UNIVERSITY OF MANNHEIM

VISUAL ANALYTICS APPROACHES FOR SEQUENTIAL DATA

OVERVIEW AND PROCESS MINING SUITABILITY

Bachelor Seminar Thesis

Submitted: 08.11.2023

By: Philemon Schulz
born on August 27, 2003
in Freiburg im Breisgau, Germany

Matriculation number: 1835051

Course of Study: Wirtschaftsinformatik B.Sc.

Supervisor: Michael Grohs
Jun.-Prof. Dr. Jana-Rebecca Rehse

University of Mannheim
Assistant Professorship of Management Analytics
L15 1-6, 68161 Mannheim
Homepage: https://www.bwl.uni-mannheim.de/rehse/

Abstract. Event logs are generated in almost every domain these days. Process mining uses this type of sequential data to analyze and improve processes. Here, appropriate visualization plays a crucial role. This paper aims to identify visual analytics approaches that focus on the visualization of sequential data and to evaluate these tools concerning their application for process mining. We identify a total of 21 approaches, which we categorize and describe in detail using specially developed coding.

Keywords: Visual Analytics · Sequential Data · Process Mining

1 Introduction

Process mining is a relatively new research technique, which works with massive amounts of sequential data, namely event logs, to discover, monitor and improve real processes [1]. Based on mining algorithms and visualization methods, process models can be produced, checked against real-world logs, and improved.

Needed sequential data is collected in all different kinds of domains [8] [9]. Healthcare providers document every treatment step for a patient to enhance underlying service processes; manufacturers keep track of every component throughout the production process to speed up production time and reduce misproduction; and marketing teams collect clickstream data of website users to improve the website interactions and appearance. As it is nearly impossible to analyze this amount of data manually [23], supporting tools are needed. Visual Analytics tools allow users to analyze data automatized while visualizing the results and offering the possibility to explore the data [23] interactively. They enable analysts to explore datasets, find similarities, compare patterns or give recommendations based on founded sequences.

Currently, in process mining, visualization is usually less important than the underlying mining methods [17]. However, as van der Aalst et al. [1] state in their process mining manifesto, interactive visualization and process mining help users better understand unstructured data and extract valuable patterns. Hence, new and innovative visual analytics approaches suitable for process mining are needed.

Therefore, this paper aims to overview current state-of-the-art approaches in Visual Analytics for sequential data like process data. We conduct an informal literature review where approaches are found in scientific articles, conference papers and books. We give an overview of different approaches, point out differences between the visualization of sequential and non-sequential data and evaluate whether they are suitable for process mining.

Hence, our research question is as follows: What are current state-of-the-art approaches in visual analytics that focus specifically on sequential data, and how are they suitable for process mining? (RQ1)

The structure of the paper is as follows: First, in sections 2 and 3, important concepts and terms will be explained, and related work will be presented. In section 4, we will present our coding and results, following a discussion and conclusion in section 5.

2 Preliminaries

This section introduces the main concepts and terms used frequently throughout this paper. Namely, those are "Sequential Data", "Process Mining", and "Visual Analytics".

Sequential Data "is any kind of data where the order matters" [39]. It appears in many different types and forms like meteorological data, time-series data, gene sequences, sensor data, and audio clips [42]. Our work primarily focuses on process data as a subform of sequential data.

Process data consists of "time-ordered sequence[s] of events" [5]. On this occasion, events are typically structured as a tuple consisting of action, context and time, where context can be optional. Process data can be inherently discrete (e.g. sequences of web-click events, treatment steps in healthcare) or continuous (e.g. cursor positions) [5].

Process Mining is a relatively young research discipline that sits between computational intelligence and data mining on the one hand and process modelling and analysis on the other hand. Its idea is to discover, monitor and improve real processes by extracting knowledge from event logs readily available in today's information systems [1]. Event logs, as one form of sequential data, are "unbiased footprints representing the process as it is" [35]. They are usually recorded by software systems and show the execution of an activity and associated information. Each activity is assigned to a particular case, representing the concrete process instance in which it occurred [27]. Further, event logs are the starting point of all process mining techniques, which can be categorized into three classes depending on their purpose: (i.) discovery, (ii.) conformance and (iii.) enhancement [35]. Discovery techniques produce process models based on the event logs. Conformance models take a process model, and the event log to check if the modeled process is represented in the actual event log. Lastly, Enhancement techniques use a process model and event logs to improve the model with the information contained in the log.

Visual Analytics tools combine "automated analysis techniques with interactive visualizations" [25]. They aim for a better understanding, reasoning and decision-making based on large and complex data sets. The focus lies on a collaborative process between human and machine [23]. Users give directions for the analysis based on their tasks or interests, while the system offers options to interact with the system meaningfully to concentrate the analysis process [25].

The challenge lies in (i.) identifying the best automated algorithm for the analysis task, (ii.) identification of limits, and (iii.) developing an appropriate solution that "integrates the best automated analysis algorithms with appropriate visualization and interaction techniques." [25]. As visualizations are essential for a visual analytics tool, Munzer [36] introduced the visual analytics framework 1 to systemize knowledge abstraction through visualizations. It splits a visualization instance into three aspects: "What?", "Why?" and "How?". "What?" refers to the data that is visualized. "Why?" refers to the task a users aims to solve with the visualization. "How?" describes the visualizing elements that are used to display the data.

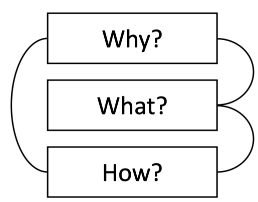

Fig. 1: Visual Analytics Framework [36]

3 Related Work

We noticed four articles that deal with similar questions as we do. The first two [17] [40] give insights into the interplay between visual analytics and process mining. The last two [45] [21] are surveys comparing visual analytics systems focusing on sequential data.

First, Geschwandtner [17] elaborated on challenges and opportunities for combining visual analytics and process mining. There, they introduced three novel visualization approaches regarding guideline conformance, process hierarchy visualization and data exploration. Further, she identified six challenges regarding the combination of process mining and visual analytics: (i.) intertwining process mining with visual analytics, (ii.) scalability and aggregation, (iii.) interaction to support process discovery and enhancement, (iv.) data quality and uncertainty, (v.) complexity of time-oriented data, and (vi.) evaluation.

Second, Sitova et al. [40] presented nine different process metrics and accordingly visualizations. A specific way of visualization was found for every metric, and the associated level of analysis and insights were described. They also proposed a system in which process mining and visual analytics can interplay reclusively.

Zinat et al. [45] made a study about visual summarization techniques for event sequences. They surveyed 14 papers that proposed new approaches and chose 3 for detailed comparison. A description of the selection criteria can be found in [45]. However, their research mainly focused on the underlying summarizing algorithms rather than the different visualization methods.

Finally, Guo et al. [21] surveyed visual analytics of event sequence data. They reviewed 30 visual analytics approaches connected to sequential data and categorized them based on analytical tasks and applications. To achieve this, they introduced four dimensions on which they evaluated each approach: 1. Data Scales, 2. Analysis Techniques, 3. Visual Representations, and 4. Interactions.

While they give a 'comprehensive attempt to survey and characterize event sequence data visual analytics methods' [21], they focus on sequential data in all its different forms and applicable domains. However, our research is interested in finding approaches especially suitable for modelling process data based on event logs.

4 Results

4.1 Qualitative Analysis of Contributions

In order to find the state-of-the-art approaches in Visual Analytics for sequential data that are applicable for process mining (RQ1), we analyse the founded approaches based on the Visual Analytics Framework presented by Munzner [36]. It identifies three main aspects on which Visual Analytics approaches can be abstracted: "Why?", "What?" and "How?". We adopted this idea for our coding and adjusted it for our needs. Therefore, we are categorising approaches based on:

1. Analytical goal (Why?)
2. Type of data (What?)
3. Analysis technique (How?)
4. Visualisation elements (How?)

We also took inspiration from [21] regarding their categorisation for Data Scales and Visualization Representations.

We didn't define our coding in advance before the start of our research. Instead, through our analysis process, we collected and aggregated the methods and tools found in the literature based on the Visual Analytics Framework. Tables 1 and 2 show our coding results.

VA Appraoches of Sequential Data			[14]	[31]	[12]	[28]	[26]	[9]	[8]	[16]	[20]	[44]	[37]
Name			DecisionFlow	CoreFlow	EventAction	MAQUI	PeekQuence	Sequence Synopsis	StageMap	Cadence	EventThread	iHPPVis	CareFlow
Tyoe of data (What?)		Event Sequences			✓		✓	✓	✓			✓	✓
		Spatio-Temporal Data											
		High Dimensional Temporal Sequence Data	✓	✓		✓				✓	✓		
Analytical Goal (Why?)		Event Summarization		✓		✓	✓	✓	✓	✓	✓		✓
		Pattern Comparasion	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
		Action Recommendation			✓								✓
		Process Performance Analysis											
How?		Interactive	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Analysis Technique (How?)	Queries	✓			✓			✓	✓	✓		✓
		Pattern Mining	✓	✓		✓	✓	✓				✓	
		Similarity Measures			✓								✓
		Filter	✓			✓	✓	✓	✓		✓	✓	
		Stage based Aggregation							✓	✓	✓		
		Process mining											
	Visualisation elements (How?)	Hirarchy graph	✓	✓		✓		✓	✓	✓	✓		✓
		Matrix			✓						✓		
		Mareys graph										✓	
		Glyph Based					✓	✓	✓		✓		
		Tooltip	✓	✓	✓			✓	✓	✓	✓		
		Timeline			✓	✓	✓					✓	
		Charts	✓		✓		✓	✓	✓	✓		✓	✓

Table 1

VA Appraoches of Sequential Data			[34]	[40]	[30]	[10]	[43]	[22]	[7]	[32]	[6]	[19]
Name					TPFlow		ViDX		SepVis			EventThread2
Tyoe of data (What?)		Event Sequences				✓			✓	✓	✓	
		Spatio-Temporal Data			✓							
		High Dimensional Temporal Sequence Data					✓	✓				✓
Analytical Goal (Why?)		Event Summarization			✓		✓			✓		✓
		Pattern Comparison			✓		✓	✓	✓	✓		✓
		Action Recommendation										
		Process Performance Analysis				✓			✓	✓		
How?		Interactive			✓	✓	✓	✓	✓	✓	✓	✓
	Analysis Technique (How?)	Queries										✓
		Pattern Mining				✓		✓				
		Similarity Measures										
		Filter			✓	✓	✓	✓	✓	✓	✓	✓
		Stage based Aggregation			✓							✓
		Process mining							✓	✓	✓	
	Visualisation elements (How?)	Hirarchy graph			✓					✓		✓
		Matrix					✓					✓
		Mareys graph				✓	✓					
		Glyph Based							✓			✓
		Tooltip					✓		✓			✓
		Timeline				✓	✓		✓			
		Charts	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 2

There are a few interesting findings and trends to observe already. First, deriving from the research question, it only makes sense that there are fewer categories of the type of data, as there are regarding the analytical goal and especially the actual implementation (How?).

