UNTANGLING THE APPLICATION MORASS
BUILDING AGILE COMPOSITE APPLICATIONS WITH SOA
UNTANGLING THE APPLICATION MORASS
BUILDING AGILE COMPOSITE APPLICATIONS WITH SOA
August 2004
Analysts: Jason Bloomberg and Ronald Schmelzer

Abstract
The complexity in many IT shops today is reaching epidemic proportions. After years of tactical purchases, company mergers, and shifting business requirements, today’s IT environment is heterogeneous, brittle, and expensive to maintain.

Such complexity presents users of applications with a number of challenges. People must often work with several different applications at once, entering data from one system into another, keeping track of business processes in their heads. Such confusion leads to errors, lost productivity, and customer dissatisfaction.

Service-Oriented Architecture (SOA) is an approach to abstracting such complex application functionality in a way that enables users to create composite applications that bring such diverse functionality into a single interface. SOAs can thus help companies automate business processes in a flexible, agile manner, reducing the costs and risks inherent in today’s heterogeneous environments.

Jacada’s Fusion product helps companies achieve the agility benefits of such composite applications by providing a non-invasive abstraction layer on today’s heterogeneous applications, including Windows, Web, and enterprise applications. Jacada thus enables companies to compose new applications from existing applications, whether they be Windows-based or Web-based. By bringing these diverse application interfaces together into services that users can incorporate into composite applications, Jacada helps companies reduce the cost and complexity within their IT environments.
# Table of Contents

I. **Composite Applications and Service-Oriented Computing** ............................................................. 4  
   SOAs in the Enterprise ........................................................................................................... 4  
   The Business Motivation for SOAs ....................................................................................... 6  
   The Economics of Business Agility ....................................................................................... 6  

II. **SOAs and Business Process** ............................................................................................... 7  
   The Importance of Business Process .................................................................................... 8  
   Connecting Business Requirements to IT Capabilities with Business Process ................. 8  
   Loosely Coupling Business Process from Application Logic .............................................. 9  
   Building and Running Composite Applications .................................................................. 10  

III. **Jacada: Non-Invasive Composition of Existing Applications** ................................................ 11  
    Comprehensive Composite Application SOA ................................................................. 12  
    Enabling Agile Business Processes .................................................................................... 13  

IV. **The ZapThink Take: Untangling Applications with SOA** ................................................... 14
I. Composite Applications and Service-Oriented Computing

In the fifty-plus years that companies have been using computers, the level of complexity has been steadily increasing. Today's information technology (IT) environment for many organizations is now numbingly complicated, as new systems and applications, company mergers and acquisitions, and an ever-changing business environment layer new technology upon the old. With such complexity comes inefficiency, risk, customer dissatisfaction, and wasted resources.

Fortunately, approaches to dealing with such complexity are evolving, as IT in general matures—the most effective being layers of abstraction. A layer of abstraction provides a simplifying interface that empowers users as it masks the underlying complexity of the technology. Compiled languages, the graphical user interface, and the World Wide Web are all examples of layers of abstraction that have put greater power into the hands of users, while hiding increasing levels of complexity under the covers.

Companies are now building a new layer of abstraction that masks the complexity of heterogeneous distributed computing environments known as Service Orientation. Service Orientation (SO) is an approach to distributed computing where software functionality is exposed as location independent, loosely coupled, and discoverable services on the network. At the core of SO is Service-Oriented Architecture (SOA), the broad set of best practices that enable companies to build a services abstraction layer that both hides the complexity of the underlying technology while at the same time providing an agile set of resources to the business.

SO's benefits to the enterprise are substantial—lowering integration costs, obtaining greater value out of existing assets, embedding their business processes within those of their customers and partners, and increasing business agility, leading to increased revenue and improved customer value. However, many challenges remain. Implementing an enterprise SOA is technically complex, and also requires substantial changes in the use and management of IT resources. Few software products on the market today address the full scope of what companies need to build, run and manage SOAs. However, the industry is increasingly converging on a set of capabilities to meet these needs.

SOAs in the Enterprise

There’s no question that today’s enterprise IT environment is complex, expensive to maintain, and short on flexibility. Coupled with ever-increasing business demands, perennially tight budgets, and a business environment of continuous, inherently unpredictable change, today’s IT manager requires a better approach to architecting IT functionality that will better address the ongoing needs of the
business.

