A Survey of Approaches for Event Sequence Analysis and Visualization using the ESeVis Framework

Anton Yeshchenko\textsuperscript{a}, Jan Mendling\textsuperscript{b}

\textsuperscript{a}Vienna University of Economics and Business, Vienna, Austria
\textsuperscript{b}Humboldt-Universität zu Berlin, Berlin, Germany

Abstract

Event sequence data is increasingly available. Many business operations are supported by information systems that record transactions, events, state changes, message exchanges, and so forth. This observation is equally valid for various industries, including production, logistics, healthcare, financial services, education, to name but a few. The variety of application areas explains that techniques for event sequence data analysis have been developed rather independently in different fields of computer science. Most prominent are contributions from information visualization and from process mining. So far, the contributions from these two fields have neither been compared nor have they been mapped to an integrated framework. In this paper, we develop the Event Sequence Visualization framework (ESeVis) that gives due credit to the traditions of both fields. Our mapping study provides an integrated perspective on both fields and identifies potential for synergies for future research.

Keywords: Event Sequence Data, Information Visualization, Process Mining

2010 MSC: 00-01, 99-00
1. Introduction

Event sequence analysis is an important field of computer science due to its relevance to a diverse spectrum of application domains such as manufacturing, logistics, healthcare, financial services, education [1], to name but a few. Despite this broad relevance across these domains, it is striking to observe that techniques for event sequence data analysis have been developed rather independently in different fields of computer science.

The most prominent research fields investigating the analysis of event sequence data are process mining and information visualization. Process mining has emerged as a subfield of research into workflow management systems [2]. Its focus is the development of new techniques for automatic process discovery from event sequence data with the ambition to provide a meaningful and understandable summary of the behaviour to the business process analyst. Information visualization is a field of computer graphics, which originated as a subfield of human-computer interaction [3]. Its focus is on devising new techniques for visualizing event sequence data in a meaningful way such that analysts can effectively explore them. Typical representations frequently used in this field are timelines that plot conceptually related sequences of events over a time axis. As similar as the ambitions of these research areas may sound, it is surprising that there is hardly any exchange of ideas. Cross-references are scarce and mutual awareness and understanding is limited. All this makes research on event sequence analysis a fragmented field with scattered contributions.

So far, the contributions from these two fields have neither been compared nor have they been mapped to an integrated framework. For this reason, it is not clear to which extent both fields have developed complementary concepts and insights. In this paper, we develop such a framework that we call EseVis and that gives due credit to the traditions of both fields. Our mapping study

\footnote{For instance, by December 2021 there are less than ten articles in IEEE Transactions on Visualization and Computer Graphics that mention the term “process mining” at all.}
provides an integrated perspective on both fields and potential synergies for future research. In this way, our work contributes towards overcoming the fragmentation of research on event sequence data analysis.

The paper is structured as follows. Section 2 discusses the background of our research. We use the example of event sequence data of sepsis treatments in a hospital to illustrate and define categories of event sequence data representations and visualizations. Based on both representations and visualizations, we develop our ESeVis framework. Section 3 describes our method of systematically reviewing the literature on event sequence data analysis in the fields of information visualization and process mining. Section 4 presents our findings. We provide descriptive results on the distribution of contributions over the different categories of the ESeVis framework. Furthermore, we illustrate key concepts for each category of the framework. Section 5 concludes the paper and gives an outlook on future research.

2. Background

In this section, we discuss the background of our research. First, we describe the various types, characteristics and formats of event sequence data. Then, we distinguish two distinct conceptual categories of event sequence representation, namely sequence representation and model representation. Moreover, we identify five visualization techniques. Based on conceptual representation and visualization technique, we develop our ESeVis framework for event sequence data analysis. We use the term representation to refer to the formal syntax and semantics of the event sequence data types and visualization to refer to specific visualization techniques, similar to the distinction made by Keim [4].

2.1. Example of Sepsis Treatment in a Hospital

Event sequence data originates from a variety of real-world processes. For example, such data is recorded by information systems, such as enterprise resource planning systems [5] or healthcare systems [6]). These systems capture sequences of transactions, web click streams [7], student learning progress [8], or
Table 1: Example of a typical steps in the Sepsis process based on data of [11].

| Steps             | Description                                                                 |
|-------------------|-----------------------------------------------------------------------------|
| ER Registration   | Register a patient                                                          |
| ER Triage         | Fill a general document to assign degree of an illness and decide the order of treatment (Triage document) |
| ER Sepsis Triage  | Fill an additional triage document if a patient is suspected to have sepsis |
| CRP               | Activity related to measurement of C-Reactive Protein (CRP)                  |
| LacticAcid        | Activity related to measurement of LacticAcid                                |
| Leucocytes        | Activity related to measurement of Leucocytes                                |
| Release C         | Patient discharge                                                           |

any other traces of human behavior [9,10]. The common characteristic of these event sequence data is that they encode sequences of events over time.

We use the example of the event sequence data originating from the transaction system of a hospital for illustrating important characteristics of such data. This example covers the event sequences of handling sepsis cases [11]. The events of these sequences refer to the progression of patients in the hospital including registration, tests, diagnoses, possible transfer to care units, and release. Table 1 summarizes the most frequent steps of this process. These represent the typical treatment sequence of a patient.

2.2. Representation of Events in Event Sequence Data

Event sequence data represents event sequences such as documented cases of treating sepsis patients. An event sequence is an ordered list of events [12]. One event represents a step in a process, such as ER Sepsis Triage for the Sepsis process. Each sequence has a start event, such as ER Registration, and finishes with an end event, such as Release A. Events can be represented as a symbolic value, numerical value, or as a complex data type comprising of a set of values of different types. It depends on the real-world phenomenon and the supporting
systems which kind of event data can be recorded. In particular, we distinguish different categories of event data [12].

- **Simple symbolic sequence:** If each event in the sequence can be described by one categorical value, then the sequence of such values forms a simple symbolic sequence. Formally, this data is an ordered list of events instantiated from some alphabet \( \{e_1, e_2, \ldots, e_n\} \). These event sequences can represent web click streams or program execution sequences. The sequence of sepsis-related treatment events for one patient are, for instance, \textit{ER Registration}, \textit{ER Triage}, \textit{LacticAcid}, \textit{Release A}.

- **Complex symbolic sequence:** When events are represented as tuples of categorical values, the event sequences formed by such events is called complex symbolic sequence. This type of data is an ordered list of vectors that capture values drawn from some alphabet \( \{\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_n\} \). One example of recording events related to procedures performed on a patient for \textit{Sepsis} case would be:

\( \langle (ER\text{Registration}, Private\text{Insurance}), (ER\text{Triage}, Dr.Garcia), (Lactic\text{Acid}, Leucocytes, Dr.Williams), \ldots, (ReleaseA) \rangle. \)

- **Simple time series:** If an event represents a single numerical value combined with the time when this value was recorded, the event sequences originating from these events is called a simple time series. This type of data is a sequence of timestamped real-valued numbers, representing a change of a value over predefined time periods. In the \textit{Sepsis} process, the time series reflecting the current number of treated patients in the clinic is one example and another one the change of health indicators of a patient over time. For instance, \( \langle (t_1, 37^\circ), (t_2, 39^\circ), \ldots, (t_n, 38^\circ) \rangle \) can represent the body temperature of a patient.
• **Multivariate time series:** In this representation, each event is captured as several numerical values. This type of data is a sequence of timestamped real-valued vectors, describing how these vectors change over time. Each single event is represented by a list of values. In the *Sepsis* example, several health statistics related to a patient can be grouped together into multivariate time series. The change of body temperature and blood sugar can be represented as multivariate time series $\langle (t_1, \langle 37.9^\circ, 190 \rangle), ..., (t_n, \langle 36.6^\circ, 120 \rangle) \rangle$

• **Complex event sequence:** When real-world phenomena are captured with rich information, recorded events can contain complex content. This type of data is the sequence of values that have complex data types, each event containing a number of values, usually including timestamp and multiple other event characteristics of textual, numerical and categorical types. For each event related to a patient in the *Sepsis* data set, the timestamp, event name, and many other characteristics can be recorded. The example of an event can be 10:05 10.05.2021 $\langle \text{AdmissionIC}, 36.6^\circ, 120\% \rangle$ that describes the event *Admission IC* that was recorded on 10:05 10.05.2021, and the person in question had a body temperature of 36.6 and a sugar level at 120.

The choice of how events are represented in event sequence data determines which analysis can be done. Simple symbolic sequence and complex symbolic sequences can be used to discover the sequential patterns of events. The simple and multivariate time series can be used to calculate regression and numerical statistics. The complex event sequences can be used for a combination of these analyses.

2.3. Representation of Event Sequences

No matter how events are stored, we can distinguish two ways how event sequences can be represented. First, we describe different *instance-based representations* of event sequences. These representations have in common that they list event sequences according to different principles. They describe the actual
event sequences that have been observed. Second, model-based representations do not describe individual sequences, but the rules by which these sequences can be generated. They abstract from actual event sequences. We describe both representations in turn.

2.3.1. Instance-based Representation of Event Sequences

Instance-based representations organize data on events of several event sequences in different ways. We distinguish event enumeration, case-event enumeration, and case-state enumeration (see Figure 1). Different file formats exist for each of these categories.

![Figure 1: Examples of different instance-based representations of event sequences](image)

**Event enumeration.** This event sequence representation stores data as a sequence of events. These events contain the event name, timestamp, and any additional attributes related to the event. Additional attributes contain important information that can be used to order events into separate event sequences (or cases). One example for event enumeration is the OCEL file format that is used to store lists of events and support multiple case notions [13]. An example of data stored in this format is shown in Figure 1A. The OCEL format was developed to solve the problem of storing events that can be grouped according to different case notions into different event sequences. For instance, events of the sepsis data collection can be grouped into sequences based on the same patient (sequence of the treatment) or based on the same nurse (sequence of a shift).
OCEL was designed to use XML-based syntax to store event sequence data with a possibility of storing also the information about the relation between objects within event sequences. That information allows the data analysis to identify the resulting case notion and resulting event sequences at the stage of analysis. OCEL provides a general process mining standard to interchange object-centric event data with multiple case notions.

