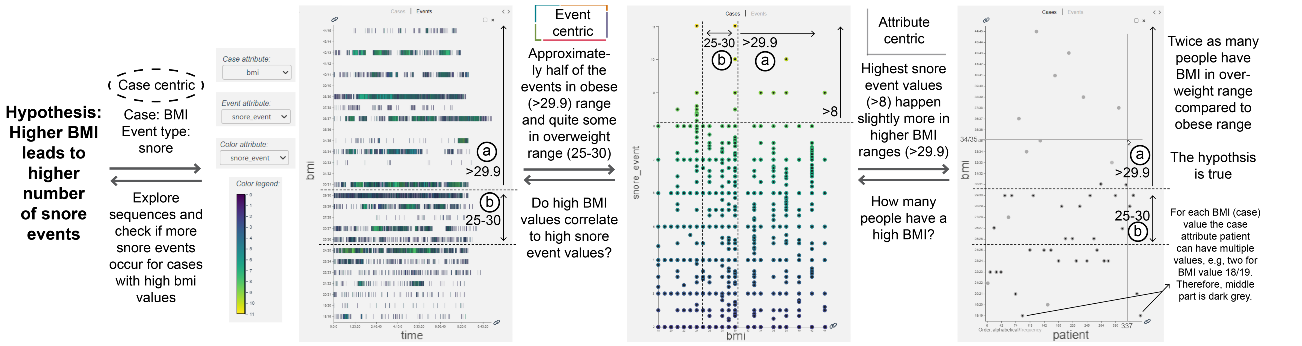

Figure 6: *The users' thought process. The color is based on the snore events values, yellow means many consecutive snores in one event and purple only a few consecutive snore in one event. Users filtered out snore events with a value of zero. They go back and forth between the plots using the different perspectives of FlexEvent and use them simultaneously to answer the hypothesis about high BMI groups (a/b).*

multiple FlexiPlot tiles for synchronized zooming and panning. Also, the size of a FlexiPlot tile can be adapted. The design choices of the individual and linked FlexiPlot tiles have the advantage that they are easy to understand, generic, and easily adjustable to provide the flexibility users require. Next, we describe the filters and derived attributes. These are linked to all the FlexiPlot tiles.

5.3. Attribute Value Filters

The scented widgets (see Figure 5b) with distributions of each attribute act as filters, to support pattern discovery (T2). For numerical attributes, we bin the data according to the Freedman–Diaconis rule [FD81]. The bars of the histograms are partially colored darker to indicate the current selection. Moreover, users can temporarily remove irrelevant attributes by deselecting the check mark, which collapses the distribution plot. Furthermore, there is a pre-, co- and post-occurring events filter (T7), see Figure 5b1. Users can select how many events they need to see before, during, and after each occurrence of an event type category, or event type value range.

5.4. Automated Methods

Regularly, the available data on its own is insufficient to answer hypotheses involving non-trivial patterns or relations. Therefore, users need the ability to enrich the data with derived attributes using automated methods (T3). Based on user interviews, we offer five automated methods: grouping, dimensionality reduction (DR), clustering, frequent patterns, and frequency computation. These (combined) methods guide users to interesting patterns and (combinations of) attributes. For example, users run DR with clustering (*e.g.*, Figure 7) and, afterwards, frequent pattern mining on all event attributes to reveal multivariate frequent pattern for further analysis. These methods all add derived attributes to the data table as new columns which can then be used similar to the base attributes. We do not make a distinction between core and derived attributes.

Grouping: users can create (non-mutually-exclusive) groups (T4) using lasso selection or scented widget filters. The different groups can be labelled and are listed in the menu together with their filter information, see Figure 5a 'rem'. Users are enabled to visualize the same or different groups in the plots for comparison (T5).

