Data Warehousing versus Event-Driven BI: Data Management and Knowledge Discovery in Fraud Analysis

Martin Suntinger, Josef Schiefer, Heinz Roth and Hannes Obweger

***Abstract*—In the growing market of online betting and gambling, fraud has reached a magnitude that cannot be overlooked. For successful fraud detection and prevention, systems strongly depend on a detailed characterization of recurring fraud patterns. This paper compares two distinct technologies to extract this knowledge: data warehouses and event-driven BI tools. Though data warehousing coupled with OLAP analysis is in wide-spread use, several limitations and shortcomings come to light. For discovering fraud patterns, the abstraction of business events into aggregated key figures makes detailed root-cause and cause-chain analyses very difficult. Furthermore, complex data mappings and comprehensive efforts for data integration are required for preparing the data for analytical purposes in the data warehouse. In comparison to data warehouses, event-based systems are easy to integrate and provide the analyst with fine-grained information for sound root-cause analyses. Business processes and behavioural patterns of users can be fully reconstructed and visually analyzed at the level of single events.**

***Index Terms*—Business Process Analysis, Data Warehouse, Event-Based Systems, Fraud Management**

I. Introduction

ONLINE betting and gambling is a popular and growing market. By making sports bets and casino games accessible electronically to a mass of customers, providers unavoidably expose themselves to various forms of fraud. Hacker attacks, money laundering, abuse of insider information in sports bets, abuse of bonus programs with multiple anonymous accounts, betting syndicates, casino game syndicates or automated gaming agents are only a small proportion of the threats providers have to face. What Koehler [6] called "an ever-evolving cat-and-mouse play between players and merchants" has become a highly critical issue today with millions of customers playing online. If a security gap is not filled before the mass of users discovers it, huge losses for the providers are the inevitable consequence. In order to detect and prevent these threats, mainly two system components are required:

- an operational component that detects and prevents known fraud patterns (in real time) and alerts security analysts in case of exceptional situations, and
- an analysis component supporting a discovery-driven identification of new fraud patterns and threats.

M.Suntinger and H. Obweger are with SENACTIVE IT-Dienstleistungs GmbH, Vienna, Austria, e-mail: {msuntinger,hobweger}@senactive.com.

J. Schiefer and H. Roth are with the Institute for Software Technology and Interactive Systems, Vienna University of Technology e-mail: {js,hr}@ifs.tuwien.ac.at.

Manuscript received Oktober 15, 2007.

Most of today's fraud analysis systems cover the first requirement with a rule-based approach [11] for generating alerts and blocking users. The latter requirement is concerned with the assessment of knowledge on how to design these rules. Many of the current systems rely on expert knowledge without providing a systematic approach and tools for generating and continuously updating this knowledge.

One technology that can be utilized to discover this knowledge is data warehousing with OLAP analysis. Via OLAP aggregated information on historic incidents can be navigated and analyzed. Single occurrences are abstracted into manageable key figures.

With the need for zero-latency business decisions, event-based systems are becoming increasingly popular and begin to settle in industrial employments. These systems cover both an operational component to sense and react to business events in real time as well as a range of analysis techniques based on the captured events.

These two approaches differentiate in the perspective onto the data. Data warehouses focus on entities like customers, bets or leagues. Event-driven BI sees the data as sequences of events which influence or are caused by such entities. This paper compares these two approaches and the pros and cons of their perspectives for fraud analysis in online betting. The structure of the paper is as follows: Section II discussed related work. Section III compares typical architectures for fraud management solutions with data warehouses and event-based systems. Section IV discusses data retrieval issues and in Section V, we show an example for the analysis of a typical fraud case. Finally, Section IV concludes our comparison.

II. Contribution and related work

In the domain of fraud detection and prevention one can distinguish between at least three major directions in research efforts: (1) fraud detection techniques, (2) fraud detection systems and architectures with a focus on execution issues, and (3) approaches for exploring and gaining knowledge on fraud patterns.

