# Real-Time Visual Analytics for Event Data Streams

Fabian Fischer
University of Konstanz,
Germany
Fabian.Fischer@uni-konstanz.de

Florian Mansmann
University of Konstanz,
Germany
Florian.Mansmann@uni-konstanz.de

Daniel A. Keim
University of Konstanz,
Germany
Daniel.Keim@uni-konstanz.de

## ABSTRACT

Real-time analysis of data streams has become an important factor for success in many domains such as server and system administration, news analysis and finance to name just a few. Introducing real-time visual analytics into such application areas promises a lot of benefits since the rate of new incoming information often exceeds human perceptual limits when displayed linearly in raw formats such as textual lines and automatic aggregation often hides important details. This paper presents a system to tackle some of the visualization challenges when analyzing such dynamic event data streams. In particular, we introduce the Event Visualizer, which is a loosely coupled modular system for collecting, processing, analyzing and visualizing dynamic real-time event data streams. Due to the variety of different analysis tasks the system provides an extensible framework with several interactive linked visualizations to focus on different aspects of the event data stream. Data streams with logging data from a computer network are used as a case study to demonstrate the advantages of visual exploration.

### Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*; H.1.2 [**Models and Principles**]: User / Machine Systems—*Human information processing*

### Keywords

data streams, event processing, event streams, real-time, visual analytics, visualization

## 1. INTRODUCTION

Event-based data streams can be found in many applications and domains. A single event can be seen as "single, time-stamped item" [11]. In the domain of system administration many real-time streams, with events matching this generic event definition, can be found. Each computer system in a large network is regularly producing status and error messages. To make use of this information most system administrators collect those messages in a centralized way to enhance accessibility and data security. A large percentage of successful attacks of computer systems could have been clearly identified when someone would have paid attention to the log data in time. This situation highlights that it is important that the system administrator is able to analyze the data and to monitor critical systems in real-time to recognize anomalies or to find occurring problems as soon as possible. The same is also true for many other domains.

The three main contributions of our work, which explicitly focus on data stream *visualization* issues, are 1) a generic processing and analysis architecture for event data streams to support real-time visualization applications, 2) a system for pluggable visualizations for *real-time* and *historical* event data and 3) a dynamic timeline visualization to directly interact with multiple streams to also address the challenge of visualizing highly co-occurring events. Moreover several requirements are introduced which help to develop data stream visualizations. Additionally, the integrated feedback loop makes it possible to push gained insights back to the analysis system, which directly affects the ongoing scoring and classification process. Therefore, this interactive system can offer more than just pure information visualization and brings visual analytics to different real-time applications. According to Thomas et al., "visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces" [13]. It is the tight integration of computational analysis by algorithms and highly interactive visualizations. Through user interaction and visual supported parameter refinement, knowledge and insights can be extracted from the underlying data. This knowledge can directly be used to influence the models used for automated processing using a feedback loop. This combines the strength of automated algorithms with human's intuition and background knowledge to provide better results.

The remainder of this paper is organized as follows. In Section 2, we review work from different fields. Section 3 describes criteria for real-time visualizations. Section 4 introduces our overall architecture and describes the developed visualizations. Section 5 provides a real case study, how the system can be applied to system log events. Finally the work is summarized and future research and improvements are discussed in Section 6.

## 2. RELATED WORK

Much research has been conducted in the area of visual representation of time series data which is highly related to

*SAC'12* March 26-30, 2012, Riva del Garda, Italy.

visualizing temporal data streams. While the basic principle of a time series chart was already described by William Playfair in 1786 [10] many improvements and other techniques were developed in the past. An extensible overview of research in the field of visualizing time-related data can be found in [1]. A space efficient technique to represent time-oriented data are pixel visualizations [2]. Those techniques help to identify patterns, exceptions and similarities. The TimeSearcher [6] allows to interactively query such data sets and [5] combines visual interaction with automated analytical methods. Most of these techniques require the full dataset to be available, which is not an option for data streams.

A very advanced system to analyze large-scale time series data streams is LiveRAC [9]. The core part of this system is a "reorderable matrix of charts". To provide a good overview for thousands of devices, the authors use semantic zooming to adapt the charts. The system takes advantage of the SWIFT backend [7], which uses a streaming pipeline model to process time series data. In contrast to our application, LiveRAC focuses on time-series data, which is great to analyze system management time-series (e.g., CPU loads, memory usage). Our system provides a similar real-time solution, but does focus mostly on textual event data instead, which is challenging to aggregate and needs more analysis and classification work to be done. LiveRAC was also used as a basis for other real-time applications to visualize time series to enhance traffic analysis based on network flow data, for example CLIQUE [3]. This system shares the same design principles than LiveRAC, but does rely on a messaging backend structure instead, which is similar to our processing approach. The integration of general aggregation functions for such data sets is often not flexible enough. Smart aggregation [12] tries to solve this problem by combining "automatic data aggregation with user-defined controls". This helps to provide situational awareness on massive data sets. Another tool, which focuses on monitoring of time series data is VizTree [8], which provides visual real-time anomaly detection for time series. The general approach is to transform the time series data to a representation of symbols. Those symbols are visualized in a suffix tree to present frequencies with different colors. A generic and flexible solution to find interesting events and patterns based on similarity ordering, which uses colored rectangles to represent the events, was proposed by Schaefer et al. [11]. This approach to use the similarity of event patterns cannot be applied to real-time data streams, because the full data set is required for calculation.