Case-event enumeration. This type of storing event sequences requires the definition of an order of events and their reference to one specific case. This yields a hierarchical representation, where each event belongs to a case, which usually represents one instance of a process. The eXtensible Event Stream (XES) format is an XML-based standard event sequence format for information systems [14]. This format is used by software tools for process mining, such as ProM [15] or Disco [16]. An example of storing data in this way is shown in Figure 1.B. It covers the Sepsis data in the XES format. Using case-event enumeration, each event is assigned to a case. Here, each case represents one patient in the clinic, and the events represent the procedures performed.

Case-state enumeration. This representation focuses on the event sequences as changes of states for a particular case. The STate Sequence (STS) format considers states as a data unit instead of events [17]. This representation still uses the tabular format to store data, but with a different structure. As shown in Figure 1.C, rows correspond to the sequence of equitemporally spaced timestamps with corresponding states for each case in that time.

2.3.2. Model-based Representation of Event Sequences

Model-based representations (or for short: models) describe how individual sequences can be generated and which characteristics they have. In this way, they aggregate event sequences into abstract specifications. Several categories of the model-based representations of event sequences exist. We describe four examples that are frequently used for process mining, namely directly-follows
graphs, Petri nets, Declare constraints, and process trees [2]. It is well known that transformations exist between many of these representations [18, 19, 20].

**Directly-follows graphs.** Event sequences can be modeled as a graph, where each node represents a type of event and arcs describe directly-follows relationships between these event types [2]. An exemplary directly-follows graph using Sepsis is shown in Figure 2. In essence, it shows for each event types like ER Sepsis Triage the event types that can follow next in an event sequence, here Leucocytes and CRP.

![Figure 2: Directly-follows graph for the Sepsis example](image)

**Petri nets.** This representation is a specific kind of a bipartite graph [21]. There are two types of nodes: places and transitions. These can be connected by arcs representing the flow relations. Each place can hold several tokens, which collectively define the current state of the system. Changing the state by firing enabled transitions produces sequences. Petri nets are more expressive than directly-follows graphs, because they can explicit describe concurrency [22]. The Petri net in Figure 3 shows that ER Sepsis Triage can be followed by either Leucocytes or CRP.

![Figure 3: Petri net for the Sepsis example](image)

**LTL (Linear Temporal Logic).** is the temporal logic with modalities that describe time. Event sequences can be described as a collection of rules by which
the underlying behavior is constrained. One such specific modeling language that uses LTL for describing event sequence data is Declare [23, 20]. Table 2 shows a set of rules showing that constrain event sequences. For example, the rule ChainSuccession(ER Registration, ER Sepsis Triage) means that that the first event type has to be directly followed by the second.

| Rule | Explanation |
|------|-------------|
| ChainSuccession(ER Registration, ER Sepsis Triage) | If ER Registration occurs then ER Sepsis Triage occurs immediately afterwards |
| ChainPrecedence(ER Sepsis Triage, Leucocytes) | If Leucocytes occurs then ER Sepsis Triage occurs immediately beforehand |
| AtMostOne(ER Sepsis Triage) | If ER Sepsis Triage occurs, then it occurs at most once |
| NotSuccession(Release A, CRP) | Release A occurs if and only if CRP does not occur afterwards |

... Process Trees. are abstract hierarchical representations of a process model [24]. This tree represents events as leaf nodes, and control-flow operators as non-leaf nodes. There are a number of operators available such as exclusive choice \( \times \), sequence \( \rightarrow \), parallelism \( \wedge \). A process tree for the Sepsis case is shown in Figure 1. The sequence operator at the root of the tree indicates that the child nodes will be sequentially executed. The event sequence starts with ER Registration, continues with ER Sepsis Triage, which follows by either Leucocytes or CRP. The sequence finishes by the final event Release A.

2.4. Visualization of Event Sequence Data

There are various archetypes of how event sequence representations can be visualized. Here, we use the term visualization to refer to an arrangement of visual elements on a canvas. In this section, we explain different visualization
types that are used to support the analysis of event sequence data. We use the taxonomy presented in [25] to differentiate five main categories. We describe these categories in the order from most specific to least specific. In Figure 5, we exemplify each of these five categories for instance-based and model-based representations using the Sepsis example.

2.4.1. Matrix Visualization
A matrix visualization is the most specific visualization of event sequences. The key feature of this visualization is that event sequences or aggregate values from event sequences are mapped to rows and columns of a matrix. Relationships between two orthogonal categories are shown as various glyphs at the matrix cell where both categories intersect. Figure 5.A shows an instance-based matrix visualization inspired by the design presented in [26]. Each row of this matrix represents one sequence from the Sepsis treatment process. The columns correspond to various events that appear in the event sequences. The model-based matrix visualization is inspired by the design of [27] shown in Figure 5.B. Here, the rows of the matrix correspond to clusters of patients based on event sequences and columns represent the events that are present in those categories.

2.4.2. Timeline Visualization
Timeline visualizations are visual representations of event sequences that use a time axis to align the elements. This visualization is usually used to convey the evolution of events over time, the evolution of aggregate values, or the order of events in event sequences. The example visualization shown in Figure 5.C is based on the dotted chart design [26]. It uses circles to position each individual
Figure 5: Example of representation using Sepsis use-case
event on a horizontal timeline for each patient of the Sepsis use-case. The model-based representation of ToPIN [8] is depicted in Figure 5.D. It shows the events associated with different activities as they are executed in event sequences. In this way, it conveys information of the general order, number of events, and about groups of events (events are shown below or above the timeline).

2.4.3. Hierarchy Visualization

Hierarchy visualizations are graph-based event sequence visualizations that use a tree topology. These visualizations convey hierarchical dependence and proximity of events in event sequences. Such visualizations can be generated using clustering algorithms. Figure 5.E shows the design of individual event sequences with an order of events. The same activities are grouped together to convey the hierarchical relationships in the data. The visualization of [28, 29] is shown in Figure 5.F. It shows the hierarchy-based model representation of event sequences, conveying information about general patterns in this data set.

2.4.4. Sankey Visualization

Sankey visualizations are inspired by flow charts, showing events as bars and transitions as lines connecting them. Lines are proportional to the number of transitions in the event data. This visualizations type also includes node-link diagrams and directly-follows graph [30]. The visualization depicted in Figure 5.G shows a sankey diagram for an instance-based representation of event sequences. Each event sequence, e.g. a sepsis patient, is visible as a line going through the bars of each event. The shown variant is based on the sequence braiding visualization [31]. The model-based representation described in [32] is depicted in Figure 5.H. It shows a directly-follows graph for a sepsis case. This visualization arranges the data as a graph, showing the aggregated number of transitions between events.

2.4.5. Other Chart Visualizations

Some visualizations are more basic or combinations of other archetypes. We put all these other visualizations into this category. These typically include visual-
izations that combine simple visualization types such as bar charts, pie charts or box plots with means to convey some sequential information of the event sequence data. An instance representation inspired by [33] is shown in Figure 5.I. It uses a bar chart to represent individual sequences in the visualization. An example of model-based visualization is shown in Figure 5.J. Here, the sequence of charts are used to convey information extracted from the raw event sequences. In this case, the progression of the patients’ condition through the hospital process is displayed with bar charts. These charts are commonly used as additional views to the main visualization.

2.5. Event Sequence Visualization Framework (ESeVis)

We have identified and discussed two orthogonal dimensions for describing event sequence data: instance-based or model-based representation as the conceptual axis and five different visualization types as the visual axis. Our Event Sequence Visualization Framework (ESeVis) builds on these two axes. Table 3 describes the $2 \times 5$ categories of our framework. In the following, we will use ESeVis for categorizing contributions from the research fields of information visualization and process mining, in turn.

3. Research design

Research on visualizations of event sequence data have been published in major outlets of process mining and information visualization research. In this section, we describe how we review these streams of literature. We follow guidelines as outlined in [34, 35] and exemplified in [36]. First, we explicate our overall review objectives. Then, we define our search and selection procedures. Finally, we report our classification procedures.

3.1. Research objectives

The goal of this survey is to analyze the visualizations for event sequence data from two notable streams of research: information visualization and process mining. While both of these fields share commonalities in terms of data types,
Table 3: ESeVis Framework

| Model representation | Instance representation |
|-----------------------|--------------------------|
| Aggregated values coming from events, their additional values, or whole event sequences are mapped to rows and columns of the matrix. Each cell then shows some characteristics of these aggregations. | Event sequences and their features are mapped to the rows and columns of the matrix. Each matrix cell shows some values such as the presence of a particular event in the sequence. |
| The event sequences are aggregated before being mapped to the visual elements on the visualization with a time axis. | Each event sequence is mapped to the visual elements on the visualization with a time axis. |
| Event sequences are hierarchically clustered and mapped to the visualization elements that convey and highlight a hierarchical relationship present in event sequences. The event sequences are usually clustered with prefix-based clustering algorithms. | The event sequences are mapped to the visual elements on the visualization with a time axis. |
| Sankey-based visualization displays event sequences after aggregations over the initial event sequences. | Each event sequence from a collection of event sequences is directly mapped to the visual elements of a Sankey-based visualization. |
| Aggregate statistics derived from an event sequence are visualized with chart-based visualization or a set of chart-based visualizations. | Chart-based visualizations or their elements are used to map event sequence data to the visual elements. |
they have developed largely disconnected from each other. Our systematic literature review aims to categorize contributions from these two research fields in order to highlight commonalities and potential for mutual inspiration. We approach these objectives by focusing on the following research question (Q):

\[ Q1: \text{Which type of event sequence visualizations have been proposed by which stream of research?} \]

We refine this research question into the following sub-questions:

- Q1.1: How can event sequence visualizations be classified?
- Q1.2: How has research on event sequence visualizations developed over time?
- Q1.3: What are the main difference between two research streams?