The first direction, fraud detection techniques, includes approaches to detect fraud in specific domains such as telecommunication or gambling industry. Research works in the first domain are concerned with the question of how to *recognize fraud in general*. For example, Fawcett and Provost [7] proposed techniques for profiling users and assessing fraud from detected, conspicuous changes in a user's behavioural pattern.

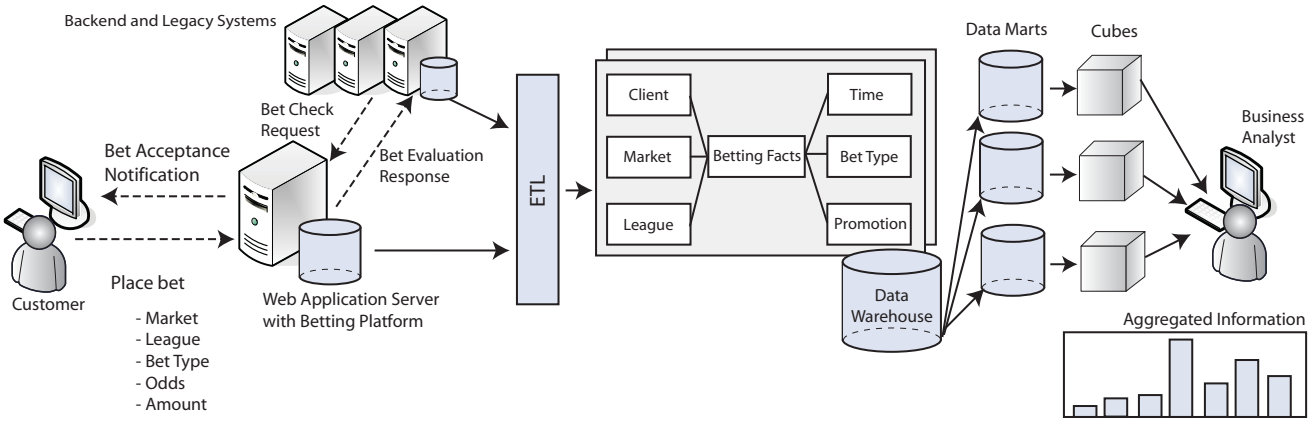


Fig. 1. Data extraction for data warehouse analysis of betting data.

Other examples are the recognition of fraudulent credit card and mobile phone usage based on outlier detection [8] or statistical and data mining algorithms for analyzing skewed data sets, as used by Phua et al. [11].

The second direction, fraud detection systems and architectures, deals with the design and implementation of systems to actively detect and prevent fraud in an operational environment. Knowing fraud cases and patterns, efficient systems are required to *apply* this knowledge. Typically, fraud patterns are comprised in complex rule sets which are continuously evaluated [10]. There has been a lot of research and development concerning knowledge updates and active rules in the area of active databases. Several techniques based on syntactic (e.g. triggering graphs or activation graphs [3]) and semantic analysis of rules [2] have been proposed. The combination of deductive and active rules has been also investigated in different approaches mainly based on the simulation of active rules by means of deductive rules [9].

This leads to the third research direction: Techniques for deriving knowledge on fraud patterns. One possible approach was presented by Rosset et al. [12] who derive rules automatically from the data in a two stages process. First, candidate rules are generated and in a second step, a fraud coverage evaluation filters applicable rules. The advantage of this process is that it can be partially automated. Drawbacks are the missing transparency in the generated rules and the incapability to discover completely unknown fraud patterns. The alternative is to analyze the operational data manually and formulate rules from the discovered patterns. Both data warehouses and event-driven BI tools are applicable. Suntinger et al. [15] presented the Event Tunnel visualization and showed how it can be applied for the analysis of fraud patterns in online betting. The Event Tunnel visually depicts business events in glyphs. Multiple data dimensions can be encoded in colors and sizes of these glyphs. The visual data representation allows to back-trace occurrences and to derive fraud patterns.