Dimensionality Reduction: users need the ability to compare more than two attributes to inspect detailed information or search for complex inter-relations (T2,6) from an attribute-centric perspective. We considered using glyphs, however, we aim to provide a generic method where tens or even a hundred attributes can be compared. To support this, we use DR combined with post-hoc clustering. Here we use t-SNE (t-distributed Stochastic Neighbor Embedding) [VdMH08] because it recovers well-separated clusters [LS19] in the data and preserves local structure [VdMH08]. Also, it is broadly available and offers easy solutions to input custom distance matrices. However, other methods, such as UMAP, are considered but currently not implemented. Running this automated method creates a derived attribute that represents the multivariate relations on a high-level. As input to the DR algorithm, users select a combination of attributes from all types, *e.g.,* numerical, and (a subset of) the data they are interested in, and indicate attribute importances. DR can be applied to both the cases and events. If only case attributes are selected, the DR is applied on the cases (circles) and otherwise it is applied on the events (rectangles). Computing the distance between a combination of categorical and numerical data attributes is an open problem and domain dependant. To address this issue, we combine the categorical and numerical attributes in a linear, quick and weighted manner using existing measures. We try to remove unwanted bias towards either categorical or numerical data as much as possible, but this is not our focus.

After normalizing the numerical attributes using Min-Max scaling, we use weighted euclidean distance to compute the distance between data points for each numerical attribute. This weight is user-defined based on the importance of the attribute for the task at hand. By default all attribute weights are one. The average distance is computed based on all individual distances of the numerical attributes between two data points. We use the weighted Jaccard distance to calculate the distance between categorical attributes in a similar manner. The total distance between two events is then the sum of the average euclidean distance and the Jaccard distance, each time their respective proportion of either numerical or categorical attributes compared to all selected attributes. Currently we picked this distance measure because it suited our needs, but others could also have been used. This distance matrix is used as input to the t-SNE DR algorithm. We adapted the algorithm to progressively return intermediate results; FlexEvent updates the derived x and y positions every 200 iterations until the Kullback–Leibler divergence between two iterations is lower than a user-defined threshold or the maximum number of iterations. If memory limitations are reached the GPGPU implementation [PTM*19] with linear complexity is used instead. As a consequence, we introduce a slight unwanted bias towards categorical attributes.

Clustering: after DR, the events are clustered using a density-based clustering algorithm [skl22]. We selected density-based clustering because it does not require to specify the number of clusters and is indifferent to the shapes of clusters, but other clustering methods can also be used. Cluster parameters, such as the minimal samples per cluster, are exposed to users. Users can adjust the parameters to rerun the clustering or manually adjust the clusters via mouse lasso selection. Each cluster has a distinct color and cluster number (Figure 7c). The tooltip includes a summary of the attribute frequencies of occurring values in the DR/clusters (Figure 7a). Also, users can save and load previously computed DRs.

Frequent patterns: navigating through a large number of sequences with additionally a long length is challenging. To help users find interesting patterns (T2), we offer frequent pattern mining. This helps to aggregate large amounts of sequences based on one or multiple attribute(s). We considered four algorithms; the algorithm from Bertens et al. [BVS16] and three from Wu et al. [WLG*21, WGW*20, WLGW22] offer multivariate frequent pattern mining. However, the computational costs are too high to be used interactively [BSKR18, WGW*20, WLG*21], it had a low efficiency on large data sets [WGW*20] or was specialized for short sequences with few attributes [WLGW22]. Therefore, FlexEvent uses the VGEN [FVGŠH14] algorithm (to compute frequent patterns of one attribute) possibly in combination with DR (to mine multivariate patterns) to provide frequent patterns in tens of seconds for our complete data sets. After DR and clustering, there is a newly derived attribute (cluster number) representing the relation between the attributes in the DR. Users can select the cluster number as the attribute for the frequent pattern mining to mine patterns of often occurring cluster numbers in a certain order that represent multivariate event patterns. Users can interactively rerun the pattern mining on different subsets of the data or with different parameters.

