Erschienen in: Expanding the Frontiers of Visual Analytics and Visualization / ed. by John Dill ... - London : Springer, 2012. - S. 69-80. - ISBN 978-1-4471-2803-8

Dynamic Visual Analytics—Facing the Real-Time Challenge

Florian Mansmann, Fabian Fischer, and Daniel A. Keim

Abstract Modern communication infrastructures enable more and more information to be available in real-time. While this has proven to be useful for very targeted pieces of information, the human capability to process larger quantities of mostly textual information is definitely limited. Dynamic visual analytics has the potential to circumvent this real-time information overload by combining incremental analysis algorithms and visualizations to facilitate data stream analysis and provide situational awareness. In this book chapter we will thus define dynamic visual analytics, discuss its key requirements and present a pipeline focusing on the integration of human analysts in real-time applications. To validate this pipeline, we will demonstrate its applicability in a real-time monitoring scenario of server logs.

5.1 Introduction

Real-time analysis is a challenging and important field, motivated through both the potential to gain value from up-to-date information as demonstrated in the financial sector, and the thread of damage to be caused if timely assessments are unavailable as, for example, in emergency situations, air traffic control or network monitoring scenarios. In many such cases, it has been shown that automated solutions alone are insufficient since the severity of consequences often requires humans to be in the loop for making adequate decisions. Therefore, applying visual analytics for real-time analysis appears to be a rewarding undertaking.

Since most analysis and visualization methods focus on static data sets, adding a dynamic component to the data source results in major challenges for both the automated and visual analysis methods. Besides typical technical challenges such as

F. Mansmann (✉) · F. Fischer · D.A. Keim
University of Konstanz, Konstanz, Germany
e-mail: Florian.Mansmann@uni-konstanz.de

F. Fischer
e-mail: Fabian.Fischer@uni-konstanz.de

D.A. Keim
e-mail: Daniel.Keim@uni-konstanz.de
url: http://infovis.uni-konstanz.de

unpredictable data volumes, unexpected data features and unforeseen extreme values, a major challenge is the capability of analysis methods to work incrementally. In the essence this means that current results need to build upon previous results, which applies both to knowledge discovery methods such as clustering as well as visualization methods. Adding a node and several edges in a force-directed graph visualization, for example, will result in significant visual changes, whereas adding a data point to a scatter plot only triggers one small visual change. Therefore, certain methods are more suitable than others to be used in an incremental fashion.

The field of visual analytics has developed significantly over the last years with more and more methods solving challenging application problems. While Thomas and Cook (2005) defined the purpose of visual analytics tools to "provide timely, defensible, and understandable assessments", to date little research has focused on the timely aspects of visual analytics, and—in particular—its applicability for real-time analysis. We therefore hope that this book chapter inspires and triggers developments in this area.

Real-time analysis task can be as diverse as monitoring (i.e., timely detection of changes or trends), getting an overview, retrieval of past data items, prediction of a future status or making decisions. Despite this diversity of tasks, common to most real-time analysis tasks is the need to provide situational awareness. Ensley describes what he calls "situation awareness" as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Endsley 1995). In particular, this definition includes Ensley's three levels of situation awareness, which are 1) the perception of elements in current situation, 2) the comprehension of the current situation, and 3) the projection of the future status.

Based on these two ground-breaking works we define dynamic visual analytics as follows:

Definition 5.1 Dynamic Visual Analytics is the process of integrating knowledge discovery and interactive visual interfaces to facilitate data stream analysis and provide situational awareness in real-time.

The goal of dynamic visual analytics is thus to provide situational awareness by minimizing the time between happening and assessment of critical events and enabling a prediction of the future status.

Besides defining dynamic visual analytics, the core contribution of this book chapter is to develop an abstract model for dynamic visual analytics, which will then be applied to the application scenario of server monitoring. The remainder of this paper is structured as follows: We will discuss previous work in relation to dynamic visual analytics in Sect. 5.2, thoroughly analyze the user's role and key requirements for dynamic visual analytics methods in Sect. 5.3 and apply our model for dynamic visual analytics to a system log analysis scenario in Sect. 5.4. The last section summarizes our contributions.

