VESPa: A Pattern-based Visual Query Language for Event Sequences

Florian Haag, Robert Krüger and Thomas Ertl

Institute for Visualization and Interactive Systems, University of Stuttgart, Universitätsstraße 38, 70569, Stuttgart, Germany

Keywords: Visual Query Language, Semantic Movement Analysis, Temporal Filtering, Timeline Queries.

Abstract: Movement data can often be enriched with additional information that enables analysts to ask new questions, for instance about POIs visited and meetings that imply interactions between persons. Information on spatio-temporal events such as visits or meetings can be especially valuable for digital forensics, marketing analysis, and urban planning. Most existing query languages for movement data, however, do not take that additional information into account. We address this gap by proposing VESPa, a pattern-based graphical query language to express, check, and refine hypotheses about spatio-temporal event sequences. Using VESPa, the analyst can sketch abstract assumptions and use the pattern to query the data for matches. The applicability of our approach is demonstrated in two case studies with different datasets. We also report on a small user study in which several construction and comprehension tasks were successfully solved in an interactive implementation of the concept.

1 INTRODUCTION

Increasing amounts of movement data are collected, and recording precision is constantly rising. Analyses of that data can reveal important insights, for example, to improve urban planning, transport efficiency, and digital forensics. While data exploration is an important task to reveal the unknown, analysts often have specific questions about the data. Regarding urban planning tasks, an analyst may be interested in checking whether a certain suburb is well connected to the inner city area, or more precisely, if there are any times of the day when roads are congested and the daily commute takes particularly long. To improve sales and quality of service, it is of interest to understand indoor behavior patterns such as the flow of customers during their stay in a mall, or visitors exploring different exhibits in a museum. Also, concerning surveillance tasks, spatio-temporal happenings need to be verified. Here, especially gatherings are of interest, that is, when several persons meet at a specific time at a specific place. More abstractly, each of these examples revolves around a sequence of consecutive spatio-temporal events that involve a single or multiple persons. Figure 1 exemplarily illustrates data of three persons' movement paths and the event sequences therein.

Detecting such situations can be complex, as multiple conditions like varying timespans, different places, persons, and orderings of events need to be

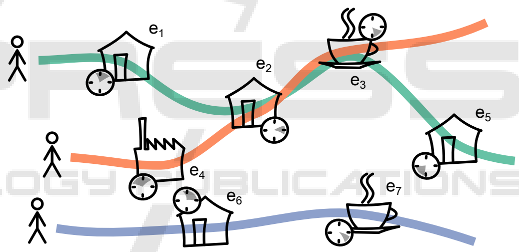


Figure 1: A sketch of a movement dataset that stores the consecutive stays and meetings of various people at specific locations and time as sequences of events. For instance, the person in the middle has the event sequence $e_4 \rightarrow e_2 \rightarrow e_3$ and meets the person on top in e_2 and e_3.

expressed. This is especially true for domain experts in the aforementioned domains who have no programming or database experience. Visual query notations can help to hide this complexity under an easier to understand set of visual items. To the best of our knowledge, none of the existing graphical query notations for event data (e.g. by Wongsuphasawat et al. (2012) or Zgraggen et al. (2015)) visualize meetings of multiple persons at certain points of interest.

To address this gap, we propose a new notation to visually express such spatio-temporal event sequences of one or more persons. Once defined, the visual event sequence can be used as a graphical representation of a hypothesized sequence of events. Likewise, it can be used as a pattern to filter the data and find any instances of the presumed event

Haag, F., Krüger, R. and Ertl, T.
VESPa: A Pattern-based Visual Query Language for Event Sequences.
DOI: 10.5220/0005716900480059
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 2: IVAPP*, pages 50-61
ISBN: 978-989-758-175-5

sequence, thereby confirming the hypothesis. Based on the results, the event sequence pattern can also be iteratively refined in a verification loop, based on the sensemaking process by Pirolli and Card (2005). Our approach is especially helpful for domain experts of an application field (e.g. urban planning, digital forensics) who are not familiar with formal text-based query languages such as SQL. Furthermore, users who could express their queries in SQL might still benefit from a visual representation to facilitate error checking and communication.

After discussing related work in the fields of event data models, event visualization, and event filtering in Section 2, we present VESPa, our Visual Event Sequence Pattern notation, in Section 3 and justify our design choices. We demonstrate the usefulness of VESPA with several pattern examples, two case studies (Section 5), and a small user study (Section 6), both using a prototypical implementation (Section 4).

2 RELATED WORK

We have looked at various options to represent concrete event sequences and to visualize the events found therein, as well as related spatio-temporal data. The findings from that literature survey provide a basis to discuss possible query visualizations for such event sequences.

2.1 Modeling of Event Data

There are several variations of modeling event-based data. In basic event logs, events are usually associated with a timestamp and carry some structured information on the event (Makanju et al., 2011; Gaaloul et al., 2004). In the case of multiple logging sources, that information can also include a hint about the origin of the event message (Abela et al., 1999).

For the purpose of representing action sequences described in texts, a timeline of events can be defined as a series of intervals that are defined by an action and related resources (Do et al., 2012). Similarly, events are defined with respect to sensor data (Atrey et al., 2006; Heydekorn et al., 2011; Kim and Giunchiglia, 2012), where a timespan can be supplied as an additional attribute (Atrey et al., 2005).

Entries in spatio-temporal event logs can be associated with timespans and locations (Peuquet and Duan, 1995). Such entries may include data that implicitly or explicitly points out a connection between related events (Huang et al., 2008; Jiang et al., 2011).

All of these works do not involve any visualization or querying of events, and therefore are not comparable to our work. However, we will base our own model of event data on these works.

2.2 Visualization of Event Data

Temporal event data are frequently visualized in a timeline, such as by lines expressing states (Plaisant et al., 1996). These data are commonly split up to distinguish different persons (Kumar et al., 1998) or event types (Tao et al., 2012; Heydekorn et al., 2011). Additional information on single events or on sets of events can be encoded into the timeline (Krstajić et al., 2011; Fischer et al., 2012). In some cases, the lines still conform to a time axis, but are reshaped to express some information themselves (Havre et al., 2000; Guo et al., 2014).

When looking at the visualization of spatio-temporal events in particular, it becomes apparent that the spatial and the temporal dimensions are often expressed in separate, but linked views (Fischer et al., 2012; Marcus et al., 2011). Alternative approaches use a space-time-cube (Kapler and Wright, 2005; Tominski et al., 2012), or they integrate aggregated spatial data into the temporal visualization (Guo et al., 2011). Moreover, additional visual elements can indicate the temporal properties of data shown in a spatial view (Sun et al., 2014).

