



Service oriented Business Applications

Ravi Koka

Contents

Executive Summary	1
Business Drives Service Oriented Business Applications	1
Evolution of the Enterprise Application Architecture	2
SOA Provides the Solution	3
Intellect SEEC's Perspective on SOA	5
Practical Results with SOA Solutions	8
Conclusion	8

Executive Summary

In today's world, an IT organization's value to the business depends upon its ability to break down existing IT silos and provide the business with the level of agility and flexibility it requires in order to compete effectively and meet emerging business needs. Current states such as "swivel chair integration" where business users are forced to switch from one business application to another in order to get their job done need to be eliminated. Dissolving these silos for the business user is critical and predictably the most dramatic way to both enrich the customer experience and improve overall efficiency and effectiveness. Practically speaking, this means being able to bring applications and their associated functionality to business users in a different way. It means delivering them as discrete shareable services. Not as an array of distributed web services; but rather as a clearly defined and manageable service layer – a layer that allows for true interoperability of existing systems, high reuse, cost effectiveness as well as a high degree of flexibility without impacting the business adversely.

Today, many organizations are attempting to meet the demands of their business by creating business processes that facilitate real time customer and business partner interaction over the Web. The method of choice up to now has been to Web-enable existing applications as driven by customer demand, but this has not allowed organizations to realize the full potential of Web-centric processes and systems. Until recently, the only alternative to Web-enablement was to replace existing systems by building all-new, Web-centric systems from the ground up. While technically feasible, this is not economically practical for companies with significant investments in systems – custom and packaged applications built on different technologies and platforms.

More recently, an application architecture has emerged that delivers the foundation for Web-centric computing while mitigating the costs and risks of replacing existing systems. This Service Oriented Architecture (SOA) allows business users to access software services without worrying about where the software is resident – it allows for a shared service layer of functionality where capabilities are available for use as needed and where needed. This new architecture completely leverages Internet standards and technologies, using XML, Java, .NET and HTTP to restructure the application environment.

In this paper, we explore the SOA-enabled shared service layer and the services which can comprise it and be assembled into Service Oriented Business Applications (SOBAs) to finally deliver true agility to business while leveraging past IT investments.

Business Drives Service Oriented Business Applications

Businesses today are facing the need to improve process efficiencies, sales productivity, revenue growth and cost savings — and many have made great strides by transforming their organizations to better focus on the needs of customers and business partners. However, to win more business from new and existing customers, and to even further streamline supplier relationships, organizations need to create outward-facing processes that meet several key requirements:

- They must create process efficiencies for better productivity and cost management.
- They must be able to quickly adapt to market changes.
- They must give customers and partners access to complete and accurate information in realtime.
- They must satisfy customer demands for personalized information and self-service.

To achieve this, organizations need to shift to a Web-centric business model where transactions are carried out seamlessly across the extended enterprise. This model requires a new information architecture in which applications may run in the network, but where the bulk of the interface and transactions are carried out over the Web, in an integrated, highly streamlined and automated way. Specifically, the new architecture must:

- Support one-to-many and many-to-many business relationships with different classes of users.
- Provide a single portal that enables users to access information based on their unique needs and access privileges.
- Leverage existing application and information assets in the enterprise while enabling business processes to be smoothly shifted to a service oriented model without major software reengineering costs or time delays.
- Enable applications to be easily customized and changed to support on-going innovation and a company's evolving business strategy and brand.
- Enable applications to support multiple languages, currencies and business practices globally.
- Seamlessly integrate other services and applications on-demand.
- Minimize total cost of ownership and implementation risks while creating sustainable market differentiation.

Evolution of the Enterprise Application Architecture

Established companies have several generations of applications that have inhibited their pursuit of Web-centric business processes. In the first generation, the mainframe environment, large centralized computers held all the information necessary for computing, including the database, application logic and user interface — information was centralized. The early 1990s brought a shift to the client-server model where application logic and the user interface moved to the client. While this enabled significant customization and data manipulation, this powerful distributed client base created islands of information that were largely disconnected. This disconnected environment required major data synchronization efforts to provide an enterprise-wide view of a company at any given point in time, which is counter to the goals of the real-time environment. Unfortunately, the vast majority of enterprise systems in place today come from the mainframe and client / server models.