Next, we can clearly see that for the "Why" Area, there is a clear focus on event summarization (12 of 21) and pattern comparison (15 of 21). For analytical tasks, getting an overview of complex datasets and comparing different variants, patterns, or clusters seems essential.

Almost all approaches allow for interactive interactions except two. That makes sense, as visual analytics exactly corresponds with interactive systems. However, we still wanted to include those two approaches because [34] presents a novel algorithm to visualize process maps, a common way of visualizing processes in other visual analytics tools, and [40] gives an overview of different visualization techniques regarding sequential data and process mining.

Regarding the analysis technique (How?), we can observe that many approaches use filters and queries. Lastly, tooltips (11 of 21) 2, hierarchies-based graphs (11 of 21) and additional charts (18 of 21) are among the most used visualization techniques.

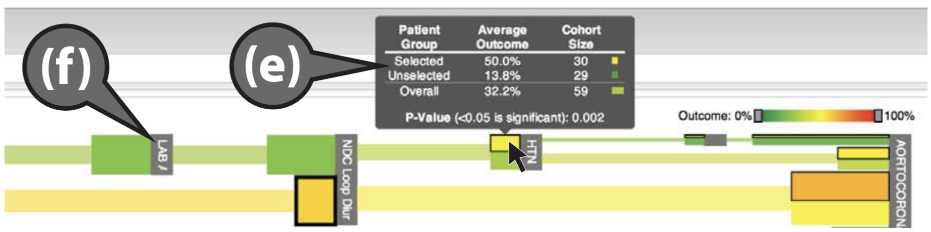

Fig. 2: Tooltip used in DecisionFlow [14]

In the next sections, we will explain our findings in great detail. We will guide it along the four categories mentioned above.

4.2 Analysis of coding results - Type of data (What?)

Type of data. Due to the specific research question, it is clear that all approaches cope with sequential data in some kind. However, sequential data varies enormously in its complexity regarding unique events or event types as well as sequence length and additional attributes. For example, LifeFlow [41], which uses colours to distinguish between event types, is demonstrated with a dataset containing only six unique event types. In comparison, CoreFlow [31] is tested on datasets with over 100,000 unique event types and over 5 million different events. Hence, while some approaches don't specifically name their data scope (e.g. PeekQuence [26], EventAction [12]), there are contributions that specialise in high-dimensional sequential data (e.g. DecisionFlow [14], MAQUI [28], CoreFlow [31]). High-dimensional sequential data refers to datasets with high event cardinality and long event sequences [31]. We found that approaches that directly address this data type almost always focus on data exploration as one of their main tasks. DecisionFlow [14], for example, helps healthcare analysts explore a given dataset to find factors that may influence a particular outcome. MAQUI introduces an analysis framework to recursive explore datasets and "modify the analytic focus and context" [28] and CoreFlow [31] aims to help analysts select effective event groupings for further analysis.

Lastly, TPFlow [30] uses spatio-temporal data for its analysis. Spatio-temporal data refers to multi-dimensional data defined by temporal, spatial and other domain-specific dimensions [30]. Spatio-temporal data plays an integral part in sequential data analysis [2] [18] [4] [38] [11] [29] [24]; however, most approaches don't allow for an application in a business process-focused area. Here, TPFlow [30] acts as a special case. Using spatio-temporal data, TPFlow allows analysts to, e.g., use sensor data of a shopping area, helping to conclude user movement behaviours through different days and times, allowing for a direct improvement of the shopping process.

4.3 Analysis of coding results - Analytical Goal (Why?)

Analytical goal (Why?) When it comes to the analytical goal of the proposed approaches, all approaches help to allow for specific data exploration (except for [34] [40]), which are only visualization advancements). In the course of this, proposed approaches mainly try to satisfy at least one of the following tasks:

Event Summarization. Event summarization is one of the major tasks the proposed visual analytics systems aim to solve. As its goal is to find meaningful patterns and groupings of the sequence entities, it helps analysts to get an overview of the data quickly. As a result of this, the different proposed approaches differ heavily regarding at which detail they try to summarize the given data. Depending on size and complexity, different approaches aim at different summarization goals. CoreFlow [31] seeks to provide a high-level overview of paths and volumes of flows, while Sequence Synopsis [9] tries to give only a coarse overview of the data while balancing between information loss and visual clutter. Further, StageMap [8] tries to provide an utterly lossless visualization.

To summarize effectively, there are various analytical methods at hand. For example, MAQUI [28], StageMap [8] and PeekQuence [26] use sequential pattern mining to mine the most frequent patterns. Sequence Synopsis [9] uses a minimum description language to create a two-part representation of the sequences, and Cadence [16] and EventThread [20] implement stage-based aggregation algorithms.

Pattern Comparison. Pattern comparison is also an important task when it comes to analyzing data. By directly showing different sorts of patterns or process flows, outliers and other abnormal situations are easy to identify. However, approaches differ regarding their way of visualizing comparison and their focus on what to compare. DecisionFlow [14], Cadence [16], SepVis [7] enable comparison through different statistics in bar charts or scatter plots. Herr et al. [22] allows comparison of the event's temporal distribution through a bar chart as well. MAQUI [28] and Maier et al.[32] allow comparison by vertically showing different clusters or panels depending on selected filters. EventAction

[12], Peekquence [26], Sequence Synopsis [9], EventThread 1 and 2 [20] [19] and SepVis [7] focus on different patterns and stages that were minded, which are then vertically ordered. Lastly, ViDX [43], iHPPVis [44] and Denisov et al. [10] focus on showing the whole process and its stations over time. Here, all three use Marey's graphs to enable a quick comparison of every process's time steps and station.

Some approaches extend the comparison task to dedicated similarities finding. Similarities finding, in contrast to comparison, goes a step further and not only allows comparison e.g. for two different mined patterns, but for specifically searching and presenting e.g. similar clusters or patterns. Finding similar patterns or records in huge sequential datasets can help analysts identify and improve current processes (e.g. [44] [20]) or give a foundation to recommend further steps like in EventAction [12] or CareFlow [31].

Action recommendation. Action recommendation tasks aim at giving predictions and recommendations based on event sequences. Two approaches tackle this task. First, EventAction [12] uses event sequences from an educational dataset to analyze the current progression state of a student by showing similar students and their results and giving recommendations to help the student achieve its desired output. It further allows for generating personalized action plans and provides feedback on the probability of success. Second, CareFlow [37] allows doctors to find similar patients to get a better overview of possible care plans. Using electronic medical records, different care plans are presented while context about success rates enables the doctor to pick the best-suited care plan.

Process performance analysis. As performance analysis is an important element in process management and process mining [33], some approaches try to visualize process performance to enable improvements. Denisov et al. [10] present a Marey's graph based approach to "investigate process performance of all process behaviors without prior aggregation" [10]. SepVis [7] uses process mining techniques to get accurate conformance and performance information pertaining to a patient's care pathway. Lastly, Maier et al. [32] use process models to explore the relation of cause and effect based on sensor data.

4.4 Analysis of coding results - Analysis Technique (How?)

Analysis technique (How?) categorizes the approaches based on how they analyze the underlying data. It contains methods of data preprocessing as well as ways of interactively analyzing the data through user interactions.

Queries. Queries allow analysts to "make domain-specific data adjustment or selection based on certain conditions, so as to eliminate noisy and irrelevant data for better analytical performance" [21]. Queries are used both for initial data preprocessing [14] [28] [16] as well as drilling down to specify the visualized data even more [8] [20]. In Cadence [16], a query is used to initially select a data subset of a given dataset to ease analysis. However, while a query would allow finding sequences related to a concrete question (e.g., in Cadence, analysts would start querying the data for patients discharged from a hospital after being diagnosed with pain), it risks missing unexpected patterns in the data [28]. Therefore, MAQUI and DecisonFlow use both queries and pattern mining to overcome this issue. Here, DecisonFlow [14] uses queries with milestones, meaning that users can specify an ordered list of event types as query constraints. In a modified version [15], a pattern mining algorithm then extracts respective patterns based on the found matching sequences and visualizes them. MAQUI [28] improves this querying → mining → visualizing pipeline to allow this process in a recursive fashion. StageMap [8], EventThread [20] and EventThread2 [19] use queries only for digging down on already processed data, using queries as a way of filtering.

Pattern Mining. Finding patterns is one of the most common tasks the found approaches tackle. Here, Pattern mining plays an important role. Starting with a mining algorithm, patterns are mined either fully automatic as a completely linear process or semi-automatic, meaning the system provides mechanisms for interacting with the mining algorithm [28]. There are different mining algorithms and metrics for approaches to choose from:

Regarding fully automated mining algorithms, SPAM and VMSP are among the most common pattern mining algorithms for event sequences. SPAM uses a bitmap representation for efficiently mining frequently occurring patterns. Generally, a "frequent" pattern refers to a pattern mined with a minimum support value of x, which means that the mined patterns should appear in at least x percent of the sequences [15]. For a detailed description of the SPAM algorithm, please refer to [3].