Rather than the exact geographical reality, a semantic view on locations (e.g. location names or categories rather than coordinates) can be extracted from geographic databases or generated from social networks (Parent et al., 2013; Krüger et al., 2014). The link between the geographical locations and the logical locations can be displayed (Zhu et al., 2013), but in some cases, the primary interest lies in the semantic locations (Nguyen et al., 2007; Westermann and Jain, 2007), or they are even the only data available or cleared for use (Andrienko et al., 2013).

While all of the approaches mentioned above show event sequences, they do not help to express and visualize queries on them.

2.3 Filtering of Event Data

A variety of approaches for visually expressing generic filter queries have been proposed in the past (Shneiderman, 1994; Seifert, 2011; Soylu et al., 2013; Russell et al., 2008). There are specialized query visualizations for spatial data, some using symbolic representations of geographical relationships (Morris et al., 2004; Wu et al., 2013) and others working directly on concrete maps (Kumar et al., 2013). The symbolic representations lend themselves to use cases that work with logical locations.

For selectively finding particular pieces of information in spatio-temporal data, the spatial and the temporal dimensions are split up in some concepts (Boyandin et al., 2011; Certo et al., 2013; Krüger et al., 2013). Also, a graphical notation that expresses exact relative relationships between locations or areas and timespans has been proposed (Bonhomme et al., 1999). The only other visual approach that we are aware of displays the query restrictions in a space-time-cube (D'Ulizia et al., 2012). Some of these concepts can be further abstracted to replace the geographical association with restrictions based on logical locations (Certo et al., 2013), and time-related works provide some further ideas on how to visually represent certain features required for temporal queries (Monroe et al., 2013).

So far, only few approaches specifically deal with the visual specification of queries for event sequences, which could be used for finding particular event patterns and validating hypotheses about event sequences. Those that do only visualize the event sequence, but they do not emphasize the query structure and its restrictions in a visual way other than aligning sequential states horizontally or vertically (Plaisant et al., 1996; Gotz and Stavropoulos, 2014; Wongsuphasawat et al., 2012). Only few concepts represent series of events to find on a timeline (Fails et al., 2006), or as nodes with temporal constraints (Dionisio and Cárdenas, 1996), rather than just displaying results in a visual way (Plaisant et al., 1996). Still, relations between distinct objects are usually not supported (Zgraggen et al., 2015). When they are, these are expressed by matching colors in otherwise visually disjoint elements (Jin and Szekely, 2009).

As opposed to the aforementioned work, we focused on the graphical representation of event sequence patterns of multiple persons. Rather than just recognizing similar sequences (Plaisant et al., 1996), we are looking for actual overlaps that allow for an interaction of persons. Moreover, we want to visually express the overlaps of event sequences by actual connections rather than just by matching properties such as node colors (Jin and Szekely, 2009) to make the connection explicit.

3 QUERY NOTATION

We contribute a new visual notation to represent patterns of event sequences. The notation can be used to express hypotheses and it can be executed as a data query, for example, to answer which kinds of locations people meet at before going to a restaurant together when evaluating the layout of a shopping mall. While this sequence of events can still be described relatively easily, queries can often become more complex: "Find all pairs of persons who travel together from one place to another place, where they arrive after 3 PM, after one of them has first visited an arbitrary location and then an ice cream parlor, while the other one has arrived from another location, where he or she has not met the first person." still describes a relatively simple sequence of events, yet the text is already quite long and possibly confusing. Queries of the same or a higher complexity can also be required, for instance, in digital forensics, when investigators try to find out whether people meet after having visited certain locations that are connected to a crime.

In the above example and the aforementioned literature (Makanju et al., 2011; Gaaloul et al., 2004), certain basic components of the query become apparent: There are several distinct *persons*. These persons stay at different locations, and the time at which they stayed there is relevant to determine whether the persons might possibly have met. Therefore, such a stay can be called an *event*. Furthermore, the *movements* of each person between the respective locations are considered, connecting events to *event sequences*. Finally, *restrictions* can be imposed on the elements mentioned so far, such as the type of a location, or an arrival time.

Based upon these components, we have defined a minimal set of visual elements. These elements can be connected in a node-link diagram, which we call Visual Event Sequence Pattern (VESPa). This visualization has two purposes:

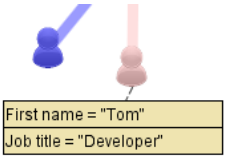
1. It can be used to express a hypothesis about the event sequences of one or more known or unknown persons.
2. It can be executed as a database query, where each element from the event sequence pattern is mapped to a database item.

3.1 Query Elements

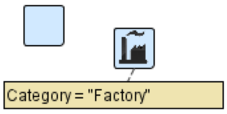
With the aforementioned goals and requirements, VESPa consists of the following elements:

Person Node: A person node as depicted in Figure 2(a) is a placeholder for a movable subject, such as a person.

Event Node: An event node, shown in Figure 2(b), is a single node in an event sequence pattern, like events found in related work (Fails et al., 2006; Plaisant et al., 1996; Dionisio and Cárdenas, 1996). In spatio-temporal contexts, it denotes a stay of one or more persons at a particular location and timespan.

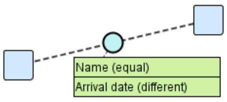


(a) Two persons, one of them has some restrictions.



(b) Two event nodes, one of them has a restriction.

(c) A transition. The central dot provides access to a settings UI.



(d) A comparison edge, configured to require equal names and different dates of arrival.

Figure 2: The basic visual elements used in VESPa.

Transition: A transition unidirectionally connects two event nodes (Figure 2(c)), comparable to the TimeSpan bars from PatternFinder (Fails et al., 2006). It represents how persons move on from one event node to another.

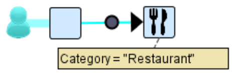
Restriction Tag: A restriction tag expresses an absolute condition that can be imposed on a person node or on an event node. Examples are depicted in Figures 2(a) and 2(b).

Comparison Edge: A comparison edge connects two person nodes, two event nodes (as shown in Figure 2(d)), or a person node and an event node, to enforce a relationship between any of their attributes. For instance, attribute values of two such nodes can be required to be equal or different. We have chosen to add this additional representation of restrictions, as conditions fulfilled with respect to another node have been considered in related concepts (Fegeras, 1999; Russell et al., 2008).

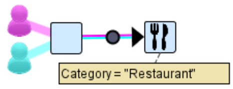
Based upon the aforementioned visual query elements, various compositions are possible. In the following, we will present some advanced compositions and restrictions by example.