Today, the current three-tier architecture (presentation/logic/data) represents a vast improvement over the traditional client-server model of the past despite the fact that it continues to evolve. The first phase of this architecture is a Web-enabled client-server model where a “thin-client” leverages a Web browser to render media rich applications while business logic and rules reside in stored procedures and packages in servers.

While Web-enablement efforts certainly improve usability of individual applications, they do not facilitate the seamless integration of disparate applications to support complex, Web-centric business processes:

- The disconnected system silos that are prevalent in legacy systems (mainframe-based or client-server) are perpetuated onto the Web.
- Customers are demanding dynamic real-time transactions and information updates, which batch-mode legacy systems are unable to provide.
- Some legacy systems cannot be easily exposed to the Web without significant changes, which may be difficult to justify.
- Most legacy systems, even those that can be Web-enabled, are designed for internal deployment and reinforce rigid business processes that difficult to change.
- Implementing end to end optimized business processes and workflows is difficult

In contrast to Web-enabled systems, the next phase of three tier architecture entails SOBAs which are intended to provide the total integration of business logic and data, spread across an extended enterprise via a unified user interface. Applications may be hosted on a variety of systems, but the key is that they are all integrated to provide unified data and transactions through a unified, customizable user interface. These Web-centric systems will support agile, fully integrated and streamlined processes that are designed to adapt to changing customer and business needs.

SOA Provides the Solution

The emerging service oriented architecture will allow organizations to re-purpose the valuable elements of existing applications and existing data to support new Web-centric processes that enhance competitiveness and efficiency. This new architecture adds a business service layer above the existing applications and packages, providing a uniform method to access information and perform transactions. It unifies processes that span multiple applications, departments and enterprises, helping organizations to increase agility, flexibility and responsiveness to business needs.

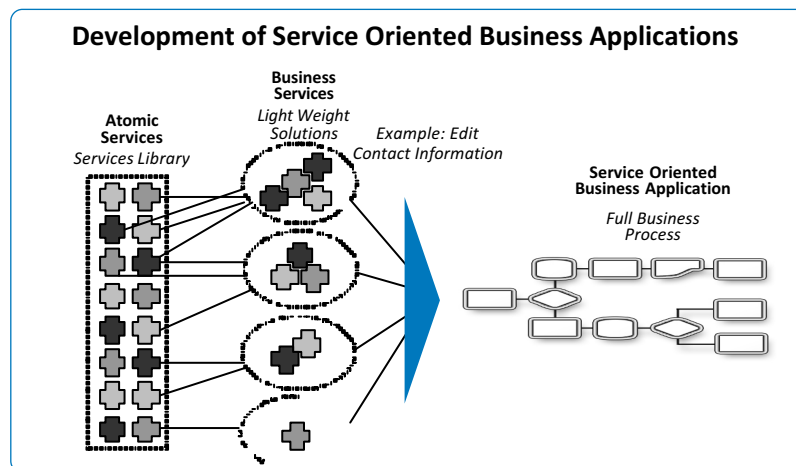
In the shift to this Web-centric SOA, significant computing power is redistributed to an expanded middle tier, where objects from disparate sources can interoperate. The middle tier contains the business process logic as well as business rules, and the database is treated primarily as a repository for data schema. Based on open standards (including J2EE, .NET, XML, and HTTP) the middle tier has the power to utilize multiple disparate applications within a single unified deployment that is driven by user centric business processes – breaking down existing silos and creating considerable flexibility.

In addition, SOA requires a separation of business processes and rules from the underlying application logic giving business users the ability to change business processes with minimal impact to application infrastructures. The separation of the business processes and rules from the application logic also provides a framework for extensive customization based on user needs. Customers, suppliers, partners, and employees can all be served through a common set of application components that are assembled and customized to their requirements in a modular way.

