Fraud Detection and Event Processing for Predictive Business
Abstract

The art and science of multisensor data fusion has emerged as an underlying foundation for Predictive Business™, including applications such as enterprise risk management and fraud detection. Enterprise fraud detection and other complex inference processing solutions require the management of real-time events from distributed sensors, agents and other processing components, including historical data-at-rest repositories. Distributed event-driven architectures such as TIBCO Enterprise Messaging Service™ or TIBCO Rendezvous® provide the underlying communications infrastructure that enables high performance rule-based event processing services. In this paper we discuss Predictive Business in the context of fraud detection with a focus on the distributed processing architecture. Emphasis is placed on the JDL data fusion model and TIBCO BusinessEvents™ in the context of event processing.
Introduction

In his latest book, The Power to Predict\footnote{Please note that the same high-level processing architecture applies to solutions in many other Predictive Business related areas, including opportunistic trading, network and telecommunications management, intrusion detection, real-time diagnosis, predictive battlespace concepts and more.}, Vivek Ranadivé, founder and CEO of TIBCO Software, discusses how Predictive Business is enabled by the fusion of historical knowledge with real-time information. The concept of the fusion of knowledge derived from data-at-rest with real-time data-in-motion is at the heart of understanding the momentum behind TIBCO’s event-driven architecture (EDA). We elaborate on the concepts of Predictive Business in the context of an overarching processing architecture that helps TIBCO customers and employees realize solutions in context to different business domains, while grounding the discussion to the application of fraud detection.\footnote{Refer to pages 46 and 47 of The Power to Predict\cite{1}.}

The Joint Directors of Laboratories (JDL) data fusion model has proven to be highly applicable to detection theory where patterns and signatures discovered by abductive and inductive reasoning processing (for example data mining) are “fused” with real-time events. The JDL processing model\footnote{Refer to pages 46 and 47 of The Power to Predict\cite{1}.}, illustrated in Figure 1, has been the dominant functional data fusion model for decades. The vast majority of real-time event processing architectures are based on the JDL model. Vivek Ranadivé indirectly refers to this model when he discusses how real-time operational visibility, in context of knowledge from historical data, is the foundation for Predictive Business.

The heart of the JDL data fusion model is a communications infrastructure that looks remarkably like TIBCO’s marquee Information Bus™ technology. This model also corresponds to the concept of a service-oriented architecture.\footnote{Refer to pages 46 and 47 of The Power to Predict\cite{1}.}

---

**Figure 1. JDL Data Fusion Model, adapted for complex event processing (CEP)**
Figure 1 illustrates the three main “meta” architectural components of the JDL model. Perhaps the most important of these components are the events. Events can be local and external and originate from myriad sources in many formats. There is no requirement that events must use the same syntax or format, as we will understand in the context of Level 0 Event Preprocessing. There is also the core event, or the so-called event stream processing architecture, which the JDL model depicts as the “data fusion domain.” In this paper, we will use the terms data fusion domain, data fusion, multisensor data fusion, complex event processing and event stream processing somewhat interchangeably, as the concepts are independent of vendor implementation and have similar technical objectives and outcomes. Finally, there is the user interface, providing operational visibility into every technical and business process depicted in the model.

The business objectives of organizations may differ in context, but the overarching technical goal to enable event-decision business objectives are the same:

*Correlate information gained from abductive and inductive reasoning processes with real-time events to infer current situational knowledge and predict both opportunities for, and threats to, the enterprise to maximize assets and minimize liabilities.*

This objective is at the heart of TIBCO’s vision for Predictive Business.

**Predictive Business Architecture**

The business outcomes of Predictive Business are realized when an enterprise can leverage knowledge and information gleaned from historical data as patterns that are applied to real-time events and situations. Figure 2 depicts a high-level communications architectural view of how business events are published to subscriber event services, such as rules or inference engines. In turn, the output of these rules or inference engines can also be events, complex events, situations, threats and/or opportunities which are published to subscribers. A similar diagram could be used to illustrate queue-based architectures.