III. System architecture and data pre-processing

The range of analysis techniques and the possibility to discover fraud patterns in data sets strongly depends on which data are available for the analysis and how these data are structured. Hence, the comparison starts at the level of data extraction and pre-processing.

A. Data pre-processing for the data warehouse

In online betting and gambling, users communicate via a web server with the betting platform to place bets, play casino games and cash-in or cash-out money. To collect analysis data in a data warehouse, data has to be extracted from the betting platform as well as eventually available legacy systems. Figure 1 illustrates the data transformation process. The data from user- and system communication-messages are mapped into a star schema, as shown in figure 1. This mapping is costly and done for a number of *technical reasons*: The information on one atomic transaction - a bet placement - is spread over multiple database tables. After the pre-processing step, incidents are transformed to key figures in the database. Each entry in the fact table represents exactly one incident.

Advantages are the efficient handling of large data sets and the abstraction to aggregated information, which simplifies analyses. On the other hand, the data integration and the mapping into the DWH schema are costly operations which are usually performed in a batch mode. Approaches for *active data warehouses* have been developed [14] which combine active mechanisms based on event-condition-action rules with the integrated analysis capabilities of data warehouse environments to extend (passive) systems with reactive capabilities. Yet, unresolved performance issues hinder practical implementations of active data warehouses [4] [14].

B. Data management in event-driven BI tools

In an event-based system, information about business activities is encapsulated in events, which capture attributes about the context when the event occurred. Event attributes are items such as the agents, resources, and data associated with an event. For example, a typical bet placement event could have the following attributes as context information: account ID, amount, market, league, sport event ID, odds and bet type. To back-trace and analyze the events of a business process, it is important to correlate temporally and semantically related events. Elements from the context of an event can be used to define its relationship to other events.

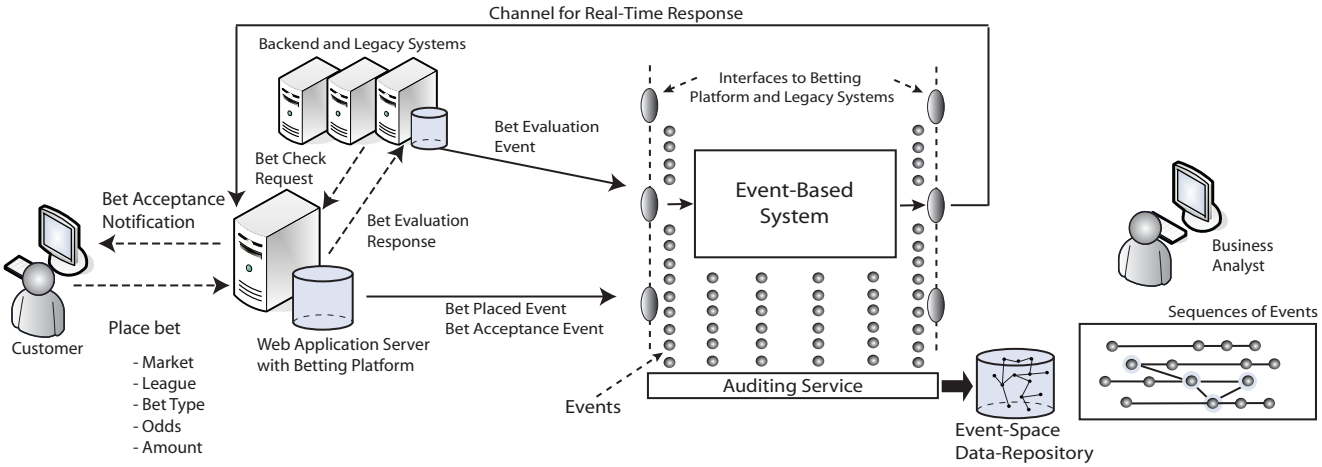


Fig. 2. Data flow and extraction for analysis with an event-based system.