Achieving this architectural framework in the highly demanding Web environment requires deeply embedding Internet technologies including XML and Java. Only by fully leveraging these technologies can a solution be constructed that is agnostic to the underlying applications powering the processes, whether those applications are legacy, mainframe or new generation Internet-ready applications.

From services to SOBAs

The business service layer and SOBAs are an aggregation of atomic services which are reusable across the enterprise. The chart below depicts how atomic services can be grouped:



Atomic Service

These are fine grained services that perform a discrete function. For example, Change Name, Change Phone Number and, Change Address. Typically these fine grained services will be re-used by several business services and SOBAs

Business Service

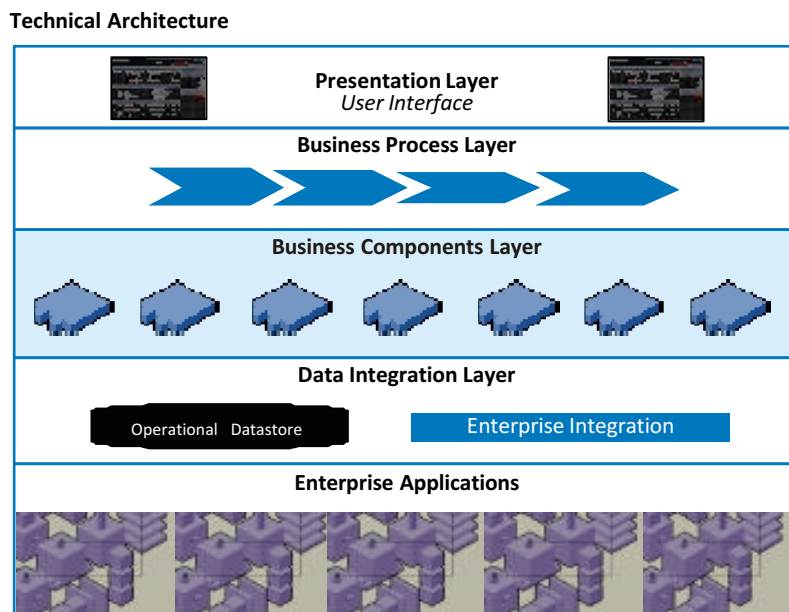
A set of fine grain services maybe combined to provide a business service. Typically these would be a logical grouping in a business scenario. For example, Change Contact Information for a customer will comprise of Change Address, Change Phone Number, and Change Email Address

Service Oriented Business Application (SOBA)

A set of business services can be assembled to build a SOBA. For example these could be grouped by business process for a consumer. As an example, an insurance agent portal will consist of Customer Search, Change Contact Info, Get Quote, Get Policy

Intellect SEEC's Perspective on SOA

Over the past five years Intellect SEEC has delivered solutions that enable Web-centric computing by leveraging legacy assets in a SOA. These SOA solutions enable disparate software products to be easily and rapidly deployed in a common environment, which is also agnostic to any underlying application service, legacy, custom or database technology. Intellect SEEC's technical architecture defines a service layer based on industry standard semantics for such as ACORD XML Schema and messages for the Insurance industry as an example. It allows a large enterprise with many divisions and varied software solutions to deploy unified Web-centric solutions specifically tailored to their business and their customer's unique business requirements as required.



Key characteristics of Intellect SEEC's technical architecture include:

Loose Coupling

The key principles of SOA are the loose-coupling of components (or "services"), and the use of nonproprietary interfaces that allow services built with different technologies to work together seamlessly. Loose-coupling requires business components that implement process and business rules to be independent from front-end services like user interface, security and entitlement, as well as from back-end data sources, including legacy applications. In a SOA, business components, back-end systems and data sources, and presentation devices can each be enhanced, changed or replaced easily with minimal changes to other pieces of the system, ensuring flexibility. Equally important, front-end services, components, and back-end systems can be easily combined and reused, maximizing the value of the software significantly.