Person nodes are always indicated next to the first event node in the sequence of the respective person and connected to it with a thick, semi-transparent line for clarification. There is only one person node for each person in the dataset. We chose this design, as restrictions on persons are globally valid throughout the whole event sequence. A simple example of a VESPa query with one person is shown in Figure 3(a). It shows a person who moves from an arbitrary place to a place categorized as a *restaurant*.

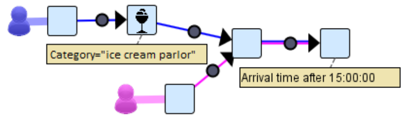
Transitions are shown as colored lines. The color(s) of each transition match(es) the color(s)



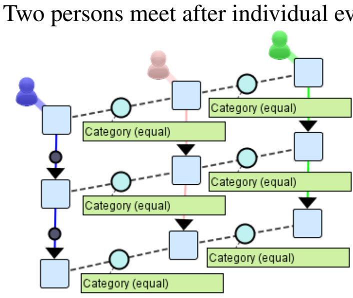
(a) A single person travels to a restaurant.



(b) Two persons travel to a restaurant together.



(c) Two persons meet after individual events.



(d) Three persons visit three places of the same category.

Figure 3: Various exemplary event sequence patterns.

of the involved persons[1]. For instance, Figure 3(b) shows how two persons meet at an arbitrary place and move on together to a restaurant from there—the example from the very beginning of this section.

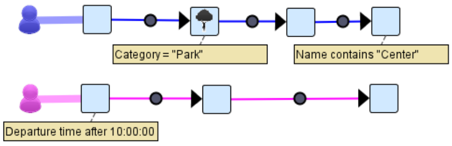
The arrowhead indicates the direction of each transition. As event nodes represent locations *at a specific time* in spatio-temporal contexts, cycles of transitions and undirected transitions are disallowed. A transition may include several legs of a journey, as a specified number of events to skip can be supplied for each transition in the pattern.

Based upon these definitions, we can also represent the complicated pattern described textually in the beginning of this section, as shown in Figure 3(c). As an example of how comparison edges can be used, Figure 3(d) shows a query that finds three persons that subsequently visit three similar (with respect to their category attribute) places in the same order.

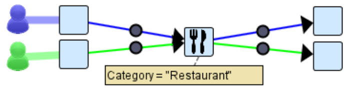
3.2 Interpretation of Event Nodes

As pointed out above, we chose to define an *event node* in a query as a place at a particular time. This allows us to focus on encounters between several persons, which thus need to coincide both in time and place. Furthermore, we define that no two event nodes can refer to stays of persons at the same place and at overlapping timespans.

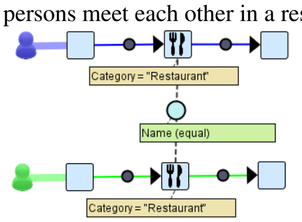
[1]For this paper, colors were chosen so as to be distinguishable both on colored and greyscale printouts. Conceptually, alternative color schemes can be chosen that are specific to user requirements, for instance, color blindness.



(a) Event sequences of two persons who do not meet each other.



(b) Two persons meet each other in a restaurant.



(c) Two persons visit the same restaurant, but at different times (and hence do not meet).

Figure 4: Contrast between persons who visit the same place at the same time and persons who visit the same place at different times.

Accordingly, Figure 4(a) shows an event sequence pattern that represents independent event sequences of two persons who do not meet. Expressed as shown in Figure 4(b), the persons meet at a restaurant (the central node marked with a fork-and-knife-symbol). In Figure 4(c), both persons visit a particular restaurant (in a dataset where each restaurant has a unique name), but they do *not* meet. As both depicted restaurant event nodes refer to the same restaurant, and no two event nodes can be mapped to the same location at the same time, it is implied that the two persons stayed there at different times.

3.3 Query Semantics

In the following, we define the exact semantics of VESPa queries. First, we describe the structure of the queried data, followed by a description of how VESPa query elements are mapped to result elements.

We assume the small database schema depicted in Figure 5, essentially a set of persons, each of which can have linear sequences of events. We have chosen this schema as its simple structure illustrates that VESPa does not require a specifically optimized dataset. Also, we expect the conversion of data from other schemas to ours to be sufficiently easy.

Formally, a dataset matching the schema from Figure 5 can be seen as a tuple $d \subseteq \text{Persons}_d \times \text{Sequences}_d \times \text{Events}_d$ of three disjoint sets. We are interested in meetings between persons, but in the above schema, each event belongs to exactly one person. Therefore, a pre-processing step is necessary to replace Events_d with Events'_d—a set of events linked to one or more persons and part of one or more event sequences. Each element $e_d \in \text{Events}'_d$ contains a set $e_d.\text{persons} \subseteq \text{Persons}_d$ of participants of the event.

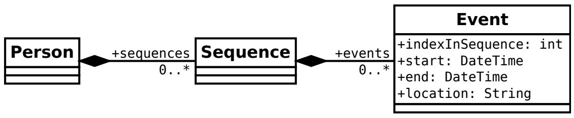


Figure 5: UML diagram of the database schema assumed for the data that can be filtered with VESPa. Persons and events may have additional columns for domain-specific attributes that can be used in restrictions.

Events'_d contains copies of all events from Events_d, as well as any events describing possible meetings. Formally, any element of Events'_d is based upon a set $\tilde{E}_d \subseteq \text{Events}_d$ of events from the original dataset. Then, $\tilde{E}_d$ must satisfy the condition

$$\begin{aligned} \forall \tilde{e}_{d,\alpha}, \tilde{e}_{d,\beta} \in \tilde{E}_d, \tilde{e}_{d,\alpha} \neq \tilde{e}_{d,\beta} : \\ (\tilde{e}_{d,\alpha}.\text{location} = \tilde{e}_{d,\beta}.\text{location}) \\ \wedge (\tilde{e}_{d,\alpha}.\text{start} < \tilde{e}_{d,\beta}.\text{end}) \\ \wedge (\tilde{e}_{d,\beta}.\text{start} < \tilde{e}_{d,\alpha}.\text{end}) \end{aligned}$$

A VESPa query $q \subseteq \text{Persons}_q \times \text{Events}_q$ can then be used to filter the dataset d. For each result of the query, the function

$$\text{valueOf} : \text{Persons}_q \cup \text{Events}_q \rightarrow \text{Persons}_d \cup \text{Events}'_d$$

maps abstract person nodes and event nodes from the query to concrete persons and events from the dataset, respectively, and no two nodes from the query can be mapped to the same person or event from the dataset. Furthermore, the conditions imposed by restrictions and comparison edges are adhered to.