3.2. Literature Search

We focused on the major process mining and information visualization journals for our systematic literature review. We consider the recent period between 2000-2020 for the search. We selected three major research journals in the information visualization field that publish articles about event sequence visualizations. These journals are IEEE Transactions on Visualization and Computer Graphics (TVCG), Information Visualization (IV), and Computer Graphics Forum (CGF). For process mining research, we selected the following three major journals: Information Systems (IS), IEEE Transactions on Knowledge and Data Engineering (TKDE), and Decision Support Systems (DSS).

Information visualization and process mining related publications use a different vocabulary to describe the works focusing on event sequence data. Therefore, we iteratively defined an initial keyword list, and extended it with the

\footnote{Note that the choice for this period is driven by the assumption that relevant visualizations should be in publications over such a span of time. Both fields look back at a more than 100 year old history\cite{37,3}.}
keywords extracted from relevant publications. For the visualization journals we searched for the following keywords: *event-based data*, *temporal event data*, *timeline*, *sankey*, *process mining*. Looking through the results we identified additional keywords and added them to our search list: *flow*, *sequence mining*, *action sequence*, *Temporal visualization*, *behaviour*, *log data*, *timenets*, *Temporal Categorical*, *sequential pattern mining*, *event sequence*, *Event sequences*, *temporal event sequences*. For process mining focused journals we used the following search keywords: *Process mining*, *Event sequence* and additional keywords *Event sequences*. For the selection of relevant papers, we used the Publish or Perish software [38] with the Google Scholar backend. We build our queries using the ISSN journal identifiers and mentioned keywords.

3.3. Selection Procedure

The selection procedure has the aim to identify step by step relevant papers. The starting point are the papers identified by the search queries. Figure 6 gives an overview of the procedure.

1. We selected all the papers returned by the keyword search.
2. We analyzed the titles of each paper for being unrelated or potentially related based on our inclusion criterion. Our inclusion criterion considers research articles as relevant that propose new visualization techniques for event sequence data.
3. We analyzed abstracts and visualizations presented in papers. If a paper contains an abstract that discusses the visualization of event sequence data, or shows visualizations related to one of the categories described in Section 2.4 we consider it to be potentially relevant.
4. We fully read the papers and identified the set of relevant papers. We documented the steps as summarized in Table 4. Overall, we identified 62 relevant papers.
Table 4: Selection of papers

|                | Stage 1 | Stage 2 | Stage 3 | Stage 4 |
|----------------|---------|---------|---------|---------|
| TVCG           | 339     | 180     | 56      | 37      |
| IV             | 66      | 34      | 6       | 1       |
| CGF            | 113     | 54      | 11      | 9       |
| Information Visualization | 518     | 267     | 73      | 47      |
| IS             | 195     | 108     | 9       | 9       |
| TKDE           | 158     | 28      | 1       | 1       |
| DSS            | 111     | 36      | 6       | 5       |
| Process Mining | 464     | 172     | 16      | 15      |
| **Total**      | **982** | **440** | **89**  | **62**  |

3.4. Classification Procedure

The aim of the classification is to identify where which type of contributions have been made. We analyzed each of the 62 included papers and their visualizations and categorized the contributions according to our ESeVis framework (see Table 3). We documented additional details about specifics of the identified visualizations. Often, research articles introduce visualization systems that contain one or more views to show a specific perspective on the event sequence data. Therefore, each contribution can belong to several categories in the 2x5 framework. Moreover, several visualizations introduce views that can be classified into more than one categories (e.g. the icicle plot in [29] is categorized into the timeline-based, model-based category and the hierarchy-based model repre-
sentation as it adheres to definitions of both). Next, we present our findings.

4. Findings

In this section, we describe the results of our literature review. First, we will give an overview of the contributions. Then, we will discuss each of the five types of visualization and their sub-types in turn.

4.1. Overview of Contributions

The number of relevant papers is shown in Table 4. 62 articles were selected from the six journals. We observe that TVCG appears to be the journal with the most articles on the visualization of event sequence data. More than half of the relevant papers, namely 37 of 62, are published in this journal. CGF and IS follow with 9 articles each. DSS has five articles, both IV and DSS one.

Research on the visualization of event sequence data has been developing in the twenty years since 2000. The distribution over years is shown for each journal in Figure 7. We observe that there are a few articles between 2008 and 2013. Only 2014 sees a drastic increase of published contributions. Since then, almost 10 articles are published each year, most of them in TVCG. Figure 8 shows how the contributions of these articles are distributed over the different categories of visualizations for event sequence data. We will now discuss each category in detail. We use "IV" for information visualization and "PM" for process mining together with the reference number of the paper to indicate to which category a reference belongs.
4.2. Matrix-based Visualizations

Matrix-based visualizations use a matrix layout to display characteristics of event sequence data. This visual representation has two dimensions, where the rows correspond to one characteristic such as unique event sequence identifiers, and columns correspond to another dimension of data, such as events in a sequence, or transitions between events. The summary of contributions that use a matrix-based visualizations is presented in Table 5. Next, we separately discuss Matrix-based Visualizations and then Extended Matrix-based Visualizations.

4.2.1. Matrix-based Visualizations

In Matrix-based visualizations, the matrix cells are used to represent elementary information that can be described as a categorical values (see Figure 9). The cells are usually distinguished with color, which represent the type of the cell.

Bose et al. use the matrix layout to display event sequence data at the sequence representation level.
They arrange event sequences in a matrix view as follows: each line represents one event sequence and each column represents an event type. This visualization supports the analysis and complements a corresponding algorithm that minimizes the number of vertical gaps in the matrix for aligning similar event sequences.

Other works use Matrix-based visualizations to show aggregated characteristics of event sequence data [27, 39, IV] by conveying categorical information with color hue. Chen et al. [27, IV] use topic modeling algorithms to aggregate event sequences into topics, and generate visualizations showing how these topics map to events in a matrix form. Kwon et al. [39, IV] use a matrix view to show medical data using hidden markov models (HMM). They lay out the discovered states of these models as columns and multiple event sequence attributes as rows.

4.2.2. Extended Matrix-based Visualizations

Some matrix visualizations are more complex. They use the cells in order to convey additional quantitative and qualitative information through a special glyph designs. This design is used to convey more information than just simple categorical values (Figure 10).
Loorak et al. [40] IV represent the medical data of stroke patients. Instead of showing the raw event sequences only, they use a matrix visualization to convey information about additional parameters attached to each sequence. In this way, the matrix cells show quantities as bar charts, categorical information with color, and shapes.

Some Extended Matrix-based visualizations convey aggregated information about event sequences. Jin et al. [41] IV use a matrix view to present a causality analysis for event sequence data. They use the rows of the matrix to represent causes and the columns to show effects. Each cell of a matrix is split into an outer and an inner region to represent two casual relation groups. Color saturation is used to represent the strength of the corresponding relations. Another usage of a matrix view is presented by Xie et al. [42] IV. They use a matrix with a custom glyph design to represent passes in football.

### 4.3. Timeline-based Visualizations

A timeline-based visualization is a visual representation of event sequence data that has a fixed time axis (usually the x-axis) to arrange visual elements that represent events or derived properties in temporal order. Next, we describe several categories of timeline-based visualizations. For a summary of papers in this category refer to Table 6.

#### 4.3.1. Fixed timeline-based

The first category is called Fixed timeline-based visualizations and is depicted in Figure 11. Visualizations in this category present event sequences as elements that are aligned along the timeline.

Bose et al. [26] PM use a representation of event sequences, in which each event has one timestamp (beginning or the end of event execution). This visualization
is called dotted chart. An event sequences is shown as a sequence of circular glyphs colored by event type. Each sequence is arranged as a row. Similar visualization are used in several visual systems [43, 39, 53, IV], with different glyph shapes including triangles [53, 44, IV] and rhombus [54, IV]. Other works also use Fixed timeline-based visualizations [45, 44, IV], but combine them with a Duration timeline-based visualization that is described as a next category.

Some works use a Fixed timeline-based visualizations to represent aggregated event sequences. The visualization system by [53, IV] shows the colored vertical rectangles on the timeline to visualize all events and when they happened on the timeline. de Leoni et al. [60, PM] use a timeline view with circular glyphs to show the events that are planned to be performed in the future. Grouping of several circles creates a bigger circle that is used as a pie chart to represent if some of the contained events were already executed.

### 4.3.2. Duration timeline-based
In this event sequence visualization type, each event is mapped to a timeline with duration information encoded as a length of a glyph (see Figure 12).

The authors of [55, 27, 46, IV] and [58, PM] use a horizontal timeline to arrange the rectangular visual elements colored according to their event type. In this way, they convey the time from the beginning to the end of the event, often to capture the duration of an activity execution. Another group of works [9, 52, IV] encodes events as a color-coded rectangle and aligns them to the vertical timeline. Rosenthal et al. [50, IV] describe a visualization system that shows problems and their severity with a color of a standard sized boxes, while the lead time (completion time) is shown as a line extending past the box instead of the enlarged box size. Other works, instead of showing individual events from event sequence data, focus on conveying some information derived from event sequences on the timeline. Richter et al. [59, PM] use a duration timeline-based visualization to show when particular transitions between events were performed.