Software architecture, however, is difficult to understand and even more challenging to put into practice. As a result, many organizations have “accidental architectures” made up of a gradual accretion of systems and applications, until the resulting organization looks like more of a rat’s nest than a planned architecture, as shown in Figure 1 below.

Figure 1: An actual accidental “rat’s nest” architecture

Clearly, organizations require a better handle on architecture. One succinct definition of architecture is the fundamental organization of a system embodied by its components, their relationships to each other and to the environment and the principles guiding its design and evolution. While the components referred to in this definition generally refer to hardware and software, the architecture also includes their relationships to their environment—namely, the users that work with the technology and the organizational constructs that form the business. Including people in the architecture helps the organization understand its business processes, and the processes in turn define the services that companies must construct when they abstract existing systems.

SOA is an evolutionary change to the practice of architecture. As companies seek to achieve the business agility promised by the practice of SOA, they face a substantial challenge, as the IT marketplace is currently in a transitional phase. Integration-based distributed computing products are gradually giving way to SOA-based approaches, but this shift is currently in progress. Furthermore, the movement toward SO computing affects many areas of IT beyond application integration. As a result, companies must be prepared to evaluate the products that are currently on the market carefully, and place them in the context of the broader shift to SO computing that the industry currently faces.

The key market reality that companies must understand is that at this point in time, the shift to Service Orientation affects all aspects of enterprise distributed computing. We are not moving toward a world where SOAs are one of several possible distributed computing architectures in the enterprise. Rather, other architectures, including n-tier, client/server, and message bus, will all come to be
SOAs provide a layer of abstraction above other architectural styles.

Business agility is the ability of a company to respond quickly and efficiently to change, and to leverage change for competitive advantage.

The ability to respond to changing requirements is the new “meta-requirement” for business.

Service Orientation embraces heterogeneity and obtaining greater value from existing legacy technology.

understood within the context of SOA. Fundamentally, SOAs provide a layer of abstraction above other architectural styles, providing an asynchronous, loosely coupled, coarse-grained service interface on top of a heterogeneous mix of architectures, systems, and applications.

**The Business Motivation for SOAs**

The difference between the practice of SOA and other approaches to architecture is in the business agility that SOA offers. Business agility is the ability of a company to respond quickly and efficiently to change, and to leverage change for competitive advantage. For the architect, building an architecture that provides business agility means creating an IT infrastructure that meets as-yet unknown business requirements—a situation that throws traditional IT planning and design out the window.

To meet the needs of today’s enterprise, the practice of SOA has the following core principles:

- The business drives the services, and the services drive the technology—In essence, services act as a layer of abstraction between the business and the technology.
- Business agility is the fundamental business requirement—The ability to respond to changing requirements is the new “meta-requirement” for business. The entire architecture—from the hardware on up—must reflect the business agility requirement, because any bottleneck in an SOA implementation can substantially reduce the flexibility of the entire IT environment, and hence the business as well.
- A successful SOA is always in flux—The everyday normal state of affairs is an IT environment that is undergoing constant change, and as a result, the work of the service-oriented architect is never done.

SOA requires a new way of thinking about architecture—placing the architecture in the human context of dynamic business and user requirements. Fortunately, SOA makes economic sense, and thus there is a solid motivation for architects to learn this new perspective.

**The Economics of Business Agility**

The current transition to SO is fundamentally different from the last major distributed computing transition: the one from client/server to n-tier architectures in 1996-97. That last build-out heralded the beginning of the dot-com boom, where Internet-related investment coupled with Y2K expenditures created a kind of IT “perfect storm” so dramatic it led to a worldwide economic boom, and subsequent downturn. Today, of course, the economic environment for technology adoption has completely changed, and this return to the “new business normal” is accelerating the move to SO. Rather than promoting massive build-out or extensive rip-and-replace, SO embraces heterogeneity and obtaining greater value from existing legacy technology. Today’s distributed computing transition, while every bit as significant as the ones that came before, has an entirely different economic model. Instead of massive IT investment, today’s IT executive is concerned with thrift.