Frequency computation: according to user interviews and tasks, see Section 3, it is important for users to see, among others, the attribute frequencies for pattern discovery and analysis (T2). Therefore, users can select attributes to compute and visualize the frequency of all categories. In case the attribute is numerical the values are binned. We provide two options; display the total frequencies of the (filtered) data, see bottom left plot in Figure 5c, or split it up per value on the axis. For example, if users display sleep stage (frequency attribute) and frequency on the axis, users see the total frequency of all sleep stages of all patients (cases) added together. If users would display patient and frequency on the axes, we visualize the frequency of the different sleep stages for each patient.

6. Evaluation

The evaluation compares our method to previous methods. In general, given enough effort and experience, similar concepts could partly be realized in existing event sequence visualization methods or generic business intelligence frameworks. However, several aspects would be hard to realize since these frameworks are not built considering concepts we incorporate, such as the distinction between cases and events. This makes it difficult to realize the carefully designed interaction, such as the instantaneous, flexible switching of the three perspectives. To achieve this interaction the data would need to be pre-processed again hampering interactivity.

Closest to our work is Wu et al. [WGW*20] a method for analyzing multivariate event sequences with a limited number of attributes. However, it is impossible to switch the case attribute, inspect the event/attribute perspective or inspect the relation between many attributes. To switch the case attribute or inspect the relation between many attributes, users must program and pre-process the data differently. Then they need to start another instance without any linking or interactions between them. For univariate frameworks users would even need one instance per event attribute. Another option to retrieve information about a different case attribute or event/attribute perspective is to go through all event and sequence attributes manually. This is cumbersome, time-consuming and relies on the users' memory. Further, users still cannot extract all the needed information, *e.g.*, the order of the events for a different case attribute. FlexEvent addresses these limitations by offering all functionality in a single system.

7. Use Cases

Here we describe two use cases with anonymized real-world data. We used FlexEvent together with two sleep researchers (U1-2) and a clinical researcher (U3). U1-3 also participated in the interviews in section 3. Currently, users manually generate static business graphics, typically through scripting or notebook documents. These require data processing, are not integrated into their workflow and suffer from the limitations described in section 6. During the use cases, we explained and demonstrated how FlexEvent solves tasks. Also, the users completed a workflow based on high-level goals, modeled as action-target pairs [Mun14]. These goals are discover features (Section 7.1) and identify similarity (Section 7.2), also see additional material. They could ask for our support if needed. The associated tasks and time it took to complete a task are indicated between round brackets. Sessions lasted an hour and a half.

7.1. Use Case 1: Discover Features of Sleep Disorder Data

The sleep data set consists of 82,052 events where each event has the same 25 attributes; 21 numerical and four categorical. In the starting case-centric perspective, there are 98 patients (cases) where the shortest sequence contains 192 multivariate events and the longest 1063 measured over two hours and 14 minutes and 10 hours and 16 minutes, respectively. U1 and U2 wanted to analyze features of patients with different sleep disorder diagnoses and ages.

First, we started with patient as case, and sleep stage as event type attribute. U1 and U2 explored what happened before or after rem sleep for young people. After filtering and using the pre-, co- and post-filter on rem sleep (T7), U1 and U2 saw that some patients have small blocks of rem spread out through the night and

