

# CBDI Report

## ***Service Oriented Architecture: An Introduction for Managers***

*An executive briefing on the value proposition for Service Oriented Architecture and Web Services, and strategies for adoption*

***Author: David Sprott***

ABSTRACT: Many organizations are now undertaking development of service oriented architectures, but the probability is that most will result in sub-optimal implementation. Most organizations will focus on a smaller set of objectives than they ought to, because they are overly influenced by project and or technical concerns, and not sufficiently focused on the broader business service view. This paper provides guidance for managers in understanding and delivering real business value from service oriented architecture.



*Independent insight for Service Oriented Practice*

# Contents

Section 1 - Introduction .....	3
Section 2 - Business Drivers and Opportunities .....	4
Section 3 - Basic Principles .....	6
Web Services - A big advance beyond software components .....	6
SOA - Enabling consistency, sharing and adaptability.....	6
SOA or Web Services .....	7
Managing Separation.....	8
Service Life Cycles .....	9
The Service Architecture.....	9
Formalizing Service Contracts .....	10
Investment Protection - Using Existing Assets .....	11
Enabling Response to Change - Delivering Business Agility.....	12
Section 4 - The SOA Roadmap .....	13
Early Learning.....	14
Integration .....	14
Reengineering.....	14
Maturity .....	15
Section 5 - IBM's Strategic approach to SOA .....	16
Business Perspective Drives Automation and Resources .....	16
Life cycle automation .....	17
Standardization and Abstraction of Service Infrastructure Management.....	17
Legacy and Existing Portfolio Integration.....	17
Section 6 - Summary .....	19
Next Steps and Resources .....	19

This report has been commissioned by IBM Inc.

## **Section 1 - Introduction**

A recent survey of Fortune 500 companies indicated that over 80% had altered their business model in a given two-year period. Two thirds of these – roughly half of the total respondents – claimed that this business change had been constrained by inflexible IT. In a survey by IBM Business Consulting Services, 90% of CEOs expect to transform their enterprise to become more responsive, particularly to customer demand, within the next five years. More than half expect to be engaged in significant company-wide transformation initiatives within two years.

A key part of many organizations' transformation program is to reorganize information resources as substantially independent, reusable services that create an inherently adaptable environment. Business and technical services may be published using open, standard protocols that create self describing services that can to be used independently of the underlying technology. Technical independence allows services to be more easily used in different contexts to achieve standardization of business processes, rules and policies. Collaborations, internal and external to an enterprise, can more easily be established enabling improvements in process and information consistency. This approach to IT systems, referred to as Service Oriented Architecture (SOA), is now widely recognized as having the potential to radically improve the responsiveness of both business and IT organizations.

Common practice however is that enterprises often take the easy way out - addressing a smaller set of objectives because it seems too difficult to do the job properly. We observe many organizations now undertaking development of service oriented architectures, but the probability is that most will result in sub-optimal implementation because the relationship to the business requirements is not fully understood and the value proposition not fully communicated. Consequently business management will not see nor realize the value of the SOA.

In this report we provide a broader context that equips decision makers with a balanced view of the rationale for SOA, guidance on how to combine short and long term objectives in a practical program and how to exert governance over the delivery process.

## Section 2 - Business Drivers and Opportunities

The idea that Services should be available and delivered "on-demand" has been recognized by the research community for a long time. Researchers in the 1990s often used video download as an example of an on-demand service, and early studies showed how collaboration between telecoms, cable and content providers could provide a film viewing service that combined the convenience and immediacy of broadcast television with the flexibility of video rental. (While such services are far less widespread today than many people a few years ago predicted they would be, most of the purely technological issues have long been solved, and the constraints have been largely commercial and regulatory ones.)

Business people implicitly understand the service concept because it is a basic mechanism by which normal business exchanges are arranged and executed. For example company A offers company/division B a service, and the service delivery is governed by a contract in which the obligations of the providing and receiving parties are clearly defined.

A business service generally has inherent flexibility, particularly for the consumer of the service. The supplier of a service can be changed because the supplying party's connections to the consuming company are deliberately engineered to be specific to a particular service delivery. We might term this as loosely coupled.

The business service will also enable collaborative arrangements - a consuming company may easily acquire services from multiple companies, each providing specialist services that combine to create a highly functional process or product. Creating composite business applications in this manner is the central principle underlying the on-demand business, enabling rapid response to changing market conditions.

