Institut für Computergraphik und Algorithmen
Technische Universität Wien
Karlsplatz 13/186/2
A-1040 Wien
AUSTRIA
Tel: +43 (1) 58801-18601
Fax: +43 (1) 58801-18698

Institute of Computer Graphics and Algorithms
Vienna University of Technology
email:
technical-report@cg.tuwien.ac.at
other services:
http://www.cg.tuwien.ac.at/
ftp://ftp.cg.tuwien.ac.at/

TECHNICAL REPORT

The Event Tunnel: Interactive Visualization of Complex Event Streams for Business Process Pattern Analysis

Martin Suntinger
SENACTIVE IT-Dienstleistungs GmbH

Hannes Obweger
SENACTIVE IT-Dienstleistungs GmbH

Josef Schiefer
Institute for Software Technology and Interactive Systems, Vienna University of Technology

M. Eduard Gröller
Institute of Computer Graphics and Algorithms, Vienna University of Technology

TR-186-2-07-07
May 2007

The Event Tunnel: Interactive Visualization of Complex Event Streams for Business Process Pattern Analysis

Martin Suntinger*
SENACTIVE IT-Dienstleistungs GmbH

Hannes Obweger†
SENACTIVE IT-Dienstleistungs GmbH

Josef Schiefer‡
Institute for Software Technology and Interactive Systems, Vienna University of Technology

M. Eduard Gröller§
Institute of Computer Graphics and Algorithms, Vienna University of Technology

May 14, 2007

Abstract

Event-based systems are gaining increasing popularity for building loosely coupled and distributed systems. Since business processes are becoming more interconnected and event-driven, event-based systems fit well for supporting and monitoring business processes. In this paper, we present an event-based business intelligence tool, the Event Tunnel framework. It provides an interactive visualization of event streams to support business analysts in exploring business cases and business processes. The visualization is based on the metaphor of considering the event stream as a cylindrical tunnel, which is presented to the user from multiple perspectives. In the Event Tunnel, relevant business events are laid out and depicted for analysts. The information of single events is encoded in event glyphs that allow for a selective mapping of event attributes to colors, size and position. Different policies for the placement of the events in the tunnel as well as a clustering mechanism generate various views on historical event data. The event tunnel is able to display the relationships between events. This facilitates users to discover root causes and causal dependencies of event patterns. Our framework couples the event-tunnel visualization with query tools that allow users to search for relevant events within a data repository. Using query, filter and highlighting operations the analyst can navigate through the Event Tunnel until the required information or event patterns become visible. We demonstrate our approach with use cases from the fraud management and logistics domain.

Keywords: Business process visualization, complex event processing, query-driven visualization.

1 Introduction

Today's networked business environment requires systems which are adaptive and easy to integrate. Event-based systems have been developed and used to control business processes with loosely coupled systems. One of the most promising concepts for gaining insight into business processes in order to support closed loop decision-making on an operative level is Complex Event Processing (CEP) [6]. CEP includes a set of technologies to process large amounts of events, utilizing them to monitor, steer and optimize the business processes with minimal latency. Typical application areas of CEP require fast decision cycles [8] based on a large number of observable business events which can be used to discover exceptional situations or business opportunities. Typically, these are areas like financial market analysis, trading, security, fraud detection, logistics like tracking shipments, compliance checks, and customer relationship management.

The success of event-driven business solutions depends on an ongoing learning process. It is an iterative cycle including the analysis and interpretation of past processing results and the conversion of these results into the event-processing logic. Analysis tools are required which are tailored to the characteristics of event data to answer questions like: Where did irregularities occur

*e-mail: martin.suntinger@senactive.com
†e-mail: hannes.obweger@senactive.com
‡e-mail: js@ifs.tuwien.ac.at
§e-mail: groeller@cg.tuwien.ac.at

in my business? Did processes change over time? Does my business slow down, or can certain processes be executed more effectively? Which of different execution paths of a process is most effective? Which contributors to my business process are most valuable? Upon which data were past, automated decisions made? Did errors occur in the automated decision process? What happened at a certain point in time at a certain location and who was involved? To answer these questions, the business analyst has to be equipped with extensive retrieval tools to extract required data sets. Also expressive visualizations are necessary to navigate through event data and recognize recurring patterns and irregularities that influence the business performance. In the following, we present an approach for visualizing event data, the Event Tunnel. It is based on the metaphor of considering the continuous stream of events as a tunnel, which we show from a top and a side view. Around this core visualization, we have built an analysis workspace which we will also briefly discuss in this paper.

The paper is structured as follows: In Section 2, we reflect on related work on event-based visualization approaches and discuss our contribution. Section 3 describes event spaces which are the data foundation for our visualization. Section 4 describes the visual metaphor of the Event Tunnel and its major components such as its views, the glyph-based event representation and the cluster display. In Section 5, we show how the event tunnel visualization is embedded in an analysis framework and discuss related implementation issues. In Section 6, we present evaluation results with respect to two use-case scenarios and, finally, with Section 7, we conclude our paper and provide an outlook for future work.

2 Related work

Currently, event-based systems and applications are gaining increasing maturity and are starting to be employed in industrial settings. Event-data specific visualization and analysis tools on the other hand are still in their infancy. In the domain of business intelligence and analysis, many approaches exist, from online-analytical-processing (OLAP) to data mining techniques. These technologies are applicable for the analysis of event-based operational business data as well, but they do not take into consideration the nature and special characteristics of events. Techapichetvanich and Datta [12] and Maniatis [7] describe visualization approaches for OLAP, which allow users to explore and analyze data cubes and data warehouses without generating sophisticated queries. Users can gain both overviews and refine views on any particular areas of interest through the combination of interactive tools and navigational functions (i.e., drill-down, roll-up, and slicing). In comparison to OLAP analysis, our approach consciously omits an abstraction of the data into a set of multidimensional key figures and focuses instead on the events as the data points for the visual data representations. High-level views on the data are provided by visual patterns and clusters, and drill-down operations are possible down to the level of a single event.

