

BEA White Paper

Applying a BPM and SOA Approach to Achieve Agile Business Integration



Copyright

Copyright © 1995–2007 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc. Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA SYSTEMS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks and Service Marks

Copyright © 1995–2007 BEA Systems, Inc. All Rights Reserved. BEA, BEA AquaLogic, BEA eLink, BEA WebLogic, BEA WebLogic Portal, BEA WebLogic Server, Connectera, Compoze Software, Jolt, JoltBeans, JRockit, SteelThread, Think Liquid, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Blended Application Development, BEA Blended Development Model, BEA Blended Strategy, BEA Builder, BEA Guardian, BEA Manager, BEA MessageQ, BEA microServices Architecture, BEA Workshop, BEA Workspace 360, Signature Editor, Signature Engine, Signature Patterns, Support Patterns, Arch2Arch, Arch2Arch Advisor, Dev2Dev, Dev2Dev Dispatch, Exec2Exec, Exec2Exec Voice, IT2IT, IT2IT Insight, Business LiquidITy, and Liquid Thinker are trademarks of BEA Systems, Inc. BEA Mission Critical Support, BEA Mission Critical Support Continuum, BEA SOA Self Assessment, and Fluid Framework are service marks of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties. All other trademarks are the property of their respective companies.

CWP1642E0807-1A

Table of Contents

- Introduction1
- Working from the top down: BPM2
- Working from the bottom up: Integration4
 - Application Integration4
 - Data Integration5
 - Multiple layers6
- The crucial role of mediation7
 - Not just a transport8
 - Multiple logical levels8
 - Best Practices9
- Bringing everything together9
 - Tooling10
 - Deployment & management10
 - Governance10
- Conclusion10
- What to do next?11
- About BEA11
- Join the BEA community11

Introduction

Every day, information technology (IT) professionals struggle with agility: how to support new business processes, how to adapt to changing conditions, and how to improve execution efficiency. They desperately want to turn their technical capability into business agility. But as engineers, they don't believe in a soft, haphazard approach. They need a methodical, repeatable procedure. Business Integration encompasses both the procedure and technology for transforming technical capability into business agility.

In the past, theorists have confused “methodical” procedure with “linear” processing. So Business Integration approaches were either top-down or bottom-up. Top-down approaches decomposed business processes further and further until arriving at specific computational operations—unfortunately, these operations often bore little resemblance to existing software functions. Bottom-up approaches built up more and more abstract computational operations—unfortunately, these operations often bore little resemblance to coherent business processes. To make matters worse, the infrastructure for linking individual operations was so rigid that enterprises couldn't adapt their business and IT mappings quickly enough to be competitive.

Service-Oriented Architecture (SOA) bridges the top-down and bottoms-up perspectives to overcome both challenges. To overcome the first, it offers a unifying conceptual unit of work that enables IT and the business to work in parallel: the atomic business service. With this concept in place, the business can work with their processes from the top down and IT can work with their resources, systems, and software from the bottom up. SOA gives each group a combination of flexibility within its own area as well as visibility into the other area—they can actually work towards each other and meet in the middle.

From the business perspective, SOA-based Business Integration starts with a defined business objective and its associated processes. Business Analysts use modeling tools to sketch or visualize the process flow. They move on to detailed process simulations and user interface prototyping to help refine the model down to a set of atomic business services. From the IT perspective, SOA-based Business Integration starts with existing assets and their capabilities. Developers convert existing assets into foundational services. Then they build up more and more sophisticated services, adding new ones where necessary, until they can compose the required atomic business services that support the business process. Obviously, there's a negotiation about which atomic services the business needs and which ones IT can deliver. SOA gives them a shared vocabulary and mutual visibility. IT can understand the business requirements driving the need for each service and the business can understand the technical challenges associated with implementing each service.

As for the challenge of flexible yet enterprise-class Business Integration infrastructure, SOA can naturally leverage the enterprise service bus (ESB). It provides a layer of indirection between service requesters and providers. As a middleman, it can mediate quality-of-service and service communication needs. It inspects each message that flows between the service consumers and business services, applying all the appropriate priority, routing, transformation, and security policies. Moreover, it allows enterprise to seamlessly swap in alternative service implementations, transition from older to newer versions, rearrange the sequence of services in a process, enforce governance policies, and monitor service execution. With the ESB, Business services can now become as plug-and-play as the business requires.