Applying this basic thinking about business services to information services can radically alter the dynamics of IT service provision:

- A service-oriented architecture that properly reflects real world business services creates **convergence of the business and IT perspective**, which enables much greater efficiency, adaptability and cost control in business relationships and structures.
- The service provides a basis for better business and technical systems design, **improving understanding and communications** and implementation of boundaries, collaborations and contracts.
- Properly designed services **create looser coupling between business models and technologies**, which reduce dependency on specific products and technologies, and break down barriers between application silos.
- Specialization of functionality into substantially independent services can facilitate common or **shared business and technical services that enable consistent policy implementation** with local variation.
- The self describing nature of the runtime service enables **automation of business rules and technical service management functions**, reducing human intervention and facilitating process improvement such as straight through processing.

- With real time, heterogeneous access services can **enable optimization of information ownership and improved data quality** replacing replication of critical data with real time access.
- Services are a run time concept - they are available to authorized users in real time. This enables **real time business intelligence**, together with real time response to events.
- Because business and technical services are self describing and runtime concepts, they may be used by consumers without development effort by the service provider. Sometimes the using parties may be anonymous or have no prior relationship with the provider. These characteristics create the potential for services to be used on a much broader basis, perhaps by unconventional consumers and markets to **change business models in terms of economics, scale and scope**.

The evolution to SOA is equally a business **and** an IT transformation. The business opportunities created by SOA potentially enable a radically different form of enterprise where business capabilities are implemented as automated services, and processes are easily integrated across the enterprise ecosystem including its partners, suppliers and customers.

The IT opportunity is to enable business and technical capabilities that have a high degree of independence from both the technology and related functionality. In the past IT architectures have focused on separation of layers. In the SOA environment, separation of concerns can be multi-dimensional, where capabilities are loosely coupled in both technology AND functional terms.

## **Section 3 - Basic Principles**

### **Web Services - A big advance beyond software components**

For experienced information technology personnel, the Web Service concept will be familiar because it builds on the well understood principles underlying software components, which offer services through an interface. The interface specification provides the external perspective and a well designed component allows the user to make use of the service(s) provided as a black box, simply by being aware of the interface details. The Web Service is a big improvement on the component architecture because it further extends the separation and independence of the service from the technical implementation providing the service.

The Web Service is a run time implementation of a component which can be invoked by an authorized consumer. It is self-describing and enables contractual, security and transaction management functions that enable the service to be freely executed across the Web. Even more important the Web Service is implemented using open standards that allow authorized users to make use of the service irrespective of their technology or application environment. The only pre-requisite is that the user complies with the relevant standards.

Web Services can potentially be put to use to implement a wide range of business and software services. Because the Web Service is simply a technology, the range of uses to which it can be put are infinite. A service might be a simple document exchange or a complex business service which allows an authorized user (which might be another computer system) to check the status of a logistics shipment. Many independent software vendors (ISVs) have already upgraded their applications to use Web Services so they can be more easily integrated into enterprise environments. Others have gone further and offer their entire functionality as services.

From the foregoing, it might be easy to conclude that Web Services are merely a better way of deploying a component-based architecture, that services are a better form of component interface. However, the service concept is even more useful because it can often map directly to the business view of the world.

It is very common for an enterprise to buy in services of one kind, and sell services of a different kind. For example, an airport operator buys in a range of responsibility-based maintenance services, and sells landing slots to airlines. A rail network operator may subcontract track maintenance to a group of engineering firms, and sell time on the network to train operators. One of the challenges facing such enterprises is that the mapping between the input services and the output services is often very complex. If you want to deliver a given quality of service to your customers – for example, a defined level of availability on an access-based service - you need to convert this service level into a series of separate service levels, which are then negotiated with your subcontractors. In the Web Services environment, the appropriate business rules can be applied automatically at runtime using the Web Services management infrastructure.

### **SOA - Enabling consistency, sharing and adaptability**

Web Services can be very useful to enable two or more enterprises to collaborate in a shared business process - the technology neutrality and standardized approach make this

much easier than using proprietary integration technologies. However the same service oriented thinking is equally applicable inside an enterprise.

An enterprise might create a set of services for CRM that set an internal company standard for customer interaction, even though there may be many different applications and databases serving different business divisions or geographies. Once the standards have been set and implemented into each of the disparate systems, it will be possible to have an enterprise wide view of a customer, without changing all of the underlying systems. This can be even more useful when a merger or acquisition occurs, allowing a newly acquired company to quickly integrate with the existing enterprise systems.

