CHAPTER 7

SOA Governance, Organization, and Behavior

Service-oriented architecture (SOA) is a challenge for both business and information technology (IT) organizations in light of the organizational and behavioral issues that attend an SOA initiative. However, SOA has the potential to impact IT governance and enterprise architecture perhaps more than any other processes.

There are many symptoms of the need for change in an IT organization. Among them are stovepiped architectures, where various applications and computing platforms cannot share data or interoperate in support of common processes or business functions. They also include costly and brittle integration strategies implemented to alleviate the problem of stovepiped architectures. Such strategies may address some of the immediate integration challenges, but they only push the root cause further under the carpet, hidden from scrutiny. Imagine that you are an archaeologist. Your job is to analyze physical remains and artifacts in order to draw conclusions about the behaviors of the people who left the artifacts behind. Often these physical artifacts must be carefully excavated and documented to record the spatial context and position in the earthen matrix in which they have been found. These artifacts include flint tools, ceramics, animal bones, fire-cracked rocks from fire pits, decorative beads, and so forth.

Now, some of these artifacts will provide immediate clues as to the date of the site and the cultural affiliation of its prehistoric peoples. Arrowheads and ceramic styles often quite accurately point to

the period in history when a particular group inhabited an area. However, other behavioral issues cannot as readily be ascertained. Why were these people at this specific location? Why were houses located as such? Around what organizational principles was the village structured? How was their society governed? What were the rules? Was there class differentiation or was this an egalitarian group?

Examination of physical remains can answer some, but not all, of these behavioral questions. No matter how skilled you are as an archaeologist, you will have a difficult time drawing conclusions about behavior from the assemblage of artifacts. The behavioral granularity is very coarse and cannot eludicate the thought processes of individuals or the collective civilization.

Now, imagine you're an IT archaeologist (of course, there are no such titles, at least not yet...). Your job is to reconstruct the behavior patterns that resulted in the assemblage of technology artifacts in an organization. What were the collective and individual decisions that led to the purchase of a particular mainframe system? What behaviors led to a decision to install client-server platforms for enterprise applications? What caused the organization to pick a particular vendor platform over another? Why are organizations so interested in Open Source software now? What behavior patterns does that choice imply?

An organization's current IT architecture is a collection of artifacts, an assemblage of physical (and even mental) artifacts in the form of employees with specific knowledge of these "heritage" systems that accumulated through years of organizational and individual behaviors and choices. Behaviors caused your current IT architecture to be in its present state.

However, behaviors not only resulted in your current assemblage of IT artifacts; they also attempted to resolve challenges by implementing processes and chartering organizational functions whose sole purpose was to make sure IT systems worked and supported business needs. Central architecture organizations were formed, sometimes as federated teams from various business units and sometimes as central organizations chartered to oversee IT architecture and govern the technology and standards allowed in the architecture.

The organizational recognition of the increased complexity of IT systems required dedicated oversight and architectural attention. This role befell the chief technology officer (CTO) and chief architect. In

the past, generally it was the CTO who had oversight for the organization's architecture and technology. Now, however, the SOA movement is presenting new challenges to enterprise architecture organizations. The architectural goal of "build things and make them work" is no longer good enough.

ARCHITECTURE'S ROLE IN AN SOA

The definition of "architect" is: *one who designs and supervises the construction of buildings or other large structures*. The appropriateness of the building construction metaphor has been discussed at length by others. Here we only say that the notion of building IT architectures that emulate rigid fixed structures has clearly been realized, much to the chagrin of business leaders who need a better way to respond dynamically to changing business conditions without being hindered by the digital concrete of current IT architectures and enterprise applications. Perhaps the very title "architect" has resulted in artifacts that are like buildings—fixed, rigid, sturdy, unchanging—as opposed to fluid, agile, flexible, nimble, or malleable. The "building" metaphor of architecture is too static to suit the requirements of IT based on SOA. "Architecture" must become an adaptive process that mediates business and technical changes and ensures that IT solutions can adapt and change in conjunction with business changes.

The current role and process of architecture must be reexamined in light of the demand for SOA and reusable services. The past role of enterprise architecture must be attuned to the nuances of SOA in today's business enterprise. Again, recall the IT artifacts we are left with. The behavior that caused these artifacts indicates processes and capabilities that did not emphasize interoperability and shared reusable services. These IT artifacts consist of rigid IT architectures characterized by legacy systems, inflexible "digital concrete" of enterprise applications, and a portfolio of applications cemented with integration software to make them interoperate.

The process and role of enterprise architecture must be reengineered to provide the vision, leadership, and active participation in the implementation of SOAs based on services. Architects must adapt to the new realities of IT and enterprise architecture—from getting systems to work to making services work together.

SOA will fail unless the process of architecture is radically changed from one of static advice and creation of color PowerPoint slides, application blueprints, and architecture roadmaps to one of actively shaping and implementing flexible and reusable IT assets that support business processes. In other words, SOA.

DYNAMIC ARCHITECTURE VERSUS STATIC ARCHITECTURE

Agile SOA is the key concept. What is agile SOA? Agile SOA is based on services that can be enhanced and extended without negatively impacting current consumers. Agile SOA is predicated on an agile services lifecycle process of identifying, modeling, and implementing services quickly in response to business and IT requirements. Agile SOA is predicated on flexible enabling technology solutions that can facilitate and accommodate the inevitable environmental, business, and technology changes. Service-oriented agility is the concept most organizations seek, yet they have not determined how "agility" translates into an operational concept of SOA, services, the enabling infrastructure and management processes SOA requires.

Once SOA is under way in organizations, they must adjust their enterprise architecture process from a static offline advisory function to an active shaping of IT flexibility and asset reusability. Exhibit 7.1 depicts the potential impact of an SOA on existing IT organizational structures—IT governance and IT architecture. The drivers of an SOA initiative, which are the motivating forces for SOA change in an organization, will superimpose an SOA governance model onto the existing structural makeup of an IT organization. The SOA governance process will impact IT governance, enterprise architecture, and other governance processes within the organization. The impact on each of these IT institutions may be minimal, but chances are the impact will be somewhat profound. Either way, an organization should be prepared to tune and adjust the IT governance and enterprise architecture models as the requirements of SOA become more mission critical.

Before taking this task on, we first have to devise a general model of IT architecture. Once this model is established and understood, we will adapt this to an SOA initiative. Chapter 8 addresses the new

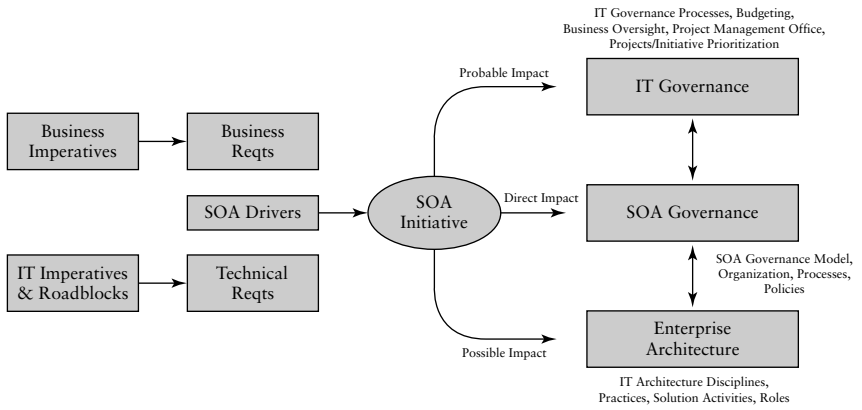

EXHIBIT 7.1 SOA Governance vis-à-vis an IT Organization

requirements of enterprise architecture to meet the demands of SOA. Here we shift our attention to the bigger picture of SOA: SOA governance and behavior.

SOA: SPATIAL AND TEMPORAL CHALLENGES

SOA is not a big bang implementation model based on a single momentous event. SOA is a conceptual IT architecture, based on reusable services, that is achieved over multiple implementations of "services" projects across an organization through time. The "services" are not implemented centrally. They are implemented through many projects over time, potentially across multiple departments, business processes, and business units, eventually to reach some critical mass of SOA benefits. SOA is accomplished through continuous iterations.