The communications model, which industry analysts often refer to as an enterprise services bus (ESB), enables organizations to take advantage of business intelligence, real-time business events and other information sources, such as advanced sensors and rule-based processing. It is interesting to note that a robust distributed communications infrastructure is an underlying technical requirement for Predictive Business. Another requirement for Predictive Business is a logical data processing model.
This is the rationale for introducing the JDL model into a discussion of both event processing and fraud detection.

The high level requirements for Predictive Business are fairly straightforward when viewed in context of an established inference processing architecture such as the JDL model, summarized in the following sections:

**LEVEL 0 – EVENT PREPROCESSING**

Event preprocessing is often referred to as data normalization and basic feature extraction. Heterogeneous events from sources across the extended value chain, both external and internal to an enterprise, exist in many data formats, accessible through local methods and from application interfaces. Level 0 preprocessing is a generic term for normalizing data in preparation for upstream event processing. The terms data normalization, data cleansing, event normalization, and object preprocessing are terms that are often used interchangeably. In context, these terms refer to similar processes of preparing events for further upstream data processing.

TIBCO customers often use TIBCO Adapters™ and TIBCO BusinessWorks™ to normalize data before presenting the data to the next processing stage within an ESB. In addition, a high performance rules-based engine such as TIBCO BusinessEvents, discussed in more detail later, can also be used for data normalization and event preprocessing.
For example, visualize a high performance network service that passively captures all inbound and outbound network traffic from a web server farm of 200 web servers. Next we normalize, track and trace the traffic based on HTTP sessions prior to further detection and predictive processing. This visualization is a real world example of Level 0 event preprocessing for enterprise fraud detection in a financial services business scenario. As you might imagine, this can be a challenging problem when there are thousands of transactions per second in real-time and requires specialized software such as TIBCO BusinessEvents.

**LEVEL 1 – EVENT REFINEMENT**

Event processing rarely stops with event preprocessing. Data normalization and basic feature extraction is a housekeeping task – dirty work – but critical if the next stages are to be successful. After event normalization, other challenges loom on the processing horizon. Level 1 processing involves selecting events for inclusion in an event track or trace.\(^3\)

In the context of fraud detection, this is often a task of identifying possible badges of fraud by methods of association and correlation — classification by pattern matching in the normalized raw event stream. Generally, this is a forward-chaining process with three steps:

1. Hypothesis Generation
2. Hypothesis Evaluation
3. Hypothesis Selection

Event Hypothesis Generation, in the context of fraud detection, tags groups of events which may represent fraudulent activity on the network. The input is the normalized event stream. The output is a matrix or ScoreCards of possible events of interest.

Event Hypothesis Evaluation refines the event hypothesis generated to rank the events based on likelihood. For example, the output determines which events have a higher likelihood of representing fraudulent activity.

---

\(^3\) This selection process often involves the use of real-time ScoreCards, discussed in more detail later.
Event Hypothesis Selection further refines the events by attempting to associate a name or classification to the suspected fraudulent event.

This level, or stage, of event processing normally requires a high performance rules-based pattern matching algorithm. TIBCO BusinessEvents uses an optimized Rete Algorithm, discussed later in this paper, regarded as one of the most efficient algorithms for optimizing mainstream rule-based systems.\textsuperscript{[5]}

**LEVEL 2 – SITUATION REFINEMENT**

Situation refinement represents a higher level of inference than object or event refinement, where estimation and prediction of event states happen based on associations between events.\textsuperscript{[4]} Level 2 processing is commonly referred to as relation-based state estimation. The state of the aggregated events is represented by a network of relations among individual events.

Another way to view this stage of processing is that we have previously taken normalized data from the raw data or event steam and extracted basic features, processing the event steam by matching with patterns obtained from historical knowledge. This historical knowledge was extracted from a priori data-at-rest and then “fused” with the real-time event stream.\textsuperscript{[5]} After an event is identified it can be correlated with other real-time events, models, patterns and data, as illustrated in Figure 3, to infer and predict more complex events and situations.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{situation_refinement.png}
\caption{Discovering Situations from Detected Events}
\end{figure}

\textsuperscript{4} In the prior inference stage (L1) the process performed high speed pattern matching with a highly efficient rules-based engine to detect objects of interest in real-time. The outcome of L1 provided both an estimation and prediction of low level entity states based on inferences from the actual raw data stream.