This idea that systems offer services has applicability everywhere; it can be used to rejuvenate legacy enterprise systems, to integrate disparate applications and packages, and to loosely couple systems and businesses that were not originally designed to collaborate. The SOA will almost certainly be more responsive to change in business requirements and organization because the service interfaces allow new components to be introduced and existing ones to be easily replaced and or upgraded.

**SOA or Web Services**

The move to SOA did not commence until around four years ago with Web Services - in fact Web Services are a step along a much longer road. The notion of service is an integral part of component thinking, and distributed architectures such as CORBA were early attempts to implement service oriented architecture.

The Web Service is the programmatic interface to a capability that complies with standard protocols, providing the interface technology, delivering platform independence and loose coupling of the transport. SOA is potentially much wider in its scope governing the policies, rules and common services that enable:

- a) a logical service bus structure for use by authorized consumers internal and external to the enterprise regardless of implementation technology.
- b) design and quality of service characteristics that enable widespread use and reuse, genuine abstraction away from implementation technology and conformance with functional and non functional service level agreements.

Enabled by Web Services	Technology neutrality	Endpoint platform independence
	Transport and interface standardization	Standards based protocols
	Separation of provider/consumer	Enabling automated discovery and usage
Enabled by SOA	Functional standardization	Use (reuse) of Service, not reuse copying of code/implementation
	Abstraction	Service is abstracted away from the implementation, enabling technology and application independence
	Formalization of relationship	Formal contract between endpoints, places obligations on provider and consumer
	Relevance	Functionality presented at a granularity recognized by the user as a meaningful service

**Table 1 - Web Services and SOA**

## **Managing Separation**

Agility in both business and technical terms is enabled by managing separation. Clear boundaries reduce the horizon and impact of change, and formalize the interfaces between services. A critical aspect of separation of concerns is to ensure that the provision and supply of services are kept rigorously apart.

For the service consumer, the process must be organized such that only the service matters, and there is no dependence upon knowledge of the implementation. If this can be achieved considerable benefits of flexibility accrue because the service designers cannot make any assumptions about consumer behaviors. They have to provide formal specifications and contracts within which consumers can then use the service in whatever way they see fit. They want to know where it is, what it does, how they can use it. The interface is the only thing of consequence to the consumer as this provides the service.

The provider has a very different set of concerns. The provider needs to develop and deliver a service that can be consumed in a completely separate process. The focus of attention for the provider is therefore the interface – the description and the contract. The nature of the relationship between the provider and consumer is therefore an important issue that needs to be managed appropriately. There are two quite different patterns for the provider/consumer relationship:

### **1. Negotiated - Consumer and Provider jointly agree service.**

When new services are developed there is an opportunity for both provider and consumer to agree how the services should work. In industries where there are many participants all dealing with each other, and where services are common to many providers it is essential that the industry considers standardizing those services. Examples include:

- Close partners that agree the service interface as a natural part of reaching and implementing a commercial agreement.
- The early stages of reaching agreement upon an industry initiative – forming standards for a vertical sector such as Accord for the insurance sector.

### **2. Mandated - This is it, take it or leave it!**

There are many examples of very large, dominant organizations that in business terms dictate business practice throughout their ecosystem. There is every reason to expect this behavioral pattern will apply to service design. Examples of dominant partner patterns include:

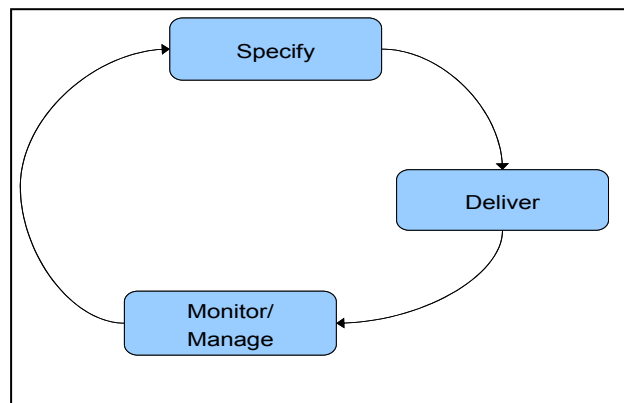
- Provider led situations - such as Ford or GM in a dealer supply chain that say “use this service or we can’t do business”.
- Consumer led situations - such as Walmart or Tesco in a retail supply chain, or Ford or GM with their parts manufacturers, where the dominant partner defines the business process.
- ISV led situations - for example SAP providing standard service interface specifications for ERP customers.