However, SOA is a spatially and temporally distributed process, and these features of SOA are very challenging for many organizations. How do you enforce a consistent set of design, reuse, and interoperability standards across a spatially diverse organization so that the ultimate benefits of SOA can be realized? How do you manage the temporal challenges of SOA, where services developed using one generation of Web services standards have the potential for incompatibility with a later generation of Web services standards? SOA

governance and policies address these issues. Policies are enforced by a combination of decree, education, employee management, incentives, and overall enforcement during service design, publishing/discovery, and at run-time.

SOA GOVERNANCE OVERVIEW

SOA governance refers to the organization, processes, policies, and metrics required to manage an SOA successfully. A successful SOA is one that meets defined business objectives over time. In addition, an SOA governance model establishes the behavioral rules and guidelines of the organization and participants in the SOA, from architects and developers to service consumers, service providers, and even applications and the services themselves. These behavioral rules and guidelines are established via a body of defined SOA policies. SOA policies are specific and cover business, organizational, compliance, security, and technology facets of services operating within an SOA.

SOA governance consists of the organization and processes required to guide the business success of an SOA. SOA governance defines and enforces the policies that are needed to manage an SOA for business success.

SOA governance is crucial to transitioning from point-to-point Web services to reusable business services. SOA governance involves defining the organizational issues, the governance processes and procedures, and the necessary SOA policies required to manage services and the SOA infrastructure throughout the SOA lifecycle. While governance addresses the organization, processes, and required policies for managing an SOA, the SOA policies are the essential ingredient that must be enforced at service design, publishing, discovery, invocation, and management. Policies can be business policies, security policies, standards compliance policies such as WS-I, or internal standards and other technical policies.

For an SOA, SOA governance:

- Provides overall SOA oversight and management
- Defines architectural standards, developer guidelines, and specific policies that are enforceable across the services lifecycle—from design, development/enablement, publishing, discovery,

and run-time and across all architecture and development processes
- Clarifies services ownership and stewardship across the organization, including budgeting processes, maintenance responsibilities, infrastructure management, and so forth
- Defines services development and lifecycle management issues (e.g., service design, development/enablement, publishing to a services registry, discovery, invocation/run-time, management, maintenance, quality assurance, versioning and reuse)

SOA governance is a master thread running through the organization, processes, and roles in an SOA. It holds everything together and guides the activities of an SOA toward achieving its stated business and technical goals. An SOA governance model includes these elements:

- *Organization.* Defines the organizational structure and management processes for SOA oversight and management control.
- *Processes.* Defines the roles, responsibilities, and procedures for managing SOA processes and activities, including design, development, publishing, maintenance, and so forth.
- *Policies.* Consists of the body of SOA policies that will be enforced at design and run-time, including business policies, industry and organizational standards, security standards and policies, release procedures, publishing, reuse.
- *Metrics.* Must include business metrics, process metrics, performance metrics, service-level agreements (SLAs), and SOA governance metrics, such as SOA conformance and developer exception reporting.
- *Behavior.* Creates a behavioral model through its body of defined policies, which instills and enforces the behaviors necessary for the business success of an SOA. Behavior includes human behavior, such as management, architects, developers, consumers, and providers of services, but it also includes behavior of services as they interact and interoperate with the context of orchestrated business processes enabled by services. Behavior, culture, and both organizational and individual incentives are critical to instilling a reuse and SOA culture. Change management practices will help organizations drive the necessary changes in order to shift behavior to support SOA initiatives.

Each of these dimensions of SOA governance is explored in subsequent sections.

ORGANIZATION OF GOVERNANCE

If you are assigning an SOA core team, an architecture oversight board, an XML core team, or the like, you are creating an organizational model for SOA governance. Marks and others have captured the impact of organizational structure on the performance of a given process.[1] SOA governance is no different. How the governance organization is established will determine how it functions in a specific enterprise context. Therefore, attention should be paid to the structure, participants, and roles of the SOA governance organization as well as how it impacts existing IT and business governance functions.

SOA initiatives can impact IT organizations in a number of ways, as shown in Exhibit 7.2. An SOA initiative, along with an appropriate SOA governance model, will impact existing IT governance processes and the existing enterprise architecture model. SOA places new decision emphasis on projects where in some cases reuse and interoperability take precedence over the needs of individual projects within business units. In other words, the SOA greater good will overrule specific requirements of a business unit if there is reuse and leverage to be obtained from such an initiative.

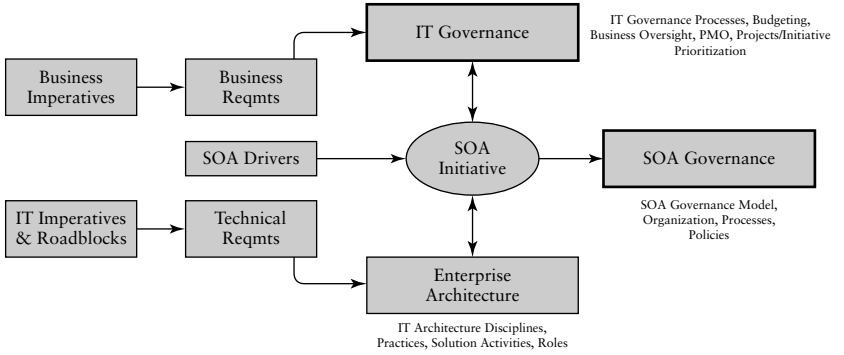

EXHIBIT 7.2 Relationship of SOA Governance to Overall IT Organization

Consider the case where a project budget may increase to obtain reuse of services. If reuse can be clearly demonstrated, then the increased budget can be justified. However, any incremental budget may have to be provided by a central SOA organization that is responsible for overall SOA projects, shared infrastructure, and special investments that are SOA-specific. Furthermore, reuse metrics suggest a 50% incremental cost to develop software for reuse. Although these numbers may or may not be appropriate for services, especially when the development process is different in many ways, the incremental cost and potential elapsed time to ensure reuse must be factored into budgeting and governance decision criteria.

SOA governance will force certain decisions to be resolved above the individual business unit and project level. The governance organization and processes must accommodate these scenarios.

SOA governance impacts existing enterprise architecture as well. (This topic is covered in Chapter 8.) Note, however, that the SOA governance model must incorporate decisions about the current architecture model, organization, process, and skills. We have documented the fact that current architecture practices are not tuned to the nuances of SOA. Enterprise architecture, application architecture, data architecture, and related architectural disciplines will have to be upgraded and tuned to a services model based on an SOA.

Exhibit 7.3 depicts how an SOA initiative may impact existing IT architecture within an IT organization.

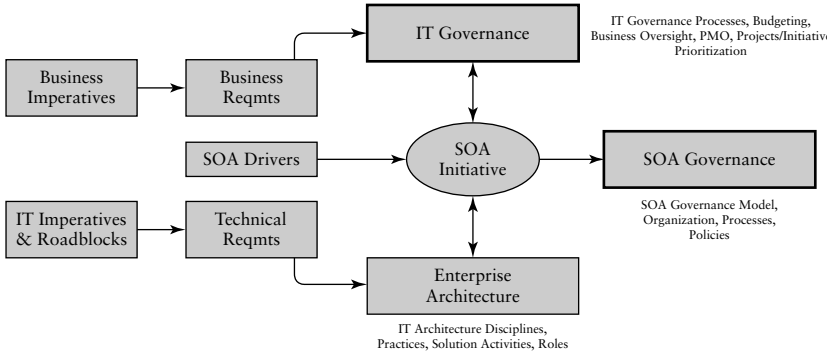

EXHIBIT 7.3 Enterprise Architecture May Be Affected by an SOA Initiative

Based on the specific SOA strategy, enterprise architecture, and IT governance model, an SOA governance model and its associated body of policies will be developed to implement and enforce those SOA and enterprise architectural goals.

WHAT DOES SOA GOVERNANCE DO?