In a modified version of DecisionFlow [15], SPAM is used as its core mining algorithm. The system provides options to configure the minimum support value and the minimum pattern length. However, as SPAM tends to produce large numbers of patterns, it can be hard to visualize and compare them in a reasonable fashion [28]. Here, PeekQuence [26] tries to create a remedy by proposing a four-view interactive visualization system to make SPAM results more interpretable.

Following up with semi-automatic mining processes, CoreFlow [31] introduces a novel Rank-Divide-Trim process to extract branching patterns from event sequences interactively. First, events are ranked based on a frequency-based function. Then, sequences are separated into two groups, and finally, subsequences leading to top-ranked sequences are removed. Sequence Synopsis [9] uses the Minimum Description Length (MDL) principle to summarize sequential patterns with a balance between informative visualization and information loss. The MDL principle trades off between 1) the complexity of the model and 2) the description length of the original data with the help of the model. It helps to create a two-part representation of the data consisting of a set of sequential patterns and a set of corrections. Denisov et al. [10] introduce their own algorithm to extract performance patterns from event logs without enforcing any prior aggregation.

Contrary to PeekQuence or DecisionFlow, which use SPAM as their pattern mining algorithm, MAQUI [28] uses the VMSP algorithm [13], which achieves to generate more compact patterns to reduce visual clutter. Further, MAQUI combines VMSP with an interactive querying system to allow for a recursive exploration of the data. Herr et al.[22] introduce a system which uses filtering paired with outlier detection and seasonal trend decomposition. iHPPVis [44] uses Apache's Hadoops2 and the Maximal Information Coefficient to extract its feature variables for the heavy plate production data.

Similarity Measures. Similarity Measures are used when systems aim to find similar sequences given an input sequence. EventAction [12] uses the Euclidean Distance of extracted feature vectors from students' academic history sequences to measure similarity. Then, a similarity score between the current student and each achieved student is calculated, and the distribution is visualized. CareFlow [37] uses its own developed patient similarity analytics system to find similar patients in electronic medical data. It then uses queries for extracting records of performed treatments and associated dates from those.

Filters. Filters are an essential feature to explore big datasets interactively. It allows the inclusion or exclusion of events in datasets as well as dynamically manipulating different views or producing extra views. We found two categories of filters that are used throughout the approaches.

First, filters that manipulate the data: Here, filtering directly influences the mined data, leading to calculating a new subset of data. EventThread [20] allows filtering for event types. EventThread2 [19] extends this filtering by introducing filtering of entity groups. Herr et al. [22] allow selecting event types, product types or process steps, and Maier et al.[32] implements a filter system for cluster affiliation or specific sensor data. DecisionFlow [14] allows to include or exclude both certain milestones or event types for its data analysis. Finally, MAQUI [28] provides filtering for all kinds of record attributes 3. Categorical ones work just as boolean values, while you have to choose a range when the attribute is numerical.

Second, filters that only manipulate the visualization, but not the underlying dataset. In PeekQuence [26], for example, clicking on a node or edge in the sequence network view filters the pattern list view, which results in highlighted areas. TPFlow [**?**] allows for selecting nodes to change statistics views, StageMap [8] shows users a more detailed overview when clicking a node in the treeview, in VIDX [43] you can choose a set of records or a timespan, and in Bodesinsky et al. [6] already mined patterns can be selected.

There are also approaches that combine both categories. With Sequence Synopsis [9], users can filter on events or attributes, resulting in a recalculation of the relevant data (data manipulation). It further allows for linked-highlighting to help users associate information (visualization manipulation). iHPPVis [44] uses a filtering technique to sort out anomalous data in the data preprocessing step. Second, it provides the user with multiple filters to select different mining algorithms (data manipulation) or switch focus between different processes (visualization manipulation). Lastly, SepVis [7] provides a similar system where users can filter for different activities, like treatments or labs (data manipulation), and click different traces to get highlighted information (visualization manipulation).

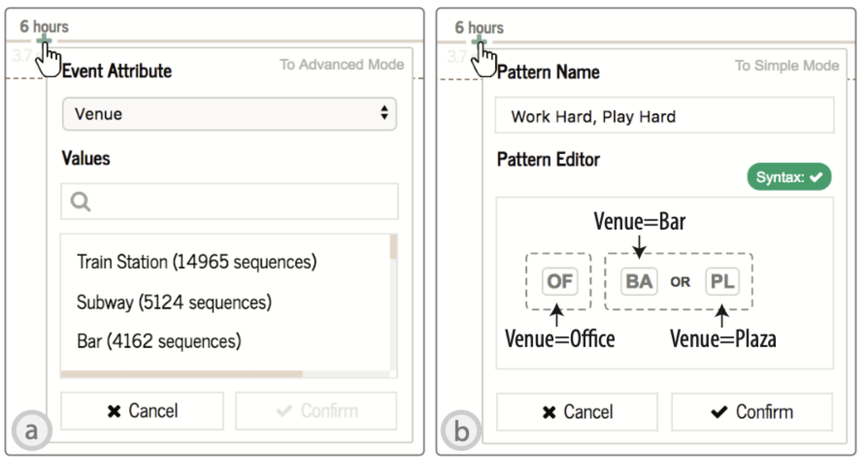

Fig. 3: Filtering interface in MAQUI [28]

Stage-based aggregation. Stage-based aggregation aims at "finding stages and their progression patterns in the dataset" [8]. A stage here is a grouping of events, and a progression pattern of a stage is a sequence of stages. Because a stage is a subsequence of events, it is "more concise than the event-based summary" [8].

StageMap [8] uses its algorithm SPTree to generate a set of possible stages. This algorithm consists of the subroutines: 1.) extracting stages, 2.) computing covers, and 3.) building trees. After SPTree finds a set of possible stages, VMSP [13] is used to identify subsequences with a higher frequency. EventThread [20] uses Tensor Decomposition for sequence clustering. Here, a tensor refers to a "multidimensional array, which extends the concepts of scalars, vectors, and matrices to higher dimensions" [20]. To create a tensor, they use a four-step process: First, events in the sequences are filtered to reduce noise. Then, sequences are aligned based on event occurrence order. Next, event sequences are segmented into stages. Finally, the stages are transformed into a tensor. Stage segmentation is based on either fixed time intervals or the number of major events. However, this system is fixed regarding the stage segmentation. To overcome this fixed-stage segmentation, EventThread2 [19] uses the Content Vector Segmentation algorithm to allow for a more flexible segmentation. For a detailed description of the algorithm, please visit [19]. Lastly, similar to stages, Cadence [16] introduces three a three-part algorithm to determine the most informative set of event groupings interactively.

Process Mining. Some approaches already use process mining techniques for their data analysis. [6] claims to use automatic pattern search and mining for interactive exploration. However, they do not specify which methods they use for searching and mining the patterns. Maier et al. [32] use automated and interactive grouping to find candidate sets sharing properties relevant for cause and effect analysis. For this, they use a process model, a directed acyclic graph and a feature vector together with an 'apriori' algorithm. Lastly, Caballero et al. [7] use the alignment-based conformance analysis technique to calculate conformance and performance information for healthcare processes. Instead of using "traditional sequence alignment techniques" [7], they provide visualization for patient flow activities in accordance with a clinical model.

4.5 Analysis of coding results - Visualization (How?)

Visualization (How?) categorizes the approaches based on different visual elements to present the sequential data.

Hierarchy-based visualization. Hierarchy-based visualizations are one of the most commonly used visualization techniques among found approaches. They use, e.g. node-link trees or icicle plots to provide aggregated views for event sequences and always act as the central visualization element.

Starting with Maier et al.[32], they use a classical Sankey diagram to visualize the flow of a process 8. Here, bars represent steps in the business process and are colour-coded regarding the average duration. Next, in Sequence Synopsis [9], different patterns are shown vertically, while a pattern itself is presented horizontally 10. The horizontal axis here represents the order of occurrence of specific events. In a pattern, events are presented as rectangles, whose height provides information about the number of sequences containing this event. Triangles are placed between two rectangles to represent the needed corrections. The size of a triangle gives information about the number of insertions required. TPFlow [30] uses a horizontal node-link visualization for presenting subsets of data. Nodes are visualized as partitioned rectangles, which are colour-coded based on the deviation rate. The first node represents the original data, while every other node on the tree represents a subset of the data. Edges give further information about containing attribute values like product types.

DecisionFlow [14], Cadence [16], and CareFlow [37] use the same horizontal node-link visualization for showing event flows. A grey (white in CareFlow, dark green in Cadence) rectangle represents a particular event or milestone. Two milestone rectangles are connected with edges of two types: First, a time edge directly next to the event rectangle, with the same height as the event rectangle. The height here gives information on the proportion of events in this node. The width corresponds to the average duration in this node. Second, link edges convey connectivity between graphical elements. The horizontal line responds to the occurrence of the nodes over time 9. All three approaches further use colour coding to indicate outcome probability. Further, in DecisionFlow, clicking on a time edge yields a tooltip for additional information. MAQUI [28] also uses a horizontal node-link visualization 11. Similar to the three approaches mentioned above, nodes are visualized as rectangles connected by rectangle edges. However, contrary to, e.g. DecisionFlow, a node rectangle can not only represent a specific event but also a defined pattern. Further, it allows for different alignment techniques regarding the width and height of edges and nodes 12. In the original view, the heights of nodes and distances between nodes are distorted for a better initial overview. However, it is possible to change alignments in both directions. First, distances between nodes can be adjusted to represent the duration between two milestone events. Second, the height of nodes can be rearranged to correspond to the number of sequences containing the milestone event. Third, the distance between nodes can also be used to show the average number of events during two milestones. Again, the horizontal axis corresponds to the order of occurrence.

