systinet™

A Practical Guide to SOA
for IT Management

February 2005

A Practical Guide to SOA for IT Management
A Systinet White Paper

February 2005

Systinet Corp. One Van de Graaff, 5th Floor Burlington, MA 01803 Phone: 1.781.362.1300
www.systinet.com

systinet™

A Practical Guide to SOA
for IT Management

Contents

Contents

Executive Summary

As a strategy for creating a flexible and agile IT infrastructure, SOA (service oriented architecture) has gained considerable momentum in recent years, largely due to the advent of standards-based Web services that make SOA a practical reality. There are some powerful business drivers for implementing SOA today:

- **IT Agility**
  SOAs make IT more responsive to changing business demands, and more flexible to changing business processes. Reconfiguring loosely coupled business services is simple, fast and low-cost.

- **Maximizing IT Investments**
  SOAs do not rip and replace your current IT application and data assets—they wrap and reuse existing IT investments and make them available to a wider audience. SOA encourages reuse and avoids unnecessary duplication and reinvention.

- **Aligning IT to Business Processes**
  SOAs transform IT systems into self-contained services that accurately reflect business processes and operational requirements. IT mirrors business operations, improving performance.

- **Creating a Standards-Based IT**
  SOAs that use standards-based components provide unparalleled interoperability for all IT systems and services, making IT more flexible and dramatically simplifying integration.

This paper discusses how to capitalize on the advantages of SOA, presenting a management framework for implementing an SOA and the technical, organizational and process issues involved. Finally, it offers recommendations on building an infrastructure so that IT management can adequately address the SOA imperative.

The Business Value of SOA

Service oriented architecture (SOA) is a design style for maximizing IT interoperability, sharing and reuse in a distributed environment. Service orientation isn't a new approach to software design, but it has become increasingly important because of the widespread adoption of Web services technology that makes creating an SOA easy and inexpensive. [Web services technologies, protocols and products are discussed in Appendix A.]

SOAs offer distinct advantages over other architectures. First, they make interoperability an innate characteristic of IT applications. Applications built using SOA and Web services become shared resources that are completely platform-independent, language-independent, and very loosely coupled based on universally accepted industry standards. Organizations no longer need to invest inordinate amounts of time and resources writing custom adapters to integrate applications, only to have to recode them when changes are made to support new business processes. With an SOA, all IT systems have interoperable applications, and so the problem of integrating them becomes moot.

Second, SOAs make IT more agile and more responsive to changing business demands. New business processes can be supported and integrated across organizational and IT systems on demand, and organizations can easily compose reusable, shared services to respond to new business challenges. In addition, since services represent high-level business logic, IT is encouraged to think in terms of business functions. With SOA, IT systems quickly and accurately adapt to organizational goals and processes. SOAs make IT highly tolerant of changes, and reconfiguring loosely coupled services is a simple and economical process.

Overall, SOAs offer an easy way to speed time-to-market, respond quickly to changing business conditions, eliminate rework and maximize the value of existing assets. The result is a very positive effect on both workflow and the bottom line.

Management Guidelines

As your organization begins the SOA planning, development and deployment process, there are a number of things to bear in mind. IT managers should consider the following guidelines as they approach an SOA rollout:

Plan for Incremental Deployment

A complete conversion to service oriented architectures, principles and practices does not have to happen overnight. SOAs can easily be deployed incrementally and still show business value.
For example, SOA can show immediate value on projects with multi-point integration involving heterogeneous or legacy applications. Reuse of legacy code and integrating diverse platforms is an ongoing challenge for most enterprises, but that challenge is easily met using SOA.

Focus on Interoperability

Innate interoperability is a key benefit of SOA. For non-mission-critical Web services, this requires standardizing the message (operations and data types) and the services descriptions (WSDL), along with implementing the services registry as soon as possible in the SOA rollout. For mission-critical Web services, it is also necessary to focus on additional interoperability issues such as security, reliability, transactional integrity, monitoring and management, along with load balancing and service provisioning. A robust program of identifying, evaluating and prioritizing services will help determine which ones should be enabled to accomplish specific business goals, and whether they are mission-critical or not. Today, existing SOA infrastructure components can support all these requirements.