Presentation Layer

Separating presentation of the user interface from the business service enables changes in either area to be made independently; delivering a highly flexible applications management environment

that supports rapid change. In effect, the presentation layer is a façade for the business component that is loosely coupled with the user interface, enabling the reuse of the business component for noninteractive, business-to-business services as well. The presentation layer is not monolithic but is built as a set of components (such as portlets) that can be flexibly combined to create a dynamic, highly configurable user interface that is personalized for each user community.

Semantic Mapping and Transformation

The data elements in the presentation layer are typically terms that are familiar to the consumer. In order to facilitate a standard way of processing requests from consumers (internal and external) it is important to adopt an industry standard semantic model (object model) such as ACORD (insurance). This requires transforming the input from the presentation layer to a standard schema using the semantics defined in the industry model. Further the XML message requests need to be transformed to the underlying object model in order to be able to perform the necessary operation on the transaction and data layers.

Business Components

Business components provide a logical abstraction of the physical reality. Business components provide meaningful business services by hiding the number and types of backend systems. This makes developing the presentation layer simple and uniform even when backend systems are evolving and/or replaced with newer packages. Business components are responsible for fulfilling the business transactions by performing appropriate functions on one or more relevant back-end systems.

They are also responsible for triggering appropriate business rules to ensure that the transactions happen as intended by different consumers of the business service.

Business Rule Framework

Business rules are fundamental policies which drive business processes. Historically these rules have been embedded within applications, which make changing and maintaining rules difficult and costly. In today's highly competitive business environment, organizations can no longer afford to wait for longer implementation cycles to adapt changes in business policies. To gain and maintain a competitive advantage, companies must be able to immediately respond to the requirements of their customers.

With SOA as the paradigm, a more consistent and cost effective approach is to externalize the business rules using external rule engines. However there are many rules for which employing an inference rule engine becomes unnecessary. With the help of a Business Rule Framework, simple business rules could be externalized and the same framework could also be used as a conduit to invoke rule engines to process complex rules like “eligibility rules” in the insurance industry. Externalized and declarative rules within this framework used in conjunction with atomic business component give flexibility for organizations to react quickly to dynamics of the business. The following are the key features to be considered for a Business Rule Framework.

- Rules should be declarative and configurable
- The business component should be able to invoke different set of rules for different consumers. Example: The rules could be different for an internal employee performing a transaction vs. an external agency performing the same transaction
- Business Analysts and IT personnel should be able to define, deploy and maintain rules faster and easier
- The framework should be able to invoke an external rule engine to process complex business rules

Data Integration Layer

In order to support the business service layer a robust and flexible data integration layer is extremely important to realize the full benefits of SOA. Connectors and adapters retrieve data for business objects and enable persistent workflows across disparate applications. Connectors are data source and protocol-specific. They enable developers to declaratively map the business object attributes to the backend objects including database and other data sources in legacy systems. Adapters are business object-specific and act as the logical aggregator for data accessed from disparate back-end systems. Adapters also manage validation, caching and two-phase transactional semantics.

Important features which should be considered for a robust data integration layer include:

- Ability to adapt to changes to the data model
- Ability to interface with different data formats and storage mechanisms
- Ability to handle data from various sources both internal as well as external
- Ability to handle batch transactions as well as online or real-time transactions

The loose coupling of the business logic and data layer allows the data model and the business services layer to evolve independently. The adoption of a common data model or canonical model, preferably an industry standard model where applicable, will allow for supporting multiple physical data models. This is achieved by mapping the various models to the canonical model and creating a single integration point at the business services layer.

Open Standards

SOA solutions should be built around open industry standards such as J2EE (Java), XML and HTTP, SOAP and UDDI interface standards for Web Services to ensure that the developed solution can be deployed on a wide variety of technology platforms.

Practical Results with SOA Solutions

The service oriented architecture represents an evolutionary technology that removes many of the risks and costs associated with the shift to a Web-centric business model by substantially speeding time-to-benefit and dramatically lowering on-going maintenance and operational burdens. SOA serves as the foundation for all processes in the product life cycle, tying them together and seamlessly delivering them to each user. Using this new architecture, Intellect SEEC delivers applications that can be changed to fit evolving business processes, geared toward end user specifications without any major reengineering or custom code development. These solutions enable enterprises to realize the benefits of Web-centric computing by linking all their suppliers, partners, customers and employees into a single value chain, and to do it faster, at lower cost, and in a flexible, future-proof architecture that supports agility.