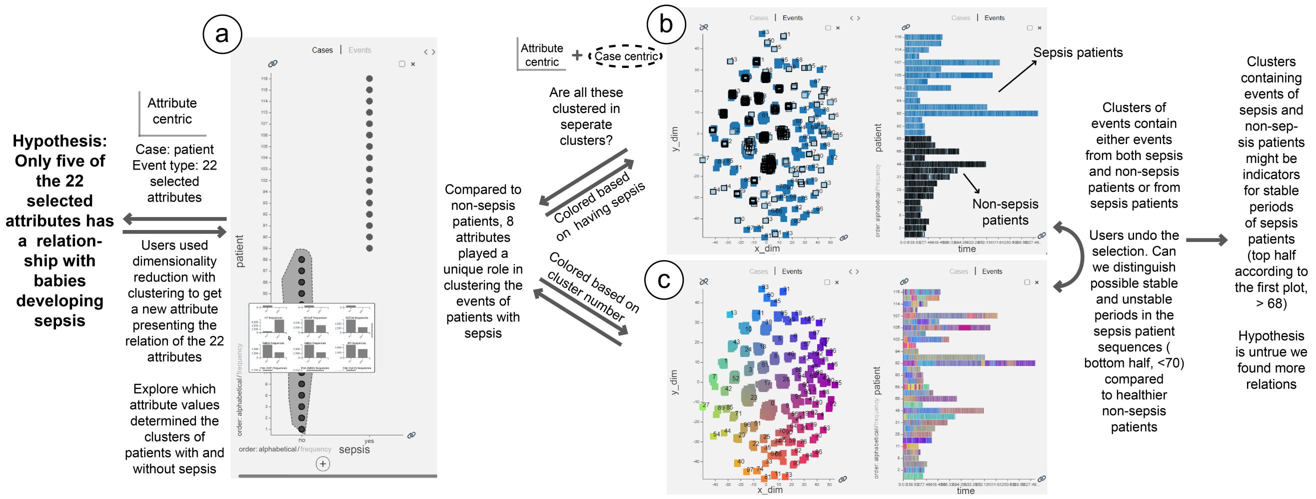

Figure 7: *Users explored the relations of the cases (patients) to 22 other attributes with three plots (a-c) to test a hypothesis. The analysis process is not linear, users go back and forth between the plots and need to combine all of them to answer the hypothesis.*

others have longer blocks of rem more at the end of their night (T2, 2:19 mins). To get more information the researchers needed the relations with other attributes from other *case and event perspectives* (T1). Using this univariate approach (similar to their current tools) we cannot explore and analyse complex (inter-)relations through different perspectives. For this rich analysis we need FlexEvent with the three perspectives because univariate sequences with fixed cases, see Section 6, are insufficient. We derived a new frequency attribute (T3) and changed the plot to Figure 5 only with more patients. We saw that sleep stages N1, N2 and awake did occur quite frequently around rem and that there was almost no N3 (T2, 2:36 mins). This is in line with sleep research and the researchers experience. Only, they mentioned that now they can explore the data from different perspectives instantaneously and interactively. The durations were often less than 200 seconds except for a few events and the diagnosis (*attribute perspective*) did not seem correlated with the amount or time of rem events (8:55 mins). This was in line with known theory and their expectations. Moreover, we ran frequent pattern mining (T3) on the sleep stages to see patterns in sleep stages before and after rem and displayed these in the left top plot, changing the cases to events (T1). Two interesting patterns emerged, such as sleep stage N2 was followed by N3 and then N2 again (T2). Using the tooltip and the linked plots we discovered that this pattern occurred 12 out of 14 occurrences for patients with an insomnia diagnosis, 7 of the 9 patients (9:25 mins). They mentioned that this data enrichment (frequent pattern mining) could help in the future to discover patterns when answering new hypotheses about unknown aspects of sleep disorders.

Secondly, U1 and U2 were interested to see the known relation between BMI and high snore events for this data set. We changed the case attribute to BMI, the event type and color attribute to snore (T1) and filtered on high snore events. We changed the *perspectives* in the plots to visualize the data as in Figure 6. We explored the data for patterns (T2) by selecting the two highest BMI groups, see a and b in the plots in Figure 6, via the lasso selection and used the tooltip and linked plots to see the number of events, snore value per event and patients (T6). We repeated this for middle and low BMI categories. We saw that BMI and number of snore events are indeed positively correlated, *e.g.,* the high BMI group has 1000 more snore events compared to the low group (5:24 mins). The ability to change between and within perspectives within a second enabled U1 and U2 to answer their hypothesis within a couple of minutes without the need to re-process the data. Moreover, the users saw the potential of FlexEvent for researching new relations. They stressed that the added value is to have all the information in context, the ability to easily switch between case attributes, such as going from patient to BMI, and see relations between multiple attributes in context with the events/sequences.