Hao et al. [4] presented VisImpact, an approach to reduce the complexity of business data by extracting impact factors that identify either single nodes or groups of nodes from business flow diagrams that influence business operations. VisImpact is able to find relationships among the most important impact factors and supports an immediate identification of anomalies. Our approach is not dependent on process models and is able to visualize process-behavior patterns based on events generated from IT-systems. It does not reduce the complexity of the data, but allows investigating causal dependencies of events in a business environment. By extracting and visualizing manageable data sets, a user is able to discover (hidden) relationships between events as well as impact factors for business operations.

Kapoor et al. [5] proposed OPAL, which uses classical visualization techniques (charts, time tables, histograms, scatter plots) for operational, multidimensional data about business operations. OPAL helps users in making faster and more well-informed decisions in order to respond to irregular process patterns.

Rudensteiner et al. [10] implemented the XmdvTool for multivariate data visualization that allows users to view data from different perspectives. The XmdvTool supports a variety of advanced visual interaction techniques, including brushing in screen space, data space, and structure space, panning, zooming and distortion. We extended this approach by a visualization technique and show events without further abstraction in the context of their occurrence. Condition-based mappings enable the encoding of various event-data aspects in the rendering.

The event-tunnel visualization is designed to display search results from event-data queries. Rosznyai et al. [9] proposed the search-engine Event Cloud for analyzing business events based on event queries. The system correlates and indexes the events in a data staging process and enables users to search in large sets of historical events. Event Cloud allows to flexibly retrieve historical event data and uses a text-based view for displaying

the search results. This is supported by the work of Sebrechts et al. [11], who showed that locating a search target is fastest in text-based views. We hypothesize that for event data, a visual representation of query results is more valuable for answering questions of a business user, since it shows the user multiple perspectives of a search result and allows at the same time to graphically display the relationships of the retrieved events.

In the domain of information visualization, different approaches exist to display the time dimension for data. Carlis and Konstan [2] and Weber et al. [13] proposed a time spiral, aiming on temporal patterns in periodic data. Arranging data in a spiral, their approaches provide the user with easy visual cues to both serial and periodic aspects of the data, along with interactions, such as the change in period over time. Yee et al. [14] show a visualization approach which uses a radial tree layout method for supporting the interactive exploration of graphs. In our approach, we combine and extend these techniques for displaying networks of correlated events which occur over a certain period of time.

3 The event space

Continuous capturing and processing of events produces vast amounts of data. Event-based business process analysis relies on these data. An efficient mass storage is required to store all events and prepare the data for later retrieval. Throughout this paper, we refer to this mass storage as the event space. During the processing of the events in the event-based system, they are captured by an auditing service and stored in a data repository. Furthermore, the events are indexed for quick retrieval of correlated events and metrics are pre-calculated. Figure 1 illustrates this process.

For the purpose of maintaining information about business activities, events capture attributes about the context when the event occurred. Event attributes are items such as the agents, resources, and data associated with an event, the tangible result of an action (e.g., the placement of an order by a customer), or any other information that gives character to the specific occurrence of that type of event. For example, a typical order event could have the following attributes as context information:

- Customer Name (string data attribute)
- Order ID (string data attribute)
- Product ID (string data attribute)
- Price (numeric data attribute)

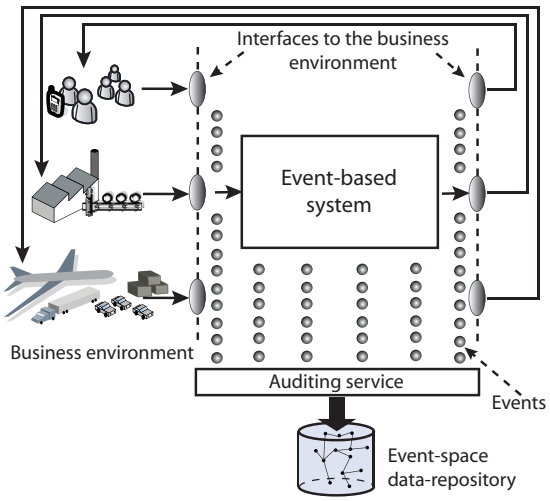

Figure 1: The event-space backend system contains processed events for analysis and retrieval.

This template of attributes defines the structure of a certain class of events and is called event type. It indicates the underlying type of state change in a business process that is reflected by the event.

To back-trace and analyze the events of a business process, it is important to correlate temporally and semantically related events. Elements of an event context can be used to define a relationship between events. Chains of correlated events reflect instances of a business process (see figure 2).

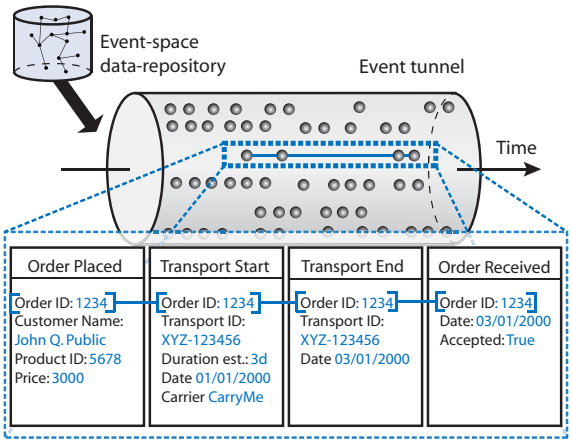

Figure 2: A business process instance is given as a set of correlated events. The semantic correlation is based on the event attribute *order ID*.

The event-tunnel visualization follows a query-driven approach. A business analyst can define queries for ex-

tracting data from the event-space data repository in order to retrieve and prepare data sets for the visualization. We have unified the access to event data by specifying event filters and patterns with a so-called event-access expression language. This language allows to easily access event attributes and supports the modeling of complex conditions including calculations and aggregations. A typical search query is a set of conditions which is used to filter the events from the event space. A simple example of an EA expression is: *Order.Price BETWEEN 100 AND 200 AND OrderDelivered.Status = 'Delayed'*. As the example shows, the query model allows defining a search scope for retrieving correlated events that match a set of conditions. Throughout this paper, we will demonstrate that our query model is applied for multiple purposes. Conditions separate the event data into (eventually overlapping) groups. Condition-based coherences can then be visually mapped to color, spatial proximity and other characteristics of a visualization.