In CoreFlow [31], both a vertical node-link visualization and an icicle plot are used. Here, the vertical axis can refer to the number of events or the average time since the beginning of sequences. The icicle plot starts with a full-width rectangle representing all events. It then gets partitioned, resulting in rectangles of different widths and heights representing the branching patterns. Each rectangle refers to an event of a pattern, where the width goes with the number of sequences containing this event. A sequential colour schema is used to indicate the level of a node. Exit branches are marked in grey. Further, they use a node-link visualization, which uses the layout of the icicle plot to more explicitly show the links between events. Here, classically, the width of an edge refers to the number of sequences. However, as this view makes the comparison more complex, they also proposed a hybrid view consisting of both the icicle and the node-link visualization. Users can switch between all three views 13. Similar to CoreFlow, StageMap [8] uses horizontal node-link and icicle visualizations 14. However, those are both always shown and not meant to be switched. Multiple icicle plots are displayed in rectangles, showing each progression pattern's structure. The area of each rectangle encodes the number of sequences matching this pattern. Inside the icicle plot, each rectangle represents a stage, while the height corresponds to the stage's frequency. When clicking on a certain icicle plot, a node-link visualization of this pattern is shown to provide the user with more details. Here, events are labelled and colour-coded based on the event type.

Lastly, EventThread [20] and EventThread2 [19] introduce a horizontal thread view approach to visualize threads, stages, events and entities 15. Threads (an event sequence cluster) are shown as vertically ordered lines. Threads are then segmented into stages, where events with a high likelihood of occurrence are visualized as nodes. Vertically, threads can be clustered into latent stage categories, where labels in each segment and thread explain the most likely event. Finally, each thread is underlined with a grey background, whose height indicates the number of entities in the thread. EventThread2 [19] further introduces a second cluster view, which again uses a horizontal node-link visualization. Clusters are represented as nodes, where each node "illustrates the frequencies of starting and ending events within a cluster as well as the frequent event patterns found within each cluster" [19].

Matrix. Matrix views help "to demonstrate a summary of event frequency or frequent patterns" [21]. In EventAction [12], a matrix view acts as the main visualization element to show the frequency of certain events ordered by timestamps 16. The size of a rectangle here refers to the amount of events that include the element. Further, colour coding is used to determine the outcome probability. In ViDX [43] and EventThread [20], matrix views are used to give additional information. ViDX uses a matrix-calendar view, which colour-codes the frequency of faults or products. It allows users to select a timespan to investigate the data. EventThread [20] uses an overview matrix with colour coding to show the distribution of occurring threads. EventThread2 [19] uses the same concept for a stage transition view, where multiple matrices are connected as a horizontal tree to show the overview changes in the different stages.

Marey's graph. Three approaches use Marey's graphs as their main visualization method 17. Marey's graphs help to show the process course over time for multiple process instances. Here, the focus mainly lies on detecting abnormalities over time to find process faults. In iHPPVis [44], the time lies on the y-axis, while process steps are presented along the x-axis. ViDX [43] and [10] axis are swapped. Further, ViDX uses coloured lines to highlight time abnormalities in the production process. While there are only a few crossings in iHPPVis and ViDX to reduce visual clutter, [10] allows for lots of crossings to get better insights into specific performance patterns. A taxonomy of elementary patterns is introduced (see appendix 18) which differs between Scope, Shape, Performance and Workload. For a detailed description, please visit [10].

Glyphs. Glyph-based visualizations are commonly used to visualize event types. Except for SepVis [7], glyphs are visualized as a circle with a colour corresponding to its event type 4 (SepVis uses different icons 19). In PeekQuence [26], Sequence Synopsis [9], SepVis [7], StageMap [8], they are used in summary views to show events of each pattern over time. In EventThread [20] and EventThread2 [19], glyphs are shown in each stage and thread to visualize the most common events. Glyphs can vary in size to give information about matching events in a pattern [9] [8] or be split into multiple sections like a pie to represent multiple events occurring at the same time [26].

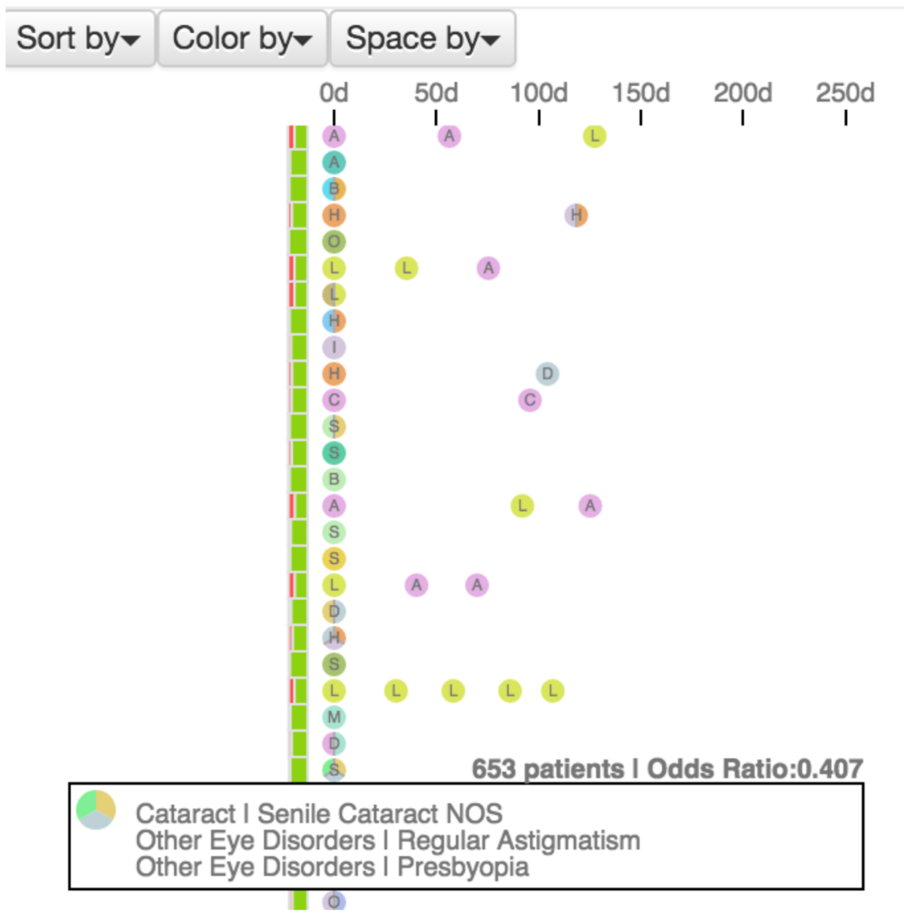

Fig. 4: Glyphs used in PeekQuence to display patterns [26]

Tooltips. Tooltips are a powerful tool to provide additional information for an analyst. Therefore, they are frequently used among found approaches. Tooltips are used to provide information about, e.g. event names, descriptions, outcome probabilities or event frequencies. Some approaches also use tables [14] [12] or bar charts [31] inside the tooltip to further visualize the information.

Timeline. Due to the importance of time in sequential data, timelines help to put information about e.g. patterns, events or stages into a time perspective. There are simple timelines like in MAQUI [28], which are just lines between two events or more advanced ones like in iHPPVis [44] or ViDX [43] where the timeline is combined with histograms to give information about e.g. production volume or fault rate as well. Timelines can be continuous like in EventAction [12] 5 or distorted like in SepVis [7].

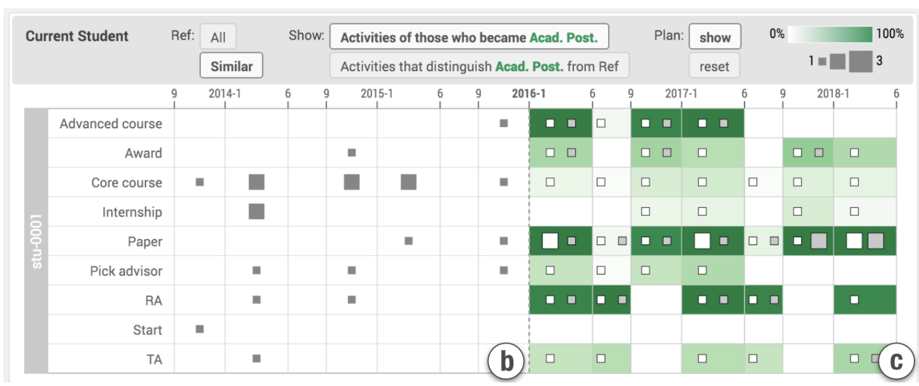

Fig. 5: Timeline used in EventAction [12]

Graphs. Besides the main visualization methods like hierarchy-based graphs or Marey's graphs, different basic chart types are used to display additional information on attributes that are connected with the sequential data.

Starting with bubble or scatter plots, they are mostly used to show events and provide an overview. In DecisionFlow [14], for example, it allows users to explore the larger event space that can not be visualized through the flow panel. It also allows for statistical comparison as events are colour-coded and differ in size to reflect correlation and odds ratio. Similar implementations can be seen in Cadence [16], iHPPVis [44], or citeSP20. In Sequence Synopsis [9], a radial scatter plot is used as an event filter, which shows the co-occurrence of events with a focus event at the centre 6.

