

Visual Querying of Semantically Enriched Movement Data

Florian Haag^(✉), Robert Krüger, and Thomas Ertl

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

{florian.haag, robert.krueger, thomas.ertl}@visus.uni-stuttgart.de

Abstract. Visual data exploration is used to reveal unknown patterns that, however, need to be validated, refined, and extracted for a final presentation and reporting. We contribute VESPa, a pattern-based visual query language for event sequences. With VESPa, analysts can formulate hypotheses gained and query the data for matches. In an interactive analysis loop the pattern can be altered with further restrictions to narrow down the result set. Our language allows for (1) hypothesis expression and refinement, (2) visual querying, and (3) knowledge externalization. We focus on semantically enriching movement data, used in law enforcement, consumer, and traffic analysis. To evaluate the applicability we present two case studies as well as a user study consisting of comprehensive and composition tasks.

Keywords: Visual query language · Semantic movement analysis

1 Introduction

Rising amounts of movement data and an increasing recording precision offer new possibilities for urban and transport planning, customer analysis, and law enforcement. While visual 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.

This publication is an extended version of our IVAPP 2016 paper [1]

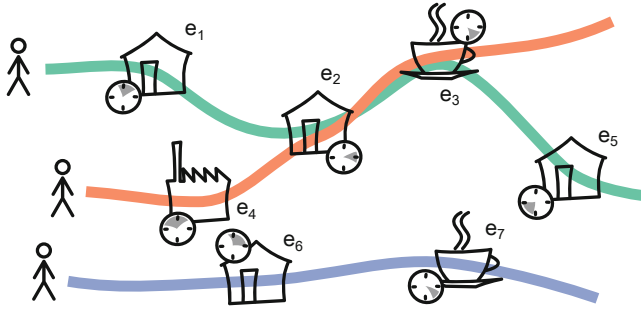


Fig. 1. A sketch of a movement dataset that stores consecutive stays and meetings of people at specific locations and time as sequences of events. For instance, the person in the middle has the event sequence $e_4 \rightarrow e_2 \rightarrow e_3$ and meets the person on top in e_2 and e_3 .

Detecting such situations can be complex, as multiple conditions like varying timespans, different places, persons, and orderings of events need to be 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. [2,3]) visualizes 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 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 [4]. 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 Sect. 2, we present VESPa, our Visual Event Sequences Pattern notation, in Sect. 3 and justify our design choices. The general idea of using the technique is outlined in Sect. 4. We demonstrate the usefulness of VESPa with several pattern examples, two case studies (Sect. 6), and a small user study (Sect. 7), both using a prototypical implementation (Sect. 5).

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 [5, 6]. In the case of multiple logging sources, that information can also include a hint about the origin of the event message [7].

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 [8]. Similarly, events are defined with respect to sensor data [9–11], where a timespan can be supplied as an additional attribute [12].

Entries in spatio-temporal event logs can be associated with timespans and locations [13]. Such entries may include data that implicitly or explicitly points out a connection between related events [14, 15].

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 [16]. These data are commonly split up to distinguish different persons [17] or event types [10, 18]. Additional information on single events or on sets of events can be encoded into the timeline [19, 20]. In some cases, the lines still conform to a time axis, but are reshaped to express some information themselves [21, 22].

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 [20, 23]. Alternative approaches use a space-time-cube [24, 25], or they integrate aggregated spatial data into the temporal visualization [26]. Moreover, additional visual elements can indicate the temporal properties of data shown in a spatial view [27].

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 [28, 29]. The link between the geographical locations and the logical locations can be displayed [30], but in some cases, the primary interest lies in the semantic locations [31, 32], or they are even the only data available or cleared for use [33].

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 [34–37]. There are specialized query visualizations for spatial data, some using symbolic representations of geographical relationships [38,39] and others working directly on concrete maps [40]. 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 [41–43]. Also, a graphical notation that expresses exact relative relationships between locations or areas and timespans has been proposed [44]. The only other visual approach that we are aware of displays the query restrictions in a space-time-cube [45]. Some of these concepts can be further abstracted to replace the geographical association with restrictions based on logical locations [42], and time-related works provide some further ideas on how to visually represent certain features required for temporal queries [46].