Intellect SEEC's service oriented architecture solutions provide enterprises with an expanding set of opportunities to extend their reach with suppliers, partners, customers and employees

across the product life cycle. Information is immediately available, transparent to all users and tailored to their specific business requirements. This real-time information environment enhances key supplier and partner relationships and overall customer satisfaction, increasing the enterprise's brand value. The shift to real-time, Web-centric processes has yielded impressive results for Intellect SEEC customers:

- Order entry times reduced by up to 90% versus manual or phone-based processes.
- Order errors minimized to less than 1% of total orders.
- Third party shipment of products enabled, reducing delivery time by 30%.
- 80% of customer support activity shifted to self-service.
- 40% of customer service/sales staff redeployed to higher-value activities.

Conclusion

SOA redefines the cost, time and risk calculations in making the shift to Web-centric computing. The focus on aligning IT to business requirements makes the definition of the business service layer the key element to ensuring a successful implementation. Getting buried in tools, technology and platforms will not gain the trust of CIOs and business executives who have been burnt again and again with promises of the silver bullet from IT vendors. The proof of value is in demonstrating that business applications can be built and deployed rapidly and can be modified easily as business requirements change. Starting with a small application and establishing the success criteria would be the right way to set the stage for moving the current assets to an SOA framework.

Ravindra Koka

COO (Chief Operation Officer), Intellect SEEC Inc.

Mr. Koka is a co-founder of Intellect SEEC and served as the company's President and Chief Executive Officer from 1988 until 2003. Prior to founding Intellect SEEC in 1988, Mr. Koka was a Visiting Scientist and Adjunct Professor at Carnegie Mellon University, where he developed an incremental transformation paradigm that provided the foundation for Intellect SEEC.

Intellect SEEC provides service-oriented business applications using the Intellect SEEC Advantage Library™, the world's largest collection of standards-based, reusable software components for the insurance and financial services industries. The Intellect SEEC Advantage Library is the foundation for a truly customer-centric approach, offering companies the freedom to choose only the software functionality they want, license the source code, customize and deploy in less than 90 days, and then reuse across any channel, any product, anywhere in the business. At a price that is a small fraction of the cost of a proprietary solution, it's the agility, speed and cost savings to create strategic business advantage from IT. Designed specifically for insurance and financial services, Intellect SEEC's service-oriented applications deliver process-rich sales and service capabilities that empower agents, producers, and service staff to work faster and sell smarter across all channels, products, and lines of business.

Polaris Software Lab is a leading Financial Technology company, with its comprehensive portfolio of products, smart legacy modernization services and consulting. Polaris is a 300 Million USD company with a talent strength of over 9,000 solution architects, domain and technology experts. The company owns the largest set of Intellectual Properties in the form of a comprehensive product suite namely **Intellect™**, the first pure play SOA based application suite for Retail, Corporate and Investment banking.

Polaris is headquartered in Chennai with offices in all global financial hubs including 30 relationship offices across 16 countries, 11 development centers (India) and 4 near shore centers in Toronto, Belfast, Sydney and Singapore.

POLARIS

live your dream

A Financial Technology Corporation

Polaris Software Lab Limited

Regd. Office: 'Polaris House',
244, Anna Salai, Chennai - 600 006, India.
Tel: +91-44-3987 4000, Fax: +91-44-2852 3280
email: marketing@polaris.co.in

- Belfast • Chennai • Chicago • Dubai
- Dublin • Frankfurt • Fremont • Hong Kong
- Hyderabad • London • Madrid • Manama
- Melbourne • Mississauga • Mumbai • Neuchatel
- New Delhi • New Jersey • Riyadh • Santiago • Seoul
- Singapore • Sydney • Tokyo • Toronto
- Utrecht • Vincennes • Wicklow

www.polaris.co.in