The mapping of event nodes from Events_q to elements of Events'_d is then defined as follows: Let $\tilde{P}_q \subseteq \text{Persons}_q$ be a set of persons from the query that is bijectively mapped to the elements of a set $\tilde{P}_d \subseteq \text{Persons}_d$. Let $e_{q,1}, e_{q,2} \in \text{Events}_q$ be two events from the query that are connected by a transition t_q from $e_{q,1}$ to $e_{q,2}$, where t_q applies to the elements of $\tilde{P}_q$. The mapping can then determine $\text{valueOf}(e_{q,1}) = e_{d,\alpha}$ and $\text{valueOf}(e_{q,2}) = e_{d,\beta}$ such that:

- $e_{d,\alpha}$ and $e_{d,\beta}$ equal the first and last, respectively, in a sequence of events $\dot{e}_{d,1}, \ldots, \dot{e}_{d,\dot{n}} \in \text{Events}'_d$ connected by transitions.
- $\bigwedge_{i=1}^{\dot{n}} \dot{e}_{d,i}.\text{persons} \supseteq \tilde{P}_d$.
- $\dot{n} - 2$ equals the number of intermediate events that may be skipped by t_q.

4 IMPLEMENTATION

We have implemented VESPa in a system based on the Java Swing toolkit. It supports all of the features described in Section 3, except for comparison edges between a person and an event node. Also, there is a hard-coded limitation to six persons.

All VESPa queries depicted in this work are based on screenshots of the prototype. Therefore, circular user interface elements for accessing configuration pop-ups are visible in the center of edges.

In the prototype, we use SQLite for executing the queries, though other SQL databases could also be used. For that reason, each event sequence pattern is translated by the prototype to an equivalent SQL query. We use the aforementioned schema depicted in Figure 5, with an additional *Person* column to directly retrieve an event owner. The extraction of Events'_d from Events_d as described in Section 3.3 happens on the fly while executing the query, as precomputing all mutual events would multiply the data volume. As we rely on the event index in a sequence, our implementation cannot recognize consecutive meetings of overlapping sets of persons at the same location.

Based on the schema, Listing 1 shows SQL code equivalent to the small event sequence from Figure 3(b). Even with such a simple schema, the SQL statement is inconveniently long and not directly comprehensible. The SQL statement might be slightly shortened by avoiding composite keys in the schema, but the basic complexity of matching events to find possible meetings between persons would remain. We consider this a sign that our visual event sequence pattern constitutes a simplification over the textual specification of the same query with a formal language.

Listing 1: SQL statement equivalent to the event sequence pattern from Figure 3(b), based on the schema from Figure 5: The graphical representation in Figure 3(b) is more concise.

```sql
SELECT P1.id, Ea_1.sequence AS "Sequence1",
  Ea_1.indexInSequence, Eb_1.indexInSequence,
  P2.id, Ea_2.sequence AS "Sequence2",
  Ea_2.indexInSequence, Eb_2.indexInSequence
FROM Person P1, Person P2,
  Event Ea_1, Event Ea_2, Event Eb_1, Event Eb_2
WHERE (P1.id <> P2.id)
  AND (Ea_1.sequence = Eb_1.sequence)
  AND (Ea_1.indexInSequence + 1
    = Eb_1.indexInSequence)
  AND (Ea_2.sequence = Eb_2.sequence)
  AND (Ea_2.indexInSequence + 1
    = Eb_2.indexInSequence)
  AND (Ea_1.Person = P1.id)
  AND (Ea_2.Person = P2.id)
  AND (Ea_1.location = Ea_2.location)
  AND (Ea_1.start < Ea_2.end
    AND Ea_2.start < Ea_1.end)
  AND (NOT ((Ea_1.location = Eb_1.location)
    AND (Ea_1.start < Eb_1.end)
  AND (Eb_1.start < Ea_1.end)))
  AND (Eb_1.Person = P1.id)
  AND (Eb_2.Person = P2.id)
  AND (Eb_1.location = Eb_2.location)
  AND (Eb_1.start < Eb_2.end
    AND Eb_2.start < Eb_1.end)
  AND (Eb_1.ATTR_STATE_CATEGORY = "Restaurant")
```

5 CASE STUDIES

In the following sections, we describe two scenarios of application for VESPa using two different datasets. Based on these, we want to show how the query notation can be applied to actual datasets to find particular event patterns and retrieve specific information. The case studies do not just present hypothetical possibilities—the described filter operations have actually been executed in the prototype described in Section 4, which then also retrieved and displayed query results mentioned in the case studies.

5.1 Social Event Detection

The first dataset was released as part of the VAST Challenge 2014 (Visual Analytics Community, 2014). Every year, a synthetic dataset is created that covers various patterns that have to be found using Visual Analytics approaches. The 2014 challenge concerns surveillance tasks. In Mini-Challenge 2, the task was to detect frequent, but also suspicious behaviors of people in the two weeks leading up to a kidnapping. In a fictitious city, people work at the company GAStech and rent cars that get tracked with GPS devices. With some additional data provided, such as credit card transactions and points of interest (POIs), we extract event sequences from the movement between logical locations as described in Figure 1.

These sequences serve as input data for VESPa, which we embedded in an analysis system with multiple linked views (see Figure 7). By means our filter approach, we are able to formulate and prove various hypotheses, including meetings of suspicious people, working patterns outside of business hours, or abnormal sequences of movement destinations.

We have adjusted the set of restrictions and attributes supported by the filter prototype to the data available in the datasets used. In this case, the available filters included *name*, *category*, *employment title* and *type*, *time* and *date* of *arrival* and *departure* at an event node, and the amount of *payments*, if any. By exploring the data, we detect fuzzy repetitive rou-

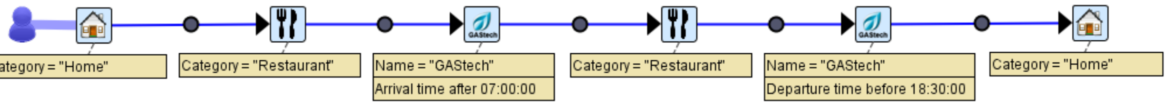


Figure 6: The event sequence pattern finds behavior sequences that fit the expected daily working routine. Most GAStech employees start their day with a coffee, work at GAStech, have a normal lunch break before working again, and leave no later than 6:30 PM to go home.