## Service Life Cycles

It seems inevitable that eventually most software capabilities will be delivered and consumed as services. Of course they will be implemented as tightly coupled systems, but the point of usage – the portal, the device, another component or endpoint, will use a service based interface. This implements a separation of concerns such that each service provides a specific capability that has its own managed life cycle, and the minimum necessary dependencies.

The service life cycle provides an integrated view of a functional capability that allows the entire life cycle spanning provider and consumer views - including requirements, design, acquisition and outsourcing, asset management, deployment and resource allocation, to be managed from the business perspective.



**Figure 1 - The Service Life Cycle**

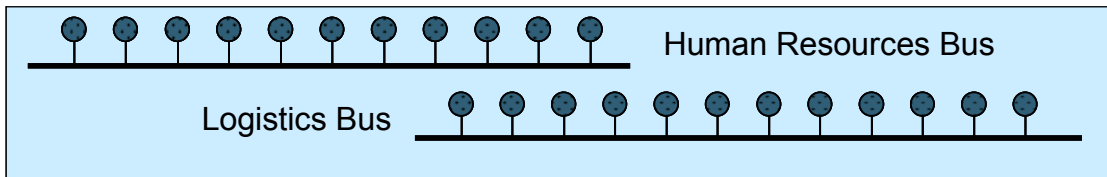
## The Service Architecture

Enterprises typically develop and maintain a series of architectural deliverables to establish strategies, policies, rules and common infrastructure functionality. These include Business, Application, Information and Technical Architectures. To this list enterprises need to add the Service Architecture, in order to manage the service perspective.

The Service Architecture provides the framework for managing the life cycle of services defining the policies and practices, infrastructure and common services for use in specific situations. A useful Service Architecture technique is to develop a Business Service Bus - a logical view of technical and business services illustrated in Figure 2. It helps answer questions such as:

- what services are required by the enterprise ecosystem?
- what services are available to the enterprise ecosystem?
- what services will operate together?
- what common semantics and business rules should apply to specific sets of services?
- what service usage should be standardized/mandated in particular circumstances? (security, management, CRM, product etc)
- what substitute services are available?

- what are the dependencies between services and versions of services?



**Figure 2 - The Business Service Bus**

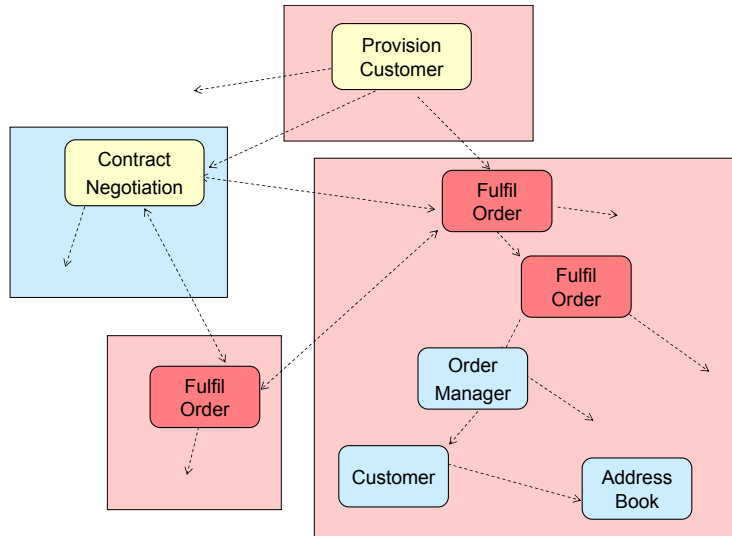
Even more important is that the Business Service Bus is defined **before** projects or programs are chartered, such that critical decisions around standardization of scope, usage and acquisition are made, and that appropriate investment programs are established. Further that critical decisions are made on how existing applications support the new service architectures.

### **Formalizing Service Contracts**

The SOA is effectively a federation of services where each service offers a specific capability according to a contract. The analogy with federated systems of government is appropriate - where parties to a federation form a unified entity for specific purposes, but remain independent in other affairs. The obligations and responsibilities of each party are defined in formal contracts.

The formality of contractual arrangement is important to ensure the agreed functionality is provided, but is also an enabler of service quality and automation. For distributed collaborations, a contract between provider and consumer needs to have three parts. The first part of the contract specifies the function of the service, typically using a series of logical assertions such as pre-conditions and post-conditions. The second part of the contract specifies the quality of service. The third part of the contract specifies the commercial arrangements, including charging for normal operation and compensation for abnormal operation.

