Three Steps Towards Information Technology Insight

A White Paper By:

David Luckham Professor
Emeritus of Electrical Engineering

Presented by:

APTSOFT™

Event-driven Application Collaboration™

AptSoft Corporation • 20 Mall Road • Burlington, MA • 01803-4123 • 781.270.4900 • fax 781.270.4904 • www.aptsoft.com

Until very recently technology for monitoring IT traffic focused on the network level. This helps to keep the networks running but it is useless for delivering business intelligence. As a result, today every enterprise is in a situation of not understanding how the events that travel through its IT infrastructures will effect its high level business transactions and strategic operations, minute by minute or day by day. Managers can't figure out in real time when events on their IT layer — and it's not just single events, but more generally patterns of many events — are significant from a business perspective. In effect, they are blind to the business implications of the events on the IT layer. I call this situation **IT blindness.** It's a real time problem. It can't be solved by storing events in databases and searching them later — that just isn't fast enough.

Not only does IT blindness prevent us from effectively monitoring and managing our business processes in real time, but it also blocks implementation of some of our grand visions for the future of eCommerce. It will soon become a major obstacle in the path towards fully utilizing new technologies like RFID that result in creating large numbers of new types of IT events.

Furthermore, much is currently being written about *alignment.* That is, aligning IT investment with business goals. One of the most important alignment targets for IT investment in a real time enterprise must be to improve business process technology, e.g., flexibility to change processes quickly, and capability to track what they are doing. Any investment in technology to predict how events in the IT layer will impact the successful execution of business processes will certainly be in alignment with business goals. And it will be a step towards solving IT blindness.

In the last three or four years people have begun to understand the challenge of solving IT blindness. Those who do, view it as a goldmine business opportunity. Gartner lists well over a hundred vendors selling tools that build on top of the current generation of middleware to track the progress of business processes, or predict violations of high level policies, or analyze the impact of overloaded IT resources on the progress of currently executing business transactions. These are some of the things you can do, or claim to do, with just a little **IT insight.** Roy Schulte of Gartner was the first to recognize the tip of this IT insight iceberg when it first became noticeable a couple of years ago and named it "BAM" for Business Activity Monitoring.

The basic "dashboard" paradigm for BAM tools is described in my article, http://www.complexevents.com/about/articles/bam_beginnings.html. To really achieve **IT insight** we need to go a lot further, technologically speaking.

Information Technology Insight *is the ability to predict how patterns of events in the IT layers of an enterprise will impact high level business goals, policies and processes.*

A little explanation is in order. Here it is.

In The Beginning: the Global Event Cloud.

First of all, there is the **global event cloud,** the environment of events on the IT layers of the modern real time enterprise. We're not talking about network packets. This is a cloud of business level events. In most cases it consists of events from inside the organization and also contains communications from outside, which is why "global". It is usually unstructured since many of the events result from activities distributed around the world, so it's a "cloud". Here's an example I used recently in another article, the event cloud on an online banking website, see figure 1.