In contrast to existing work, our system moves visual analytics for generic event data to a real-time level. Additionally, the proposed visualizations allow interactive exploration and monitoring of streaming event data and are able to provide context and historic time series information accordingly.

## 3. VISUALIZATION REQUIREMENTS FOR STREAMING DATA

Chin et al. [4] points out, that visual search tools are effective if they visualize the data in an intuitive context. This is the reason, why temporal data is often mapped to timelines, geographic data to maps and hierarchical data to trees. This helps the user to understand the visualization quickly. On the other hand, if the visualization focuses on other requirements (e.g., using the space efficiently) other techniques might be more useful, which might be less intuitive, but provide a higher information density.

When having real-time visualizations it is particular important to convey the data in an human-recognizable way to reduce the perceptual complexity to support the decision-making process for the dynamic event stream. To design and implement applicable visualizations, we identified four criteria which are helpful to assess and to design appropriate techniques.

**Interactive Exploration** is an important requirement for real-time visual analytics applications. The user needs to interactively explore the event stream anytime. This is a challenging visualization constraint for dynamic data streams, because new data is added to the view even when the user is simultaneously exploring the data. Aggregation of the event data is helpful in the overview, but at least at a particular zooming level the user needs to select individual events to get details on demand, to provide a seamless transition between monitoring and exploration. Semantic zooming can help to switch between those different levels of details smoothly.

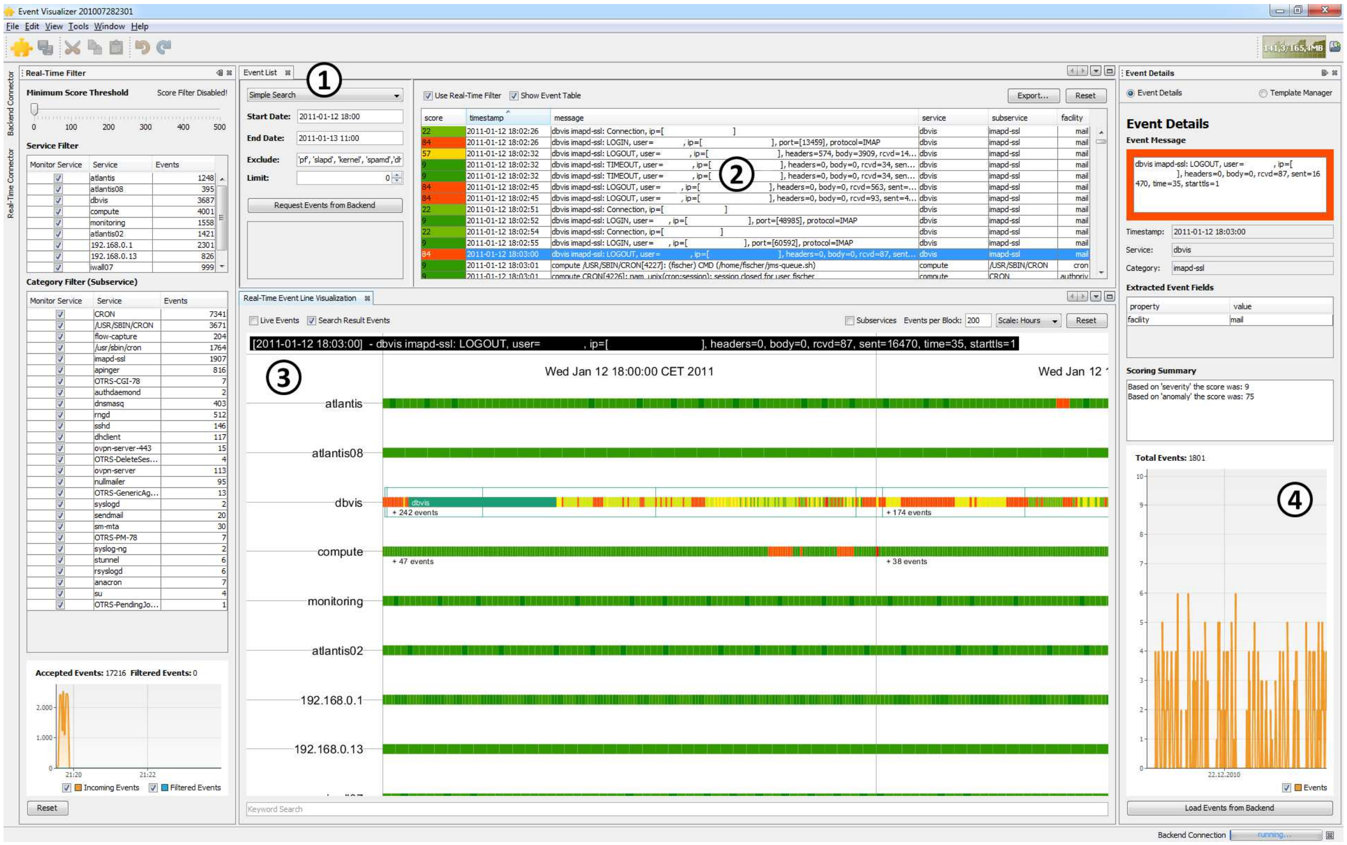
**Updatability in Real-Time** is another challenging aspect. This means that the complexity of the used visualization algorithms needs to be as low as possible. This also causes, that the view must provide ways for incremental appending of new events. To recalculate the whole layout every time a new event comes in is not affordable in general.