The two categories discussed so far are often considered to be pure event-based visualizations. They convey information about the time of when an event happened as one dimension of the visualization of the event sequence. The following two categories deal with more complex visualizations, in which additional information is integrated.

4.3.3. Converging-diverging timeline-based

This type of timeline visualization illustrated in Figure 13 displays event sequences as converging and diverging lines. Each of these lines corresponds to an event sequence. The lines that lay close together usually correspond to some common characteristic such as location proximity, same state, or joint resource executing these events.
This category has only been discussed in the field of information visualization. Some works build on Marey graphs to represent event sequences. Xu et al. [47, IV] use a Marey graph to represent the assembly line performance at a factory, showing progression of events as vertical lines that follow a sequence of predefined events along the graph. This visualization uses color to highlight those parts of the lines that represent delays. Another Marey graph based visualization is shown in [39, IV], using color to highlight the event types for the use case of disease progression pathways.

Baumgartl et al. [49, IV] use a vertical Converging-diverging timeline-based visualization in a system for the analysis of pathogen transmission in hospitals. The horizontal interactive timeline contains the lines representing patients and their health status. The lines that converge show the patient contacts. The visualization by Liu et al. presented in [48, IV] uses lines to represent the characters of a story and events that involve them as converging lines of several lines. Another Converging-diverging timeline-based visualization by Reda et al. [57, IV] shows the evolution of how social groups evolve with time. The lines that stretch horizontally represent the group membership, while diverging and converging of the lines convey the change of the group membership.

4.3.4. Evolution timeline-based

The Evolution timeline-based visualization (Figure 14) conveys event sequence information as a density chart arranged along the timeline. Event types are often distinguished by color and the area size is used to show additional characteristics.

Wu et al. [50, IV] introduce the Opinionflow visualization that shows the opinion diffusion on social media as a sankey-based-density map visualization. In this visualization, where the event sequences are aggregated in several horizontal timelines. The size of the area shows the number of opinion expressions on social media and the color shows the type
of opinion. Sung et al. [8, IV] present a Evolution timeline-based view as part of their visualization system. This view is called a theme river and it summarizes a number of events as a color-highlighted area displayed over a horizontal timeline. A similar view is presented as part of the ChronoCorrelator visualization system [54, IV]. It includes a density plot showing the number of all events that happen as an area arranged with a horizontal timeline.

The papers in this category focus on displaying the features derived from event sequence data in the timeline-based graphs. Suriadi et al. [61, 62, PM] use line graphs to show how different event sequence parameters change with time. These graphs allow for inspection of the time-related changes in the event sequences.

4.3.5. Combinations

In this section we discuss visualization systems that combine several visualization categories in their designs.

The visualizations presented in [44, 45, IV] show a combination of Fixed timeline-based and Duration timeline-based visualization systems. They use triangular [44, IV] and circular [45, IV] glyphs to represent the individual events, and range-plot visual elements to convey the information about events with a duration. The ToPIN visualization [8, IV] captures the online learners behavior with a ToPIN timeline-based visualization as a combination of Fixed timeline-based and Duration timeline-based. In this visualization different events from event sequences are grouped in the rectangles that convey the average time range of when the events grouped to this rectangle happened. Additionally, these rectangles connected with a timeline axis with curved lines show the exact timestamps of individual events that they contain.

The LiveGantt visualization [51, IV] combines Duration timeline-based and Evolution timeline-based designs. The visual elements are placed on the horizontal timeline with the length representing the duration of the event, and the height representing the number of events of the same type that occurred at a given time. The Coreflow visualization [29, IV] presents the main visualization
Table 7: Selection of papers for timeline-based visualizations

| Node-link   | Flow          | Treemap       |
|-------------|---------------|---------------|
| Law et al.  | Guo et al.    | Jang et al.   |
| Vrotsou et al. | Jang et al.   | Jang et al.   |
| Vrotsou et al. | Liu et al.    |               |
| Krause et al. |               |               |
| Wongphasawat et al. | Guo et al. |               |

that shows the tree-like patterns discovered from event sequences. Coreflow is presented in two ways, as an icicle plot and as a node-link visualization showing branching patterns of event sequence data. Both visualizations also preserve aggregated timestamps for each event. While this visualization is also showing a hierarchy-based representation, it is also a Duration timeline-based and Evolution timeline-based visualization.

4.4. Hierarchy-based

Hierarchy-based visualizations convey hierarchical relationships found in event sequence data. These visualizations group sequences by their prefix similarity. A summary of contributions that design visualization systems using this visual representation is shown in Table 7.

4.4.1. Node-link hierarchy-based

This type of visualizations (Figure 15) captures information about event sequences as a node-link diagram, where nodes usually represent events and the links represent transition between events.
In many visualization designs an elementary node-link diagrams is used, e.g. in [53, 63, IV] and [26, 69, 60, PM]. Law et al. [63, IV] and Bose et al. [26, PM] use color to distinguish different event types. Also Leite et al. [53, IV] and de Leoni et al. [69, 60, PM] use various glyph designs for event types additionally to color.

A second group of Node-link hierarchy-based visualization designs consider the node size, link size or both to show additional dimension in the data. For instance, Vrotsou et al. [28, 29, IV] use the size of a link in node-link diagram to represent the frequency of the corresponding transition between two events. The visualization design by Krause et al. [64, IV] takes the size of the links, size of nodes, and the color of nodes to convey the information about the frequencies and additional characteristics of a cohort of event sequence data.

While many hierarchy-based visualizations aggregate the event sequence data, the work by Vrotsou et al. [9, IV] defines a node-link based visualization that only groups sequences containing the same sub-sequence of activities using user defined queries. This visualization shows hierarchically grouped sequences that are still individually identifiable.

4.4.2. Flow hierarchy-based

Flow hierarchy-based visualizations aggregate event sequences in the sankey based visualization that shows the hierarchical dependencies in the event sequence data (see Figure 16). These visualizations use the horizontal or vertical layouts with sequentially ordered elements, in which the width of a transition shows its frequency.

Liu et al. [29, IV] show Coreflow and its icicle plot as Flow hierarchy-based visualization. This vertical design shows event sequences grouped by prefixes starting from the top of the visualization where similar prefixes are laid
out proceeding downward to more specific events. The width of the events show the number of transition. Similar design, but laid out horizontally is used in the Outflow visualization [65, IV]. In this visualization, the patterns are grouped and colored by an outcome by the scale green to red for a positive or a negative outcome. The EventThread system [66, IV] extends the EventFlow visualization and its icicle plot design with color to highlight the different event types. Jang et al. [10, IV] present the MotionFlow system that uses glyphs to identify events, and the transition width and color to show separate aggregated event sequences.

Several special cases of Flow hierarchy-based visualizations are described in [66, 67, 63, 68, IV]. Guo et al. [66, 68, IV] present a system for the analysis of event sequence data that splits event sequences into stages. Their systems use nodes of a horizontally laid out graph to represent information about stages of event sequences. These nodes are connected with semi-transparent lines that use width to indicate the frequency of a certain path. Law et al. [63, IV] show the MAQUI system in which the workspace view illustrates patterns as the vertical colored rectangles. Event transitions are shown with the gray horizontal rectangles using height to indicate the frequency of a transition. Finally, the system by Liu et al. [67] shows the icicle view for patterns for different groups of users. Unlike the standard icicle plot, this visualization shows separate patterns for each group.

4.4.3. Treemap hierarchy-based

This category includes visualizations of the event sequences that are aggregated into a treemap view (Figure 17). This visualization uses nested figures such as rectangles to represent hierarchical dependencies in the data.

The Motionflow system by Jang et al. [10, IV] features a treemap view to complement a Flow hierarchy-based view. It uses a nested rectangle layout, coloring the rectangles in the same way as the paths in the flow.
visual representations. This view then shows a hierarchical relationship of which patterns exists in the event sequence data.

4.5. Sankey-based

Sankey-based visualizations show event sequence data as a graph of transitions between event types. Articles that employ sankey-based visualization designs are summarized in Table 8.

4.5.1. Node-link sankey-based

The category of Node-link sankey-based visualizations use a graph layout in which the events are represented as nodes and the transitions are shown as links (see Figure 18). A node-link visualization can convey additional information by
the size of the node or the width of links such as frequency of occurrence of a
specific type of transition.

The authors of [26, 78] use node-link diagrams as an aggregated model derived from event sequence data. They annotate the transitions and nodes with frequencies. Song et al. [69] show the social network derived from event sequences as a node-link based representation. They use colors to distinguish event types. Baumgartl et al. [49] use a node-link representation to show the contact network view for the data of pathogen transmission between patients in the hospital. They use color to highlight the patient status and their contacts. The ViSeq visualization [70] defines a node-link design that arranges the events vertically and the transitions as arcs.

4.5.2. Extended node-link sankey-based

Extended node-link sankey-based visualizations combine designs that convey information beyond the transitions between event types and their frequencies (see Figure 19). These include characteristics such as the type of transition, the distribution of events as a single node, and nodes with different semantics (such as Petri nets [79]).

Some works distinguish transition types in the node-link visualizations. Leoni et al. [80] show the node-link graph that describes the Declare rules applying to pairs of events. They add arrows, circles, and other glyphs on top of transitions to indicate the rule type as a transition. Bolt et al. [32] use colors of transitions and dashed lines to represent differences between data and some predetermined rules.

There are works that describe node designs that convey additional information. Leite et al. [53] uses the
color and shape of the events to show the frequency and the type of the node. Jang et al. [10, IV] define water-drop nodes in Motionflow that contain representations of the human motion patterns. The transitions in this node-link diagram show the change of a person’s position. So-called Extended Compliance Rule Graphs by Knuplesch et al. [81, PM] visualize the compliance of an event sequences with the model. These graphs use different node shapes to represent various node types. Song et al. [69, PM] adopt Petri nets and color the transitions with two-color schema (color of a circle and its border) to show performance measures.