The integration of an event-based system (EBS) strongly differs from the DWH approach. As the event-based system is actively involved in the execution of a process, it integrates directly into the operational environment. Figure 2 illustrates the system architecture and essential data processing flows from the source system to the data analysis. The EBS receives events on notable occurrences such as bet placements via unified interfaces to the betting platform and legacy systems. It can evaluate the received events and, if necessary, respond in real-time by carrying out automated decisions. In comparison to the data warehouse, which periodically takes snapshots for historic data analysis, the EBS integrates into the runtime of the operational systems. Therefore, the pre-processing of events for analysis purposes is performed in real-time. An auditing service pushes meaningful events into a data repository and coherences between events are directly captured. After the pre-processing, any relevant incident is available as a coherent, multi-dimensional data unit (event). In addition, knowledge on sequences of correlated incidents (i.e., the series of bets from one specific user) and process measures are available (e.g., average bet amount for each user, number of bets per user etc.).

The message and service-oriented architecture of event-based systems preserves an event-perspective by capturing and processing incidents as coherent units and preserving these units for the analyst instead of mapping them into a predefined schema. This eases the adaptation of processes by avoiding known schema evolution issues. The data are available for the analyst in near real-time. For fraud analysis this is advantageous since it is possible to immediately back-trace the root-cause in case of a system alert without waiting for the data to be loaded into the DWH. The event-driven approach omits abstraction of the data into entities and stores the data as it occurred in the real world: in the form of events. The advantage is to not lose any information on an occurrence: The behavioural patterns of a user can immediately be reconstructed based on the captured events. One drawback is the huge mass of data: Efficient retrieval mechanisms are required to extract the desired information. Another problem is that aggregated information cannot be extracted efficiently from the data set since this information is encapsulated in hundreds of thousands of events. Instead, such information is pre-calculated in form of measures immediately when capturing the events.

IV. Data retrieval

In the first stage of our comparison, we reviewed required data pre-processing steps and the integration effort in order to collect the analysis data. The second criterion for the comparison is how to query the collected data and extract data sets relevant to an analysis task. Data retrieval is crucial to the analysis process. In order to identify a fraud pattern from a data set, the analyst needs flexible ways to query data sets.

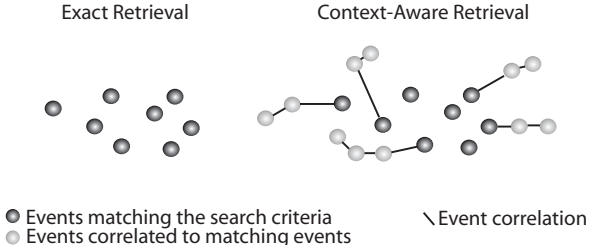


Fig. 3. Exact data retrieval in comparison to context-aware retrieval, returning events correlated to the core search result.

In fraud analysis, experience showed that the analysis process consists of two steps: (1) data tracing and (2) discovery and mining. Data tracing refers to the process of reconstructing an incident from the available data. Tracing is driven by an initial clue (i.e., a system alert that exposes a certain user). The analyst has to extract all data in the context of this clue. The second step is the discovery and mining step. By starting from the available dataset, the analyst will narrow or broaden the analysis horizon to discover similar cases or related incidents.

These retrieval requirements do not exactly match the basic retrieval metaphor of a DWH. In a DWH, retrieval typically starts at a high level of aggregation, whereas here the analyst begins at a detailed level. Instead of extracting key figures, incidents have to be retrieved. Although a DWH