Along with agility, thrift is a primary business motivation behind SOA. Hundreds of enterprises have already learned that taking an SO integration approach could reduce the cost of an integration project dramatically when compared to traditional integration techniques. Thrift, however, means more than simple cost savings. True thriftiness means making do with what you have—squeezing value
out of every asset. One of the clear benefits of an SOA is that such architectures help companies get more value out of existing resources by wrapping legacy applications in Web services interfaces and then making those services available on the network. Fundamentally, SOA promotes the reuse of existing IT assets, which leads to cost savings from reduced developer time and increased productivity among business users. In fact, most businesses have much of the functionality they require in the form of siloed applications in the enterprise. Such capabilities are not well coordinated across systems.

A second thrift benefit that SOAs provide is that they facilitate heterogeneous IT environments. Instead of “ripping and replacing” existing corporate IT systems by installing new systems and throwing the old ones out, SOAs enable users to build bridges between different systems and applications and leverage existing IT assets.

IT is often the area most relevant to discussions of thrift and business agility, because achieving agility begins with removing the bottlenecks that impede it, and IT has traditionally been the source of most bottlenecks. In fact, companies are so used to the fact that IT decision-making and implementations impede their organization that technology and its limitations often drive business decisions. SO, however, has the potential to change this equation, and enable business decisions to finally drive technology decisions.

The rearchitecture needed to move to SOAs does not take place in a vacuum; companies must transition their systems from the existing architectures to SOAs in a manner that does not impede the ongoing necessary functionality of the technology. Furthermore, the act of rearchitecting is not sufficient enough by itself to guarantee that the resulting business services will meet the needs of the business. Companies must also make a more fundamental change to a new SO mindset.

II. SOAs and Business Process

Today, we’re still in an early phase for Web services and SOAs, where people tend to apply yesterday’s approaches, theories, and concepts to new technology approaches. The traditional mindset that needs changing is the view that Web services are an extension of the component object model. To many developers, Web services are simply “another interface to a compiled object.” Instead, enterprises should approach SOA as fundamentally a process-driven architecture that leverages distributed processes in addition to distributed services. Distributed processes are all about the creation of business processes that in turn depend on other business processes that may be defined anywhere in the organization. Such distributed, SO processes are the key to composite applications that run on an SOA.

Approaching SOAs and Web services from this perspective simplifies and clarifies many of the troublesome issues relating to distributed computing and Web services. Integration goes from being a troublesome chore that must be accomplished through implementing increasing layers of complicated and expensive technologies to a side effect of process execution. In fact, it’s virtually impossible to create an important business process that does not provide the fundamental benefits of application and business integration. The mere act of orchestrating a composite application achieves most integration goals.
The Importance of Business Process

Organizations dedicate considerable sums of money and resources to developing and enhancing their business processes in a constant quest to deliver the best services and products for the least cost. In fact, enterprises can be thought of as an aggregation of their processes, the current state of execution of those processes, and the resources that comprise those processes. It is the role of business management to make sure that those processes are performed adequately or to change them to meet new business requirements. As a result, when companies look to effect change in their organization, they look first at the processes that comprise the organization and then to the tasks that support those processes.

Carrying out business processes efficiently also provides a competitive advantage. The efficient execution of a business process requires the use of some measurable criterion, such as cost, process execution time, or customer satisfaction. This requirement mandates that business not only learn how to establish business processes, but also learn how to measure and instrument processes so that they can be optimized. If a company determines, for example, that providing a self-service interface to a business process will be more efficient than assigning the same process to internal personnel, then the company can change the associated processes to meet the new business requirements.

Connecting Business Requirements to IT Capabilities with Business Process

High-level business requirements eventually need to be filtered down to actual activities connected to human-based tasks and IT-based systems. Yet, when talking with IT and business managers, it is clear that many organizations consider today’s IT infrastructure to be a bottleneck to operational efficiency. It is not that these systems do not work, but rather, that they were designed without change and flexibility in mind. Furthermore, many organizations record details of processes on paper or electronically as nothing more than reference documents. However, few of these documents are maintained to reflect day-to-day reality. As a consequence this valuable knowledge is often outdated and unmanaged, and key process details are often lost as soon as the expert leaves the building. In essence, process definition has been disconnected from process implementation.

The challenge with accomplishing many process-driven tasks is that enterprise computing architectures have not been architected from a perspective of business agility. This concept sounds like much the same goal as what is promised by continuous process improvement. The difference is subtle: where business agility is the goal of the enterprise, continuous process improvement provides a framework to meet that goal, and the underlying activities and systems power that framework.