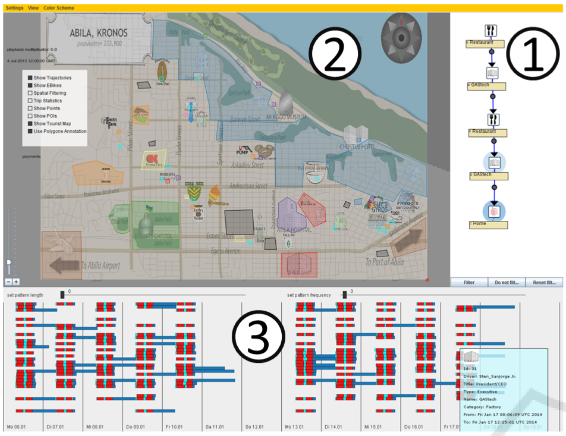
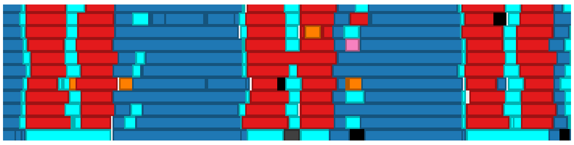
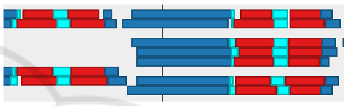


Figure 7: VESPa ① is embedded in an interactive visual analysis system (Krüger et al., 2015) that further consists of a map ② and a temporal sequence view ③ that shows the sequences of all persons (rows) along a horizontal time axis. The coloring of events is based on the annotated map regions. Here, the analyst filtered for a daily working routine similar to Figure 6. The temporal view shows the result set.

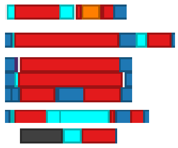
tines using multiple spatial and temporal views. Obviously, these are daily working patterns that are mostly similar, except for some minor temporal differences and some other outliers (see Figure 8(a)). To create a hypothesis based on our suspicion, we express these routines with our filter notation. This visual query is then executed, which means that SQL query code is automatically generated from the filter pattern. It is sent to the database, whereupon any matching event sequences are retrieved. Step by step, we can refine our hypothesis by adding various restrictions. For example, we define the working day to start at 7 AM at the earliest and to end no later than 6:30 PM with a lunch break in between, according to the usual business habits of GAStech. Lastly, after an iterative refinement, our pattern shows that employees start their day with a coffee at one out of various restaurants before they go to work till noon. For lunch, they also mostly go out, and then work till the early evening. Then, some of them go home directly, while others first meet at various bars. The final pattern is depicted in Figure 6. From this query, we get all daily routines that exactly fit this pattern (see Figure 8(b)). Accordingly, when we invert the results, we get all sequences that do *not* match the pattern (see Figure 8(c)). These sequences might be interesting to look into. For example, there are some people who work outside the usual working hours, who do not take a lunch break, or who do not appear at work, but go shopping.



(a) A snippet of unfiltered movement behavior sequences over three days.



(b) A snippet of filtered sequences over two days. Here, the pattern shown in Figure 6 is applied.



(c) Inversion of results reveals some anomalies.

Figure 8: A sequence view shows event sequences over time. The events are colored according to their category: home in blue, restaurant and cafés in cyan, business places in red, and other store types in orange and pink.

In our further analysis, we want to investigate the behavior of one of these persons in depth. He is a GAStech CEO (chief executive officer) who arrives late during the week. Thus, his behavior significantly differs from that of other employees. After we have explored the data with various visual tools, we wonder whether the CEO might be involved in any kinds of suspicious activities outside of work. We therefore construct a filter pattern to verify the hypothesis as follows: First, we create an event node, that we set not to match the GAStech company building. We then set the person for this event to be the CEO. Lastly, we add another person node that is not restricted any further, either. By doing so, we formulate a query that finds events during which the executive meets any other person outside of GAStech (see Figure 9).

As a result we get three main events. Besides official meetings at the company GAStech on Friday, the CEO also meets three other executives at the golf course. Furthermore, he meets other employees for lunch on Saturday and for dinner on Sunday. While this result does not fully confirm our hypothesis, knowledge about the meetings outside of work proved helpful for a complete overview of the events

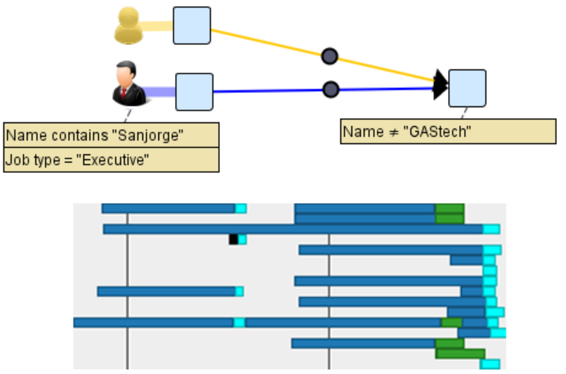


Figure 9: The event sequence pattern (top) finds events where the GAStech CEO named Sanjorge meets somebody else outside the company building of GAStech. The sequence view (bottom) visualizes the results, revealing that the CEO meets people at restaurants (cyan) and at the golf course (green). The person node representing the CEO has been configured to use a custom icon to emphasize the restriction to one particular person.

leading up to the kidnapping. Our findings match the official solution of the challenge.

5.2 Transportation Analysis

The second use case is based on a large real-world dataset (Bracciale et al., 2014) that is retrieved from CRAWDAD, an online platform for open source data. Movements of 320 taxis in Rome were recorded in the course of 30 days at a high resolution. Overall, the data consists of more than 12,000 trips. Again, we mapped trip origins and destinations to areas of interest, such as business-related and historic city districts. As the dataset is very large, we applied a semi-automated enrichment approach (Krüger et al., 2014) that extracts POIs (points of interest) from Foursquare, a social media service. We then manually reviewed and refined the results. Lastly, we enriched the trips of the taxis with this information, which resulted in a temporal sequence of POI visits for each taxi (event sequence) as described in Figure 1. This data serves as input data for our filter approach.

Exemplarily, we showcase a query to find movements starting at the airport, which is far out of town. We suspect that taxis are often used to get to the requested destination. The result of this query reveals that taxis frequently travel to several hotels, located in the city and the upper north of Rome. We then refine our filter query for sequences from an airport to hotels only (see Figure 10).

For this scenario, we use an interactive geographic map that shows the location of the events and the corresponding transitions (routes), as can be seen in Figure 11. This reveals that there are various taxi trips to pick up hotel guests at the International Airport outside of the city to drive them to their hotels (see Figure 11, bottom). Continuing our analysis, we query the dataset for any tourist activities starting from these hotels. As suspected, taxis are used to visit common sights, such as Vatican City and the Colosseum. Further queries could investigate taxi usage with respect to temporal patterns. For example, one might expect more trips to historic places during weekends.

