

zapthink
white paper

THE SOA GOVERNANCE TIMELINE



THE SOA GOVERNANCE TIMELINE

June 2007

Analyst: Jason Bloomberg

Abstract

Governance consists of creating, communicating, and enforcing policies in a corporate environment. In many ways, it is the key to maintaining the balance between executive control and employee and customer empowerment.

Implementing SOA requires governance in order to ensure that the organization applies and enforces the policies that apply to the Services that the organization creates as part of its SOA initiative. But more importantly, organizations can leverage SOA best practices to represent policies broadly in such a way that the organization can achieve better policy management, flexibility, and visibility into policy compliance across the enterprise. Because of these two characteristics, enforcing policies and leveraging SOA best practices, it is critical for all organizations to deploy SOA governance as soon as they begin their SOA initiative.

With its acquisition of Mercury and its Systinet division, HP has propelled itself into a leadership position in the SOA governance space. HP is well-positioned to help its customers leverage SOA for IT governance, and more broadly, for corporate governance.

All Contents Copyright © 2007 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.

I. Executive Summary

Running any organization, whether public or private sector, large or small, requires a careful balance. Too little management control can lead to chaos, while too much often results in low morale and poor productivity. Maintaining this balance is especially critical in today's complex, dynamic business world, as information technology (IT) increases the speed and flexibility of progressive organizations. Rarely does a day go by without news about exposed confidential information or accounting irregularities hitting the headlines. Today's globally connected, always-on business environment presents heretofore unknown challenges to any executive looking to govern their companies effectively.

In fact, it is *governance* that is the key to maintaining the balance between executive control and employee and customer empowerment. Governance does not merely consist of setting corporate policy; it also encompasses communicating those policies, giving people the tools they need to follow the policies, as well as the enforcement of policies, mitigation of policy breaches, and visibility into levels of policy compliance.

IT organizations must improve their responsiveness so they can better meet the continually changing needs of their organizations. This need for increased responsiveness in the face of business change is the core motivation for *Service-Oriented Architecture* (SOA). Properly implemented, SOA can provide the agility that organizations require to remain competitive in today's cutthroat business environment. And yet, with flexibility comes risk, because increasing agility leads to greater user empowerment, and with greater user empowerment comes the possibility of policy breaches. Therefore, SOA governance is essential as an early part of any successful SOA initiative.

In fact, SOA governance not only offers the means to control business Services, it also bridges corporate and IT governance as well. SOA describes business Services that enable workflow that drives the management and enforcement of corporate policies. While organizations are familiar with enforcing corporate policies manually using written guidelines and human interactions, with the right infrastructure in place, SOA governance can automate corporate policy communication and enforcement, reducing both the risk and cost of compliance.

SOA governance not only offers the means to control business Services, it also bridges corporate and IT governance as well.

II. The Business Motivation for Governance

The concept of governance is drawing substantial attention in corporate boardrooms and technical meetings alike, as companies struggle with complex regulatory compliance pressures, increasing globalization, enhanced competition, and the maturation of their markets. Perhaps the greatest challenge facing organizations who are implementing some kind of corporate governance initiative is simply the fact that people don't particularly like to be

Thank you for reading ZapThink research! ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences 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.

Earn rewards for reading ZapThink research! Visit www.zapthink.com/credit and enter the code **HPGOV**. We'll reward you with ZapCredits that you can use to obtain free research, ZapGear, and more! For more information about ZapThink products and services, please call us at +1-781-207-0203, or drop us an email at info@zapthink.com.



governed. From the corner office, governance appears to be a clear win, and yet the rank and file tend to resist the implications of governance, especially when they perceive the program to be too onerous.

And yet, management requires control, and in the absence of an effective governance program, such control typically devolves into micromanagement—which is every bit as bad, if not worse than an onerous governance initiative. The challenge for management, therefore, is to scale management control while avoiding micromanagement—empowering users while avoiding policy breaches.

A related business motivation for governance is regulatory compliance. It's a now-familiar joke that ROI for regulatory compliance doesn't stand for *Return on Investment*, but rather for *Risk of Incarceration*. Clearly, criminal liability has made regulatory compliance one of the primary drivers for corporate governance initiatives. And yet, as a business motivator, such compliance has an abrupt limitation, as companies must spend however much it takes to become compliant, but not a penny more.

