

The slide features a decorative design with three orange circles of varying sizes, each with a white ring and a drop shadow, arranged in a descending staircase pattern from top right to bottom right. Two thin blue lines intersect at the top left, forming a large 'V' shape that frames the text area.

Business Empowerment by SOA

How Legacy Integration makes business agile

The core benefits of SOA: interoperability, reuse is making business very agile and flexible. Companies have invested billions of dollars into the legacy applications that they can't simply retire. Standards based legacy integration will transform business to achieve greater productivity and utility.

Arkin Software Technologies
3/26/2008



Table of contents

Executive Summary.....	3
SOA Defined	4
SOA Success Factors.....	4
SOA Pitfalls.....	6
Legacy Integration.....	7
SOA Methodology.....	9
Gather Phase.....	9
Explore Phase.....	10
Build Phase.....	11
Deploy and Maintain Phase	11
Business Process Modeling (BPM)	11
Development Approaches	12
Composite Applications	13
Summary	14

Gartner estimates that by 2010, 80 percent of the mission-critical operational applications and business process applications will be based on SOA.

“Cost reduction, increased asset reuse, increased business visibility, and greater business agility – the four core benefits of SOA are giving rise to an additional benefit: business empowerment”

--Zapthink

Executive Summary

Today, service-oriented architecture (SOA) is a mainstream IT initiative that ranks among the top priorities of almost every organization. SOA dramatically improves the flexibility and adaptability of organizations by accelerating the time to market for new applications and processes, driving down IT costs by making services highly reusable and building business processes to support change. As a result, organizations can react faster, seize new opportunities and respond more quickly to competitive threats.

Although SOA is commonly associated with the creation of new systems, SOA and integration are also closely linked. A key driver of SOA is interoperability, which of course is at the core of integration. By adopting Web services standards to achieve a greater degree of system –to-system interoperability, SOA will ease the integration challenges. SOA will not replace the need for higher-level integration capabilities. Enterprises can achieve higher level of flexibility and reuse by taking a SOA approach to implementing integration. In essence, SOA makes integration easier, and integration technologies enable SOA to achieve greater productivity and utility. Legacy applications integrated in a standards-

based approach will make the SOA orchestrations using BPEL very easy to maintain. Composite applications will enable the organizations to reap the benefits of investments made in service-oriented business processes.

SOA Defined

SOA, in its purest form, promotes across-the-board reuse of IT assets, across-the-board standardization so systems inside and outside the business can exchange data and launch services, and breathes new life into the billions and billions invested in back-end legacy systems over the past century. However, it takes time to reap the benefits of SOA. Organizations cannot expect to see a turn-around in the business agility and efficiency right away. Organizations should keep one thing in mind, SOA is neither a product nor a piece of software; this is just an architectural concept.

The service-oriented approach delivers IT systems as a set of reusable services that can be assembled easily to create a composite application that automates a business process. This “assembly approach” accelerates the time to market for new applications and reduces IT costs. SOA better aligns business and IT and shifts the focus away from the nuances of underlying technologies toward abstracted services that make sense to the business. These abstracted services are often called “business services.”

SOA provides significant benefits by offering the following:

- **Reuse** of components, packages, and legacy applications—contributes to increased IT productivity (reduced project costs, faster cycle time) and lower maintenance costs.
- **Business agility**—enhances an organization’s business responsiveness, operational effectiveness and efficiency, and its ability to innovate.
- **Technology independence**—fosters independence from software vendors and enables the organization to choose best-of-breed products in growing the IT infrastructure.
- **Monitoring**—or the ability to monitor points of information and points of service in real-time, to determine the well being of an enterprise or trading community. Detailed performance metrics can be used to track traffic, response times and other key performance indicators (KPI).
- **Extend reach**— or the ability to expose certain business services to other external entities for the purpose of inter-enterprise collaboration or shared processes.

