

# Getting on Board the Enterprise Service Bus

*Sixto Ortiz Jr.*

**O**ne of the biggest challenges facing businesses today is getting their diverse applications, often built on different platforms, to work together when necessary. For example, a purchase order generated via a company's e-commerce application might need to query a customer-loyalty database to determine whether the buyer has done enough business in the past to merit special pricing or handling.

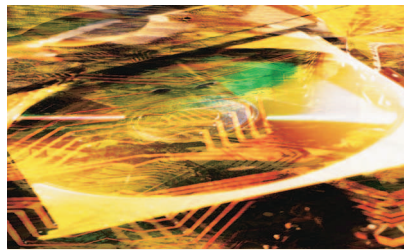
To accomplish this, companies must implement effective application integration flexibly, quickly, and easily.

Businesses initially turned to manual integration and then enterprise application integration (EAI) but subsequently have focused on service-oriented architectures. To leverage SOAs' benefits effectively, companies are beginning to use an approach known as the enterprise service bus, offered by a growing number of large and small vendors.

ESB is the middleware glue that holds an SOA together and enables communication between Web-based enterprise applications.

"Fundamentally," said Larry Fulton, senior analyst with Forrester Research, "ESB provides the connectivity between service requestors and service providers, physically conveying the requests and the responses."

Proponents say ESB offers advantages over earlier application-integration approaches, such as ease and



flexibility of use and ready scalability to larger deployments.

However, ESB also faces challenges, such as implementation costs and complex migration and management.

## DRIVING FORCES

Enterprises often must provide employees, suppliers, customers, and partners with on-demand services and information culled from various data sources. To do this, they must be able to combine the capabilities of their many applications, including some legacy programs.

This type of integration became an important issue in the mid-to-late 1980s, said Fulton, when businesses began deploying software that helped management by combining information from disparate data-generating applications.

Initially, companies had to manually integrate applications so that they could work together. However, this required considerable time and expense, and it worked only for the applications that were manually linked. Adding additional applications required more effort.

## EAI

According to Fulton, organizations began using EAI in the late 1980s.

EAI systems can use a hub-and-spoke architecture in which applications send messages via adapters to centrally located message and integration brokers, noted Wayne Kernochan, senior IT analyst for market research firm Illuminata. The integration broker is the system's "brains," handling tasks such as data transformation and message routing.

In a bus-like EAI architecture, disparate source applications send messages via a central "pipe" to receiving applications. The system contains software adapters and integration engines at all nodes, thereby distributing the intelligence.

EAI enables automatic, machine-to-machine communications. However, Fulton explained, it works via point-to-point interfaces, which organizations must define and build one at a time, between applications. As companies use more applications to provide additional services, the amount of integration work required mushrooms, and managing the system becomes increasingly difficult.

In addition, the integrated application sets are not reusable for other purposes.

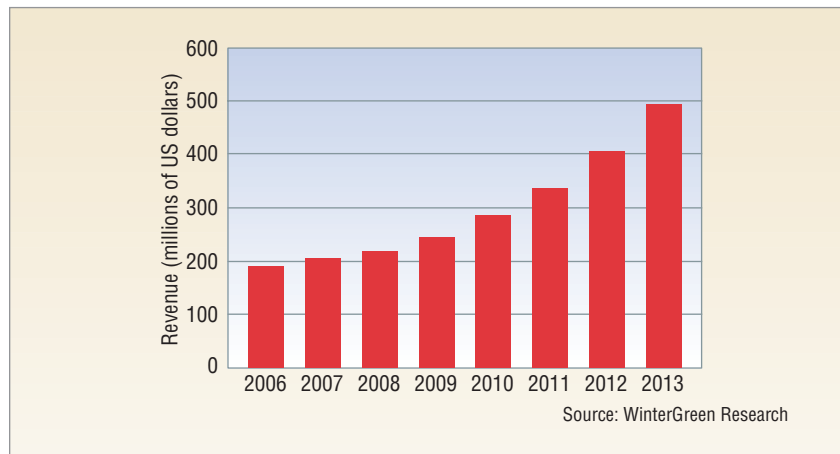
## SOA

Companies began implementing SOAs, which use EAI technology for integration, in the mid-to-late 1990s.

An SOA enables flexible connectivity and communication among applications by representing each as a service with an interface that lets them communicate readily with one another.

Like EAI, SOA can use Web services to allow standards-based applications to interact over the Internet. This enables machine-to-machine communication, speeding the application-interaction process and eliminating the need for human intervention.

Standardized Web-services interfaces include XML for data descrip-



**Figure 1. The worldwide enterprise-service-bus market will grow steadily, doubling between this year and 2012, according to WinterGreen Research.**

tion; HTTP for message transfer; the Simple Object Access Protocol (SOAP) for message exchange; the Web Services Description Language (WSDL) for service description; and the Universal Description, Discovery, and Integration (UDDI) protocol for publishing and discovering services, said Jim Rivera, Cape Clear Software's vice president of product management.

SOAs can also work with proprietary interfaces to communicate with legacy systems.

Unlike EAI, SOA lets companies reuse the services they create—via the integration of multiple applications—in more than one business process without additional programming.

In essence, SOA sets up a repository of these services, which business-process developers can discover via Web services protocols.

This reusability is what makes SOA so useful for helping companies accomplish their business goals and needs, stated Steve Craggs, the president of Saint Consulting and the vice chair of the Integration Consortium ([www.integrationconsortium.org](http://www.integrationconsortium.org)) industry group.

### BOARDING THE BUS

Initially, SOAs were configured in a hub-and-spoke architecture, working via a central broker that provides the system's intelligence.