What is governance?

There are five aspects to governance:

- Establishing and communicating the policies that employees must follow
- Giving employees the tools they need to be compliant with those policies
- Enforcing policies
- Providing visibility into the levels of compliance in the organization
- Mitigating any deviations from established policy

There's nothing in the above five bullets that requires that management involve technology in any way, and in fact, most managers today handle governance in an essentially manual fashion, or where IT serves a secondary support role. One challenge for SOA governance, therefore, becomes identifying which policies are automatable, and then leveraging the benefits of SOA to automate the enforcement of those policies in a flexible way, in order to support changing policies. And yet, even for the most mature SOA implementations, many governance tasks fall outside the realm of automation. Even so, when architecture drives IT governance, taking a Service-oriented approach to such architecture can improve the policy management, flexibility, and visibility necessary for IT governance, and more broadly, corporate governance.

What is a policy?

In order to identify which policies lend themselves to automation, it's worthwhile to take a look at the definition of policy. According to Merriam-Webster, a policy is "a high-level overall plan embracing the general goals and acceptable procedures," which is essentially a broad business definition that would apply in the United Nations as well as the corporate boardroom. However, it's clearly quite difficult to automate a policy given that definition. Another definition of policy is "a set of rules that apply to the performance or behavior of a system and its users," which is a more technical definition of policy that is easier to automate.

This dichotomy between different perspectives on the nature of policies in the organization presents challenges across the organization as both business and IT managers get a handle on governance. Before SOA, business and IT managers shared little common ground with respect to policy enforcement. SOA, however,

One challenge for SOA governance becomes identifying which policies are automatable, and then leveraging the benefits of SOA to automate the enforcement of those policies.

The first step in any governance initiative is to define the governance framework.

helps automate policy activities by treating policies as *metadata* – policy information represented in a standard, machine readable format. Once policies appear as metadata, it becomes possible to bridge the gap between the business and IT perspectives on policies.

Defining the governance framework

The first step in any governance initiative is to define the governance framework. The governance framework essentially prompts the organization to consider the various aspects of policy creation, communication, and enforcement that are relevant to the project or part of the organization that is currently of concern. In particular, to create an effective governance framework, you should answer the following questions:

- Which policies should you implement in the current iteration?
- What policies should you implement first?
- Who in your organization is responsible for creating policies?
- Which policies are automatable?
- How will you create and communicate policies?
- How will you represent policies? In other words, what is the format for your policies?
- How will people within your organization discover policies?
- What tools will people use to follow policies?
- How will management get visibility into policy compliance?
- How will you deal with policy violations? What mitigation approaches will you use?

This governance framework then becomes an outline of the organization's governance initiatives. Note, however, that the first question you should ask involves your current iteration of the governance framework. In fact, taking an iterative, or step-by-step approach to governance is a critical best practice, because it lowers risk and helps to deliver business value as requirements change.

In early iterations, the governance framework will be a simple document, but in each successive iteration, it is important to flesh it out, delineating in increasing detail how you will define and enforce policies as your governance initiative matures. This iterative approach to governance is critical for SOA governance initiatives, as well. In fact, considering what should go in your first SOA governance iteration, and then fleshing out your SOA governance strategy as your SOA implementation matures, is important for achieving a successful SOA initiative. The next section discusses some considerations for the implementation order of different types of policies.

To ensure proper SOA governance, organizations must manage business Services in the context of specific business, IT and regulatory policies that apply to those Services.

III. The Context for SOA Governance

SOA governance at its core focuses on establishing a framework for assuring Service quality over the course of the Service lifecycle. To ensure proper SOA governance, organizations must manage business Services in the context of specific business, IT and regulatory policies that apply to those Services and the consumers that interact with them.

It is vitally important to implement governance in the very early stages of a SOA initiative.

Many organizations are pursuing SOA to abstract their IT infrastructures from inflexible, monolithic application silos to loosely-coupled, business-oriented Services that various parts of the organization can easily share and reuse. SOA can have a transformative impact on organizations by making them more agile, and it can also reduce the cost of integration and application development as well. In effect, SOA enables IT to provide more powerful, flexible tools to business users, as they leverage the power of these business Services in increasingly flexible processes.