Low et al. [58, PM] use pie charts as nodes to show the extent of swap of resources that work on events. EmoCo [46, IV] define views that convey additional information about the aggregated events as colored pie charts. Wynn at al. [83, PM] present Petri nets in the three-dimensional space including bar charts on the nodes to represent additional information, such as injury severity. Munoz-Gama et al. [79, PM] also use Petri nets to support visual conformance checking, as much as van Dongen et al. [82, PM] who convey additional information as horizontal bar charts in the nodes and various transition designs (solid lines, dashed lines, etc.).

4.5.3. Flow sankey-based

In the Flow sankey-based visualizations the event sequences are aggregated showing the transitions of events in the event sequences (see Figure 20). These visualizations are horizontally arranged from left to right according to the temporal order of events, and the width of transitions show the frequency of transitions.

Nguyen et al. [71, IV] present a sankey-based flow visualization, in which each event type is associated with a histogram to show its variability across processes. Gotz et al. [72, IV] use a sankey diagram to aggregate the flow of events representing them as colored rectangles and their size scaled to their
aggregated frequency. Jin et al. [41, IV] arrange cause-effect relations in a sankey diagram. The RoseRiver visualization system [73, IV] shows the change in groups of event sequences as a sankey-inspired diagram. For each step, color indicates the event sequence group and height frequency. The Loyaltracker [74, IV] visualizes the loyalty of search engine users. They use color hue to differentiate between three groups users: strictly loyal, medium loyal, and barely loyal. The flow of users represents those that join or leave a search engine at each point in time. The Opinion Flow system [50, IV] shows the information about the flow of interconnected events as a sankey-based-density map, where the color intensity and hue capture the state of the system and the separate flows indicate the frequency of a particular opinion discovered from raw event sequence. On demand, parts of the raw event sequence with the directly follows relations can be shown. The EmoCo visualization [46, IV] is a three-steps sankey diagram showing facial-emotion change along the sequence of particular words.

The STBins visualization system [75, IV] is a sankey-based design for analyzing multi-thread execution log data. Event sequences are grouped in threads and visually encoded as rectangles containing circles that represent events. The transitions between circles within the thread and across threads encode the change in sequences.

Guo et al. [68, IV] use several chart-based visualizations (area charts, bar charts) in combination with a Flow sankey-based visualization to convey different characteristic of event aggregations.

### 4.5.4. Converging-diverging sankey-based

Converging-diverging sankey-based visualizations have in common that they use horizontal lines that converge and diverge showing change of state of a sequence or a group of sequences (Figure 21).

Bartolomeo et al. [31, IV] present a Sequence Braiding visualization system that uses vertically lines grouped into categories and highlighted with color. These lines...
show how the status diabetic patients change with events of food consumption.

Chou et al. [76, IV] present a similar visualization design, grouping similar sequences with the same color. Vrotsou et al. [52, IV] define a related design, but instead of arranging the events vertically by one categorical measure, each step is represented based on quantitative measure. This visualization focuses on information derived from event sequence data, and not on the sequential order.

Agarwal et al. [77, IV] present a Converging-diverging sankey-based design that represents group membership of event sequence changes over time. Line width indicates additional characteristics and bar chart display distribution information related to an event type.

4.5.5. Chord sankey-based

The Chord sankey-based visualization captures events as circles with transitions as arcs (see Figure 22).

Chen et al. [70, IV] present a Chord sankey-based representation as part of their visualization system. It is used for showing aggregated transitions of event sequence data in different time windows (weeks). The width of each flow in this chord diagram shows the frequency of this transition. The design allows for better comparison between different event sequence groups.

4.6. Other Chart-based

This group of event sequence data visualization designs use well-established visualization charts to display information derived from event sequence data. The examples of these charts are bar chart, histogram, pie chart, and sequence chart. Visualization systems often integrate a combination of several chart-based visualizations in one design. A summary of related contributions is shown in Table 9.
| Sequence | Sequence-Duration | Bar-chart | Other chart | Composition |
|----------|-------------------|-----------|-------------|-------------|
| Kwon et al. [39] | Wang et al. [84] | Unger et al. [33] | Liu et al. [67] | Vrotsou et al. [28] |
| Nguyen et al. [7] | Chen et al. [70] | Guo et al. [66] | Guo et al. [68] | Jin et al. [41] |
| Loorak et al. [30] | Xu et al. [47] | Jin et al. [41] | Xu et al. [47] | Xu et al. [33] |
| Chen et al. [70] | Unger et al. [33] | Jin et al. [41] | Xu et al. [47] | Xu et al. [47] |
| Guo et al. [66] | Xue et al. [47] | Kwon et al. [33] | Xie et al. [42] | Xu et al. [47] |
| Nguyen et al. [7] | Chen et al. [70] | Kwon et al. [33] | Xie et al. [42] | Xu et al. [47] |
| Jin et al. [41] | Law et al. [43] | Isaccs et al. [86] | Jo et al. [51] | Xie et al. [42] |
| Cappere et al. [27] | Chen et al. [33] | Bernhard [6] | Xie et al. [42] | Xu et al. [47] |

| Sequence | Sequence-Duration | Bar-chart | Other chart | Composition |
|----------|-------------------|-----------|-------------|-------------|
| IV       | CGF               |           |             |             |
| Rosenthal et al. [50] | Dortmont et al. [53] | Agarwal et al. [72] |             |             |
| IS       | DSS               |           |             |             |
| Leoni et al. [62] [PM] | Low et al. [59] | Bolt et al. [72] | Richter et al. [59] | Low et al. [59] |
| TKDE     | ENDS             |           |             |             |
| Suriadi et al. [62] [PM] |             | Lie et al. [74] |             |             |

35
4.6.1. Frequently used Chart Types

In this section we describe frequently used chart types for visualizing event sequences.

A. Sequence chart-based. In this category, events of sequences are mapped to visual elements such as different glyphs or shapes. Color is often used to show the event type (see Figure 23). Visualizations in this category are arranged horizontally or vertically, such that the order of events is easily understood. In two-dimensional layouts, lines are used to connect glyphs. Sequence chart-based visualizations are similar to Fixed timeline-based, but without the requirement to show elements arranged along the timeline.

![Figure 23: Sequence chart-based visualization](image)

The first group of sequence chart-based works focus on providing an overview of event sequences. They visualize events as sequence of circles [43, 39, IV], triangles [84, IV], rectangles [33, IV], squares [28, 67, IV], and other shapes [80, PM]. Each event sequences is shown as a separate line. Some of these visualizations can highlight a particular event type in all sequences [28, IV] or align events according to one specific event type [67, 84, IV]. These visualizations show from as few as five [84, IV] to hundreds [67, IV] of event sequences in one view. Representations in this category convey also patterns of aggregated event sequence data. Nguyen et al. [7, IV] use visual highlighting to show aggregations. Chen et al. [70, IV] distinguish different event types using different shapes.

A second group of Sequence chart-based visualizations display one or more event sequences using visual elements. Long sequences are split into several rows and glyphs are connected with lines [66, 63, 68, 41, IV]. This design is used to show details of a concrete event sequence for analysis. This representation can also be used for aggregated event sequences. MAQUI [63, IV] shows the major patterns discovered from event sequence data. Capper et al. [85, IV] present a visualization system for individual event sequences and aggregated event se-
quences, using a sequence of annotated rectangles capturing rules discovered from event sequences. Chen et al. [43 IV] introduced a visualization system that uses a Sequence chart-based representation. They use a sequence of colored rectangles to represent the aggregated event sequences. This visualization design supports interactive exploration of details related to specific patterns. These patterns are displayed as taller rectangles and aligned below the original aggregated sequence.

B. Sequence-duration chart-based. Sequence-duration chart-based visualizations map event sequences to the visual items in such a way that the size of the elements are proportional to characteristics such as the event duration (see Figure 24).

Loorak et al. [40 33 IV] present a vertical design of Sequence-duration chart-based visualization of event sequences and their duration. These sequences are arranged along the x-axis and the y-axis showing the duration of individual events in event sequence. Xu et al. [47 IV] describe a radial representation, arranging the event sequence around a circle. Rosenthal et al. [56 IV] use a one-dimensional Sequence-duration chart-based overview of event sequence data. The color of the rectangles indicate the event type and size shows the number of aggregated events.

C. Bar-chart-based. Bar charts visualize data of different categories as rectangular visual elements, where width is fixed and height represents some quantitative value (see Figure 25). Stacked bar charts offer a comparison between different subcategories of data.

Event sequence data visualization systems use bar chart views to present some distributional aspects of the
data. Some works use bar charts to show the distribution of gender [64 [72 [41 [IV]], race of participants [72 [IV]], or event frequencies [72 [41 [51 [47 [IV]]. Kwon et al. [39 [IV] provide bar charts to summarize various other characteristics of the event sequence data. Agarwal et al. [77 [IV] give an overview of the number of events per a specified time ranges. Nguyen et al. [71 [IV] use the bar chart with a chart for each sequence. One bar represents two variables, one captured by its height and the other by its color. Suriadi et al. [62 [PM] use bar charts to show the average length of queue before each event in the sequences. Low et al. [58 [PM] show the difference in resource utilization between two event sequence datasets. Bose et al. [26 uses bar charts as a additional view, in which each bar corresponding to one event and the extent its traces are aligned with a predefined model.

4.6.2. Other Chart Types

D. Histogram. A histogram is a visualization of data that is based on the bar chart. Instead of displaying bars for a categorical value, the histogram replaces them with ranges derived from a quantitative values.