\textsuperscript{5} There might be a bit of overlap and disagreement in emerging terms and concepts at this point in the discussion. Hence, for purposes of this paper, and given the liberty as the writer, I will refer to what “just happened” in Level 1 Event Refinement as “event stream processing” (ESP) and what “happens now” in Level 2 Situation Refinement as “complex event processing” (CEP) – but this is just a subjective opinion. In fact, a good argument could be made that ESP and CEP are the same thing. Objectively, there is not enough definition and vocabulary of either CEP or ESP to provide the required “absolute” context; and folks from different vendor camps are competing for position and influence in the marketplace. This is one reason I have chosen to ground the discussion in the JDL data fusion model, which is much better defined and less subject to marketing hyperbole.
The correlation of events into more complex or interesting events is what is referred to as situation refinement in the JDL data fusion model. We use the term “situation refinement” and “complex event processing” interchangeably because the output of this level of processing has the same goal: inferred situations, or complex events, from aggregated individual events.

Figure 4 represents how an organization might detect the fraudulent situation referred to as identity theft based on phishing. In this figure external event sources are represented as gray nodes and internal event sources are represented as blue nodes to illustrate an aggregate network of relations among individual events.

Each node in the graph is connected to other nodes by arrows which represent a cause-and-effect (causal) relationship. Nodes may be associated with conditional probabilities. As a graph, Figure 4 represents the organization’s a priori knowledge of how they might detect identity theft in the context of phishing, which is a situation inferred by the correlation of aggregated events.  

---

6 Detecting situations with the method represented in Fig. 3 was invented by Thomas Bayes in 1731. Bayesian probability is the name given to a family of related interpretations of probability, which have in common the application of probability to any kind of statement, not just those involving random variables. “Bayesian” has been used in this context since about 1950 [6]. There are numerous other mathematical techniques for detecting situations. Bayes’ theorem is the dominate technique in fraud detection and similar classes of detection problems including most modern email SPAM filters.
LEVEL 3 – IMPACT ASSESSMENT

After event detection and situation refinement, organizations are generally interested in ascertaining or predicting the intent of those responsible for the current situational threat. Processing situations based on models to infer intent is referred to as Level 3 Impact Assessment. Impact assessment is defined as the estimation and prediction of events and situations of planned, estimated, and/or predicted threat (or opportunistic) actions by the participants.

Therefore, Level 3 processing is normally implemented as a prediction based on higher level inferences built upon Level 2 event-object associations. At this stage of the model, we may estimate the impact of an assessed situation, which includes likelihood and/or cost/utility measures associated with potential outcomes of a player’s planned actions. From this inference, loss projections and liabilities (or gains) may be estimated.

For example, organizations are generally interested in detecting fraudulent situations in their networks. Organizations would also like to know, to the extent possible, the intent of fraudsters and the potential damage to the organization.

LEVEL 4 – PROCESS REFINEMENT

Event process refinement is an element of resource and task management. Functionally, Level 4 is where adaptive data management and real-time resource and process control take place. The focus of Level 4, in contrast to Levels 0-3 in the JDL model, is overall planning and control, not detection, estimation and prediction. Process refinement is the event-decision actions of assigning resources to tasks, performed by people or automated processes. This is also the stage where event-decision variables and models are updated, removed, added, tuned or otherwise refined.

DATABASE MANAGEMENT AND OPERATIONAL VISIBILITY

Supporting the overall model are databases of features and patterns extracted from historical data and other supporting databases. These databases may also contain models created by domain experts. For example, there may be a database of known IP addresses of Internet fraudsters or a database of known anonymous proxy servers used to hide the true origin of the fraudster. It goes without saying that operational visibility at all levels of the inference process is important.

---

Readers may be aware that TIBCO Hawk® provides a mature communications infrastructure for monitoring, command and control of distributed complex event processing components.