What specifically are the activities that SOA governance provides oversight for? How is SOA governance accomplished? And who does it? SOA governance encompasses high-level activities and processes. SOA governance:

- *Determines SOA architecture oversight.* Who is responsible for the SOA technical architecture? Who owns the standards and monitoring of conformance to the SOA policies? How does the role and process of enterprise architecture change in an SOA context? Who determines appropriate levels of business service granularity and generality?
- *Establishes SOA policies.* Defines and enforces policies that will ensure conformance to the SOA goals, standards, and overall objectives across all process of SOA, including design, publishing, discovery, and run-time. Who will have access to the service? How will credentials be managed? What are the security policies for the SOA?
- *Establishes funding models.* Budgeting practices and funding models are challenges that must be addressed early in the SOA process. Who will pay for building and maintaining services? Who will pay for new shared SOA enabling technology when it is required by a specific project yet will be shared across business units? How will the *SOA greater good* be funded for shared services and infrastructure? Many organizations budget at the project level, where the project and its funding are subsidized by one business unit. This model creates conflict when SOA seeks the development of shared reusable services across business domains. A funding model that creates organizational incentives to develop reusable services for the greater good of the organization is essential. Creating this will require some creativity, new incentive models, and authority to implement these kinds of changes.

- *Implements the SOA governance process.* How will the interdependencies of shared services be managed within the SOA? What organizational and process challenges will be faced? Who will mediate conflicts between organizations?
- *Governs services definition, creation, and publishing.* How will services be defined, developed, and later modified? Who will have design authority? Upon whose requirements? Who owns the services? Who governs publishing and discovery? What technology platforms are necessary to implement SOA governance?
- *Establishes policies and processes for quality of service/SLA management.* What quality of service will be provided? Is high availability required by some but not others? Who will enforce the SLAs? What enabling technology will enforce policies and implement management for services?

SOA governance affects more areas, but this list sets the stage for its complexity and criticality.

SOA GOVERNANCE ORGANIZATIONAL MODEL

The SOA governance model below, based on typical preexisting structures within most organizations, may prove useful. The categorization of services follows a tiered model based on whether the services are business process services, infrastructure services, or hybrids.

Exhibit 7.4 illustrates this tiered model for SOA governance based on a tiered approach.

Supporting this generic tiered model, the following SOA governance organizational model may make sense. Depending on the organization, its IT organization and its enterprise architecture model, newly formed teams may be required to implement SOA governance. An SOA core team can assume multiple responsibilities until a formalized SOA governance and organizational model is established.

As you determine the organization, structure, and roles of your SOA governance model, you must consider the existing structures and processes you have as well as possibly adding overlay organizations onto them. This is a challenging exercise, as the process of SOA governance must not be additive to already-burdened job tasks. SOA governance must become the "company way" in all behaviors

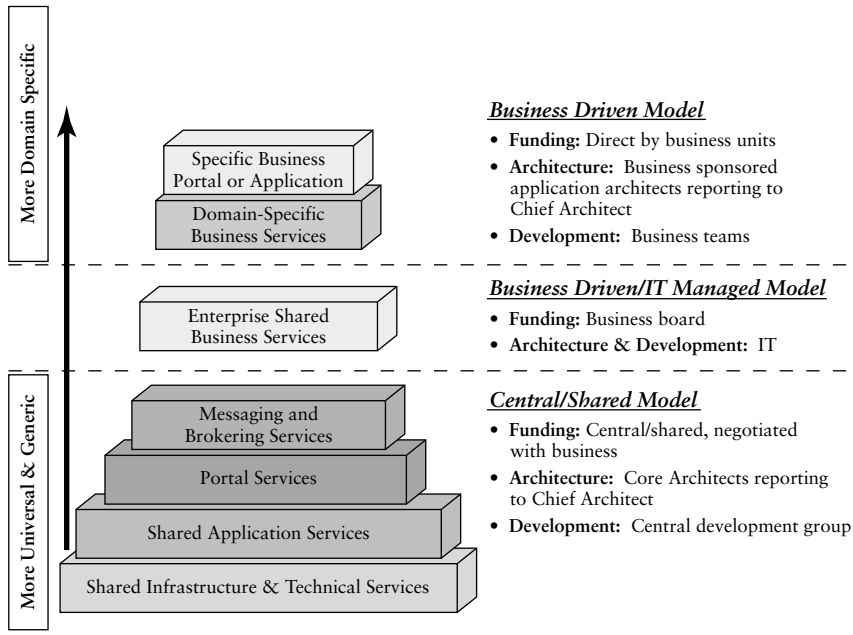

EXHIBIT 7.4 Tiered Governance Example
Source: Graphic Courtesy of BEA

and decision-making processes. The list below describes common SOA governance organizations and structures that may apply to your organization:

- *SOA leadership team (steering committee).* Executive team comprised of business and IT leadership.
  - *Goal.* Ensure SOA efforts align to business and IT strategic goals. Ensure budgets and funding are in place for SOA infrastructure and initial services rollouts. Review and approve SOA roadmap and business initiative roadmaps, project plans, budgets, and so on.
  - *Duration.* Ongoing, quarterly meetings or on-demand as projects dictate.
- *SOA core team.* Senior team composed of senior business and IT leadership.
  - *Goal.* Develop the initial SOA strategy, vision, policy, and governance model, standards and infrastructure, and spearhead

the initial services rollouts. Serve as a catalyst for ongoing SOA efforts. Evangelize the SOA benefits to the IT and business organizations. Be SOA coaches for the enterprise.
    - *Duration.* May disband once formal SOA governance structure and processes are in place and these functions are absorbed by other processes and structures to be described.
- *Process services team.* Senior business team composed of business leadership, process owners, and IT support personnel.
    - *Goal.* Identify and prioritize business initiatives for SOA inclusion. Identify opportunities for services within and across business units. Determine ownership for business services, common process services, and budgeting for these initiatives. Develop business initiative roadmap with SOA core team. Review business initiatives with SOA core team/SOA leadership team, architecture services team (via SOA review board).
    - *Duration.* Ongoing. Chairs Process Services Review Board for inclusion into business initiative roadmap.
- *Architecture services team.* CTO, chief architect, and IT services leads (enterprise services, information service, process services).
    - *Goal.* Create SOA policy and governance model. Identify and enforce architecture compliance to standards, development goals and guidelines, security policies, and business policies. Chair the SOA Architecture Review Board for infrastructure and process services proposals. Ensure all initiatives conform to the SOA governance model.
    - *Duration.* Ongoing.
- *Enterprise services team.* IT infrastructure services team members.
    - *Goal.* Implement and manage the enabling infrastructure for the SOA. Includes baseline horizontal services for SOA enablement as well as security, messaging, audit, and related functions. Member of the SOA core team and Architecture Review Board.
    - *Duration.* Ongoing.
- *Information/data services team.* Data warehousing, analytics, data modules and information delivery team members.
    - *Goal.* Implement and manage the enabling capabilities for information harvesting and delivery to consuming business units, processes, and users. Includes identifying and selecting infrastructure unique to delivering information services, such

as metadata management. Also includes development and ongoing stewardship of the canonical data model.
- *Duration*. Ongoing.

Specific SOA governance roles and responsibilities must be defined for each organization based on its SOA strategy, governance model, and specific business services. This SOA governance organizational reference model may prove useful in establishing an organizational model suited to your particular needs.

SOA GOVERNANCE PROCESSES

The SOA governance process is more than establishing the governance model and the policies that will be enforced. It actually is the process of governing the SOA. The governance process can be challenging because it may be partially manual. SOA governance can include design-time activities, such as design reviews, code reviews, testing and quality assurance processes, and the like. However, SOA run-time processes may require automated management platforms to ensure quality of service, reliability, load balancing, and failover, among many other requirements. Clearly these processes must be automated, using automated policy-enforcement. In the big picture, SOA requires policy enforcement across all SOA lifecycle processes. We call this closed loop SOA governance.

CLOSED LOOP SOA GOVERNANCE

SOA governance must also be enforced in various SOA and IT processes, such as services lifecycle processes (e.g., design, development, and deployment), as well as during SOA and services management, monitoring, analysis, and optimization. We advocate a closed loop SOA governance model. By "closed loop governance," we mean the ability to centrally define governance and SOA policies as well as enforce them across all SOA lifecycle processes—from service design and development to publishing and discovery, and ultimately through

services operations and run-time. Policy and run-time feedback should be captured and fed back into the service design process to provide important feedback on services performance, SLA effectiveness, and overall consumer experience with services. Doing this ultimately provides the closed loop SOA governance model.