4 The Event-Tunnel Visualization

In the previous section, we presented the event space as a queryable data repository for historical events. In this section, we propose a visualization technique called event tunnel to produce a highly interactive, visual depiction of this data. The event-tunnel visualization is based on the metaphor of seeing the past stream of events as a 3D cylinder and providing two views onto this cylinder: a top view that looks into the stream of events along the time-axis, and a side view plotting the events in temporal order (see figure 3).

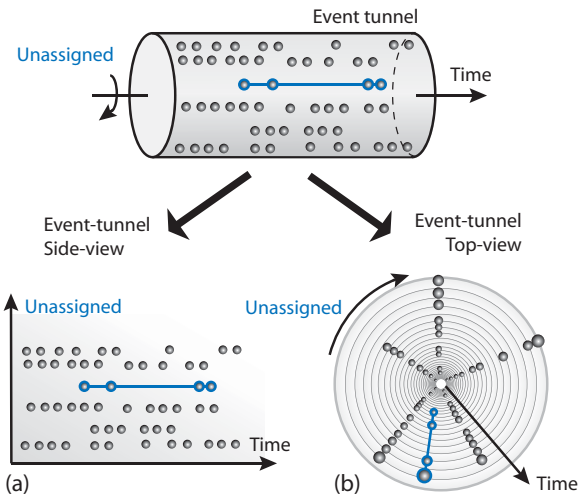

Figure 3: The event-tunnel visualization: Side view and top view onto the stream of events.

Following the metaphor of an event-stream cylinder, one axis is determined and occupied unambiguously by time, whereas the remaining axis is assignable by a placement policy. For the top view and the side view, we will show placement policies that either aim on efficient space filling or try to encode data attributes in the spatial arrangement.

4.1 Event-Tunnel Top-View

The event-tunnel top-view maps the 3D view into the event-stream cylinder to a radial 2D rendering (see figure 3b). Events on the inner circles of the tunnel are displayed smaller to simulate perspective projection. The motivation behind this technique is an observation we made early in the planning phase: During the analysis of historical event data we found out that the entropy and business value of the most recent events are usually higher in comparison to similar but older events. This tendency results from the fact that the most recent events are more relevant for a business analyst as well as the final events of a business process generally characterize best the outcome of a business case. The perspective projection in the event-tunnel top-view reflects this requirement to emphasize recent events. In the tunnel, the latest events appear larger at the outer rings and are well visible.

4.1.1 Placement policies

Following the underlying metaphor of the event tunnel, the distance from the center of the top view is defined by the dimension mapped to the cylinder axis, i.e., usually time. Another parameter, the angular position of each event, is not implicitly defined and can be controlled by the business analyst with placement policies.

The strategies for effectively placing the events on the concentric rings of the tunnel depend on the objectives of the business analyst. Possible objectives might be:

- Avoid overlapping events
- Display correlated events close to each other
- Keep sequences of events on nearly straight lines to be able to easily recognize process behavior patterns

It is hard to find a single strategy that fits all requirements at the same time. However, it is possible to define specialized policies that concentrate on certain aspects and try to optimize the output with respect to these aspects. Ultimately, the user can switch between the placement strategies to find the output that is most effective

for his/her purposes. In the following, several placement policies are proposed, and their results are discussed.

4.1.2 Sector placement-policy

The sector placement-policy focuses on distributions of events. Events are placed in non-overlapping, angular sectors, depending on their particular attributes. The overall appearance of the outcome resembles well-known pie charts. *Conditions* are used to define sector-memberships, though restrictions must be introduced to avoid multi-set memberships. When using numeric event attributes for the sector distribution instead of having discrete sector conditions, additional expressiveness can be achieved. In this case the events are arranged around the tunnel surface by performing a linear mapping of the event attributes to the position.

In many cases, the sector placement policy shows expressive results on medium to large event sets (from several hundreds up to 40,000 events). Accumulations, missing events over time, as well as single, isolated outliers become visible. Combined with adequate color and size configurations, patterns can be discovered across the sectors.

4.1.3 Centric event-sequence placement-policy

The centric event-sequence placement-policy (CESP policy) is focused on sequences of correlated events, such as the events of a business process instance. A position of an event depends solely upon its membership in a group of correlated events. The CESP algorithm follows event sequences and plots them in coherent chains.

Algorithm
4.1: CESPLACEMENT(*events*)

comment: Plot *events* according to the CESP

ORDERBYTIME(*events*)
for $i \leftarrow 0$ **to** *events.length*
do
- **if** $i = 0$
  - **then** PLOTEVENT(*events*[*i*])
- **else**
  - *events*[*i*].*Angle* $\leftarrow$ *events*[$i-1$].*Angle*
  - **if** OVERLAP(*events*[*i*], *events*[$i-1$])
    - **then** SHIFTCLOCKWISE(*events*[*i*]) (i)
  - PLOTEVENT(*events*[*i*])

Algorithm 4.1 avoids overlapping of events by clockwise shifts: on line (i), the event's representation is moved clockwise as far as not to overlap its predecessor. The algorithm results in specific patterns, being abstract representations of underlying business process instances. As regular instances of a business process proceed more or less according to a template, characteristic patterns emerge. This enables the analyst to derive fuzzy, visual template representations for certain business processes. Anomalies can then be assessed by comparing the resulting pattern with the expected template. Therefore, the CESP policy is well suited for the analysis of smaller data sets to detect fine-grained causal relationships. Although most of the emerging patterns are application and data specific, several basic, high-level patterns can be characterized. Figure 4 schematically shows these patterns. Table 1 lists how these patterns can be interpreted.

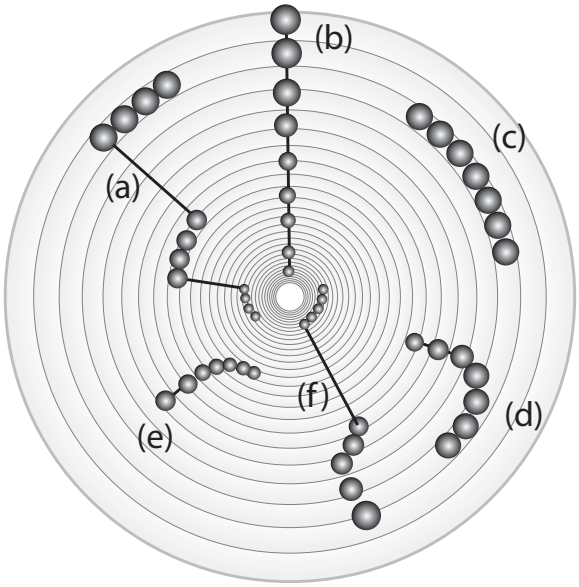

Figure 4: Examples of characteristic business-process patterns in the CESP policy. Refer to table 1 for an explanation of the individual patterns.

