

# Complex Events in Enterprise IT

**NOW Magazine:** Let's start by defining the difference between an event-driven SOA and a classic SOA.

**Dr. David Luckham:** SOA, which is a very good idea by the way, consists of two things basically—services and users of the services—and communications between the two. So the two concepts are first, modules that are groups of services that are logically related in some way, for example, online banking services or credit rating services and so on; and second, communications that let the users access the services. The emphasis is on grouping services in modules. As you know these ideas of modular design have been around for a long time in object-oriented programming.

**NOW:** Do you have a picture of a classic SOA?

**Luckham:** Yes, indeed. (Editor's note: refer to Figure 1.) This figure is a paradigm depiction of an SOA. It's actually using the analogy of a hardware motherboard, which is a very good example of SOA indeed. And it shows an e-marketplace actually, so you've got the services of the e-market, and then you've got transaction settlement

services such as PayPal, and then you've got users, buyers, and sellers working through the marketplace.

Now in traditional SOA the communication between the users and the services is by remote procedure call, RPC. An everyday example of RPC is a telephone call, that's typical. As we all know it can be very inefficient and wasteful. Typically you phone up, you get a call center, and the automated multiple choice defense first line, and then you might get a human being who may be able to answer your question, or perhaps has to put you on hold, or transfer you elsewhere. It can be an endless runaround and waste of time.

**NOW:** So how does an event-driven SOA differ from the classic SOA view?

**Luckham:** Very simple, it's a change in the communication between users and services. Instead of using RPC, event-driven SOA uses events. As a result, of course, the services have to be able to react to events. (Editor's note: refer to Figure 2.) In this figure, you won't see very much difference, but the upside is that event-driven communication is much more versatile and

Today's business environment **DEMANDS AGILITY**. Event-driven SOA, or a new idea called complex event processing, offers a new way to provide business insight that can help IT contribute toward bottom-line operational excellence.

Dr. David Luckham, Professor Emeritus at Stanford University, is the world's leading expert in this emerging field. His latest book, "The Power of Events," was published by Addison-Wesley in 2002. Dr. Luckham was recently interviewed by *NOW Magazine* about his favorite topic. This is part one of that interview. The final installment will appear in the next issue.



David Luckham

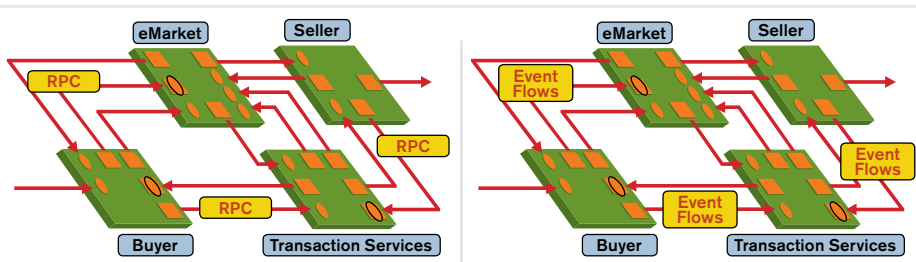


Figure 1

Figure 2

efficient than RPC. It allows all the actors, the users and the services to multitask.

Instead of a phone call you send an event over the Internet, for example. This request event for service is distributed to a set of possible servers. The servers collaborate in answering your request, and there's a guaranteed time limit in which you're going to get a response. You're no longer hanging on the end of a phone, you're free to do something else. Just imagine—you can send a message to your bank asking what's happened to my funds transfer, and have a guarantee that you'll get an answer within, say two hours, and there'll be an explicit event trail explaining the answer. That's a much more efficient way to organize service-oriented architectures.

**NOW: So what do you think is the most compelling use case for event-driven SOA?**

**Luckham:** Well, event-driven information technology has turned out to be what's referred to now as a disruptive technology. Network messaging such as that provided by the Internet has revolutionized the way that modern enterprises are organized and do business. This has changed society. The way the Internet works is a paradigm for event-driven SOA.

So if you think about your own enterprise, you'll probably understand that it is event-driven, too. Pretty much any modern enterprise's IT operations today are organized as an event-driven SOA, especially if they use the Internet or commercial networks such as SWIFT. This includes a whole gamut of areas—from financial services and stock markets, cellular service providers, through automated supply chains, and even the SCADA control systems that control our electricity grids, dams, and power stations. To be honest with you, I've got a difficult time thinking of anything that isn't event-driven these days; it's not a matter of a use case, it's a matter of the fact that it's a way of life.

**NOW: So if it's a way of life, how come we haven't been doing this...**

**Luckham:** Well, people are not conscious of it, it's as simple as that. When you start thinking about it and you say well, I'd like to organize my IT a bit more efficiently, then you've got to come to terms with the actual technology. Events have been around since the very early operating systems.

Most of the early operating systems were event-driven just in the way they organized

their work stacks. Messaging networking in the 1970's was all event-driven, it's just bubbled up through the layers of the enterprise to the business level now.

**NOW: In your book "The Power of Events," you wrote about complex event processing, and coined the acronym CEP for it. How does CEP fit into the event-driven SOA picture?**

**Luckham:** Now we get into the issue of how to manage the event-driven SOA. That's where CEP comes in. We've got all this event activity going on in the modern enterprise, so lots of questions arising in real time. How are my business processes doing? What are my competitors doing with their prices? Are my customers happy? What's happening to those hazardous materials I shipped? Is our market trading in compliance or will the government come after us? And what are the crooks out there doing to me?

There's a good example that addresses these questions from a national retailer with a highly automated IT support and network monitoring system. They had every network monitoring tool known to mankind and then some. What happened was that the pricing process went berserk right around Christmastime one year. That's a process—one of a number of processes in the business processing system, which updates prices to 600 stores every 24 hours. Pricing errors at Christmas were not caught for three days and resulted in \$3 million worth of revenue loss.

Top-level management was really upset by this error—after all they're in a cutthroat business with razor slim margins—so they told the IT department to fix it. With its large IT department, well, we all know what happens; several different

initiatives were started. These varied from formal modeling of the process system through to event content validation, which was exactly what they needed. But for two years nothing came to completion.

Eventually, top-level management lost patience with the IT department and outsourced the whole IT department, and 650 employees lost their jobs.

**NOW: This sounds like a catastrophic example.**

**Luckham:** And the last chapter in this is still to be written, because the smart money says that the outsourcing deals like this in the retail business don't work. And it's important to note that this happened in an enterprise that has every network monitoring tool known, as well as some homegrown ones. They could tell you what their latencies are, and what their queue lengths are anytime, anywhere. But they were totally unaware of the business content of the events in their business processes!

**NOW: How does CEP differ from a tool for business intelligence, data warehousing, or analytics?**

**Luckham:** Well, one of the pressing needs in the modern event-driven enterprise is real-time operations. What happened in this case of the retailer was that they needed rather simple event-processing, business-event content validation, and that would have been very easy to introduce. These kinds of errors need to be caught as they happen, not after they're put in a database, and they may be analyzed after close of business. The main lesson here is that this kind of error can be caught a lot earlier. You should save a lot of losses with business-event content validation. ●