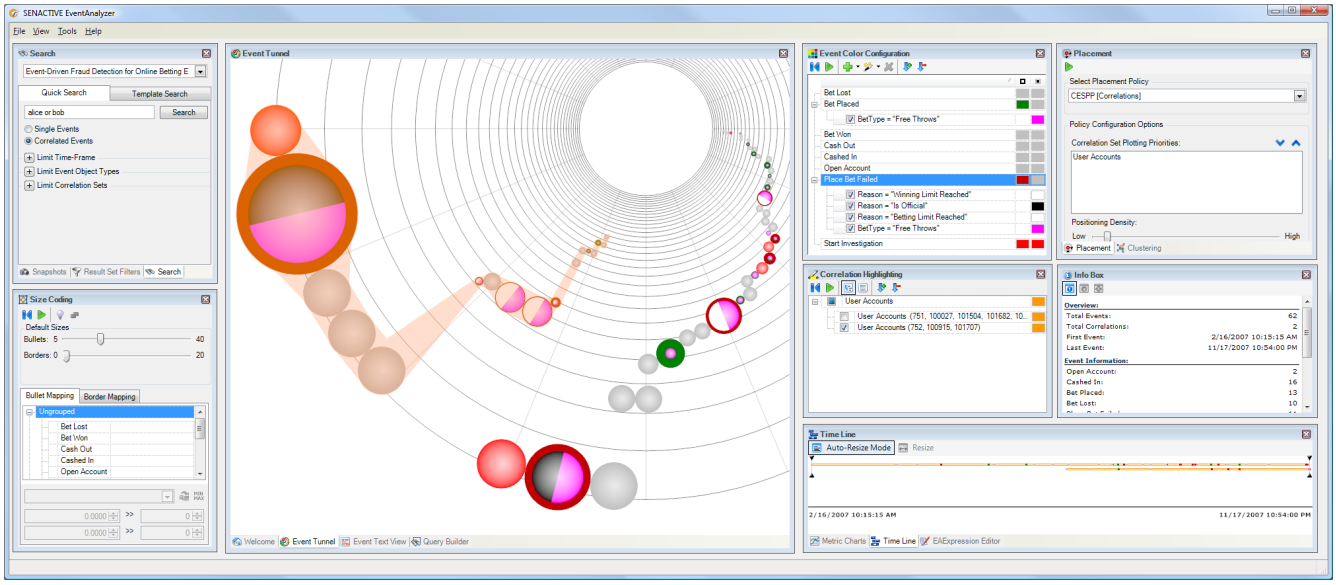


Fig. 4. The analysis workspace of an event-driven BI tool: the SENACTIVE EventAnalyzer. The view shows an event visualization and several configurations options for mapping various data dimensions to colors, sizes and positions in the visualizations.

can be searched for the initial clue by providing a full-text index, only facts which match the search term (e.g. a suspicious user ID) will be returned and provide a starting point for navigating through the data. In contrast, the retrieval metaphor provided by an event-perspective naturally fits the requirements for fraud analysis. Rozsnyai et. al. [13] presented an event space implementation providing full-text search for business incidents. An event-based system correlates events at runtime. Using these correlations, a context-aware depth search is enabled where not only events exactly matching a search query are retrieved, but also events in the context of the base result. Figure 3 illustrates this optional fuzzy retrieval mode.

V. Analysis Opportunities

After having reviewed data pre-processing and retrieval, the third level of the comparison is concerned with analysis opportunities provided by both data warehouses and event-based analysis-systems. We show a real-world use case to outline major characteristics of the analysis requirements in fraud management and assess general capabilities of the compared analysis technologies. As data warehousing and OLAP analysis is wide-spread and well known, we decided to leave out a general introduction to available analysis techniques. Yet, event-driven BI is still in its infancy and not widely known. Therefore, we will shortly introduce one event-driven analysis tool, the SENACTIVE EventAnalyzer.

A. The putter-on use case

Working with a provider of online sports bets revealed that one specific fraud pattern is the so-called *putter-on* pattern. A putter-on is a customer that places a bet for a person who is not allowed to (e.g. referees or players). Detecting and preventing the bets from players and referees can be easily achieved by cross-checking the customer's names in lists of game officials. Even if pseudonyms are used, mismatches in the financial data on cash-out operations unmask most of these customers. Nevertheless, the identification of putter-ons using solely an entity-perspective on the data is much harder. The solution is to identify a behavioural pattern of such customers.