With greater user empowerment, however, comes greater responsibility. The last thing IT management desires is to enable an increased circle of users to wreak havoc with their more powerful capabilities. Hence SOA governance becomes a critical part of any SOA initiative. Basically, it's a mistake for organizations to discount governance as optional or a later-phase aspect of their SOA initiative. It is essential that governance planning begin with the initial SOA deployment, providing the framework, processes and practices for scaling the SOA initiative.

Getting started with SOA governance

SOA governance focuses on the creation, communication, and enforcement of policies that apply both to the design time aspects of Service creation, publication, and reuse, as well as the runtime aspects of Service operations, including service levels and the management of Quality of Service (QoS) metrics. It is vitally important to implement governance in the very early stages of a SOA initiative. Even for SOA pilots, which may only deliver a few basic Services, organizations should still deal with certain design time and runtime governance requirements, including the creation and enforcement of the following policies:

- Core security policies – unless your Services are for general, public consumption, you'll need to establish security policies for access control as well as threat management.
- Versioning policies – if you don't plan on updating Services over time, then you will find you must update Service consumers whenever you have a new Service version, which defeats the purpose of building loosely coupled Services.
- Publication policies – to lay the groundwork for future Service reuse, you should create and enforce policies for the publication of reusable Services.
- Quality of Service policies – how your SOA management capability will handle runtime policies for QoS.

In addition to managing the policies above, early SOA iterations should also include the following considerations:

- High-level policy lifecycle – even early on in your SOA planning, you should sketch out how you plan to update and replace policies.
- Auditing and measurement planning – your SOA pilot might not have an auditing requirement, but you should still plan ahead for what metrics you will want to track and how you will want to track them.

It's clear from the above tasks that getting started with SOA involves policies that specifically apply to the Services that form the core of the SOA rollout.

IV. Implementing SOA Governance

SOA governance, in addition to the more traditional software development lifecycle (SDLC) checkpoints and role-based review signoffs, focuses on the creation, communication, and enforcement of Service policies. Service policies are metadata that consist of a set of constraints and capabilities that govern how Services and their consumers interact. Simple policies typically include rules describing who can access a Service and what credentials they need, how messages should be routed to the Service, and what service-level agreements (SLAs) apply to the Service.

The bottom line for SOA governance centers on metadata. Metadata, or data about data, are information about business Services that enable discovery and appropriate usage of those Services. There are many different types of metadata relevant to SOA implementations: Service contracts, policies, XML schemas, and process and Service configurations are among the many different types of metadata relevant to the Services in a SOA implementation. In fact, SOA governance is in essence a combination of policy, process and other metadata.

In the tightly-coupled world of traditional integration, each system and application deals with its own metadata separately. In the SOA world, however, Services externalize these metadata to enable the use and governance of the Services. As a result, managing metadata is a critical task within any SOA initiative. Because SOA policies are metadata, one of their key advantages is that they are declarative, which separates the policy from other business logic. In other words, organizations can change policies without having to change Services. This externalized, declarative nature of SOA policies requires careful management of the policy lifecycle, which requires a trusted system of record for policies, and the processes to create, modify and approve those policies.

Managing metadata is a critical task within any SOA initiative.

SOA Policies: the Key to SOA Governance

Because of the distributed, heterogeneous, and dynamic nature of SOA, it's especially important to govern business, technical, and other policies as they apply to SOA artifacts. Such policies define configurable rules and conditions that affect Services during their lifecycle. It is important, therefore, to validate Services before publishing them, as well as to enforce specific standards and behaviors at runtime.

Furthermore, because SOA implementations center on metadata, it is critical that organizations represent SOA policies themselves as metadata. In essence, the organization can codify reusable policies that they can then associate with Services. This linkage between Services and policies enables the automated validation of Services and the enforcement of particular policies. Encoding policies as metadata enables organizations to automate the process of validating and enforcing compliance to those policies both at design time and runtime.

SOA governance requires that organizations take business policies and transform them into metadata-based rules that can help automate the process of validating and enforcing compliance with those policies.

SOA governance requires that organizations take business policies, typically in written form, and transform them into metadata-based rules that can help automate the process of validating and enforcing compliance with those policies in both design time and runtime environments. Companies must then manage policies through their entire lifecycle. In general, policy lifecycle management within SOA focuses on ensuring the quality, performance and applicability of available Services, enabling Service consumers to discover and reuse Services as well as other artifacts, managing Service versions, handling the security of Services and other SOA artifacts, and assessing and managing the impact of change across all Service consumers. Managing policies also includes providing visibility into whether people are following policies, as well as handling policy

infractions. Such policy management tasks are also an inherent aspect of IT governance, as well.