4.2 Event-Tunnel Side-View

Some of the event-tunnel top-view's most valuable characteristics result from the perspective projection. On the other hand, the perspective projection results in a non-linear transformation from time to distance, which makes it difficult to estimate absolute time values. In some cases, an adequate representation of the event space's temporal relationships is indispensable to a detailed analysis process. Therefore, we provide an additional view of the event tunnel, the side view (figure 3a).

Table 1: Interpretation of business-process patterns in the CESP policy.

Pattern	*Interpretation*
Stair pattern (a)	Represents a business process whose execution is characterized by several idle times (potentially exceptional delays).
Non-interfering chain (b)	A long-running process, whose stages are passed straight forward in regular time steps.
Parallel chain (c)	A fast-executing process (probably automated or machine controlled) without idle times.
Acceleration worm (d)	Represents a process whose execution accelerated continuously.
Deceleration worm (e)	Represents a process whose execution decelerated continuously.
Rattlesnake (f)	Reflects one extreme delay in the execution of a process.

It is intended as an extension to the event-tunnel top-view that accurately presents temporal coherences.

For the event-tunnel side-view, the question arises again how to utilize the unassigned axis of the tunnel. We decided to exclusively focus on temporal relationships between correlated event sequences and plot these sequences on horizontal lines in temporal order. Event sequences (which reflect business process instances) appear as continuous bars surrounding the single events contained in the sequence. The result closely resembles process charts such as GANTT diagrams without the depiction of dependencies.

4.3 Mapping data to event glyphs

The placement policies described above work on the basis of events' characteristics either attribute values or correlation-group memberships. Hence, placement policies perform a mapping of data attributes to position. Another possibility to visually encode event attributes is the shape of a single event itself. An event is represented by a spherical event glyph (see figure 5). The event glyph consists of a sphere surrounded by an outer ring. The size and color of these elements do encode event attributes. The sphere surface and the surface of the outer ring may be subdivided into more than one sector to simultaneously encode several attributes.

4.3.1 Size

By its nature, coding the size parameters of the event glyphs is suitable for the mapping of continuous attributes. Per event type, separate mappings can be defined for the sphere diameter and the diameter of the outer ring.

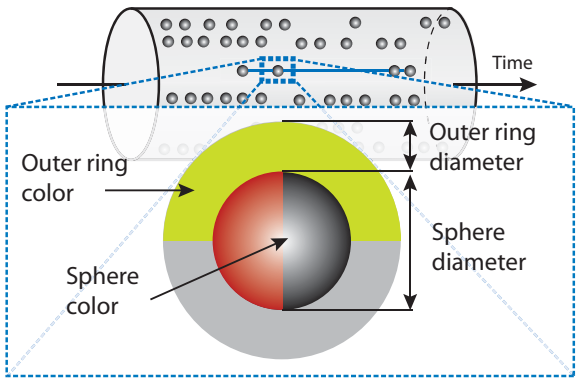

Figure 5: Event attributes can be mapped to colors and sizes of event glyph parameters.

4.3.2 Color

In contrast to size coding, the color parameters of the event glyph can be utilized to visualize arbitrary event attributes. Anyhow, the complexity of analysis tasks requires a flexible and powerful approach far beyond the basic mapping of single, discrete attributes to color. Therefore, the application enables the user to define conditions, each associated with a customizable color. In case of an event fulfilling a condition, the event's bullet area is filled with the respective color. As described in section 3, conditions may lead to multi-set memberships. This is incorporated by subdividing the event-glyph sphere into differently colored sectors of equal size.

4.4 Displaying event correlations

So far, in the event-tunnel side-view correlations between events are naturally encoded and visually depicted. In the event-tunnel top-view the only technique available to visualize correlations is via the spatial arrangement (CESP policy). In the course of this work, we elaborated an additional technique to visualize event correlations in the top view. The approach simply connects related events on demand. The connections are drawn as semi-transparent bands. Each band reaches from one event to the next, in temporal order. The thickness of the band is continuously adjusted to the events' sizes.

4.5 Drawing clusters

The techniques presented so far are suitable for small and medium sized data sets of up to 10,000 events.

A distribution analysis in the sector placement-policy can also be performed with data sets of up to 40,000 events. To cope with result sets that are larger than that, we implemented a clustering mechanism that aggregates groups of events into single data points. The implementation supports the formation of clusters based on correlations, equality of event types and arbitrary event-access expression conditions. Each cluster possesses an anchor event (see figure 6a): The temporal occurrence of this event determines the plotting position of the cluster in the event tunnel. A cluster can be collapsed or expanded. In the collapsed state, its appearance resembles a single event node, in the expanded state, the events contained in the cluster are plotted on a resizable circle. The metric determining the events' distance from the center of this circle is customizable (figure 6b). Expanded clusters can be dragged around by the user: a thin line indicates the connection to the anchor event. Clustering does not only aim at a sparse visual representation of large result sets. The organization of events in clusters additionally allows a distribution analysis and a direct comparison of distinct event groups. For example, having grouped the events from business process A in cluster 1 and the events from business process B in cluster 2, the analyst can bring the clusters to an equal size and compare them to each other. For the generation of clusters, we support both automatic clustering of the complete result set or selective clustering of a selection of events. To cluster the complete result set, the user can define so-called clustering rules. These rules configure the event aggregation (e.g., generate one cluster for each correlation chain) and the selection of the anchor event in each cluster.