Figure 3 illustrates a typical SOA where an enterprise service provisions a customer and has the option to delegate contract negotiation to a third party. Either party can then delegate fulfillment of the order which in turn may be further delegated. Some of these services may be inter-company transactions; others may be intra-company.



**Figure 3 - The Federated Service Architecture**

### **Investment Protection - Using Existing Assets**

Web Services and SOA technologies and approaches are evolutionary - they do not require the wholesale replacement of existing applications. Many early Web Service deployments will be published from existing applications, and provide stable and performant environments. However, as the scale and complexity of the Web Services expand, including advanced behaviors such as for security and collaboration, the reuse of existing functionality requires careful consideration to ensure that existing functionality does not compromise performance or transactional integrity. Further as services become the unit of on-demand management, the monolithic nature of most existing applications becomes an inhibitor to optimizing resources.

To assess existing systems for Web Services and SOA we need to consider a number of issues, all of which have to be related to a specific set of new requirements. The following list provides a checklist of issues to consider:

1. Map existing systems to new service requirements and perform gap analysis in terms of functional and non functional requirements. Particular attention needs to be given to integrated security design, integrated business rules applied at the application and the service orchestration layers, scalability and data integrity.
2. Assess architectural fit - for example what is the roadmap to deliver on-demand Services, and will the existing application prevent resource optimization?
3. Develop an SOA integration plan that enables progressive upgrade of the existing application - for example initially offering key services that provide or support shared services, progressively moving to develop service interfaces for all significant interactions.
4. Determine application boundary changes driven by needs to decompose monolithic applications. Develop decomposition plans.
5. Develop technical integration with service architecture.

6. Define upgrade and synchronization policies between the existing application and any facade and or service consuming applications.
7. Plan a transition strategy for the existing system for each step in the roadmap

**Enabling Response to Change - Delivering Business Agility**

IBM refers to the notion of an On-Demand business - where many different types of resource are arranged in a manner that allows an appropriate level of rapid response to change.

If the business or market operates on a quarterly cycle, then we want something better than an IT response that typically takes 18 months - and then only after a new budgetary cycle. If we have a standard service we may wish to be able to use this in multiple situations by reassembling it into different business processes.

The core value proposition for SOA is that a loosely coupled environment has inherently greater adaptability than a tightly coupled environment, because the loosely coupled components have lower cohesion and the impact of change is minimized. In Table 2 we summarize some of the ways that we expect to see diversity in the on-demand response.

Type	Examples	Response
Physical resources	CPU cycles, bandwidth or storage	Subsecond
Dynamic Consumer Service discovery and consumption	Consumer Services such as Video on demand Marketing Services such as MyMusic Relatively trivial business or personal Services with no security requirements	Minutes
Dynamic Business Service discovery and real time consumption	Commodity process elements - that are published with a status that enables consumers to assemble and use with no communications between provider and consumer other than the published specifications Public Services	Subsecond
Dynamic Business Services assembly and rapid consumption	Sub process elements - that are published with a status that enables consumers to assemble into a new or modified business process. Probably requires communication over and above the published specification Private or trusted Services between known partners	Days or weeks
Change of Business Rules	Update to existing services which requires a service bearing component to be re-implemented can be achieved rapidly because the horizon of change is minimized by design.	Days or weeks
Business Process Outsourcing	Credit card processing Call center management Billing and Collections	Weeks to months
Business Reengineering	Reframing the business process using standardized parts (components and services).	Weeks or months

**Table 2 - Many Types of On-Demand Change**

## Section 4 - The SOA Roadmap

Most new technologies go through a similar process of evolution. In the early stages the new technology gets used for solving today's problems. Then as the technology is better developed and understood it enables radical new applications and patterns. In a final stage the technology is integrated with existing technologies that combine to deliver mature, integrated solutions.

Sometimes it seems that Web Service technologies have been a long time in gestation. But it was always inevitable that adoption of Web Services would be slow because, while the basic concepts and developer tools are quite straightforward, the standardization and management aspects are quite complex. Further, Web Services are the foundational interoperability technology and as discussed, SOA is the **application** of the interoperability functionality that implies considerable change in business and IT practices.

This magnitude of business and technology change requires some level of management. In order to synchronize business requirements, technology capabilities and investments with ROI, CBDI advises on a roadmap approach which identifies stages of maturity for the various areas impacted. In Figure 4 we show a generic SOA Roadmap which links SOA drivers and strategies to stages of maturity.