At design time, the goal of SOA policy management is to detect and resolve quality issues before putting the Services into production. At runtime, then, organizations must also implement runtime policy management for monitoring and automatically enforcing policies during the usage of Services. Such runtime policies may focus on security, QoS, or other requirements for the behavior of the Services in a production environment.

SOA governance requires the following capabilities:

- *Policy Management* – defining and maintaining reusable policies over the course of the Service lifecycle.
- *Policy Association* – applying policies to Services, often through the use of a SOA registry. It's usually preferable to publish such policies to the registry similar to Service contracts.
- *Policy Enforcement* – Enforce SOA policies in practice, either via the Service repository at design time, via SOA management tools at runtime, or via various types of policy enforcement points on the network, depending upon the type of policy.
- *Policy Reporting* – providing visibility into policy compliance via reports that the Service repository can store.

Here are some practical steps for automating policies. First, conduct a policy inventory to uncover the policies that are a priority. Next, decide which policies are automatable. In other words, identify those policies that you can represent as metadata that your policy management tools can understand. Then, decide on level of granularity for your policies. Note that not every policy management or enforcement tool represents policies with the same level of detail, so it's important to develop a consistent format for representing the policies.

At this point you must translate policies into a system-understandable format. Standards like WS-Policy and WS-SecurityPolicy can aid somewhat with this formatting issue, but unfortunately, these standards can only help in rather narrow situations. In the general case, it will be important to either develop your own XML-based policy specification, or encode the policies directly into the policy enforcement system, which represents policies according to the tools' own internal specifications.

Once you have fully defined your policies, you must figure out how to enforce policies in practice. Policy enforcement essentially depends on the type of policy. For example, an XML firewall might enforce a security policy, while a Service repository might enforce a Service reuse policy. SOA management tools enforce many runtime policies, while identity and access control solutions are adept at enforcing access management policies.

Finally, it's important to identify techniques for long-term policy maintenance, as the organization creates, modifies, and retires its policies. In addition to these policy lifecycle considerations, you should also consider how you will scale policy management, either through a centralized or federated approach.

The Role of the Registry/Repository

To provide the capabilities we described above, it is essential to leverage a tool known broadly as a registry/repository, because it combines the capabilities of registries and repositories. The dictionary definition of *registry* is a place to keep

official records. In the technical sense, a registry is an authoritative store of information that relates to a particular task at hand. Like a gift registry that provides listings of the items you wish your friends to purchase, or the Windows registry that stores pointers and configuration information for how the operating system should handle the various resources on your machine, registries store metadata that reference the particular assets of interest, without actually containing those assets.

The store that actually contains those assets is the *repository*. A repository, like the gold repository at Fort Knox, stores the assets and typically controls access to them. So, while a registry records official information that relates to an asset, the repository stores the assets themselves. Here's where the confusion starts: what assets should you register in a Services registry, and what assets and underlying artifacts should you place in a repository? In the context of SOA, consider all the artifacts that relate to the Services. Where do models go? Where do functional specs go? How about policies, schemas, and transformation maps? While a registry might be the right place to store a Service definition and associated policy metadata, it is not the right place to store service artifacts such as models, maps, shared keys, and transformational schemas.

There's no reason, however, for a single product not to offer the capabilities of both a registry and repository, giving rise to the merged category of registry/repository. Such products offer the reference metadata of the registry, as well as the asset storage and management of the repository. Even though a single product might offer all such capabilities, however, developers have different needs for registry/repositories at design time than systems require for runtime operation.

Design time registry/repositories help developers locate assets, make decisions about which ones are best to use, and understand the various costs involved in their consumption. Runtime registry/repositories help systems make automated, policy-based decisions about Service selection and access. The SOA registry/repository forms the system of record for policies and provides the foundation not only for governing policies, but also for managing associations among policies, Services and associated artifacts. Registry/repositories are also able to provide for advanced SOA governance capabilities, including collaborative policy management, Service promotion, Service usage reporting, taxonomy management, visibility into SLAs and levels of performance, as well as integration with quality assurance and production management tools.