7.2. Use Case 2: Identify similarity of Neonatal Data

The neonatal intensive care unit data set consists of 25,336 events where each event had the same 73 attributes; 17 numerical, 2 ordinal and 54 categorical. In the starting situation, there are 30 patients/sequences (cases) where the shortest sequence has 164 multivariate events spanning 240 hours and the longest sequence has 2160 multivariate events covering 1131 hours. U3 needed to explore factors contributing to babies developing sepsis.

At the start *case-centric perspective*, U3 selected as event type if a certain type of blood pressure sensor was used (T1), which U3 commonly analyses in relation to sepsis. She was surprised because for seven patients this usage did not occur at the beginning of the sequence, which is typically the case (16 secs). This piqued her interest (T2) and she wanted to combine this with three other attributes to confirm the possible hypothesis that this has a relation with low birth weights, having sepsis and the occurrence of a disorder. Answering this hypothesis was not possible from a univariate case-centric perspective without the option to switch the case attribute, see Section 6. Now the added value of FlexEvent occurred clearly. We switched to an *attribute-perspective* in the left plot by changing the attribute on the x-axis to birth weight, to visualize this attribute for each case (T1). She confirmed part of her hypothesis that the blood pressure sensor usage only occurs for patients with low birth weights using the two linked plots (T2, 1:25 mins). Also, by using grouping on (non-)sepsis patients (T4) and the pre-, co- and post-filter based on a certain disorder category (T7), U3 noticed that only two out of the seven patients, one with and one without sepsis, had usages of that blood pressure sensor attribute in context to that disorder (3:35 mins). U3 expected that there would be clear relation between the sepsis, disorder and blood pressure attributes (T2). To explore the relation between attributes of sepsis and non-sepsis patients further and see if it might be influenced by NEC (necrotizing enterocolitis), we reset the pre-, co- and post-filter and selected 22 attributes of interest for dimensionality reduction with clustering (T3). We purposefully omitted attributes known to give separation for non-sepsis and sepsis patients. We use three plots with different perspectives (T1) in the analysis, see Figure 7. By changing the color attribute to sepsis, we saw that there are quite some clusters belonging only to events with sepsis. Also, non-sepsis patients have fewer clusters. U3 thought this was very unexpected, see Figure 7b. She could not analyze this easily with her current analysis tools. By changing the color back to the cluster attribute, see Figure 7c, we observed that clusters that sepsis and non-sepsis patients have in common might be stable periods for the sepsis patients (22:28 mins). It is valuable to see when they happen over time in the sequences. By selecting the sequences of patients with NEC (T4) with the mouse, we discovered by using the tooltip that there were eight blood pressure, temperature and three oxygen-related attributes that often occurred (T2,6) compared to non-sepsis patients (T5). The tooltip displayed the categories of the attributes present in the dimensionality reduction clusters (3:32 mins). U3 mentioned these results are very interesting because U3 would not have associated that with NEC. They might point to additional parameters they need to take into account in their machine learning models for predicting sepsis, which they had not thought of before.

7.3. Overall Feedback

The two use cases show the usefulness of this method by answering users' hypotheses based on the defined tasks that go beyond state-of-the-art. All users mentioned they would like to use FlexEvent in the future because it would speed up their exploration and analysis process, users do not need to re-process the data and all data is interactively linked, see Section 6. Further, all functions are well-integrated. U1 and U2 mentioned: “You can visualize many different aspects of the data in a simple manner, that is why this tool would help us to explore big data sets and get in-depth insights.”. Also, U3 mentioned: “You can explore unexpected relations by switching between perspectives and deriving attributes [...] FlexEvent could give us more information than our current way of analyzing.”. Further, the users completed all the different tasks without problems. Moreover, users can gain more in-depth insights by looking at the data from different perspectives. As mentioned in Section 6, it is difficult to look at the data from different perspectives with current methods. This leads to less in-depth insights for hypotheses where multiple perspectives are needed. U1-3 used the automated methods to enrich the data to discover non-trivial relations successfully. Furthermore, visualizing the plots (FlexiPlot tiles) side-by-side helped users in tracking their thought process in analyzing the data for the current hypothesis. It was unexpected that U3 found several surprising findings because she was familiar with the data content beforehand. However, users also mentioned they need time to get used to FlexEvent, but they were confident they would learn it quickly. U3 also mentioned an unexpected goal for FlexEvent; the ability to help with disease predictions.