5.2 Background

5.2.1 Visual Analytics

In 2005 Thomas and Cook from the National Visualization and Analytics Center coined the term *visual analytics* in their research and development agenda (Thomas and Cook 2005). This document explicitly stated “the need for analysis techniques for streaming data” (recommendation 3.13, p. 91) to deal with the problems of situational awareness, change assessment and information fusion.

Likewise, the recently published book *Mastering the Information Age: Solving Problems with Visual Analytics* (Keim et al. 2010) identifies data streaming as one of the major challenges for visual analytics: “Streaming data presents many challenges—coping with very large amounts of data arriving in bursts or continuously (. . .), tackling the difficulties of indexing and aggregation in real-time, identifying trends and detecting unexpected behavior when the dataset is changing dynamically.”

In contrast to these two fundamental theoretical publications, we exclusively focus on the real-time aspects of dynamic visual analytics for data streams in this book chapter by defining dynamic visual analytics, listing technical requirements and describing the user’s role in this interactive real-time analysis process.

5.2.2 Data Streams: Management and Automated Analysis

From an infrastructural point of view data in streaming applications are difficult to manage in comparison to occasionally updated data sets. As stated in Chaudhry et al. (2005), a number of key characteristics change when moving from traditional database management systems (DBMS) towards streaming applications. First, one-time queries are replaced by continuous queries that not only once evaluate the query against the database, but also stay active and continuously produce results. Second, traditional DBMSs do not necessarily possess a notion of time since an update of an attribute overwrites the previous value, whereas streaming applications might issue continuous queries such as the average speed of cars over the last 5 minutes. Third, the unbounded nature of streams can lead to changes during query execution against such an unbounded data set. Fourth, while data reliability is assumed for a traditional DBMS, failures or delays of sensors might introduce unreliability into streaming applications since records might be reported out-of-order. Lastly, the monitoring nature of streams requires streaming application to have a reactive capability whereas traditional DBMS only passively react to human-issued queries.

A recent review of currently methods and systems can be found in the book *Data Stream Management* by Golab and Özsu (2010). Besides data stream management systems, storing and accessing both historic and real-time data in streaming data warehouses are the core topics of their book.

Not only has data management of streams become an active area of research, but deriving valuable information from data streams through knowledge discovery has also emerged. Besides applying traditional data mining techniques such as clustering, classification, frequent pattern mining and forecasting to data streams, stream-specific topics such as sliding-window computations, stream indexing and querying are emerging and discussed in detail in Aggarwal (2007). In many cases, the unbounded and high-frequency characteristics of data streams pose additional challenges for the analysis from an algorithmic and computational point of view: Evolving data over time might require changes in the derived models. As a result of high data volumes, it might no longer be possible to process the data efficiently more than once. In addition to this book, Gama's comprehensive book (Gama 2010) focuses on a number of stream mining solutions based on *adaptive learning algorithms*. Thereby the set of examples is not only incremented for a given learning algorithm, but also outdated examples are forgotten. Gama furthermore argues that machine learning algorithms have to work with limited rationality, which he describes as the fact that rational decisions might not be feasible due to the finite computational resources available for making them.

Dynamic visual analytics needs to build upon the emerging infrastructure technology to query and store data streams. Furthermore, knowledge discovery from data streams can also be considered a relatively young research field. As a result of the technology's immaturity, intensive training to application-specific solutions and continuous adaption to still changing concepts are currently characteristic to data streaming since off-the-shelve solutions are for most application fields not yet available.

5.2.3 Time Series Visualization

So far, the most common field to include real-time visualization was *time series visualization* as surveyed in the recently published book *Visualization of Time-oriented Data* (Aigner et al. 2011). While there is a large body of work about interactive visualization and exploration of static time series (e.g., the TimeSearcher, Hochheiser and Shneiderman 2004), further research aspects in this field include the combination of knowledge discovery and visualization techniques as, for example, demonstrated in the classical paper *Cluster and calendar based visualization of time series data* by Van Wijk and Van Selow (1999). Yet another interesting work is VizTree (Lin et al. 2004), a system that can be used to mine and monitor frequently occurring patterns in time series through a data abstraction and pattern frequency tree visualization.