**Locality of Changes** means that appending new events doesn't heavily affect the whole visualization. Stability of the resulting view is an important criteria, how the user will perceive the visualization. Many changes will distract the user and will make exploration rather difficult. Additionally, if there are changes to events in the past, the user might easily loose the context and reference points. Therefore, a key principle when designing visualizations for monitoring scenarios, is the idea of keeping changes on already visualized events as small as possible.

**Preservation of Temporal Context** is an important criteria, if a visualization should be able to convey historic and recent data items. To see temporal patterns at least the relative time should be recognizable. Preserving the time information is challenging, because it is a trade-off between giving an overview and still be able to visually link the event's representation with a precise point in time.

## 4. SYSTEM IMPLEMENTATION

When event data streams should be visually analyzed, an infrastructure is needed, which is able to collect, analyze and store the incoming data. To provide visualizations for incoming events in real-time, a forwarding mechanism is required to push events to the visualizations immediately. This points out that the backend is indeed a crucial part of the whole visual analytics system, because the whole backend architecture must consider the specific needs of novel user interfaces. In the scope of visual analytics, it is also required that the user needs to interactively push results back to the automatic analysis process (feedback loop). This can be used, for example, to directly influence event classifica-



Event Visualizer 201007282301

**Minimum Score Threshold** Score Filter Disabled!

**Service Filter**

| Monitor Service | Service | Events |
|---|---|---|
| ☑ | atlantis | 1248 |
| ☑ | atlantis08 | 395 |
| ☑ | dbvis | 3687 |
| ☑ | compute | 4001 |
| ☑ | monitoring | 1558 |
| ☑ | atlantis02 | 1421 |
| ☑ | 192.168.0.1 | 2301 |
| ☑ | 192.168.0.13 | 826 |
| ☑ | iwall07 | 999 |

**Category Filter (Subservice)**

| Monitor Service | Service | Events |
|---|---|---|
| ☑ | CRON | 7341 |
| ☑ | /USR/SBIN/CRON | 3671 |
| ☑ | flow-capture | 204 |
| ☑ | /usr/sbin/cron | 1764 |
| ☑ | imapd-ssl | 1907 |
| ☑ | apinger | 816 |
| ☑ | OTRS-CGI-78 | 7 |
| ☑ | authdaemond | 2 |
| ☑ | dnsmasq | 403 |
| ☑ | rngd | 512 |
| ☑ | sshd | 146 |
| ☑ | dhclient | 117 |
| ☑ | ovpn-server-443 | 15 |
| ☑ | OTRS-DeleteSes... | 4 |
| ☑ | ovpn-server | 113 |
| ☑ | nullmailer | 95 |
| ☑ | OTRS-GenericAg... | 13 |
| ☑ | syslogd | 2 |
| ☑ | sendmail | 20 |
| ☑ | sm-mta | 30 |
| ☑ | OTRS-PM-78 | 7 |
| ☑ | syslog-ng | 2 |
| ☑ | stunnel | 6 |
| ☑ | rsyslogd | 6 |
| ☑ | anacron | 7 |
| ☑ | su | 4 |
| ☑ | OTRS-PendingJo... | 1 |