The SOA registry/repository forms the system of record for policies and provides the foundation for governing policies.

Avoiding SOA pitfalls from the lack of governance

To fully understand the importance of SOA governance, let's take a look at what happens when a SOA initiative does not have proper governance in place at the beginning of the SOA initiative. First of all, an ungoverned SOA implementation can lead to unintended consequences, including the lack of sufficient reuse, inconsistent or incompatible Services, increased support costs, and challenges with updating the versions of Services. Furthermore, because regulatory compliance can be very complex, with multiple regulations and jurisdictions in play at once, and also because regulations are essentially arbitrary, lack of governance can lead to unexpected noncompliance with regulations, by failing to associate key policies with Services.

Lack of sufficient governance can also lead to various security breaches by allowing arbitrary access to Services, leading to the exposure of confidential information, unwanted access to internal systems, and other security threats. Fundamentally, without proper governance, all of the business benefits of SOA,

including increased agility, reduced costs, increased asset reuse, and improved visibility, are all at risk.

Finally, organizations who delay the implementation of governance and roll out their SOA initiative without a proper SOA governance framework in place run the risk of implementing Services as well as applications that consume those Services that are not able to follow critical policies that a future governance effort would require. For example, without proper support for versioning and depreciation policies in place at the beginning of a SOA initiative, an organization may have to scrap their efforts and start over when they realize they should have built support for such policies into their implementation from day one. Implementing SOA governance at the beginning of the SOA initiative, therefore, can reduce the overall risk of the initiative, and also avoid expensive and time-consuming rework down the line.

V. Expanding the Context for SOA Governance

Of all the SOA best practices that ZapThink espouses, perhaps one of the most important is to take an iterative approach to any SOA rollout. Take the initiative one step at a time, and show business value at each step. Such an iterative approach can lower risk, avoid pitfalls, and build broad acceptance for SOA across the organization.

Taking an iterative approach to SOA governance is a best practice.

Taking an iterative approach to SOA governance, therefore, is also a best practice. Early iterations should typically focus on creating, communicating, and enforcing essential policies for a few key Services. As the SOA rollout progresses, however, SOA governance becomes less of a separate project and more a set of best practices for the organization as a whole. For this reason, organizations should think about SOA governance in the broader context of IT and corporate governance.

Taking an iterative approach, however, does not mean putting off the more advanced tasks. Rather, it means getting started on tasks further down the roadmap as you make progress with the more immediate activities. SOA governance is no different. In fact, ZapThink believes one of the critical success factors for SOA is the establishment of strong SOA governance as one of the first tasks in an overall SOA initiative.