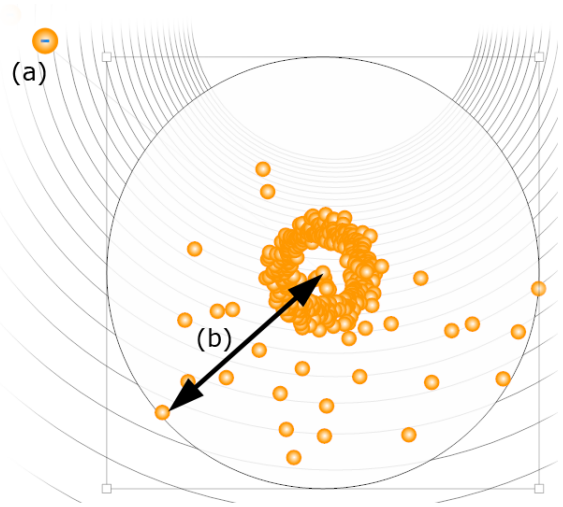

Figure 6: A cluster of transport events. The anchor event (a) indicates a demand; the events in the cluster are transports available to fulfill the demand. Distance metric in the cluster (b): estimated transport costs.

The representation of clusters that are in the collapsed state is similar to an event glyph. This again offers several parameters (colors and sizes) to map data. In this way, the mapping characterizes the aggregated group of events. For example, if a cluster aggregates the events of one correlation, the metrics defined for this correlation can be mapped to the cluster glyph. In the simple transport chain correlation introduced in figure 2, possible metrics to map could be transport duration, transport costs, satisfaction status, shipment delay and order volume.

5 The Analysis Framework

The event-tunnel top and side view visualizations generate interactive views on the event data. We have embedded these views into a configurable workspace for event analysis and mining purposes. It is designed as an extendable framework offering the possibility to plug-in customized event-data visualizations without having to implement basic functionalities. Besides the event-tunnel views, the analysis workspace includes facilities for querying and retrieving event-data from the event-space data-repository, selective filter operations, displaying metrics and the content of events, discovering similarities of correlated event sequences, and managing analysis results. All visualizations are linked together. Highlighting an event in one of the views highlights it in all other views. Figure 7 provides a screenshot of the analysis workspace.

Besides the event-tunnel top-view (a) and the event-tunnel side-view (b), the analysis framework currently consists of a text view (c) that generates a text-based output of an event query result, a metric charting view (d) that plots calculated metrics for selected business processes, a text box (e) that displays information on the current selection in the visualizations, a graphical query builder (f), a filter manager (g), a management console for clustering (h), a configuration manager to control the mapping of data attributes, select placement strategies and configure a condition-based highlighting (i) and a snapshot management console (j) to capture the current view state and configuration and restore past analysis results. The analysis framework targets business analysts and developers of business logic which is based on company-specific event processing. The goal is to accomplish back-tracing tasks, draw conclusions on the

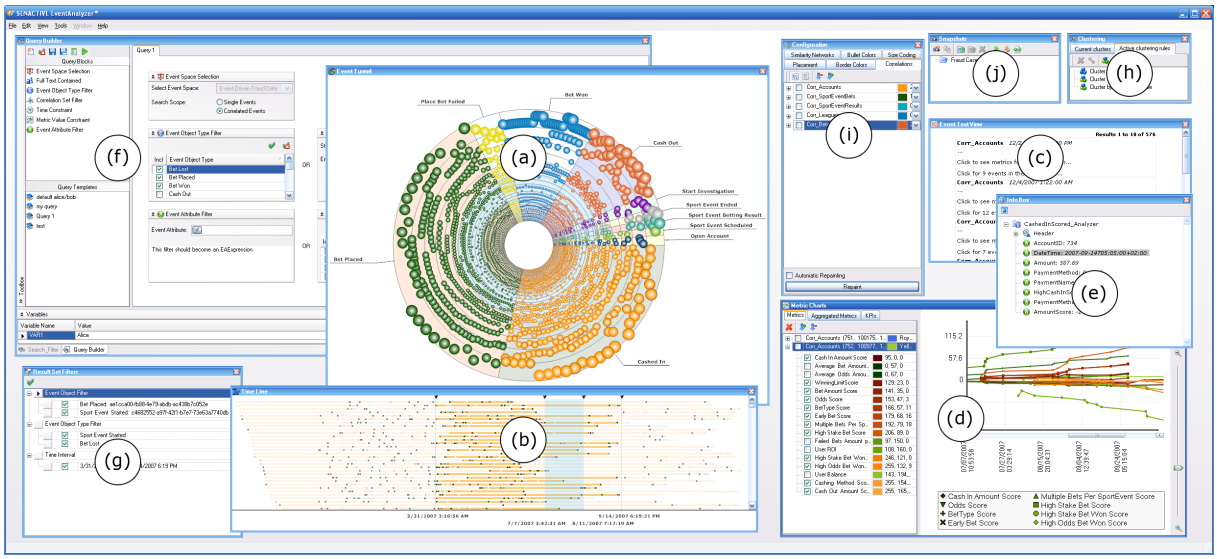

Figure 7: The analysis workspace: event-tunnel top-view (a), event-tunnel side-view (b), text view (c), metric charting view (d), text box (e), graphical query builder (f), filter manager (g), clustering management console (h), configuration manager (i), snapshot management console (j).

business performance and discover undetected correlations and patterns. These findings can then be employed in the automated event processing.

The analysis framework has been written in C#. For enabling simple plug-in integration, it is built upon the Castle Project's Windsor IoC container for managing modules and components. We tried to keep the visualizations as slim as possible. Therefore, we provide a range of interfaces, each with a certain limited scope. A visualization module *can* be notified of event selections, and it *can* support clustering, each through implementing the respective interface.

The introduced views on the event tunnel combine an extensive graphical rendering with a rich user interaction model. Therefore, we made use of the Piccolo.NET framework [1], a graphical toolkit that supports zooming and panning as well as object-level event handling. Due to the high computational effort of transformation operations, we decided to reduce rendering details to an absolute minimum during zooming and panning.

6 Applications and Results

To evaluate the above-presented visualization techniques, we have applied the analysis framework for two selected business applications: automated fraud detection in online betting platforms and real-time monitoring of logistics and transportation processes. We experienced that a typical analysis task can be split up into:

- data tracing
- anomalies discovery
- pattern characterization

Data tracing means the process of a step-wise reproduction of recent, influential occurrences. Typically, this step is triggered by some clue that serves as the starting point for further investigation efforts. The framework's query-driven visualization-approach complies well with data tracing tasks: The initial clue can be translated to a query in order to extract a limited dataset for the visualization. The analyst can subsequently narrow or broaden the search scope to the required granularity level.