**Accepted Events:** 17216 **Filtered Events:** 0

Simple Search

**Start Date:** 2011-01-12 18:00

**End Date:** 2011-01-13 11:00

**Exclude:** 'pf', 'slapd', 'kernel', 'spamd','dh

**Limit:** 0

☑ Use Real-Time Filter ☑ Show Event Table

| score | timestamp | message | service | subservice | facility |
|---|---|---|---|---|---|
| 22 | 2011-01-12 18:02:26 | dbvis imapd-ssl: Connection, ip=[ ] | dbvis | imapd-ssl | mail |
| 84 | 2011-01-12 18:02:26 | dbvis imapd-ssl: LOGIN, user= , ip=[ ], port=[13459], protocol=IMAP | dbvis | imapd-ssl | mail |
| 57 | 2011-01-12 18:02:32 | dbvis imapd-ssl: LOGOUT, user= , ip=[ ], headers=574, body=3909, rcvd=14... | dbvis | imapd-ssl | mail |
| 9 | 2011-01-12 18:02:32 | dbvis imapd-ssl: TIMEOUT, user= , ip=[ ], headers=0, body=0, rcvd=34, sen... | dbvis | imapd-ssl | mail |
| 9 | 2011-01-12 18:02:32 | dbvis imapd-ssl: TIMEOUT, user= , ip=[ ], headers=0, body=0, rcvd=34, sen... | dbvis | imapd-ssl | mail |
| 84 | 2011-01-12 18:02:45 | dbvis imapd-ssl: LOGOUT, user= , ip=[ ], headers=0, body=0, rcvd=563, sent=... | dbvis | imapd-ssl | mail |
| 84 | 2011-01-12 18:02:45 | dbvis imapd-ssl: LOGOUT, user= , ip=[ ], headers=0, body=0, rcvd=93, sent=4... | dbvis | imapd-ssl | mail |
| 22 | 2011-01-12 18:02:51 | dbvis imapd-ssl: Connection, ip=[ ] | dbvis | imapd-ssl | mail |
| 9 | 2011-01-12 18:02:52 | dbvis imapd-ssl: LOGIN, user= , ip=[ ], port=[48985], protocol=IMAP | dbvis | imapd-ssl | mail |
| 22 | 2011-01-12 18:02:54 | dbvis imapd-ssl: Connection, ip=[ ] | dbvis | imapd-ssl | mail |
| 9 | 2011-01-12 18:02:55 | dbvis imapd-ssl: LOGIN, user= , ip=[ ], port=[60592], protocol=IMAP | dbvis | imapd-ssl | mail |
| 84 | 2011-01-12 18:03:00 | dbvis imapd-ssl: LOGOUT, user= , ip=[ ], headers=0, body=0, rcvd=87, sent... | dbvis | imapd-ssl | mail |
| 9 | 2011-01-12 18:03:01 | compute /USR/SBIN/CRON[4227]: (fischer) CMD (/home/fischer/jms-queue.sh) | compute | /USR/SBIN/CRON | cron |
| 9 | 2011-01-12 18:03:01 | compute CRON[4226]: pam_unix(cron:session): session closed for user fischer | compute | CRON | authpriv |

□ Live Events ☑ Search Result Events □ Subservices Events per Block: 200 Scale: Hours

[2011-01-12 18:03:00] - dbvis imapd-ssl: LOGOUT, user= , ip=[ ], headers=0, body=0, rcvd=87, sent=16470, time=35, starttls=1

● Event Details ○ Template Manager

**Event Details**

**Event Message**

dbvis imapd-ssl: LOGOUT, user= , ip=[ ], headers=0, body=0, rcvd=87, sent=16470, time=35, starttls=1

Timestamp: 2011-01-12 18:03:00

Service: dbvis

Category: imapd-ssl

**Extracted Event Fields**

| property | value |
|---|---|
| facility | mail |

**Scoring Summary**

Based on 'severity' the score was: 9
Based on 'anomaly' the score was: 75

**Total Events:** 1801

**Figure 1: System administrators can use the Event Visualizer to visually review the most important system log events of large time windows. With (1) the system can query the backend database. The results (2) are shown as a list and are visualized (3) accordingly. Selection provides further details on demand (4).**

tion in the automatic real-time analysis process. To provide this high flexibility, the system was divided into several modules namely the *Event Service*, *Event Analyzers* and *Event Visualizers*. To support communication of the modules, taking scalability, high-availability, high-performance and distributed nodes into account, the Java Message Service[1] (JMS) was used.

## 4.1 Event Service & Analyzers

The core functionalities of the Event Service are to listen or register on given data streams, preprocess, parse and convert the events to generic event objects. Those events are immediately forwarded to the message broker's incoming queue.

