

## Event-Driven SOA: A Better Way to SOA

Experts agree that service-oriented architecture (SOA) is the right approach for new enterprise architecture development. And chances are that your company has already started on the path towards SOA. But are you implementing the right SOA for the long term? You may not be if your SOA implementation doesn't support event processing.

**TABLE OF CONTENTS**

1	EVENT-DRIVEN SOA : A FOUNDATION FOR THE FUTURE .....	3
2	SOA AND EVENT PROCESSING – PUTTING IT ALL TOGETHER .....	8
3	CONCLUSION .....	9
4	ABOUT TIBCO .....	10

## 1. Event-Driven SOA : A Foundation for the Future

SOA has gained acceptance as a way to unlock the business value of a company's IT infrastructure. SOA permits architects to "break down" monolithic applications into discrete services that can be reused in a variety of applications. By exposing information and transactions as services in an abstracted, standards-based way, an SOA provides a foundation for new classes of business applications – "composite" applications that are assembled from a mix of existing and new service logic as opposed to being built entirely from scratch. It also enables organizations to deploy new business services faster, at lower cost.

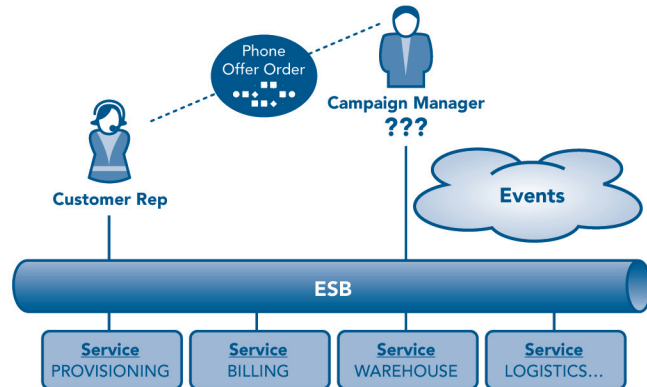
However, an SOA infrastructure doesn't address all the capabilities needed to respond to the dynamics of real-time business. Consider the following scenario, one that probably plays out with slight differences in a variety of industries.

A telecommunications firm targets a particular consumer to make the switch to their service, promising a new model of phone and rate plan. The promotion proves a resounding success, and the particular customer gets swayed to change services by a telephone customer service representative.

The core business operation, in this example, is a triumph of the SOA application model. Services exposed trends in a data warehouse about which prospects would be most susceptible to a change in service. Services made the functional integration possible that inserted the customer name, telephone number, and other information into a CRM/call center application. Another service accessed a service contract and processed it with a billing system. And finally, an order process was invoked that used services to invoke the inventory/warehouse actions and logistics processes to send the new phone and rate plan contract to the customer.

However, this story is not over yet. It turns out this promotion is so successful that the particular brand of phone is being sold off the shelves and inventories are running low. By the time this particular order goes through, the phone is gone from the warehouses. The phone isn't shipped. The order process gets "stuck" and sends an alarm that the shipment hasn't been completed. But it's too late. By the time the customer service representative can contact the customer about the delay or about ordering a different phone, the customer has already lost faith.

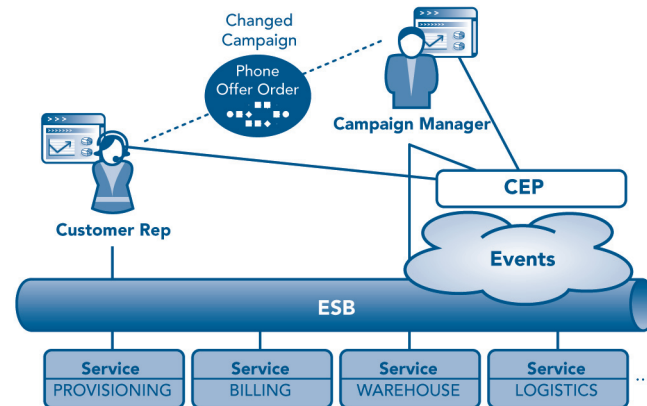
**Figure 1.**  
An SOA sets up a successful promotion, but something is missing.



He's decided to take up an offer from his original provider on a new service contract and phone.

What just happened? Was this failure related to SOA? No, it's an example of an enterprise that can't sense and rapidly respond to fast moving business events and situations. The events that would indicate a situation such as "phone being out of stock" were lost in an "event cloud." This example points to the need for SOA to

**Figure 2.**  
Adding complex event processing (CEP) allows the business to dynamically respond to time-sensitive business situations.