Wang et al. [84 [IV] use histograms and stacked histograms to aggregate and show the distribution of event types on a daily basis. Similarly, the authors of [70 [64 [IV] use histograms to show the distribution of outcomes of an event sequence. Gotz et al. [72 [IV] and Jin et al. [41 [IV] use histograms to display age distribution in the data. Nguyen et al. [71 [IV] utilize modified histograms to show the frequency of different process variants. These modified histograms use a second x-axis with another variable connecting the two axes with shadowed lines.

E. Line Chart. Line chart displays the series of data points connected by a line in a two-dimensional graph. It shows a change of a variable over time. Xie et al. [32 [IV] use line charts to display statistics of event sequences derived from football data. Jo et al. [51 [IV] use a line chart to represent outcome changes per group of event sequences. They color slopes of this line chart use hue and saturation for an increase or decrease of value.
F. Scatter plot. Scatter plots show the distribution of values of a two-dimensional numerical variable. These values are plotted on the x- and y-axes. Nguyen et al. [71, IV] show a scatter plot of how two attributes correlate. In this way, they compare exclusive and inclusive run-time of event sequences from parallel program executions. Gotz et al. [72, IV] present a so-called scatter-and-focus visualization. This interactive visualization uses a green-red color scale for differentiating the points and for choosing the focus points for further analysis. Chen et al. [70, IV] use a scatter plot as a projection view of their visualization. They display the event sequences aggregated and sorted by similarity with respect to a predefined time period.

G. Pie chart. Pie charts are charts that slice a circle for showing proportions taken by different groups. Bolt et al. [32, PM] use pie charts to present the results of a classification algorithm for detection of differences between behavior and business rules. Liu et al. [87, PM] show a pie chart of the aggregated distribution of different causes of anomaly.

H. Heat map. Heat maps visualize quantities of data using a color scale projected onto a two-dimensional space. Isaacs et al. [86, IV] employ the heat maps to visualize program execution traces. They arrange the event sequences in rows and their time in columns. The color scheme from light blue to red identifies the delay of execution of certain events. Lie et al. [87, PM] use a heat map to highlight the columns in a matrix according to the extent of abnormal behavior present in corresponding event sequences.

I. Map. This visualization type positions visual elements on a two-dimensional map. Leoni et al. [63, PM] display individual events of event sequences on a city map. They use several glyph types (rectangles, circles) to distinguish event types.

J. Calendar. This visual representation utilizes a calendar of dates in the month to present information. Xu et al. [47, IV] use a calendar view to represent the number of faults for each day. They color squares scaled by color lightness.
**K. Density Plot.** A density plot is a smoothed version of a histogram that is used to show how the numerical variable is distributed. It is used as a building block among others by Xu et al. [47, IV], Wu et al. [50, IV], and Dortmont et al. [54, IV].

**L. Box Plot.** A Box plot is a visualization method to aggregate and display a set of numerical values. This representation displays data through quartiles, drawing a rectangle showing second and third quartiles with lines extending further to show lower and upper quartiles. Bernard et al. [6, IV] use it as part of their visualizations.

### 4.6.3. Compositions of Chart Types

A key property of technology is its compositional nature [88]. Also visualization techniques inherit this compositionality. Here, we describe works that integrate different chart-based techniques in one visualization view. Several authors use a combination of Sequence chart-based visualizations together with other chart-based visualizations.

Chen et al. [70] display individual event sequences as sequences of glyphs as in Sequence chart-based overlaying with line charts to show how quantities change with progress. Xie et al. [42, IV] use a map visualization to position elements of a Sequence chart-based. They use the map of a football pitch to show the sequence of passes that lead to a certain situation (e.g. a goal).

Low et al. [58, PM] combine a Sequence chart-based visualization to show which activities have a time shift, and a diverging bar chart to display the aggregated time shift between two event log datasets. Xie et al. [42] use a combination of Sequence chart-based and Bar-chart-based visualizations. They use the bar charts to show the aggregate current state of a system, while on each bar an event can be displayed as a circle to show individual event sequences. Isaacs et al. [86, IV] use a combination of Sequence chart-based, Bar-chart-based and heatmaps in their visualizations. The individual events are arranged by rows and columns as rectangles that are colored by hue and saturation to show how
late events are due to some predetermined criteria. Additionally, at the bottom of this view, the bar chart shows the overview of the whole event sequence while the main view only shows the chosen part.

Richter et al. [59, PM] and Zeng et al. [46, IV] use the Sequence-duration chart-based to show individual event sequences next to a line chart to show how additional quantitative characteristic changes with time of the sequence. Richter et al. [59, PM] also combines line chart and scatter plot using the same x-axis to display two different characteristics for each transitions between two event types. The scatter plot shows the duration of these transitions and a line chart for identifying when the changes in behavior occur.

Xu et al. [47, IV] use a Node-link sankey-based visualization arranged vertically. For each event type, a small density plot is displayed showing the occurrence distribution of this event in the data. Chen et al. [27, IV] use a scatter plot to arrange pie charts on the two dimensional space. Bernard et al. [6, IV] group several different charts into a group, and a sequence of instances of this group of charts to signify how the characteristic of a group of event sequences changes with time. In this design the bar charts, box plots, pie charts, matrix views are combined into a group to show the characteristics. This system was designed for a medical system for monitoring to analyse cohorts of patients.

4.7. Discussion

We analyzed contributions from the fields of process mining and information visualization using the ESeVis framework. We observe that both fields differ in terms of their emphasis on sequence representations and model representations.

Contributions focusing on instance representations are dominated by the field of information visualization. A larger share of these works are in the category of timeline-based visualizations and in the category of other chart types. The articles of our review include few works that build on matrix-based, hierarchy-based, or sankey-based visualizations of event sequences. There are very few contributions from process mining research on instance representation. Contributions on model representations are more balanced between both re-
search fields. A larger share of these works are sankey-based or in the category of other chart types. There are some hierarchy-based, some timeline-based, and few matrix-based works. Process mining contributions are to a larger share related to sankey diagrams and the category of other chart-based visualizations.

These findings highlight some strengths of the process mining field. They reflect the fact that process mining research puts a strong emphasis on discovery algorithms that generate process models such as variants of node-link diagrams or Petri nets from event sequence data \cite{2}. We observe that this focus on designing new algorithms integrates into a fertile discourse on which knowledge can be extracted from event sequence data. The findings also point to strengths of the information visualization field. This field puts a strong emphasis on the creative development of powerful visual interfaces to interact with complex analysis algorithms, often designed for the specific requirements of a given application domain. Formal notations like Petri nets are hardly considered. Our findings underline the potential for stronger synergies of research on process mining and information visualization.

Our review extends insights from previous review articles. Several frameworks on corresponding review articles have been written in the field of information visualization. McNabb et al. \cite{89} introduce a survey of surveys of information visualization. They categorize the field into topic clusters using a 2-dimensional classification schema. For the first dimension, the phase of information visualization pipeline \cite{90} is used. In the second dimension, the survey papers are categorized by the subject of the study \cite{90} (e.g. graphs and networks vs. geospace). While some of the categories mention the use of data including events (geospace+time, data-centric surveys), this survey of surveys does not point to an existing survey that reviews event sequence data visualizations.

Aigner et al. \cite{3, 91} focus their survey on time-oriented data visualizations. The survey in \cite{91} categorizes visualization designs by three categories: time, data, and representation. In their taxonomy, the notion of event data is introduced as part of the data category, showing that events represent one level of data abstraction. The recent survey, by Gue et al. \cite{25} introduces a summary
of visual analysis approaches for event sequence data. They propose a taxonomy based on research articles in the field of information visualization. They categorize visualizations into design space clusters: data scales, analysis techniques, visual representation, and interactions. Another focus of the survey is identifying and clustering the papers according to visual analytics tasks. Our survey adopted five visual representation categories from [25]. Within each of these categories, we extended the categorization into subcategories bringing a fine-granular distinction in visualization designs. None of these review articles explicitly considers contributions from process mining.

There are also several review articles on process mining. Their focus is on the comparison of process discovery algorithms, while the visual representations of discovery results are usually limited to assumed options. Weerdt et al. [92] present a multi-dimensional quality assessment of process discovery algorithms. Their quality assessment focuses on distinguishing process discovery algorithm types, and typical challenges of discovery (e.g. a problem of incorrectly detecting a loop in a model). They evaluate the results of process discovery for each of the surveyed approaches after conversion of the result to Petri nets representation. Visualization is not considered. A more recent survey article, by Augusto et al. [93] explicitly clusters articles by model languages produced by their process discovery algorithms in categories such as process trees, casual nets, BPMN models, state machines. To assess the quality of results, this survey still converts these representations into Petri nets, and does not focus on visualizations of these models. Only recently, the effective visualization of process mining results has become a subject of research [94], building on insights on the cognitive effectiveness of diagrams [95]. Contributions from information visualization provide concepts to inform this emerging research.

5. Conclusions

In this paper, we have presented a review of visualization approaches for event sequence data. To this end, we have developed the ESeVis Framework for cate-
gorizing visualizations from information visualization and process mining fields. Our findings highlight strengths of the process mining field in terms of generating formal models and the information visualization of creatively developing powerful visual interfaces to interact with complex analysis algorithms. Our findings underline the potential for stronger synergies of research on process mining and information visualization. Future research can utilize our work for identifying useful contributions from both fields, which will eventually help towards the integration of these fields.

Acknowledgement

The research by Jan Mendling was supported by the Einstein Foundation Berlin.

References

[1] C. dos Santos Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, E. E. Scalabrin, Process mining techniques and applications—a systematic mapping study, Expert Systems with Applications 133 (2019) 260–295.

[2] W. M. P. van der Aalst, Process Mining: Data Science in Action, Second Edition, Springer, 2016. doi:10.1007/978-3-662-49851-4. URL https://doi.org/10.1007/978-3-662-49851-4

[3] W. Aigner, S. Miksch, H. Schumann, C. Tominski, Visualization of Time-Oriented Data, Human-Computer Interaction Series, Springer, 2011.

