Complex Event Processing in Financial Services

By David Luckham
Stanford University

David Luckham, Professor Emeritus of Electrical Engineering, Stanford University, is the author of The Power of Events and the inventor of Complex Event Processing (CEP).

In this white paper, developed exclusively for TIBCO Software, he shows how CEP can help organisations in the financial services industry detect patterns of events in the IT layers of the enterprise and predict how they will impact high level business goals, policies and processes.

AN EVENT DRIVEN INDUSTRY

The financial services industry is just about as information intensive as it gets. Everything in this industry is event driven.

Figure 1 depicts the messaging collaboration between financial enterprises of all kinds: the Federal Reserve and investment banks on the left, hedge funds and full service brokers on the right, and many others, not least of which are the individual online traders, shown at the top. Collaboration takes place over a variety of media from the Internet to cell phones to private networks like SWIFT. Quite possibly someone is still using ticker tape. How does collaboration take place?

Well, events are flowing to and from each enterprise across the media. We're not talking network packets here. These collaboration events are application level events such as email messages, stock quotes, buy, sell and stop loss orders, contract negotiations and so on. That is, they are events created by the use of applications – the tools of the financial industry. There are billions of events per second. For example, today a mid-sized banking institution estimates that it has 1500 business events per second on its IT infrastructure – and that does not include the external events in other infrastructures that it may want to factor into its decisions.

Each enterprise in Figure 1 is reacting to the events it receives, say by taking the next step in a transaction to trade stocks or negotiate a contract. It receives events and reacts to some of them by creating new events, which are sent out to other enterprises, and at the same time it may change the state of some of its internal records. In this sense, the activities of each of the enterprises in this collaborative scheme are event driven.

THE GLOBAL EVENT CLOUD

There’s a problem with this picture of collaboration. The events in the IT infrastructures of any single enterprise, or of the whole system, form a disorganised cloud. They are created in a distributed system, so many of them are created independently and may arrive at their destination in any order – not necessarily a
nice orderly sequence one after the other. Not only that, but you may not be able to tell which events are related to one another — for example, which events caused any event you happen to look at?

Consider a simple example. Something happens in your Tokyo office; something else happens in your New York office. Did one event cause the other, or did they happen independently? What information would you need in order to know? Perhaps they were independent events and happen to conflict by overwriting the same record in a database in your San Francisco office. Sometimes separate business processes aren’t coordinated!

So, from the viewpoint of any of these enterprises in this collaborative stock market system, the events in its IT infrastructure at any moment form a disorganised cloud. We call this the global event cloud. The problem for the enterprise is to infer as much information — some call it business intelligence — from the event cloud as it can. Most important, it needs to detect in real time any events that are going to impact its own high level business objectives. After business hours is too late!

Consolidation within financial services has often compounded the problem of making sense of the global event cloud by increasing the diversity of applications and networks within the enterprise. Consequently it is sometimes difficult for a single enterprise, especially one that has grown through acquisition, to get a complete end-to-end view of its own IT event cloud to see what’s in it. Making business sense out of it is quite beyond the capabilities of such enterprises.

**FINANCIAL SERVICES**

—the forest of information problems

Some people see problems as opportunities. They may be right. Here, an information problem refers to a situation that cannot be handled adequately without a new technology for extracting information from the event cloud.

The different kinds of information problems that are cropping up in the financial services industry today are too numerous to list here. Some of the broad categories follow.

**Fraud and compliance:** The problems in this category are multiplying exponentially. They include (1) credit card fraud, most common being the result of stolen cards or card information, (2) insurance fraud, common examples of which involve home insurance, automobile accidents or health insurance scams, (3) money laundering, which typically involves multiple accounts and institutions, and (4) various stock trading scams such as market timing in mutual funds and trading ahead on large orders.

**Advantageous trading situations:** Typically in finance, these are arbitrage situations occurring across different markets. Given the developing situation in markets such as oil, the modern set of event processing problems now spans situations that might not traditionally have been “financial services”. These problems involve using event clouds external to the enterprise’s environment.