Some more recent publications explicitly focus on the real-time analysis and visualization aspects of time series: The work of Kasetty et al. (2008) focuses on real-time classification of streaming sensor data. The authors use Symbolic Aggregate Approximation (SAX) (Lin et al. 2003) to transform the time series data and then show how time series bitmaps representing pattern frequencies can be updated in constant time for classifying high-rate data streams. Hao et al. (2008) investigate the use of cell-based time series for monitoring data streams to reveal root causes of anomalies. In this work they focus on circular overlay displays, which start overwriting the screen once full, and variable resolution density displays, which adaptively increase the resolution of the display once more data is available. The LiveRAC system (McLachlan et al. 2008) follows a reorderable matrix of charts approach with semantic zoom to monitor a large number of network devices in real-time.

In contrast to these dynamic time series visualizations, a number of domain-specific real-time visualization systems exist. The VisAlert system (Foresti et al. 2006), for example, aims at establishing situational awareness in the field of network security. A circular layout is used to visualize the relationships between where (w.r.t. the network infrastructure), what (type of alert) and when network security alerts occurred. While real-time aspects were not explicitly considered in this work, the system of Best et al. (2010) puts its focus there. Using a high-throughput processing platform, the authors base their network monitoring work on the above mentioned SAX technique to model behavior of actors in the network in real-time and visualize these through a glyph representation. In addition to that, they use LiveRAC and a spiral pixel visualization to interactively investigate streamed network traffic. Other domain-specific work focuses on near real-time visualization of online news (Krstajic et al. 2010). The technique merges several articles into threads and uses categories to display the threads in a temporal colored line visualization.

As shown in this review of related work, combining knowledge discovery and visualization methods for real-time analysis remains challenging and therefore only little work about dynamic visual analytics exists to date. The purpose of this publication is thus to establish a link between the worlds of data stream management, real-time knowledge discovery and interactive dynamic visualization to tackle extremely challenging data analysis problems in streaming applications in the near future.

5.3 Dynamic Visual Analytics

Dynamic visual analytics applications are different from traditional visual analytics systems because a number of requirements from the perspective of data management, knowledge discovery and visualization need to adhere to the incremental nature of streams. Furthermore, the user's role changes since his focus on exploration is extended to include real-time monitoring tasks.

5.3.1 Requirements for Dynamic Visual Analytics Methods

When dealing with massive data streams requirements especially from the data management perspective need to be fulfilled before knowledge discovery or interactive

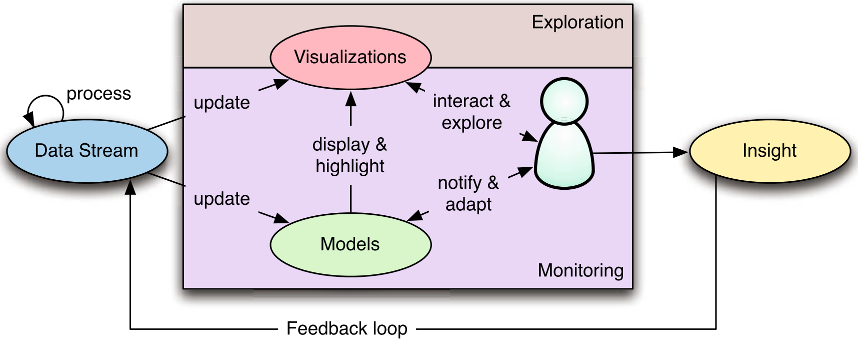


Fig. 5.1 The dynamic visual analytics pipeline

visualization methods can be applied. In particular, this means that (distributed) data gathering & processing in real-time, stream query languages, methods to deal with uncertainty (e.g. error bounds in sensor networks) and reactive capabilities of the database management system are readily available. Furthermore, many applications might not only require querying the stream itself, but also need to access to historic records stored in streaming data warehouses.

Knowledge discovery methods then need to deal with the output of the processed data streams. One key requirement thereby are incremental algorithms that can deal with limited computational resources on the one side and the unbound and possible uncertain nature of the streams on the other side. Again, depending on the application, both the notion of uncertainty as well as the one of real-time might be considerably different.