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 [2,16,47]. Only few concepts represent series of events to find on a timeline [48], or as nodes with temporal constraints [49], rather than just displaying results in a visual way [16]. Still, relations between distinct objects are usually not supported [3]. When they are, these are expressed by matching colors in otherwise visually disjoint elements [50].

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 [16], 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 [50] 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 [5,6], 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 Sequences 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, for each result.

3.1 Query Elements

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

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

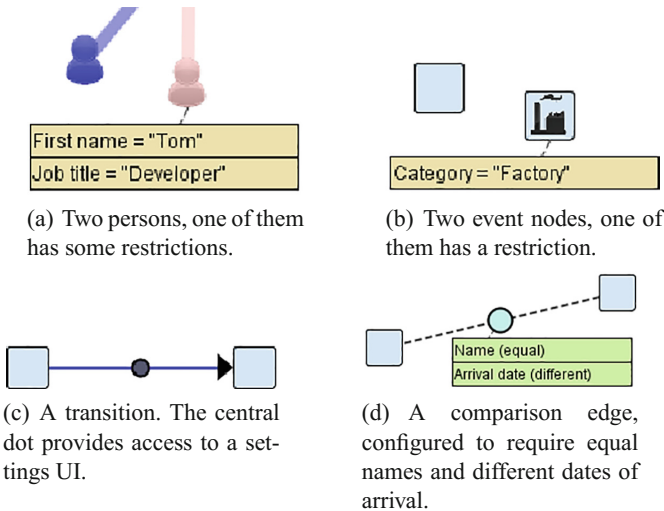


Fig. 2. The basic visual elements used in VESPa.

Event Node: An event node, shown in Fig. 2(b), is a single node in an event sequence pattern, like events found in related work [16,48,49]. In spatio-temporal contexts, it denotes a stay of one or more persons at a particular location and timespan.

Transition: A transition unidirectionally connects two event nodes (Fig. 2(c)), comparable to the TimeSpan bars from PatternFinder [48]. 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 Figs. 2(a) and (b).

Comparison Edge: A comparison edge connects two person nodes, two event nodes (as shown in Fig. 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 [37,51].

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 Fig. 3(a). It shows a person who moves from an arbitrary place to a place categorized as a *restaurant*.

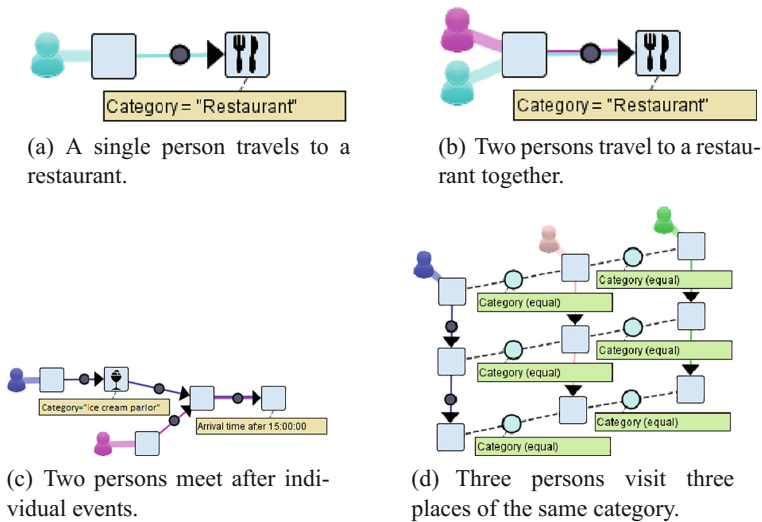


Fig. 3. Various exemplary event sequence patterns.

Transitions are shown as colored lines. The color(s) of each transition match(es) the color(s) of the involved persons¹. For instance, Fig. 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 Fig. 3(c). As an example of how comparison edges can be used, Fig. 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.