**Focused marketing:** A classic example of this would be a bank using its consumer events, say credit card transactions, to focus the marketing of its financial products on suitable consumers. Take the case of a bank that detects a baby crib purchase on a customer’s credit card transaction. Consumer behaviour models show the majority of couples buying cribs are having their first babies. The models show a predictable progression of needs and purchases, including a bigger home, life insurance, etc. The bank can exploit that knowledge to cater to the customer for products such as life insurance, trust accounts, etc.

**Business intelligence:** While this terminology is much overused, it serves here to categorise a slew of real time problems to do with understanding what is happening in the enterprise’s own IT operations. For example, recognising in real time how an enterprise's own business processes and trading transactions are progressing, which ones are stalled and why, how to prioritise processes that are in progress but are competing for resources or otherwise conflicting. And, most of all, everyone wants to know what their competitors are up to! That means being able to view and analyse the event cloud from a very wide set of sources.

There is, however, one fundamental capability common to building solutions to any of these different kinds of information problems. To deal with any of them, one must be able to process the global event cloud in real time to detect the presence of...
patterns of events that can impact the problem concerning us. Essentially any event driven enterprise needs an ability to have insight into how the events at any layer in its IT infrastructure may affect its business objectives. For example, stock price quotation feeds from different markets may contain information indicating an advantageous trading opportunity. That information needs to be extracted in real time and fed to trading algorithms. Another class of examples is contained in the previous focused marketing example from CRM.

Typically, the patterns of a consumer’s banking events that need to be detected are more complicated than a single credit card transaction.

The events that might impact a management problem might be high-level business events created by steps in business processes, or lower-level events created by applications, or they might even be messages in the middleware layer. The capability to detect the management implications of events in the IT layers is called IT to business insight. Not to be confused with network management, it helps to understand the enterprise management implications of events at any level in the IT infrastructure.

One of the keys to this definition, of course, is what is meant by a “pattern of events”, which will be explained a little later. But first, a brief account of the scramble to provide IT to business insight.

**THE FIRST ANSWER: BAM**

“Business Activity Monitoring” is a term coined by Gartner, Inc. to describe the wealth of point solutions that have grown over the past four or five years to address some of the problems in financial services. Gartner lists over 50 vendors of business event technology products. The list runs from the largest vendors through to a myriad of emerging startups.

The classic paradigm of a BAM solution is the dashboard model shown in Figure 2.

On a BAM dashboard those enterprise objects (applications, services and resources) in the IT infrastructure that are critical to the running of the enterprise’s business processes are wired up to activity monitors. That is, the events going into and coming out of each object are read by an activity monitor. This is non-invasive monitoring, which means that the events are not delayed or altered. The types of enterprise objects that can be monitored may vary from high-level applications and services to low-level IT resources. Monitors compute predefined (often called “out-of-the-box”) metrics using the events from these applications. Common examples are:

- Number of website calls in past hour
- Inventory level in warehouse
- Average database response time in last five minutes
- Load on a server
- Number of active instances of a business process.

These metrics are often called statuses. Statuses have warning levels, e.g. normal, abnormal, critical, which can be set to numbers that are sensible for an individual customer’s business activities. Monitors create events containing status information (status events) that are displayed on the BAM dashboard. These dashboards present quite sophisticated graphical representations of the statuses, the motivation being to help the user interpret the implications of status information for the business activities that are in progress.

This is the basic dashboard paradigm of how BAM solutions work. What is involved is simple event monitoring. Events are fed into the activity monitors in a continuous stream, they are crunched according to fixed metrics and the results (statuses) are displayed graphically. The smarts, i.e. the interpretation of the status information, are in the head of the user. These tools do not provide any facility for matching complex patterns against the event streams, or combinations of event streams from different sources. There is no global view – the user must produce that.

Nowadays BAM solutions are certainly going beyond the dashboard model and offering more capabilities. One direction is automatic warnings when statuses reach criticality. The warnings can be configured by the user. They may consist of email messages to various managers, or cell phone calls and other means of getting people out of bed.
Another direction is the use of event-condition-action rules. These are reactive rules triggered by single status events. They can be used not only to automate warnings, but also to control or modify processes in the enterprise.