<b>Drivers</b>	Short term ROI Experimentation	Portfolio Rationalization	Business Reengineering	Industry/ Ecosystem
<b>Service Perspective</b>	Information Access Specific function	Core Business Capability	Business Product Business Collab'n	Domain Standard
<b>Service Deployments</b>	Standalone WS Projects	Architected Shared	Measured	Managed
<b>Collaborations</b>	Opportunistic, Narrow	Many internal Few external	Many external Supply chain	Virtual Business
<b>Service Process</b>	Momentum	Extended Momentum	Reengineered	Automated, Resource Virtualization
	<b>Early Learning</b>	<b>Integration</b>	<b>Reengineering</b>	<b>Maturity</b>

Figure 4 - A Generic Roadmap

### Early Learning

Basic capabilities are put in place largely as experimental activity. Short term ROI and specific project based activity makes this a low risk and impact stage.

Service deployments	Key Roadmap actions
<ul style="list-style-type: none"> <li>- mostly internal</li> <li>- using existing security mechanisms</li> <li>- not mission critical</li> <li>- better application integration</li> <li>- platform neutral information sharing</li> </ul>	<ul style="list-style-type: none"> <li>- Establish understanding of key drivers for SOA - the strategic requirement for portfolio rationalization, information rationalization, process improvement etc</li> <li>- Assess organizational readiness and requirement for preparation stage, establish conditions that facilitate cross organizational learning</li> <li>- Create basic levels of understanding through an appropriate preparatory stage</li> </ul>

### Integration

Once basic capabilities are established the organization is likely to see opportunities from integration with existing core systems. At this stage SOA considerations are required to support shared services and separation of provider and consumer. Some organizations may choose to enable key services which can provide immediate business benefit, others will choose to service enable a complete set of common services which they mandate usage across their ecosystem. This strategy is often highly relevant for large, disparate organizations.

Service deployments	Key Roadmap Actions
<ul style="list-style-type: none"> <li>- business process oriented</li> <li>- project level implementation of shared services</li> <li>- mostly internal usage</li> </ul>	<ul style="list-style-type: none"> <li>- establish common technical infrastructure</li> <li>- establish basic semantics for cross organizational services</li> <li>- rationalize portfolio, expose widely used services for consistency and efficiency</li> <li>- establish basic SLA and management capabilities</li> </ul>

### Reengineering

Once basic business and technical services are available, together with the service infrastructure, many enterprises will see opportunity in reengineering business models using real time or shared service characteristics. Also at this stage it is likely that the increasing business criticality will require more sophisticated management techniques are employed, to monitor and manage service provision at the level of the individual service.

Service deployments	Key Roadmap Actions
<ul style="list-style-type: none"> <li>- common services provided for intra and inter organization usage</li> <li>- services implemented as an integral part of business products</li> </ul>	<ul style="list-style-type: none"> <li>- implement collaborative architectures across organization and with third parties</li> <li>- implement comprehensive and standards based measurement and monitoring systems allow end to end management</li> <li>- implement comprehensive SLAs</li> </ul>

## **Maturity**

Relative maturity of service architecture will be realized when most business products and processes are aligned with the service perspective. In a mature state, there is seamless integration throughout the enterprise and its ecosystem of divisions, customers and suppliers, and there is a high level of automation of the entire service life cycle. There is integrated management of business process execution through to deployed resources.

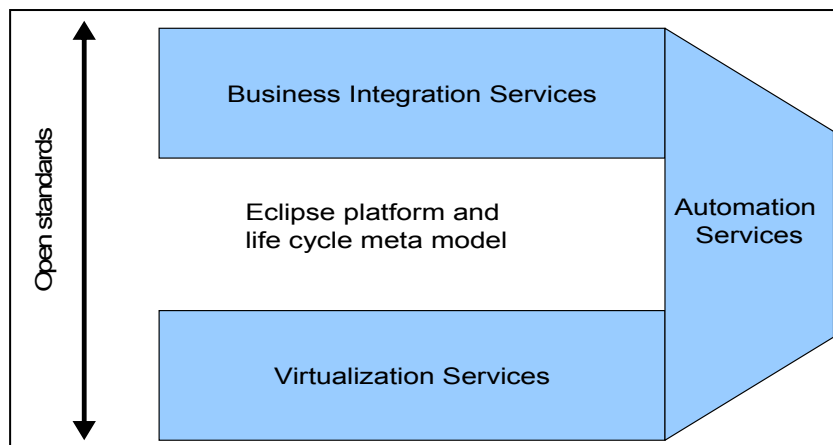
<b>Service deployments</b>	<b>Key Roadmap Actions</b>
<ul style="list-style-type: none"><li>- all primary capabilities organized as services</li><li>- business processes spanning intra and inter organization, are delivered by orchestrating pre-existing services</li><li>- service assets and resources are delivered on demand</li></ul>	<ul style="list-style-type: none"><li>- business service is unit of investment and resource</li><li>- business services and associated assets are monitored and controlled at a business level of abstraction</li></ul>