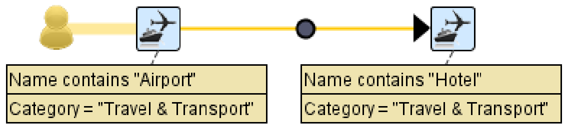


Figure 10: The filter pattern finds sequences starting at any airport and finishing at any hotel.

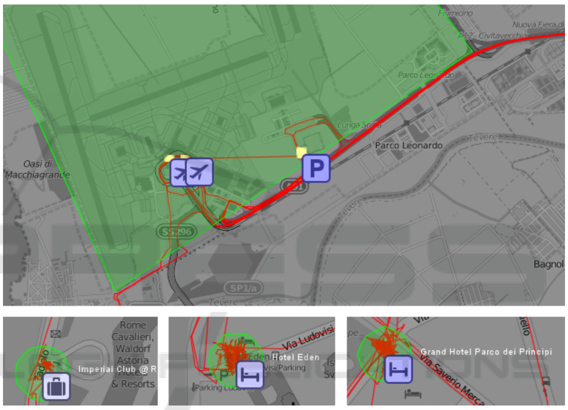


Figure 11: In an interactive geographic map, the filter results can be investigated. Here, icons visualize location types. Movements between locations are drawn in red. The upper image shows the departure of taxis at the airport. The images on the bottom show arrivals of taxis at various hotels.

6 USER EXPERIMENT

To evaluate comprehension and composition understandability of VESPa, we have conducted a small experiment with five users. Our goal was to find out whether users can, after a short introduction, express event sequences with our notation, and whether they are able to recognize what event sequence patterns assembled by someone else mean. Moreover, we wanted to gain some insight on possible issues and improvements for our visual notation, based on user knowledge about visualization and impressions from reading and creating event sequence patterns based on our concept. For that purpose, we prepared six comprehension tasks and six composition tasks that make use of all visual elements.

6.1 Comprehension Tasks

In the six comprehension tasks, participants were provided with completed event sequence patterns. The task was to describe in detail what event sequence patterns are depicted, and what restrictions and other conditions were expressed. Moreover, participants were asked to briefly speculate on the nature of the events that might match each of the event sequence patterns. Task complexity ranged from very simple graphs (like the one depicted in Figure 3(a)) to more complex ones, such as the one shown in Figure 12.

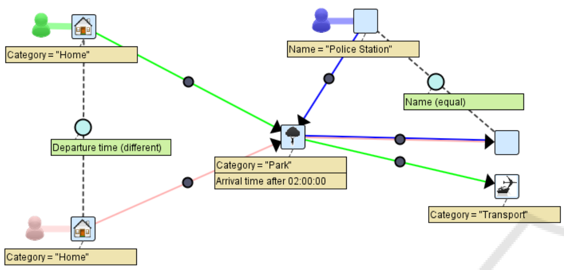


Figure 12: One of the most complex comprehension tasks from the user study: This event sequence pattern matches two individuals convening in a park, possibly at night, while a third person arrives from a police station. One of the two initial persons joins the third individual to return to the police station, while the remaining person heads for a transportation place (for instance, a train station or an airport).

6.2 Composition Tasks

For the composition tasks, textual descriptions of six queries for event sequences were prepared. Participants were asked to use the interactive prototype to assemble an appropriate event sequence patterns that would help to solve each of the textual queries. Again, the first few queries were quite basic (e.g. “Find all users who started at a restaurant, traveled to a factory, and traveled to another restaurant from there.”), while the last ones were more complex (“Are there any users that, after starting at different locations, meet at one location, travel to another location together, and move on to different places?”).

6.3 Materials and Equipment

All materials for the user study were prepared in German, the native language of the participants. We prepared a brief printed description of the general idea and the graphical elements found in the concept, similar to Section 3 in this work. The twelve tasks were printed on paper, as well. As we wanted to emphasize and evaluate the intuitive mental transition between the real-world environment and our event sequence patterns, we used layman’s terms in the descriptions and tasks rather than the scientific terminology found in this paper—for instance, a *spatio-temporal event node* became an *event at a particular time and place*, and a *comparison edge* became a *comparison line*.

The implemented prototype was displayed on a 19” monitor. While there was no written documentation for the user interface, participants had an opportunity to get used to the interaction specifics of the implementation before starting to work on the tasks. In addition, they were allowed to ask questions concerning the interaction throughout the user study, as the evaluation focused on the visual notation, not the context menus or settings dialog boxes of the editor.

6.4 Participants and Procedure

We recruited five participants (four males, one female), all of whom are researchers from the field of visualization. Each participant conducted the study separately, while two of the authors were present to give directions and note down any responses.

After reading the description of the concept and familiarizing themselves with the prototype, participants were first given the sheets with the comprehension tasks and solved them. Subsequently, the sheet with the composition tasks was handed out to the participants, and they used the implemented prototype to create solutions. Eventually, participants were asked for general thoughts and suggestions concerning the visual concept, as well as for ideas of additional use cases for VESPa. During this concluding interview, expected solutions of the completed trials were gradually revealed to the participants, to give them an opportunity to reflect on the discrepancies.

6.5 Results

Overall, the performance of the study participants was very promising—each of them could find a solution for each question, and all answers were at least partially correct. Out of the total of 60 trials, 42 were answered correctly—23 comprehension trials and 19 composition tasks—while answers for the others usually contained only a single mistake related to the interpretation or the identity of event nodes. For the additional question about the possible nature of the events described by the event sequence patterns, the participants’ understanding matched the core of our back-story. They merely interpreted minor parts differently, e.g., the movement of the third person to the police station in Figure 12.

Comments were mostly positive; several of the participants noted that the notation was entirely clear to them once difficulties had been discussed after completing the trials. Two participants found the visualization "intuitive" and easy to learn. Another participant pointed out that achieving an overview with our event sequence pattern notation was inherently easier than with any text-based language.

While we had told participants to interpret event nodes with the same name as referring to the same location, several of the participants struggled slightly with this definition, and one explicitly wished for an *identical place* connector between event nodes. In general, participants were quick to identify each event node with one location, and thus none of the participants correctly recognized that in one of the comprehension tasks, two event nodes actually represented the same location at different (non-overlapping) times. One participant pointed out that the category icons displayed in some event nodes to express categories such as *Home* or *Factory* visually conveyed to her that each event node was equivalent to a place. In turn, some of the participants neglected the necessity to enforce that two event nodes be mapped to two different locations, when the locations were explicitly required to be different.

Minor remarks about the graphical representation, which were partially related to our particular implementation rather than the concept in general, asked for an increased thickness of transitions especially when several persons were involved and a smart positioning of restriction boxes. Moreover, a clearer distinction between transitions and the (equally colored, but thicker and partially transparent) connector lines between person nodes and event nodes (cf. for example Figure 12) was suggested.

