

Is There a Commercial Need for a Quantum Leap in CEP Technology?¹

Part 2

by
David Luckham

Suppose a quantum leap in commercially available CEP technology were to happen. What would it consist of? I would guess that one of the new components would be something that we haven't seen offered up till now, tools to help users make greater use of abstraction than they do at present. Figure 1 illustrates the creation of abstract events that summarize information contained in patterns of events detected in a company's IT layers. The abstract event might summarize activity in stock trading feeds or sales performance at a number of retail stores. The abstract events would be generated continuously over a sliding time window.

Interestingly, if you look at commercial CEP applications today, you will see abstract events being created all over the place. But nobody pays any attention to the *concept* of abstraction. The immediate problems can be solved without thinking explicitly about how one abstract event relates to another. At present the power of event abstraction is used in unconscious, ad hoc and informal ways. If we are to stop muddling along and apply abstraction more precisely, we will need new tools that are not on the market at the moment. And there is always the question of whether added formality will lead to added pay-off!

There is one particular aid to employing abstraction that I have a special interest in, and I have sometimes questioned whether it has a commercial future at all.

That piece of CEP technology is *computable event hierarchy definitions*.

This is a key component in applying the full power of event abstraction. Its role in CEP applications is to convert the results of the event detection phase, that is detecting business situations of interest, into actionable information for business managers.

¹ © 2008 David C Luckham

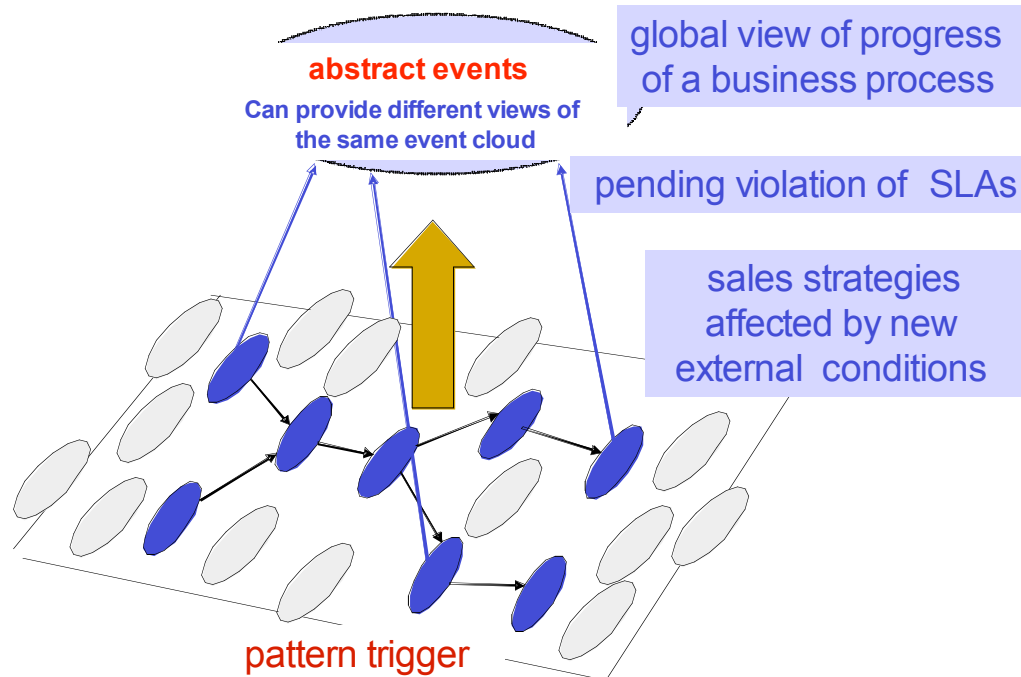


Figure 1: Creating higher level abstract events to summarize patterns of lower level events

I have always believed that at some point in the future, some of the more far reaching commercial CEP applications will be based upon precisely defined hierarchies of events – just like the messaging layers in the networks we all take for granted today. The most likely CEP applications to use event hierarchy definitions will be the kinds of holistic event processing applications that were described in “Holistic Event Processing”.² This is because these kinds of applications will feed different views to such a diverse audience that the inter-relations between the abstractions they use will have to be precisely defined if the users want to communicate with one another. Perhaps those event abstractions might be adopted as standards eventually. But this kind of event processing is a few years off as yet. So, I’m still wondering if we might see an event hierarchy definition in some new application any time soon.

