



The cornerstones of SOA governance: policies, registries and repositories.

*Steve Graham,
senior technical staff member, IBM SOA Foundation*

*Guy Loewy,
chief technical officer, WebLayers*

*Sunil K. Murthy,
product manager, IBM WebSphere Service Registry and
Repository software*

*Dan Potter,
vice president of marketing, WebLayers*

Contents

2	<i>Introduction</i>
3	<i>Scenario</i>
7	<i>SOA governance</i>
8	<i>Overview of IBM WebSphere Service Registry and Repository software</i>
12	<i>Overview of WebLayers Center software</i>
15	<i>A complete solution for SOA governance</i>
30	<i>Summary</i>

Introduction

In this paper, we will describe two cornerstones of a service-oriented architecture (SOA) governance solution: policy infrastructure and an integrated service registry and repository capability. We will briefly describe what SOA governance is and why SOA governance is critical to the overall success of an organization's adoption of service orientation. We will also outline some typical problems related to SOA governance, and how the policy infrastructure and service registry and repository capability can be combined to address those problems.

We begin by introducing you to Jane Dunham, an SOA architect working for a fictional insurance company. We're going to explore the SOA governance-related challenges she is facing, and then briefly define SOA governance in terms of IT governance and overall governance. Following this, we will briefly describe the features and capabilities of the IBM WebSphere® Service Registry and Repository software and the WebLayers Center solution. We will tie things together by examining how the combination of the WebSphere Service Registry and Repository and WebLayers Center products address the SOA governance challenges faced by Jane. We will conclude the paper with a review of the benefits that SOA governance can bring to organizations adopting service orientation to achieve better alignment between the needs of their businesses and the capabilities of their IT investments.

Scenario

As she put down her coffee mug, Jane Dunham reminisced about how easy life was in early 2004 when her company, a large insurance provider, piloted its first Web services-based claims-processing application. Things seemed so simple: Use the new Simple Object Access Protocol (SOAP) engine inside the IBM WebSphere Application Server, Version 5.0.2 software to front-end Enterprise JavaBeans (EJBs) technology connecting to a suite of IBM Customer Information Control System (IBM CICS®) applications to generate some Web Service Definition Language (WSDL) and voilà!—integration challenges would be a thing of the past.

But that was then. In the intervening time, Jane and her firm have learned a lot. Jane has evolved from programmer to the firm's first SOA architect, the person who understands the technical needs of agile application integration as well as the business concerns of how to align the capabilities of the IT portfolio with the needs of the business. The SOA architect also needs to establish flexible business processes that can be easily and quickly reconfigured to adjust to new technical and business realities.

One thing Jane has learned is that point-to-point Web services, the kind where the Web service and the consuming application are designed at the same time, are easy to create. But, when the needs of the business change, those point-to-point Web services can quickly become outdated. Worse, because these sorts of Web services were so easy to build, her firm had an exuberant explosion of Web services creation, and now there is a Wild West of services in various states of quality in use throughout the firm.

Within less than three years, the firm has developed so many Web services with a confused history of ownership and support that no one is sure which applications are using which Web services. In fact, recently Jane heard the story of a peer in another business unit who nearly lost his job because an application he was responsible for went down for three days. The outage, which cost the organization several million dollars in lost business, was caused by a change in a Web service no one had ever heard of. No one was tracking the dependence between this critical application and the service that had been quickly put together under the radar.

SOA is great, but unless it's controlled, it's just another chaotic IT process that fails to meet business needs. For this reason, Jane's firm – with the help of IBM Global Business Services – put into place a Center of Excellence for SOA to act as a focal point to review newly created Web services; to communicate and enforce standards related to Web services design and deployment; and to encourage a higher degree of service reuse.

Things have been out of hand for some time, but Jane and her team had been holding things together through lots of coffee and many late evenings. Jane's boss, Jerry Hall, director of IT architecture and development, appreciated all the hard work and had recently conferred with Jane about the current mode of operation – the Web services Wild West was not sustainable and could not support the need for the department to provide reliable, predictable applications to internal business clients.

Jane thinks that she and Jerry finally understand the difference between Web services and SOA. It is easy to build Web services; it takes discipline to deliver serious business value from them, and that is where SOA comes in. The spreadsheet approach to tracking Web services and all their dependencies is no longer sufficient; too many other folks inside IT and in the business units have been asking for more details about the various services. And keeping the spreadsheet up to date is a dull and error-prone task. Maybe the notion of a registry or repository for SOA is exactly what Jane needs. Otherwise, how can her department encourage reuse if no one can find information about what services might be available to reuse!

Alan Peckman, a senior manager in the auto insurance division, had approached Jerry to discuss some changes his department needed in its Web service-enabled claims fulfillment process. Not only did various claims-processing services need to be consumed by Microsoft® .NET clients as well as by the current Java™ code-based client, but – according to Alan, for time-to-market reasons – several Business Process Execution Language (BPEL)-based processes would need to be tested, and updated if necessary, to integrate with a third-party service to locate preferred auto repair partners.