In most cases, there is the need to enrich the events with additional data, which are results from the analysis and classification process. It is obvious that those algorithms might be costly and time-consuming. To still keep up with the stream's flow-rate, we make use of *distributed and multi-threaded* analyzers, to take full advantage of cluster nodes and multi-processor environments. The number of running analyzers is elastic and can be adjusted as needed. Each analyzer subscribes to the message broker's queue. The broker ensures that an incoming event is pushed to exactly one of the connected consumers for further analysis. To provide an implicit load-balancing the events are equally distributed to all connected analyzers. In interactive real-time applications, the user expects to be able to retrieve overall statistics and visualizations, to see historical time plots and trends of any specific event type. In many other visual analytics applications, it is possible to generate those statistics on demand when the user actually requests this information (e.g., by issuing a query to the database system). In an environment of dynamic large-scale streams such queries would take too long and would not satisfy the need of interactive response times. Besides of that, the analysis, classification and scoring process takes the occurrence of events in the past into account to detect anomalies or event bursts. This is done by keeping occurrence statistics for all event types and user-defined event patterns to calculate the weighted average for incoming events. Therefore, we need fast lookup-mechanisms to efficiently retrieve how often an event type has occurred in the recent past. As a reasonable solution, we made massive use of incremental counters.

## 4.2 Event Visualizer

The *Event Visualizer* is the graphical user interface for the framework. The main goal of this part of the system is to make historic and real-time data available to the user using a set of different visualizations which can be applied to the data stream. Having the criteria of Section 3 in mind we propose a generic timeline visualization to provide a time dependent overview of events to monitor multiple simultane-

[1] http://www.oracle.com/technetwork/java/jms/index.html

ous streams or categories. One of the challenges in dynamic streams are highly co-occurring events at the same point in time. Plotting those events to an absolute timeline leads to overplotting which would possibly hide important events. Stacking those events as solution [11] is not useful for real-time monitoring, because rescaling the visualization to have enough space for those events would strongly violate our criteria for *Locality of Changes* and would make *Interactive Exploration* rather difficult. As generic default visualization for our system we implemented a relaxed timeline, to visualize important events in their relative temporal order.

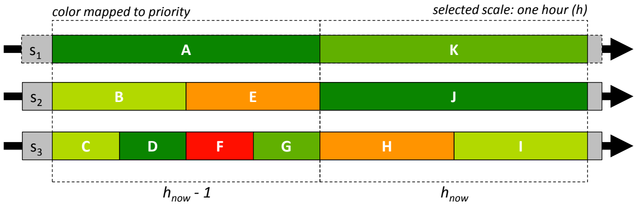


**Figure 2: Multiple Relaxed Timelines for visualizing the events of different streams.**

The basic idea is sketched in Figure 2. Each horizontal bar represents a timeline for a particular stream or category. The scale of the timelines is defined by a global value (e.g. minute, hour or day). A single event is represented by colored rectangles. The area within such an interval is used in a space-filling way for all the events, which occurred in this particular time interval for the respective stream. When there are many co-occurring events in a specific interval, the width for each rectangular shrinks to make space for newly added events. As a result, the absolute time information is lost, but the relative temporal information is still preserved. Additionally, all right-most intervals always convey more recent events than those intervals on the left. Therefore, general comparison between the different streams and the identification of trends are still possible.

To achieve *Interactive Exploration* while new events are added to the visualization, we decided to embed the whole visualization to a zoomable user interface, which allows us to smoothly pan and zoom into any area to investigate the event streams. Hence, switching between historic events and the most recent events for monitoring purposes is possible as smooth transition. Because of the fixed intervals the *Locality of Changes* and *Updatability in Real-Time* are guaranteed. Incrementally adding new items will only slightly influence the neighboring events within the same interval. *Preservation of Temporal Information* is limited to a relative preservation of time within an interval.

This attempt also has several limitations, we would like to discuss. A major drawback becomes obvious, when there are plenty of events in a single interval. At one point the rectangles are getting too small to be visible. Using semantic zoom this drawback can be qualified, because a deeper zoom level would provide more space again. Automatic scoring and classification algorithms, which are applied to all incoming events help to rank and filter for highly relevant events. As a result we introduce a maximum value of number of events per interval. After reaching this limit, the visualization will remove the least interesting events in favor of a more interesting ones. On high-frequency peaks, this helps the analyst to focus at least on the most important events. This prevents an information overload in such situations by reducing the high number of very common (which are most likely less interesting) events from the monitoring view.

Besides of the mentioned interaction techniques the user needs to be able to select and highlight events. To highlight selected events, we change the color according to a qualitative color map. To see occurrence patterns of similar events (or events with the same type of event) in all streams, symmetric lines (boxes) are introduced to connect all related events within each timeline.

Because of the variety of different tasks, we need the possibility to focus on different aspects of the event data to emphasize a particular event attribute. To address geographic related tasks we integrated a real-time geographic map. In this visualization each event is mapped to a circle, which can be seen in Figure 4. To provide a smooth transition between historic and the real-time incoming events, we integrated a time slider. Besides of that, a traditional textual representation and a tag cloud visualization available to spot the most common keywords in the current stream, have been integrated.