[4] D. A. Keim, Information visualization and visual data mining, IEEE Transactions on Visualization and Computer Graphics 8 (1) (2002) 1–8.

[5] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, Second Edition, Springer, 2018.
[6] J. Bernard, D. Sessler, J. Kohlhammer, R. A. Ruddle, Using dashboard networks to visualize multiple patient histories: A design study on post-operative prostate cancer, IEEE Trans. Vis. Comput. Graph. 25 (3) (2019) 1615–1628.

[7] P. H. Nguyen, C. Turkay, G. L. Andrienko, N. V. Andrienko, O. Thonnard, J. Zouaoui, Understanding user behaviour through action sequences: From the usual to the unusual, IEEE Trans. Vis. Comput. Graph. 25 (9) (2019) 2838–2852.

[8] C. Sung, X. Huang, Y. Shen, F. Cherng, W. Lin, H. Wang, Exploring online learners’ interactive dynamics by visually analyzing their time-anchored comments, Comput. Graph. Forum 36 (7) (2017) 145–155.

[9] K. Vrotsou, J. Johansson, M. Cooper, Activitree: Interactive visual exploration of sequences in event-based data using graph similarity, IEEE Trans. Vis. Comput. Graph. 15 (6) (2009) 945–952.

[10] S. Jang, N. Elmqvist, K. Ramani, Motionflow: Visual abstraction and aggregation of sequential patterns in human motion tracking data, IEEE Trans. Vis. Comput. Graph. 22 (1) (2016) 21–30.

[11] F. Mannhardt, D. Blinde, Analyzing the trajectories of patients with sepsis using process mining, in: RADAR+EMISA@CAiSE, Vol. 1859 of CEUR Workshop Proceedings, CEUR-WS.org, 2017, pp. 72–80.

[12] Z. Xing, J. Pei, E. J. Keogh, A brief survey on sequence classification, SIGKDD Explor. 12 (1) (2010) 40–48.

[13] A. F. Ghaahfarokhi, G. Park, A. Berti, W. van der Aalst, Ocel standard (2000).

[14] Ieee standard for extensible event stream (xes) for achieving interoperability in event logs and event streams, IEEE Std 1849-2016 (2016) 1–50 doi: 10.1109/IEEESTD.2016.7740858
[15] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, W. M. P. van der Aalst, The prom framework: A new era in process mining tool support, in: ICATPN, Vol. 3536 of Lecture Notes in Computer Science, Springer, 2005, pp. 444–454.

[16] Disco Fluxicon, Disco. URL https://fluxicon.com/

[17] A. Gabadinho, G. Ritschard, N. S. Mueller, M. Studer, Analyzing and visualizing state sequences in r with traminer, Journal of Statistical Software 40 (4) (2011) 1–37.

[18] M. De Backer, The use of petri net theory for business process verification., Ph.D. thesis (2007).

[19] D. Deutch, T. Milo, A structural/temporal query language for business processes, J. Comput. Syst. Sci. 78 (2) (2012) 583–609.

[20] J. Prescher, C. D. Ciccio, J. Mendling, From declarative processes to imperative models, in: SIMPDA, Vol. 1293 of CEUR Workshop Proceedings, CEUR-WS.org, 2014, pp. 162–173.

[21] W. Reisig, Petri nets: an introduction, Vol. 4, Springer Science & Business Media, 2012.

[22] W. M. van der Aalst, A practitioner’s guide to process mining: Limitations of the directly-follows graph (2019).

[23] F. M. Maggi, A. J. Mooij, W. M. P. van der Aalst, User-guided discovery of declarative process models, in: CIDM, IEEE, 2011, pp. 192–199.

[24] M. Arriagada-Benítez, M. Sepúlveda, J. Munoz-Gama, J. C. A. M. Buijs, Strategies to automatically derive a process model from a configurable process model based on event data, Applied Sciences 7 (10). doi:10.3390/app7101023 URL https://www.mdpi.com/2076-3417/7/10/1023
[25] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, N. Cao, A survey on visual analysis of event sequence data, IEEE Transactions on Visualization and Computer Graphics (2021) 1–1 doi:10.1109/TVCG.2021.3100413.

[26] R. P. J. C. Bose, W. M. P. van der Aalst, Process diagnostics using trace alignment: Opportunities, issues, and challenges, Inf. Syst. 37 (2) (2012) 117–141.

[27] S. Chen, N. V. Andrienko, G. L. Andrienko, L. Adilova, J. Barlet, J. Kindermann, P. H. Nguyen, O. Thonnard, C. Turkay, LDA ensembles for interactive exploration and categorization of behaviors, IEEE Trans. Vis. Comput. Graph. 26 (9) (2020) 2775–2792.

[28] K. Vrotsou, A. Nordman, Exploratory visual sequence mining based on pattern-growth, IEEE Trans. Vis. Comput. Graph. 25 (8) (2019) 2597–2610.

[29] Z. Liu, B. Kerr, M. Doncheva, J. Grover, M. Hoffman, A. Wilson, Core-flow: Extracting and visualizing branching patterns from event sequences, Comput. Graph. Forum 36 (3) (2017) 527–538.

[30] S. J. J. Leemans, E. Poppe, M. T. Wynn, Directly follows-based process mining: Exploration & a case study, in: ICPM, IEEE, 2019, pp. 25–32.

[31] S. D. Bartolomeo, Y. Zhang, F. Sheng, C. Dunne, Sequence braiding: Visual overviews of temporal event sequences and attributes, IEEE Trans. Vis. Comput. Graph. 27 (2) (2021) 1353–1363.

[32] A. Bolt, M. de Leoni, W. M. P. van der Aalst, Process variant comparison: Using event logs to detect differences in behavior and business rules, Inf. Syst. 74 (Part) (2018) 53–66.

[33] A. Unger, N. Drager, M. Sips, D. J. Lehmann, Understanding a sequence of sequences: Visual exploration of categorical states in lake sediment cores, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 66–76.
[34] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele University 33 (2004) (2004) 1–26.

[35] H. Cooper, L. V. Hedges, J. C. Valentine, The handbook of research synthesis and meta-analysis, Russell Sage Foundation, 2019.

[36] M. Leitner, S. Rinderle-Ma, A systematic review on security in process-aware information systems—constitution, challenges, and future directions, Information and Software Technology 56 (3) (2014) 273–293.

[37] J. Mendling, Business process modeling in the 1920s and 1930s as reflected in fritz nordsieck’s phd thesis.

[38] Harzing, A.W., Publish or perish (2019).
URL https://harzing.com/resources/publish-or-perish

[39] B. C. Kwon, V. Anand, K. Severson, S. Ghosh, Z. sun, B. Frohnert, M. Lundgren, K. Ng, Dpvis: Visual analytics with hidden markov models for disease progression pathways., IEEE transactions on visualization and computer graphics.

[40] M. H. Loorak, C. Perin, N. Kamal, M. D. Hill, S. Carpendale, Timespan: Using visualization to explore temporal multi-dimensional data of stroke patients, IEEE Trans. Vis. Comput. Graph. 22 (1) (2016) 409–418.

[41] Z. Jin, S. Guo, N. Chen, D. Weiskopf, D. Gotz, N. Cao, Visual causality analysis of event sequence data, IEEE Transactions on Visualization and Computer Graphics 27 (2) (2021) 1343–1352. doi:10.1109/TVCG.2020.3030465

[42] X. Xie, J. Wang, H. Liang, D. Deng, S. Cheng, H. Zhang, W. Chen, Y. Wu, Passvizor: Toward better understanding of the dynamics of soccer passes, IEEE Transactions on Visualization and Computer Graphics 27 (2) (2021) 1322–1331. doi:10.1109/TVCG.2020.3030359
[43] Y. Chen, P. Xu, L. Ren, Sequence synopsis: Optimize visual summary of temporal event data, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 45–55.

[44] M. Monroe, R. Lan, H. Lee, C. Plaisant, B. Shneiderman, Temporal event sequence simplification, IEEE Trans. Vis. Comput. Graph. 19 (12) (2013) 2227–2236.

[45] J. Fulda, M. Brehmer, T. Munzner, Timelinecurator: Interactive authoring of visual timelines from unstructured text, IEEE Trans. Vis. Comput. Graph. 22 (1) (2016) 300–309.

[46] H. Zeng, X. Wang, A. Wu, Y. Wang, Q. Li, A. Endert, H. Qu, Emoco: Visual analysis of emotion coherence in presentation videos, IEEE Trans. Vis. Comput. Graph. 26 (1) (2020) 927–937.

[47] P. Xu, H. Mei, L. Ren, W. Chen, Vidx: Visual diagnostics of assembly line performance in smart factories, IEEE Trans. Vis. Comput. Graph. 23 (1) (2017) 291–300.

[48] S. Liu, Y. Wu, E. Wei, M. Liu, Y. Liu, Storyflow: Tracking the evolution of stories, IEEE Trans. Vis. Comput. Graph. 19 (12) (2013) 2436–2445.

[49] T. Baumgartl, M. Petzold, M. Wunderlich, M. Hohn, D. Archambault, M. Lieser, A. Dalpke, S. Scheithauer, M. Marschollek, V. M. Eichel, N. T. Mutters, H. Consortium, T. V. Landesberger, In search of patient zero: Visual analytics of pathogen transmission pathways in hospitals, IEEE Transactions on Visualization and Computer Graphics 27 (2) (2021) 711–721. doi:10.1109/TVCG.2020.3030437.