As Jerry explained this new situation to Jane over lunch earlier today, she realized that the era of simple, point-to-point Web services was over. A whole new world of more flexible relationships between service provider and service consumer was facing her team. This, together with the known problem of better understanding the portfolio of Web services and the applications that depend on them, was critical. Of course, several organizations have been pressing for the creation of more services and changes to many existing

services. And, to add more urgency to the situation, the newly formed Center of Excellence for SOA was pressing Jane's organization to enforce a new set of IT standards related to Web services and to drive more reuse of services.

So, with the last sip of coffee, Jane pondered how to prioritize and execute a solution in the face of sometimes conflicting needs:

- *How can we deliver more services to the business more quickly?*
- *How can we prioritize the services that need to be built?*
- *How can we increase reuse so that costs and the time required to get applications running can be reduced?*
- *How can we get a handle on the quality of services being deployed and which services are easy to reuse? And how can we identify and improve those hard-to-integrate services?*
- *How can we identify best practices and patterns for better services? How can best practices be enforced? Which services already follow those best practices? How can we ensure that good Web services are not changed to deviate from these best practices?*
- *How can we get a better handle on the entire inventory of services, the parts of the infrastructure they are dependent upon, and those other services and composite applications that depend on them?*

Jane knows that without answers to these questions, the firm's ability to adopt SOA to deliver more flexible IT implementations of business processes may be in jeopardy. Jane recalls a recent article that described the discipline of SOA governance. Back in the early days of Web services, all the talk about registries and governance didn't make sense to her. Maybe now things will. Jane resolves that it is time to spend a little more energy finding out how SOA governance might be able to help her.

SOA governance

A definition for governance can be found on the Wikipedia Web site.¹

Governance involves:

- *Establishing chains of responsibility, authority and communication to empower people (decision rights).*
- *Establishing measurement, policy, standards and control mechanisms to enable people to carry out their roles and responsibilities.*

The article further expands on this definition to describe IT governance in terms of “that subset of corporate governance that pertains to an organization’s IT processes and the way those processes support the goals of the organization.”

SOA governance is a specialization of IT governance that focuses on the lifecycle of services and composite applications in an organization’s SOA. Deploying an SOA frequently serves as a catalyst for an organization to start thinking about improved corporate and IT governance, and how to best implement SOA governance practices.

According to a recent article,² SOA governance can enable organizations to maximize the business benefits of SOA, which can include increased process flexibility, improved responsiveness and reduced IT maintenance costs. It also mitigates many of the business risks inherent in SOA adoption by establishing decision rights, guiding the definition of appropriate services, managing assets and measuring effectiveness.

To achieve SOA governance, Jane begins to understand the need to address these key capabilities:

- *The importance of understanding the end-to-end lifecycle of various types of artifacts related to services and composite applications, including XML artifacts such as WSDLs and XML Schemas, as well as other artifacts such as project documents created in the Microsoft Word or Microsoft Excel application*
- *The set of decision rights, policies, standards, controls and key measurements associated with the lifecycle of services and composite applications*
- *The mechanism for enforcing decision rights, policy enforcement and processes surrounding the end-to-end lifecycle of SOA artifacts*

The following sections provide a brief overview of two of imperative elements that Jane identifies in her mandate to achieve SOA governance: IBM WebSphere Service Registry and Repository and WebLayers Center software.

Overview of IBM WebSphere Service Registry and Repository software

IBM WebSphere Service Registry and Repository software is a key enabler for SOA governance featuring an optimized service registry and integrated service metadata repository to govern the service lifecycle. The seamless publish and find capabilities across all phases of SOA foster reuse of services and enrich connectivity with dynamic and efficient interactions between services at run time. The robust registry and repository functions for the entire software lifecycle and integrated component with the IBM SOA Foundation³ make the WebSphere Service Registry and Repository software an essential tool for the success of SOA. Regardless of the level of SOA maturity in your enterprise, the WebSphere Service Registry and Repository software can bring immediate benefits that will accelerate your firm's adoption of SOA.

To address customers' SOA needs, the WebSphere Service Registry and Repository software has been developed using a highly interactive iterative development approach. This approach has helped to suitably adopt the product capabilities for prime SOA usage.

Service metadata

Service metadata artifacts can be stored in WebSphere Service Registry and Repository software, enabling broader visibility and improved reuse, management and governance of services. These artifacts are essentially documents containing descriptive information about and related to services. WebSphere Service Registry and Repository software supports key document types such as XML, WSDL, XML Schema Definition (XSD), Web Services Policy (WS-Policy) and Service Component Description Language (SCDL). The information from these standard document types is organized into meaningful groups to facilitate easy browsing of service information. The information is also organized into a set of logical objects derived from the documents. This allows the SOA run times, such as enterprise service bus or process servers, to access the service metadata in an optimal fashion.

For the key document types, WebSphere Service Registry and Repository software also understands predefined properties and makes an effort to detect service relationships based on linkages to other documents. These relationships are recorded in the content model. Other types of service metadata can be stored using a generic content type such as an XML document.