## Section 5 - IBM's Strategic approach to SOA

The development of Web Services concepts and standards has been led primarily by two major vendors, - IBM and Microsoft. These two major vendors have led the creation of standards in an accelerated process that has been driven by technology specification and development, rather than the de facto standards approach which seeks to arrive at consensus prior to specification. Whether the accelerated process delivers superior specifications may be a matter of debate, but it is clear that the process has delivered an unprecedented technological advance in a very short space of time.

Today all vendors have embraced Web Services. However from this paper it will be clear there is much more to SOA than simply better interoperability. While enterprises will see real value in establishing service interfaces today from the existing application portfolio, over time enterprises will be looking to the vendors for the next level of support for the service concept - automation of the entire life cycle enabling the business perspective to govern the application of rules and resources.

Predictably amidst all the marketing hype it's hard to assess and compare vendor support for SOA a) because it is widespread practice for vendor marketing to treat Web Services and SOA as synonymous and b) the SOA story is, as discussed very much work in progress. To make the comparison we need to examine the vendor strategies, and of course assess their ability to deliver.



**Figure 5 - Life Cycle Integration (Based on IBM concept)**

IBM has made it clear for the last 24 months they are fully committed to a life cycle integration strategy in support of SOA based automation, as outlined in Figure 5. We will consider a number of critical success factors that will have a major impact on their ability to deliver.

### **Business Perspective Drives Automation and Resources**

For several years now IBM has adopted a strong business driven approach to all of its products and services. The IBM definition of on-demand business is where many different types of resource are arranged in a manner that allows an appropriate level of rapid response to change. With the acquisition of Holosofx and the creation of the WebSphere Business Integration Modeler, IBM has established the framework for business monitoring



and management and model based development, which will drive business process implementation from the business perspective.

### **Life cycle automation**

Conventional business IT alignment is no longer adequate. In the SOA world there needs to be an underlying meta model that allows multiple stakeholders to see their own perspective, while sharing common information across the life cycle. Under the Eclipse platform IBM is, together with the open source community, developing the Eclipse Modeling Framework (EMF) which provides the rigorous cross life cycle mapping, which will allow consistency and efficiency between business, developer and deploying functions.

### **Standardization and Abstraction of Service Infrastructure Management**

Service assets and resources need to be monitored and managed in a manner that ensures successful completion and service level compliance. The IBM vision, like other major vendors is to allow users to see a virtual view of the physical resources, so that key monitoring, management and control tasks can be undertaken with a high level of automation from the business perspective. IBM is strongly committed to a standardized view of the service life cycle and is working to deliver the necessary protocols, specifically WSDM<sup>1</sup>, WSRF<sup>2</sup> and GGF<sup>3</sup>, which together facilitate the life cycle management of a heterogeneous system of resources and services. IBM indicates product support for this vision will be delivered on a progressive basis as the standards process evolves. It is a reasonable assumption that IBM will, in due course, deliver an integrated life cycle management system that integrates the tasks of the architect, developer and deployer constituencies, providing automated policy based management of services implemented across heterogeneous systems.

### **Legacy and Existing Portfolio Integration**

Given IBM's breadth of platform support, management of legacy is a major priority. IBM indicates they expect extensions to the current WSRF specification to support legacy resources. Also a critical component of the EMF model (referred to above) is a mapping of the elements required to manage service provision and resource management to each of

---

<sup>1</sup> WSDM - Web Services Distributed Management

An OASIS Technical Committee defining Web Services management, including using Web Services architecture and technology to manage distributed resources and developing the model of a Web Service as a manageable resource. [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsdm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm)

<sup>2</sup> WSRF - The Web Services Resource Framework

A standards initiative by the Globus Alliance, IBM and HP with the objective of defining conventions for managing state so that applications discover, inspect, and interact with stateful resources in standard and interoperable ways. The WS-Resource Framework defines these conventions and does so within the context of established Web services standards. <http://www.globus.org/wsrf/>

<sup>3</sup> GGF - The Global Grid Forum

A community-initiated forum from industry and research leading the global standardization effort for grid computing. <http://www.gridforum.org/default.htm>

the IBM platforms, current and legacy. These models will enable IBM customers to manage services delivered by the various platforms such as CICS and OS400, at the same level of abstraction as the WebSphere platform, creating a virtualized service management platform.