Here’s the kind of example that could arise, an on-line retail website. Let’s suppose it is a very successful website doing business across the world. In that

² <http://complexevents.com/?p=420>

case, all kinds of tools that monitor and measure performance will already be installed. The results will be displayed on graphical dashboards, so called “views” of the system’s KPIs³. How might an event hierarchy be a useful addition in monitoring and managing this business?

First of all, how would we define an event hierarchy for this retail website?

There are many different *levels* of events being created in this website, see Figure 2. Perhaps the most familiar is the *consumer activity* level. That is, events where a consumer logs onto the site, searches the site’s catalog, chooses items, puts them in a shopping cart, discards some of them, adds others, and eventually goes through a checkout process which again comprises several different types of events. It is neither the highest nor the lowest level, but lies somewhere in the middle – or would do so if we had a defined set of event levels!

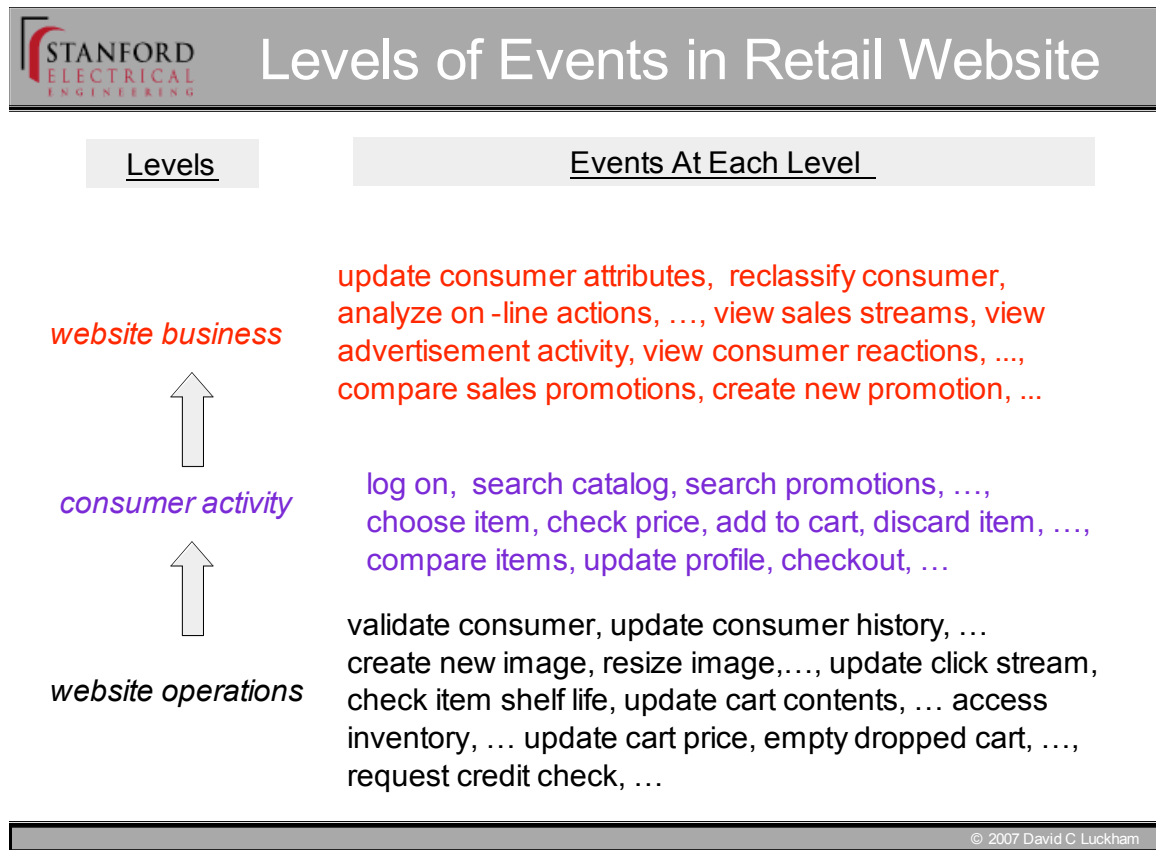


Figure 2: Some of the levels of events in a retail website event hierarchy

³ Key Performance Indicators.