User-defined metadata

WebSphere Service Registry and Repository software supports a number of user-defined metadata types that are used to decorate the service metadata to explain their semantics; we refer to those metadata as service description metadata.

WebSphere Service Registry and Repository software supports three types of service description metadata: properties, relationships and classifiers. All three can be used to decorate entities in the physical or logical model. This enables semantic queries to target individual elements of the service metadata, and it allows meaningful dependency analyses to take place prior to making changes. User-defined category systems are imported and shared through the use of documents encoded using the Web Ontology Language (OWL).

User-defined properties and relationships can be used to customize the set of predefined properties and relationships provided in the WebSphere Service Registry and Repository metamodel. Users can add properties to a WSDL document, or they can configure logical model elements with properties and relationships to represent the document's structure. To simplify the process of customizing WebSphere Service Registry and Repository entities, a simple template mechanism is supported that can be used for definition of properties and relationships.

The WebSphere Service Registry and Repository software supports a fine-grained access control model that allows users to define which user roles can perform what kind of actions on which artifacts. It allows users to define and import classifier systems from simple classifier sets to taxonomies and classification hierarchies. It also provides interfaces to analyze the impact of changes to WebSphere Service Registry and Repository content, and it can audit such changes.

Governance

WebSphere Service Registry and Repository software governs services by promoting visibility and consistency, and by reducing redundancy in SOA. By promoting consistent usage of services across SOA and managing the service lifecycle, the WebSphere Service Registry and Repository software provides

business flexibility and control, driving IT alignment to business needs. Rich governance functions include a lifecycle model for governed service meta-data elements using a state machine that describes these states, the valid transitions between them, the plug-in validators to guard the transition and the (notification) actions to be taken as result of the transition. Supported SOA-focused service lifecycle phases include service inception, service reuse, service versioning, service retirement, service promotion, service availability, service funded and more.

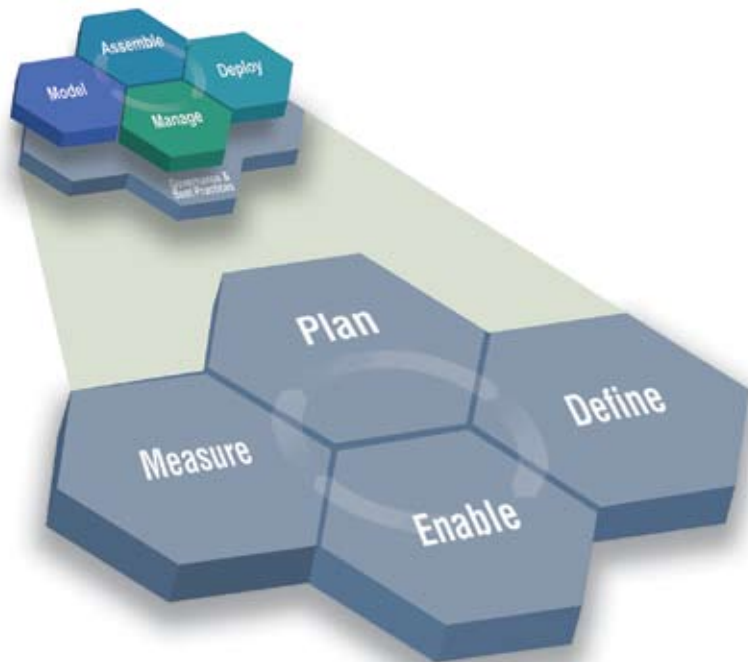


Figure 1. The WebSphere Service Registry and Repository software plays a key role in the end-to-end governance of the SOA foundation lifecycle. It is used in the enable phase of the governance lifecycle where policies and processes are actually implemented, and in the measure phase where governance requirements are monitored and measured.

As the SOA landscape evolves with needs focusing on service lifecycle management and governance, the WebSphere Service Registry and Repository software is geared to meet the need to federate service metadata with Reusable Asset Specification (RAS) repositories, Composite Application Management and Configuration Management Databases, as well as the metadata management solutions.

Overview of WebLayers Center software

WebLayers Center software is an enterprise policy infrastructure for SOA governance solution. The WebLayers Center product enables enterprises to define, enforce and audit policies across all of the existing systems to support the SOA lifecycle from design to deployment.