### A SHORT COURSE IN SOA

In order to understand the business pitfall described above, it is necessary to understand what an SOA is and is not. An SOA is an abstract distributed architecture that is open and available to anyone or any application on the network with proper security authorizations. Data and functionality are exposed as reusable services described using a standardsbased format (WSDL, SOAP, XML) and transport (HTTP, JMS) that can be used by other applications. This includes, if privacy concerns are properly addressed, items such as customer phone numbers and addresses or buying trends by categories of users. Perhaps one of the biggest

advantages is that this information transaction is shared out of its proprietary silo – be it an enterprise resource planning (ERP) system, a customer database, or a desktop application.

### **REQUEST / REPLY**

An SOA based on web services (WS) uses a classic request/reply type of message exchange pattern as a form of communication. There is a service and consumers of the service who have the right to access the service. The consumer requests access and is granted the information or processing capability of that particular service, at that particular time. The relationship between the service and the consumer is generally loosely coupled yet synchronous due to the nature of request/reply. It is a passive, demand-based infrastructure.

### **ORCHESTRATION**

For most WS applications companies generally use service orchestration. WS orchestration is now a core capability of most SOA toolsets and architectures. Service orchestration is the graphical wiring of a set of services to automate system-to-system information processing. For example, if the business objective is to get customer information into a customer relationship management (CRM) system, a process may need to access a data warehouse with customer profile information, billing information, and address/ phone type information and insert it into the CRM application. There may be some data transformation and some simple rules used for processing. Generally, this process is kicked off by an event, which then kicks off a string of request/reply interactions. It's event based, but based on explicit procedures, in that an event kicks off a prescribed hardcoded procedure, rather than declarative where a set of business goals is determined and the process goes out and works to achieve the goal based on a set of facts.

The business events that trigger an orchestrated business process, or invoke a web service, are generally not predictable and are not requested. An event is triggered, in some cases, by a random business activity. In many cases, these classes of events are recorded within an application database.

### **ESB: DISTRIBUTING EVENTS IN THE ENTERPRISE**

An enterprise service bus (ESB) is an accepted solution to the problem of a rigidly orchestrated SOA infrastructure. An ESB is generally the way service requests move on an SOA as messages. An ESB guarantees that the message, whatever it may be (the request to see customer data, the actual data, etc...), gets delivered from consumer to service or vice versa. It also provides transaction-grade delivery

of messages to/from software components that are not web services – for example, Java objects.

The publish/subscribe and notification services in many messaging products ensure that the ESB can handle events, even unexpected events, if they are applied properly. With proper planning, an ESB can be a common foundational element for SOA and event processing.

Users need to process events and respond and adapt to situations. In most cases, this does not happen. The result is that businesses cannot detect business events and situations, including rapidly occurring opportunities or threats. This is an emerging area in business optimization often referred to as complex event processing (CEP). We will discuss CEP in more detail later in this paper.

## WHY EVENTS MATTER

### ***What's a Business Event?***

A business event is any notable business activity that affects your business. In an eventdriven SOA, an event is created and recorded in your software infrastructure, either directly or indirectly, and has some type of influence over your business.

Depending on the size of your business, there are hundreds to millions of notable events that occur on a daily basis. A service representative inputs a new address for a customer; a network device emits a warning message that the temperature is too high; a snowstorm occurs in one region of the country causing employees to be late; a pallet of goods passes through an RFID reader in another portion of the country. These are all business events, and somehow these events affect the operations of a company. Some business events have a negative impact on a business, some have a positive impact, some will be an opportunity, and others may be unremarkable.

All businesses are event driven. In fact, the entire world is event driven!

Consider the pallet of goods that passes through an RFID reader in a warehouse on the East Coast. Most IT managers will have a system in place that sends that information to an inventory tracking application that records exactly what quantity of that product is now available in that warehouse. The systems may also send information to a logistics application to confirm the order was received. It may even

match that product description to the original purchase order to confirm what was ordered was delivered. These are all business applications that are either enabled or made much easier by an SOA. So far so good, but that's not the whole story.

Taken in context, the shipment of this pallet of products is actually bad news, since the product is piling up in a warehouse on the East Coast, while sales are booming in the Midwest due to an aggressive set of promotions in that region. Responding to this situation over the phone may be problematic since key employees are not in the office due to a snowstorm in one region of the country.

This is an example of events blindsiding a business in retail. Every industry is driven by events. Businesses spend most of their time dealing with event-driven processes. Misdirected resources, upset customers, fraud, theft, sudden increases in sales, and every other kind of situation imaginable demonstrate how events affect businesses.

#### ***What Is Complex Event Processing?***

"Complex event processing is a new technology for extracting information from messagebased systems."<sup>1</sup>

Many observers believe that the capability to process complex events is required for businesses to respond to rapidly changing business scenarios and situations.