Below the consumer activity level is the *website operating* level. This consists of events involved in the execution of consumer activities, such as consumer validation, database accesses, image rendering, click stream tracking, shopping cart tracking, accounting, credit operations using other websites, and so on. And above the consumer activity level might be the *business level* involving events dealing with inventory maintenance, consumer behavior analysis and classification, advertisement tracking, sales support and promotion, and many other event types we haven't thought of yet. Below these three levels are many other levels, down to network level events. Also there may be more abstract levels above the ones we have mentioned such as a level of events involved in long term planning.

The preceding paragraphs raise a lot of questions. How do we know what "the events" are? And what determines whether an event is "above" or "below" another event, or "at the same level"? The levels and events in Figure 2 might all seem rather ad hoc and open to dispute.

First, the events we choose signify familiar activities that we know happen in the website. We may not think of all the events at once. So the definition of a hierarchy must be flexible and easy to change in the sense that it lets us add new events in the future, or delete others from consideration, or move them between the levels.

Secondly, the levels are intended to capture the notion of abstraction. To classify events into levels we must be able to show that a "higher" event can be defined from a set of "lower" events. If A can be defined from B, C and D, and not conversely, then A is at a higher level than B, C, and D.

Thirdly, we must give computable rules for defining higher level events from sets of lower level events. That is, given examples of B, C and D the rules will compute the corresponding example of A, and conversely. We call them "*abstraction*" rules. They must be reversible so they enable us go up and down the hierarchy.

An example is:

A customer logs on, searches the catalog, chooses an item and purchases that item. Given this sequence of events, the abstraction rules should let us compute the resulting new higher level events such as updates to the customer's sales record and new statistics on the item. Conversely, if we want details about a higher level event such as the appearance of a new sales record, we can apply the rules in reverse to retrieve the event history that resulted in that sales event. This is often called "drill down". A computable hierarchy based upon abstraction rules allows us to go in both directions, up and down, in analyzing events.

One point that must be emphasized is that a computable event hierarchy enables events at every level to be computed continuously in real time as the events at

the level of website activity are created by customer activities. Figure 3 tries to illustrate a real time flow of events upwards through the levels of an abstraction hierarchy, driven by events being created at the website level.

How would a hierarchy like the one in the figures be used?

Perhaps the most obvious answer is to deliver *focused real time information*. Various people involved in the management of the website in would be interested in different types of events and certainly not in all the events if they are engaged in real time management. They need to be able to specify what they want to “view” – or have a consultant do it for them! A computable hierarchy enables viewers to define just the sets of events they want to see⁴, and not be bothered with a lot of events they’re not interested in.