BAM solutions are using monitoring to offer new capabilities. These include (1) Business process viewing: monitoring event feeds from multiple applications to deliver real time views of the progress of business processes. Also, analytics can then be applied to business process performance to deliver a wealth of performance statistics on processes. This can be especially useful in cases where the process owner outsources steps in the process to several different subcontractors as is often the case in mortgage loan processing. (2) Business impact analysis: here monitoring of lower level IT assets is combined with the use of graphs of the dependencies between IT resources to predict when a critical status of a low-level object (say an overloaded website or database) will impact the execution of higher level applications or processes. However, since these dependency graphs do not account for runtime variations, the analyses are worst case predictions.

Looking over the current set of BAM solutions, two things are quite clear:

- All of the solutions have limited applicability because the underlying event processing they rely upon is very limited. Detecting single events, or very simple patterns of events such as A and B or A or B, isn’t sufficient to deliver IT to business insight.
- This marketplace has developed in a totally chaotic manner. This is particularly evident in the fraud detection area which is a large collection of point solutions.

BAM has developed without any standards. No standards to aid integration of solutions from different vendors. No standards to aid construction of BAM solutions based upon existing middleware, although some proposed standards such as the Common Event Infrastructure (CEI) are beginning to appear. No standard specifications of the information processing problems being addressed. And nothing by way of modular components of BAM solutions, say event processing rules languages and scalable rules execution engines.

Philosophically, one can take the view that chaos is perhaps always the way a new area of software applications begins. And when the market is established, then the standards and modular components for building solutions follow. But also, a new applications area cannot mature without an underlying scientific basis for the new things it brings to the table. One of the basic foundations of solutions to provide IT to business insight is Complex Event Processing (CEP).

**COMPLEX EVENT PROCESSING: A SIMPLE EXAMPLE**

Here’s an example I have used in other articles, the event cloud on an online banking website.

The cloud of events on an online banking website consists of account logins, deposits, withdrawals, transfers, etc. These events can be created at any computer linked to the website. Figure 3 depicts an event as an oval and the relationship where two events must have been executed one after the other as an arrow. If you look closely on the left, you can see a sequence of events where an account login is followed by a deposit followed by an account activity check. Those events would be created by a customer’s banking activity on a single account. They would be normal behaviour. And because they are on the same account, they must be performed one at a time. There’s a lot of activity on other accounts going on simultaneously. Events on different accounts don’t have to be executed in a one-after-the-other order. They can happen at the same time or different times. Figure 3 shows several threads of activity on different accounts, each thread being a sequence of events connected by arrows. Today, an online banking website may have millions of active customers. There are possibly thousands of banking events per minute in the website’s event cloud at certain times of a day. Banking websites have been frequent targets of identity theft over the past two years. A Gartner telephone survey of online

![Figure 3. The event cloud in an online banking website](image-url)
banking customers in the US alone showed that in a one-year period during 2003–2004, over two million customers had money stolen from their accounts, the total loss being about $2 billion. Also, phishing attacks aimed at identity theft have increased by an average of 26 per cent per month since July 2004. No matter how hard banks try to educate their customers, a percentage of the online accounts will always be at risk to theft resulting from the legitimate customers losing their account information to crooks.

Figure 4 shows patterns of banking events that are typical of what crooks do when they gain access to accounts by means of stolen identities. Here we see patterns of event activity on accounts in which a login is followed by a password change followed by a new automatic payment order all within a short time. The two instances of this pattern in Figure 4 happen in different contexts. In the first, all three events follow each other directly. But in the second there’s an account balance enquiry in between. In fact, several other events might happen in between, but as long as the three critical events happened in a short time, they would be flagged as suspicious. For example, a crook might login, execute password change and logout. Then wait a short time before another login and the new automatic payment order. Why the short time span? Because the crooks want the transfer to execute before anyone notices the identity theft.

Figure 4 also shows what a monitor for suspicious banking activity should be able to do. Detect a sequence of several (in this example, three) critical events in different contexts, and if they happen in a short time span, create an alert event. Monitoring for suspicious account activity needs to be able to detect many similar types of patterns of events in different contexts and with different timing constraints. In more complex cases, suspicious activity may encompass several accounts and involve more devious operations such as installing new beneficiaries to stolen accounts. So the event patterns won’t be simple sequences of events in a given timeframe, but may involve concurrent and independent sequences of events on separate accounts over long time periods. However, a capability to monitor patterns of simple sequences of events shown in Figure 4 would be a good start towards detecting cases of possible online banking theft and holding up the money transfers for verification.