The set of suitable visualization methods is also dramatically reduced for streaming applications since not all visualizations are designed in a way to accommodate for changes. By nature, time plays an important role in dynamic visualizations and should thus be treated in a way that the age of both historic and recent data items on the screen can be easily distinguished. One prominent way of realizing this is to use the x-position for time, either by shifting historic items to one side in order to make space for new arriving data, by removing and aggregating old data, or by rotating and overwriting old data. However, many more options exist and should be investigated with respect to the specifics of the stream application. In addition to time, visualizations for dynamic visual analytics should be capable of enhancing currently displayed items based on the continuously arriving results of employed online knowledge discovery methods. This could, for example, be done by using a color scales to express the abnormality of an event in a system monitoring scenario. Note that the unbounded nature of streams could lead to the need to make changes to data elements which are already displayed, for example, when a cluster emerges in the stream that extends over both recent and historic items.

After considering these data management, knowledge discovery and visualization requirements for dynamic visual analytics, we will now discuss what role the user plays in interactive streaming applications.

5.3.2 The Role of the User in Dynamic Visual Analytics

Besides different methods, the core difference between automated streaming solutions and dynamic visual analytics is the role of the user. In the essence, background knowledge and decision making capabilities of humans are key to maintain situational awareness by monitoring and exploring data streams. Therefore, we adapted the visual analytics process model of Keim et al. (2008) to match the interactive analysis of data streams.

Figure 5.1 shows the *dynamic visual analytics pipeline*. The nature of data streams differs from static data sets in so far that dynamic updates in the subsequent models and visualizations will be automatically triggered. Likewise, traditional preprocessing will not only be conducted once, but will also be applied to new incoming data on the fly. Visualization or modeling of raw data is an extremely rare and in many application areas unrealistic. Transformation of this raw data into meaningful aggregates or assignment of relevance scores are thus usually the first steps of the analysis. Note that we discard the details of stream data management in our model due to the fact that we focus on the interactive component of streaming applications. Therefore, we consider stream data management as a one-time setup with only rare changes through the user. However, such changes are implicitly modeled through the feedback loop.

Classical stream monitoring applications are built in a way that the incoming data is continuously matched with one or several models. The analyst is then notified about unexpected results and will assess how relevant these are. If exploited near real-time, the gained insight is the outcome that will translate into value for his company. As an example, a company could make money using this information through fast trading on the stock exchange. However, fine-tuning these real-time models is difficult because nobody knows what the future will bring in such a dynamic scenario. Therefore, in critical situations the system might either not pass the relevant notifications or it will trigger an abundance of irrelevant notifications to the user.

Dynamic visualizations can be used to give more control over the data stream into the hands of the analyst. These visualization can either be updated through processed results from the data stream itself, or from derived results from the dynamic models that display the essence of what is going on and highlight supposedly important parts of the data. It is then up to the user to interact with and explore the visualizations to verify his hypotheses and to come up with new ones. Note that our dynamic visual analytics pipeline does not contain an explicit arrow from visualizations to models. However, the two double arrows connecting to the user implicitly allow for adaption of the dynamic knowledge discovery models based on the findings in the visualizations.

Probably the largest part in most dynamic visual analytics application scenarios is devoted to the task of monitoring data streams, which includes both methods from knowledge discovery and visualization. Analysts therefore watch trends, wait for certain events to occur or make projections of future states. Exploration on the other side is a mostly visual process that in most cases will work on a static snapshot of the data. Normally, elements that capture the attention of the analyst might be investigated in detail to assess their relevance with respect to the application dependent analysis goals. Visual methods for exploration of dynamically changing scenarios are so far the exception, but might obtain a more important role for specialized purposes in the near future.

Due to the central role of the user in dynamic visual analytics, human factors play an important role. Individual factors, such as the analyst's long term memory capability, the degree of automaticity for the tasks that he routinely performs and his individual information processing capabilities will influence the analysis results. In particular, the degree of situational awareness, decisions and subsequent actions are strongly influenced by these individual factors. Besides having different preconceptions and expectations, the goals and objectives might vary between analysts. In addition, there might be a variance of the attention capabilities among the analysts, which can also influence the final results.