The discovery of anomalies in data sets was the subject of many recent research efforts [3] [4]. Our approach is currently limited to a manual anomaly-discovery process supported by the following techniques: Outlier detection in the sector placement policy, accentuation of outliers in event clusters, pattern comparison in the CESP policy, highlighting of abnormal event attribute values by color and size coding and interpretation of temporal accumulations in the event-tunnel top-view and the event-tunnel side-view.

After a data tracing step, information on anomalies and conspicuous occurrences is available. However, in order to exploit this information, in most cases a generalization and characterization of the discovered patterns is essential. The characterization of a pattern includes influential factors (i.e., key figures and threshold levels) as

well as behavioral patterns in event sequences, whose combination makes up a reference pattern. This reference pattern can be used by event-based systems for discovering similar cases. Tools that support the characterization of reference patterns are the event tunnel itself, the text view and the metrics view. From the event tunnel, the analyst can extract a sequence pattern of events and generalize it to an event sequence model. The metrics view shows the temporal evolution of key figures which helps to assess threshold levels. Finally, from the text view the events' exact data values can be extracted.

6.1 Automated Fraud Detection and Prevention

Fraud detection and prevention is a major issue in technology-driven business domains relying on online payment solutions and web-based customer interactions. A market that has been heavily affected by fraud cases recently is online betting and gambling. Various forms of fraud have been reported, reaching from physical attacks by hackers over money laundering to the abuse of insider information about sport events. With event-based systems, fraud cannot only be detected, but proactively prevented in near real-time by a continuous rule-based evaluation of customer interactions. An automated intervention is then possible in case of a conceived suspicion [8]. To implement an efficient fraud detection and prevention system, exhaustive knowledge on the underlying business processes and occurring fraud patterns is a prerequisite. Continuous learning and the knowledge of chain fraud patterns are necessary to keep an event-based system up-to-date [3]. We propose the event-tunnel analysis-framework to attain this knowledge and evaluated it with an event-based fraud detection and prevention system for online betting providers. The data in the following examples were generated using a simulation model that emprises known user-behavior patterns in the simulated events. For the evaluation, we simulated ordinary bet placing, cash-in and cash-out events and randomly added about three percent fraud cases from several available fraud templates. The templates were parameterized to vary in structure and degree of conspicuity.

6.1.1 Tracing of Customer Activities

In a requirements study for an online betting provider we observed that a complete department was concerned with data tracing. For fraud investigations, security analysts had to regularly back-trace customer actions in case of system alerts. Fraud investigations target single users or groups of users or the analysis of complete markets and leagues. In addition, automated system interventions are regularly investigated. In the following, we assume that an event-driven fraud-detection system automatically generates alerts for users with suspicious behavior. The security analysts regularly receive clues (the account IDs of users with fraudulent behavior) that serve as starting points for a data-tracing analysis step. In figure 8 an excerpt of a step-wise tracing and discovery process is shown. It is based on an illustrative case discovered by a test-user in the simulated data set. Figure 8 shows the historical events of two user accounts that are to be investigated. Both users triggered an alert for the same sport event and bet type. The plot shows that they failed to place a bet because the system recognized them as officials. It is further visible that the sequence of actions of both users strongly correlates. The bet placing failed for both users nearly concurrently. Mapping the event attribute *bet type* to color reveals that both users exclusively place bets of type *free throws*.

The graphical rendering of events together with the density of coded information allows security analysts a fast reconstruction of recent incidents. In addition, the two user profiles can directly be compared: Concurrent accumulations of bets or similar behavioral patterns become visible and may expose networks of fraudulent actors. The efficiency of the event-tunnel visualization in comparison to a text-based output of the sequence of user events results from the ability to display multidimensional information in one view: While a text-based list of events provides one degree of freedom (the sequential order), the event-tunnel visualization provides a time-based display, two size dimensions (sphere diameter and outer-ring diameter) and several color dimensions. The advantage over a text-based view becomes more obvious in the analysis of multiple account profiles: The exact temporal order of all users' events is reflected in the visualization and temporal coherences are immediately visible.

6.1.2 Discovering Anomalies in Betting Behavior

Data tracing activities focus on a very specific business entity, e.g., a user, and track the events related to this entity. A more general approach is to start at a higher granularity level and drill down to more detailed data for discovering different types of anomalies. The previous example showed bet placing failures in the profile of two users. The bet type in the example was *free throws*. Having this information available, the analyst can broaden the search scope to all recent bets of the type *free throws* in a certain market. Figure 8b shows