Atomic business services and the ESB are enablers. Realizing their potential to turn IT agility into business agility requires a 360-degree approach to Business Integration. Neither the business nor IT is monolithic. Each encompasses many different roles that contribute throughout the process automation lifecycle. Enterprises need tooling that will support each role individually and coordinate their combined efforts. They also need proven infrastructure for executing the resulting processes. BEA offers a 360-degree solution that delivers the benefits of cooperation, visibility, governance, time-to-market, and enterprise-class quality of service (QoS) to the challenge of Business Integration.

Working from the top down: BPM

As shown in Figure 1, the goal of Business Integration from the business perspective is to provide process participants agile business process management on top of atomic business services. This process management has two parts: the models that describe how processes should execute and the software that makes this execution a reality.

Creating the process models is analogous to writing a story. Instead of outlines, business analysts use flow diagrams. Instead of brainstorming, they use simulations. Instead of storyboards, they use user interface prototypes. Together, these tools dictate what should happen. Actually making it happen requires executable software—the evocative prose in the story analogy—which is where the business organization must rely on the technical wizardry of the IT organization. Of course, unlike a traditional story, business analysts also need to continually re-write processes to respond to change and improve efficiency. Moreover, executing a sophisticated process is far more complex than printing the pages of a story. SOA is their key to delivering this extra adaptability and responsiveness.

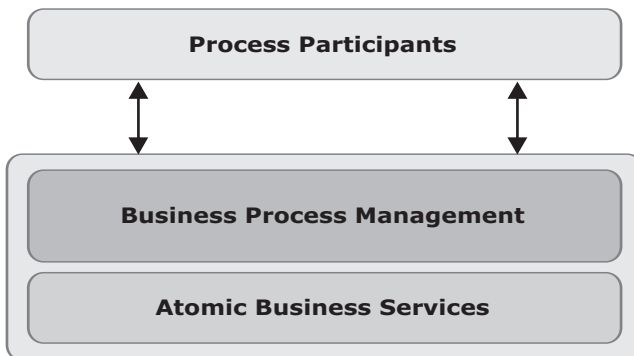


Figure 1

Business perspective on Business Integration.

Modeling tools, such as the Process Designer in BEA's AquaLogic® BPM Suite, enable business analysts to outline the “story” of each business process using preferred conventions such as Business Process Modeling Notation (BPMN) or Unified Modeling Language (UML). Analysts start with a very high level process such as “Deliver Order.” Then they decompose it into ever more fine-grained processes. So a contract manufacturer might drill down from “Build Order”, to “Pick Components”, to “Order Parts”, and finally to “Send Purchase Order”. Once they get down to a level where they can't decompose the process further without specifying technical implementation details, they have reached an atomic business process.

At some point in the decomposition, analysts may need to simulate the process to either identify missing tasks or opportunities for improvement. That's why BEA's AquaLogic® BPM Designer includes extensive simulation tools. For each task in a process, analysts can assign such parameters as the mean and variance in processing time, queuing strategy, and likelihood of different possible outcomes. It can even simulate multiple processes that utilize the same resources so analysts can identify resource contention issues.

In addition to accounting for process dynamics through simulation, analysts must account for human dynamics through interface prototyping. Mocking up wireframes for the screen that users might access to complete a task is the most evocative way of modeling how they will interact with the process. It ensures that analysts have accounted for all the information the process must provide the user and all the information the process expects from the user. The end result of process decomposition, process simulation, and interface prototyping is a complete encapsulation of the desired behavior for atomic business services, just without any implementation. The business relies on IT to provide it.

As discussed above, the process is not linear. For certain processes, IT may have already made some of the necessary atomic business services available. The business can just directly reference and access them at the BPM layer. Of course, after the initial deployment of any process, analysts will be able to make changes via modeling as well. They simply rearrange how the process definition uses the available atomic business services. Naturally, some changes will require additional programming and a true Business Integration solution should allow them to seamlessly push down these requirements to IT.