Focus on Business Agility

Business and IT agility should always be a primary and overarching goal of your SOA strategy. IT systems begin to mirror business processes, making it easier to map business change to system change.

SOA technology makes it easier to implement IT change because systems are composed of *loosely coupled business services*. This means that changes in services do not interfere with connections between services, and reconfiguring processes is straightforward. For example, if your business rules for processing payroll are changed, the retirement plan system that depends on the payroll service is unaffected.

The Stages in SOA Adoption

Systinet defines three stages of evolution to a service oriented architecture.

Phase One: Web Services Enablement
This is a tactical implementation where existing applications have standard Web services interfaces, reducing integration complexity. These developer-driven implementations have modest returns on investment in the short term.

Phase Two: Business Services Enablement
Success with simple Web services encourages a systematic, enterprise-level approach in the next phase. This architect-driven phase transforms Web services into business services with a focus on reuse. The primary benefit of reuse is increased alignment to business, and consequentially, improved agility to facilitate business change. Impact analysis and reporting are key outcomes.

Phase Three: Dynamic Interoperability
The desire to support composite applications and agile or "on-demand" computing drives this phase, which is characterized by a concept Systinet calls Dynamic Business Interoperability. By managing business services using high-level definitions called metadata, change to IT or business processes can be dynamically managed, further increasing agility and dramatically lowering operating costs.

The business services that compose an SOA represent a *coarse-grained view* of IT assets—i.e., it defines services around business concepts rather than technical details. This allows business analysts to easily understand and work with business services to implement change without turning to IT.

Recognize the Registry's Critical Role in SOA

The basic SOA model is defined by interoperable service providers and service consumers matched by a Business Service Registry. Service providers and consumers can be individuals, departments, organizations or applications that provide or use services in the SOA according to pre-defined business policies. In this model, the registry is an essential component for the important SOA lifecycle phases of enablement, publishing, discovery and management.

Beyond these baseline considerations, other issues will become important as the SOA is incrementally deployed across the enterprise. These include the following:

Define and Enforce Application Interoperability Policies

An organization must define an interoperability architecture and policy to manage all integration efforts. These must specify how Web services will be used and what standards must be defined and enforced. Organizations should also define a reference architecture for migrating point-to-point Web services to reusable business services.

Change IT Procurement Policies

Over time, the ongoing management of your SOA will make it necessary for vendors to comply with your SOA strategy and interoperability policy. All vendor solutions should be evaluated according to your SOA strategy and reference architecture. Packaged software should support Web services standards and be able to produce and consume (i.e., use) other Web services.

Transform Your IT Development Processes and Policies

SOA represents a way to drastically improve IT processes, especially in application development. Development organizations should create modular, component-based applications using Web services design standards. Compliance to WS-I and internal standards should be enforced at design and runtime using available Web services management (WSM) and SOA enforcement tools.

Define and Enforce Your Business Interoperability Policies

As your organization expands its SOA and services portfolio, you will increasingly interact with other SOAs from customers, business partners and other organizations within your enterprise. The resulting business integration challenge will be easier to manage when your SOA defines a policy for B2B interoperability and a strategy for leveraging industry XML standards.

Monitor, Measure and Analyze Your SOA Service Network

SOA metrics should be defined early in the SOA strategy and planning cycle. These metrics will help define the overall effectiveness of the SOA by answering the following kinds of questions:

- Are the services leveraging one another in a symbiotic, networked fashion?
- Are we getting the maximum interoperability and services reuse from our SOA? If not, why not? Are policies being enforced at design and runtime?
- Can services be easily discovered and invoked?
- Finally, have initial SOA business goals been met? How can we improve our performance? If our goals were not met, why not? What must be done?

Creating an SOA Foundation and Lifecycle

Properly implemented, SOA is an excellent approach for managing change—change of service owners, service implementations, provider-consumer relationships and even entire business processes. The key to managing change in an SOA is managing the lifecycle of SOA business services—what we call Business Service Lifecycle Management.