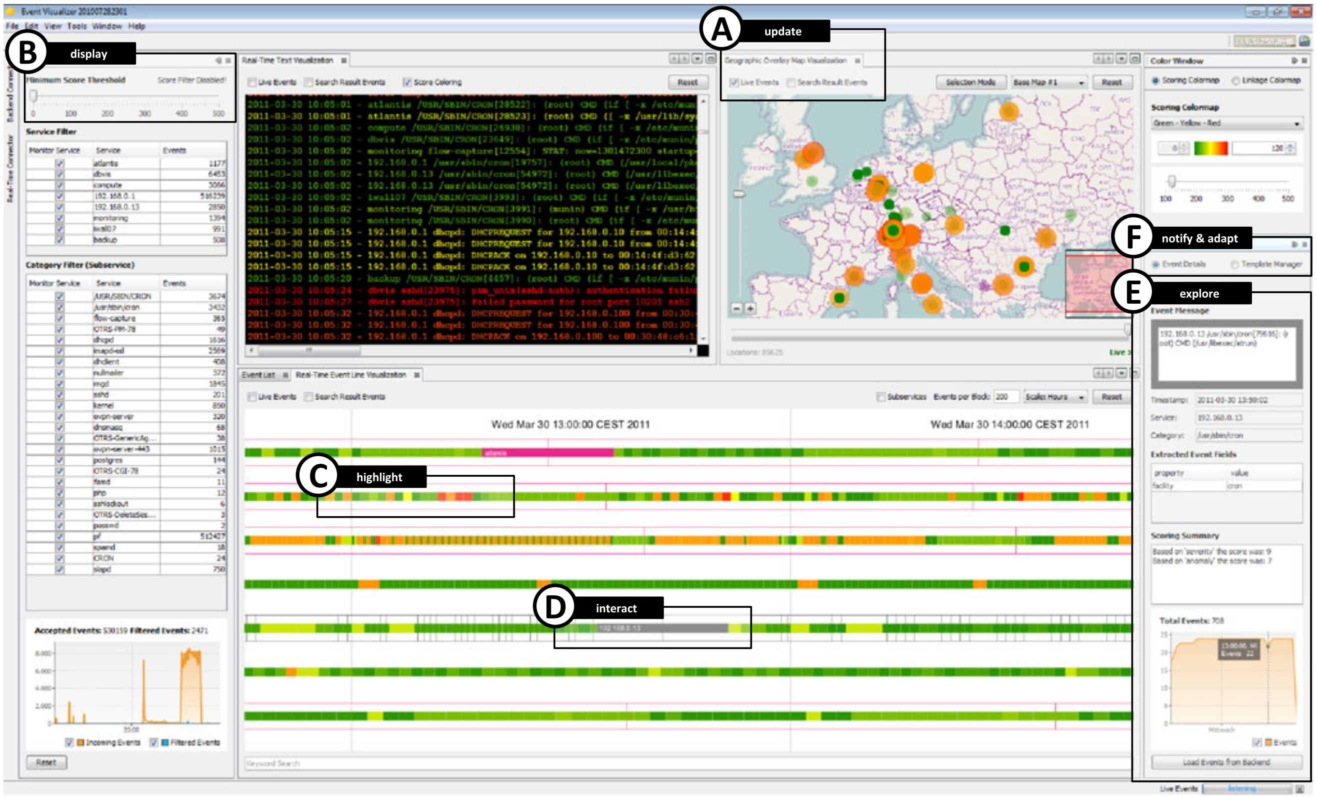
5.4 Server Log Monitoring Application Example

In this section we describe how to visually analyze a large-scale real-time event data stream, which is based on log entries of several servers. In particular, this application example represents a prototypical implementation of the proposed dynamic visual analytics pipeline as discussed above.

System administration and *log analysis* is a typical domain in which experts have to analyze and react to such data streams. Moreover many tasks discussed in the previous sections are relevant in this domain. Large computer networks with all their servers and devices, produce a vast amount of status, error and debug messages or alerts. Not only do the server's operating systems produce this continuous flow of information, but the software, the processes and services running on those machines also generate even more log data (e.g. access log data). The amount of this data is unbounded and is heavily affected by the utilization of the services. A busy web server will produce a higher number of error and status messages than an idle server system. This makes it obvious that the data stream is unpredictable with peaks and bursts, which need to be processed by the analysis system. Fully automated systems, which try to cluster and extract relevant or unusual events out of this data streams, are often not sufficient for further analysis, because the context is not preserved and the analyst is not able to gain an overview of these potentially critical events.