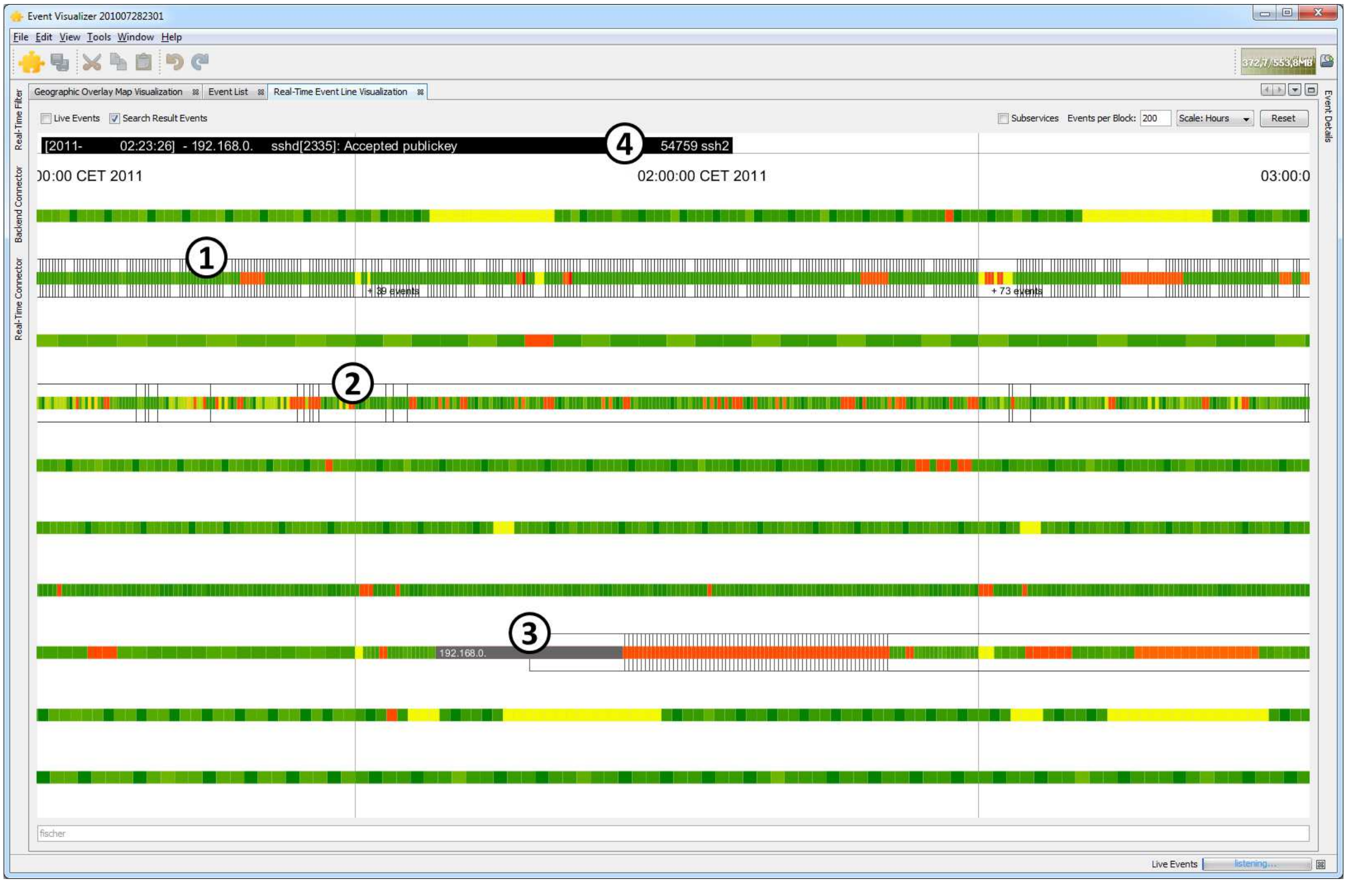
## 5. CASE STUDY: MONITORING OF SYSTEM LOG EVENT STREAMS

The system was deployed for the system administrators (SA) in our working group to visually support and improve their daily tasks to monitor their stream of system log events. The system was operating without major problems over the last months and has successfully processed over 100 million events. The Event Analyzers were deployed on a multi-core server utilizing ten threats in total. In this scenario, there were peaks up to 425,000 events/hour. The SA arrives in the morning in his office and opens the *Event Visualizer* (Figure 1) and connects to the backend database[2]. After loading the events (1) of the last night to the system, the different visualizations will be automatically enriched with the relevant events. Additionally the SA starts the real-time connector, which directly connects the visualizations to the live stream. New events will then continuously be added to the loaded visualizations. The time-consuming work of scoring and classifying the events has already been done by the distributed Event Analyzer modules. Those modules calculate an anomaly score based on the uniqueness and frequency of the event's message, based on the event type and based on user-defined rules which can be generated from within the Event Visualizer's user interface.

This overall scoring value is mapped to a color value according to user-defined color maps. By default, reaching from green (lowest score), over yellow to red (high interestingness). To ensure meaningful representations for colorblind users, there are several other optimized color maps integrated.