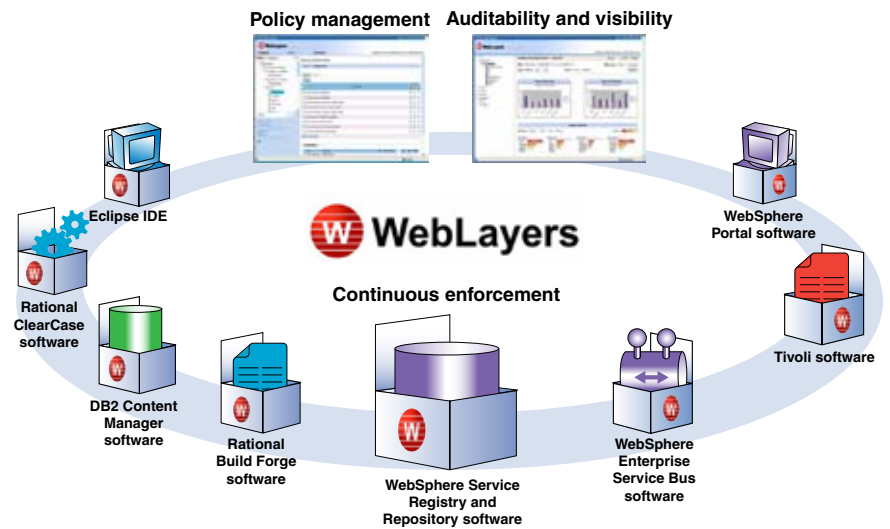


Figure 2. The WebLayers Center solution enables enterprises to define, enforce and audit policies across an enterprise's systems.

Best practices policy library

The WebLayers SOA policy library jump-starts the SOA learning process that is critical to any enterprise. The library contains a set of best practices derived from working with Fortune 500 companies on their SOA implementation projects, as well as industry consortia recommendations and standards defined by the World Wide Web Consortium (W3C) and Web Services Interoperability Organization (WS-I). The policy libraries have been designed to support each phase of the lifecycle.

Policy definition and management

The WebLayers Center solution has a complete policy infrastructure that enables the creation, configuration and management of policies. Policies can be technical, architectural or business in nature, and their scopes can span within and across corporate, divisional or departmental levels, supporting multiple stakeholders. This allows policymakers to establish the appropriate policies for the entire enterprise and for each line of business.

Enforcement across the lifecycle

Noninvasive enforcement agents, called WebLayers Governors, transparently examine content and enforce compliance across development processes and the various infrastructure components—including development tools, registries and repositories—to drive policy adherence across the entire lifecycle without changing existing lifecycle processes or creating impediments to the governance process. By providing continuous enforcement across the entire lifecycle, the WebLayers solution provides the visibility to identify and resolve issues as they occur, rather than just governing at specific transition points of a service lifecycle. Business and technical policies can be enforced in the design and development phases to provide the necessary guidance to get the service right from day one.

Throughout the SOA lifecycle, there are various infrastructure components generating unique document types, artifacts and supporting code. The WebLayers Center solution provides prebuilt policies and governance for all of these SOA layers, including the following:

- *Documents (e.g., Microsoft Word, Microsoft Excel, Text)*
 - *Inspect the requirements documents from the business group to ensure appropriate information and approvals are complete.*
- *User interface artifacts (e.g., HTML)*
 - *Ensure that the HTML complies to internal user interface standards and federal accessibility requirements.*
- *Messaging artifacts (e.g., XML, SOAP)*
 - *Verify compliance with security and regulatory requirements.*
- *Service description (e.g., WSDL, XML Schema)*
 - *Ensure that services can be easily reused by different types of clients.*
- *Code artifacts (e.g., Java)*
 - *Ensure that service implementations follow architectural standards and best practices.*

Detailed compliance auditing and reporting

The WebLayers Center management console enables business and IT stakeholders to have real-time visibility into their enterprise SOA deployment and operational status. Management dashboards include a comprehensive set of drill-down reports to provide statistics and impact analysis in both summary and detailed perspectives. This information is critical to an organization implementing SOA in determining the effectiveness of people, services and business initiatives in relation to policies and their impact points.

A complete solution for SOA governance

IBM and WebLayers have teamed to provide the industry's most comprehensive solution for SOA governance. At the core of the solution is a tight integration between the WebLayers Center solution, a policy-driven infrastructure, and IBM WebSphere Service Registry and Repository software. This combination of products and services from the IBM SOA governance consulting practice delivers a holistic SOA governance solution to enterprises.

Let's explore how this combination solves the problem Jane and her team are facing.

Define policies and stakeholders

Jane's initial step is to begin to define the policies for her company. This is often done in conjunction with a Center of Excellence for SOA to help communicate and establish buy-in for the policies and other SOA-related decisions. Policies comprise both technical and business requirements, and help create a common language of information and process. Jane has a variety of policies that need to be defined at different levels.

Jane's policies start at the business level with regulatory compliance issues:

- *Policyholder name and contact information may not be transmitted as clear text.*
- *Each message must carry information to uniquely identify its source, destination, timestamp and transaction ID for mandatory archiving requirements.*

Then Jane establishes specific policies for information security:

- *Messages must contain an authorization token.*
- *Password elements must be at least six characters long and contain numbers and letters.*
- *Every operation message must be uniquely identified and digitally signed.*

Finally, Jane develops detailed technical policies that ensure architectural strength:

- *Do not use Remote Procedure Call (RPC)-encoded Web service operations.*
- *Do not use a solicit-response style of operations.*
- *Do not use XML “anyAttribute” wildcards.*

The WebLayers SOA policy libraries allow Jane to get started quickly. Jane can quickly select from a starter set of policies that are appropriate for her organization, tailoring them as she chooses.

Using the WebLayers Director module, the policy management component of the WebLayers Center solution, Jane can quickly and easily write new policies. And using a wizard-driven interface, she can add details such as the policy source and include code examples in the policy explanation. She can also identify the specific business impact that will result when artifacts are not in compliance.