Bar charts, column charts and histograms are other commonly used visualization methods. In DecisionFlow [14], Cadence [16] and SepVis [7], they show

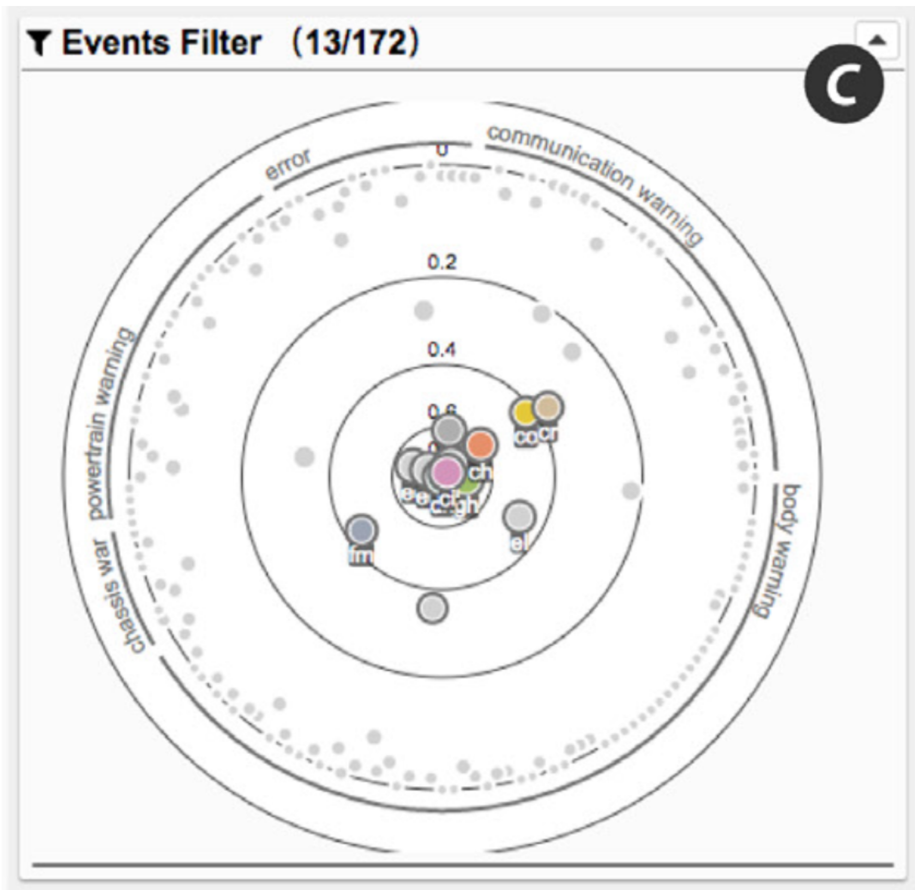

Fig. 6: Scatter plot based event filter in Sequence Synopsis [9]

the age distribution of patients; in PeekQuence [26], a histogram shows the frequency of events co-occurring given a selected pattern; EventAction [12] uses a histogram to show a similarity distribution for different students and in [22] histograms are used to show the distribution of event types.

Next, process maps are used to "discover the real production process" [40]. In ViDX [43], a modified process map is used to show the assembly line schema and represent the different production steps for the Marey's graph. Further, as classical process maps tend to represent the actual process poorly, [34] propose a novel graph layout algorithm to show the underlying process in an insightful way.

There are a lot of other different graphs used in different approaches. Line charts are commonly used to display trends [44] [22], patterns [30] or correlations [12]. Tabular views are used to show information like sequences [9] (glyph-based) or clusters [32] (text-based). Instance charts can give insight into events feature markedness over time, like workers' load in manufacturing [40]. Lastly, radial graphs in their simplest form are used to give insight into event distributions [40] or, in a modified way, encode multiple attributes at once like in ViDX [43] where all currently ongoing processes are visualized in one radial plot 7.

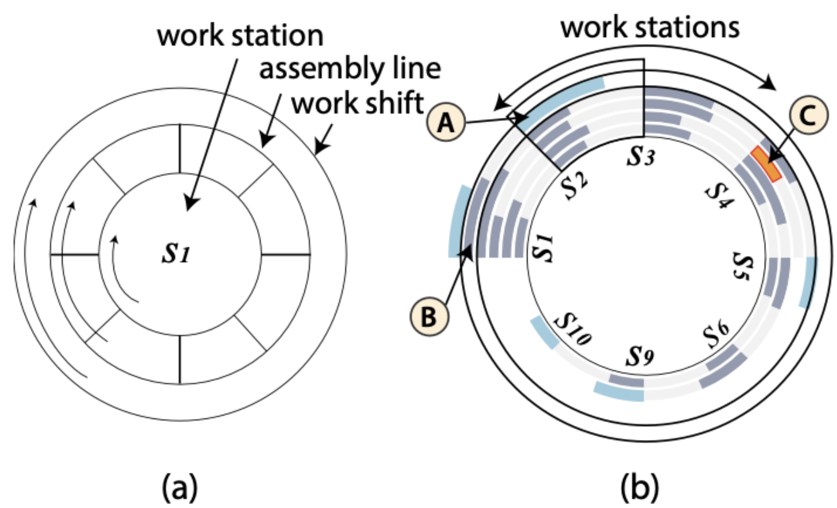

Fig. 7: Radial graph concept a) and implementation b) in ViDX. For a detailed description please visit [43]

4.6 Differences to non-sequential data

Due to time's importance in sequential data, we found that many approaches use visualizations with a time axis (Hierarchy-based, timeline or Marey's graphs). Because order is essential for sequential data, approaches always use some flow-based visualization to show the e.g. patterns or stages. Further, additional classical graphs are commonly used to visualize additional information like attributes or statistics. Here, scatter plots/bubble charts are helpful in displaying clusters, while line charts help to shop similarity or outcome predictions.

5 Discussion and Conclusion

This paper aims to identify current state-of-the-art approaches in visual analytics that specialize in sequential data and discuss their applicability to process mining (RQ1). Based on an informal literature review, we recognized 21 approaches that have been published by different researchers on different platforms. It is to be noted that we do not claim our identified literature as complete or exclusive. There can be more relevant literature that uses different wordings or is published in other journals or databases than we used.

Approaches mostly use regular event sequences or high-dimensional temporal sequence data as their data basis. One approach uses spatio-temporal data. Most approaches focus on event summarization or stage aggregation and pattern comparison. Further, there are two approaches that focus on action recommendations and six that allow for dedicated process performance analysis.

Almost all approaches are interactive and use filters. Pattern mining, queries and stage-based aggregation are commonly used techniques to analyze the data in the backend. Some approaches already use process mining techniques for their data analysis. Regarding visualization, custom hierarchy-based and glyph-based visualizations are mostly used. They allow for a clear sight of patterns, sequences or stages. Marey's graphs are used to show process courses over time for multiple process instances, and matrix views are used to display event frequency or frequent patterns. To display attributes, statistics and other additional information, graphs like scatter plots, bar- and line charts or radial plots are used.

Identified approaches can be used in all three categories of process mining, especially for process enhancement. However, they act more as additional tools that should be used together with existing process mining tools.

- **Discovery:** While no approach directly creates a process model, systems like CoreFlow, DecisionFlow or EventThread can help understand the complexity and variety of the data. This can help, e.g. to modify process mining discovery algorithms' parameters to get better results.
- **Conformance:** Further, when a process model is already at hand, visual analytics approaches can help verify that the model represents the process.

- **Enhancement:** Here, the identified scientific approaches are most applicable. They allow digging deep into the data, validating it with process models and getting recommendations. They offer new perspectives on the process by mining new and unique patterns, converting the process into new stages or visualizing performance and time distribution over time.

Through the application of process mining and visual analytics tools, analytics can get an even better understanding of underlying processes, yielding weaknesses and allowing for steps to address those.

However, there are also two hurdles to applying visual analytics approaches with process mining.

First, the trade-off between information loss and visual clutter. This difficulty occurs in all approaches. Approaches always have to decide which level of aggregation they choose. If you aggregate little, the section of the data set becomes smaller, but if you want to analyze the entire data set, a high aggregation level must be selected to avoid visual clutter. Sequence Synopsis and MAQUI present initial approaches, but it is very likely that there is still room for improvement.

Second, the importance of process graphs in process mining. The process graph is of essential importance in process mining. However, only ViDX integrates a form of process graph into its approach. Although visual analytics approaches can provide valuable information for process mining even without a process graph, integrating process graphs into visual analytics tools could contribute to even better support of the tools for process mining.

References

1. van der Aalst, W.e.a.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) Business Process Management Workshops. pp. 169–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
2. Andrienko, N., Andrienko, G.: Spatio-temporal visual analytics: a vision for 2020s (2020). https://doi.org/10.5311/JOSIS.2020.20.661, https://publica.fraunhofer.de/handle/publica/263284
3. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 429–435. KDD '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/775047.775109, https://doi.org/10.1145/775047.775109
4. Bai, J., Zhang, H., Qu, D., Lv, C., Shao, W.: Fgvis: visual analytics of human mobility patterns and urban areas based on f-glove. Journal of Visualization **24**(6), 1319–1335 (2021). https://doi.org/10.1007/s12650-021-00775-x, https://doi.org/10.1007/s12650-021-00775-x
5. Bergner, Y., von Davier, A.A.: Process data in naep: Past, present, and future. Journal of Educational and Behavioral Statistics **44**(6), 706–732 (2019). https://doi.org/10.3102/1076998618784700, https://doi.org/10.3102/1076998618784700