Accordingly, Fig. 4(a) shows an event sequence pattern that represents independent event sequences of two persons who do not meet. Expressed as shown in Fig. 4(b), the persons meet at a restaurant (the central node marked with a fork-and-knife-symbol). In Fig. 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 Fig. 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.

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.

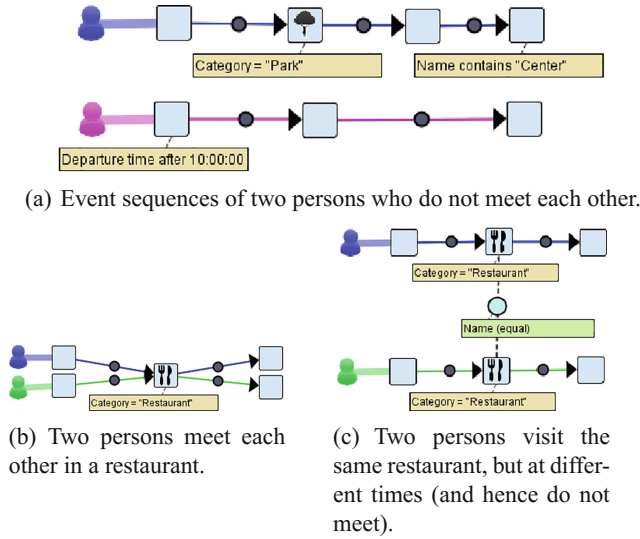


Fig. 4. Contrast between persons who visit the same place at the same time and persons who visit the same place at different times.

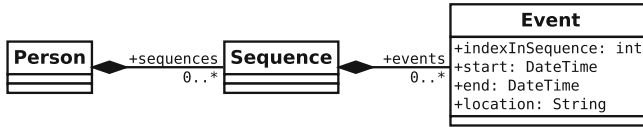


Fig. 5. UML diagram of the database schema assumed for the data that can be filtered with VESPa. Persons and events may have additional columns for domain-specific attributes that can be used in restrictions.

Formally, a dataset matching the schema from Fig. 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.

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

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

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 $\dot{e}_{d,1}, \dots, \dot{e}_{d,\dot{n}} \in \text{Events}'_d$ 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 Usage Schema

VESPa is most powerful when it is coupled with a visualization to explore the query results. Figure 6 shows a typical analysis process and explains how VESPa can be integrated.

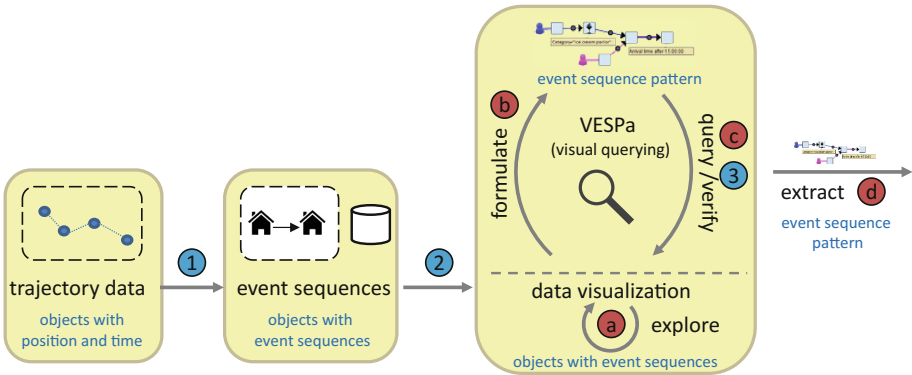


Fig. 6. Usage schema of VESPa in a visual analysis process. The blue numbers represent computational steps. The red labels highlight analysis steps of the user. (Color figure online)

The blue numbers in the schema figure indicate computational steps, while the red labels highlight analysis steps conducted by the user. As a computational

preprocessing step, trajectory data (e.g., retrieved via GPS), is transformed into event sequences (① in Fig. 6). This can be done with any semantic enrichment process, e.g., mapping the destinations of trips to points or areas of interest, as proposed by Parent et al. [28]. This is, however, beyond the scope of this paper.