After a decade or more when business process definition and execution was promoted by a series of theories such as business process re-engineering, quality management, six sigma and supply chain management, business process is now entering a new era with the onset of IT architectures that support business process goals. As a result, past models of monolithic application design are no longer an effective option. In fact, the cost and complexity of accommodating change in existing IT architectures is the major impediment to increasing productivity today.

Today’s enterprises have IT architectures and processes, whether or not they have been formally defined or established without planning. As a result, business requirements, process definition, and IT implementation are all jumbled together—applications are tightly-coupled with process logic, and business
process definition is disconnected from IT implementations. Before companies can attempt to architect systems that are capable of handling business requirements, they must separate these components into four distinct layers:

- High-level business requirements that result from senior management imperatives and overarching corporate goals.
- Processes that result from translating requirements into a practical set of activities.
- Lines of control that connect individual tasks within a process.
- IT capabilities that are represented by discrete, individual tasks, or services, performed by users or devices.

In order to connect those dots, companies must look at developing composite applications with a business process-centric approach.

**Loosely Coupling Business Process from Application Logic**

Application development itself has been changing over the past few decades. At first, applications were just programming language-encoded functionality connected tightly with the data that the application required. However, data were locked in the application itself, making programs difficult to update and reuse. Soon, our desire to abstract functionality was extended to client connectivity, first in the form of the client/server model, and then when the n-tier model abstracted the presentation layer form the business logic layer. But now we’re at the point where business processes are locked away in stovepipe applications.

Each added layer of abstraction has helped to facilitate business agility by isolating higher levels from implementation or process details. In essence, each layer of abstraction serves to loosely couple the consumers of the functionality from the technology underlying the layer of abstraction, facilitating agility by simplifying and reducing the cost of change.

The movement to various levels of abstraction touches upon an industry holy grail: application functionality reuse. A primary focus of IT architectures is the ability to reuse functionality. Applications are more maintainable because the business process is not hard-coded, but instead described by separate process definition layers that can be changed at runtime without requiring change to the underlying services. This abstraction enables flexibility and responsiveness to change at the most frequently changing architecture layer, that which implements business processes and procedures.

Moving process into a separate process layer above the SO layer of abstraction considerably facilitates business agility for the following reasons:

- A change in the process definition does not require a modification of underlying application functionality. Processes can be rapidly changed, amended and redeployed as business requirements demand with minimal, if any, programming.
- Businesses can easily monitor, audit, and escalate critical business processes since they have greater visibility into the overall process, rather than having the processes hidden by discrete application functionality.
- Processes can be composed of sub-processes, thus delegating business rules authority to different parts of the organization or to external organizations.
Processes can be implemented using any service that fulfills the basic requirements of that process flow, allowing for variable implementations depending on the requirements of a user.

However, if we want to create composite applications that successfully extract the "process-oriented" logic from the underlying applications, we must first agree on how to abstractly represent process flows, and then how to automate and execute those flows. In addition, there are significant layers of complexity involved when we aren’t talking about individual, atomic pieces of functionality, but rather the interactions of entire systems of resources. Furthermore, enterprises need technologies, products, and solutions that have been developed with process in mind, rather than just application functionality.

**Building and Running Composite Applications**

The first step to implementing business process in an enterprise is to create a representation of a business process that can then be executed by human and/or machine-based systems. Business managers and analysts tend to think visually. When they describe a process, they often reach for a pad of paper or marker to sketch out their ideas. To higher-level managers and line-of-business users, a process is something that can be written down and visualized in a way that can be communicated to others in the organization, as shown in the Figure 2 below. We can thank flowcharting and graphic design packages for the idea that these business users can represent theoretical concepts such as processes using abstract notation.

**Figure 2: Business Process Visualization**

![Business Process Visualization Diagram]

However, historically there has been little way to connect these meaningful diagrams into true composite applications. With the advent of SO approaches to process, users can finally put into practice their abstract business process representations. Since business-oriented users will rarely be successfully converted into programmers, these users are expecting the very same visual representations to be the primary way in which they can build composite applications that will implement their process flow desires.

The second step to building composite applications is to identify the services that are available to the enterprise. There are many challenges facing companies as they seek to define services, including:

- How granular should the services be to maximize the reuse of those services?
Which software functionality should be exposed as services? Where is the best place to start?

How should a company build the management, security, and scalability infrastructure to support the services, and to keep them loosely coupled?

How should the organization handle the information that the services deal with? What vocabularies and schemas should the organization use?