6. Bodesinsky, P., Alsallakh, B., Gschwandtner, T., Miksch, S.: Visual process mining: Event data exploration and analysis. In: 2014 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 229–230 (2014). https://doi.org/10.1109/VAST.2014.7042504
7. Caballero, H.S.G., Corvò, A., Dixit, P.M., Westenberg, M.A.: Visual analytics for evaluating clinical pathways. In: 2017 IEEE Workshop on Visual Analytics in Healthcare (VAHC). pp. 39–46 (Oct 2017). https://doi.org/10.1109/VAHC.2017.8387499
8. 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). pp. 975–981 (Dec 2018). https://doi.org/10.1109/BigData.2018.8622571
9. Chen, Y., Xu, P., Ren, L.: Sequence synopsis: Optimize visual summary of temporal event data. IEEE Transactions on Visualization and Computer Graphics **24**(1), 45–55 (2018). https://doi.org/10.1109/TVCG.2017.2745083
10. Denisov, V., Fahland, D., van der Aalst, W.M.P.: Unbiased, fine-grained description of processes performance from event data. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) Business Process Management. pp. 139–157. Springer International Publishing, Cham (2018)
11. Doraiswamy, H., Freire, J., Lage, M., Miranda, F., Silva, C.: Spatio-temporal urban data analysis: A visual analytics perspective. IEEE Computer Graphics and Applications **38**(5), 26–35 (2018). https://doi.org/10.1109/MCG.2018.053491728
12. Du, F., Plaisant, C., Spring, N., Shneiderman, B.: Eventaction: Visual analytics for temporal event sequence recommendation. In: 2016 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 61–70 (Oct 2016). https://doi.org/10.1109/VAST.2016.7883512
13. Fournier-Viger, P., Wu, C.W., Gomariz, A., Tseng, V.S.: Vmsp: Efficient vertical mining of maximal sequential patterns. In: Sokolova, M., van Beek, P. (eds.) Advances in Artificial Intelligence. pp. 83–94. Springer International Publishing, Cham (2014)
14. Gotz, D., Stavropoulos, H.: Decisionflow: Visual analytics for high-dimensional temporal event sequence data. IEEE Transactions on Visualization and Computer Graphics **20**(12), 1783–1792 (2014). https://doi.org/10.1109/TVCG.2014.2346682
15. Gotz, D., Wang, F., Perer, A.: A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. Journal of Biomedical Informatics **48**, 148–159 (2014). https://doi.org/https://doi.org/10.1016/j.jbi.2014.01.007, https://www.sciencedirect.com/science/article/pii/S1532046414000094
16. Gotz, D., Zhang, J., Wang, W., Shrestha, J., Borland, D.: Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation. IEEE Transactions on Visualization and Computer Graphics **26**(1), 440–450 (Jan 2020). https://doi.org/10.1109/TVCG.2019.2934661
17. Gschwandtner, T.: Visual analytics meets process mining: Challenges and opportunities. In: Ceravolo, P., Rinderle-Ma, S. (eds.) Data-Driven Process Discovery and Analysis. pp. 142–154. Springer International Publishing, Cham (2017)
18. Guo, C., Xu, S., Yu, J., Zhang, H., Wang, Q., Xia, J., Zhang, J., Chen, Y.V., Qian, Z.C., Wang, C., Ebert, D.: Dodeca-rings map: Interactively finding patterns and events in large geo-temporal data. In: 2014 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 353–354 (2014). https://doi.org/10.1109/VAST.2014.7042562

19. 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), 417–426 (Jan 2019). https://doi.org/10.1109/TVCG.2018.2864885
20. 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), 56–65 (Jan 2018). https://doi.org/10.1109/TVCG.2017.2745320
21. Guo, Y., Guo, S., Jin, Z., Kaul, S., Gotz, D., Cao, N.: Survey on visual analysis of event sequence data (2020)
22. Herr, D., Beck, F., Ertl, T.: Visual analytics for decomposing temporal event series of production lines. In: 2018 22nd International Conference Information Visualisation (IV). pp. 251–259 (2018). https://doi.org/10.1109/iV.2018.00051
23. Icke, I., Sklar, E.: Visual analytics : A multi-faceted overview (2009), https://api.semanticscholar.org/CorpusID:14038874
24. Jiang, H., Yang, K., Qin, Y., Yang, Y., Zhao, Y., Zhou, F.: Visual analytics of spatio-temporal uncertainties for radiation monitoring in a nuclear leakage crisis. In: 2019 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 132–133 (2019). https://doi.org/10.1109/VAST47406.2019.8986925
25. Keim, D., Andrienko, G., Fekete, J.D., Görg, C., Kohlhammer, J., Melançon, G.: Visual Analytics: Definition, Process, and Challenges, pp. 154–175. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70956-5_7, https://doi.org/10.1007/978-3-540-70956-5_7
26. Kwon, B.C., Verma, J., Perer, A.: Peekquence: Visual analytics for event sequence data (08 2016)
27. Laue, R., Koschmider, A., Fahland, D. (eds.): Grundlagen. De Gruyter Oldenbourg, Berlin, Boston (2021). https://doi.org/doi:10.1515/9783110500165, https://doi.org/10.1515/9783110500165
28. Law, P.M., Liu, Z., Malik, S., Basole, R.C.: Maqui: Interweaving queries and pattern mining for recursive event sequence exploration. IEEE Transactions on Visualization and Computer Graphics **25**(1), 396–406 (Jan 2019). https://doi.org/10.1109/TVCG.2018.2864886
29. Li, C., Dong, X., Liu, W., Sheng, S., Qian, A.: Ssrdvis: Interactive visualization for event sequences summarization and rare detection. Journal of Visualization **23**(1), 171–184 (2020). https://doi.org/10.1007/s12650-019-00609-x, https://doi.org/10.1007/s12650-019-00609-x
30. Liu, D., Xu, P., Ren, L.: Tpflow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. IEEE Transactions on Visualization and Computer Graphics **25**(1), 1–11 (2019). https://doi.org/10.1109/TVCG.2018.2865018
31. Liu, Z., Kerr, B., Dontcheva, M., Grover, J., Hoffman, M., Wilson, A.: Coreflow: Extracting and visualizing branching patterns from event sequences. Computer Graphics Forum **36**(3), 527–538 (2017). https://doi.org/https://doi.org/10.1111/cgf.13208, https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13208
32. Maier, S., Kühnel, H., May, T., Kuijper, A.: Visual interactive process monitoring. In: Yamamoto, S. (ed.) Human Interface and the Management of Information. Information and Knowledge Design. pp. 74–85. Springer International Publishing, Cham (2015)
33. Măruşter, L., van Beest, N.R.T.P.: Redesigning business processes: a methodology based on simulation and process mining techniques. Knowledge and Information Systems **21**(3), 267–297 (2009). https://doi.org/10.1007/s10115-009-0224-0, https://doi.org/10.1007/s10115-009-0224-0

34. Mennens, R.J., Scheepens, R., Westenberg, M.A.: A stable graph layout algorithm for processes. Computer Graphics Forum **38**(3), 725 – 737 (2019), http://www.redi-bw.de/db/ebsco.php/search.ebscohost.com/login.aspx%3fdirect%3dtrue%26db%3daph%26AN%3d137771657%26site%3dehost-live
35. Munoz-Gama, J.: Introduction, pp. 1–8. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-49451-7_1, https://doi.org/10.1007/978-3-319-49451-7_1
36. Munzner, T.: Visualization analysis design. A K Peters visualization series, Boca Raton, Florida, 1st edition edn. (2015)
37. Perer, A., Gotz, D.: Data-driven exploration of care plans for patients. In: CHI '13 Extended Abstracts on Human Factors in Computing Systems. pp. 439–444. CHI EA '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2468356.2468434, https://doi.org/10.1145/2468356.2468434
38. Pham, V.V., Dang, T.: Mtdes: Multi-dimensional temporal data exploration system. In: 2018 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 100–101 (2018). https://doi.org/10.1109/VAST.2018.8802440
39. Sarker, I.H.: Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science **2**(6), 420 (2021). https://doi.org/10.1007/s42979-021-00815-1, https://doi.org/10.1007/s42979-021-00815-1
40. Sitova, I., Pecerska, J.: Process data analysis using visual analytics and process mining techniques. In: 2020 61st International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS). pp. 1–6 (Oct 2020). https://doi.org/10.1109/ITMS51158.2020.9259296
41. 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. pp. 1747–1756. CHI '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/1978942.1979196, https://doi.org/10.1145/1978942.1979196
42. Xie, Z., Yang, Y., Zhang, Y., Wang, J., Du, S.: Deep learning on multi-view sequential data: a survey. Artificial intelligence review **56**(7), 6661—6704 (2023). https://doi.org/10.1007/s10462-022-10332-z, https://europepmc.org/articles/PMC9707228
43. Xu, P., Mei, H., Ren, L., Chen, W.: Vidx: Visual diagnostics of assembly line performance in smart factories. IEEE Transactions on Visualization and Computer Graphics **23**(1), 291–300 (2017). https://doi.org/10.1109/TVCG.2016.2598664
44. Zhang, T., Ma, Y., Xu, D., Liu, C., Ding, J.: ihppvis: Interactive visual analysis of industrial data in heavy plate production. IFAC-PapersOnLine **53**(2), 12050–12055 (2020). https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.745, https://www.sciencedirect.com/science/article/pii/S2405896320310648, 21st IFAC World Congress
45. Zinat, K.T., Yang, J., Gandhi, A., Mitra, N., Liu, Z.: A comparative evaluation of visual summarization techniques for event sequences. Computer Graphics Forum **42**(3), 173–185 (2023). https://doi.org/https://doi.org/10.1111/cgf.14821, https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14821

A Hierarchy-based approaches

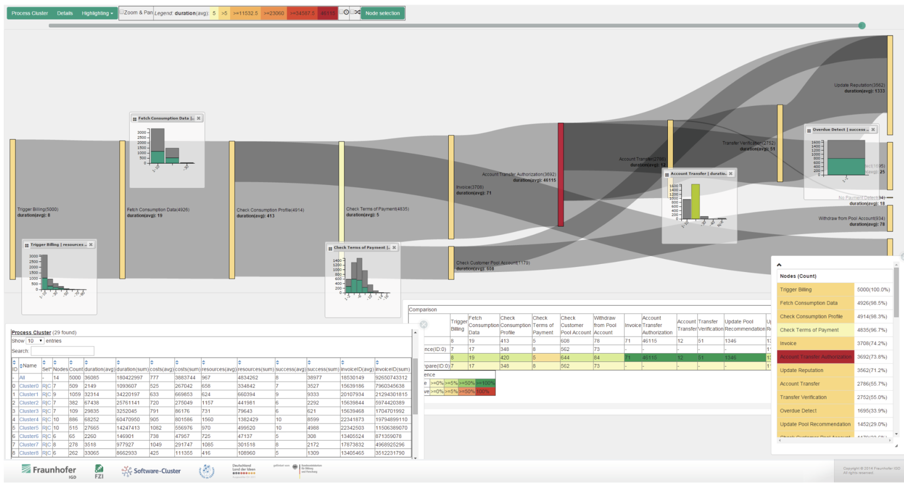

Fig. 8: Sankey diagram in Maier et al. [32]

