

Title: **Are You Being Served?**

Date: **1 June 2007**

Service-oriented architecture still holds sway, but not in areas where speed is the key. By *Emily Fraser*

The unending quest for low latency in electronic trading has pushed the bright promise of service-oriented architecture (SOA) out of much of the front office, where stream processing architectures now dominate.

SOA is a design philosophy, as opposed to a particular type of technology, which can save firms time and money by letting them reuse software. It offers great flexibility by allowing any application to interact with any other application and borrow pieces of code.

For example, a firm might develop a complex yield calculator for a front-office application, and then discover that the same tool would be very useful for a back-office accounting application. The back-office team does not need any of the other features of the front-office application, but would greatly benefit from borrowing the calculator component. In an SOA model, the back-office application would provide a link to the calculator, bring back results and use them in the accounting platform.

SOA is therefore well suited to middle- and back-office functions, says Steve Grill, global head of corporate, investment banking and markets (CIBM) IT architecture at HSBC. "It is very good in terms of addressing settlement and clearance and working across legal entities and operating units," he says.

However, when it comes to the front office and applications, which need streaming data and real-time updates, SOA does not deliver, says Grill. "SOA does not perform well at a sub-millisecond need," he says. In order to allow for flow processing and to support algorithmic trading and direct market access, firms need the capability to do very flexible, high-performance computing, based upon either market data or various amounts of transactional-type data, he says.

Sending a link, either via messages or Web services, to another application and waiting for a response can take longer than is tolerable in high-volume, low-latency front-office trading and market data applications.

XML, the programming language associated with SOA, is not well suited to low-latency applications either. Every message sent using XML is wrapped in an assortment of brackets and slashes to the extent that "the markup dramatically dwarfs the size of the actual content it is trying to carry," says Buzz Moschetti, chief architecture officer at Bear Stearns.

### **Caught in the Slip Stream**

Event stream processing (ESP), also known as complex event processing (CEP), more accurately reflects the needs of the front office as they relate to pricing, order management and such, says Grill. HSBC views technology implementation as a two-part solution, he says. HSBC's CIBM division is developing two separate architectural backbones: one with SOA for the back and middle offices, and one with ESP for the front office.

Examples of these front-office backbones include Reuters Market Data System (RMDS) and 29West messaging solutions, which achieve sub-millisecond latency. "The implication is that we have two separate messaging backbones but one standard set of business processes where appropriate, so we can support our event streams or our service-based technologies using our common process models," he says.

Moschetti says that a front-office trading system can contain elements of both types of architectures. SOA and enterprise service buses (ESBs), the technology SOA is built around, are not ideal in the more exciting front-office applications, but other parts of the trading environment can absolutely take advantage of SOA, he says.

"The more glamorous parts of the trading system typically involve super high-performance networks, data-distribution models, and technologies such as CEP that are 'in line' with information coming from and going to the markets," says Moschetti. "But other aspects of the system, such as entitlements, configuration, loading of baskets, lookups on reference data, and so on, can use what the industry conceives as SOA—that is, Web services, XML over message busses, and so on," he adds.

CEP is an emerging technology rather than a design philosophy, which has started to gain traction in the financial services arena over the last couple of years. "Event-stream processing is already becoming the standard application for driving algorithmic trading, with firms like Progress Apama, StreamBase, Vhayu Technologies and others having had a good bit of success," says Matt Nelson, senior analyst at consultancy TowerGroup. Becoming an integral part of algorithmic and strategy trading, ESP is starting to spread toward other areas like data monitoring, compliance, transaction cost analysis (TCA), risk management, proprietary data derivations, market-making and more.

According to a hype-cycle report on emerging technologies, released last summer from researcher Gartner, event-driven architecture is on the upward slope, expected to reach mainstream maturity in five or 10 years. The ESP model is therefore still maturing. Grill says he foresees that middle-office services will migrate toward ESPs and data fabrics as they evolve, and that ESP and SOA/ESB technologies will come together. "ESBs are strong in areas where ESPs are weak, and vice versa," he says. "So, they will either merge or evolve to replace each other," he says. Standard process and information models, such as the FIX protocol, are changing the integration landscape, and hence the requirement of these tools, he says. "Only the agile and strong will evolve and survive," he adds.

Others see SOA and ESP architectures living alongside each other in harmony. "The business use cases are different," says Brenda Michelson, principal consultant and analyst with Elemental Links, who specializes in SOA and event processing. Not everybody in an enterprise needs that real-time capability, and not everyone has high volumes of data to stream and analyze, she says. For this reason, she says, "Event processing is very important but it will not be as pervasive as SOA."

## **The SOA Strategy**

Executives at SunGard Consulting Services (SCS) have seen a lot of growth in SOA for enterprise-level services. "A lot of our clients are moving in that direction," says Alex Raitt, principal at SCS and formerly a director of consulting firm Finetix, which was recently acquired by SunGard. "A lot of front offices make use of those enterprise-level services, but you don't see much birth and incubation of SOA inside front-office environments," he adds.

SOA can also be suitable for certain slower-performance front-office applications where instant response is not critical, says Bill Hobbib, vice president of marketing at CEP software vendor StreamBase. Conventional, non-automated trading could make good use of SOA, he says. "Customer relationship management, call center and billing applications might all feature some sort of SOA in a front-office situation where there is no need for an instantaneous response," he says. A dashboard that monitors business activity and is checked every half an hour or so might feature SOA as well, he adds.

A secondary benefit of SOA, beyond offering flexibility, is that it can "help extend the lifespan of legacy systems," says TowerGroup's Nelson. In many cases, firms have "very old, very clunky technology" in the back office, he says. The legacy processing engines may do certain core processing really well, but may lack certain features. With SOA, firms can build code that integrates with newer applications to borrow their updated functionality. Keeping legacy systems alive even a bit longer has major cost benefits, he says. The cost for replacing these systems altogether could reach into seven or eight figures, so the longer the firm can continue paying maintenance fees on the old systems without replacing them, the better.

Feverous merger-and-acquisition activity in the financial services sector is another compelling driver for the adoption of SOA. "Financial services firms in general are dealing with a lot of consolidation, and an SOA model provides an easy way to integrate various applications," says Nelson.

However, SOA does have to be managed properly in order to realize these advantages. "One criticism that has been leveled against SOA is that if it is not implemented correctly, firms end up with a component architecture rather than a service-oriented architecture," says SunGard's Raitt. With SOA, "you partition up your problem into separate pieces or into components and if you do the job right, those pieces deliver business functionality," he says. "If you don't do a good job, they end up as components just providing functions."