Furthermore, policy conformance and validation of artifacts can be fully automated by defining and implementing the conditions for policies using industry-standard dialects.

Figure 3 illustrates the policy authoring interface. Policymakers can create collections of policies known as libraries. The libraries are classified so they apply to specific SOA artifact types via a configurable classification system that can examine artifact content and properties.

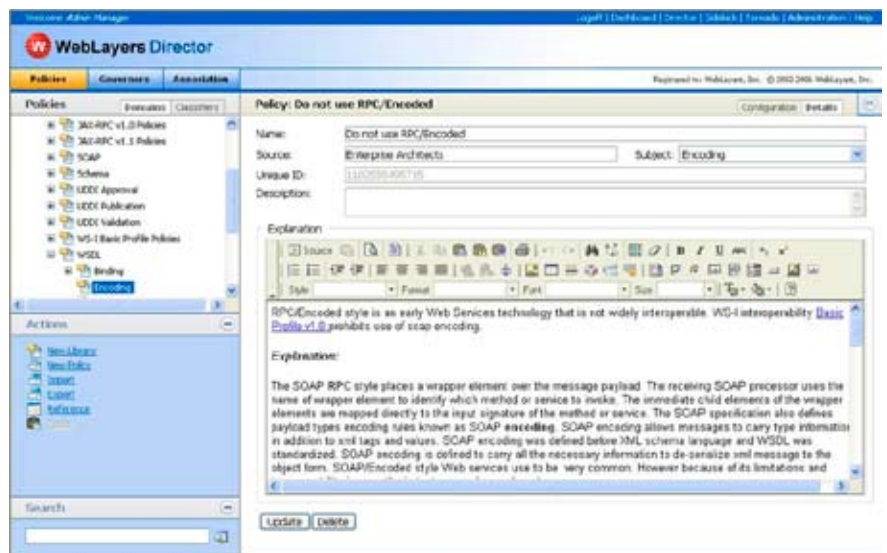


Figure 3. The WebLayers Director module enables you to author policies.

Policies can be federated by project, business unit or across the entire enterprise. These federation levels are known as policy domains. If policy conflicts are detected, configurable conflict-resolution rules specify which domains take precedence. While Jane can create company-wide policies and enforce them across all enterprise divisions, the life insurance division can also manage its own specific policies in addition to the inherited enterprise policies.

When enterprise-level policies conflict with one or more suborganizational policies, the WebLayers Center solution allows policy offenders to request exclusion or an override from the stakeholder of enterprise policies. Stakeholders can choose to grant or reject policy exception requests. The entire policy exception process is tracked by the WebLayers Center solution and streamlined by the policy management console.

The WebLayers Center solution uses its event mechanism to provide full synchronization of policies into the WebSphere Service Registry and Repository software. Because the WebSphere software can act as the single system of record for all service metadata artifacts within the organization, policy additions, deletions or modifications with the WebLayers Center

solution can be automatically synchronized to update the appropriate WS-Policy information stored in the WebSphere Service Registry and Repository software. Figure 4 shows a WebLayers policy artifact within the WebSphere software.

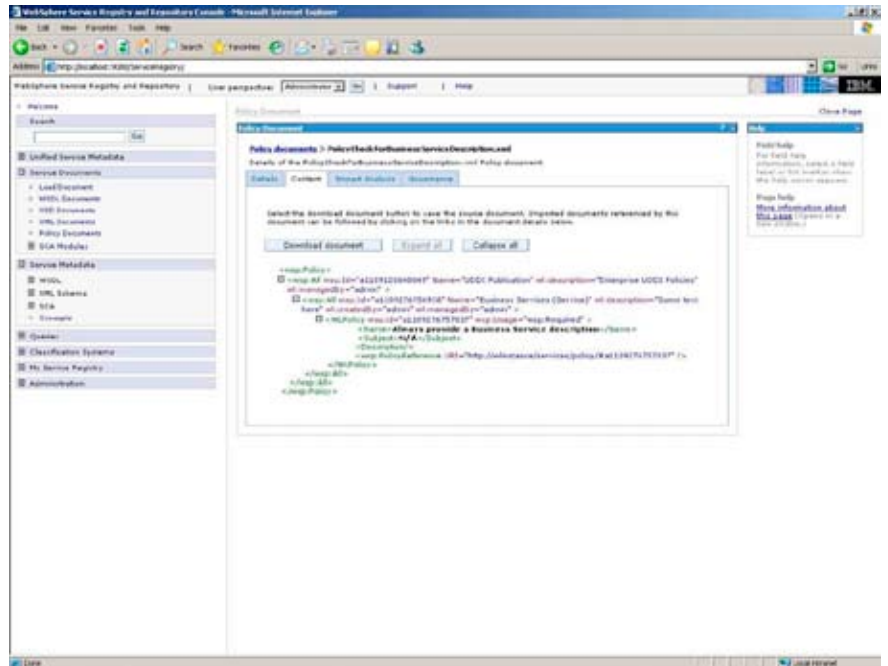


Figure 4. The WebSphere Service Registry and Repository software records all service metadata artifacts, and the WebLayers Center solution automatically synchronizes policy information.