How should a company bring together diverse functionality in the enterprise into services, when the applications that provide that functionality might be Web-based, client/server, or enterprise business application-based?

And finally, once the organization has a visual representation of their business processes and clear definitions of the services available, what remains is to tie the business requirements to the functionality of the architecture in an agile way. The technology work is done at this point—all that remains is the human side: issues of governance, change management, and responsiveness to customer needs.

In fact, governance issues can be among the most challenging facing companies as they move to SOAs. Business units often do not own or control the systems that automate the functions that comprise their business processes, either because line-of-business and IT have separate spheres of control, or because processes cross multiple business units—or even multiple companies. Therefore, it is often impossible to create composite applications by restructuring the component applications.

Other problems can also prevent the invasive restructuring of applications. There are often additional licensing requirements for access to source code or data structures. Furthermore, such restructuring projects lead to increased training and support costs, and can be very expensive to upgrade. For all these reasons, a non-invasive approach to composite application creation that leverages the power and flexibility of SOA without changing the underlying applications provides companies with the greatest combination of flexibility, capability, and thrift.

III. Jacada: Non-Invasive Composition of Existing Applications

Of the five key challenges facing companies as they create and define the services that they will incorporate into composite applications, the ability to bring together diverse functionality in an agile, loosely coupled manner is one of the most significant. Jacada focuses on addressing this challenge. Jacada offers the ability to compose new applications from existing applications with its Fusion product. Jacada Fusion integrates business applications, delivering a simplified desktop to the user. It is able to pull together a variety of application types, including Windows applications, applications with Web interfaces, and host-based applications.

The key to Jacada's solutions is their non-invasive nature, which means that customers do not need to rewrite or replace existing applications to take advantage of business processes in a SO environment. In fact, companies don’t even need to have the source code for those existing applications.
Jacada Fusion helps companies build an SOA by providing a loosely coupled abstraction layer on top of existing applications. Jacada Fusion provides service interfaces to existing applications and enables users to compose those applications into agile business processes, as shown in Figure 3 below.

**Figure 3: Jacada Fusion Provides an SO Abstraction Layer**

The benefits of Jacada’s approach include the ability to:

- Service-enable business processes that are embedded in existing business applications, regardless of the type of application.
- Integrate legacy systems with other enterprise applications to allow users to change the business processes as needed.
- Eliminate redundant data entry, thus improving data quality.
- Provide Web and rich client interfaces to host applications.

Fundamentally, Jacada Fusion provides the ability to map processes to services regardless of their source system, which increases both the speed of deployment and the productivity of the users of the composite application. Jacada Fusion also prevents companies from feeling locked into a particular composite application. If requirements change, the composite application has the flexibility to change as well. And finally, composite applications built with Jacada Fusion enable the ability to map processes to services regardless of their source business application.

**Comprehensive Composite Application SOA**

Jacada Fusion enables the delivery of composite applications, leveraging any Windows, Web, or host-based, custom or packaged application. Jacada Fusion
leverages SOA as an underlying principle to deliver process flexibility using methods that deliver results quickly. Jacada Fusion neither relies upon existing application APIs nor accesses the graphical display layer, offering a more agile approach to service-enabling existing applications than more traditional API-based or screen-scraping approaches. Direct API access requires an often unavailable knowledge of the functionality of an application, while screen scraping techniques are particularly brittle, breaking whenever any small variation in the display of an application occurs. The Jacada approach is thus the most loosely coupled of the three approaches, making it more suitable for building composite applications in an SOA.

Jacada Fusion also offers the flexibility for users to either designate one of the existing applications as the controlling one, or if none of the applications supports the ability to drive the resulting business process, users can then use Jacada Fusion to create a new composite application with an entirely new interface.

Jacada Fusion works well in call centers, where customer service representatives (CSRs) must frequently deal with multiple application interfaces in an environment where promptness and quality of service is vital to the success of the business. Examples include “in-call” tasks, such as the process of placing an order or looking up information. Jacada can also improve “Post-call” wrap-up activities by reducing or eliminating the administrative data entry that most CSRs execute to synchronize systems and meet compliance requirements.

