Enterprise Service Bus 101

Marty Wasznicky
Director, Product Business Development
Neudesic
Table of Contents

Abstract ......................................................................................................................... 3

Understanding the Enterprise Service Bus ................................................................. 4
  Defining the ESB ......................................................................................................... 4
  An Infrastructure for SOA .......................................................................................... 4
  The Convergence of EAI, MOM, and SOA ................................................................. 5

Business Benefits of an ESB ....................................................................................... 6
  Improve Business Agility ............................................................................................ 6
  Streamline Business Execution .................................................................................. 6
  Expand Business Intelligence ..................................................................................... 6
  Decrease Costs ........................................................................................................... 6

IT Benefits of an ESB ................................................................................................... 7
  Increase Flexibility ...................................................................................................... 7
  Lower Total Cost of Ownership ................................................................................ 7
  Strengthen Operational Reliability ............................................................................ 7
  Decrease Risk ............................................................................................................ 7

Making the Case for ESB ............................................................................................ 7
  Responding to ESB Critics ....................................................................................... 8

Conclusion .................................................................................................................. 9
  About the Author ...................................................................................................... 9
  About Neudesic ......................................................................................................... 9
Abstract

For organizations looking to streamline and simplify their infrastructure and support Service Oriented Architecture (SOA) initiatives, an Enterprise Service Bus (ESB) can be the answer. Unfortunately, this technology remains a mystery to many companies, and even the definition of an ESB can vary depending on who you talk to.

This white paper attempts to clear up some of the confusion around ESBs and help your organization determine if this technology is needed to support your integration and SOA initiatives.
Understanding the Enterprise Service Bus

Since gaining popularity in the last few years, ESB has become the subject of intense interest by enterprise customers and heated debate by technologists—a clear sign that the idea either carries significant merit or is compelling hype without substance. Certainly the noise level alone on ESB warrants an open-minded examination of the concept.

Before diving into the technology, let's first take a look at what an ESB is.

**Defining the ESB**
If you've done your research, you've probably seen that ESB can be described in a number of ways. After examining the most common definitions (below), it becomes apparent that these descriptions are actually pieces of a larger, composite definition that accurately describes this technology.

- An ESB is a backbone for integrating an enterprise's applications and services.
- An ESB provides the necessary infrastructure to create a Service Oriented Architecture.
- An ESB is based on open standards such as XML, SOAP and WS-*.
- An ESB provides intelligent routing, such as publish-subscribe, message brokering, and failover routing.
- An ESB provides mediation and resolves security differences between endpoints.
- An ESB integrates with legacy systems using standards-based adapters.
- An ESB provides logical centralized management but is physically decentralized.
- An ESB is able to monitor and throttle activity as per a Service Level Agreement (SLA).

It is possible for an ESB solution to include some, but not all, of the functionality listed above. At this point, it's up to you and your organization to determine if the solution meets your unique needs and will support your projects.

Although it's not part of the composite definition, I would propose adding one more point that should help distinguish a true, well-intentioned ESB design:

- An ESB is modular and product-agnostic; its parts can be replaced with new implementations.

Notice that I haven't specified the development environment in any definition for ESB. There are ESB solutions available for both Java and .NET (and very rarely, for both). If you are a .NET developer specifically, you would naturally look for a solution that supports .NET. Why this is important goes beyond the environment you are working in to include how the ESB solution will actually support your overall initiatives and simplify your .NET development processes.

Therefore, it's as equally important to know how an ESB supports your organization as it is to know what it is and how it works. In the next section, we'll look at how an ESB supports a common – and critical – enterprise initiative: Service Oriented Architecture (SOA).

**An Infrastructure for SOA**
SOA has given us a good set of principles (such as loosely coupled communication), an excellent set of standards that are composable and ongoing (WS-*), and compelling new technologies such
as Windows Communication Foundation (WCF). While SOA is still young, it’s real enough to put into action now in your organization.

As enterprise-wide adoption of services continues, the need for SOA will start to be felt as certain issues arise, such as: how to effectively manage large numbers of services; how to overcome differences between services; how to enforce SLAs; and how to enforce enterprise policies across distributed collections of services.

What would an ideal SOA look like? It’s instructive to visit past architectures that have fallen in and out of favor over the years (Figure 1). Point-to-point architecture works all right on a small scale, but its problems become apparent when used at the enterprise level. If each system has to know the connection details of every other system, then each additional system increases the problem of configuration and management exponentially. This was the impetus that led us to hub-and-spoke architectures, which most Enterprise Application Integration (EAI) products use. This architecture was a vast improvement over point-to-point architectures, and each system needed to communicate with only the hub. In addition, the hub could provide excellent management features since it was a party to all communication.