To deliver the benefits of automation and adaptability, a business process model requires a robust execution medium. The execution components translate a process definition into executable instructions and then invoke the necessary atomic business services. They also provide a web-based workspace for people to execute their tasks and manually handle exceptions. Monitoring components track real-time process flow and long-term process metrics to minimize response times to both sudden crises and evolving environments. BEA's AquaLogic® BPM Suite delivers all these modeling, execution and management components in a package with proven reliability, availability, and scalability.

Working from the bottom up: Integration

As Figure 2 shows, the goal of Business Integration from the IT perspective is to build up atomic business services from existing assets. This aggregation occurs through two mechanisms: Application Integration and Data Integration.

While the business must create the stories behind all of an enterprise's processes, IT must provide the vocabulary with which the business can articulate these stories. IT staff package the necessary "verbs" using application integration and the necessary "nouns" using data integration. In most cases, they won't be able to directly map current asset capabilities to atomic business service functions. Instead, they'll need several layers of intervening services to achieve good abstraction and flexible reuse. Of course, every layer must deliver high service levels to ensure reliability of the larger processes. SOA provides the framework for ensuring both good abstraction and reliable service delivery.

Application Integration

Application Integration technologies provide the action-oriented units of the atomic business service vocabulary. For example, consider an action such as "Verify Order" for an enterprise that offers build-to-order computer products. This action ensures that a customer order can be built and delivered within a specified time frame. It would require cooperation among configuration, inventory, manufacturing, and logistics applications.

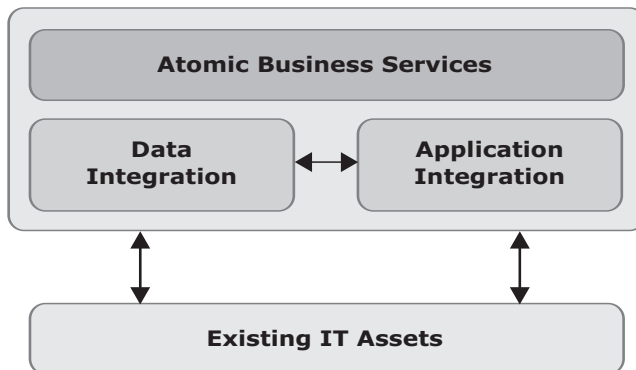


Figure 2

IT perspective on Business Integration.

Application Integration is necessary because this kind of action isn't just a series of application calls with format transformations. The primary flow might be perfectly linear, going from one application to another. However, accounting for all the possible flows requires specifying more complex interaction. In the Verify Order example, some configurations may correspond to pre-defined SKUs while others require custom assembly. There's a conditional branch. Then what if the inventory application shows that a component specified by the configuration application is out of stock? There should be special logic for iteratively identifying potential substitutions. Moreover, some exceptions may not be easily predictable or resolvable. When unforeseen problems arise, such as a deadlock between the manufacturing scheduler and the logistics scheduler, the integrated application should route orders to a specialist for manual assistance.

To address the full range of action-oriented possibilities, enterprises need an application integration platform. At the lowest level, this platform offers connectivity for communicating with a comprehensive range of packaged applications, as well as both legacy and modern platforms running custom applications. On top of the connectivity layer, it requires its own workflow engine for specifying task sequences, conditional branching, and exception handling. As noted above, certain real world issues require a more flexible response than even a workflow engine can conveniently handle. Therefore, the solution also needs facilities for plugging in specialized logic modules, rules engines, and brokering manual intervention.

BEA's WebLogic® Integration provides an ideal platform for Application Integration as part of a Business Integration Solution. Developers can visually define flows among almost any applications, leveraging its full suite of back end connectivity options, and can insert new Java logic modules where needed. A built-in Worklist capability automates the task of providing user interfaces for performing approvals and handling exceptions. Its underlying process engine is fully transactional, ensuring that enterprises can maintain consistency across applications in fine-grained process flows. These advanced capabilities enable IT to rapidly build up the library of actions that underlie the atomic business services.