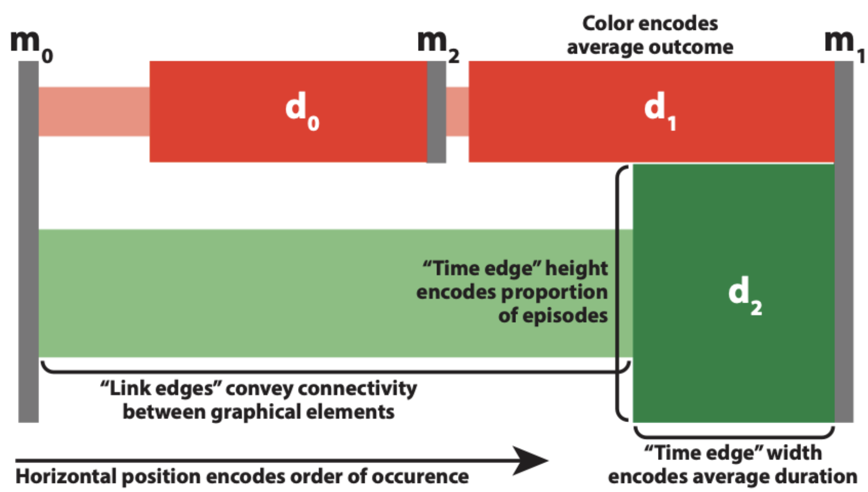

Fig. 9: Basic node concept in DecisionFlow [14]

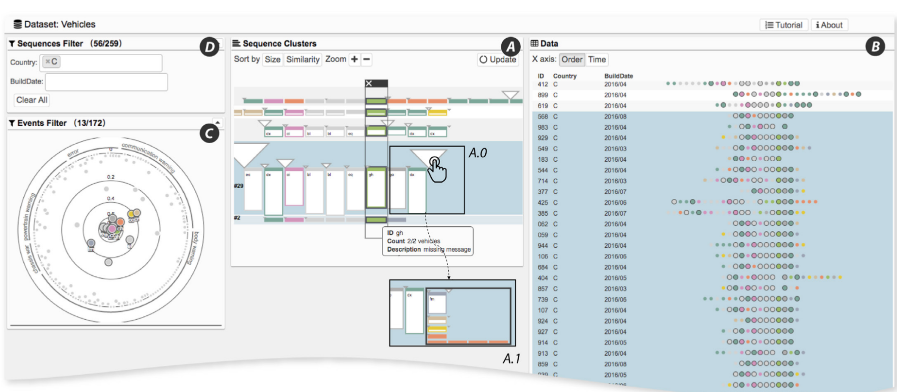

Fig. 10: Pattern view in Sequence Synopsis [9]

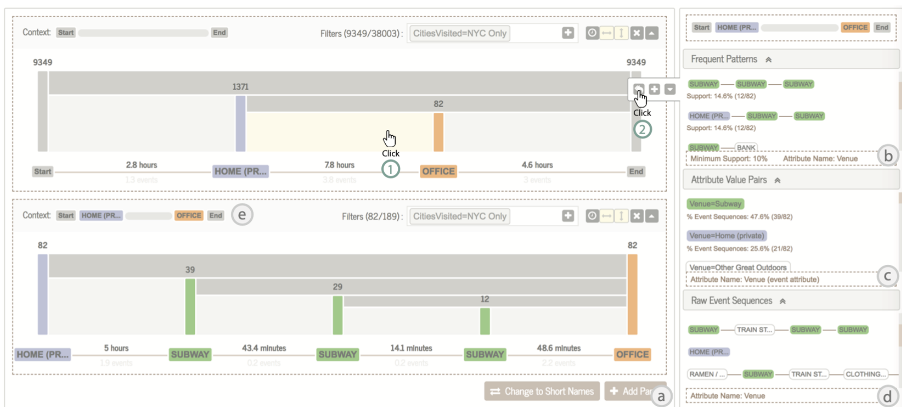

Fig. 11: MAQUI visual analytics approach [28]

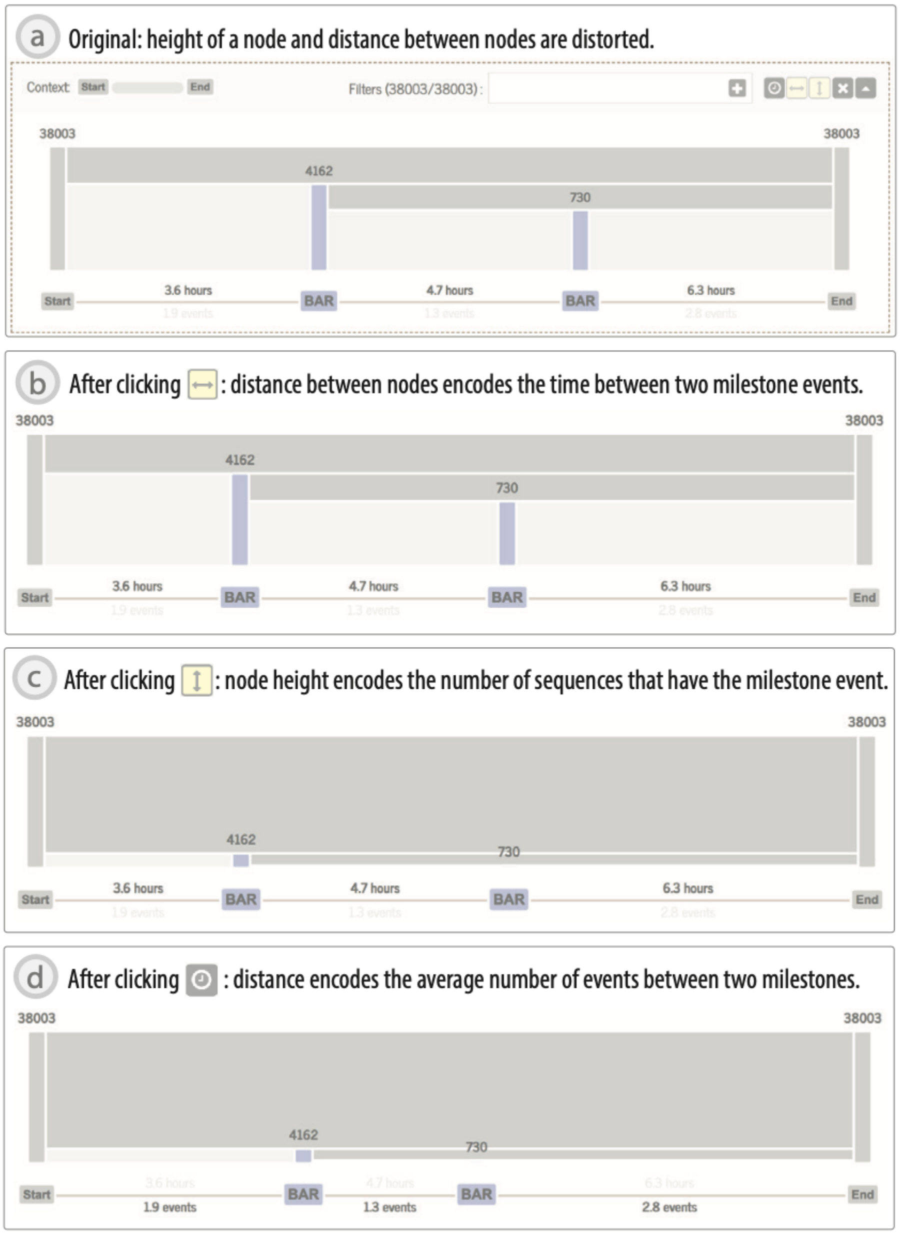

Fig. 12: Alignment options in MAQUI [28]

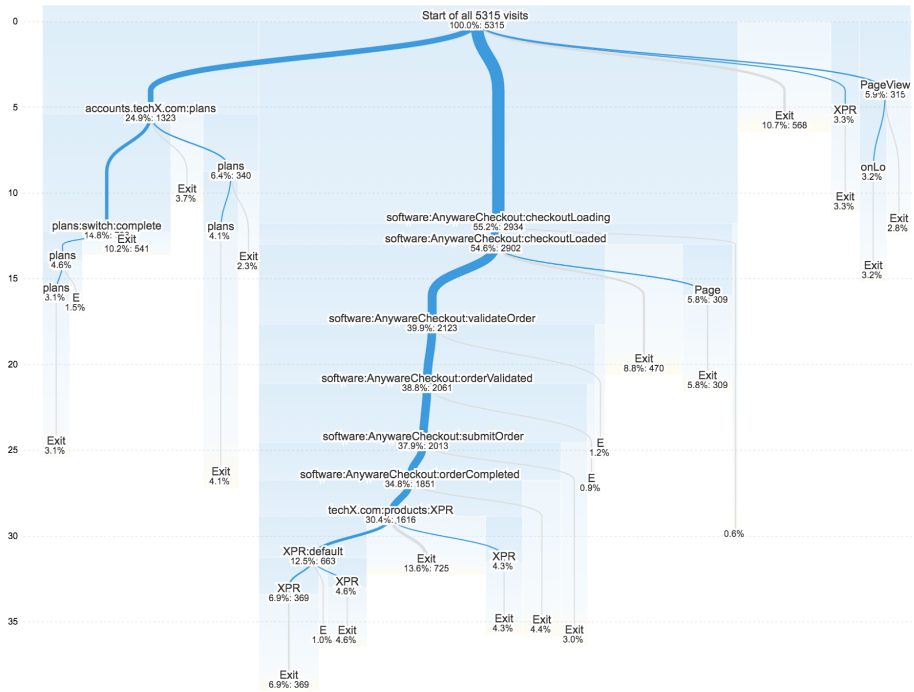

Fig. 13: Hybrid view in CoreFlow [31]