[50] Y. Wu, S. Liu, K. Yan, M. Liu, F. Wu, Opinionflow: Visual analysis of opinion diffusion on social media, IEEE Trans. Vis. Comput. Graph. 20 (12) (2014) 1763–1772.
[51] J. Jo, J. Huh, J. Park, B. H. Kim, J. Seo, Livegantt: Interactively visualizing a large manufacturing schedule, IEEE Trans. Vis. Comput. Graph. 20 (12) (2014) 2329–2338.

[52] K. Vrotsou, A. Ynnerman, M. Cooper, Are we what we do? exploring group behaviour through user-defined event-sequence similarity, Inf. Vis. 13 (3) (2014) 232–247.

[53] R. A. Leite, T. Gschwandtner, S. Miksch, E. Gstrein, J. Kuntner, NEVA: visual analytics to identify fraudulent networks, Comput. Graph. Forum 39 (6) (2020) 344–359.

[54] M. A. M. M. van Dortmont, S. van den Elzen, J. J. van Wijk, Chronocorrelator: Enriching events with time series, Comput. Graph. Forum 38 (3) (2019) 387–399.

[55] Y. Han, A. Rozga, N. Dimitrova, G. D. Abowd, J. T. Stasko, Visual analysis of proximal temporal relationships of social and communicative behaviors, Comput. Graph. Forum 34 (3) (2015) 51–60.

[56] P. Rosenthal, L. Pfeiffer, N. H. Müller, P. Ohler, Visruption: Intuitive and efficient visualization of temporal airline disruption data, Comput. Graph. Forum 32 (3) (2013) 81–90.

[57] K. Reda, C. Tantipathananandh, A. E. Johnson, J. Leigh, T. Y. Berger-Wolf, Visualizing the evolution of community structures in dynamic social networks, Comput. Graph. Forum 30 (3) (2011) 1061–1070.

[58] W. Z. Low, W. M. P. van der Aalst, A. H. M. ter Hofstede, M. T. Wynn, J. D. Weerdt, Change visualisation: Analysing the resource and timing differences between two event logs, Inf. Syst. 65 (2017) 106–123.

[59] F. Richter, T. Seidl, Looking into the TESSERACT: time-drifts in event streams using series of evolving rolling averages of completion times, Inf. Syst. 84 (2019) 265–282.
[60] M. de Leoni, M. Adams, W. M. P. van der Aalst, A. H. M. ter Hofstede, Visual support for work assignment in process-aware information systems: Framework formalisation and implementation, Decis. Support Syst. 54 (1) (2012) 345–361.

[61] S. Suriadi, C. Ouyang, W. M. P. van der Aalst, A. H. M. ter Hofstede, Event interval analysis: Why do processes take time?, Decis. Support Syst. 79 (2015) 77–98.

[62] S. Suriadi, M. T. Wynn, J. Xu, W. M. P. van der Aalst, A. H. M. ter Hofstede, Discovering work prioritisation patterns from event logs, Decis. Support Syst. 100 (2017) 77–92.

[63] P. Law, Z. Liu, S. Malik, R. C. Basole, MAQUI: interweaving queries and pattern mining for recursive event sequence exploration, IEEE Trans. Vis. Comput. Graph. 25 (1) (2019) 396–406.

[64] J. Krause, A. Perer, H. Stavropoulos, Supporting iterative cohort construction with visual temporal queries, IEEE Trans. Vis. Comput. Graph. 22 (1) (2016) 91–100.

[65] K. Wongsuphasawat, D. Gotz, Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization, IEEE Trans. Vis. Comput. Graph. 18 (12) (2012) 2659–2668.

[66] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, N. Cao, Eventthread: Visual summarization and stage analysis of event sequence data, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 56–65.

[67] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, A. Wilson, Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths, IEEE Trans. Vis. Comput. Graph. 23 (1) (2017) 321–330.
[68] S. Guo, Z. Jin, D. Gotz, F. Du, H. Zha, N. Cao, Visual progression analysis of event sequence data, IEEE Trans. Vis. Comput. Graph. 25 (1) (2019) 417–426.

[69] M. Song, W. M. P. van der Aalst, Towards comprehensive support for organizational mining, Decis. Support Syst. 46 (1) (2008) 300–317.

[70] Q. Chen, X. Yue, X. Plantaz, Y. Chen, C. Shi, T. Pong, H. Qu, Viseq: Visual analytics of learning sequence in massive open online courses, IEEE Trans. Vis. Comput. Graph. 26 (3) (2020) 1622–1636.

[71] H. T. Nguyen, A. Bhatele, N. Jain, S. P. Kesavan, H. Bhatia, T. Gamblin, K. Ma, P. Bremer, Visualizing hierarchical performance profiles of parallel codes using callflow, IEEE Trans. Vis. Comput. Graph. 27 (4) (2021) 2455–2468.

[72] D. Gotz, J. Zhang, W. Wang, J. Shrestha, D. Borland, Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation, IEEE Trans. Vis. Comput. Graph. 26 (1) (2020) 440–450.

[73] W. Cui, S. Liu, Z. Wu, H. Wei, How hierarchical topics evolve in large text corpora, IEEE Trans. Vis. Comput. Graph. 20 (12) (2014) 2281–2290.

[74] C. Shi, Y. Wu, S. Liu, H. Zhou, H. Qu, Loyaltracker: Visualizing loyalty dynamics in search engines, IEEE Trans. Vis. Comput. Graph. 20 (12) (2014) 1733–1742.

[75] J. Qi, V. Bloemen, S. Wang, J. J. van Wijk, H. van de Wetering, Stbins: Visual tracking and comparison of multiple data sequences using temporal binning, IEEE Trans. Vis. Comput. Graph. 26 (1) (2020) 1054–1063.

[76] J. Chou, Y. Wang, K. Ma, Privacy preserving visualization: A study on event sequence data, Comput. Graph. Forum 38 (1) (2019) 340–355.

[77] S. Agarwal, F. Beck, Set streams: Visual exploration of dynamic overlapping sets, Comput. Graph. Forum 39 (3) (2020) 383–391.
[78] K. Winter, F. Stertz, S. Rinderle-Ma, Discovering instance and process spanning constraints from process execution logs, Inf. Syst. 89 (2020) 101484.

[79] J. Munoz-Gama, J. Carmona, W. M. P. van der Aalst, Single-entry single-exit decomposed conformance checking, Inf. Syst. 46 (2014) 102–122.

[80] M. de Leoni, F. M. Maggi, W. M. P. van der Aalst, An alignment-based framework to check the conformance of declarative process models and to preprocess event-log data, Inf. Syst. 47 (2015) 258–277.

[81] D. Knuplesch, M. Reichert, A. Kumar, A framework for visually monitoring business process compliance, Inf. Syst. 64 (2017) 381–409.

[82] B. F. van Dongen, J. D. Smedt, C. D. Ciccio, J. Mendling, Conformance checking of mixed-paradigm process models, Inf. Syst. 102 (2021) 101685.

[83] M. T. Wynn, E. Poppe, J. Xu, A. H. M. ter Hofstede, R. Brown, A. Pini, W. M. P. van der Aalst, Processprofiler3d: A visualisation framework for log-based process performance comparison, Decis. Support Syst. 100 (2017) 93–108.

[84] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, M. S. Smith, Temporal summaries: Supporting temporal categorical searching, aggregation and comparison, IEEE Trans. Vis. Comput. Graph. 15 (6) (2009) 1049–1056.

[85] B. C. M. Cappers, J. J. van Wijk, Exploring multivariate event sequences using rules, aggregations, and selections, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 532–541.

[86] K. E. Isaacs, P. Bremer, I. Jusufi, T. Gamblin, A. Bhatele, M. Schulz, B. Hamann, Combing the communication hairball: Visualizing parallel execution traces using logical time, IEEE Trans. Vis. Comput. Graph. 20 (12) (2014) 2349–2358.
[87] C. Liu, H. Xiong, S. Papadimitriou, Y. Ge, K. Xiao, A proactive workflow model for healthcare operation and management, IEEE Trans. Knowl. Data Eng. 29 (3) (2017) 586–598.

[88] W. B. Arthur, The nature of technology: What it is and how it evolves, Simon and Schuster, 2009.

[89] L. McNabb, R. S. Laramee, Survey of surveys (sos) - mapping the landscape of survey papers in information visualization, Comput. Graph. Forum 36 (3) (2017) 589–617.

[90] S. K. Card, J. D. Mackinlay, B. Shneiderman, Readings in information visualization - using vision to think, Academic Press, 1999.

[91] W. Aigner, S. Miksch, W. Müller, H. Schumann, C. Tominski, Visualizing time-oriented data - A systematic view, Comput. Graph. 31 (3) (2007) 401–409.

[92] J. D. Weerdt, M. D. Backer, J. Vanthienen, B. Baesens, A multidimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs, Inf. Syst. 37 (7) (2012) 654–676. doi:10.1016/j.is.2012.02.004 URL https://doi.org/10.1016/j.is.2012.02.004

[93] A. Augusto, R. Conforti, M. Dumas, M. L. Rosa, F. M. Maggi, A. Marrella, M. Mecella, A. Soo, Automated discovery of process models from event logs: Review and benchmark, IEEE Trans. Knowl. Data Eng. 31 (4) (2019) 686–705.

[94] J. Mendling, D. Djurica, M. Malinova, Cognitive effectiveness of representations for process mining in: A. Polyvyanyy, M. T. Wynn, A. V. Looy, M. Reichert (Eds.), Business Process Management - 19th International Conference, BPM 2021, Rome, Italy, September 06-10, 2021, Proceedings, Vol. 12875 of Lecture Notes in Computer Science, Springer, 2021,
[95] M. Malinova Mandelburger, J. Mendling, Cognitive diagram understanding and task performance in systems analysis and design, MIS Quarterly 45 (4) (2021) 2101–2157.