Additionally, many administrative back-office processes are also a good application for Jacada Fusion. Any process that requires a user to interact with multiple, disparate, application interfaces can benefit from Jacada Fusion. Jacada Fusion is also equally applicable for building corporate portals that must access multiple applications. After all, the value of the portal is often to provide single sign-on capabilities and intuitive access to the application functionality needed to complete a business task. Instead of shoehorning application interfaces together at the presentation layer, the way that traditional portals operate, a company can use Jacada Fusion to create either a new or simplified composite application and then expose that application to the portal.

However, self-service applications like call center interfaces and corporate portals are only a few examples of how Jacada Fusion can enable agile composite applications. Jacada’s focus is the simplification of processes and work that people perform which today prevents them from delivering efficient high-quality interactions.

Enabling Agile Business Processes

The majority of desktop inefficiency is a result of the need for frequent switching between applications to complete an assigned task. Users must perform redundant data entry or information look-ups, or manually continue a business process, all of which are time-consuming and error prone.

With Jacada Fusion, organizations can change and improve their task-based business processes without rewriting or replacing their underlying applications. Jacada Fusion provides a simplified user interface that eliminates the need to navigate through and between applications. Jacada Fusion eliminates redundant data entry, cuts keystrokes and streamlines process steps.

The primary barrier to achieving such business improvements is often the stovepiped, inflexible nature of the applications that support the processes.
through acquisitions, have created a level of complexity that is strangling corporate speed and agility while stifling user productivity and innovation. Jacada Fusion helps to resolve these issues by simplifying how users access and manage data in multiple systems, accelerating their ability to respond to customers and changes in the business environment.

IV. The ZapThink Take: Untangling Applications with SOA

Business users work daily with business processes, but today's business applications are more likely to automate functions rather than processes. As a result, users must execute processes manually from a tangled collection of application interfaces. Such users must frequently copy data from one application into another, use different application interfaces for tasks within the same process, and remember process steps in their head—problems that often fall under the label “swivel chair integration,” because of the need to move from one application to another to get the job done. Each of these limitations leads to lost productivity, errors, and poor customer service.

Jacada Fusion cuts through this complexity by providing a non-invasive abstraction layer on top of a broad range of heterogeneous application interfaces. Companies can thus use Jacada Fusion to solve many of the problems with swivel-chair integration, leading to a clear tactical advantage in environments with heterogeneous interfaces to different applications. Jacada Fusion also plays a strategic role within organizations looking to take advantage of the agility benefits of SOAs, because it enables companies to craft composite applications from services. Such composite applications empower users to craft and execute business processes in an agile, flexible manner, spelling the end to swivel-chair integration.
Copyright, Trademark Notice, and Statement of Opinion

All Contents Copyright © 2004 ZapThink, LLC. All rights reserved. The information contained herein has been obtained from sources believed to be reliable. ZapThink disclaims all warranties as to the accuracy, completeness or adequacy of such information. ZapThink shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.

About ZapThink, LLC

ZapThink is an IT market intelligence firm that provides trusted advice and critical insight into XML, Web services, and Service Orientation. We provide our target audience of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing—a vision of IT meeting the needs of the agile business.

ZapThink’s role is to help companies understand these IT products and services in the context of SOAs and the vision of Service Orientation. ZapThink provides market intelligence to IT vendors who offer XML and Web services-based products to help them understand their competitive landscape and how to communicate their value proposition to their customers within the context of Service Orientation, and lay out their product roadmaps for the coming wave of Service Orientation. ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into how to assemble the available products and services into a coherent roadmap to Service Orientation. Finally, ZapThink provides demand intelligence to IT vendors and service providers who must understand the needs of IT users as they follow the roadmap to Service Orientation.

ZapThink’s senior analysts are widely regarded as the “go to analysts” for XML, Web services, and SOAs by vendors, end-users, and the press. They are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry.

ZapThink was founded in October 2000 and is headquartered in Waltham, Massachusetts. Its customers include Global 1000 firms, public sector organizations around the world, and many emerging businesses. ZapThink Analysts have years of experience in IT as well as research and analysis. Its analysts have previously been with such firms as IDC and ChannelWave, and have sat on the working group committees for standards bodies such as RosettaNet, UDDI, CPExchange, ebXML, EIDX, and CompTIA.

Call, email, or visit the ZapThink Web site to learn more about how ZapThink can help you to better understand how XML and Web services impact your business or organization.

ZAPTHINK CONTACT:
ZapThink, LLC
11 Willow Street, Suite 200
Waltham, MA 02453
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com