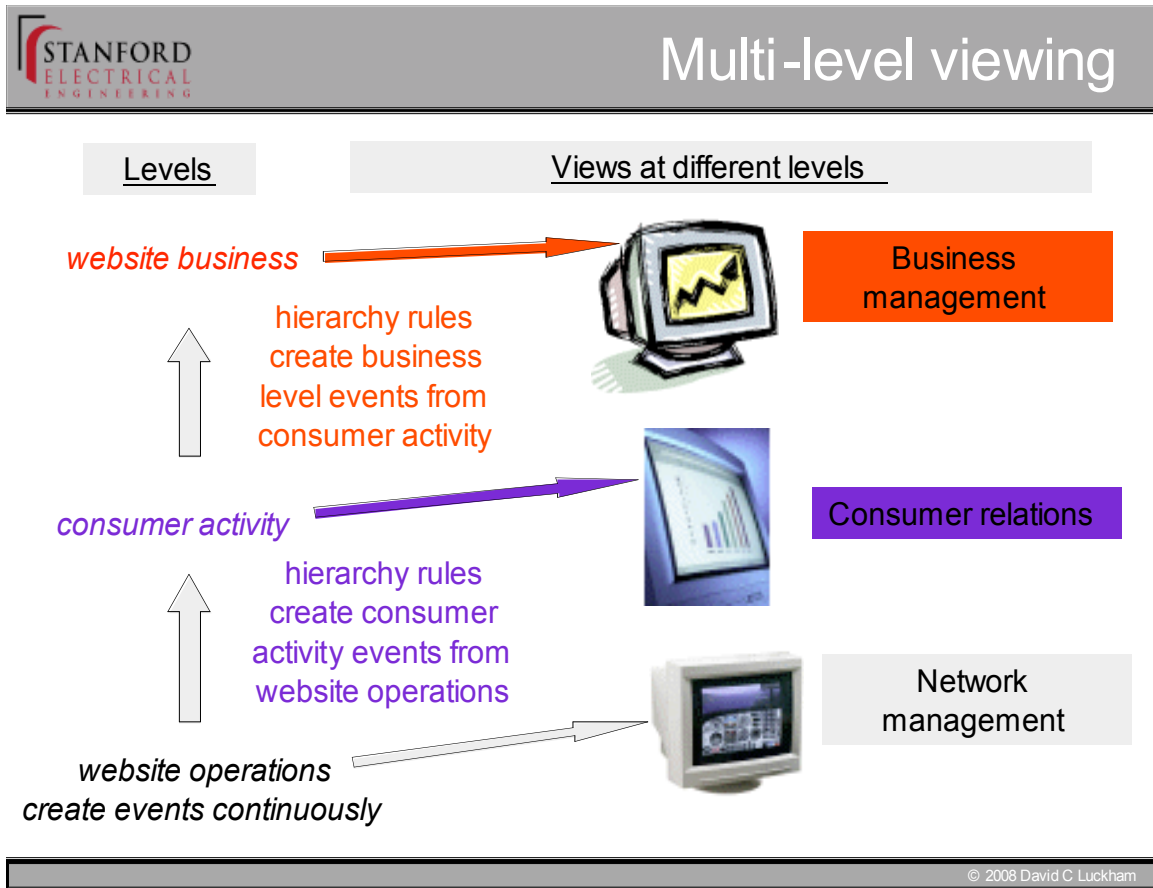


Figure 3: real-time viewing of retail website events for different management roles

⁴ Called a “view”.

For example, a business manager would be interested in viewing the sales streams for particular items, or how a consumer group is contributing to those sales, or how the sales of new items are varying as advertisements are added to the website. Also the effect of a new item on the sales of older items could be useful. An advertising manager will want to see different types of events such as the on-line viewing of various advertisements as a real time function of the time window in which a page is being viewed, how that activity varies with the advertisement's position on the page, and the subsequent behavior of a viewing customer. A network manager on the other hand would be interested in a totally different space of events at the lower network levels. But when complaints reach him from the business level, then he might want to know how the higher level events being viewed by the business and advertising managers are related to the network level. That requires drill down through the various levels.

So, if we have a computable hierarchy, the views that the different managers want can be computed from the appropriate levels of events in real time and fed to them. At any time they can request and get other views composed from different events. At some point we may expect that real time management decisions based on higher level viewing will be automated. The business may eventually be run by a bunch of robot managers with very high level human overseers checking the bot managers activities! Perhaps that's a little over the top, but its not impossible! In fact its already happening in some areas like stock trading where the number of human traders can be reduced by the use of algorithmic trading engines.

A computable hierarchy definition has to be *modifiable* so that the events at any level and the rules for constructing them from lower level events can be changed easily. Indeed, over a hierarchy's application lifetime, we may need to change the levels themselves, delete some levels, introduce new ones, and insert new levels between existing ones.

A discussion of methods of defining modifiable computable hierarchies is the topic of a forthcoming article. But for now, we end with a question:

Can you see a market for this kind of CEP application yet?