The business service lifecycle is a defined set of activities and functions that continuously support SOA business service development, deployment and management. This approach recognizes the need to achieve defined business outcomes via SOA, and acknowledges that there are new and unique requirements to help an organization successfully make the transition and ultimately manage dynamic business interoperability. The SOA business service lifecycle can be summarized by the following diagram:

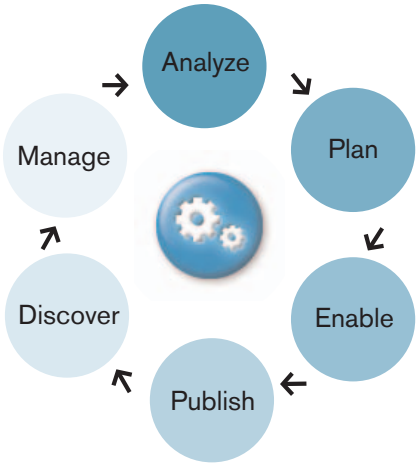

Figure 1—The SOA business service lifecycle.

Many will recognize these steps as the way business is done and IT is managed today. The difference is the importance of publishing and discovery as part of a rigorous, managed process. Most methodologies don't consider publishing and discovery as highlighted steps; however, for SOA they are mandatory, as described below.

SOA Planning

As in any management activity, planning is critical—in this case, planning the transition from Web services to reusable business services. Here are some considerations:

SOA Governance

SOA governance consists of the corporate, business and IT processes and rules required to guide the business success of an SOA and Web services. SOA governance defines and enforces the Web services policies needed to manage SOA applications and data for business success.

The SOA governance model will result in policies for services reuse, IT compliance and security.

Reuse: A reuse policy would describe technical and business aspects of services reuse, such as WSDL design conventions and WS-I compliance for interoperability, as well as a review process to ensure existing services are used before developing new ones.

Compliance: an SOA compliance policy would describe what internal and industry standards will be followed for all services, whether internal or from external providers.

Security: A security policy might specify what security standards and credentialing processes will be enforced during services design and consumption, such as SAML, WS-Security, XML Signature and others. It should also tell developers how to determine the risk factors associated with a given service, and what tools and techniques they should use to secure the service.

Business Services and SOA Governance

Business services are Web services operating in an SOA, augmented with the necessary compliance (e.g., HIPAA, Sarbanes-Oxley), security and manageability for governance and enterprise-wide use.

Effective governance involves three steps:

- Definition of policies
- Tools and infrastructure to support policy implementation
- Process and procedures that verify compliance and policies

Effective governance involves mapping corporate, business and IT policies to specific business services, and then ensuring policy enforcement. Governance defines, enables and enforces the rules for managing business services, and the service lifecycle infrastructure ensures proper implementation and operational support during service enablement, publishing, discovery and management. Policies can be business, security or standards-compliance policies (such as WS-I), or they can be internal standards and other technical policies.

Metadata and Classification Management

Using metadata to fully describe services is essential to getting the most visibility, productivity and reuse from your SOA. This requires creating common services, and a shared understanding of how to manage, change and use these services. Modeling your enterprise using metadata is similar to other types of data modeling in your organization, helping to align IT infrastructure with the business model and keeping it aligned as both the business and the IT infrastructure change.

New SOA Infrastructure

During the planning stage, steps must be taken to evaluate new technical components that establish a new foundation of SOA infrastructure. Key to this new infrastructure is the Business Service Registry, which provides a centralized location for managing all the descriptions of services and related SOA information. The Business Service Registry provides the central point of SOA governance to manage policies and drive enforcement at the services level. In essence, it becomes the "system of record" for the SOA.