Using the *Relaxed Event Timelines* the SA is able to investigate the occurred events of many servers (3). By default, each timeline represents the event stream of a particular server. When there are hundreds of different servers it might be more appropriate to switch this mapping to categories (e.g., process names) instead. This would for example mean that all mail related events originating from different servers, would be visualized together in a single timeline.

[2] Only possible if the stream *can* be stored effectively in a distributed database system.



[2011- 02:23:26] - 192.168.0. sshd[2335]: Accepted publickey 54759 ssh2

**Figure 3: The Relaxed Event Timeline can be used to visually highlight search results (1), which can help to track suspicious user behavior (2) or to notice abnormal patterns (3). Hovering over events provides tooltip information (4).**

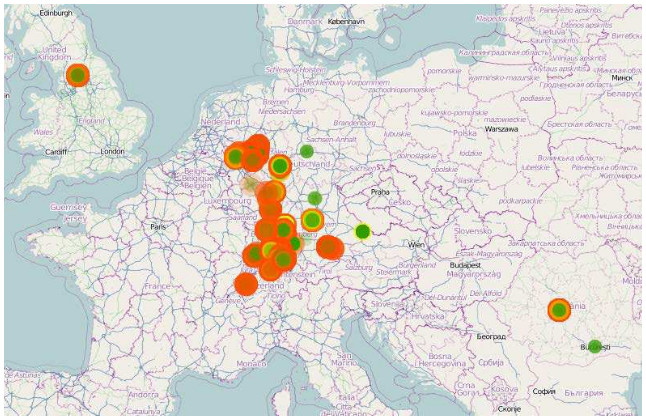
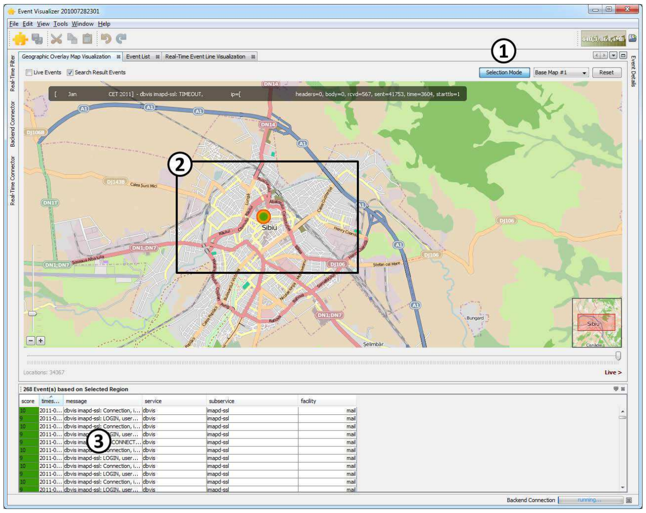


**Figure 4: Visualization of IMAP login events according to their geographic locations based on IP addresses.**

Using the real-time filter, it is possible to filter the incoming stream according to different attributes or discard events with a very low scoring value. The *Event Datails View* (4) provides more information and an historic graph about the occurrences of the classified event type in the past. To enhance such explorations, the the tool can be used to do a visual search. This helps to identify insider threats. At the bottom of the visualization showed in Figure 3 the user can type in ad-hoc search terms. The system will continuously highlight all events matching the search query using lines in the data stream. This quick search for an user name reveals interesting usage patterns which can be directly explored without switching the context. In the second stream, there are plenty of events (1) which are generated by that particular user. Selecting those events provides details on demand, with a graph to show occurrences of this event in the past. With the help of this additional information the SA can judge the relevance of these events immediately. In (2), the events of an e-mail server are visualized. These can be identified as requests from the user's mobile phone during the night at around 2:00 am. In (3) the user has selected a particular event. The corresponding tooltip is shown at (4). This event has a relatively high scoring value (orange color) and is executed many times in this time interval. The contents of the message (4) gives a hint that those are successful public key authentication logins to a server, which seems to be suspicious massive file transfers late at night. The SA is now able to save the knowledge about this pattern directly to the analysis system, by simply comment and apply a scoring modifier to the selected event, which will influence the classification and helps other SA to make use of his insights.