Data Integration

Data Integration technologies provide the entity-oriented units of the atomic business service vocabulary. For example, consider an entity such as "Customer" that spans almost every department in an enterprise. This entity encompasses information about contacts, orders, payments, and support. Assembling a coherent picture of the Customer would require assembling data from every data source that holds any of these information types.

Data Integration is necessary because these diverse data sources use a wide variety of models and the services that consume the data have a wide variety of semantic requirements. In the Customer example, contact information may be split between an SFA application, billing database, and CRM application. Order information may be split between an online orders database, ERP application, and billing database. Obviously, packaged applications behave differently from databases. Furthermore, different applications have different API semantics and different databases have different data models. Some consuming services might want individual Customer records by ID, others may want all Customer records by zip code, while still others may want to filter based on different kinds of status. Then there's the challenging of preserving transactional integrity and access control across all the different sources.

To make entity information available to all the applications that need it, enterprises need a data integration platform. At the lowest level, this platform enables analysts to map data sources to data services. Then it has to

provide all the query planning and optimization features necessary for these services to respond quickly. It must bridge different transaction and security models, translating the policies for the aggregated entity into the individual calls necessary to implement those policies in each data source.

BEA's AquaLogic Data Services Platform provides the ideal foundation for Data Integration as part of Business Integration. Analysts can visually define mappings from data sources or use an advanced XQuery builder. It optimizes interactions with data sources through such techniques as pushing down operations to the underlying data source engines, passing join values from one SQL source to another in batches, and taking advantages of particular database implementation features. For transactions, it uses a combination of XA two-phase commits and compensating transactions for non-XA sources. For security, it offers an extremely fine-grained model that can not only redact fields like social security numbers which are generally confidential, but can also dynamically restrict access based on query properties such that only a patient's physician of record could see his medical records. These advanced capabilities enable IT to rapidly build up the library of entities necessary to support atomic business services.

Multiple layers

To achieve maximum flexibility and reuse, enterprises often require multiple layers of both application and data integration. For example, the inventory application used in the Verify Order example may itself be an aggregation of different inventory modules, perhaps one for externally sourced components like disk drives and one for internally sourced components like motherboards.

The number of layers really depends on the existing assets. If developers are lucky, a modern package may already provide high level action and entity abstractions. In this case, they may need only a simple wrapper layer or perhaps even no wrapper at all. If developers are unlucky, legacy packages may require extremely low-level synchronization to remain in a consistent state. Most cases will fall somewhere in between. Few modern packages will be perfect for emerging needs and most low-level synchronization will already be implemented.

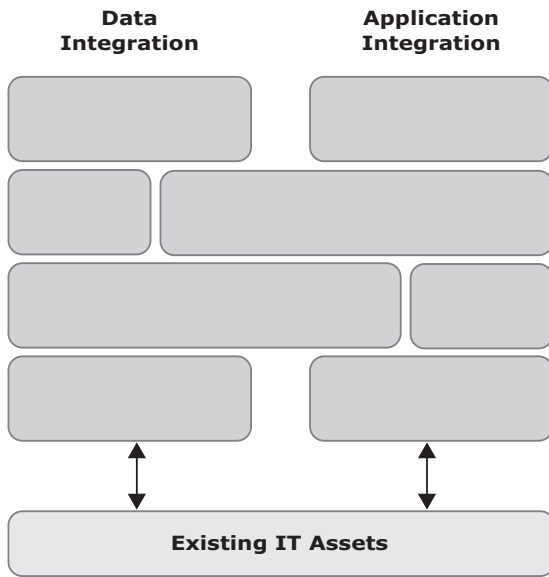


Figure 3

Multiple Data and Application Integration Layers.

The driving forces that determine the number of layers are modularity and dependency. Data analysts may determine that the concept of a “Contact” is general enough that it should be a standard building block for higher level data services. As noted above, the general inventory application may depend on several specific inventory applications. Modularity and dependency can even cross the boundary between application and data integration. For example the aforementioned configuration application may depend on a data service that standardizes the “bill of materials” entity. Or the aforementioned Customer entity may actually depend on an “add customer” action that coordinates ERP and CRM applications. Zooming in on the integration architecture produces a picture of stacked, interleaved layers like that in Figure 3. The advantage of SOA is that it allows enterprises to use the same paradigm, no matter how many layers are necessary to fit a set of existing assets to a desired set of atomic business services.