Subsequently, the sequence data is stored in a database with a schema like the one presented in Fig. 5. As a next step, a data visualization can be computed (②). This can include maps and timelines that allow the analyst to explore the data in time and space (a). Gained hypotheses can then be formulated as a graphical pattern by using the pattern-based visual query language VESPa (b). When the analyst executes the pattern (c), the database is queried and returns results that satisfy the query (③).

More precisely, the dataset is scanned based upon the VESPa queries—suitable instructions or textual queries to the underlying database are automatically generated. The result of this scan is a set of mappings of the elements from the VESPa query to elements from the actual dataset. The analyst can explore them in the result visualization and use the information to iteratively refine the query with VESPa or create new queries. Finally, a verified pattern serves as a compact representation of the finding and may be included in a report document.

5 Implementation

We have implemented VESPa in a system based on the Java Swing toolkit. It supports all of the features described in Sect. 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 Fig. 5, with an additional *Person* column to directly retrieve an event owner. The extraction of $Events'_d$ from $Events_d$ as described in Sect. 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 1.1 shows SQL code equivalent to the small event sequence from Fig. 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 1.1. SQL statement equivalent to the event sequence pattern from Fig. 3(b), based on the schema from Fig. 5: The graphical representation in Fig. 3(b) is more concise.

```

1 SELECT P1.id, Ea_1.sequence AS "Sequence1",
2   Ea_1.indexInSequence, Eb_1.indexInSequence,
3   P2.id, Ea_2.sequence AS "Sequence2",
4   Ea_2.indexInSequence, Eb_2.indexInSequence
5 FROM Person P1, Person P2, Event Ea_1, Event Ea_2, Event Eb_1, Event
6   Eb_2
7 WHERE (P1.id <> P2.id)
8   AND (Ea_1.sequence = Eb_1.sequence) AND (Ea_1.indexInSequence + 1
9     = Eb_1.indexInSequence)
10  AND (Ea_2.sequence = Eb_2.sequence) AND (Ea_2.indexInSequence + 1
11    = Eb_2.indexInSequence)
12  AND (Ea_1.Person = P1.id) AND (Ea_2.Person = P2.id)
13  AND (Ea_1.location = Ea_2.location) AND (Ea_1.start < Ea_2.end
14    AND Ea_2.start < Ea_1.end) AND (NOT ((Ea_1.location = Eb_1.location)
15    AND (Ea_1.start < Eb_1.end) AND (Eb_1.start < Ea_1.end)))
16  AND (Eb_1.Person = P1.id) AND (Eb_2.Person = P2.id)
17  AND (Eb_1.location = Eb_2.location) AND (Eb_1.start < Eb_2.end
18    AND Eb_2.start < Eb_1.end) AND (Eb_1.ATTR_STATE_CATEGORY = "
19    Restaurant")

```

6 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 presented filter operations have actually been executed in the prototype described in Sect. 5, which then also retrieved and displayed query results mentioned in the case studies.

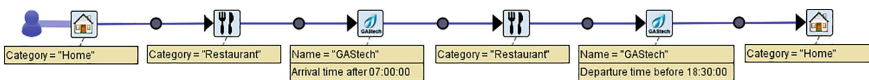


Fig. 7. The event sequence pattern finds behavior sequences that fit the expected daily working routine. Most GAStech employees start their day with a coffee, work at GAStech, have a normal lunch break before working again, and leave no later than 6:30 PM to go home.

6.1 Social Event Detection

The first dataset was released as part of the VAST Challenge 2014 [52]. 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 Fig. 1.

These sequences serve as input data for VESPa (see Sect. 4), which we embedded in an analysis system with multiple linked views (see Fig. 8). 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.