B. Analysis with a DWH

When trying to derive this knowledge from a data warehouse, the first difficulty is the extraction of relevant data. Assuming that knowledge on an official whose bet was prevented is available and there is suspicion that some customer placed the bet for him, with OLAP analysis the relevant data cannot be extracted since there is no direct link between the official's bet and the putter-on bet. In section IV we mentioned the differentiation between exact and fuzzy retrieval. The actions taken by the putter-on are a typical example for context-incidents related to the official whose bet was prevented.

In case a potential putter-on is known, aggregated information can be used to characterize a behavioural profile. For example, irregular bet placement with a skewed distribution of bet amounts and rare but high cash-ins are characteristics of a putter-on. This knowledge is valuable, but it is but one of many indicators required for identifying a putter-on.

C. Analysis with an event-driven BI tool

For our comparison, we utilized an event-driven BI tool which we developed recently: the SENACTIVE EventAnalyzer. It offers a range of visualization opportunities tailored to the characteristics of event data. The central component of the analysis workspace is the so-called *Event Tunnel*, an interactive view into a stream of events [15]. Figure 4 provides an impression of the tool: With the Event Tunnel it is possible to look into the past stream of events. Multiple linked views ease the analysis process. Via a set of configuration panels, the analyst can map data dimensions to colors, sizes

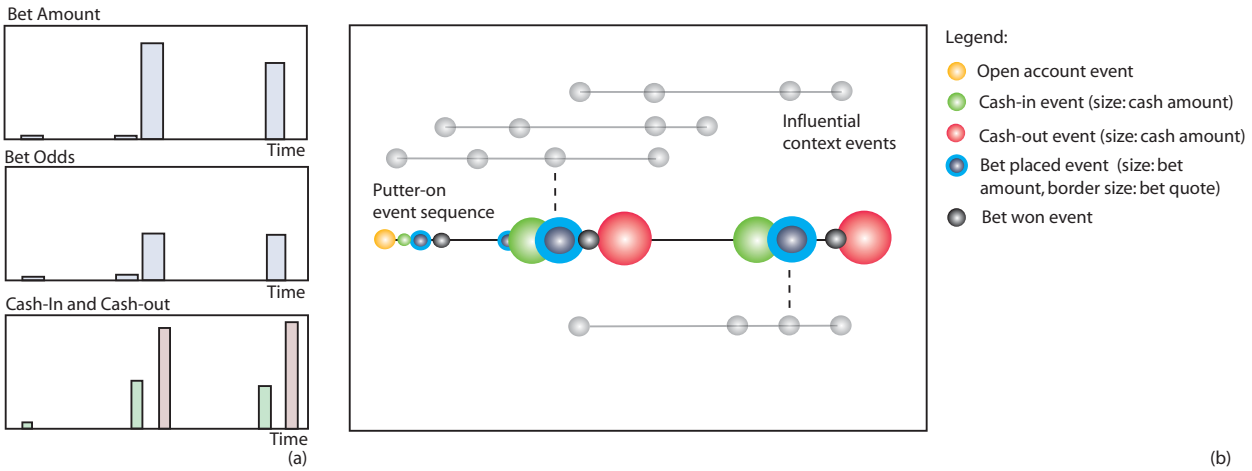


Fig. 5. Putter-on pattern characterization based on (a) aggregated data (OLAP) and (b) the sequence of events as behavioural pattern (event-driven BI)

and positions of events. A graphical query builder helps to formulate complex queries in order to optimize data retrieval. Client-side event filters help to filter out irrelevant event data.

With an event-driven BI tool such as the EventAnalyzer, relevant data can be easily extracted with an initial clue on hand (i.e., an official's prevented bet). The analyst can search the event data for events of users who placed the same bet within a certain time-frame before or after the prevented official's bet with a similar or higher bet amount. The retrieved result set can be further narrowed with visual outlier analysis and filtering mechanisms until a user is found that placed a bet which could be a putter-on bet [15]. The captured events related to this user represent a full qualified account profile. From the sequence of events, a detailed pattern can be derived. Decisive information such as that a putter-on always performs a cash-in before a high-stake bet, and then cashes out within a couple of hours, is directly visible. Once a first indicator pattern is found, the analyst can use this pattern to search the historic data for similar occurrences, and generalize the pattern.