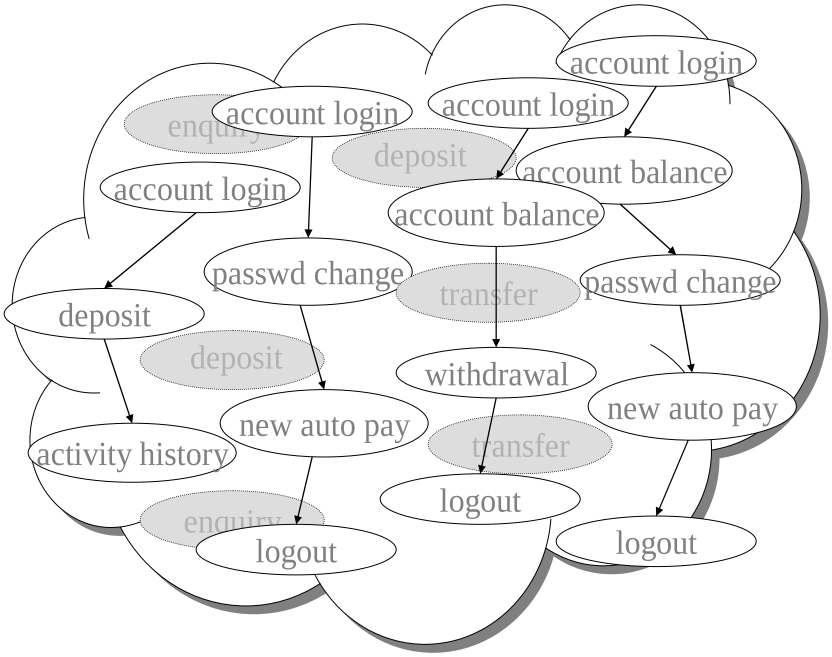

Figure 1: the event cloud in an online banking website

There is a vast cloud of events on an online banking website, **account logins**, **deposits**, **transfers** etc. These events can be created at any computer linked to the website. Figure 1 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 behavior. 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 1 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 day.

If you look at the IT layers of any other kind of enterprise you'll find a similar cloud of events pertaining to the business activities of that enterprise. Examples of patterns of IT events that have critical implications for strategic business objectives crop up in all electronically based business contexts -- supply chains, trading in stocks, futures or financial instruments, electrical power transportation, consumer relations management, SCADA control systems, and very often in electronic barter involving multiple enterprises. It's a common situation. We're sitting in a cloud of events, we want to know what's happening to our business, and we want to know now!

Step 1: Describing Patterns of Events.

The first step towards IT Insight is to be able to detect what's happening in that cloud. How? Let's go back to online banking for examples of event patterns and why they are important.

Figure 2 shows patterns of banking events that are typical of what crooks do when they gain access to accounts by means of stolen identities. There 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 2 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, we would want to flag them 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.

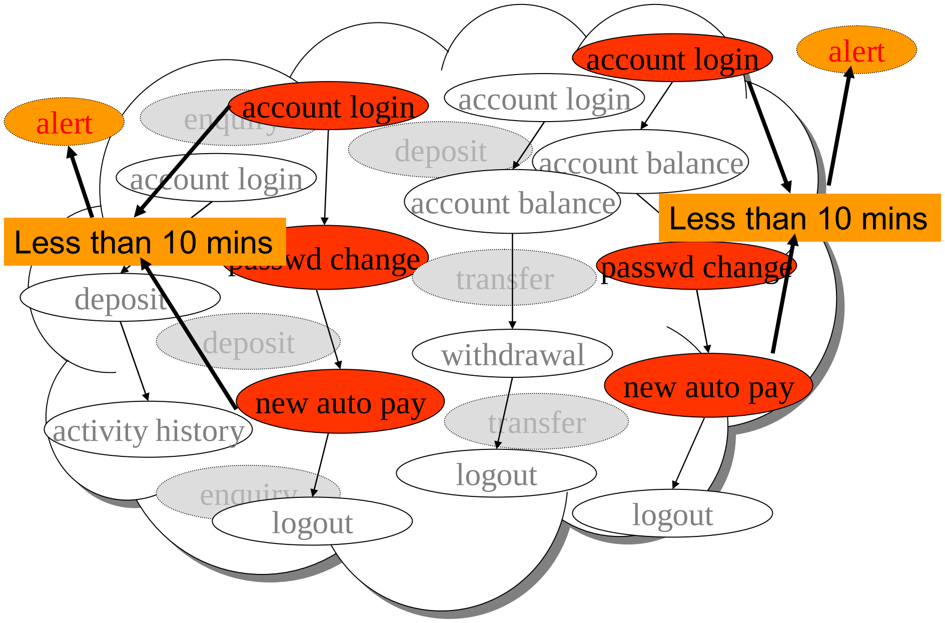

Figure 2: patterns of suspicious activity in an online banking event cloud

Figure 2 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, so the event patterns won't be simple sequences of events with a time bound, 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 our figures would be a good start towards detecting cases of possible online banking theft and holding the money transfers for verification.

The point I want to make in the banking example is that an event pattern usually involves

- 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.

Event patterns can get quite complex. And describing them needs precision. But 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 1** towards **IT insight** requires *developing precise descriptions of event patterns.*

Software scientists have been doing similar things with programming languages for forty years. And computer scientists have been developing formalisms for events and timing logics for almost as long. So this step isn't something that requires any great new science. It just needs to get done!

Step 2: Detecting Patterns of Events.
Electronic commerce is speeding up. Transactions between enterprises that used to take a week or more to negotiate may now take hours. Management must speed up too, and needs the tools to do so.