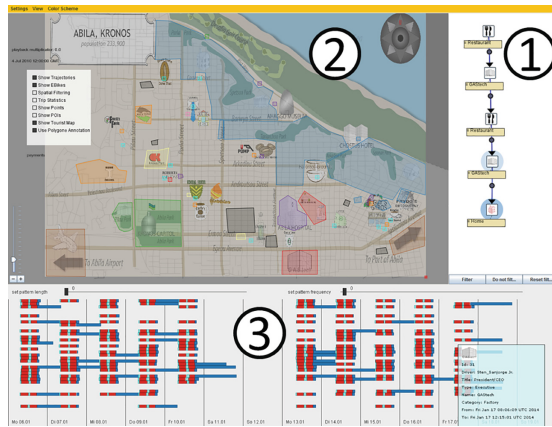


Fig. 8. VESPa ① is embedded in an interactive visual analysis system [53] that further consists of a map ② and a temporal sequence view ③ that shows the sequences of all persons (rows) along a horizontal time axis. The coloring of events is based on the annotated map regions. Here, the analyst filtered for a daily working routine similar to Fig. 7. The temporal view shows the result set.

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 routines 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 Fig. 9(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 Fig. 7. From this query, we get all daily routines that exactly fit this pattern (see Fig. 9(b)). Accordingly, when we invert the results, we get all sequences that do *not* match the pattern (see Fig. 9(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.

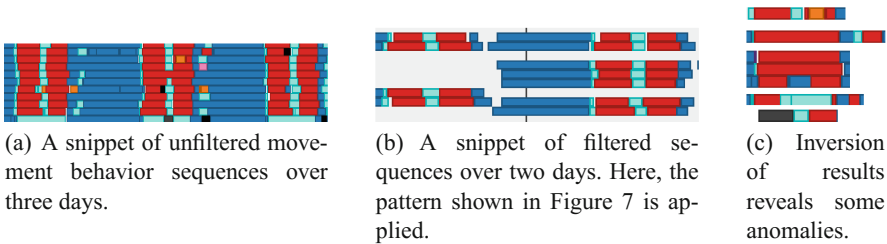


Fig. 9. A sequence view shows event sequences over time. The events are colored according to their category: home in blue, restaurant and cafés in cyan, business places in red, and other store types in orange and pink. (Color figure online)

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 Fig. 10).

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

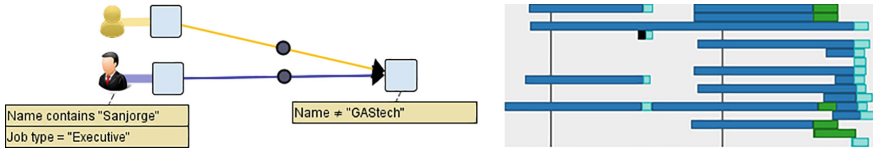


Fig. 10. The sequence pattern (top) finds events where the GAStech CEO named Sanjorge meets somebody else outside the company building of GAStech. The sequence view (bottom) visualizes results, revealing that the CEO meets people at restaurants (cyan) and at the golf course (green). The person node representing the CEO has been configured to use a custom icon to emphasize the restriction to one particular person. (Color figure online)

6.2 Transportation Analysis

The second use case is based on a large real-world dataset [54] 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 [29] 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 Fig. 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 Fig. 11).

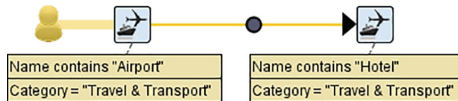


Fig. 11. The pattern finds sequences starting at any airport and finishing at any hotel.

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 Fig. 12. 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 Fig. 12, 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.



Fig. 12. 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. (Color figure online)

7 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.

7.1 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 Fig. 3(a)) to more complex ones, such as the one shown in Fig. 13.

7.2 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?”).