Implementation of closed loop SOA governance must include the key lifecycle processes of an SOA, including design, publishing/discovery, and run-time operations. The SOA governance aspects of these major lifecycle processes follow.

Design-Time Governance

Design-time SOA governance is accomplished by discovering, identifying, and inventorying business and technology assets using metadata catalogs. Metadata catalogs are repositories for various IT assets, including executables, design patterns and related knowledge assets, object libraries, software modules, and even services and related artifacts. Metadata catalogs provide support for developers who are implementing reuse policies and best practices. These design-time metadata catalogs integrate with developer tools and integrated development environments (IDE) for all major application development platforms to enable developers to use their normal development tools and processes when they reuse services and other software development assets.

Increasingly, these design-time metadata catalogs provide tools that support SOA governance where the specific policies intersect with the software or services development process. The increased convergence of offline design-time metadata repositories with run-time metadata solutions, such as service registries, will be interesting to watch as SOA implementation efforts mature.

At the completion of service design, the service will be prepared for publishing to a service registry.

Design-time governance requirements include:

- Application of SOA policies to services development processes
- Process policies, such as reuse, design reviews, code reviews, release procedures

- Technical policies, such as schema usage, WS-I conformance, security policies, compliance policies
- Automation through service validation processes
- Access to operational and run-time metadata

Publishing and Discovery Governance

When publishing services to a service registry, there are clear governance processes and policies to be enforced. For example, the publishing process may require eight predecessor steps to be completed satisfactorily first:

1. Complete exposing or development of service.
2. Unit test service.
3. Check SOA conformance of the service to governance model and policies of your SOA.
4. Receive "certification" that the service complies with your policies sufficient to be published.
5. Store the certification into a metadata registry with an association to that service.
6. Begin publishing process; verify that user has authorization to publish services to the registry.
7. If user does not have publishing authorization, he or she must submit the service and conformance certification to the registry owner or librarian who has authorization to publish to the registry.
8. Upon review of the service, test data, conformance certification, it will be published to the registry.

With respect to discovery governance, when locating services available in an SOA, whether by role, function, authorization, or what have you, policies are ultimately what determine a system or individual's access to services.

Publishing and discovery governance issues include:

- Application of policies controlling the service publishing process
- Roles, security, authorization, validation of services and metadata

- Conformance validation prior to publishing
- Application of policies affecting the discovery of services (design-time and run-time discovery)

Run-Time Governance

When consuming or invoking services, policies are enforced by inspecting the SOAP message headers for WS-Policy metadata in the form of assertions about policies asserted by the service providers. Run-time governance and policy enforcement will be essential sooner than most people expect, as major software vendors are planning to offer their software products as bundles of services contained in a services registry that will ship with their software. The real issue here is the potential proliferation of registries in the enterprise with no clear path toward federating them into a single view of the enterprise. A single federated view of all the metadata in an SOA or in an enterprise is essential to optimize reuse of these assets and to manage them all under a given set of governance policies. When there is no federated view of assets, services, and the associated metadata in an SOA, chaos is likely to ensue. Multiple fiefdoms of metadata and services will arise with no possibility of reuse, central management, or overall SOA policy enforcement. Failure to enforce SOA policies means that services may not interoperate because there is no consistent implementation of interoperability conventions and standards or implementation of specific standards and policies specific to that particular organization.

Run-time governance requirements include:

- Enforce policies during service consumption.
- For internal services, enforce internal policies, monitor services, feedback.
- For external services, enforce policies using acceptance criteria to allow consumption of external services.
- Close the loop to design governance by pushing metadata back to the design process.

WHAT IS THE SOA GOVERNANCE PROCESS?

Defining and implementing SOA governance is a series of steps that begin with SOA strategy and planning, business and IT objectives, and the standards and guidelines that are targeted for the SOA. SOA governance is a process that occurs through three high-level steps:

1. Define overall SOA governance model, organization, and process.
2. Define SOA policies to be enforced:
3. Implement SOA governance policy and enforcement

Define the Overall SOA Governance Model, Organization, and Process

The first task is to define the overarching governance model, which determines high-level organization, governance processes, services ownership, budgeting, and funding issues for an SOA. This step establishes ownership and funding models for various classes of services that will be defined and implemented in your enterprise. This overall SOA governance model establishes the operating model and rules for the SOA.

- Define SOA goals and objectives. (This step should have been completed already during the SOA strategy and planning process.)
- Define the SOA metrics, such as business, process, return on investment (ROI), performance and SLA metrics, as well as SOA conformance metrics.
- Define the SOA governance organizational model and governance processes required.
- Define services ownership across the organization and process model. Note that a service taxonomy may be required first to determine who owns what kinds of services. We suggest a simple service taxonomy initially: process services, enterprise services, technical services, and infrastructure services.

Define SOA Policies to Be Enforced

Next we turn to the policies, or the specific "rules of engagement," for designing, building/exposing, and operating services within an SOA. SOA governance is an exercise in futility without enforceable policies that will drive conformance to the SOA vision, goals, and standards. The policies that will be enforced include specific design-time and run-time policies. These policies must support and enable the higher-level SOA governance model. Four steps are necessary for defining policies to be enforced during SOA governance:

1. Define SOA policies needed based on business and technical requirements.
2. Define conformance processes across the services lifecycle (e.g., design, development/enablement, deployment, publishing, discovering, operation/run-time, management, and maintenance activities).
3. Govern the SOA and associated services using the defined policies.
4. Measure conformance to the SOA governance model by examining multiple areas of conformance.[2]

   Policies. What are our policies? Where are they implemented? How are they enforced during design, development, and run-time? Where are the gaps?

   Enterprise services. What enterprise services are being developed or exposed? How are policies being enforced during development? Is policy enforcement automated during the services lifecycle?

   Conformance status. Do our services (and others we consume) conform to our policies? What is the impact of nonconformance on service operations or business processes (e.g., security intrusions, SLA degradation, inoperable services)?

   Impact analysis. What happens to the SOA and associated business processes and business services if a policy is changed (e.g., SOAP policy, adding new metadata to SOAP message headers, message encoding policies, etc.)?

Interdependencies. How will business processes and operations be impacted by changes to services? What mission-critical processes will be impacted or fail due to a service change or enhancement? What regression testing processes must be followed when a service changes and other processes or business units rely on that service?

Exception management. How will policy exceptions be granted for services used by a specific project? What is the impact of policy exceptions? What minimal tier of policies must always be enforced in order for a service to be consumed? Should there be tiers of policies to handle the exception process?

The concept of SOA policies is explained in detail next.

SOA POLICIES: WHERE SOA GOVERNANCE GETS REAL

SOA governance is the body of policies that drives the overall behavioral model of the participants of the SOA and ensures the interoperability of the services operating in the SOA. Behavior of services and behavior of the participants on the SOA are the real challenges of an SOA. Policies define the parameters for the acceptable behaviors of both.

SOA governance is accomplished by policies. Policies are the specific rules that services adhere to at design time and run-time as well as the behavioral policies that developers and architects adhere to. There are thus enterprise policies that all SOA parties must adhere to (e.g., "Reuse services before developing/exposing new services") as well as granular technical policies that ensure architectural compliance, such as "avoid RPC Encoded Web services operations," or "use document-centric messaging wherever possible." The nature of the policies is driven by business and technology requirements, which feed into the overall goals of the SOA.

SOA governance is achieved through the definition of policies. However, it is critical to understand that defining clear enforceable policies as part of the SOA governance model is not enough. Policies must be enforced, at design time, at publishing and discovery time, and at run-time. Enforcement of policies in these offline and online capacities brings into play the technical implementation of policies

that comprise the SOA governance model. But what do we mean by offline enforcement versus active online enforcement of SOA policies?