SOA Enablement

There are two components to SOA enablement:

- Creating business/Web services
- Installing new SOA infrastructure technology, such as a Business Service Registry or Web services management platform

Creating Business Services

SOA enablement also refers to the process of enabling or developing new business services, or of exposing existing application functionality (e.g., packaged or MOM-based applications) as business services. There are numerous Web service enablement platforms available today. In selecting a solution, it is essential to ensure that important standards such as WS-I (Web Services-Interoperability) and WS-S (Web Services-Security) are fully supported. [For a list of recommended standards, see the SOA Lifecycle Planning Matrix in this document.]

SOA Enablement Infrastructure

At its most basic, the enabling infrastructure can be simplified as follows:

- Service provider enablement
- Service consumer enablement
- Standards-based Business Service Registry
- Supporting infrastructure, such as Web services management, identity management and service-oriented messaging services

Service Provider Enablement

For a service provider, enablement involves publishing business service definitions, descriptions, information and security/access policies to the Business Service Registry.

Service provider enablement also includes aspects of management. These include instrumenting the code, enabling version control and providing for dependency analysis.

Service Consumer Enablement

Since reuse and composite applications are key business drivers for SOA, a service consumer must be able to discover service providers—at both runtime and design-time—and the technical, policy, ownership and other information about them.

Service consumers may also need to negotiate with service providers at either the business or technical level, according to predefined, acceptable policies. Service consumers must be able to accommodate changes to the service providers, on demand and reliably. It is inevitable that the service provider will need to be updated and maintained independently of the service consumer.

Business Service Registry

The Business Service Registry is a core foundation in the new layer of SOA infrastructure, providing the single thread of visibility and control of all SOA interoperability activities and related information across the SOA lifecycle.

Web Services Management

Web services management provides essential monitoring, metrics and control of Web services, and works with the Business Service Registry to provide lifecycle management of services.

SOA Publishing

Publishing services in an SOA is a policy-driven process that needs to answer questions such as:

- Who is allowed to publish a service to the registry?
- What release procedures must be followed?
- How will various designs, standards and security policies be approved, certified and enforced in the SOA?

These are all issues that are typically supported by a Business Service Registry. The registry provides the means to publish a service and can also host validation routines that verify compliance with policies. Publishing is critical in the SOA lifecycle because Web services are not truly "business" services until they can be discovered and shared by all SOA participants.

The registry exists at the core of the SOA and consequently must support diverse user communities. Business classifications provide context for various audiences—developers, business analysts, IT architects and line-of-business management. For example, a developer might refer to a service by a very technical name that only has meaning for the programming community, but that same service might be consumed by an external business user or customer. A reference using a technical name won't have meaning for a business, so business must provide meaning and context to non-technical SOA users and services consumers. The Business Service Registry's ability to provide meaning and context to services in an SOA is critical to increasing usage, supporting service discovery, and maintaining control of services in the SOA.

SOA Discovery

An SOA is predicated on the concept of discovery. Until business services can be found, shared and discovered on demand, there is no SOA.

Standards-based registries use the Universal Description, Discovery and Integration (UDDI) specification to support the finding, sharing and discovery of services. UDDI is a mature and widely adopted standard supported by many leading companies including IBM, Microsoft, SAP and others.

Registry-based discovery allows organizations to leverage existing services and support the development of new ones, as well as support business processes in production. It facilitates service reuse so that developers and service consumers may use existing IT assets and services while providing a foundation to accelerate time-to-market for new business and IT functionality.

A registry also allows dynamic deployment of services, enabled by WSDL service descriptions and additional service policies described in XML-based schemas and documents. With dynamic, on-demand discovery, implementing new business processes and functions is significantly faster and, more importantly, controlled.

The registry also supports other infrastructure services, including management, event and workflow services. These services support the coordination and orchestration of providers and consumers, as well as ensuring control across the full service lifecycle for ongoing maintenance, version management, release management and more.