Event pattern matching is the fundamental technology needed to detect patterns in a global event cloud. This technology is as yet in its infancy. The trend in business activity monitoring is towards developing pattern detection engines to monitor for complex patterns of several events involving sequencing and timing and concurrent events. For some applications the speed of pattern matching execution becomes critical in deployment — so-called *scalability of implementation.* There are three dimensions to scalability, the number of patterns to be detected, the speed with which events are entering the event cloud, and time.

A good benchmark for event pattern detection engines to aim for would be the **thousand-by-a-thousand-per-second** test:

> *an ability to monitor a thousand different patterns of events, similar in complexity to the one in our banking example, on an event cloud that is creating new events at the rate of a thousand per second.*

This is quite a challenging benchmark, but the technology is getting there. Meanwhile, there are many areas of ecommerce where IT insight can be delivered by an event pattern detection engine with less than this level of performance.

Step 3: Event Pattern Abstraction.
This step towards **IT insight is** the great leap forward – or rather, *upwards*!

Event patterns can be complicated enough to defy understanding by personnel who are novices at event processing. And why indeed should they need to know about that – they have enough to do managing the business! **IT insight** involves delivering the information contained in patterns of events in the IT layers to personnel with a variety of different roles in the enterprise. This is done by *aggregating* and *abstracting* the data in the event pattern.

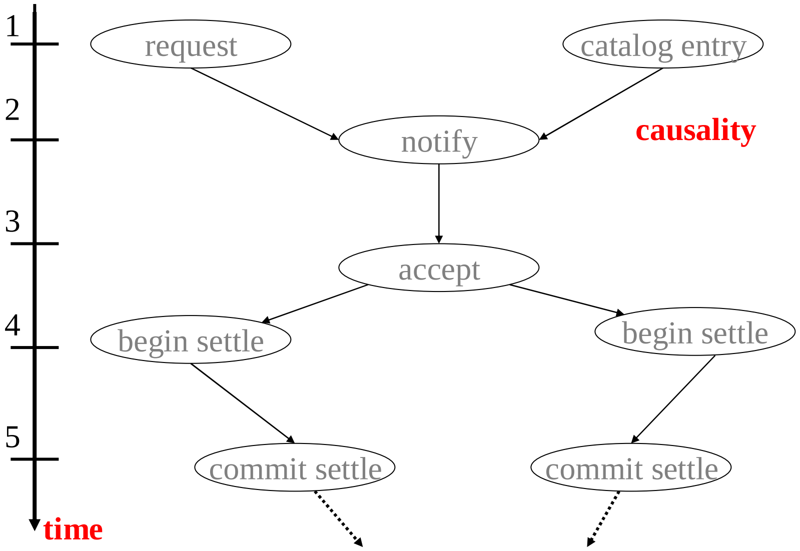

Figure 3: Pattern of events in a supply chain transaction

For example, figure 3 shows an instance of a transaction involving four enterprises, a *buyer*, a *seller*, an *auction facility* and a *settlement facility*. Essentially we are looking at the event activity in an eMarketplace or electronic supply chain. The events involve the buyer making a **request** for product to the auction facility. A seller independently makes an **entry** into the catalogue of the auction facility. These two events cause the auction facility to **notify** the buyer of a product that may meet its requirements. Here, this causes the buyer to send an **acceptance** of the seller's product and price back to the auction facility. The auction facility then triggers the settlement phase of the transaction by sending **begin settlement** events to the seller and the settlement facility, which then takes over as the go-between in completing the exchange of product and payment between the buyer and seller. The pattern instance shows the causal and timing relationships between events at the trading level.

You don't really want to see event patterns like this, do you?

Anyone at a business management level in any of these four enterprises does not want to see all these event communications in a negotiation. In fact the details can get a lot more complicated than this example, involving several buyers and sellers. Managers want abstractions of the trade level event patterns that relate to their roles in the enterprise. A CFO for example, might simply want summary data about products, prices, payment dates. And he may want the data on each successful transaction streamed into his spread sheets 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 4 shows the concept of abstracted views.