Offline policy enforcement occurs in meetings according to the governance model, organization, and overall governance process. It can involve design reviews, code walk-throughs, and other checks and balances during the development lifecycle that help architects understand how well SOA policies are being incorporated into various IT projects and adhered to. This is not far from the normal architectural enforcement model of the pre-SOA enterprise. Policies are reduced to documentation, which must be distributed to architects and developers and reinforced to them with active mentoring and ongoing education and training.

However, policies should not be institutionalized as documentation only. Somehow policies must be integrated into the services design, development, and deployment processes and the services publishing, discovery, and operational processes, or at run-time. Policies must be enforced at run-time by consumers and providers as well. Remember, behaviors are conditioned and shaped for all participants and roles in an SOA—human participants, services, applications, and enabling infrastructure.

Enforcing policies in an automated fashion using various technology solutions is essential for run-time SOA policy enforcement. SOA policy enforcement requires the appropriate enabling technology, including tools such as Web services management (WSM) platforms, policy validation engines, service registries, and metadata management solutions (for both run-time policy enforcement and offline enforcement during development). For example, consuming a service from an outside provider requires that the service contract, or WSDL document, be validated for compliance to the consuming organization's SOA and policies, such as the security assertions contained in the SOAP message headers, and the message encoding specified in the WSDL (e.g., RPC encoded versus Document-Literal, etc.).

Even when consuming an internal service, the policies supported by that service should be validated against the SOA policies to verify conformance. This step is important; in some cases, there may not be a solid process for enforcing policies during the development/enablement process and subsequent publishing of the service

to a registry. In fact, a service registry may not even be implemented as part of the SOA enabling technology. Although service registries can help with the enforcement of policies prior to publishing, there is often debate as to when a service registry is needed to manage a particular volume of services. How many services dictate the need for a service registry? Gartner Group, for example, has arbitrarily settled 50 as the number of services at which registries and other SOA infrastructure will be necessary. These are all decisions that must be made case by case, as there are not enough empirical data to suggest a general pattern.

Who Defines Policies?

Policies are defined by multiple members of the IT organization who play a role in the definition of the SOA governance model and overall SOA vision and strategy. IT managers, chief technology officers, chief architects, architects, development managers, team and/or project leaders all can play a role in defining the policies that will comprise the SOA governance model.

Policies ultimately are derived from the business and technical requirements of the SOA initiative and the portfolio of services that will operate in the SOA over time. Therefore, it is likely that an initial body of policies will be defined by an SOA core team to spearhead the implementation of services and SOA in a given organization. In fact, many organizations define their initial policies without calling them policies at all.

Many organizations begin their SOA effort by defining their services design guidelines and best practices within various business process domains. These initial service design guidelines will become the basis for identifying and enforcing specific policies through code reviews and manual SOA governance processes under the oversight of the architects and IT management. Eventually these policies can be implemented as enforceable policies using automation and tools that provide centralized policy definition, management, and policy enforcement across the organization and SOA lifecycle processes.

What Policies Are Required?

Many types of policies must be defined, including:

- *Enterprise policies*. Policies that affect all business units, processes, and roles, such as reuse, security policies, design best practices and standards.
- *Business policies*. Address business issues, including process policies, SLAs and performance criteria, approval levels, spending limits for external services, and more.
- *Process policies*. Who is allowed to publish a service? What minimal standards must be adhered to for a service to be published to a registry? How will versioning of services be managed? How many versions will be allowed? How will new versions of services be advertised to consumers? How will deprecation of older versions be handled?
- *Compliance policies*. Policies that implement regulatory compliance standards and other industry-specific standards, such as HIPAA for healthcare, FIXX and IFX for banking and financial services, and ACORD for Insurance.
- *Technology standards compliance*. Web services standards also apply here, such as compliance to WS-I, appropriate versions of SOAP, WSDL, and UDDI, as well as other related standards including XML Schema, Xpath, and Xquery.
- *Security policies*. Policies that implement the organization's security model and technical standards, such as authorization and authentication policies as well as the standards that will be used to implement security policy. WS-Security standards, SAML, XML Signature, and XML Encryption may be specified for specific use cases of services or the messages sent or received by services.

The body of specific policies will be determined by the overall SOA governance model, defined standards, goals of the SOA, and, of course, the nature of the services that will be exposed or developed internally as well as services consumed from external service providers.

SOA GOVERNANCE IMPLEMENTATION AND INTEGRATION

Implementing SOA governance occurs through a combination of tactics. For organizational aspects of SOA governance, such as services ownership, budgeting issues, and mediating conflicts between organizations and functions, a series of SOA governance forums will suffice. However, some thought must be given to the organizational model for SOA governance. In addition, once an organization model is determined, the processes that implement SOA governance must be considered, such as how SOA governance will be implemented during the service design process, during the architecture process, and during key design reviews and development lifecycle checkpoints. Finally, the nuts and bolts of SOA governance revolve around enforceable policies. Who defines policies, and how will these policies be enforced and results reported on such that the SOA vision and goals can be achieved? Regularly scheduled SOA governance reviews should be planned, along with design reviews, architecture compliance reviews, conformance reviews, and the like.

Eventually, when the SOA enabling technology is fully deployed, an organization may be able to automate enforcement of policies across the full SOA lifecycle, from centralized policy definition and management to the automated enforcement from design to publishing and discovery to run-time operations. At a minimum, organizations should consider automation options when defining their SOA governance processes. The more automation that is put into place, the less intrusive governance becomes to the organization and the more likely that governance processes will be executed consistently.

SOA Governance: Three Basic Steps

SOA governance is a three-step process.

1. The overarching governance model determines high-level organization, governance processes, services ownership, budgeting, and funding issues for an SOA.
2. Policies, or specific "rules of engagement," are created for designing, building/exposing, and operating services within an SOA.

3. SOA governance is implemented and integrated. Often this requires multiple solutions working together to enforce policies across the many processes of an SOA. The integration of services management, messaging platforms, service registries, metadata repositories, development tools, and security solutions must be considered to achieve SOA governance across the SOA lifecycle.

Exhibit 7.5 depicts a generic SOA governance model in two ways: organizationally and functionally. Governance often begins with addressing the organizational aspects of SOA, such as ownership of broad categories of services, budgeting and cost allocation for services and shared enabling infrastructure, and aspects of the development lifecycle that may be impacted by SOA. Exposing and/or developing services is different in some respects from traditional software development, for example, in that additional steps are necessary before services may be consumed. The transition from the requirements-driven waterfall process of software development to a producer-distributor-consumer model requires new processes for asset management, application and enforcement of design-time and run-time standards and policies, and management processes. These may include publishing to a service registry, which would require services to be discovered prior to being invoked.

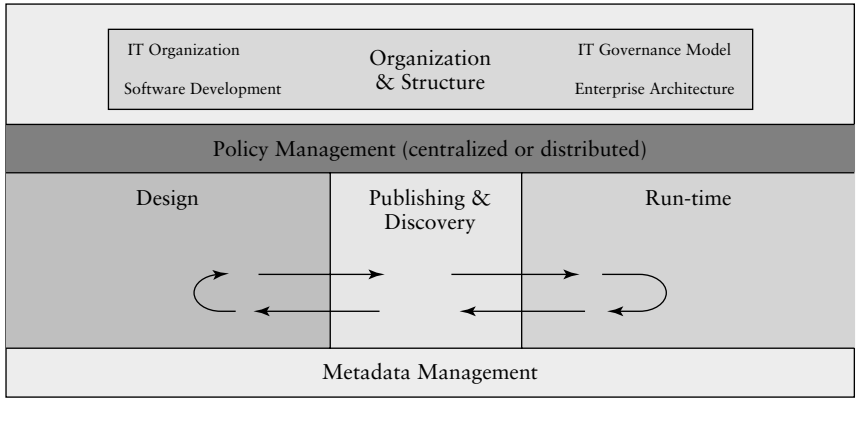

EXHIBIT 7.5 SOA Governance Model: Organizational and Functional View

In addition to organizational and procedural aspects of SOA governance, there are other lifecycle issues to be considered once an overall governance model has been devised. In order for governance to be effective, it must be built on a foundation of specific enforceable policies that will be used to encourage conformance to the goals, standards, and specifications of an organization's SOA governance model. This body of policies will be enforced at multiple points of the services lifecycle, including design time, during the publishing and release process, during the discovery process, and ultimately at run-time.