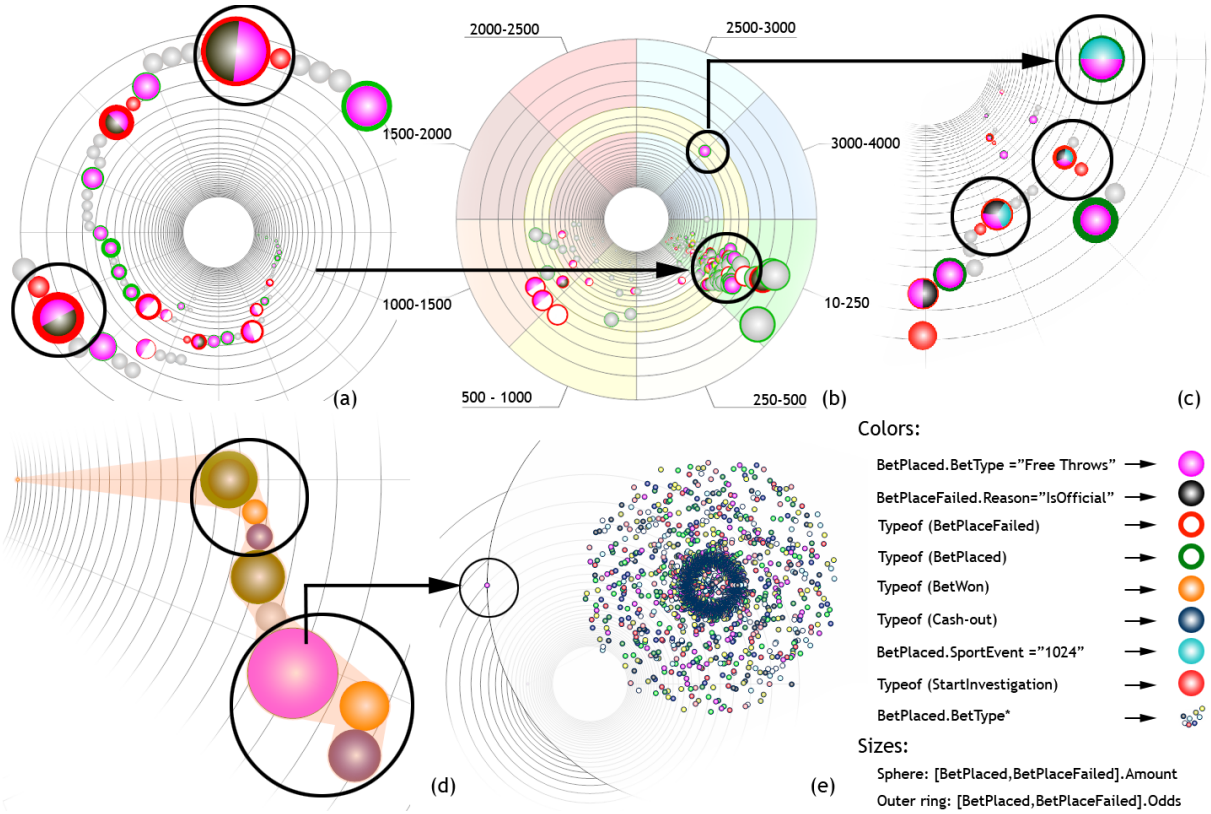

Figure 8: Example of a step-wise investigation of online-betting data. Account histories of two users (a), bet amount distribution for a selected sport event (b), suspicious occurrences at a certain point in time (c), a putter-on account profile (d), cluster of high-stake bets with abnormal outlier (e).

a plot of bet events with the sector placement policy. The bets scatter around the tunnel according to the bet amount. The bet placing failure events detected in the previous step appear in the bulk of average bet amounts. The plot exposes one salient outlier temporally related to the detected bet placing failure events. For the further analysis steps, this data point presents a valuable link to conspicuous data. In figure 8c the account history events of the user who placed this bet are plotted in relation to the already detected suspicious account profiles. From the color and size mapping the analyst can conclude that the third user successfully placed a high stake bet equivalent to the bet that failed for the two other users. One possible interpretation could be that user three is a so-called ”putter-on” who placed bets for people that are prohibited to bet, i.e., officials and players. This hypothesis is supported by the account profile of user three plotted in figure 8d. The marked areas represent a recurring sequence of bet placing, bet won and immediate cash-out. This sequence characterizes the putter-on pattern. One strength of the event-tunnel analysis method becomes visible in the example: A single visual clue in a plot can immediately be exploited to navigate to formerly unconsidered data. We experienced that in practice such navigation chains can be followed by analysts over more than 10 steps before performing a completely new query. For example could step 8(d) lead to a further broadening of the search scope to investigate if the high-stake bet of user three should have been prohibited by the system. Figure 8e plots recent high-stake bets in a cluster. The plot shows that one bet is an outlier and therefore abnormal. This could be a data error or a failure of the event-processing system.

The above example demonstrated that anomaly discovery is currently supported by the event-tunnel analysis-framework in the sense of checking and exploring interesting or exceptional situations. When discovering outliers, a user can continuously detect anchor points in the navigation chain. One major advantage over classical methods like histograms or scatter plots is that the level of abstraction in the output does not have to be changed. In a cluster, the scattered data points are still single events embedded in their context. The techniques for size mapping and color coding can be applied in

the same way as with other placement techniques. One trade-off is the difficulty of labeling. In comparison to a diagram, reading the exact data values from the visualization is hardly possible. We tried to alleviate this problem by showing tool tips with event data information and metrics to the user when hovering with the mouse over an event.

6.2 Real-Time Logistics and Resource Planning

Transportation and logistics service providers are finding themselves confronted with increasing cost pressures. Many organizations have already introduced large-scale IT systems to increase the efficiency in logistics management. The goal is to improve and shorten supply chains to provide just-in-time deliveries.

In the area of analysis and visualization approaches in the supply chain domain, two types of systems are relevant to logistics applications: Geographical tracking systems like popular parcel trackers and business intelligence tools that are able to extract and analyze key figures on the organization's performance. A combination of these techniques covers the current scope of logistics management systems. But with the integration of market and production planning into one homogeneous system, supply-chain tracking exceeds the potential of current visualizations when the analyst is concerned with an in-depth analysis of causal and temporal dependencies of incidents. In an event-driven approach, all the required information from the production systems over the transport chain to the market is incorporated and unified in the captured events. In the following, we present typical analysis tasks that can be accomplished with the event-tunnel analysis-framework. The results are based on long-term tests of an event-driven logistics application with simulated data.

6.2.1 Supply-Chain Tracing

Questions like: "What delayed the delivery of order 12345?", "Why are the transports of carrier A slower than those of carrier B" or "Why is the current demand in Madrid not covered and the stock level continuously increases?" arise regularly in the daily business of logistics managers. All these questions contain a valuable clue, an order ID, a carrier name or a location, that allows querying the event space for events related to these entities. An immediate reconstruction of related occurrences is possible. Figure 9a shows a plot of logistics supply chains related to a selected location. The sequence of occurrences can be interpreted as follows: After a demand is detected (*demand* event, figure 9a.1), available transports are proposed by the system (multiple *available transport* events, figure 9a.2). After a transport is chosen, the shipment is created and the transport starts (*transport start* event, figure 9a.3). Concurrently, several internal checks are performed, reflected in a set of simultaneous events (figure 9a.4). After the transport is completed, a final fulfillment check is performed (*fulfillment checked* event, figure 9a.5). This sequence of activities presents a template of the company's supply-chain processes. If a process strongly diverges from this template, an abnormal process execution can be assessed.