Figure 1: Messaging Architectures

In time, shortcomings with the hub-and-spoke approach were revealed, including concerns about scalability, single point of failure, and vendor lock-in. Fortunately, there is a sound compromise in the bus architecture which provides the benefits of logical centralization but is physically decentralized. The bus architecture in earlier days was often used in message bus systems based on proprietary technologies, but a modern ESB implements this architecture using WS-* standards.

So, think of an ESB as a set of infrastructure services that complement your business services.

**The Convergence of EAI, MOM, and SOA**

While the ESB can be described as a recent approach to enterprise integration, it really stands on the shoulders of three disciplines in wide usage: SOA, EAI, and Message Oriented Middleware (MOM).

The superset of capabilities that comes from combining SOA, EAI, and MOM is extremely compelling. SOA makes loosely-coupled, decentralized solutions possible that are enterprise-
ready and based on interoperable standards. EAI allows integration of any combination of systems, with sophisticated message brokering, message translation, business process orchestration, and rules engine processing.

MOM provides intelligent routing such as publish-subscribe topical messaging and strong managerial controls over routing, auditing, activity monitoring, and throttling.

However, each discipline has some weak areas: SOA needs better enterprise manageability; EAI needs to become decentralized; MOM needs to get away from proprietary technologies. Combining these disciplines properly in an ESB overcomes these weaknesses.

**Business Benefits of an ESB**

Now that you’ve seen some common definitions for ESB and how it supports critical initiatives, like SOA, the next question most likely is: “How does it benefit my organization?”

Simply put, an ESB:

- Improves business agility
- Streamlines business execution
- Expands business intelligence
- Decreases costs

**Improve Business Agility**

An ESB makes an organization more nimble, more responsive and more adaptive. The ESB matches up events—whether expected or unexpected—with appropriate IT services. Unlike many traditional infrastructures that are resistant to change, an ESB is designed to accommodate change easily and painlessly. An ESB’s flexibility makes it easier to add new partners at a moment’s notice, pursue a sudden business opportunity, or improve time to market for a new product or service.

**Streamline Business Execution**

Additionally, an ESB improves an organization’s ability to execute efficiently and consistently meet its commitments. Business rules can be enforced across the breadth of the enterprise. The availability, capacity, and responsiveness of enterprise systems can be managed to match service level agreements. And enterprise activity can be captured for compliance auditing.

**Expand Business Intelligence**

If your organization needs insight into enterprise activity, an ESB can be very helpful. As the clearinghouse for all messaging, the ESB has a full view of enterprise activity. Combined with business activity monitoring, your organization would be equipped to track key performance indicators for the business based on the messaging between enterprise systems and access business information in real-time.

**Decrease Costs**

Finally, an ESB decreases an organization’s operating costs, namely because it enables the business to do more with less. Additional value can be created from existing software assets and
existing technical skills by extending their reach. The time and cost for IT to develop, integrate or deploy, integrate or deploy solutions is reduced.

There are clear benefits and justifications for an ESB solution at the business-level... but what about benefits for the business unit that will be most closely linked to the tool: the IT organization?

**IT Benefits of an ESB**

For the IT organization, an ESB delivers several significant benefits, including:

- Greater flexibility
- Lowered total cost of ownership (TCO)
- Strengthened operational reliability
- Decreased risks

**Increase Flexibility**
The ESB’s event-driven architecture is both responsive and adaptive. Change management becomes simpler and more powerful. Communication and routing details are no longer embedded in application code and are easily reconfigured. Changes can be applied dynamically without disrupting systems in operation. And real-time integration replaces lengthy integration projects. Due to these benefits, an ESB can also greatly accelerate mergers and acquisitions.

**Lower Total Cost of Ownership**
An ESB contributes to a lower TCO in several ways. Its long-term architecture avoids the need to periodically reinvent the enterprise. Additional value is created from existing software by extending its reach and maximizing reuse. And expensive development and integration projects can become smaller or disappear entirely in favor of configuration changes.

**Strengthen Operational Reliability**
Due to its physical decentralization, a service-based architecture is highly scalable. The logically centralized management promotes high availability, as does the freedom to apply configuration changes without taking down critical systems. Operational health and activity monitoring, throughput controls, and failover routing collectively help enforce service level agreements. And with an ESB, you can typically adjust the Quality of Service standards to provide desired throughput, security, and delivery assurances.