8. Discussion and Future Work

In this section we discuss limitations and future research directions. We aim for FlexEvent to be generic and not specifically tuned for one domain. This is reflected in the manner we deal with the data and the associated tasks. No element of the design is domain specific. These are valid for other domains with the same data structure and generalized tasks. There are several limitations to our study. First, ideally, we would monitor users using FlexEvent over a longer period of time to identify the added value of FlexEvent in their work environment. Other limitations to our method are, the use of homogeneous event sequence data; each event has the same set of attributes. It would be interesting to extend this to non-homogeneous event data. Second, visual scalability is a problem if there are thousands of sequences or if the sequence length is large. The width and height of a FlexiPlot tile (maximum of approximately 1000 pixels) determines the maximum length and number of sequences, respectively, that can be displayed in one tile. At least one pixel per event is needed. It would be interesting to investigate in future work how sequences can be aggregated to fit more and longer sequences on the screen. Also, distinguishable colors for categorical data is limited. Currently, we only color the most frequent categories if the color limit is exceeded. Fourth, if users do not have any idea yet of what they want to investigate, it can be hard to start. The frequent patterns and DR helps users to find multivariate frequent patterns. Finally, our visualization suffers from all short-comings that scatterplots also have, *e.g.,* overdraw.

There are several directions for future research. First, additional interactions are needed to make our solution more scalable. For example, by supporting more levels of granularity. Second, users indicated that the pattern mining gave some interesting patterns, but also missed some expected or manually spotted ones. It could be more specified to users' needs while analyzing complex multivariate relations, *e.g.*, focus on order and frequency of certain attributes instead of only order. Finally, further research is needed to support heterogeneous event sequences (*i.e.,* text, images, etc.)

9. Conclusions

This paper presents a novel, domain independent visualization and analysis method for multivariate event sequence data based on case-, event-, and attribute-centric perspectives. We offer users the flexibility to switch between and within perspectives using FlexiPlot tiles (based on scatterplots), in contrast to solely analyzing multivariate events from one fixed, predefined case-centric perspective. We identified user tasks from related work and interviews. Moreover, we described two use cases with real-world data to show the need and added value of our method. Overall, FlexEvent is a generalized method to analyze and explore the complex multivariate event sequence data different domains collect in abundance.

References

[BC06] Braun V., Clarke V.: Using thematic analysis in psychology. *Qualitative research in psychology 3*, 2 (2006), 77–101. 4

[BHS*13] Brewer C., Harrower M., Sheesley B., Woodruff A., Heyman D.: Colorbrewer 2.0 color advice for cartography, Oct. 2013. URL: https://colorbrewer2.org/. 5

[BSKR18] Bernard J., Sessler D., Kohlhammer J., Ruddle R. A.: Using dashboard networks to visualize multiple patient histories: a design study on post-operative prostate cancer. *IEEE Transactions on Visualization and Computer Graphics 25*, 3 (2018), 1615–1628. 1, 3, 8

[BVS16] Bertens R., Vreeken J., Siebes A.: Keeping it short and simple: Summarising complex event sequences with multivariate patterns. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pp. 735–744. 8

[CPYQ18] Chen Y., Puri A., Yuan L., Qu H.: Stagemap: Extracting and summarizing progression stages in event sequences. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), IEEE, pp. 975–981. 3

[CVW11] Claessen J. H., van Wijk J. J.: Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2310–2316. 6

[CvW17] Cappers B. C., van Wijk J. J.: Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 532–541. 1, 2, 3

[CXR17] Chen Y., Xu P., Ren L.: Sequence synopsis: Optimize visual summary of temporal event data. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 45–55. 1, 3