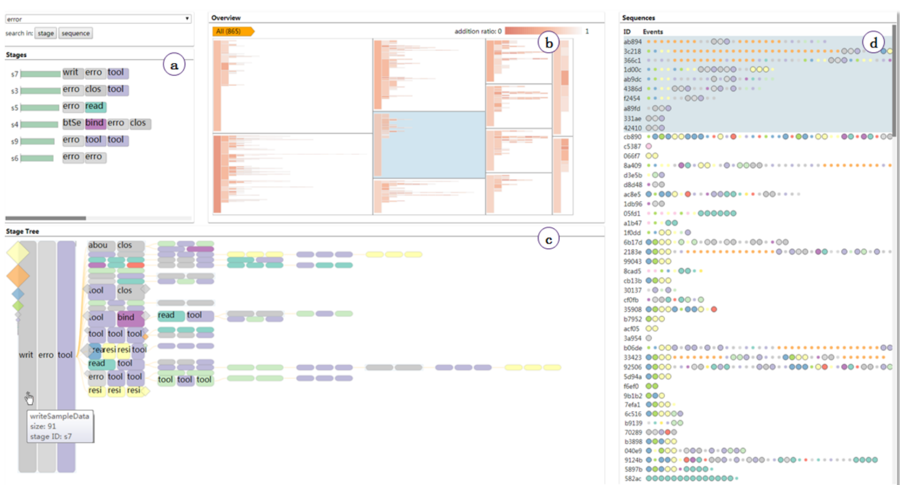

Fig. 14: Stage view in StageMap [8]

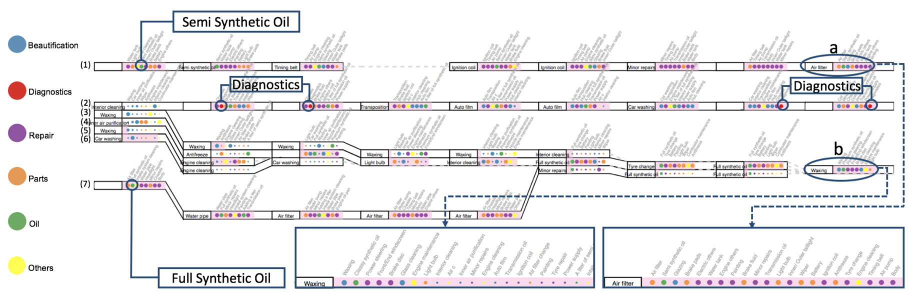

Fig. 15: Thread view in EventThread [20]

B Matrix-based visualization

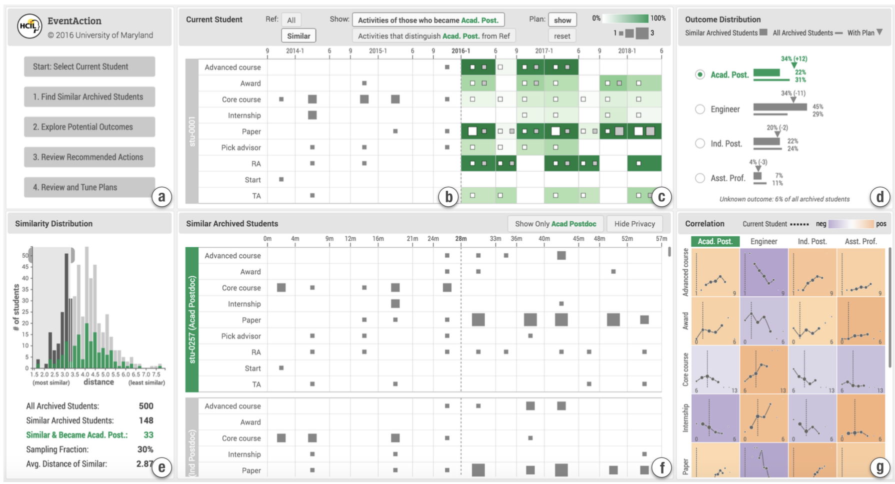

Fig. 16: Matrix view in EventAction [12]

C Marey's graphs

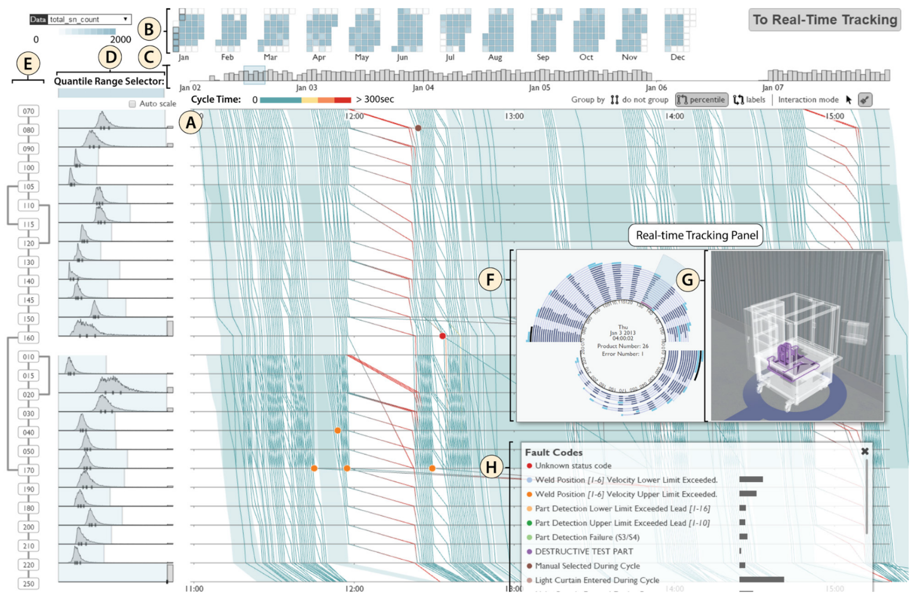

Fig. 17: Usage of Marey's Graphs in ViDX [43]

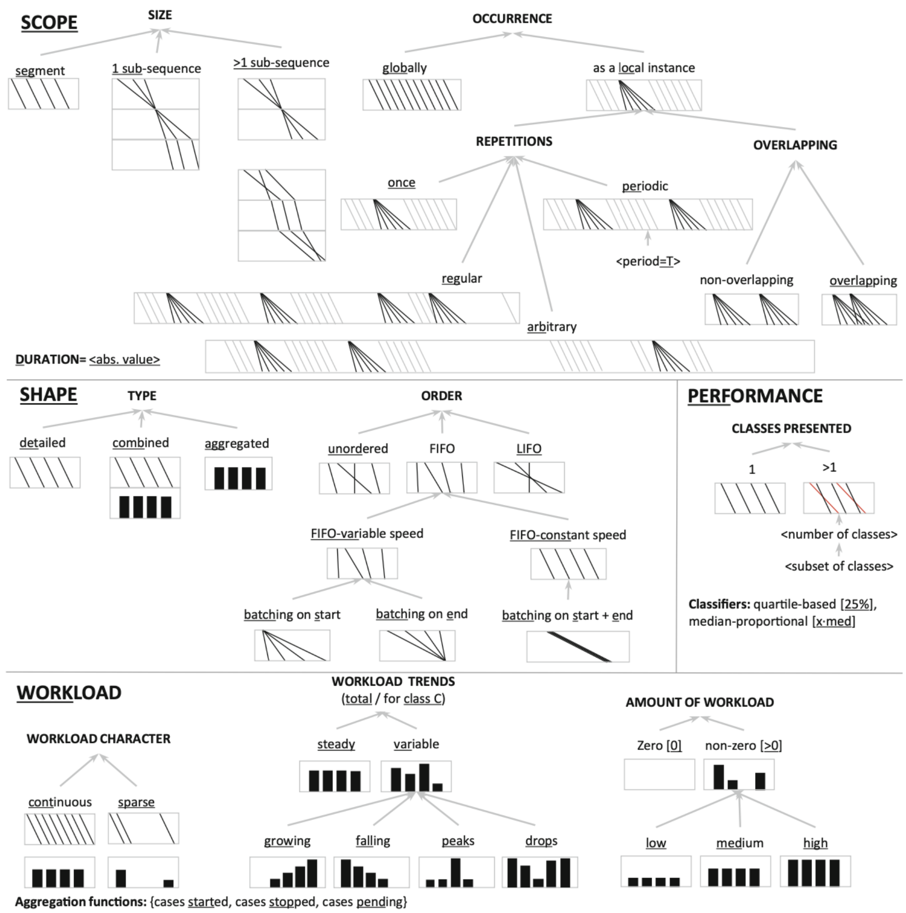

Fig. 18: Taxonomy of elementary patterns

D Glyph-based

Fig. 19: Icon "glphys" in SepVis [7]

Affidavit

I hereby declare that I have developed and written the enclosed seminar thesis entirely on my own and have not used outside sources without declaration in the text. Any concepts or quotations applicable to these sources are clearly attributed to them. This seminar thesis has not been submitted in the same or a substantially similar version, not even in part, to any other authority for grading and has not been published elsewhere. This is to certify that the printed version is equivalent to the submitted electronic one. I am aware of the fact that a misstatement may have serious legal consequences.

I also agree that my thesis can be sent and stored anonymously for plagiarism purposes. I know that my thesis may not be corrected if the declaration is not issued.

Mannheim, November 4, 2023

Philemon Schulz

Philemon Schulz