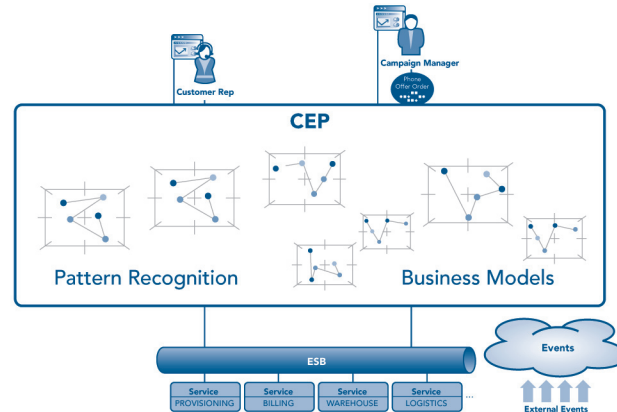
A complex event processing (CEP) system enables organizations to process distributed business events and identify opportunities or threats. Business events may be tracked individually, such as a stream of stock trades, or correlated with other events, producing derived or "complex" events often called "situations."

Making sense of these business events requires event processing technology and domain expertise. In general, users must understand and create business models that model the dependencies and relationships within an event stream between event objects and in the context of historical information.

Events and trends related to business events need to be mapped and correlated against historical business models. Mapping events against business models forms the basis of recognizing patterns that may be occurring. Rules play an important part in the development of business models as well as in the correlation between business events and business models. For businesses, the important part of event

<sup>1</sup> David C. Luckham and Brian Frasca, "Complex Event Processing in Distributed Systems"

**Figure 3.**  
A CEP system enables organizations to process events for trends and patterns and predict the impact of identified situations.



processing is the ability to rapidly respond to negative situations or take advantage of emerging opportunities.

To summarize, event processing, and in particular CEP, provides the capability for businesses to track, trace, and correlate events; to process events for trends and patterns; and, finally, to predict the impact of identified situations. Rapid action can then be taken to prevent or minimize damage from threats to businesses – such as a customer not receiving a product on time resulting in a cancelled order. The same is true on the upside. A positive situation can be capitalized upon rapidly and effectively –such as an emerging trading pattern or customer up-sell opportunity. Situations can also be analyzed to improve the underlying business processes and applications that are the “DNA” of business operations.

## 2. SOA and Event Processing – Putting It All Together

SOA and event processing are both required for an optimized business and, when combined, can create extreme value to business operations. SOA and event processing also help expose business information that is otherwise locked in application silos and databases.

Services and SOA are used to help unlock information and IT processes so they are available to be leveraged in other applications and to build new composite applications. Events are the nervous system of the enterprise and, when leveraged by an SOA, can disseminate business events to all authorized parties in the

extended enterprise and ecosystem of today's modern businesses. CEP extracts and creates value by identifying threats and opportunities from distributed enterprise events.

On a functional level, SOA and events need each other. SOA can profit from events when it comes time to build an actual event-driven application, large or small. For example, an event can trigger the launching of a service or string of services to solve a business problem. Modeling event-processing as services is also important because events and CEP applications benefit from the business objectives of an SOA. In fact, services can be used in an event-driven application at practically every functional step in the architecture. In a similar manner, an SOA powered with events facilitates agile, adaptive business processes that can respond to ever changing opportunities and threats. The outcome of a business service or orchestration of services is often another business event.

### 3. Conclusion

To return to the original question: Are you implementing the right SOA for the long term? If you're implementing an ESB that supports event processing and messaging for any eventdriven communication, the answer is yes. If you're not implementing an ESB – for example, if you're implementing an SOA using application servers, integration brokers, and code – the answer is no. You need an ESB.

This brings up the next question: When do you need CEP? The best strategy is to ask that question now and identify the first CEP projects in the same way you identified your first projects for SOA. Think about any area of the business where tracking the state of a process or handling unexpected events adds significant business value.



## 4. About TIBCO

**TIBCO Software Inc.** (NASDAQ: TIBX) is a provider of infrastructure software for companies to use on-premise or as part of cloud computing environments. Whether it's optimizing claims, processing trades, cross-selling products based on real-time customer behavior, or averting a crisis before it happens, TIBCO provides companies the two-second advantage™ – the ability to capture the right information at the right time and act on it preemptively for a competitive advantage. More than 4,000 customers worldwide rely on TIBCO to manage information, decisions, processes and applications in real time. Learn more at [www.tibco.com](http://www.tibco.com)



Global Headquarters  
3303 Hillview Avenue  
Palo Alto, CA 94304

**Tel:** +1 650-846-1000  
+1 800-420-8450  
**Fax:** +1 650-846-1005

[www.tibco.com](http://www.tibco.com)