SOA Success Factors

The following section provides a list of factors that are critical to the success of SOA adoption:

Select a small SOA project with greater ROI

In most of the cases, the single cause of failure for SOA is inappropriate scoping of the SOA project. Companies too often seek to make SOA an enterprise-wide effort, even though the business case for that is typically not justified. The rationale goes that SOA is an aspect of Enterprise Architecture and therefore its scope is enterprise-wide, or because it is so important and strategic, it must be implemented at an enterprise-wide level. SOA is simply not appropriate for all problems, and even for problems that need to be solved enterprise wide, not all parts of the solution should be Service-oriented. A good enterprise architect knows how and when to apply SOA. Managing the Service lifecycle, including continuous quality, Service modeling, governance, and management is the major problem. When companies seek to implement hundreds of unproven Services against a business case that is not justified using millions of dollars of untested technologies, they risk significant failure. This will set up the SOA for failure.

On the flip side, well-scoped SOA projects are often remarkably successful. Most case studies of SOA success relate to organizations fixating on a particular business problem, perhaps at even the departmental level, and solving that in a Service-oriented way. The champions of SOA know that success comes from focusing the solution on a particular problem and solving it well.

Foster Team Approach

The problem with current state of expertise is that very few individuals have the skills to understand the business combined with the technical acumen necessary to understand how SOA best practices can drive business solutions. In most successful SOA projects, there exists a strong, multi-role, cross-functional team that helps to provide all the perspectives needed to understand the full scope of enterprise architecture.

IT departments have necessarily grown into isolated organizations that have neither the skills nor resources to appropriately manage and view all aspects of the organization. While one way to get that visibility is to hire enterprise architect experts that have the necessary skills to see all parts of the business and technology, getting those resources will become an increasing challenge. Most firms can achieve the same effect simply by building cross-functional teams that include line of business representatives, application development, data modeling, process modeling, security, and network operations roles, just to name a few. While it is certainly preferable to have a knowledgeable EA individual on staff, sometimes the team is necessary. Even EA Teams need to be guided by best practices and have validation and auditing, especially when there's no single individual with EA expertise.

Lines of Business Champions vs. Tech Insiders

Many successful SOA projects can identify one fact that someone from the business has identified a specific need that can only be solved by one of SOA's four points of ROI. Where business can see a solution, sometimes IT is blind.

The IT organization, more often than not, resists implementation of SOA even in the face of compelling business propositions. Technologists often get stuck in defensive positions regarding particular

technological approaches (REST vs. Web Services anyone?). While technologists pride themselves in technological understanding, they are often the ones that get suckered into buying the Vendor's story. While technologists understand little about the business, they can easily understand a technology story, which vendors tend to pitch. Business users already abstract technology in their thinking. In their mind, all technologies are alike as long as they meet the business needs, policies, costs, and timeframe. This makes line of business champions the most successful starting point for SOA. Where an LOB champions SOA, we see rapid success. Where the success of SOA depends on an IT-only champion, we see SOA projects stuck in the IT-political and technology-religious quagmire.

Technology Best-of-Breeds

SOA thrives in an environment of heterogeneity and continuous change, and as such, proper architectures abstract technology implementations, which would mean that the value of a suite in a SOA is no greater or less than the value of best-of-breed components. Technology change, like business change, is continuous, and as such it makes little sense to solidify the architecture to a particular technology implementation. Yet, too many organizations let the vendors drive their architectural decisions.

Not all technology providers are pushing their own story that is so detrimental to the long-term success of SOA. Best-of-breed SOA infrastructural technologies aim to solve point problems such as simply providing reliable Service intermediary capabilities, policy enforcement, or metadata management. Such technologies and vendors are not in the business of trying to lock you into their suite and will correctly advise their customers that the success of the architecture is entirely the responsibility of the end user. Where companies treat technology suppliers as simply enablers and not as packaged "architecture" vendors, we see success. Where companies come to rely on vendors to solve their problems without requiring the customer to create and manage their own architecture, we see problems.