SOA Management

Web services management (WSM) solutions provide visibility and control for an SOA. This includes:

- Monitoring (e.g., metric computation, service-level objective evaluation)
- Automation (e.g., deploy, un-deploy and upgrade)
- Auditing and utilities (e.g., alert notification and logging)

WSM must also manage the provider-consumer relationship, or what many call the "contract." Since integration is built-in, SOA often mitigates the need for traditional middleware. The key to this relationship is managing separation. Clear boundaries reduce the impact of change and formalize the interfaces between the services. The service consumer must be managed such that only the service itself matters; a service should not need to know the details of the underlying implementation. When this is achieved, significant benefits accrue to service providers who do not need to make any assumptions about consumer behavior. The service provider has a different set of concerns: the service must be consumable by any SOA consumer, requiring a focus on the interface, the description, the contract and the way that services are found. The consumer-provider relationship can be either negotiated or mandated, or a combination of the two. An example of a negotiated relationship might be a service interface that is agreed upon as part of a commercial contract. Mandated relationships are those typically defined by a dominant market or organizational player, and can be either consumer- or provider-led. A market example might be a large purchaser (such as a major discount store chain) mandating a particular interface for its suppliers. By furnishing information about service providers and service consumers, and the relationships between them, the Business Service Registry is a key enabler for managing this relationship.

SOA Analysis

The SOA lifecycle also includes the analysis, monitoring and feedback mechanisms to help optimize the SOA, and ultimately the organization's business processes it enables. Analysis activities include monitoring service status, users, usages and metrics to indicate the relative success of various business services in the SOA. This may also provide impact and dependency analysis of individual services, groupings of services based on classification, and reporting on a wide variety of issues such as reuse, policy violations or compliance reporting.

SOA Lifecycle Planning Matrix

The table below summarizes the SOA lifecycle, requirements, enabling infrastructure and relevant standards, as well as potential vendors of various solutions.

SOA Lifecycle	SOA Requirements / Activities	SOA Infrastructure	SOA Standards	Vendors
Planning	SOA governance & management policy enforcement SOA metrics Quality of service/ reliability/latency policies Security policies Classification design	Registry Security Management Governance	Corporate standards & policies WSDL WSDM WS-Policy	Microsoft System Integrators Systinet WebLayers
Enablement	Business modeling Corporate, business, IT governance SOA foundation development Web services development Infrastructure development	SOAP–WS Server and runtimes Application servers Management and/or SOA network Security proxies WSDL, classification, XML and other modeling tools ESB, SOA networks, EAI/message brokers	BPEL4WS WS-Addressing WS-I Basic Profile WS-Notification WS-Eventing WS-RM WS-Security XML, SOAP, WSDL	AmberPoint BEA IBM Microsoft Oracle Salesforce SAP Security vendors Systinet TIBCO
Publishing	Business service approval Certification process Change management Registration process & management Categorize services and create classifications from service interface Enrich service interfaces with policy- related metadata	Registry, classification creation, content management Policy design Process design	UDDI	IBM Microsoft Systinet
Discovery	Find and invoke business services Walk classification trees Design time usage Runtime usage Configuration and change Operational management Introspect metadata	Registry	UDDI	IBM Microsoft Systinet
Management And Security	Operate and manage business services Create, monitor and enforce SLAs and other policy Enforce security and identity Control service provider access Track and manage provider- consumer relationships Create parameters to monitor and provision monitoring tools Create and change classifications Create and change service providers	Registry Identity server, hardware/ software firewalls Management proxies and instrumentation tools Alerting systems Discovery tools	UDDI WS-DM WS-Security	Actional AmberPoint CA HP IBM Microsoft Systinet
Analysis	Analyze performance SOA metrics management SOA performance analysis	Registry Management console Data mining/ analysis Visibility solutions	UDDI WS-Policy assertion	Actional AmberPoint Service Integrity

Business Service Registries Enable SOA Lifecycle Management

The SOA lifecycle requires visibility, control and management across all stages. This level of visibility and control can only be provided by a registry-based approach to SOA.
A Business Service Registry is a core foundation for SOA infrastructure that paves the way to faster ROI using Web services. The registry not only speeds the transition from Web services to reusable business services; it is the hub that connects all SOA participants—providers, consumers, developers, customers, partners, business executives, IT executives and more. The registry provides the pathway to a business-driven SOA by supporting the following benefits:

- Speed and control of SOA deployment
- Speed and control of business services rollout and deployment
- Speed of internal business and IT SOA usage
- Speed of partner SOA usage
- "Increasing returns" through registry-driven SOA
- Accelerated benefits and ROI

The Business Service Registry supports the entire SOA lifecycle and enables the transition from initial Web services to business services, to Dynamic Business Interoperability.