[DBZSD20] Di Bartolomeo S., Zhang Y., Sheng F., Dunne C.: Sequence braiding: Visual overviews of temporal event sequences and attributes. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 1353–1363. 3

[DSP*16] Du F., Shneiderman B., Plaisant C., Malik S., Perer A.: Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics 23*, 6 (2016), 1636–1649. 3

[Ent15] Enthought: A better default colormap for matplotlib | scipy 2015 | nathaniel smith and stéfan van der walt, Oct. 2015. URL: https://www.youtube.com/watch?v=xAoljeRJ3lU. 5

[FD81] Freedman D., Diaconis P.: On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 57*, 4 (1981), 453–476. 4, 7

[FKSS06] Fails J. A., Karlson A., Shahamat L., Shneiderman B.: A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *2006 IEEE Symposium On Visual Analytics Science And Technology* (2006), IEEE, pp. 167–174. 3

[FVGŠH14] Fournier-Viger P., Gomariz A., Šebek M., Hlosta M.: Vgen: fast vertical mining of sequential generator patterns. In *International conference on data warehousing and knowledge discovery* (2014), Springer, pp. 476–488. 8

[GGJ*21] Guo Y., Guo S., Jin Z., Kaul S., Gotz D., Cao N.: A survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics* (2021). 1, 2, 3

[GJC*19] Guo S., Jin Z., Chen Q., Gotz D., Zha H., Cao N.: Visual anomaly detection in event sequence data. In *2019 IEEE International Conference on Big Data (Big Data)* (2019), IEEE, pp. 1125–1130. 3

[GJG*18] Guo S., Jin Z., Gotz D., Du F., Zha H., Cao N.: Visual progression analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 417–426. 3

[GS14] Gotz D., Stavropoulos H.: Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1783–1792. 1, 2, 3

[GXZ*17] Guo S., Xu K., Zhao R., Gotz D., Zha H., Cao N.: Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 56–65. 3

[HLI*15] Huang C.-W., Lu R., Iqbal U., Lin S.-H., Nguyen P. A. A., Yang H.-C., Wang C.-F., Li J., Ma K.-L., Li Y.-C. J., et al.: A richly interactive exploratory data analysis and visualization tool using electronic medical records. *BMC medical informatics and decision making 15*, 1 (2015), 1–14. 3

[HPK*21] Han D., Parsad G., Kim H., Shim J., Kwon O.-S., Son K. A., Lee J., Cho I., Ko S.: Hisva: a visual analytics system for learning history. *IEEE Transactions on Visualization and Computer Graphics* (2021). 3

[JGC*20] Jin Z., Guo S., Chen N., Weiskopf D., Gotz D., Cao N.: Visual causality analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 1343–1352. 2, 3

[KPS15] Krause J., Perer A., Stavropoulos H.: Supporting iterative cohort construction with visual temporal queries. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 91–100. 2, 3

[LPK*15] Loorak M. H., Perin C., Kamal N., Hill M., Carpendale S.: Timespan: Using visualization to explore temporal multi-dimensional data of stroke patients. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 409–418. 3

[LS19] Linderman G. C., Steinerberger S.: Clustering with t-sne, provably. *SIAM Journal on Mathematics of Data Science 1*, 2 (2019), 313–332. 7

[LWD*16] Liu Z., Wang Y., Dontcheva M., Hoffman M., Walker S., Wilson A.: Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2016), 321–330. 3

[Mic23] Microsoft: Turn your data into immediate impact. website, 2023. https://powerbi.microsoft.com/en-au/. 3

[MSD*16] Malik S., Shneiderman B., Du F., Plaisant C., Bjarnadottir M.: High-volume hypothesis testing: Systematic exploration of event sequence comparisons. *ACM Transactions on Interactive Intelligent Systems (TiiS) 6*, 1 (2016), 1–23. 3

[MSM*21] Magallanes J., Stone T., Morris P. D., Mason S., Wood S., Villa-Uriol M.-C.: Sequen-c: A multilevel overview of temporal event sequences. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2021), 901–911. 3