SOA Pitfalls

SOA sure sounds exciting for organizations to get started on that path. However, organizations have to separate the substance from the hype with thoughtful planning and risk awareness. Here are few common pitfalls to avoid:

Attempting Large Initiatives – Do not attempt to convert the entire enterprise to a new architecture in the first phase. Most successful SOA projects are the ones that begin with fewer than dozen services, integrating a handful of systems, and are completed in less than 6 months. After the initial success, these projects typically grow incrementally.

Thinking SOA as a technology – Do not think SOA as a technology. It is just a concept that guides how organizations design their architecture and how to choose technology to facilitate reuse. Focus on establishing reuse, standards based integration and loose-coupling.

Departments working in isolation – IT managers alone cannot analyze business processes to improve efficiency any more than business managers can redesign systems for a better bottom-line. Business and technical departments should work closely to realign their priorities and bridge the gap to be agile.

Replacing Legacy Applications – Do not rip and replace existing legacy applications, it can be very costly. Instead open these monolithic applications to SOA by wrapping them as web services.

Ignoring Web Services security – While many SOA projects begin behind the firewall, adding business-to-business connectivity is a natural extension and often a core benefit. It is important to ensure that an environment includes enterprise qualities of service, or can be easily extended to include them later.

Legacy Integration

The word integration triggers an array of impressions ranging from trepidation—about configuring connections between a variety of applications and working with complex tools—to frustration about large projects that run over budget and schedule. And while the road to IT efficiency is littered with failed integration initiatives, it remains IT's holy grail to use integration to achieve that efficiency.

The path to IT efficiency is looking much clearer today however, now that technologies have turned a corner away from complex and rigid integration solutions. IT has a new approach that is service-oriented, open, and extensible that promises to ease integration challenges, making the rapid creation and delivery of new composite applications and business processes attainable. Today's integration solutions are providing IT with the tools and technologies to infuse fluidity—a combination of flexibility and responsiveness—into its infrastructure.

Before legacy applications were considered the participants in the SOA, solutions involved moving applications off the legacy systems onto the application server to expose them to the Web. This is as expensive as it is time-consuming, and by removing this step, enterprises stand to save some serious money.

Businesses also prize increased agility, essentially defined by speed to market. If businesses can reuse the proven solutions on the mainframe or access them faster, they stand to become leaner, meaner, and richer.

Reliability is the big plus in the legacy application's especially mainframes favor. Developers are well-acquainted with the solid, time-tested services on their mainframes, and reusing them instead of relying on less reliable distributed platforms increases stability in an agile, cost-saving environment.

The number of application interfaces and data formats that developers must learn and support affects the complexity and flexibility of integration. The more interfaces and formats the greater the costs to implement and maintain an integration framework. Web services provide a common interface to applications and XML provides a common data format — together, they simplify integration and lower integration costs.

Traditionally, developers create application programming interfaces (APIs) for each application in the integration project to allow inter-application communication.¹ Because the APIs for each application are different, companies integrating large numbers of applications develop complex integration frameworks that require specialized development skills and lead to increased integration costs. Increased complexity also makes integration architectures less flexible, so that changes in the network and business processes (following M&A, partnership changes, etc.) are slow and cumbersome.

Reducing the number of APIs reduces complexity and increases flexibility. Web services based on Internet technologies are not specific to any platform and provide a single API for any application to use. They are loosely-coupled and can be invoked directly as traditional APIs or requests can be sent to a queuing system using where transactions can occur at specified dates or times.² Web services also provide greater flexibility when designing integration architectures. Because web services use a common interface, changes to the network or business processes do not affect the ability of individual applications to communicate with each other. Another source of complexity is the proliferation of data formats used by each application. Reducing the number of data formats reduces the number of data transformations that take place as information passes through an integration framework. It is well known fact how a common data format based on XML can reduce the complexity in an integration framework. However, it will be a very long time, if ever, before all enterprise data is accessible in XML format, the data format for Web services. Different applications – built with different business requirements for different purposes- define business entities (such as customers, products, or financial securities) their own way.