6.6 Discussion of Results

The most frequent issue encountered by participants was the interpretation of an event node as a place *at a specific time*. None of them had a definitive idea how to improve the notation, however, and all agreed that the definition was logical, just not intuitive at first.

A possible way to mitigate this difficulty is a time-related symbol on an event node, such as a little clock, as a visual clue that the event node also has a temporal aspect. That clock symbol might even be modified in a domain-specific way to emphasize temporal restrictions, if any, such as an absolute start or end time or a minimum or maximum duration. Another suggestion was related to placing event nodes on some kind of a timeline, which may work well when the temporal relationship between two event nodes (before, after, same time) is known. On the other hand, this can create additional problems for event nodes whose relative temporal relationship is unknown (or irrelevant), or between event nodes with absolute and relative temporal restrictions.

Two participants were unsure about the notion of persons *meeting* one another at the beginning of an event sequence. They ended up adding one additional "entry event node" for each person, rather than starting with a mutual event node (cf. Figure 3(b)) right away. Based on the discussion with the respective participants, it seems likely that a stronger connection of person nodes with the event nodes, rather than just with connected transitions like in our prototype, might emphasize the idea that the persons "meet" when they start out at an initial mutual event node.

The answers of one participant in the user study also revealed the importance of a careful selection of category names and icons. That particular user generally interpreted event nodes restricted to locations of the category *Transport* as the process of transportation itself, rather than a location related to transportation (a train station, an airport, etc.), as the icon in question showed the vehicles rather than the building.

7 CONCLUSION AND FUTURE WORK

We have presented VESPa, a visual notation for specifying patterns of event sequences. With this notation, ideas about possible event sequences can be visualized in an abstract and concise way. The visualization can then be used to filter databases of actual event sequences, to check hypotheses about the events in such a database, and also to express common patterns found therein. As opposed to existing approaches, overlapping event sequences of several persons can be explicitly expressed in VESPa. Rather than keeping space and time separate like related work, VESPa uses spatio-temporal event nodes that make queries for meetings of persons very straightforward.

While our event sequence pattern visualization seems adequate for expressing certain complex patterns, and was accordingly positively commented on by participants of our qualitative study. The experiment also revealed some issues that need to be addressed. The temporal aspects of event nodes needs to be clarified; future research might also lead to domain-specific solutions. Possible extensions to the visual notation include directedness of comparison edges to support asymmetric relationships between event nodes or person nodes, and restrictions on transitions. More importantly, concepts that are currently not supported, such as repeated or alternative subsequences of events, need to be considered.

We have shown the applicability of the concept to certain spatio-temporal movement information in our case studies, but we see the potential to use the event sequence pattern notation in more generic contexts, some of which were suggested by study participants. Beside the use case of surveillance and digital forensics, event sequence patterns might also help recognize movement patterns for other purposes such as marketing analysis, website visitor navigation (where event nodes could map to single pages), wild animals in their habitat, or of fictional characters to describe or find specific scenes in novels or movies. Furthermore, the event sequence patterns might be useful for different kinds of analysis tasks, for instance gaze patterns found in eye-tracking data, data packets in network traffic, or cross-thread resource access. Lastly, our visual event sequence pattern notation can be used in a prescriptive way to specify abstract itineraries for time management assistance systems, which could then be automatically completed to a concrete itinerary based on transportation schedules, business location information, and calendars of co-workers or friends.

ACKNOWLEDGEMENTS

This work was supported by the graduate program Digital Media of the University of Stuttgart and the Stuttgart Media University (HdM) as well as the Horizon 2020 project *CIMPLEX*, grant no. 641191.

REFERENCES

Abela, J., Debeaupuis, T., and Herve Schauer Consultants (1999). Universal format for logger messages. http://tools.ietf.org/html/draft-abela-ulm-05.

Andrienko, N., Andrienko, G., and Fuchs, G. (2013). Towards privacy-preserving semantic mobility analysis. In *EuroVis Workshop on Visual Analytics*, pages 19–23. Eurographics Association.

Atrey, P., Maddage, M., and Kankanhalli, M. (2006). Audio based event detection for multimedia surveillance. In *ICASSP '06 Proc.*, volume 5, pages 813–816. IEEE.

Atrey, P. K., Kankanhalli, M. S., and Jain, R. (2005). Timeline-based information assimilation in multimedia surveillance and monitoring systems. In *Proc. VSSN '05*, pages 103–112. ACM.

Bonhomme, C., Trépied, C., Aufaure, M.-A., and Laurini, R. (1999). A visual language for querying spatio-temporal databases. In *Proc. GIS '99*, pages 34–39. ACM.

Boyandin, I., Bertini, E., Bak, P., and Lalanne, D. (2011). Flowstrates: An approach for visual exploration of temporal origin-destination data. *Comput. Graphics Forum*, 30(3):971–980.

Bracciale, L., Bonola, M., Loreti, P., Bianchi, G., Amici, R., and Rabuffi, A. (2014). CRAWDAD data set roma/-taxi (v. 2014-07-17). http://crawdad.org/roma/taxi/.

Certo, L., Galvão, T., and Borges, J. (2013). Time Automaton: A visual mechanism for temporal querying. *J. Visual Lang. Comput.*, 24(1):24–36.

Dionisio, J. D. and Cárdenas, A. F. (1996). MQuery: A visual query language for multimedia, timeline and simulation data. *J. Visual Lang. Comput.*, 7(4):377 – 401.

Do, Q. X., Lu, W., and Roth, D. (2012). Joint inference for event timeline construction. In *Proc. EMNLP-CoNLL '12*, pages 677–687. ACL.

D'Ulizia, A., Ferri, F., and Grifoni, P. (2012). Moving GeoPQL: A pictorial language towards spatio-temporal queries. *GeoInformatica*, 16(2):357–389.

Fails, J., Karlson, A., Shahamat, L., and Shneiderman, B. (2006). A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *VAST '06*, pages 167–174.

Fegeras, L. (1999). VOODOO: A visual object-oriented database language for odmg oql. In *W13. The First ECOOP Workshop on Object-Oriented Databases*.

Fischer, F., Mansmann, F., and Keim, D. A. (2012). Real-time visual analytics for event data streams. In *Proc. SAC '12*, pages 801–806. ACM.

Gaaloul, W., Bhiri, S., and Godart, C. (2004). Discovering workflow transactional behavior from event-based log. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, volume 3290 of *LNCS*, pages 3–18. Springer.

Gotz, D. and Stavropoulos, H. (2014). DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE TVCG*, 20(12):1783–1792.