The point I want to make in this online banking example is that the kinds of event patterns we need to detect usually involve:

- Several events, possibly some of them sharing common data elements.
- The events may need to happen in a specific order, possibly allowing events that are not part of the pattern to happen in between.
- Some of the events in a pattern may happen independently, and in any order, while others may be causally related.
- There may be timing bounds within which the events must happen and other kinds of constraints on the data in the events.
- In some cases, the event pattern may require detection over long time periods since the activity may be designed to fly under the radar of the average detector.

Event patterns can get quite complex. So describing them requires precision. We have to be able to do this before we can take the next step, building pattern recognition engines to automate detecting patterns in an event cloud. So step one towards IT to business insight requires developing precise descriptions of event patterns.

**HOW WE GAIN BUSINESS INSIGHT**

Suppose we have gotten to step one. We have powerful event pattern descriptive facilities and scalable performance rules engines to detect patterns of events on our IT layers in real time. What other features might we need to deliver IT to business insight?

In the remainder of this paper, let me deal with just one additional CEP feature. I happen to think it’s the next big step, after step one.
Remember Nick Leeson? It might be a good bet that if the board of Barings Bank had been faced with his raw set of futures trades on the Osaka Stock Exchange and the Singapore International Monetary Exchange in 1995, they still might not have been able to understand exactly what he was doing. I haven’t seen his actual trades across both exchanges during the months following the Kobe earthquake, but they were probably quite complicated. So, even if the head office in London had gotten real time reports of all his trades and hedges, Barings might still have not taken any action, and consequently gone bankrupt.

Once we detect a complicated pattern of events that ought to tell us something about our business goals or policies, we may still need help understanding it — if only to save time. This is the point of Figure 5. What you would really like is a facility that gives you just the information from this event pattern that you need to know. We call this facility, event pattern abstraction, see Figure 6.

Anyone at a business management level in any of these enterprises does not want to see all these event communications in a transaction. Managers want abstractions of the trade level event patterns that relate to their roles in the enterprise. A CFO, for example, might simply want financial data summary. And he may want the data on each successful transaction streamed into his spreadsheets as it happens. On the other hand, an IT manager may want an entirely different view of the same event pattern — timing data on the IT support for the same transactions, particularly if the other parties complained about slow response.

The role of abstraction is to deliver relevant views of the events at the trade level to different managers. This is done by creating new events that are computed from the trade level patterns and contain abstracted data. We think of them as higher level events. Figure 6 shows the concept of abstracted views.

What happens when a trading deal is almost complete but slows down or hits a glitch? For example, the settlement phase seems to be taking longer than normal to complete. Are there events in the stock market system signifying a competitor has just entered the marketplace with a better price? Can the settlement be completed by offering better terms? In this case it is just as important to have a real time abstraction of the progress of an incomplete transaction pattern. This is real time trading dependent upon detecting incomplete transaction patterns and factoring in additional patterns from other sources.

The role of event pattern abstraction is to give you views of the cloud of events on your IT layer that you need. The technology to deliver event pattern abstraction needs to be flexible so that different abstract views of the same event pattern instance can be delivered to different personnel at the same time.

At present a flexible technology for abstracting many types of event patterns is only just beginning to emerge. It truly goes beyond BAM and is one of the basic concepts in complex event processing (CEP).

Figure 5. Events in the two-phase commit termination to a trading protocol between three banks

Bank requests completion of an eTransaction

ETC polls the participants in parallel

Participants reply in parallel

ETC’s decision coordinates all replies

Confirmation phase
To summarise, three of the first technology developments towards IT to business insight are:

- Facilities (graphical or textual) for precise description of complex patterns of events.
- Scalable performance, modular rules engines to detect complex patterns of events.
- Facilities for defining and composing event pattern triggered rules for pattern abstraction.

There is, of course, a lot more to accomplish. But these first steps will certainly help to put BAM on a unifying technology basis. My book on CEP, The Power of Events, provides detailed principles on how to accomplish these three steps and more, together with descriptions of proof of concept studies.