The crucial role of mediation

An ESB provides both a conceptual and technical foundation for SOA-based Business Integration execution and shields each layer in the SOA stack from disruption caused by changes in other layers. Conceptually, the business determines which atomic business services it needs on the bus and IT works to put them on the bus as shown in Figure 4. They can actually meet in the middle because an ESB does much more than simply transport messages—it actively mediates among services so that they can cooperate effectively. It provides a medium that simultaneously offers reliable, flexible service connectivity and effective insulation from changes in service implementations. The ESB mediates for more than just the atomic business services. It operates at multiple levels so every logical unit, from low-level legacy functions to encapsulated business activities, can get the benefit of mediation. In fact, the ESB offers so many capabilities that it’s sometimes hard to identify when there’s a better tool for the job at hand. It does not supplant the need for business process or application integration tools. Establishing a set of best practices can help guide when to use an ESB directly versus when to use one of these other tools to create a new service that connects to the ESB.

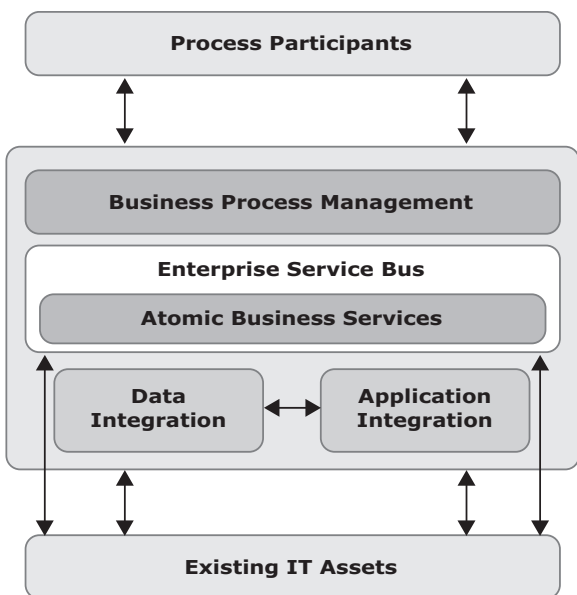


Figure 4

Not just a transport

Interaction among services requires more than just moving messages from one point to another. In fact, providing transport is secondary. The primary mission of the ESB is to promote cooperation. The basic approach is to offer a layer of smart insulation between service requesters and service providers. Without this insulation, every requester would have to know so much about every possible provider that service interactions would quickly become ossified. Instead, the ESB mediates among services, doing whatever is necessary to deliver reliable, seamless cooperation.

By acting as an intermediary, the ESB can certainly bridge transport protocols. If a requester uses JMS and the provider uses HTTP, they won't even notice the difference with the ESB in the middle. In fact, as long as the ESB has appropriate connectors, services can use such transports as proprietary MQ messaging or even email protocols. But the transport is just the tip of the iceberg. What if the requester and provider use different versions of a message format or alternative formats from different standards bodies? The ESB can automatically transform them in flight. As long as two services understand semantically equivalent messages, the ESB allows them to cooperate.

Security comprises another area where the ESB facilitates smooth service interaction. Different services may use different security models, such as applying integrity protections at the transport level through HTTPS versus at the message level through signatures. The ESB can use its trusted position to deliver equivalent protections across such boundaries as well as propagate security identities so each service can make its own access control decisions. In addition to security policies, the ESB can also implement business policies. Content-based routing allows the enterprise to direct messages from requestors to particular providers based on any characteristic, such as customer geography, transaction risk, or order value. The centralized logging, alerts, and exception handling SLAs that an ESB provides give enterprises complete situational awareness of how these service interactions are operating.

Multiple logical levels

Atomic business services are just one of the logical layers where the ESB delivers value. Previous sections discussed how the business reaches them by decomposing business processes and how IT reaches them by aggregating technical services. Both the higher level process layers and the lower level technical layers can also benefit from the ESB facilities to cooperate.