Guo, H., Wang, Z., Yu, B., Zhao, H., and Yuan, X. (2011). TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *Proc. PacificVis '11*, pages 163–170. IEEE.

Guo, X., Li, J., Yang, R., and Ma, X. (2014). NEI: A framework for dynamic news event exploration and visualization. In *Proc. VINCI '14*, pages 121–128. ACM.

Havre, S., Hetzler, B., and Nowell, L. (2000). ThemeRiver: Visualizing theme changes over time. In *Proc. InfoVis '00*, pages 115–123. IEEE.

Heydekorn, J., Nitsche, M., Dachselt, R., and Nürnberger, A. (2011). On the interactive visualization of a logistics scenario: Requirements and possible solutions. In *Proc. IWDE '11*, Technical report (Internet): Elektronische Zeitschriftenreihe der Fakultät für Informatik der OVGU Magdeburg, pages 1–7.

Huang, Y., Zhang, L., and Zhang, P. (2008). A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Trans. Knowl. Data Eng.*, 20(4):433–448.

Jiang, F., Yuan, J., Tsaftaris, S. A., and Katsaggelos, A. K. (2011). Anomalous video event detection using spatiotemporal context. *Comput. Vision Image Understanding*, 115(3):323–333.

Jin, J. and Szekely, P. (2009). QueryMarvel: A visual query language for temporal patterns using comic strips. In *Proc. VL/HCC '09*, pages 207–214.

Kapler, T. and Wright, W. (2005). GeoTime information visualization. *Information Visualization*, 4(2):136–146.

Kim, P. H. and Giunchiglia, F. (2012). Life logging practice for human behavior modeling. In *Proc. SMC '12*, pages 2873–2878.

Krstajić, M., Bertini, E., and Keim, D. (2011). CloudLines: Compact display of event episodes in multiple time-series. *IEEE TVCG*, 17(12):2432–2439.

Krüger, R., Herr, D., Haag, F., and Ertl, T. (2015). Inspector Gadget: Integrating data preprocessing and orchestration in the visual analysis loop. In *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association.

Krüger, R., Thom, D., and Ertl, T. (2014). Visual analysis of movement behavior using web data for context enrichment. In *Proc. PacificVis '14*, pages 193–200. IEEE.

Krüger, R., Thom, D., Wörner, M., Bosch, H., and Ertl, T. (2013). TrajectoryLenses – A set-based filtering and exploration technique for long-term trajectory data. *Comput. Graphics Forum*, 2013(3):451–460.

Kumar, C., Heuten, W., and Boll, S. (2013). Geographical queries beyond conventional boundaries: Regional search and exploration. In *Proc. GIR '13*, pages 84–85. ACM.

Kumar, V., Furuta, R., and Allen, R. B. (1998). Metadata visualization for digital libraries: Interactive timeline editing and review. In *Proc. DL '98*, pages 126–133. ACM.

Makanju, A., Zincir-Heywood, A. N., and Milios, E. E. (2011). Storage and retrieval of system log events using a structured schema based on message type transformation. In *Proc. SAC '11*, pages 528–533. ACM.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. (2011). Twitinfo: Aggregating and visualizing microblogs for event exploration. In *Proc. CHI '11*, pages 227–236. ACM.

Monroe, M., Lan, R., Morales del Olmo, J., Shneiderman, B., Plaisant, C., and Millstein, J. (2013). The challenges of specifying intervals and absences in temporal queries: A graphical language approach. In *Proc. CHI '13*, pages 2349–2358. ACM.

Morris, A., Abdelmoty, A., El-Geresy, B., and Jones, C. (2004). A filter flow visual querying language and interface for spatial databases. *GeoInformatica*, 8(2):107–141.

Nguyen, T., Loke, S., and Torabi, T. (2007). The Community Stack: Concept and prototype. In *Proc. AINAW '07*, volume 2, pages 52.–58.

Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., and Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4):42:1–42:32.

Peuquet, D. J. and Duan, N. (1995). An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *Int. J. Geogr. Inf. Syst.*, 9(1):7–24.

Pirolli, P. and Card, S. (2005). The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. Int'l Conf. on Intelligence Analysis*, pages 2–4. MITRE.

Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. (1996). LifeLines: Visualizing personal histories. In *Proc. CHI '96*, pages 221–227. ACM.

Russell, A., Smart, P., Braines, D., and Shadbolt, N. (2008). NITELIGHT: A graphical tool for semantic query construction. In *Proc. SWUI '08*, volume 543 of *CEUR-WS*.

Seifert, I. (2011). A pool of queries: Interactive multidimensional query visualization for information seeking in digital libraries. *Information Visualization*, 10(2):97–106.

Shneiderman, B. (1994). Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77.

Soylu, A., Giese, M., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., and Horrocks, I. (2013). OptiqueVQS: Towards an ontology-based visual query system for big data. In *Proc. MEDES '13*, pages 119–126. ACM.

Sun, G., Liu, Y., Wu, W., Liang, R., and Qu, H. (2014). Embedding temporal display into maps for occlusion-free visualization of spatio-temporal data. In *Proc. PacificVis '14*, pages 185–192. IEEE.

Tao, C., Wongsuphasawat, K., Clark, K., Plaisant, C., Shneiderman, B., and Chute, C. G. (2012). Towards event sequence representation, reasoning and visualization for EHR data. In *Proc. IHI '12*, pages 801–806. ACM.

Tominski, C., Schumann, H., Andrienko, G., and Andrienko, N. (2012). Stacking-based visualization of trajectory attribute data. *IEEE TVCG*, 18(12):2565–2574.

Visual Analytics Community (2014). VAST 2014 Challenge – the Kronos incident. http://va community.org/VAST+Challenge+2014.

Westermann, U. and Jain, R. (2007). Toward a common event model for multimedia applications. *IEEE MultiMedia*, 14(1):19–29.

Wongsuphasawat, K., Plaisant, C., Taieb-Maimon, M., and Shneiderman, B. (2012). Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interact. Comput.*, 24(2):55–68.

Wu, S., Otmane, S., Moreau, G., and Servières, M. (2013). Design of a visual query language for geographic information system on a touch screen. In *Human-Computer Interaction. Interaction Modalities and Techniques*, volume 8007 of *LNCS*, pages 530–539. Springer.

Zgraggen, E., Drucker, S. M., Fisher, D., and DeLine, R. (2015). (s—qu)eries: Visual regular expressions for querying and exploring event sequences. In *Proc. CHI '15*, pages 2683–2692. ACM.

Zhu, X. Y., Guo, W., Huang, L., Hu, T., and Gao, W. X. (2013). Pan-information location map. *ISPRS Archives*, XL-4(4):57–62.