During analysis the geographic coordinates had been added to all system log messages based on contained IP addresses.



```
[     Jan              CET 2011] - dbvis imapd-ssl: TIMEOUT,          ip=[                    headers=0, body=0, rcvd=567, sent=41753, time=3604, starttls=1
```

268 Event(s) based on Selected Region

| score | times... | message | service | subservice | facility |
|---|---|---|---|---|---|
| 10 | 2011-0... | dbvis imapd-ssl: Connection, i... | dbvis | imapd-ssl | mail |
| 9 | 2011-0... | dbvis imapd-ssl: LOGIN, user... | dbvis | imapd-ssl | mail |
| 10 | 2011-0... | dbvis imapd-ssl: Connection, i... | dbvis | imapd-ssl | mail |
| 9 | 2011-0... | dbvis imap GIN, user... | dbvis | imapd-ssl | mail |
| 9 | 2011-0... | dbvis ima CONNECT... | dbvis | imapd-ssl | mail |
| 10 | 2011-0... | dbvis imap onnection, i... | dbvis | imapd-ssl | mail |
| 9 | 2011-0... | dbvis imapd-ssl: LOGIN, user... | dbvis | imapd-ssl | mail |
| 10 | 2011-0... | dbvis imapd-ssl: Connection, i... | dbvis | imapd-ssl | mail |
| 9 | 2011-0... | dbvis imapd-ssl: LOGIN, user... | dbvis | imapd-ssl | mail |
| 10 | 2011-0... | dbvis imapd-ssl: Connection, i... | dbvis | imapd-ssl | mail |
| 9 | 2011-0... | dbvis imapd-ssl: LOGIN, user... | dbvis | imapd-ssl | mail |

**Figure 5: Visual analysis of suspicious login attempts from unusual countries. Switching to selection mode (1), provides the possibility to select a region (2) to retrieve the underlying raw events (3).**

This helps to investigate suspicious events from unusual locations. The access to a valid IMAP mailbox from an unusual geographic location could be a first hint for an successful credential theft and misuse. Figure 4 shows a live view of about 34,000 events. Most of the events cluster on different locations within Germany. However, some outliers originating from other countries are more interesting and need further exploration. Using panning and zooming the user can interactively explore the map, which is shown in Figure 5. After switching to the selection mode (1), rectangular selection (2) can be used to get and interpret the underlying events (3).

## 6. CONCLUSIONS AND FUTURE WORK

In this work we presented a loosely coupled modular visual analytics system for collecting, processing, analyzing and visualizing dynamic real-time event data streams. To achieve this, we designed 1) a generic processing and analysis architecture for event data using a distributed messaging infrastructure. Additionally, we proposed 2) a pluggable visualization application for event data and 3) the Relaxed Event Timeline using scaled intervals to analyze highly cooccurring events. The advantage of this visualization is that it can be used to smoothly switch between historic events and the most recent events for monitoring purposes and relate incoming data with historic events. Furthermore, criteria and requirements were defined, which can help do develop or adjust more visualization techniques appropriate for data streams.

In the future we would like to conduct a controlled user study to evaluate the system, especially for the day by day usage. Integrating more sophisticated algorithms for burst and anomaly detection is intended. In addition, we want to focus on developing and integrating other novel visualizations for real-time event data according to the given design principles learned from this work. The strengths of this visual analytics approach can be found in the combination of automatic algorithms to classify and score anomalous behavior and visual exploration.

## 7. REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data.* Human-Computer Interaction. Springer Verlag, 1st edition, 2011.

[2] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of Visualization '95, Atlanta, GA*, pages 279–286, 1995.

[3] D. Best, S. Bohn, D. Love, A. Wynne, and W. Pike. Real-time visualization of network behaviors for situational awareness. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security*, pages 79–90. ACM, 2010.

[4] G. Chin, M. Singhal, G. Nakamura, V. Gurumoorthi, and N. Freeman-Cadoret. Visual analysis of dynamic data streams. *Information Visualization*, 8(3):212–229, 2009.

[5] M. C. Hao, U. Dayal, D. A. Keim, D. Morent, and J. Schneidewind. Intelligent visual analytics queries. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 91–98, 2007.

[6] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[7] E. Koutsofios, S. North, R. Truscott, and D. Keim. Visualizing large-scale telecommunication networks and services. *Proceedings Visualization '99 (Cat. No.99CB37067)*, pages 457–461, 2008.

[8] J. Lin, E. Keogh, S. Lonardi, J. Lankford, and D. Nystrom. VizTree: a tool for visually mining and monitoring massive time series databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1269–1272. VLDB Endowment, 2004.

[9] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: interactive visual exploration of system management time-series data. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1483–1492. ACM, 2008.

[10] W. Playfair and J. Corry. *The commercial and political atlas.* printed for J. Debrett; GG and J. Robinson; J. Sewell; the engraver, SJ Neele; W. Creech and C. Elliot, Edinburgh; and L. White, Dublin, 1786.

[11] M. Schaefer, F. Wanner, F. Mansmann, C. Scheible, V. Stennett, A. T. Hasselrot, and D. A. Keim. Visual Pattern Discovery in Timed Event Data. In *Proceedings of Conference on Visualization and Data Analysis*, 2011.

[12] D. R. Tesone and J. R. Goodall. Balancing interactive data management of massive data with situational awareness through smart aggregation. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 67–75, 2007.

[13] J. Thomas and K. Cook. *Illuminating the path: The research and development agenda for visual analytics.* IEEE Computer Society, 2005.