Equally important are abstractions of transactions that are almost complete but seemed to slow down or hit a glitch. For example, a transaction reaches the settlement phase but is taking longer than normal to complete. What is happening? Are there events in the eMarketplace signifying a competitor has just entered the marketplace with a better price? Can the settlement be completed by offering the buyer a discount? This is real time trading dependent upon detecting event patterns.

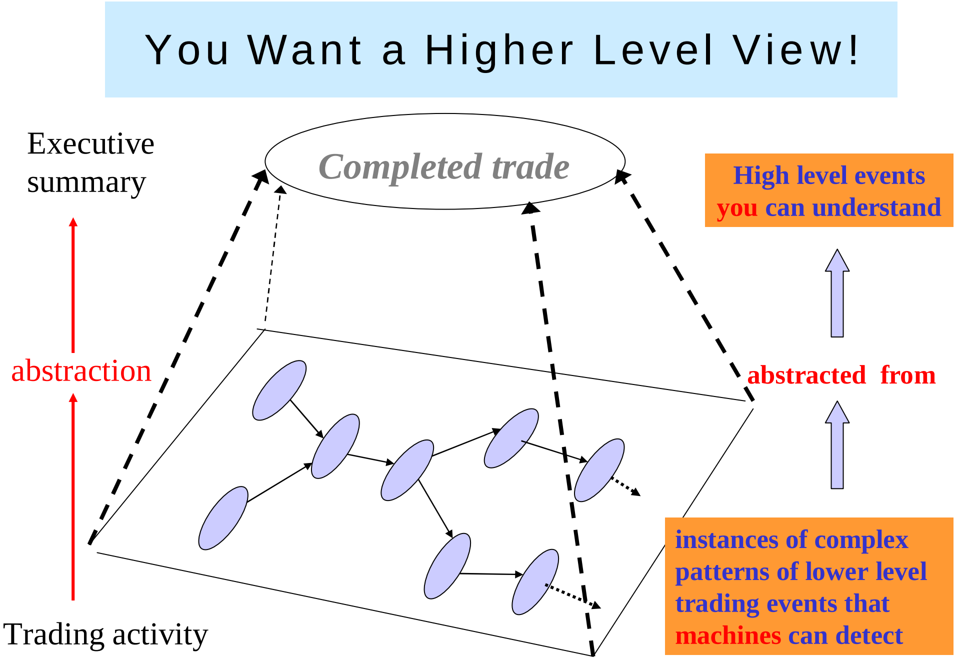
You Want a Higher Level View!
Executive summary
abstraction
Trading activity
Completed trade
High level events you can understand
abstracted from
instances of complex patterns of lower level trading events that machines can detect

Figure 4: Abstracting essential details from instances of complex patterns of events.

The role of event pattern aggregation and abstraction is to give you a view 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 the technology for event pattern aggregation and abstraction is only just beginning to emerge. It truly goes beyond BAM and requires a capability to do complex event processing (CEP). I believe some first product announcements might be appearing this Fall, 2004.

To summarize, three of the steps towards **IT insight** are:

1. precise description of patterns of events,
2. scalable implementations of event pattern matching,
3. event pattern aggregation and abstraction.

It is beginning to happen, and happen quicker than any of us predicted. In 2003 those of us discussing the progress towards developing commercial tools to deliver IT insight based upon event processing thought that it would take the industry up to ten years to tackle all three steps. Now, in 2004, it looks like it might all happen in five years. One day soon we will all be able to buy IT insight to understand exactly what is going on in our enterprises all of the time.

About the Author:

David Luckham has held faculty and invited faculty positions in mathematics, computer science and electrical engineering at eight major universities in Europe and the United States. He was one of the founders of Rational Software Inc. in 1981, supplying both the company's initial software product and the software team that founded the company. He has been an invited lecturer, keynote speaker, panelist, and USA delegate at many international conferences and congresses. Currently, he is Professor Emeritus of Electrical Engineering, Stanford University.

His research and consulting activities in software technology include multi-processing and business process languages, event-driven systems, complex event processing, business activity monitoring, commercial middleware, program verification, systems architecture modelling and simulation, and artificial intelligence (automated deduction and reasoning systems).

He has published four books and over 100 technical papers; two ACM/IEEE Best Paper Awards, several of his papers are now in historical anthologies and book collections. His latest book, The Power of Events, deals with the foundations of complex event processing in distributed enterprise systems.