However, as an organization grows, the addition of more applications and services could overwhelm a single messaging broker. This led to development of ESB's bus-like architecture.

ESB software vendors include Cape Clear Software, Fiorano Software, Progress Software, and PolarLake. Some large vendors—such as IBM and Oracle—are also entering the marketplace. Open source offerings include Bostech's ChainBuilder ESB and LogicBlaze's FUSE.

### The 411 on the ESB

The ESB, which runs via software that can be loaded onto desktop computers or servers, provides an SOA's connectivity infrastructure to enable communications between applications running on different platforms, written in different programming languages, and using different programming models, said Craggs.

The ESB functions like a software version of a PC's traditional hardware bus, moving data along a common pipe between applications, according to Illuminata's Kernochan. The integration engines are distributed throughout this architecture, at the computers that host the ESB software, rather than located in a central broker.

An ESB lets one application send data to another by first accessing it via a locator and then sending it a message.

The ESB handles the transformation of data formats; routing; message acceptance, processing, and routing; and the sending of multiple messages at the same time.

**Elements.** At its most basic, an ESB architecture consists of three elements, according to Craggs.

Mediation services automatically handle functions—like application mapping and intelligent routing—necessary to get Web services—such as an e-commerce application and a customer-loyalty database—to work together to perform business processes.

The services also allow information transfer between applications by taking data in the format of a sending application and converting it to one compatible with a receiving application.

These function via basic adapters that work with standard environments such as Web services and the Java Connector Architecture. Additional adapters connect to other environments, such as SAP's IDoc (intermediate document) data structure. This is important for legacy software that wasn't built to communicate automatically with other applications.

Adapters work via preprogrammed data-format-mapping information. Without them, developers would have to manually code or use custom development tools to create connections to applications, said Kevin Clugage, product director for Oracle Fusion Middleware. "That would add unnecessary complexity," he noted.

Messaging protocols enable communication between service requesters and providers. Progress Software's Sonic ESB, for example, supports about 200 open and proprietary messaging protocols, including Web services, the Java Message Service, e-mail, FTP, and IBM's Customer Information Control System, said Hub Vandervoort, chief technology officer of Progress' Enterprise Infrastructure Division.

**Capabilities.** ESBs' programming model provides a rich abstraction of both the application and the interface. Thus, a business-process architect need not worry about the coding necessary to connect applications or resources but instead can just utilize an automated ESB service, explained Christopher Vavra, product manager for IBM's WebSphere ESB.

ESBs can be centrally managed and thus can perform integration throughout an organization's infrastructure.

Fiorano CEO Atul Saini noted that ESBs offer additional important capabilities such as making sure a message is formatted correctly, finding the most resource-efficient way to perform tasks, providing redundancy in case of failure, and logging events or errors.

For security, ESBs typically utilize authorization and authentication to limit application access to those with permission.

ESBs also conduct some of the auditing and monitoring necessary to ensure applications perform as specified by service-level agreements between customers and vendors.

### Advantages

When ESB integrates applications, it configures the services for reuse in other situations. Over time, a system can end up with many such tools, making them useful in numerous situations.

Via ESB, SOA systems can perform application integration once and then automatically reuse the integration services whenever necessary. This keeps business-process developers from having to manually recode the services each time integration takes place, as is the case with EAI.

By providing a standardized, automated way for programs to exchange information, ESB makes expanding existing services easier than with EAI, Craggs noted. The architecture's decentralization lets companies add applications to a service incrementally, when and where needed.

Because an ESB has no single, central messaging broker but instead distributes messaging services around a network, it can better accommodate growth in an organization's activities and eliminates the potential bottleneck or point of failure that a single broker represents, Craggs explained.

### ROADBLOCKS FOR THE BUS

Because ESB does work previously performed by humans, the host system may need additional or more powerful hardware to handle the extra processing.

Learning to use an ESB optimally takes time, so organizations don't always realize a return on investment for the first few projects.

According to Forrester's Fulton, some potential users don't adopt ESB because of confusion about whether they should purchase similar Business Process Management products instead.

And some companies are delaying adoption because they must decide between the numerous ESB products hitting the market now.

Moreover, migrating from today's EAI-based integration infrastructures to a more SOA-centric approach can be challenging. For example, large-scale ESB implementations can be time-consuming and expensive, and they can entail complex management and high consultancy costs, said Rob Hailstone, software-infrastructure practice director for the Butler Group, a market research firm.

Industry observers say the ESB market is starting to take off. The technology has demonstrated its benefits, and companies are moving beyond pilot implementations, according to Fulton.

For the next five years, SOAs, and thus ESB, should remain the dominant application-integration approach, predicted Illuminata's Kernochan.

In fact, as Figure 1 shows, Winter-Green Research, a market analysis firm, predicts the global ESB market will grow from \$203.8 million this year to \$494.4 million in 2013.

Very large organizations—primarily in finance, telecommunications, and government—have been the first to implement ESBs, said the Butler Group's Hailstone. However, he said, with more ESB vendors—as well as a couple of open source products—in the market, prices are starting to fall, which should make the technology more accessible by midsized organizations.

“Companies will initially struggle with what functionality should be configured into their ESBs and what really belongs elsewhere,” said Fulton. “But at the end of the day, IT organizations embracing SOA on a large scale will certainly embrace ESB as well.” ■

*Sixto Ortiz Jr. is a freelance technology writer based in Spring, Texas. Contact him at [sortiz1965@gmail.com](mailto:sortiz1965@gmail.com).*

**Editor: Lee Garber, *Computer*,  
[l.garber@computer.org](mailto:l.garber@computer.org)**

**Renew your  
IEEE Computer Society  
membership today!**

**[www.ieee.org/renewal](http://www.ieee.org/renewal)**