The SOA Imperative

The evolution to an SOA requires new thinking about application design, IT assets reuse, the lifecycle of business services, and the IT roadmap for a new foundation of SOA infrastructure. SOA provides an approach for the design of a new generation of modular, standardized software services. The SOA lifecycle provides a logical and formal way to understand the business and technology issues required to evolve your business model into an SOA model. A business-driven SOA strategy will help focus on the goal of Dynamic Business Interoperability.

The SOA lifecycle helps clarify the SOA pathway; the Business Service Registry provides the visibility, management and control of the SOA information and function. Both lead to Dynamic Business Interoperability, dramatic business results and an agile IT.

Appendix A: Web Services Enablement Products

Examples of Web Services Enablement Products

Web services enablement products are available for most computing platforms and environments today, whether the goals are exposing mainframe CICS as services or enabling SAP using NetWeaver or third-party tools. The speed and ease of enabling Web services are among the greatest benefits of this approach. Below are some examples of Web services enablement tools for various computing environments.

Enablement of Packaged Applications:

- SAP NetWeaver allows SAP users to expose SAP functionality as services as part of an end-user's SOA strategy.
- Salesforce.com Sforce allows end-users to customize, integrate and extend Salesforce.com's CRM solution to meet their business needs.
- Third-party tools and adapters are available for popular packaged applications to help create a vendor-neutral SOA model.

Enablement of Legacy Applications:

- This is possible using many third-party Web services enablement solutions:
  - Mainframe CICS, IMS
  - Proprietary applications

Enablement of Strategic Business Applications:

- Amazon.com Merchant Network allows Amazon to extend its sales processes to affiliates and others using its Web services enablement framework. This is an example of "syndicating a process" to trading partners.

Enablement with Application Platforms:

- Microsoft .Net provides many tools and solutions for rapidly enabling Web services within the .NET environment.
- The J2EE platform is supported by a host of tools for Web services enablement.

Cross-Platform Enablement:

- Cross-platform enablement fills a critical need for end-users with heterogeneous infrastructures. In these cases, SOA truly becomes the unifying architecture for an enterprise.
- Systinet has feature-rich, standards-based solutions for Web services enablement for such cross-platform SOA requirements.

Appendix B: SOA Defined

Service orientation is an approach to designing software systems. A service oriented architecture (SOA) is a system consisting of modular software components with standardized component-access and usage interfaces that are independent of any specific platform or implementation technology. More importantly, an SOA enables software components to become standard services that can be invoked at runtime or on demand, rather than repeatedly designed and programmed by traditional means. In SOA, a "service" is typically a group of software components that together carry out a high-level function or business process, such as placing an order or making a credit approval on a purchase. At its most basic, an SOA is simply a collection of standardized services on a network that communicate with one another in the context of a business process. This approach dramatically eases integration in heterogeneous environments and provides a major enhancement in agility.

All services share some common characteristics:

- Services have interfaces that are platform or implementation-technology independent. Services are exposed using standards-based, identical interfaces that make them easy to use and reuse, and guarantee dynamic interoperability.
- Services are "loosely coupled." Services can be created without any forethought as to how or who will consume them. In addition, changes made to the service implementation will have no ripple effect on the consumers.
- **Services are "coarse grained."** Services focus on high-level business processes using standard interfaces, and thus mask the underlying technical and operational complexities of how a service is implemented.
- **Services are modular.** A service represents a discrete unit of business, application or system functionality. Multiple services can be combined to deliver more valuable services. This modular approach gives organizations great flexibility in system design. By reassembling services into a new configuration, a business can create a new business service to support a different business objective.