Above the atomic business service layer, the ESB gives enterprises process flexibility. The initial process decomposition yields reusable business modules, all connected by the ESB. If a change in the environment occurs or if an idea for innovation arises, analysts can respond by simply reconfiguring their models. The ESB then facilitates the new connections among modules. This adaptability yields an agile, model-driven enterprise.

Below the atomic business service layer, the ESB gives enterprises technical flexibility. Altering existing assets, inserting new logic, and enhancing integration all become easier with the ESB. Because the ESB can intelligently route requests to the appropriate version of a service, developers can make new capabilities available without breaking old connections, facilitating version management and upgrade support. The same routing capability makes it possible to seamlessly insert a new value-added service in front of an existing one. Better transactional, security, and exception integration can all happen under the covers without intruding on operational service interaction. So while the ESB is conceptually a backbone of flexibility for Business Integration, it is actually a single utility available throughout the stack.

Best Practices

While the ESB is a very powerful tool within SOA, it is not always the right one. Because BEA provides the AquaLogic® Service Bus, an industry-leading ESB, as well as other leading enterprise middleware products, BEA's staff has a tremendous amount of experience with when to use an ESB.

The biggest lesson BEA has learned after working on a wide variety of customer-driven projects is that trying to do the wrong things on the ESB can degrade both performance and flexibility. The most important rule of thumb is that the things connected to the ESB should truly have service semantics. For example, consider the previous example of synchronizing an add customer operation across ERP and CRM applications. If the ERP and CRM application both have service APIs, it would be technically straightforward to provide this synchronization via the ESB. However, it would require maintaining a lot of state on bus itself, which reduces performance. The need to maintain state outside of any service indicates inadequate semantic richness. A better solution would be to encapsulate that state as part of application integration done through WebLogic Integration.

In addition to degrading flexibility, using the ESB inappropriately can also degrade performance. For example, it would be possible to put RFID messages directly on the bus for applications like inventory control or baggage handling. But this approach would flood the bus with low-level, almost meaningless messages. Every requesting application would have to worry about filtering and aggregating those messages itself. A better solution would be to encapsulate the direct RFID communication in an event server, such as BEA's WebLogic Event Server, that can then interact at a more abstract level with services via the ESB. BEA routinely works with AquaLogic Service Bus customers to educate staff about general best practices and develop customized ones.

Bringing everything together

SOA offers the basic ingredients required for achieving an agile state of Business Integration. However, Business Integration is fundamentally a cooperative undertaking whose ultimate effectiveness relies on coordinating contributions from all the individuals working for the business and IT on their shared objective. To maximize return-on-investment, enterprises need mechanisms to direct and magnify this collaborative energy. BEA's comprehensive Enterprise360° approach and associated infrastructure, which incorporates Business Integration along with Governance, Interaction, and a Rock-Solid Foundation, promotes collaboration, smoothes operations, and facilitates oversight.

Tooling

Promoting collaboration means that everyone contributing to the Business Integration effort has productive tools that integrate with each other. BEA's approach is founded on the concept of WorkSpace 360 to provide integrated capabilities that support both team and project based collaboration. Teams can define shared project workspaces that include both business and IT contributors. Within these workspaces, they can have interactive discussions, work jointly on documents, and set notifications for changes to shared items while each maintaining their appropriate view. These capabilities will ensure that each contributor understands their role in the larger context of the Business Integration effort.

At the hub of WorkSpace 360 is a rich metadata model necessary for full traceability in Business Integration projects by tracking the correspondence between business process and software integration tasks. In addition,

WorkSpace 360 provides a tailored workspace for each audience in the development process. Business analysts get their modeling and prototyping tools. Architects get Service Component Architecture tools and customized dashboards. Developers get IDEs and reuse aids. Where concerns overlap or tasks flow from one role to another, Workspace 360 provides the necessary shared view. Aligning everyone's vision of Business Integration directly improves agility.

Deployment & management

Smoothing operations means that enterprises can efficiently deploy and manage Business Integration software components. All of BEA's components are able to run on and take advantage of the same rock-solid WebLogic foundation while interoperating with heterogeneous infrastructure. This enables enterprises to centrally manage all the execution infrastructure and deployment of Business Integration modules. Whether administrators want to upgrade BEA products or deployed services, they have the necessary capabilities to ensure a smooth transition.