ENABLING TECHNOLOGY OF SOA GOVERNANCE

SOA governance, like SOA in general, is more than technology. Implementing SOA governance as a body of enforceable policies, or what is known as policy-driven governance, requires automation of policy enforcement as well as an integration and interoperability model across multiple platforms for governance, design, publishing/discovery, and run-time. For example, policy engines would define and manage enterprise governance and policies across the services lifecycle. Design-time governance would enforce design policies during the design, construction, and unit testing of services, as well as determining when they are allowed to be published to a registry. Design-time governance requires processes and policies, and can leverage metadata repositories to provide enforcement. Run-time governance, however, requires a different set of enabling technology in conjunction with a centralized policy engine, such as an intermediary-based architecture with agents and interceptors to enforce run-time policies as services are invoked and routed between consumers and producers. If the vision of a closed loop SOA governance model is to be realized, the governance integration and interoperability issues must be solved.

Centralized SOA Governance and Policy Engines

A new class of SOA enabling technology provides a centralized metadata catalog of enterprise policies as well as the ability to import and export policies from various run-time and design-time platforms. A

centralized policy engine allows the decoupling of policies from service design and implementation, which is critical for services version management and maintenance of services. Commercial solutions specialize in providing enterprise SOA governance and automated enforcement across the diverse portfolio of SOA enabling technology that supports design and run-time lifecycle processes. In addition, service registry vendors are extending their UDDI-based (Universal Description, Discovery, and Integration) solutions to include SOA governance and policy management capabilities, as well as adding repositories to their registry solutions.

Policy Enforcement Models

The enforcement of policies within an SOA governance architecture can be accomplished in a variety of ways. One common scenario is to use an SOA intermediary model, where an agent or Web services intermediary actively intercepts SOAP messages and then references a central policy engine to apply the appropriate policies for that service before allowing the message to be routed to its next destination. The SOA intermediary model or agent model is implemented most often in Web services management frameworks and similar run-time fabric implementations where a distributed active intermediary model is used.

In an enterprise service bus (ESB) solution, where end points are integrated by virtue of a highly distributed run-time container, the policy information is provided through configuration of the ESB through centralized administration of the solution. In this policy enforcement approach, care must be taken that policies are clearly abstracted or decoupled from the services that run over the bus. In this model, the ESB acts as a distributed run-time container. Therefore, the policies are applied by "rules" that are defined and managed centrally for the container, or ESB. However, each end point will have its own policies for services, and the ESB must be able to aggregate or know the policies for all participating end points and represent them as enforceable and decoupled policies.

In an application server model, central administration and enforcement of policies will follow a similar set of rules. In this model, policies will be centrally defined, but ensuring that they are decoupled may be

a challenge because the rules and administration of SOA policies in an application server architecture are closely related to the design process. Abstracting policies from the service design and development may pose a challenge for developers.

SOA Governance Architecture and Integration

The discussion of SOA governance always turns to the enabling technology and the mechanisms that will be used to enforce policies. Of course, SOA policies may be enforced through manual oversight processes, which were always the purview of design reviews, architecture compliance reviews, and traditional IT governance. However, given the nature of an SOA and the spatial and temporal distribution of services projects in a large enterprise, automating aspects of policy enforcement will help facilitate conformance to the SOA standards and goals that the policies represent.

Therefore, the concept of an SOA governance architecture is important. In addition to the enabling technology required to develop and operate services, which absorbs much of the attention of SOA practitioners in the early adoption phase of SOA, there is a need to ensure that the tools and technology solutions will support an SOA governance model with automated policy enforcement. For example, many organizations are exploring various SOA run-time and integration technology solutions, including ESBs, Web services management (WSM) solutions, application server suites, business process management (BPM) tools, service orchestration solutions, as well as enterprise application integration (EAI) solutions. In addition, supporting these core run-time stacks with service registries, metadata management platforms, and supplemental development tools such as XML modeling and diagnostics solutions adds to the mix. Organizations should also consider expanding services policy enforcement back into the services development lifecycle to minimize the cost of design errors by identifying them early in the development process.

The challenge, given this enabling technology confusion, is to define the SOA governance model and enforceable policies, as well as how those policies will be enforced, *prior* to selecting the enabling technology solutions. We believe that the SOA governance model

and policies should be defined in parallel with identification and appropriate modeling of an organization's services before beginning to select technology platforms. This "services-driven architecture model" helps ensure that the technology solutions will support the technical requirements of the targeted business services, which is not always the case when a technology platform is selected and then identification of appropriate services begins.

SOA governance must also be considered in a similar fashion. Identify the SOA governance model and policies that must be enforced for the targeted services, then ensure that the chosen SOA enabling technology will be able to implement automated policy enforcement, either immediately or at least in some future versions of the particular class of technology.

In all cases, seek to decouple your SOA policies from your service.

Technology and Standards of SOA Governance and Policies

SOA governance as a discipline requires technology to implement. The technology and standards of SOA governance, and in particular policy enforcement, are relatively immature. Implementing policy-driven SOA governance relies on a body of extended Web services specifications that includes:

- WS-Policy
- Web Services Policy Language
- WS-MetadataExchange
- WS-Addressing
- WS-MessageDelivery

These emerging specifications fundamentally build on the established standards for Web services such as SOAP, WSDL, UDDI, XML, and XML Schema. However, the standards for policy management and SOA governance will continue to evolve in parallel with standards and approaches to managing metadata within an SOA. Here we focus briefly on the standards relating to policies at a high level.[3]

The primary standard for defining policies is WS-Policy. WS-Policy is actually comprised of three specifications: WS-PolicyFramework, WS-PolicyAssertions, and WS-PolicyAttachment. WS-PolicyFramework is the "container" specification that includes WS-PolicyAssertions and WS-PolicyAttachment and is referred to as WS-Policy.

Policies are simply assertions about a service that allow the consumer to find, evaluate, and invoke the services according to an agreed-upon SLA. Policy assertions "inform the requester about any additional information beyond 'plain' WSDL that may be needed to successfully invoke the provider's service."[4] The provider's service publishes its policy information so that potential consumers can access it, consume and process it, and successfully invoke the service. WS-Policy is an XML grammar for expressing policies such that they can be consumed and evaluated using rules or algorithms to determine whether the SLA can be met and thus the service can be consumed. Some policy assertions will be mandatory, while others may be optional. Some policy assertions will offer choices such as "exactly one," "all," or "one or more." For example, enclosing policy assertions in these various operators will tell a consumer what policies are mandatory, whether there are choices as to one or the other policy (e.g., security options or alternate transports), or whether a group of policies must all be applied (e.g., the "All" operator).

Without digging into deep technical details, the challenge of policy-driven SOA governance is to define the specific policies that will be enforced during services consumption. The body of policies will be codified in XML using the WS-Policy specification. A potential consumer of a service requests the policy information as an XML document conforming to the WS-Policy specification, so the consumer can format the request for the WSDL that will be used to invoke the service. There are a few issues and challenges related to SOA governance.

First, there is no consensus about how to codify and enforce policy in an SOA. As mentioned, three standards specifications cover SOA policy:

1. *WS-PolicyFramework*. Developed by BEA, IBM, Microsoft, and SAP
2. *Web Services Policy Language (WSPL)*. Created by a subgroup of the OASIS XACML Technical Committee

3. *WSDL 2.0*. Includes the features and properties portion of WSDL devised by the WSDL Work Group at W3C to accommodate policy

The disputes range from which standard should prevail to questions around the inclusion of policy assertions within the WSDL documents. Policy management is a relatively immature domain, and the number of standards combined with the widespread industry buzz about SOA governance will ensure some volatility around policy for some time to come.

Another area of discussion involves whether policy assertions should be contained in the WSDL document. There has been recent discussion of the need to decouple policies from service descriptions because it is likely that an organization may apply different policies to the same service depending on who is consuming it (internal or external consumer), how it is being consumed, and by what process. Given this reality, decoupling policies from the service contract makes sense so an organization can centrally manage, modify, and update policies in an abstract fashion separate from the WSDL descriptions.