As shown in the picture(Figure 1 Enterprise App Strategies for 2008) below, enterprises are realizing the importance of integration as their strategic priority to leverage existing investments and to adopt SOA strategies.

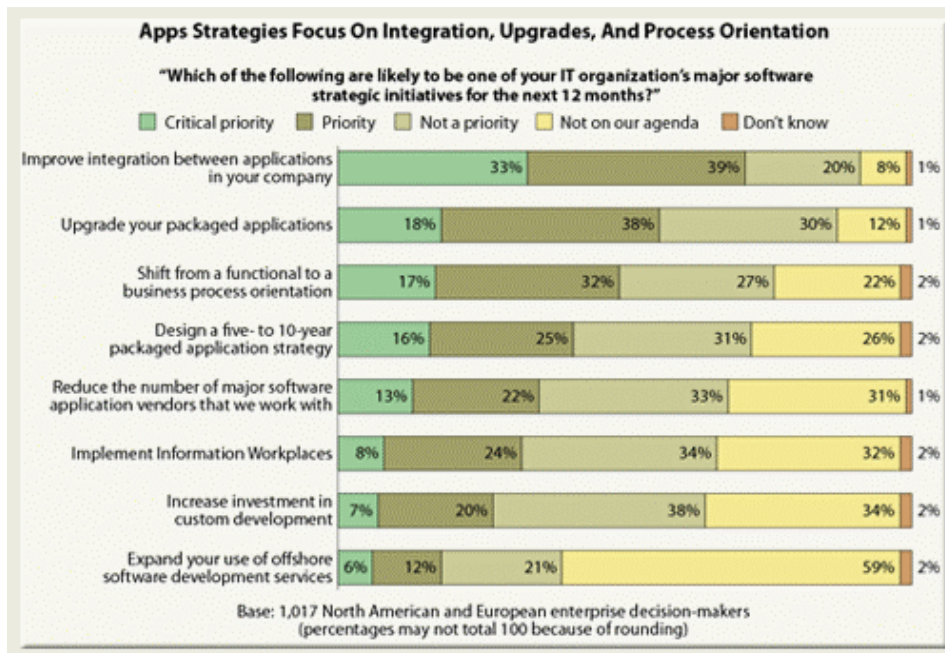


Figure 1 Enterprise App Strategies for 2008

SOA Methodology

Moving to SOA is not just about implementing new technologies, defining interfaces and creating services. It's also about creating, fostering and enabling a more collaborative development process with best practices and repeatable frameworks. The following methodology practiced by Arkin is a platform-agnostic framework, because this will survive the constantly evolving technology platform implementation. This defines a process that spans from pre-production planning, development, testing and production deployment. Once the overall SOA based models are agreed upon, these can be transformed into platform-specific models using standard transformation rules and exchanged using a standard interchange format.

The SOA methodology consists of four phases:

- Requirements Gathering - Inception, SOA Vision, SOA strategy, Project planning
- Explore – Detailed analysis, service discovery and design of the solution
- Build – Development, Create Meta data library
- Roll Out, Maintain – Operations Management and support