Establishing a compliance baseline

Jane can assess the business impact of introducing new or changing policies against the current artifacts and services already published to the WebSphere Service Registry and Repository software. The WebLayers Center solution provides an impact simulation to analyze compliance results against a new candidate set of policies without interfering with the production WebSphere environment.

The WebLayers solution automatically navigates through the WebSphere Service Registry and Repository software, checking the compliance of relevant artifacts, publishing metadata regarding the compliance results and business impact, and dynamically adding association between each artifact and its original location within the WebSphere software. This provides Jane with a baseline compliance level; it shows her the quality of existing assets; and it highlights the areas that need review. For the first time, Jane has a set of reports and dashboard views that she can share with her managers and peers to provide visibility on the current state of their SOA.

Jane is now armed with information about where to start, which policies have the highest impact on the business and which are the most out of compliance. Jane can focus on a campaign for improving services that do not comply with the most critical policies and have the highest impact on the business. She can incrementally bring these existing services into conformance.

While most of Jane's initial efforts are focused on internal-facing projects, soon after deploying the system, she will realize the benefit of navigating partner registries and repositories to verify compliance and foresee interoperability or other policy challenges that may rise. She will follow the same procedure and establish a compliance baseline with her firm's partners.

Continuous governance

Once the compliance baseline is completed, Jane will enforce the policies against any new artifacts published into the WebSphere Service Registry and Repository software to ensure continuous governance over new services and changes to existing services. For example, Jane can implement a WebLayers Governor to block users from publishing services to the WebSphere software that violate certain security policies or that may introduce liability to the enterprise.

Additionally, Jane wants to ensure continuous compliance of artifacts that are published to the WebSphere Service Registry and Repository software through its Web services application programming interface (API), as well as start to establish control over the evolution of new implementations. To achieve this, Jane will implement the WebLayers Filter Governor module that intercepts any communication going to or coming from the WebSphere software, including publication requests and lifecycle transition requests. The WebLayers Center solution checks compliance of published artifacts and populates metadata regarding compliance results and business impact while dynamically adding association information. As part of the policy enforcement, Jane can configure the criteria of publication acceptance and rejection. In certain cases, she can allow publication even though the artifact is not compliant, and she can mark through metadata that this artifact should not be exposed as production quality.

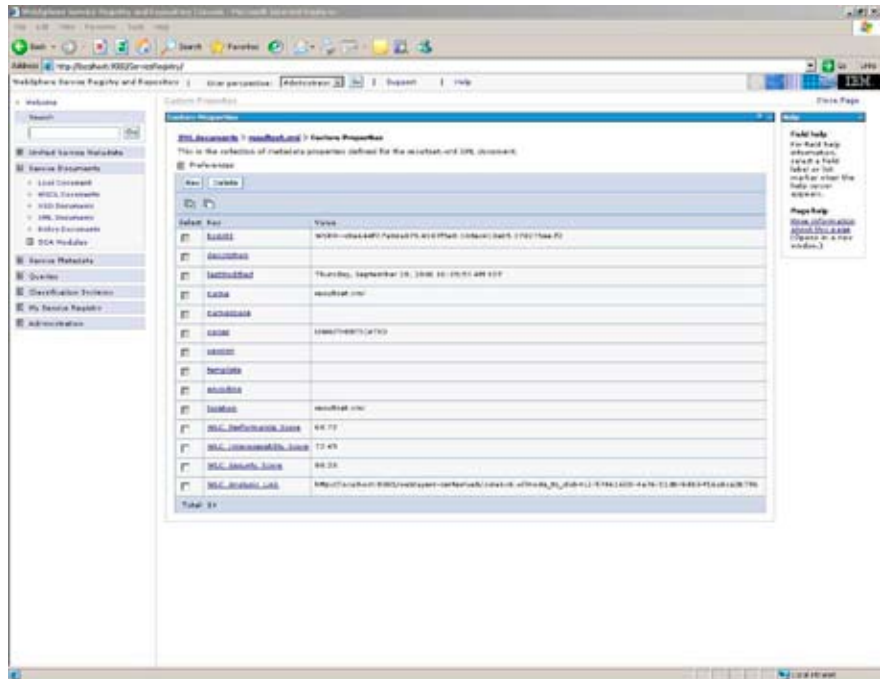


Figure 5. Impact scores for performance, interoperability and security for a service are created by the WebLayers Center solution and stored in the WebSphere Service Registry and Repository software.

While the above deployment guarantees that any artifact published to the WebSphere Service Registry and Repository software is tested for compliance, Jane will want to add and/or change policies over time. With these additions or changes, artifacts that were compliant in the past may become noncompliant. Jane schedules analysis of the WebLayers Center solution on a weekly basis to ensure current compliance and business impact information. Compliance status can then be viewed over time as policies change.