Finally, the process of evaluating policy assertions and determining which ones are mandatory versus optional is in flux. WS-Policy relies on a process whereby policies are expressed as a checklist that is matched between the provider and consumer, and numerical scores determine the relative preference for policies. If the checklist matches well enough, according to the mathematical criteria, then the service can be invoked successfully. However, WSDL relies on a scheme where policies are expressed as rules that are evaluated prior to invoking the service. The rules are evaluated as a tree structure, where the priority of the rules is established by the sequence in which they are specified.

As with the other standards of SOA and Web services, eventually the policy management standards will be resolved. In the meantime, workarounds for SOA governance are quite straightforward: Use manual policy enforcement for design-time governance, and automate policy enforcement of basic mandatory policies within the WSDL document. When the standards mature and the clear winner emerges, then the notion of decoupling policies from WSDL will most likely be realized. Decoupling policies from

services will allow the central definition, management, and enforcement of policies in a holistic SOA governance and policy enforcement model.

To summarize, metadata management requirements for SOA governance:

- Provide a management framework across the entire SOA governance process.
- Must integrate software asset metadata (design time) as well as operational metadata (run-time).
- Must incorporate a federated view of metadata, including registries and repositories.
- Must support the processes and roles across the SOA lifecycle.

SOA Governance Integration and Interoperability

SOA governance requires the federation and integration of multiple solutions in an SOA depending on how various enabling technology solutions are implemented to support a given SOA strategy. The following SOA enabling technology solutions could be part of an SOA governance architecture:

- Policy enforcement engine
- Service registry
- Metadata repository (development and run-time, which may be provided by two separate solutions: one by software asset reuse repositories and one typically provided by WSM vendors)
- Web services management solution (to provide intermediary services)
- ESB (if no WSM is installed, this will provide the intermediary services)
- SOA run-time solutions

To implement an enforceable governance model, the various pieces of SOA enabling technology must be integrated in support of a coherent governance process.

Battle for Control of SOA Governance

In light of the amount of vendor activity focused on it, SOA governance is shaping up as a dynamic SOA subdiscipline. It seems as if all SOA software vendors are claiming to deliver or manage some aspect of SOA governance. The various SOA vendors may indeed have a role to play in the implementation of policy-driven SOA governance. However, the real question is one of control. Where should SOA governance be controlled, and by what solutions?

Recent entrants into the SOA software fray have created a new approach to SOA governance based on a policy-driven model. These solutions implement an approach to SOA governance that is based on two broad requirements.

1. SOA policies should be defined and managed centrally in a policy engine that manages and enforces all SOA policies across the entire SOA lifecycle.
2. Policies must be enforced across all SOA processes, from service design, to publishing and discovery, and at run-time.

This approach, which is fundamentally the right one, creates two further SOA governance requirements:

1. SOA policies must be decoupled from the services, not embedded in the implementation of the service.
2. SOA governance must be implemented across multiple technology solutions that maintain control of those SOA lifecycle processes (e.g., service design, publishing/discovery, and runtime). This creates a potential SOA governance integration issue.

Service registries, based on the UDDI standard, are trying to assert control of SOA governance by being the primary solution for defining and managing policies in addition to managing for publishing and discovery of services. This seems somewhat reactive since UDDI has not lived up to its originally envisioned role in an SOA. Furthermore, service registries do not maintain control of the design

process or the run-time process. Thus a distributed model with a centrally defined and managed body of policies must be used to implement SOA governance.

SOA governance promises to be an interesting domain. Although there is much more to SOA governance than technology and integration, these challenges certainly will be very real over the next few years as automated enforcement of policies becomes mainstream for achieving the goals of SOA initiatives across widely distributed IT organizations and business enterprises.

Governance Summary

SOA governance is an essential ingredient for SOA success. We have shown what governance is comprised of, how policies implement an SOA governance model, and how these policies can be enforced using technology solutions. We also reviewed an approach to developing an SOA governance model and the required policies to achieve business and technology objectives. We also highlighted kinds of policies you will need in your SOA as you evolve it over time. SOA governance is critical to SOA success.

SOA BEHAVIORAL MODEL: BEYOND SOA GOVERNANCE

One area of possible change in many organizations to enable a more successful SOA initiative is the architecture process. However, the success of SOA also demands a new behavioral model for success. The behavioral model for an SOA is partially defined in the governance model, through the body of policies that will be enforced to drive conformance to the SOA standards, guidelines, and best practices. However, the behavior of an SOA also depends on structural and organizational factors, the roles and participants, and the processes that thread through the organization and roles and tie them together to achieve the stated mission and goals.

Many organizations now realize that the success of their SOA will demand the formulation of an SOA governance model and a body of enforceable SOA policies that will guide the desired management, architectural, and developer behavior within the context of the

SOA initiative. But attaining the desired SOA behavior from all its participants demands more than an SOA governance model.

SOA governance establishes the overall behavioral model of the SOA as it relates to the current IT organization, behaviors and skills, and culture of the organization. SOA governance is more than just the business and technical policies that define accepted development and run-time standards and procedures for services. It also guides the expected behavior of management, architects, developers, service consumers and providers, as well as IT management regarding the overall success of the SOA in achieving its defined objectives.

SOA governance specifies and enforces conformance to SOA policies, which define the overall behavior pattern of the participants of an SOA, such as architects, developers, services, service consumers, service providers, and others. As a recent WebLayers whitepaper notes, "Policies are the cornerstone of Governance. Policies set goals by which you direct and measure [SOA] success. Without policies there is no Governance."[5]

SOA governance is a major determinant of the organizational, technical, and behavioral success of an SOA. Governance is so essential that it must be built into the SOA planning and deployment from day one. In an SOA, the services are the lasting assets. Designing and implementing a portfolio of services in an SOA that are reusable, interoperable, and meet the needs of the business is fundamentally a behavioral problem. The necessary SOA behavior favors reuse over custom software development. The desired SOA behavior favors the SOA greater good over the needs of individuals, departments, and business units. The desired SOA behavior favors conformance to SOA policies such that interoperable reusable services can be achieved, which enables the additional SOA benefits of service and process orchestration, time to market, and increased business agility. SOA is a lifestyle change. It begins and ends with behavior and culture.

Role of Culture and Behavior in an SOA

How does behavior and culture affect the relative success of an SOA initiative? What are the moving parts of the behavioral and cultural machine that can be leveraged to positively influence behaviors toward a "services" behavior pattern? Behavior and cultural issues

are major determinants of SOA success, because SOA is ultimately a composite behavior pattern that emphasizes multiple SOA themes, such as:

- Values reuse of services over developing new services
- Values reuse of components and other IT assets
- Requires conformance to SOA guidelines, principles and standards, and overall policies
- Achieving IT productivity through reuse
- Reusing fundamental services available within the SOA to develop business solutions faster, cheaper, and better
- Achieving faster time to market for IT services to the business

These behaviors all derive from the firm's organizational and cultural fabric. These behaviors have to be defined, agreed on, and enforced in order to achieve the benefits of SOA. The role of behavior in an SOA initiative is often overlooked because it is a very challenging aspect of SOA to solve. The organizational, process, and behavioral issues are among the most difficult to manage in an SOA.

Exhibit 7.6 depicts a high-level behavioral model that brings together the aspects of an SOA that relate to cultural and behavioral forces. Ultimately, the behaviors that will help make an SOA succeed derive from the current corporate culture and must be reinforced, modified, or completely reprogrammed. Changing organizational behavior is a challenging process.

It is important to recognize the impact of organizational factors on behavior and performance of an organization. Chapter 8 discusses the impact of SOA on enterprise architecture and suggests approaches to tuning the process of architecture to the needs of an SOA.