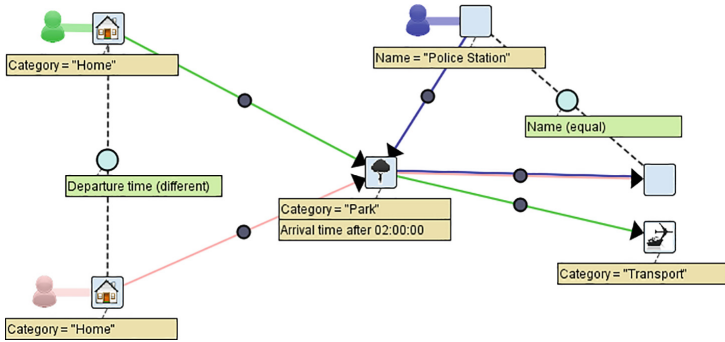


Fig. 13. One of the most complex comprehension tasks from the user study: this event sequence pattern matches two individuals convening in a park, possibly at night, while a third person arrives from a police station. One of the two initial persons joins the third individual to return to the police station, while the remaining person heads for a transportation place (for instance, a train station or an airport).

7.3 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 Sect. 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.

7.4 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.

7.5 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 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 Fig. 13.

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 Fig. 13) was suggested.

7.6 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. Fig. 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.

8 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 Horizon 2020 project *CIMPLEX*, grant no. 641191.

References

1. Haag, F., Krüger, R., Ertl, T.: VESPa: a pattern-based visual query language for event sequences. In: Proceedings of the 7th International Conference on Information Visualization Theory and Applications (IVAp. 2016), vol. 7 (2016)
2. Wongsuphasawat, K., Plaisant, C., Taieb-Maimon, M., Shneiderman, B.: Querying event sequences by exact match or similarity search: design and empirical evaluation. *Interact. Comput.* **24**, 55–68 (2012)
3. Zraggen, E., Drucker, S.M., Fisher, D., DeLine, R.: (s—qu)eries: Visual regular expressions for querying and exploring event sequences. In: Proceedings of CHI 2015, pp. 2683–2692. ACM (2015)
4. Pirolli, P., Card, S.: The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In: Proceedings of International Conference on Intelligence Analysis, MITRE, pp. 2–4 (2005)
5. Makanju, A., Zincir-Heywood, A.N., Milios, E.E.: Storage and retrieval of system log events using a structured schema based on message type transformation. In: Proceedings of SAC 2011, pp. 528–533. ACM (2011)
6. Gaaloul, W., Bhiri, S., Godart, C.: Discovering workflow transactional behavior from event-based log. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 3–18. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-30468-5_3](https://doi.org/10.1007/978-3-540-30468-5_3)
7. Abela, J., Debeaupuis, T., Consultants, H.S.: Universal format for logger messages (1999). <http://tools.ietf.org/html/draft-abela-ulum-05>
8. Do, Q.X., Lu, W., Roth, D.: Joint inference for event timeline construction. In: Proceedings of EMNLP-CoNLL 2012, pp. 677–687. ACL (2012)