## Section 6 - Summary

The implementation of an SOA is not a quick fix, enabled by the acquisition of the latest silver bullet. SOA is a long term strategy for information provision and process change that will challenge enterprises to reengineer their technology, processes and organization to optimize their systems as federations of relatively independent services. It is reorganizing business and IT systems to be able to continuously adapt to change.

The migration to SOA is also a journey during which many aspects of the environment will change. In the early stages management attention needs to be on basic infrastructure and processes, understanding the optimum organization structure and technology infrastructure which will further enable future change.

The successful approach to SOA will be based upon federated principles of governance, where clear objectives are set, but then implemented with good separation of concerns, enabling strong accountability.

Plans should recognize the strengths in existing and legacy portfolios and build upon them - SOA need not be a replacement strategy, rather a managed, evolutionary strategy which responds to immediate needs by casting solutions in a longer term framework. In many ways implementing an SOA is similar to city planning, where some common services together with well thought through policies facilitate the gradual development of a better environment.

The core assertion of this report is that enterprises can, with proper governance processes in place, create more adaptable information technology environments which will have direct and measurable business value in enabling company-wide transformation initiatives. The transition to Service Oriented Architectures is therefore a critical business issue that requires management time to develop and implement appropriate IT and business strategies.

## Next Steps and Resources

Conduct an internal SOA or application portfolio assessment.

IBM Services offerings for SOA - <http://www-1.ibm.com/services/bcs/webservices.html>

CBDI Services offerings for SOA - [http://www.cbdiforum.com/public/services\\_bro.php3](http://www.cbdiforum.com/public/services_bro.php3)

Recommended resources:

CBDI Roadmap Reports - A collection of practice guidance for enterprises and ISV's adopting SOA. <http://roadmap.cbdiforum.com/>

IBM SOA Resources - <http://www-306.ibm.com/software/info/openenvironment/soa/>

CBDI Research Reports - <http://www.cbdiforum.com>



## *Independent insight for Service Oriented Practice*

### **CBDI Raison d'être**

We aim to provide unique insight on component and web service technologies and processes for the software industry and its customers. To provide high quality analysis and other information resources on best practice in business software creation, reuse and management. To maintain the highest level of independence.

### **Modus Operandi**

CBDI has three channels:

- Subscription services - provision of continuous commentary and information.
- Workshops and seminars - providing in-depth briefing and guidance on advanced architectures, processes and practices.
- Consulting - including related product management and marketing advice, application audit and guidance, technical and business evaluation for investment

### **How we compare with others**

We are not a mass market, media oriented organization. All material published by the forum is unique. The majority of material is published by our own analysts, or commissioned from others, and under strict editorial control to ensure accuracy. We rigorously exclude spurious marketing.

We aim to provide depth insight that covers a narrow topic footprint in a deeper way than the other analysts, and in particular cover not just the technology, but also the architectures, usage, practices and processes.

Also we are unusual as analysts we do not simply echo what the vendors say, we are a think tank, identifying new ideas, opportunities and providing stimulus for thinking.

We try to be thought leaders providing ideas generation and a rich source of conceptual thinking based on practical, real world feedback.

### **Who reads CBDI Journal?**

Technical and Application Architects, Business analysts, Consultants, CTO's, Designers, Product strategists, Senior Developers, Project Managers etc . Subscription is split 40% USA, 50% Europe.

### **Contact us:**

For further information on any of our services contact us at: [info@cbdiforum.com](mailto:info@cbdiforum.com) or +353 2838073 (Ireland)

IMPORTANT NOTICE: The information available in this publication is given in good faith and is believed to be reliable. CBDI Forum Limited expressly excludes any representation or warranty (express or implied) about the suitability of materials in this publication for your purposes and excludes to the fullest extent possible any liability in contract, tort or howsoever for implementation of, or reliance upon, the information contained in this publication.

### **Subscribe to the**

### ***CBDI Journal***

*The CBDI Journal is published monthly. In addition to the CBDI Journal, subscription includes electronic access to all back numbers that now represent a significant resource library. There are individual and corporate subscriptions schemes.*

*Corporate subscription includes access to our workshop materials.*

*For more details and to subscribe see:  
[www.cbdiforum.com](http://www.cbdiforum.com)*