[Mun14] Munzner T.: *Visualization analysis and design*. CRC press, 2014. 8

[MXC*19] Mu X., Xu K., Chen Q., Du F., Wang Y., Qu H.: Moocad: Visual analysis of anomalous learning activities in massive open online courses. In *EuroVis (Short Papers)* (2019), pp. 91–95. 3

[PTM*19] Pezzotti N., Thijssen J., Mordvintsev A., Höllt T., Van Lew B., Lelieveldt B. P., Eisemann E., Vilanova A.: Gpgpu linear complexity t-sne optimization. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 1172–1181. 8

[PW14] Perer A., Wang F.: Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th international conference on Intelligent User Interfaces* (2014), pp. 153–162. 3

[skl22] sklearn: sklearn.cluster.dbscan. website, 2022. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html. 8

[Tab23] Tableau: The world's leading analytics platform. website, 2023. https://www.tableau.com/. 3

[vDBCDW18] van Dongen B., Burattin A., Claes J., De Weerdt J.: 14th international workshop on business process intelligence 2018, Sept. 2018. URL: https://www.win.tue.nl/bpi/doku.php?id=2018:start#organizers. 4

[vdEvW13] VAN DEN ELZEN S., VAN WIJK J. J.: Small multiples, large singles: A new approach for visual data exploration. In *Proceedings of the 15th Eurographics Conference on Visualization* (2013), EuroVis '13, The Eurographs Association & John Wiley & Sons, Ltd., p. 191–200. 6

[VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 7

[Vis17] VISUAL ANALYTICS COMMUNITY: Vast challenge 2017. website, 2017. https://www.vacommunity.org/VAST+Challenge+2017. 4

[Vis18] VISUAL ANALYTICS COMMUNITY: Vast challenge 2018. website, 2018. https://www.vacommunity.org/VAST+Challenge+2018. 4

[WG12] WONGSUPHASAWAT K., GOTZ D.: Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2659–2668. 3

[WGGP*11] WONGSUPHASAWAT K., GUERRA GÓMEZ J. A., PLAISANT C., WANG T. D., TAIEB-MAIMON M., SHNEIDERMAN B.: Lifeflow: visualizing an overview of event sequences. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2011), pp. 1747–1756. 3

[WGW*20] WU J., GUO Z., WANG Z., XU Q., WU Y.: Visual analytics of multivariate event sequence data in racquet sports. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2020), IEEE, pp. 36–47. 3, 8

[WHA07] WILLETT W., HEER J., AGRAWALA M.: Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (nov 2007), 1129–1136. 3

[WLG*21] WU J., LIU D., GUO Z., XU Q., WU Y.: Tacticflow: Visual analytics of ever-changing tactics in racket sports. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2021), 835–845. 3, 8

[WLGW22] WU J., LIU D., GUO Z., WU Y.: Rasipam: Interactive pattern mining of multivariate event sequences in racket sports. *IEEE Transactions on Visualization and Computer Graphics 1* (2022). 8

[WLS*21] WANG Y., LIANG H., SHU X., WANG J., XU K., DENG Z., CAMPBELL C. D., CHEN B., WU Y., QU H.: Interactive visual exploration of longitudinal historical career mobility data. *IEEE Transactions on Visualization & Computer Graphics*, 01 (2021), 1–1. 3

[WMH*21] WANG Q., MAZOR T., HARBIG T. A., CERAMI E., GEHLENBORG N.: Threadstates: State-based visual analysis of disease progression. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2021), 238–247. 2, 3

[XSZX22] XU S., SUN M., ZHANG Z., XUE H.: Exploring multivariate event sequences with an interactive similarity builder. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 271–282. 3

[YM22] YESHCHENKO A., MENDLING J.: A survey of approaches for event sequence analysis and visualization using the esevis framework. *arXiv preprint arXiv:2202.07941* (2022). 3

[ZDFD15] ZGRAGGEN E., DRUCKER S. M., FISHER D., DELINE R.: (sl qu) eries: Visual regular expressions for querying and exploring event sequences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 2683–2692. 3