Many factors influence the behaviors related to SOA success. The major influences are:

- SOA vision, goals, and guidelines
- SOA governance model
- SOA metrics
- SOA organization and structure (vis-à-vis existing IT and business structures)

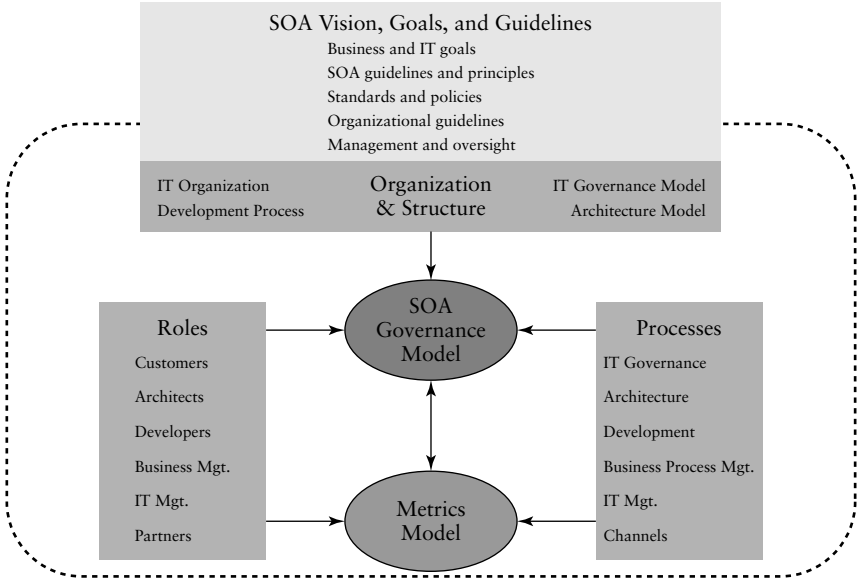

EXHIBIT 7.6 SOA Governance and Metrics Influence SOA Behavior

- SOA processes
- SOA roles and participants
- Input behaviors and the emergent behaviors
- Corporate culture and organizational behavior

SOA governance is a major contributor to the SOA behavioral model we seek. But SOA governance is not enough. Governance and its body of SOA policies require metrics and other social reinforcement mechanisms in order to drive the organizational behavior toward the norms and expectations of an SOA. Archieving the desired SOA behavior requires an understanding of the behavioral interactions in an SOA and how SOA governance, metrics, and other behavioral reinforcement mechanisms interoperate in their own right to achieve SOA success.

The behavioral interaction model of an SOA melds together the governance model, metrics, organization, processes, and roles of the SOA into a cohesive entity that can achieve the stated SOA goals. Let's explore the elements of an SOA behavioral interaction model

further. This model is comprised of four major entities with two connecting subentities. The major entities are:

1. SOA vision, goals, and organizing principles
2. IT organization and structures
3. SOA/IT processes
4. SOA roles and participants

These four major entities are connected by two crucial subentities: (1) the SOA governance model and (2) the SOA metrics model. The SOA governance model and metrics model act to bind the other behavioral elements into a body of desired SOA behaviors, norms, and cultural expectations. The total model creates an SOA behavioral interaction model, which defines the expectations for the collective behavior of the SOA overall. These quotes are instructive regarding the importance of culture and behavior in an SOA:[6]

> *"Your current IT architecture is a behavioral artifact that resulted from patterns of organizational behavior over time, driven by corporate strategy and business goals."*
>
> *"The only way to achieve SOA is to address the cultural and behavioral issues first, then architect toward your SOA goals."*

Governance and Metrics Influence SOA Behaviors

You may be asking yourself what makes this behavioral model work. The answer is in the interaction of two mechanisms: the policies of an SOA, which are defined in the SOA governance model, and the SOA metrics, which provide the performance monitoring of elements of the SOA, including behavior of services, enabling technology, consumers and providers, and the human participants.

SOA metrics are critical. You need SOA metrics to know where you are and where you are going with your SOA initiatives. In other words, SOA metrics put a steering wheel on your SOA. Very often metrics are the afterthought of SOA initiatives because much of the early focus is on getting the technology implemented and working, then measuring the results later. We believe that metrics must be built

into the SOA planning process, up front, and then assiduously monitored to help ensure goals are met.

The interplay of SOA governance and SOA metrics is how the total behavior of the SOA is determined and managed. For example, as discussed, SOA governance accommodates metrics for:

- SLAs
- Conformance reporting and policy breaches
- Enforcing reuse of existing services versus novel development of new services
- Enforcing "good reuse" versus "bad reuse," or reusing published proven services and not reusing rogue services
- Enforcement of service design best practices enterprise-wide as opposed to one-time design principles

The list could go on and on. The point is that from the body of policies in the SOA governance model, as well as the metrics defined during the SOA planning process, the overall target state behavior for SOA participants will be determined. These target behaviors must be supported by a combination of business metrics, process metrics, performance and SLA metrics, conformance metrics, and reuse metrics in order to really monitor and evolve the behaviors of an SOA.

Managing Individual SOA Behavior: Big Carrot, Big Stick

How are individual behaviors governed within the context of an SOA? Governing behavior requires a combination of clear metrics of the SOA, as discussed, and a means to relate overall SOA metrics to individual and group goals. All of these metrics and goals should be related and reinforce one another. For individual behavior, these approaches should be considered:

- Document SOA performance and behavioral expectations in annual plans for employees and contractors.
- Implement SOA performance and behavioral elements into employee review processes.

- Implement an SOA review process that helps reinforce the expectations and objectives of the SOA overall as well as the roles of various departments and individuals within the SOA context.
- Build SOA behavioral reinforcement into employee incentives and compensation plans. Consider a "profit" sharing approach for costs saved from SOA reuse and other hard-dollar and soft-dollar business benefits of SOA.

Influencing SOA behavior is going to require embedding enforcement of SOA policies and metrics within all employee annual plans and reviews as well as in compensation and reward systems.

Service-Oriented Culture and SOA

What is a service-oriented culture? In a service-oriented culture, SOA becomes the lifeblood of the IT organization. This is achieved after the organizational behavior model is implemented and there is a thorough understanding of the importance of SOA within the organization. A service-oriented culture is replicated by corporate tradition and reinforced behaviors through time. Like human culture, service-oriented culture is transmitted through learning and behavioral reinforcement.

Service-oriented culture binds the firm's vision, strategy, and objectives with its SOA strategy, vision, and governance model. We believe that our SOA behavior model describes the necessary interplay of SOA governance and SOA metrics to influence the overall behavior of the SOA, including all processes and participants. There must be ongoing training and reinforcement of the SOA goals, mission, metrics, and behavior in order to truly achieve a service-oriented culture. This is what leads to SOA results.

In order to help ensure SOA success, organizations must spend time understanding and planning for a behavioral model that will enable SOA success. Remember change management as a discipline that accompanied business process reengineering projects? At least change management was an explicit attempt to model behaviors that would help instill the process changes that attended BPR initiatives in the 1990s. What we need for SOA success is a new model, a behavioral model that begins with behavior and factors in the organizational,

process, and behavioral elements that will result in a successful SOA. We have to begin with the behavior of SOA—the behaviors that lead to services reuse, SOA conformance, governance, and metrics—that will lead you to your SOA business goals.

SUMMARY

Governance is critical to the success of an SOA. We have discussed the overall requirements of SOA governance, including the elements of SOA governance, the organizational and process requirements, and the overall approach to SOA governance. Ultimately, SOA governance enforces an organizational behavior and cultural model. The interplay between SOA governance and a metrics model will determine the effectiveness of SOA governance and the overall culture and behavior that will determine SOA success. While we spend only a few pages on the cultural and behavioral challenges of SOA, in reality the effort will be the opposite. The organizational dynamics and behavioral aspects of SOA will require far more effort than the technology. The effort, however, will be worthwhile.

NOTES

1. Eric A. Marks, *Business Darwinism: Evolve or Dissolve* (Hoboken, NJ: John Wiley & Sons, 2003).
2. WebLayers, Inc., *SOA Governance Introduction* (Cambridge, MA: WebLayers, 2005), p. 11.
3. For a detailed discussion of the metadata management requirements for SOA, see Eric Newcomber and Greg Lomow, *Understanding SOA with Web Services* (New York: Addison-Wesley, 2005).
4. Ibid., p. 298[0].
5. WebLayers, *SOA Governance Introduction*, p. 9.
6. Eric A. Marks, *SOA Governance Overview* (AgilePath Corporation, 2005).