Jane knows that there is always an exception to every rule. As artifacts are checked for compliance, the WebLayers Center solution may block certain operations, and users may want to request exception from complying with these policies. The WebLayers Center solution manages the exception request process by notifying the appropriate policy owners and taking the appropriate action, including granting exception, providing conditional authorization or rejecting. The WebLayers Center solution provides the exception information to WebSphere Service Registry and Repository software as metadata, which can be used to decide whether certain artifacts can transition to the next phase in the service lifecycle. Now Jane can focus on managing with an eye on exceptions, rather than wasting time on the same type of issues.

Governance throughout the entire SOA lifecycle

The ability to enforce policies with the WebSphere Service Registry and Repository software is an absolute necessity. However, to fully achieve SOA governance, Jane must continually enforce policies in between control points – while promoting service artifacts – as well as continually enforce policies before and after the interaction with the registry and repository.

The WebLayers Center solution provides enforcement across the IBM family of products to support all phases of the SOA lifecycle.

Examples include the following:

- *Model and assemble phases*
 - *Integrated development environments (IDEs) – including plug-ins for IBM WebSphere Studio software and other Eclipse platform-based IDEs*
 - *Source Code Control – including IBM Rational® ClearCase® software, Concurrent Versions System (CVS) software*
 - *Registries and repositories – including IBM WebSphere Service Registry and Repository, Universal Description, Discovery and Integration (UDDI)*

- *Deploy phase*
 - *Registries and repositories*
 - *Messaging systems—including IBM WebSphere Enterprise Service Bus software*
 - *Application servers—including IBM WebSphere Application Server software, JBoss products*
 - *Databases—including IBM DB2® software*
- *Manage phase*
 - *Process integration—including IBM WebSphere Business Integration software, etc.*
 - *Systems management—including IBM Tivoli® software, etc.*
 - *Business intelligence and reporting tools—including Alphablox® software*

Delivering value in the model and assemble phases

Engineers should see SOA governance as a benefit, not an obstacle. The combination of WebSphere Service Registry and Repository software and the WebLayers Center solution delivers value in the initial phases of the SOA foundation lifecycle by providing:

- *A central registry and policy system to discover existing artifacts and understand existing policies and which policies apply to the current project.*
- *A repository for capturing the knowledge and intellectual property being created throughout the enterprise.*
- *Best practices libraries to help design better applications.*
- *Enforcement of compliance to certain policies by blocking commits to Rational ClearCase software and notifying errors on nightly builds. This way, Jane can verify that specified design patterns are not being used, that certain security and performance issues are not being introduced by new or modified code, and that configuration and deployment are going as smoothly as possible.*

Delivering value in the deploy stage

Like many other aspects of SOA, policies are used by many different components within the infrastructure. One of the challenges Jane faces is how to manage and reuse policies across the enterprise. The WebSphere Service Registry and Repository software provides the perfect central location for storing and managing metadata around policies. These policies may be coming from run-time management solutions, hardware devices, security solutions or other sources.

The WebLayers Center solution provides full synchronization of policies into the WebSphere software through its event mechanism. Any creation, deletion or update of a policy within the WebLayers solution automatically updates the WS-Policy information stored in the WebSphere software and makes it available to other solutions. This interaction also eases the integration of the WebLayers Center solution and reuse of policies from other tools.

The WebLayers product also provides additional information, or Quality Marks, from the data gathered by all the systems being governed. These Quality Marks are generated based on impact weights. When a policy is created, Jane can assign an impact weight to it to provide an understanding of the business and technical effect of policy nonconformance.

Technical impact weights include information on service performance, security, interoperability, portability and so on. For example, a policy may state that SOAP messages should not contain comments, as they can degrade performance. If the service does not conform to this policy, it would not rate high on the performance quality mark.

Business impact weights include information on cost of ownership, time to market, legal risk and so on. For example, Jane's company has an enterprise license for IBM DB2 information management software, and she has created a business policy that states that a DB2 database should be used. If a proprietary database is used instead, the service will have a negative impact on cost of ownership and maintenance, because of the additional license and personnel costs incurred using the proprietary database.

The registry and repository become the system of record for SOA assets within the enterprise. Service consumers can check with the system to locate the most appropriate service that satisfies its functional and performance needs. Because the system is governed by an active policy management system, you can be assured that all services are in compliance and have the necessary insight into service quality, performance, availability and security. This capability helps make SOA deployments more dynamic and more adaptable to changing business conditions.

Detailed SOA compliance auditing and reporting

Jane's biggest challenge is to communicate the benefits and status of the SOA initiative to IT and business managers. The ability to view business impact as well as trend information of services published to the WebSphere Service Registry and Repository software provides Jane with the tools needed to fine-tune and adjust the governance processes and policies and make sure that the governance model is modified accordingly.

The WebLayers SOA governance dashboard helps Jane keep track of how her portfolio of services complies with various policies established by her and the Center of Excellence for SOA. This helps Jane and her team understand which policies are being followed and the business impact of noncompliance, providing important feedback to the team to improve the overall governance of the company's SOA. An example dashboard is shown in Figure 6.



Figure 6. The WebLayers Dashboard helps users keep track of how services are complying with policies.