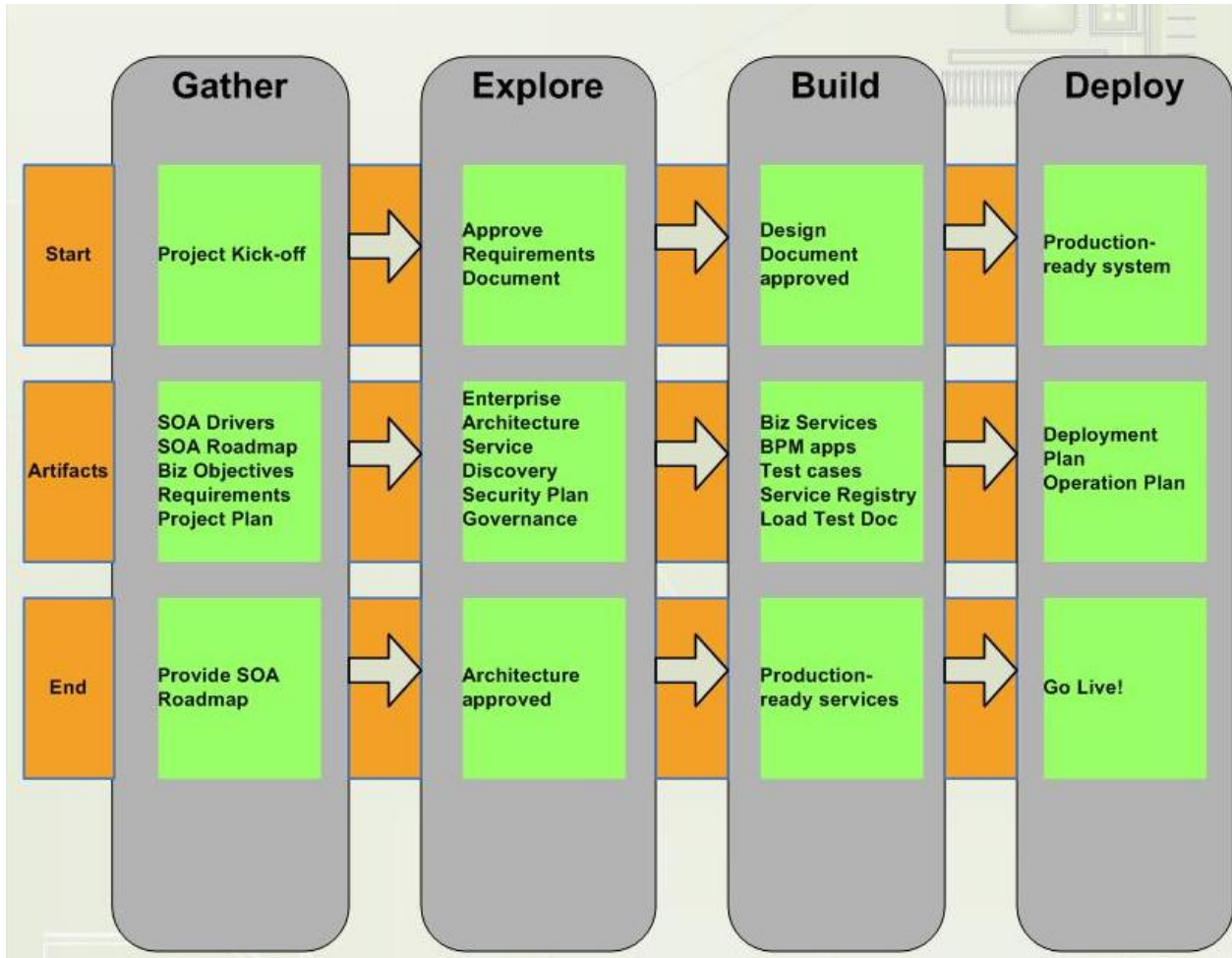
Each phase consists of a number of activities that require inputs, such as business needs and integration goals, and deliverables, such as design documents or a configured development environment. SOA Governance will be part of all the phases of SOA life cycle.

Gather Phase

During this phase business stakeholders and technical analysts will work together to identify and prioritize the business needs. Based on the business requirements business analysts will document high-level business processes, rules and requirements. This document includes the following:

- High level view of the overall business processes
- Definition of individual business process and the identification of the owners
- Identification of objectives and gaps in the current business service gaps
- Understanding of current and planned business goals and IT initiatives

- Identify initial candidates for Pilot projects for SOA adoption
- Estimation of effort, period and cost for SOA adoption, for the identified initiatives
- Roadmap for immediate-term and long-term vision for SOA adoption



Explore Phase

The Explore phase begins when the Customer approves the Requirements document. It ends when the Architecture/Design documents are reviewed and approved by the Customer team.