For SOA modules running on top of the BEA infrastructure, administrators get unified configuration, monitoring, and reporting. They can even define and enforce security policies across the entire platform. While BEA infrastructure works seamlessly with third party components via standard protocols, products in the same family can use special high performance native messaging to deliver maximum throughput, known as "fast-path" integration. Of course, the proven enterprise-class capabilities of the WebLogic and AquaLogic families mean that all the services they run get the benefit of their reliability, availability, performance, and scalability.

Governance

Facilitating oversight means that managers can exercise effective governance by identifying appropriate policies, communicating them to contributors, and tracking compliance throughout the software lifecycle. While governance is important in all SOA initiatives, the breadth of Business Integration initiatives make it crucial in this type of installation—the enterprise must know that it is meeting its business goals and IT QoS goals in a cost effective manner. Once managers can measure the initial impact of a Business Integration initiative, they can begin refining future objectives.

Through Workspace 360, managers can map all design milestones and runtime statistics back to business objectives. For a given technical service, they can see all the atomic business services it supports, and in turn all the business processes that require it. They can actually measure how technical improvements contribute to business improvements. And when change occurs, they can predict how much technical adaptation will be necessary to support a given process adaptation. At a more detailed level, IT staff can use Workspace 360 to enforce architectural compliance and SLA requirements. Everyone is accountable for how their efforts contribute to the overall Business Integration goals.

Conclusion

An Enterprise360° approach to Business Integration maximizes benefits while minimizing risks. Clearly, enterprises must achieve and maintain alignment of IT capabilities with business goals. Flexibility is important. However, this alignment naturally requires coordinating resources across many organizational units over extended periods of time. Manageability is also important. Applying SOA to Business Integration delivers both flexibility and manageability.

An ESB serves simultaneously as a backbone that bestows flexibility and as a gateway that promotes manageability. The business works top-down to figure out what business services it needs on the bus. IT works bottom-up from existing assets to put those business services on the bus. At run time, the bus provides the layer of indirection necessary for rapidly adapting the available services to the current needs. The ability to continuously monitor the flow of messages among services improves manageability. At development time, the bus provides a goal line that focuses everyone's efforts. The ability to measure the difference between which services are needed and which services are available also improves manageability.

For development, BEA's Business Integration solution provides the business the tools it needs to describe how processes should execute. It gives IT the tools it needs to build the software that executes processes properly. Just as importantly, it also provides the development time infrastructure necessary to promote effective collaboration and the run time infrastructure necessary to ensure robust execution. IT gets agile software. The business gets agile processes. BEA is leading the charge towards agility with industry-leading products, practices, and people committed to implementing Business Integration across the enterprise.

What to do next?

Whether you're just starting out with Business Integration, or already well on your way, BEA has amassed a wealth of information to help you, and offers solutions to meet your needs. BEA offers a wide variety of resources at its SOA Resource Center, including a self-service SOA assessment to analyze your SOA readiness. Visit the BEA SOA Resource Center at bea.com/soa. BEA offers a similar readiness assessment for BPM at bea.com/bpmready.

About BEA

BEA Systems, Inc. (NASDAQ: BEAS) is a world leader in enterprise infrastructure software. BEA® Enterprise 360°, the industry's most advanced SOA-based offering, is a comprehensive approach to delivering business results that includes technology, professional services, best practices, and world-class partners. Information about how BEA helps customers build a Liquid Enterprise™ that transforms their business can be found at bea.com.

Join the BEA community

At BEA, we understand that developers need different kinds of resources than IT managers. And that architects face different challenges than executives. That's why we've created four unique communities that give you exclusive access to a formidable group of your peers, to a world of shared thinking, and to the kind of meaningful information that can make you more effective and more competitive. To join one or more of the BEA communities, simply register online at bea.com/register.



BEA Systems, Inc.
2315 North First Street
San Jose, CA 95131
+1.800.817.4232
+1.408.570.8000
bea.com

CWP1642E0807-1A