In some interpretations of the JDL model, operational visibility is referred to as a fifth level.
RBS and TIBCO BusinessEvents

Rule-based systems (RBS), historically referred to as a class of expert systems, are widely used to represent the knowledge of domain experts. For this reason, RBS are used extensively in a wide variety of business applications such as customer relationship management, fault and medical diagnosis, mortgage evaluations, credit card authorization, fraud detection and e-commerce. These systems use declarative programming to tackle problems involving control, diagnosis, intelligent behavior and problem solving by describing what the computer should do rather than the exact procedural instructions on how to do it. Rule-based systems contain rules from a specific domain and use these rules to derive solutions to a wide range of business problems.

Rule engines are processing architectures which are designed to follow rules that are written in their language. These engines are used to execute decision-making logic on data and event streams to support decisions based on tens, hundreds or thousands of facts. Rule engines accomplish this by decomposing large sets of rules into an efficient network of nodes, which can process and react to changing facts far more efficiently than can be programmed using procedural methods. Therefore, rule engines, if properly designed, scale well for numerous classes of problems, including event-decision processing.\footnote{This section is intended as only a brief review of rules engines, the Rete Algorithm, and a few key concepts related to TIBCO BusinessEvents. Please refer to the references for a more detailed discussion or explanation.}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure_5_TIBCO_BusinessEvents_High_Level_Architecture.png}
\caption{TIBCO BusinessEvents High Level Architecture}
\end{figure}
The optimized Rete Algorithm\textsuperscript{10} is a well-known efficient RBS implementation that also creates a network of nodes. Each node in the network represents one or more test conditions found on the left-hand side (LHS) of a rule set. At the bottom of the Rete network are nodes representing individual rules. When a set of events filter all the way down to the bottom of the network, the set has passed all of the tests on the LHS of a particular rule and this rule set becomes an activation and potentially generates one or more events as input to other processing or alerting solutions. The associated rule may have its right-hand side executed (fired) if the activation is not invalidated first by the removal of one or more events from its activation set.

TIBCO BusinessEvents was designed to provide organizations the capability to execute event-decision logic without the programming complexity. Using a declarative programming model, business and technical architects can define rules that will execute on both individual and combinations of events and facts in working memory. The BusinessEvents high-level runtime architecture is represented in Figure 5.

The core rules-engine architecture of BusinessEvents is illustrated in Figure 6.\textsuperscript{11} A TIBCO BusinessEvents rule has three components. The declaration component is used to specify which concepts, ScoreCards and events\textsuperscript{11} the rule requires (as well as naming attributes). The condition component represents facts that evaluate to a Boolean condition.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure6.png}
\caption{TIBCO BusinessEvents Rules Engine}
\end{figure}

\textsuperscript{10} As mentioned earlier, the BusinessEvents rules engine is based on an optimized Rete Algorithm. The Rete Algorithm is an efficient pattern matching algorithm for implementing rule-based systems designed by Dr. Charles L. Forgy in 1979. Subsequently, Rete became the basis for many expert systems.

\textsuperscript{11} TIBCO defines an event as an immutable object representing a business activity that happened at a single point in time. An event includes information for evaluation by rules, meta-data that provides context, and a separate payload — a set of data relevant to the activity.
All conditional statements must be true for the rule’s action to be executed. The action component of a BusinessEvents rule is a collection of statements to be executed when all the conditions are true.

A BusinessEvents event represents an instance of an event definition, an immutable activity that occurred at a single point in time. An event definition includes properties evaluated by the BusinessEvents rules engine, including time-to-live (TTL) and comprehensive information related to the event. A concept in BusinessEvents is an object definition of a set of properties that represent the data fields of an application entity and can describe the relationship among entities. Likewise, an instance of an event is persistent and changeable; whereas an event expires and cannot be changed.

A ScoreCard is a BusinessEvents resource that serves as a container for global variables and can be used to track information across the application. For example, in the JDL data fusion model, matrices are used for assignments at all levels of the inference processing model. As illustrated in Figure 7, multiple ScoreCards can be used, conceptually, for this dynamic assignment at Level 0 through 4 of the processing model.