In this phase, the requirements and conceptual architecture are taken as inputs and the solution design activities are carried out. The activities include identification of various components such as services, wrapper services, transformation and rules and of various orchestrations among them. The activities also include the identification of utility services for various business processes, such as audit trail, error

and exception handling, and the design of related components. Service implementation rules and governance policies are identified.

Build Phase

The Build phase begins when:

- Requirements and Design documents are approved by customer
- Development environment is available

The Build phase ends when the individual services are tested and the required business solution is integration tested.

The Build phase involves the development of integration services as specified in the Design Documents. It also involves the building of common frameworks, building of process models, configuring of adapters, building of integration logic to handle data transformations and validations, custom building and testing of wrapper services. Services are tested for their reuse and any performance requirements.

Various applications and systems are integrated as specified in the integration requirements. A testing strategy and plan is drawn to facilitate the testing processes.

Deploy and Maintain Phase

The Deploy phase begins when:

- Acceptance Testing is complete and the system is ready for deployment
- The Deployment / Operational manual is available

In Deploy phase, Arkin's team working with IT operations is responsible for configuring the staging and production environment with the production environment taking the highest priority. The business solution will be deployed following the guidelines outlined in Rollout Manual.

During the Maintain phase of the life cycle, Arkin provides monitoring service to provide metrics to the business to track business KPIs (Key Performance Indicator) and review IT SLAs (). The operations guide documents the process for adding new consumers for the services without reducing service performance and availability. Guidelines are established for deploying new versions of services without disrupting consuming applications.

Business Process Modeling (BPM)

Businesses focus on processes and the impact of enabling such processes to achieve business objectives. Their language is the language of business operations and BPM, and they seldom think about the complexities of implementing business processes across IT infrastructure. IT, on the other hand, thinks in terms of service enablement, system and application resources, and underlying infrastructure. IT may

struggle with translating its infrastructure expenditures into business benefits and as a result, has difficulty trying to justify investments in modernizing infrastructure for greater agility and business responsiveness. IT also struggles with constantly changing business requirements because it lacks insight or input when the requirements are created. As a result, IT can't get the requirements quickly or clearly enough to translate them across its infrastructure in a timely manner.

IT's investment in enterprise integration can help solve the above problems, but only partially. IT will have a more flexible, dynamic infrastructure base, but until it is brought into the iterative process of understanding business requirements and business-process fundamentals, there will be a gap between what business wants and what IT interprets and can deliver. Embracing business integration solves this problem. Business integration is achieved by embracing two fundamental changes across the organization. The first is organizational: creating the support and processes for interaction between business analysts who identify and define business processes, and the IT developers who implement integrated processes and services. The organization must fundamentally commit to connecting these two groups so that business analysts embrace IT's role in the business process lifecycle, and IT can utilize the process models and requirements the analysts generate in an open and iterative fashion

The second fundamental change lies in infrastructure—investing in tools that seamlessly integrate the process models created by business and IT implementations. It is this second enabler that is addressed by business integration infrastructure, which combines the capability of a business process management suite with enterprise integration for SOA. BPM is aimed at enabling the business analyst to model their current (as-is) and new (to-be) processes using the modeling environments most familiar to them, simulate their processes to test for effectiveness and impact, define their processes, and manage business processes proactively, feeding business activity management information back to the analyst for ongoing optimization. When BPM's power is combined with the power of enterprise integration for SOA to have IT translate business processes into integrated, mediated services, the business can more rapidly create, deploy, and optimize processes that can truly transform it. Arkin's BPM experts can translate business models to artifacts that can be discovered, developed to access backend resources and data, mediated, and interconnected into deployed composite applications. Once business process models are implemented and automated across the SOA's enterprise integration infrastructure, optimization occurs when runtime feedback is returned to the business analyst via integrated business-activity monitoring. This lets business users see where process improvement needs to occur, in real time. Once improvements are identified, the business analyst can update both the models and the business, and the development cycle begins anew. True business transformation and optimization are realized through this iterative business-integration cycle.