**Decrease Risk**
Because an ESB provides the freedom to select and combine best-of-breed products, *while* preserving the enterprise architecture when those products need to be changed out, the solution can significantly decrease risk for the enterprise. Additionally, when you implement an ESB that uses open standards, it opens up a greater spectrum of compatible tools and systems.

**Making the Case for ESB**
In this paper, we’ve reviewed what an ESB is – both how it is commonly defined and how it functions. Additionally, we’ve looked at how an ESB solution provides benefits to both the IT
organization and the business as a whole. That information alone might be enough to make the case for implementing an ESB in your organization … but perhaps not!

Responding to ESB Critics
There are some critics of ESB technology, and some common arguments against the solution state that ESBs are:

- Ill-defined
- Hype without substance
- Proprietary
- Overly centralized
- Short-term, transitional solutions
- Just another name for EIA
- A departure from SOA
- Or, simply unnecessary

Our examination of ESB demonstrates the opposite, and generates the following responses to that criticism:

- **Ill-defined:** While there are many ESB definitions, a composite definition shows more similarities than differences.
- **Hype without substance/Proprietary:** While some ESB solutions may turn out to be hype without substance or involve proprietary solutions, not all of them are.
- **Overly centralized:** It’s misleading to call ESBs centralized; rather, they are logically centralized but physically decentralized.
- **Short-term, transitional solutions:** An ESB is not a short-term solution but a long-term architectural pattern; it may certainly leverage products but is not tied to any particular product.
- **Just another name for EIA:** An ESB is not another name for EAI. Rather, it is a convergence of EAI, MOM, and SOA.
- **A departure from SOA:** An ESB is not a departure from SOA. Instead, an ESB is a service oriented architecture enabler.
- **Simply unnecessary:** ESBs are necessary for enterprises that plan to take service orientation adoption to mature levels, or who require easy, low-cost change management.

Another case for ESB comes from industry analysts. A 2006 Forrester study states:

ESBs are also typically less costly than other ways of doing integration, such as integration-centric business process management suites (IC-BPMS), for three reasons:

- Configuration is easier
- Standards support drives skills availability
- Costs are lower
Conclusion

While ESB technology is gaining traction, it is still relatively new and for that reason it’s natural that there is still some confusion about what it is – and, more importantly, how it supports organizations. Before investing in any technology, it’s essential to ensure the solution meets your company’s needs and an ESB investment is no exception. But it’s equally important to separate the facts from the misplaced criticism, and hopefully this paper helped you start down that path.

At the end of the day, any ESB you choose to implement should:

- Fit the overall, compound definition of an ESB, and offer the functionality needed by your organization – including support for your development environment (.NET)
- Support your long-term enterprise goals, especially if you are implementing SOA
- Deliver benefits to IT and the business as a whole by lowering overall costs, increasing total cost of ownership, easing development processes, ensuring SLAs are met, and supporting a scalable, maintainable enterprise

About the Author

Marty Wasznicky serves as the Director of Business Development for Neudesic, bringing with him over 18 years of experience in the software development industry. Marty joined Neudesic after 6 years as a Principal Program Manager at Microsoft. At Microsoft, Marty worked in the Connected Systems Division where his responsibilities ranged from building out Microsoft’s BizTalk Server product integration business and managing a team of SOA/ESB/BPM field specialists, to building strategic partner alliances and collaborating on architecture and features for Microsoft’s distributed technologies. Marty also led the vision, development and architecture of the Microsoft’s Enterprise Service Bus Guidance. Marty has published nearly a dozen whitepapers and articles on various technologies. Most recently he was a contributing author for the Pro BizTalk 2006 book and currently sits on the Corporate Advisory Board for SOA Software, a leading provider of SOA Governance solutions.

About Neudesic

Neudesic is a Microsoft National Systems Integrator and Gold Certified Partner with a proven track record of providing reliable, effective solutions based on Microsoft’s technology platform. Neudesic’s technical and industry expertise empowers enterprises to enhance their technological capacity and respond to business opportunities with a greater level of efficiency. Established in 2001 and headquartered in Irvine, California, Neudesic offers its products and services nationwide with offices located throughout the United States, and a global presence based out of Hyderabad, India.

For more information about Neudesic products, including Neuron-ESB, visit: http://products.neudesic.com.