Appendix C: Glossary

BPEL—Business Process Execution Language: an XML-based language designed to enable task-sharing for a distributed computing or grid computing environment, even across multiple organizations, using a combination of Web services.

Business Services—Web services operating in an SOA with the necessary governance, policies and business taxonomies that enable business customers, IT applications and data, business partners, and internal enterprise users to access them. Business services provide the visibility, reusability, adaptability, and manageability for true business interoperability using an SOA. Business services can be service producers, service consumers or (most typically) both.

CICS—Customer Information Control System: an online transaction-processing (OLTP) program from IBM that, together with the COBOL programming language, has formed the most common set of tools for building customer transaction applications in the world of large enterprise mainframe computing over the past several decades.

CRM—Customer Relationship Management: all aspects of interaction between a company and its customer, whether sales or service related.

IMS—Information Management Software: IBM's premier transactional and hierarchical database management system for critical online operational and e-business applications and data.

MOM—Message-Oriented Middleware: a specific class of middleware that supports the exchange of general-purpose messages in a distributed application environment.

OASIS—Organization for the Advancement of Structured Information Standards: a not-for-profit, international consortium that drives the development, convergence and adoption of e-business standards.

SAML—Security Assertion Markup Language: an XML-based framework for exchanging security information.

Service Consumers—business services that consume service providers. Typically, they discover, retrieve and introspect service information that they obtain via a WSDL description obtained from a known URL or a Business Service Registry. They are quite different from service providers in many respects, including security and system usage requirements. Typical examples include Web services, Web users and applications, PC users and applications, and special devices (e.g., cell phones).

Service Instance—basically a concrete realization of a business service. An instance is sometimes also called an "endpoint," which denotes a runtime instantiation of a logical Web service, accessible via a particular technical protocol and transport.

Service Owner—a system entity that provides a collection of services. A service provider usually represents a business model and related process(es), generally claiming ownership and management responsibilities over its services. A service owner may host one or more Web services, which may be hosted on one or many physical machines.

Service Providers—business services that publish business service definitions, descriptions, information, and access control and authentication rules. Typically, service providers will be categorized along a range of business, functional and technical taxonomies based on a business model. Examples include data marts/warehouses, business processes and commercial off-the-shelf (COTS) applications.

SOA—Service Oriented Architecture: a simple software design approach and system in which all software functions are modeled as modular components and are implemented as platform and implementation technology independent services that can be consumed over a network using standards-based interfaces.

SOA Governance—the organization and processes required to guide the business success of an SOA and Web services. SOA governance defines and enforces the Web services policies needed to manage SOA applications and data for business success.

SOA Lifecycle Management—an approach to continuously managing the development, deployment and management of SOA business services through a set of defined activities and processes including planning, enablement, publishing, discovery, management and analysis of services.

UDDI—Universal Description, Discovery and Integration: an OASIS standard for Web services publishing and discovery using a service registry. Web Services—a set of standard interoperability specifications for loosely coupled, self-describing software functions accessed programmatically across a network.

WSDL—Web Service Description Language: an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. WS-I—Web Services Interoperability: an open, industry organization chartered to promote Web services interoperability across platforms, operating systems and programming languages.

XML—Extensible Markup Language: a simple, very flexible text format derived from SGML (ISO 8879).

About Systinet Corporation

Systinet provides the leading foundation for SOA governance and lifecycle management. Founded in 2000, Systinet's award-winning, proven, and standards-based products enable IT organizations to rapidly leverage existing technology investments, provide interoperability between heterogeneous systems, and better align business processes with IT. Customers receive the benefits of a simpler, faster, standards-based way to dramatically improve IT responsiveness and technology asset reuse, while maximizing the ROI for SOA. Systinet's customer base of over 150 Global 2000 clients includes Amazon.com, BMC Software, Interwoven, JPMorgan, Motorola, Defense Information Systems Agency, and SAIC. Headquartered in Burlington, Massachusetts, Systinet is a privately held company with over 100 employees.

To find out how Systinet can help your business, visit **http://www.systinet.com**, call 1.781.362.1300, or email us at sales@systinet.com.