9. Atrey, P., Maddage, M., Kankanhalli, M.: Audio based event detection for multimedia surveillance. In: *Proceedings of ICASSP 2006*, vol. 5, pp. 813–816. IEEE (2006)
10. Heydekorn, J., Nitsche, M., Dachselt, R., Nürnberger, A.: On the interactive visualization of a logistics scenario: requirements and possible solutions. In: *Proceedings of IWDE 2011*, pp. 1–7. Technical report (Internet): Elektronische Zeitschriftenreihe der Fakultät für Informatik der OVGU Magdeburg (2011)
11. Kim, P.H., Giunchiglia, F.: Life logging practice for human behavior modeling. In: *Proceedings of SMC 2012*, pp. 2873–2878 (2012)
12. Atrey, P.K., Kankanhalli, M.S., Jain, R.: Timeline-based information assimilation in multimedia surveillance and monitoring systems. In: *Proceedings of VSSN 2005*, pp. 103–112. ACM (2005)
13. Peuquet, D.J., Duan, N.: An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *Int. J. Geogr. Inf. Syst.* **9**, 7–24 (1995)
14. Huang, Y., Zhang, L., Zhang, P.: A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Trans. Knowl. Data Eng.* **20**, 433–448 (2008)
15. Jiang, F., Yuan, J., Tsaftaris, S.A., Katsaggelos, A.K.: Anomalous video event detection using spatiotemporal context. *Comput. Vision Image Underst.* **115**, 323–333 (2011)
16. Plaisant, C., Milash, B., Rose, A., Widoff, S., Shneiderman, B.: LifeLines: visualizing personal histories. In: *Proceedings of CHI 1996*, pp. 221–227. ACM (1996)
17. Kumar, V., Furuta, R., Allen, R.B.: Metadata visualization for digital libraries: interactive timeline editing and review. In: *Proceedings of DL 1998*, pp. 126–133. ACM (1998)
18. Tao, C., Wongsuphasawat, K., Clark, K., Plaisant, C., Shneiderman, B., Chute, C.G.: Towards event sequence representation, reasoning and visualization for EHR data. In: *Proceedings of IHI 2012*, pp. 801–806. ACM (2012)
19. Krstajić, M., Bertini, E., Keim, D.: CloudLines: compact display of event episodes in multiple time-series. *IEEE TVCG* **17**, 2432–2439 (2011)
20. Fischer, F., Mansmann, F., Keim, D.A.: Real-time visual analytics for event data streams. In: *Proceedings of SAC 2012*, pp. 801–806. ACM (2012)
21. Havre, S., Hetzler, B., Nowell, L.: ThemeRiver: visualizing theme changes over time. In: *Proceedings of InfoVis 2000*, pp. 115–123. IEEE (2000)
22. Guo, X., Li, J., Yang, R., Ma, X.: NEI: a framework for dynamic news event exploration and visualization. In: *Proceedings of VINCI 2014*, pp. 121–128. ACM (2014)
23. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: *Proceedings of CHI 2011*, pp. 227–236. ACM (2011)
24. Kapler, T., Wright, W.: GeoTime information visualization. *Inf. Vis.* **4**, 136–146 (2005)
25. Tominski, C., Schumann, H., Andrienko, G., Andrienko, N.: Stacking-based visualization of trajectory attribute data. *IEEE TVCG* **18**, 2565–2574 (2012)
26. Guo, H., Wang, Z., Yu, B., Zhao, H., Yuan, X.: TripVista: triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In: *Proceedings of PacificVis 2011*, pp. 163–170. IEEE (2011)
27. Sun, G., Liu, Y., Wu, W., Liang, R., Qu, H.: Embedding temporal display into maps for occlusion-free visualization of spatio-temporal data. In: *Proceedings of PacificVis 2014*, pp. 185–192. IEEE (2014)

28. Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M.L., Gkouloulas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., Yan, Z.: Semantic trajectories modeling and analysis. *ACM Comput. Surv.* **45**, 42:1–42:32 (2013)
29. Krüger, R., Thom, D., Ertl, T.: Visual analysis of movement behavior using web data for context enrichment. In: *Proceedings of PacificVis 2014*, pp. 193–200. IEEE (2014)
30. Zhu, X.Y., Guo, W., Huang, L., Hu, T., Gao, W.X.: Pan-information location map. *ISPRS Archives XL–4*, 57–62 (2013)
31. Nguyen, T., Loke, S., Torabi, T.: The community stack: concept and prototype. In: *Proceedings of AINAW 2007*, vol. 2, pp. 52–58 (2007)
32. Westermann, U., Jain, R.: Toward a common event model for multimedia applications. *IEEE Multimedia* **14**, 19–29 (2007)
33. Andrienko, N., Andrienko, G., Fuchs, G.: Towards privacy-preserving semantic mobility analysis. In: *EuroVis Workshop on Visual Analytics*, pp. 19–23. Eurographics Association (2013)
34. Shneiderman, B.: Dynamic queries for visual information seeking. *IEEE Softw.* **11**, 70–77 (1994)
35. Seifert, I.: A pool of queries: interactive multidimensional query visualization for information seeking in digital libraries. *Inf. Vis.* **10**, 97–106 (2011)
36. Soylyu, A., Giese, M., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I.: OptiqueVQS: towards an ontology-based visual query system for big data. In: *Proceedings of MEDES 2013*, pp. 119–126. ACM (2013)
37. Russell, A., Smart, P., Braines, D., Shadbolt, N.: NITELIGHT: a graphical tool for semantic query construction. In: *Proceedings of SWUI 2008*, vol. 543 of CEUR-WS (2008)
38. Morris, A., Abdelmoty, A., El-Geresy, B., Jones, C.: A filter flow visual querying language and interface for spatial databases. *GeoInformatica* **8**, 107–141 (2004)
39. Wu, S., Otmane, S., Moreau, G., Servières, M.: Design of a visual query language for geographic information system on a touch screen. In: Kurosu, M. (ed.) *HCI 2013*. LNCS, vol. 8007, pp. 530–539. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-39330-3_57](https://doi.org/10.1007/978-3-642-39330-3_57)
40. Kumar, C., Heuten, W., Boll, S.: Geographical queries beyond conventional boundaries: regional search and exploration. In: *Proceedings of GIR 2013*, pp. 84–85. ACM (2013)
41. Boyandin, I., Bertini, E., Bak, P., Lalanne, D.: Flowstrates: an approach for visual exploration of temporal origin-destination data. *Comput. Graph. Forum* **30**, 971–980 (2011)
42. Certo, L., Galvão, T., Borges, J.: Time automaton: a visual mechanism for temporal querying. *J. Visual Lang. Comput.* **24**, 24–36 (2013)
43. Krüger, R., Thom, D., Wörner, M., Bosch, H., Ertl, T.: TrajectoryLenses - a set-based filtering and exploration technique for long-term trajectory data. *Comput. Graph. Forum* **2013**, 451–460 (2013)
44. Bonhomme, C., Trépied, C., Aufaure, M.A., Laurini, R.: A visual language for querying spatio-temporal databases. In: *Proceedings of GIS 1999*, pp. 34–39. ACM (1999)
45. D’Ulizia, A., Ferri, F., Grifoni, P.: Moving GeoPQL: a pictorial language towards spatio-temporal queries. *GeoInformatica* **16**, 357–389 (2012)

46. Monroe, M., Lan, R., Morales del Olmo, J., Shneiderman, B., Plaisant, C., Millstein, J.: The challenges of specifying intervals and absences in temporal queries: a graphical language approach. In: Proceedings of CHI 2013, pp. 2349–2358. ACM (2013)
47. Gotz, D., Stavropoulos, H.: DecisionFlow: visual analytics for high-dimensional temporal event sequence data. *IEEE TVCG* **20**, 1783–1792 (2014)
48. Fails, J., Karlson, A., Shahamat, L., Shneiderman, B.: A visual interface for multivariate temporal data: finding patterns of events across multiple histories. In: VAST 2006, pp. 167–174 (2006)
49. Dionisio, J.D., Cárdenas, A.F.: MQuery: a visual query language for multimedia, timeline and simulation data. *J. Visual Lang. Comput.* **7**, 377–401 (1996)
50. Jin, J., Szekely, P.: QueryMarvel: a visual query language for temporal patterns using comic strips. In: Proc. VL/HCC 2009, pp. 207–214 (2009)
51. Fegeras, L.: VOODOO: a visual object-oriented database language for ODMG OQL. In: W13. The First ECOOP Workshop on Object-Oriented Databases (1999)
52. Visual Analytics Community: VAST 2014 Challenge - the Kronos incident (2014). <http://vacommunity.org/VAST+Challenge+2014>
53. Krüger, R., Herr, D., Haag, F., Ertl, T.: Inspector gadget: integrating data pre-processing and orchestration in the visual analysis loop. In: EuroVis Workshop on Visual Analytics (EuroVA). The Eurographics Association (2015)
54. Bracciale, L., Bonola, M., Loreti, P., Bianchi, G., Amici, R., Rabuffi, A.: CRAW-DAD data set roma/taxi (v. 2014-07-17) (2014). <http://crawdad.org/roma/taxi/>