Figure 5 schematically illustrates the analysis results of the putter-on pattern in OLAP and an event-driven BI tool. In 5a distributions of bets, odds, cash-ins and cash-outs support the characterization. In 5b the profile is characterized based on the sequence of events. The full pattern may include context events as well. Such events are not directly related to the user profile, but influence it (in case of the putter-on a relevant context event could be the bet placement failure of an official).

The example reveals that in order to fully qualify a fraud pattern, the sequence of customer actions is required. Patterns are often characterized by coherences among multiple data dimensions. Some specific patterns like the putter-on pattern or syndicates span multiple incoherent user accounts - relations between these accounts only result from related, concurrent or similar actions. In terms of the four enterprise data models (categorical, exegetical, contemplative and formulaic) characterized by Codd et al. [5], this implies the necessity for a contemplative model for fraud analysis that indicates unknown relationships between incoherent dimensions. Current OLAP implementations do not cover this requirement. Today, only categorical and exegetical models are available.

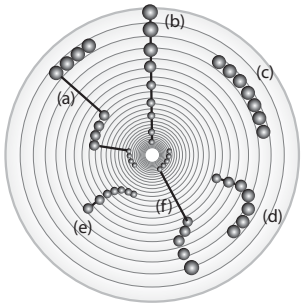
D. Moving from analysis to knowledge discovery

Figure 5 sketches the results available after a basic analysis. In the data warehousing discipline, this information serves as the basis for *feature selection* which is the starting point for data mining and knowledge discovery. Similarly, the basic indicator pattern identified in the event-driven analysis is the starting point for event mining.

We argue that the main difference between data mining and event mining is solely the perspective onto the data: Traditional data mining techniques usually depend on an abstract view on the data as provided by the entity-perspective of data warehouses to perform complex analyses such as the creation of decision trees, the discovery of classification or association rules or the arrangement of the data into clusters. For example, classification rules or clustering algorithms (taking a rather entity-perspective) can be used to spot potentially fraudulent customers, as long as they are not characterized by their dynamic behaviour. Event-driven BI on the other hand emphasizes an event-perspective onto the data. It focuses on the actions taken by or affecting such entities, instead of the entities themselves. Thus, the transition from analysis looking at sequences and distributions of events to event mining is floating. The first category of mining techniques applicable for an event-perspective on the data is *visual event sequence mining*. Patterns such as the putter-on pattern classify categories of event sequences. For visual event mining graphical depictions of patterns characterizing the chronology of events and their data attributes enable a fuzzy similarity search and immediate recognition of an event sequence's meaning. Several basic examples of visual event sequence patterns emphasizing the chronology of events are shown in figure 6. The plot shows the Event Tunnel visualization metaphor proposed by Suntinger et. al. [15] which looks into the past stream of events like a cylinder. The events are beaded along the cylinder mantle. Characteristic patterns result from the temporal order of events. Both techniques can independently

Property	*Data warehouse*	*Event-driven BI*
Granularity of data	Facts	Events
Latency	Periodic updates	Continuous updates, near real-time
Tracing related actions	Via conformed dimensions	Correlation of events
Extension and adaptation	Evolution of star-schema	Evolution of event types
Aggregation level	Aggregated	Detailed
Integration	Isolated, pull integration	Integrated, push mechanisms
Basic orientation	Subject-oriented	Event-oriented
Root-cause analyses	Drill-down, drill-across of aggregated information	Navigation through events at different levels
Level-of-detail control	Drill-down, drill-up	Event clustering
Dataset reduction	Slicing, dicing	Multivariate event filters