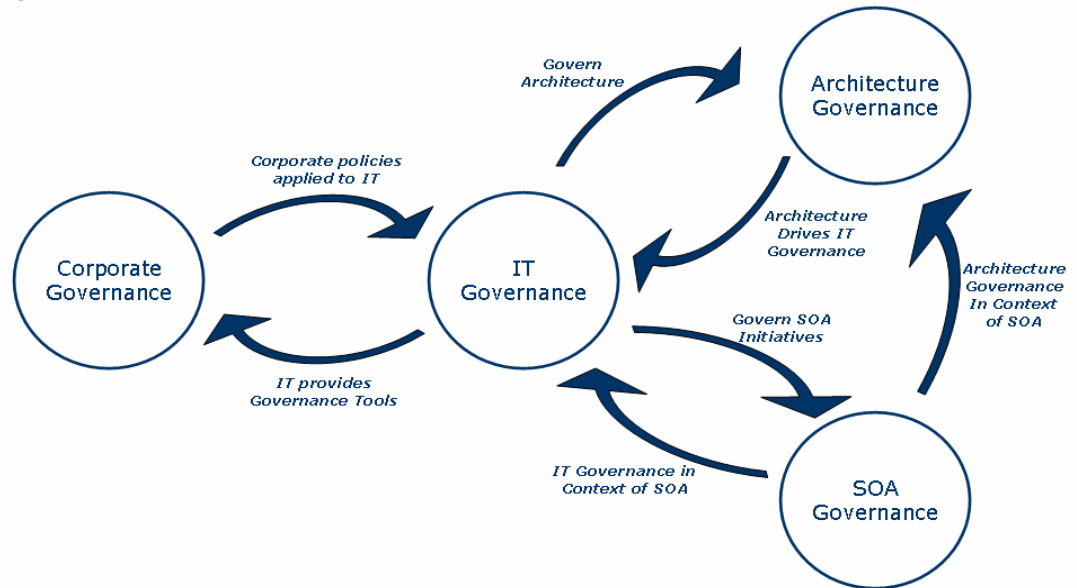
Governance in the context of SOA

Building a governance framework is one step toward implementing what ZapThink refers to as SOA governance “in the broad.” SOA governance in the broad goes beyond the governance of Services within a SOA initiative (SOA governance “in the narrow”), and essentially considers how having SOA in place will improve IT governance overall, and more broadly, corporate governance as well. Basically, SOA governance in the broad involves governance in the context of SOA more so than governance of SOA initiatives. The question is, therefore, what benefits does SOA provide that are particularly appropriate for satisfying governance requirements.

In fact, SOA offers three core capabilities that enable governance: policy management, visibility, and flexibility. Because SOA represents policies as metadata, it’s possible to represent a broader set of policies as metadata than simply those that apply to Services. Because SOA abstracts heterogeneous data sources across an organization, SOA techniques can provide visibility into levels of compliance with policies across the company. And finally, SOA’s core agility benefit helps organizations deal with policy change.

In fact, while SOA governance in the narrow applies IT governance to the governance of SOA initiatives, SOA governance in the broad both places IT governance in the context of SOA, as well as architecture governance in the context of SOA, as the figure below illustrates:

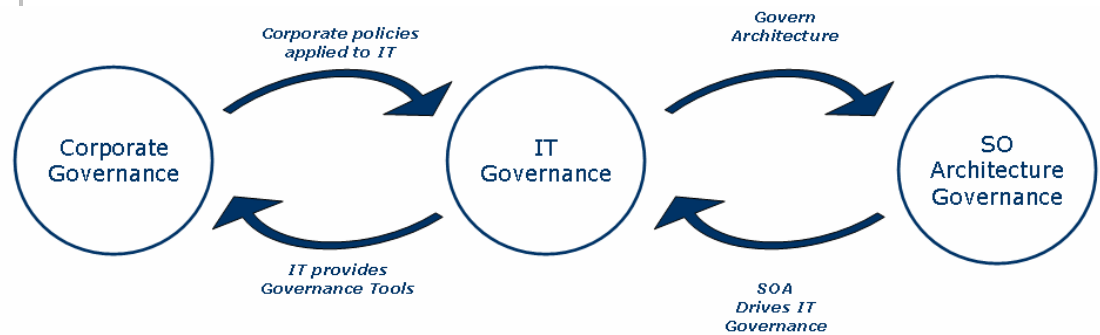
Governance Relationships as Organizations Implement SOA



Source: ZapThink

The figure above, however, suffers from an element of serious shortsightedness. While architecture governance enables IT governance, ZapThink believes that over time, Enterprise Architects will take a Service-oriented approach to architecture governance, as SOA best practices become the standard approach for organizing IT resources to meet the needs of the business. At that point, the architecture governance and SOA governance circles will merge into a Service-oriented approach to architecture governance, as the final illustration below suggests:

Governance Relationships once SOA is Established



Source: ZapThink

In fact, ZapThink believes the figure above illustrates the true future of SOA governance for enterprises around the world, as they fully implement SOA.

VI. HP Business Technology Optimization for SOA: Combining SOA Governance, Quality, and Management

HP's offers a solution for SOA governance as part of the *HP Business Technology Optimization (BTO) for SOA* strategy. This strategy offers the ability for customers to initially focus on either governance, testing or operations. *HP SOA Systinet* software is at the core of HP's SOA governance solution. Systinet acts as a single system of record for discovering and understanding business Services, It provides visibility into how organizations are publishing and discovering Services. Systinet also provides policy management that enables the creation and enforcement of policies.

Systinet includes the following capabilities:

- *SOA repository* – manages Services, metadata, and artifacts, and relationships among them.
- *Registry* – provides for standards-based access and interoperability.
- *Policy management* – automates policy and conformance validation.
- *Contract management* – formalizes the relationships between Service consumers and providers.