Closely monitoring the events in its context can help to minimize possible downtimes by reacting to and pro-actively solving issues as early as possible. Implementing a dynamic visual analytics approach helps to minimize expenses or prevent compensations. In particular, such an approach provides situational awareness and enhances both monitoring and exploration through visualization techniques.

In the following we explain how the dynamic visual analytics pipeline, sketched in Fig. 5.1, influenced the implementation of the prototype framework. The visual user interface, which is presented to the analyst, is shown in Fig. 5.2.



Minimum Score Threshold   Score Filter Disabled!

0 100 200 300 400 500

□ Live Events □ Search Result Events ☑ Score Coloring

```
2011-03-30 10:05:01 - atlantis /USR/SBIN/CRON[28522]: (root) CMD (if [ -x /etc/mun
2011-03-30 10:05:01 - atlantis /USR/SBIN/CRON[28523]: (root) CMD ([ -x /usr/lib/sy
2011-03-30 10:05:02 - compute /USR/SBIN/CRON[26938]: (root) CMD (if [ -x /etc/munin
2011-03-30 10:05:02 - dbvis /USR/SBIN/CRON[23649]: (root) CMD (if [ -x /etc/munin/
2011-03-30 10:05:02 - monitoring flow-capture[12554]: STAT: now=1301472300 startup
2011-03-30 10:05:02 - 192.168.0.1 /usr/sbin/cron[19757]: (root) CMD (/usr/local/pk
2011-03-30 10:05:02 - 192.168.0.13 /usr/sbin/cron[54972]: (root) CMD (/usr/libexec
2011-03-30 10:05:02 - 192.168.0.13 /usr/sbin/cron[54972]: (root) CMD (/usr/libexec
2011-03-30 10:05:02 - iwall07 /USR/SBIN/CRON[3993]: (root) CMD (if [ -x /etc/munin
2011-03-30 10:05:02 - monitoring /USR/SBIN/CRON[3991]: (munin) CMD (if [ -x /usr/b
2011-03-30 10:05:02 - monitoring /USR/SBIN/CRON[3990]: (root) CMD (if [ -x /etc/mun
2011-03-30 10:05:15 - 192.168.0.1 dhcpd: DHCPREQUEST for 192.168.0.10 from 00:14:4
2011-03-30 10:05:15 - 192.168.0.1 dhcpd: DHCPREQUEST for 192.168.0.10 from 00:14:4
2011-03-30 10:05:15 - 192.168.0.1 dhcpd: DHCPACK on 192.168.0.10 to 00:14:4f:d3:62
2011-03-30 10:05:15 - 192.168.0.1 dhcpd: DHCPACK on 192.168.0.10 to 00:14:4f:d3:62
2011-03-30 10:05:20 - backup /USR/SBIN/CRON[4457]: (root) CMD (if [ -x /etc/munin/
2011-03-30 10:05:24 - dbvis sshd[23975]: pam_unix(sshd:auth): authentication failu
2011-03-30 10:05:27 - dbvis sshd[23975]: Failed password for root port 10201 ssh2
2011-03-30 10:05:32 - 192.168.0.1 dhcpd: DHCPREQUEST for 192.168.0.100 from 00:30:
2011-03-30 10:05:32 - 192.168.0.1 dhcpd: DHCPREQUEST for 192.168.0.100 from 00:30:
2011-03-30 10:05:32 - 192.168.0.1 dhcpd: DHCPACK on 192.168.0.100 to 00:30:48:c6:1
```

Service Filter

Monitor Service	Service	Events
☑	atlantis	1177
☑	dbvis	6453
☑	compute	3066
☑	192.168.0.1	516239
☑	192.168.0.13	2850
☑	monitoring	1394
☑	iwall07	991
☑	backup	508

Category Filter (Subservice)

Monitor Service	Service	Events
☑	/USR/SBIN/CRON	3674
☑	/usr/sbin/cron	3432
☑	flow-capture	365
☑	OTRS-PM-78	49
☑	dhcpd	1636
☑	imapd-ssl	2569
☑	dhclient	408
☑	nullmailer	372
☑	imapd	1845
☑	sshd	201
☑	kernel	850
☑	ovpn-server	320
☑	dnsmasq	68
☑	OTRS-GenericAg...	38
☑	ovpn-server-443	1015
☑	postgres	144
☑	OTRS-CGI-78	24
☑	famd	11
☑	php	12
☑	sshlockout	6
☑	OTRS-DeleteSes...	3
☑	passwd	2
☑	pf	512427
☑	xpand	18
☑	CRON	24
☑	slapd	750