TABLE I
CONCEPTUAL COMPARISON



(a) Stair pattern (e.g. regular placement of multiple bets at once)
(b) Regular chain (e.g. a user placing one bet each day)
(c) Parallel chain (e.g. multiple nearly concurrently placed bets)
(d) Acceleration worms (e.g. a user who continuously places bets in higher frequencies (possible addictiveness)
(e) Deceleration worms (e.g. users decreasing the number and frequency of bet placements)
(f) Rattlesnake (e.g. sleeper accounts)

Fig. 6. Visual event sequence mining: characteristic event patterns.

be used to solve the same kind of problems but their differing perspectives can make their application more efficient as well as delivering more precise results. This leads us to believe that data warehouses and event-driven BI tools are not worlds apart but rather that their different perspectives can be seen as the two extremes of looking at the data. A certain family of data mining algorithms named frequent pattern discovery [1] which focuses on the order of different types of events shows that there have already been approaches towards an event-perspective in the data mining domain.

VI. SUMMARY AND CONCLUSIONS

In this paper, two technologies to detect, analyze and deal with fraudulent behaviour, especially in the growing market of online betting and gambling, have been presented and compared: data warehouses and event-driven BI. It was argued that the perspective on the underlying data is an important distinguishing feature allowing to decide which analysis solution suits a certain problem. Almost all data warehouse approaches require organizing the data into entities with the downside of losing an explicit interrelationship between those entities. Event-driven concepts using an event-perspective on the other hand aim at preserving these relationships on a low level of abstraction. For fraud analysis it is of importance to perform root-cause and cause-chain analyses on a detailed level. Hence, the query-driven approach of event-driven BI tools with fine data granularity and the possibility to fully reconstruct historic occurrences based on chains of events outperform the data warehouse analysis opportunities. Event-driven BI tools currently lack in providing aggregated information. Anyway, for fraud analysis and fraud pattern characterization this information is not as useful as in other business domains. Other differentiating factors like the architecture of event-based systems which integrates better into the operational structure are summarized in table I.

REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. *icde*, 00:3, 1995.
[2] J. Bailey, L. Crnogorac, K. Ramamohanarao, and H. Söndergaard. Abstract interpretation of active rules and its use in termination analysis. In *ICDT '97: Proceedings of the 6th International Conference on Database Theory*, pages 188–202, London, UK, 1997. Springer-Verlag.
[3] E. Baralis and J. Widom. An algebraic approach to rule analysis in expert database systems. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 475–486, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
[4] R. M. Bruckner, B. List, and J. Schiefer. Striving towards near real-time data integration for data warehouses. In *DaWaK 2000: Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pages 317–326, London, UK, 2002. Springer-Verlag.
[5] E. Codd, S. Codd, and C. Salley. Providing OLAP to user-analysts: An IT mandate. E. F. Codd & Associates, Technical Whitepaper, 1993.
[6] D. G. Conway and G. J. Koehler. Interface agents: caveat mercator in electronic commerce. *Decision Support Systems*, 27(4):355–366, 2000.
[7] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.
[8] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.
[9] B. Ludäscher. *Integration of Active and Deductive Database Rules*, volume 45 of *DISDBIS*. Infix Verlag, St. Augustin, Germany, 1998.
[10] 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.
[11] C. Phua, D. Alahakoon, and V. Lee. Minority report in fraud detection: classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1):50–59, 2004.
[12] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas. Discovery of fraud rules for telecommunications challenges and solutions. In *KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 409–413, New York, NY, USA, 1999. ACM Press.
[13] S. Rozsnyai, R. Vecera, J. Schiefer, and A. Schatten. Event cloud - searching for correlated business events. In *Proceedings of the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, pages 409–420, Tokyo, Japan, 2007.
[14] M. Schrefl and T. Thalhammer. On making data warehouses active. In *DaWaK 2000: Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, pages 34–46, London, UK, 2000. Springer-Verlag.
[15] M. Suntinger, H. Obweger, J. Schiefer, and M. E. Gröller. The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. To appear in *Pacific Vis 2008 - IEEE Pacific Visualization Symposium*, 2008.