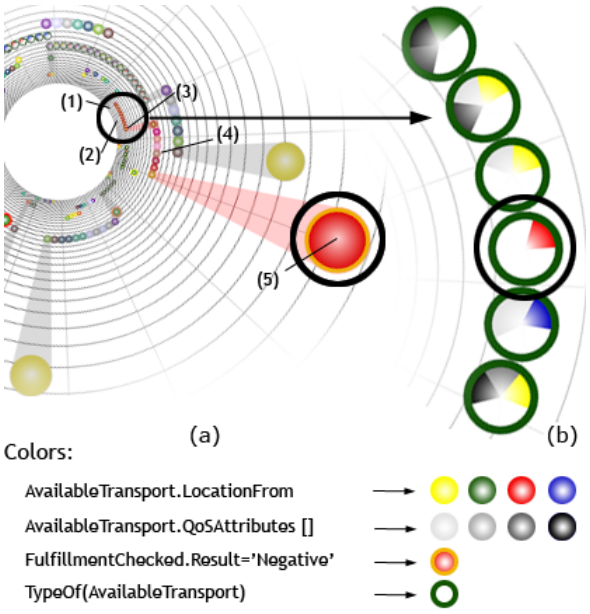

Figure 9: Supply chains in an event-driven logistics application (a); available transports and their decision-relevant parameters (b)

6.2.2 Evaluation of Automated System Regulations

By tracing supply chains, the analyst is able to perform root cause and cause-effect analyses. Especially when systems automatically carry out decisions made in the planning process (e.g., automated reorganization of transports, reordering of products for stock level regulation), it is crucial to monitor these decisions and to perform regular checks. Knowing the parameters that influence automatic decision processes, the event-tunnel analysis-framework helps to evaluate the system behavior. Figure 9b illustrates this process: As described above, transports are automatically scheduled by the

system if a demand notification is received. The system assesses all available transportation entities (carriers) and supply locations in real-time according to the following parameters: current stock level at the supplying location, transport costs, transport duration and recent carrier reliability. We have configured a relative mapping of these four parameters to sphere colors: The best values in the result set map to white, the worst to black. The six available transport events shown in figure 9b do provide varying quality of service (with respect to the current demand). In this case, the system has chosen the transport from London (red *location* mapping) to fulfill a demand in Madrid. The transport results in a negative fulfillment check (see highlighted *fulfillment checked* event in figure 9a). One interpretation could be that the system omits decisive parameters. In this case, the spatial proximity of the demand and available transport location was not taken into consideration.

7 Conclusion and future work

In this paper, we presented the event-tunnel visualization, an interactive view into a stream of complex business events. We demonstrated the integration of the event tunnel into a complete analysis workspace coupled with query mechanisms and various configuration options. The possibility to encode multiple data dimensions in colors and sizes and to configure placement policies enables precise investigations and allows to track and pinpoint specific occurrences and coherences. We intentionally chose a simple metaphor for our time-centered event visualization, i.e., a cylindrical tunnel which we show from a top and a side view. Early user feedback on the system confirms that this metaphor is understood quickly and eases the analysis work. Event clustering enables the handling of large data sets by providing an in-place drill-down exploration mechanism. It supports the analysis of value distributions among groups of correlated events.

We set our focus on data navigation which is reflected in the evaluation results. The integration with query and filter mechanisms allows for a continuous oscillation between overview and detail, from detected outliers and visually depicted conspicuities to event-level traces. We consider the presented implementation as a solid basis that will be extended by future projects. For example, condition-based operations (i.e., coloring, filtering, querying) are limited to absolute matching of event data attributes. Semantic similarity operations would enrich the power of these retrieval and highlighting mechanisms. The characterization of event-sequence patterns is one of the strengths of the event-tunnel visualization. Mechanisms to extract data that follow a given visual reference pattern would be a logical extension to the current query engine.

Acknowledgements

The authors thank the development team at SENACTIVE IT-Dienstleistungs GesmbH for their outstanding support throughout the implementation of the analysis framework.

References

[1] B. B. Bederson, J. Meyer, and L. Good. Jazz: an extensible zoomable user interface graphics toolkit in Java. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 171–180, New York, NY, USA, 2000. ACM Press.

[2] J. V. Carlis and J. A. Konstan. Interactive visualization of serial periodic data. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 29–38, New York, NY, USA, 1998. ACM Press.

[3] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

[4] M. C. Hao, D. A. Keim, U. Dayal, and J. Schneidewind. Business process impact visualization and anomaly detection. *Information Visualization*, 5(1):15–27, 2006.

[5] S. Kapoor, D. L. Gresh, J. Schiefer, P. Chowdhary, and S. Buckley. Visual analysis for a sense-and-respond enterprise. In *IASTED Conf. on Software Engineering*, pages 136–141. IASTED/ACTA Press, 2004.

[6] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[7] A. S. Maniatis, P. Vassiliadis, S. Skiadopoulos, and Y. Vassiliou. Advanced visualization for olap. In *DOLAP '03: Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, pages 9–16, New York, NY, USA, 2003. ACM Press.

[8] T. M. Nguyen, J. Schiefer, and A. M. Tjoa. Sense & response service architecture (SARESA): an approach towards a real-time business intelligence solution and its use for a fraud detection application. In *DOLAP '05: Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, pages 77–86, New York, NY, USA, 2005. ACM Press.

[9] S. Rozsnyai, R. Vecera, J. Schiefer, and A. Schatten. Event cloud - searching for correlated business events. To appear in *The 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, 2007.

[10] E. A. Rundensteiner, M. O. Ward, J. Yang, and P. R. Doshi. XmdvTool: visual interactive data exploration and trend discovery of high-dimensional data sets. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 631–631, New York, NY, USA, 2002. ACM Press.

[11] M. M. Sebrechts, J. V. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller. Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 1999. ACM Press.

[12] K. Techapichetvanich and A. Datta. Interactive visualization for olap. In *Proceedings of the International Conference on Computational Science and its Applications*, pages 206–214, 2005.

[13] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001*, pages 7–14, Washington, DC, USA, 2001. IEEE Computer Society.

[14] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001*, pages 43–51, Washington, DC, USA, 2001. IEEE Computer Society.