Accepted Events: 530159 **Filtered Events:** 2471

☑ 🟧 Incoming Events ☑ 🟦 Filtered Events

Reset

☑ Live Events □ Search Result Events

● Scoring Colormap ○ Linkage Colormap

Scoring Colormap

Green - Yellow - Red

0 120

100 200 300 400 500

● Event Details ○ Template Manager

Event Message

192.168.0.13 /usr/sbin/cron[79616]: (root) CMD (/usr/libexec/atrun)

Timestamp: 2011-03-30 13:59:02

Service: 192.168.0.13

Category: /usr/sbin/cron

Extracted Event Fields

property	value
facility	cron

Scoring Summary

Based on 'severity' the score was: 9
Based on 'anomaly' the score was: 7

Total Events: 708

☑ 🟧 Events

Fig. 5.2 A modular dynamic visual analytics software to visually monitor and explore event data streams in real-time

Processing: To support the needed functionality a robust distributed backend system, which relies on a central message broker, was developed. This message broker handles the communication between the different parts of the system. A service module collects incoming messages using the so-called Syslog protocol, which is used in UNIX systems for logging to remote systems. The incoming events are *processed* and transformed to generic messages and forwarded to the message broker.

Update Models & Visualizations: Several analyzer modules receive *updates* about the current data stream, which are processed using the appropriate *models* for the specific tasks and scenarios. These analyzers are responsible for classification, scoring and enriching the events with additional information. In the implemented system the analyzed events are also stored to a distributed database system to provide historical analysis of the last 24 hours of the stream. It is furthermore possible to push the stream data directly to the visualization. The user can decide, which visualizations should be updated in real-time using check boxes. These are available in each of the implemented visualization windows, which can be seen in Fig. 5.2 (A).

Display & Highlight: The result of the automatic analysis using the aforementioned models makes it possible to display the analyzed data stream in visualizations. Based on the models and algorithms, each visualization can be filtered to display or highlight only the interesting events according to the proposed process. The user is able to decide which data items and which stream should be displayed. In our application, this can be done using filter and threshold sliders, as seen in Fig. 5.2 (B). Highlighting (C) is done with a color scheme (green for low and red for high priority) for the calculated score or interestingness of the event. This helps the analyst visually identify important events within the temporal context of the data streams. This timeline visualization can be used in real-time monitoring situations. Each row represents log messages of one server and each colored rectangle stands for a single event in the stream. New data events are continuously added to the right of the visualization.

Interact & Explore: The user is then able to *interact* (D) with the data stream. This is done through interaction techniques like selecting, zooming and panning. Details-on-demand (E) support the *exploration* of selected events. Similar events can be visually highlighted as well. This helps visually identify event patterns and bursts of particular log messages. Interaction makes it possible to easily switch between monitoring and exploration tasks by zooming into the rightmost area of the timeline visualization or panning to the left for historical events. Another example, in which switching between monitoring and historical exploration is combined, is the geographic map in Fig. 5.2. The map shows the currently incoming events, but the user can always go back in time using the time slider at the bottom of the visualization.

Notify & Adapt: Switching to the template manager in Fig. 5.2 (F) allows the user to *adapt* the used models and rules according to his specifications. Applying score modifiers for particular event types or defining regular expressions, which are proposed by the system based on the currently selected events, influence the classification process. In a production version of the system, the usage of special alerting templates could be integrated to directly *notify* the user over different reliable communication channels about events, which need immediate response.

Feedback Loop: Adapting and influencing the models and processing algorithms through the user, is one implementation for a feedback loop. Such changes are directly forwarded to the message broker and automatically distributed to collecting services and available analyzers. This makes it possible to push background knowledge or insights formulated as rules, parameters and score modifiers to the algorithmic processes. As a result, gained insights are not lost, but pushed back to the system's backend to improve future classification or to support other analysts by presenting the corresponding annotations in the user interface.