![Figure 7. ScoreCards and JDL Processing](image)

---

Note that although some applications of the JDL model show ScoreCards used for Level 0-4 association and state estimation variables, ScoreCards, from a BusinessEvents implementation perspective, may or may not be the most efficient method to extract features from an event stream. ScoreCards are introduced here as both a generic concept (scorecard) from the JDL model used for explicit associations in performing state estimation and as the BusinessEvents ScoreCard resource. In practice, BusinessEvents implementations of association and state estimation will often use the BusinessEvents Concept resource.
The multi-level fraud detection ScoreCards example in Figure 7 illustrates how an event stream might be processed into “interesting events” by extracting events from the event stream. These events are ranked, or scored, according to a likelihood estimation of relevance to fraudulent activities. These events, ranked according to likelihood, may be further processed to infer more complex events and situations. These complex events and situations are then evaluated for business impact. The entire event stream should be reconfigurable in real-time to provide the capability to fine-tune all detection and estimation processes.

This section only scratched the surface with regard to the runtime capabilities of BusinessEvents, rules-based systems, rules engines and the optimized Rete Algorithm. For more information, see the Additional Reading list at the end of this paper.

---

13 Inference and multi-sensor data fusion can be a difficult subject to grasp. You are not alone if you feel this way! That is one of the reasons I have grounded parts of the discussion in fraud detection — to hopefully provide an element of business context to the dialog.
Conclusion

TIBCO BusinessEvents defines situations as an abstraction that results from patterns detected among related aggregated events. This definition of a complex event corresponds to the JDL Level 2 inference abstraction referred to earlier as a situation. Figure 7 illustrates how building inference in a fraud detection scenario may be represented by using the built-in BusinessEvents ScoreCard resource.

Our goal in this paper was to illustrate that real-time sensor data, often described as the “event cloud,” can be processed and refined using rules-based event-decision methods. From event streams it is possible to extract features in real-time based on matching patterns in the event stream. Features of the raw data or event stream become “events (objects) of interest,” which can be scored according to likelihood estimates constructed from high speed pattern matching algorithms with low latency.

Events may be aggregated and correlated in runtime to infer complex events also referred to as “situations.” Complex events and situations discovered in runtime can be processed with patterns developed from historical data to predict business impact and other future events and situations. This, in turn, brings us back full circle to Predictive Business, the theme of TIBCO CEO Vivek Ranadivé’s latest book, The Power to Predict.

Real-time events, combined with patterns and features extracted from historical data and knowledge, are the foundation for businesses to anticipate exceptional situations, estimate the impact on both businesses and their customers, and take corrective actions before exceptional situations become problems. In other words: Predictive Business leverages business assets to maximize opportunities and minimize future organizational liabilities.

Enterprise risk management and fraud detection is but one of many examples of how the concepts of Predictive Business can help organizations minimize threats and liabilities to their customers, the enterprises that serve them, and the investors who place trust in their organizations.
References


Additional Reading


About the Author

Tim Bass (tbass@tibco.com), CISSP, is a Principal Global Architect for TIBCO Software Inc. He is currently focusing on emerging commercial applications for Predictive Business and event-decision processing for TIBCO. He has provided independent Senior Subject Matter Expertise to both government and industry for over 15 years, including the United States Air Force (USAF), the Office of the Secretary of Defense (OSD/NII), and global multi-national financial organizations, for example, Chase Manhattan Bank, the Swiss Bank Corporation (SBC) and the “Society for Worldwide Interbank Financial Telecommunication” (SWIFT). Mr. Bass graduated B.S.E., Tulane University, School of Engineering, 1987 Magna Cum Laude, Electrical Engineering. His work on Internet security and countermeasures for the DOD has been featured in Popular Science Magazine and Federal Computer Week. He is internationally recognized as a thought leader in next-generation intrusion and distributed multi-sensor data fusion architecture, in part, based on his paper, Intrusion Detection Systems & Multisensor Data Fusion, Communications of the ACM, pp. 99-105, Vol. 43, No. 4, April 2000.