Development Approaches

Top-down Approach (Recommended for new development efforts)

This approach is also known as "analysis first" that mandates the organization's business model to be derived of business processes that are service-oriented. This approach results in the creation of numerous reusable business and application services.

In the top-down approach, the analyst draws a diagram of the business process. The analyst assigns data or documents with placeholders for specified fields. Top-down modeling does not require that the analyst have development skills, nor does the analyst need to know precisely which systems exist. The analyst must, however, be able to diagram and envision the process flows.

Bottom-up Approach (Use only for orchestration of existing services into a process)

Legacy integration is the key motivator for the bottom-up design strategy. Legacy application services are modeled as Web services to encapsulate the application logic to best serve the immediate requirements of the solution.

These Web services developed following open SOAP standards can be incorporated it into the business process model. This approach requires the analyst to have more development skills than does the top-down approach. Generally, these services are not offered at the right level of granularity for business analysts and this limits their potential for reuse.

Arkin's Fusion Approach

Arkin had been very successful in combining the top-down and bottom-up approaches. This is most recommended approach for organizations that have been doing EAI for an extended period of time and have compiled a substantial code base. In this hybrid model, existing services are wrapped within business process steps to provide visibility and orchestration into what was previously a process built from unassociated service calls. Wherever a BPM solution is approached by bottom-up service enablement coupled with top-down process engineering, SOA projects delivered the highest return.

Composite Applications

The immediate gain organization can reap from SOA is the ability to extend integrated processes and applications as services, so they can be loosely coupled for reuse in new and evolved composite applications. Composite applications refer to an application assembled from components that may originate from various underlying systems. Composite applications enable new business solutions to be created by flexibly combining new and existing application functionality.

Much of this is possible because of new open standards, which abstract away the entire technology stack, so it doesn't matter what hardware, operating system, programming language or other products are being used to provide the services. The composite application can interact with any standards compliant service. Technologies like XML, Web services (SOAP, WSDL, etc.) and others make this possible.

This level of interoperability will usher in a new wave of innovative applications that were not possible before. For example, if you had to wake up at 5:00am to catch an 8:00am flight, wouldn't it be nice if you had an alarm clock that was smart enough to check the status of your flight on united.com before waking you up? This type of integration is right around the corner. While this is a whimsical example, it shows the vast opportunity for people to creatively link mundane applications together in new and interesting ways.

While SOA standards and the use of metadata promise to make assembling composite applications easier through compositional approaches, the automation of business functions will always have inherent complexities that require extremely fine control, such as atomic transactions that comprise multiple steps involving heterogeneous application access and synchronization. Successful enterprise integration for SOA must let IT move seamlessly between composing coarse-grained service interactions and drilling down to the fine-grained aspects of a service-enabled process. Arkin's approach to Composite Application Development follows the same methodology as outlined above. The key differentiator is our ability to design most sophisticated user interfaces with the latest technologies like AJAX, GoogleGears and Yahoo User Interface Library, etc. SOA combined with RIA (Rich Internet Application) will deliver loosely coupled web applications for different media and devices to empower the business.

Summary

SOA has become a strategic imperative for organizations today. Organizations without a strategy for SOA risk being outpaced and outperformed by competition. Organizations made billions of dollars of investments in the legacy applications, packaged, and custom-built applications. Standards based integrations could bring these applications into SOA fabric and make them first-class services for composing BPM and composite applications. Arkin Software Technologies, a CMMi Level 3 & ISO 9001:2008 certified company provides a complete solution across the lifecycle of SOA. Combined with our Global Delivery Model, we are capable of helping IT organizations realize the benefits of SOA. Many global companies are already leveraging Arkin's expertise in SOA implementations to integrate legacy applications and create more agile business processes.