In addition to the Systinet registry/repository, the *HP Business Availability Center* software helps businesses manage and optimize SOA environments at runtime so that business Services deliver the QoS and SLAs that the business requires. Business Availability Center provides runtime policy management and service-level management capabilities. Further, *HP Quality Center* software and *HP Performance Center* software provide organizations with a quality management platform that enables them to manage the quality process, ensure functional quality, and validate performance of Services and composite applications.

Systinet for SOA Governance

Systinet is a SOA governance and lifecycle management platform, providing a SOA system of record and a set of capabilities for establishing the visibility, trust and control essential for successful SOA implementations. Systinet offers policy management capabilities that centralize policy-driven management of distributed SOA components. By providing a standards-based approach to defining and managing SOA policies, Systinet automates the enforcement of Service validation, compliance and integrity. Systinet also provides real-time visibility into the status, efficiency and overall conformance of the Services in an enterprise SOA implementation.

In particular, Systinet offers the following capabilities:

- *Policy definition* – HP preloads Systinet with a library of policies and assertions for WS-I Basic Profile, checks for the validity of WSDL and XML schemas, document integrity, and other standard and best practices policies.
- *Policy lifecycle management* – Systinet offers a policy verification and approval process, along with a standards-based approach to associating policies with business Services.

- *Policy validation, compliance and reporting* – Systinet provides both design-time and runtime governance capabilities, including policy validation and compliance at design time, and enables the sharing of policies within a run time environment.

VII. The ZapThink Take

As organizations achieve levels of success with SOA, they are coming to realize that SOA should apply to the enterprise as a whole. Enabling the business to leverage IT capabilities in flexible, governed ways is a fundamental requirement of the business. As such, IT is an enabler of Service Orientation, but SOA in essence is more than an IT initiative—it is a business initiative.

Governance, as well, is more of a business imperative than a simple IT project. And yet, the enterprise calls upon IT to provide tools for governance, and IT is now calling upon the architects to leverage SOA for better governance. For this reason, SOA governance is at the eye of the SOA storm in most enterprises.

SOA governance is clearly critical for realizing the promise of SOA, and organizations should consider SOA governance at the beginning of any SOA initiative. Furthermore, SOA governance helps to resolve the conflict between user empowerment and IT management control. Before SOA, IT management sought to maintain control, and doled out limited capabilities to users, because if they provided too much in the way of user capabilities, the users might violate any number of important policies.

The Service-Oriented approach allows IT to empower a wide range of users to meet the needs of the business in myriad ways, because now we have formalized the practice of SOA governance, which enables business user empowerment in the context of policy-based control. In a fundamental way, therefore, governance is the key to the success of SOA.

With its acquisition of Mercury and its Systinet division, HP has propelled itself into a leadership position in the SOA governance space. For organizations who are ramping up their SOA initiatives, HP Systinet and the rest of the HP BTO for SOA Strategy can assist in what ZapThink calls SOA governance in the narrow. But as those projects become broad, enterprise initiatives, HP is well-positioned to help its customers leverage SOA for IT governance, and more broadly, corporate governance—in other words, SOA governance in the broad.

With its acquisition of Mercury and its Systinet division, HP has propelled itself into a leadership position in the SOA governance space.

Copyright, Trademark Notice, and Statement of Opinion

All Contents Copyright © 2007 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 advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences 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 helps its customers in three ways: by helping companies understand IT products and services in the context of Service-Oriented Architecture (SOA) and the vision of Service Orientation, by providing guidance into emerging best practices for Web Services and SOA adoption, and by bringing together all our audiences into a network that provides business value and expertise to each member of the network.

ZapThink provides market intelligence to IT vendors and professional services firms that offer XML and Web Services-based products and services in order to help them understand their competitive landscape, plan their product roadmaps, and communicate their value proposition to their customers within the context of Service Orientation.

ZapThink provides guidance and expertise to professional services firms to help them grow and innovate their services as well as promote their capabilities to end-users and vendors looking to grow their businesses.

ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into the best practices for planning and implementing SOA, including how to assemble the available products and services into a coherent plan.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOA by vendors, end-users, and the press. Respected for their candid, insightful opinions, 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 November 2000 and is headquartered in Baltimore, Maryland.

ZAPTHINK CONTACT:

ZapThink, LLC
108 Woodlawn Road
Baltimore, MD 21210
Phone: +1 (781) 207 0203
Fax: +1 (815) 301 3171
info@zapthink.com