5.5 Conclusions

This book chapter discussed dynamic visual analytics. In particular, we defined it as the process of integrating knowledge discovery and interactive visual interfaces. Its purpose is to facilitate interactive data stream analysis and provide situational awareness in real-time.

The core of this chapter was the discussion of requirements and the user's role in dynamic visual analytics. In contrast to automated streaming solutions, the presented dynamic visual analytics pipeline accentuates that the analyst's background knowledge and intuition is integrated in the stream analysis process through continuously updated visual interfaces and interaction. Furthermore, automated analysis methods could retrieve continuous user feedback through a notification and adaptation loop between the employed models and the user.

As an example of how the model can be applied in practice, we discussed a dynamic visual analytics application for real-time analysis of log entries from multiple servers. Besides depicting dynamic visual analytics in a demonstrative scenario, it made clear how visualization and user interaction could be used to foster insight and provide situational awareness in time-critical situations.

References

Aggarwal, C. C. (2007). *Data streams: models and algorithms*. New York: Springer.

Aigner, W., Miksch, S., Schumann, H., & Tominski, C. (2011). *Visualization of time-oriented data. Human-computer interaction series*. New York: Springer.

Best, D., Bohn, S., Love, D., Wynne, A., & Pike, W. (2010). Real-time visualization of network behaviors for situational awareness. *Proceedings of the seventh international symposium on visualization for cyber security* (pp. 79–90). New York: ACM.

Chaudhry, N., Shaw, K., & Abdelguerfi, M. (2005). *Stream data management* (Vol. 30). Berlin: Springer.

Endsley, M. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32–64.

Foresti, S., Agutter, J., Livnat, Y., Moon, S., & Erbacher, R. (2006). Visual correlation of network alerts. *IEEE Computer Graphics and Applications*, *26*, 48–59.

Gama, J. (2010). *Knowledge discovery from data streams*. *Data mining and knowledge discovery series*. Boca Raton: Chapman & Hall, CRC Press.

Golab, L., & Özsu, M. (2010). *Data stream management*. Morgan & Claypool Publishers.

Hao, M. C., Keim, D. A., Dayal, U., Oelke, D., & Tremblay, C. (2008). Density displays for data stream monitoring. *Computer Graphics Forum*, *27*(3), 895–902.

Hochheiser, H., & Shneiderman, B. (2004). Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, *3*(1), 1.

Kasetty, S., Stafford, C., Walker, G., Wang, X., & Keogh, E. (2008). Real-time classification of streaming sensor data. In *Tools with artificial intelligence, 2008. ICTAI'08. 20th IEEE international conference* (Vol. 1, pp. 149–156). IEEE.

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., & Ziegler, H. (2008). Visual analytics: scope and challenges. In S. Simoff, M. H. Boehlen & A. Mazeika (Eds.), *Lecture notes in computer science (LNCS)*. *Visual data mining: theory, techniques and tools for visual analytics*. Berlin: Springer.

Keim, D. A., Kohlhammer, J., Ellis, G., & Mansmann, F. (Eds.) (2010). *Mastering the information age—solving problems with visual analytics*. Eurographics.

Krstajic, M., Bertini, E., Mansmann, F., & Keim, D. A. (2010). Visual analysis of news streams with article threads. In *StreamKDD '10: proceedings of the first international workshop on novel data stream pattern mining techniques* (pp. 39–46). New York: ACM.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery* (p. 11). New York: ACM.

Lin, J., Keogh, E., Lonardi, S., Lankford, J., & Nystrom, D. (2004). VizTree: a tool for visually mining and monitoring massive time series databases. In *Proceedings of the thirtieth international conference on very large data bases* (Vol. 30, pp. 1269–1272). VLDB Endowment.

McLachlan, P., Munzner, T., Koutsofios, E., & North, S. (2008). Liverac: interactive visual exploration of system management time-series data. In *Proceedings of the twenty-sixth annual SIGCHI conference on human factors in computing systems* (pp. 1483–1492). New York: ACM.

Thomas, J., & Cook, K. (2005). *Illuminating the path: the research and development agenda for visual analytics*. Los Alamitos: IEEE Computer Society.

Van Wijk, J., & Van Selow, E. (1999). Cluster and calendar based visualization of time series data. In *Infovis*. Los Alamitos: IEEE Computer Society.