The Dashboard views provide a high-level understanding of the impact of the SOA initiative on achieving business goals such as reducing total cost of ownership, improving time to market and mitigating security risks. Navigation using a filter enables custom views on how specific organizations and projects perform over time, and allows business users to quickly focus on important issues. These views can also be integrated with any standard portal product using portlets. For the first time, Jane has reports she can share with management to convey the need for SOA and prove its value.

Benefits to the enterprise

But how does SOA governance benefit the organization? Let's take a look at the effect strong SOA governance has on various business functions.

IT management (Jerry Hall's position)

- *Verifies consistency of development of various IT components, resulting in fewer surprises and exceptions of deployed services and applications at run time. Greater consistency results in lower maintenance costs, fewer unexpected outages of key applications and lower SOA operational costs.*
- *Allows oversight of the entire portfolio of SOA artifacts, services and applications, and the relationships between them. Also provides insight into the progress of current projects against committed delivery schedules, budgets, etc.*
- *Reduces SOA project risks because the tools provide a means by which quality can be detected, visualized and audited, and provides insight on how quality deviation can be anticipated.*

Business management (Alan Peckman's position)

- *Enables faster time to market with new applications to generate more revenue and/or reduce costs.*
- *Results in fewer surprises from the IT systems when interoperating with customers' and business associates' systems, resulting in better client satisfaction.*
- *Improves communication between the business staff and the IT staff; for example, by building scorecards and dashboards on the SOA portfolio and ensuring the quality of that portfolio.*

SOA developers (Jane Dunham's position in 2004)

- *Provides a place to go to find out the "real deal" with respect to services available for reuse, allowing faster application development.*
- *Provides a level of assurance that the services being reused are conformant with key architectural principles.*
- *Ensures that developers won't be surprised when components change; they will have some control over change processes, and they will be notified.*
- *Makes it easy to stay current on the evolving standards and best practices associated with Web services and SOA standards.*
- *Provides rapid feedback on issues and guarantees higher-quality deliverables.*

SOA architects (Jane Dunham's current position)

- *Enables an ability to define best practices and to communicate and encourage their use by the development community, in conjunction with a Center of Excellence for SOA*
- *Helps to operationalize policies (not just shelfware ignored by busy developers)*
- *Allows more projects to be reached and therefore brought under better control of the business as governance becomes less onerous.*
- *Provides visibility on an enterprise level to overall compliance to policies, and provides the feedback necessary to do a better job at policy creation and modification. Impact analysis of proposed policy additions or changes can be used as a trial balloon to help govern the governors. This provides valuable support to the feedback loop between the measure phase and subsequent plan phase in the SOA governance framework.*

Asset manager (Jane Dunham's peer position)

- *Provides the ability to translate governance policies and processes into production.*
- *Oversees the correct use and management of IT assets, including the service metadata in the WebSphere Services Registry and Repository software.*
- *Drives discovery and publication of new services.*
- *Drives the definition of lifecycle models for services.*
- *Assigns classifications to service descriptions so they can be located and used.*
- *Supports analysis of upcoming changes, communicates the changes and ensures that those changes are authorized.*

Summary

In this paper, we reviewed the importance of SOA governance to the problems Jane and her organization were facing as they adopt a service-oriented approach to aligning business with information technology. We defined SOA governance and outlined the benefits that it can provide to the organization.

We focused on two cornerstones of a SOA governance infrastructure: policy management and enforcement capabilities, and service registry and repository. We examined two industry-leading products: IBM WebSphere Service Registry and Repository software and WebLayers Center software. And we reviewed the benefits Jane and her firm receive when they combine the two robust solutions.

Clearly, SOA is a strategic imperative to the organization, but if the architecture isn't accompanied by an effective governance strategy, the whole initiative is destined to fail. Take the time to explore and establish an SOA governance strategy and infrastructure that's right for your business.

For more information

To learn more about how the IBM WebSphere Service Registry and Repository software and the WebLayers Center solution can help you achieve SOA governance, visit:

ibm.com/soa

weblayers.com/ibm



© Copyright IBM Corporation
and WebLayers, Inc. 2006

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

WebLayers, Inc.
125 CambridgePark Drive
Cambridge, MA 02140
U.S.A.

Produced in the United States of America
10-06
All Rights Reserved

CICS, ClearCase, DB2, IBM, the IBM logo,
Rational, Tivoli and WebSphere are trademarks
of International Business Machines Corporation
in the United States, other countries or both.

WebLayers, WebLayers Center, and the WebLayers
logo are trademarks of WebLayers, Inc.

Alphablox is a registered trademark of Alphablox
Corporation in the United States, other countries,
or both.

Java and all Java-based trademarks are trademarks
of Sun Microsystems, Inc. in the United States, other
countries, or both.

Microsoft is a trademark of Microsoft Corporation in
the United States, other countries, or both.

Other company, product and service names may
be trademarks or service marks of others.

The information contained in this documentation
is provided for informational purposes only. While
efforts were made to verify the completeness
and accuracy of the information contained in this
documentation, it is provided "as is" without war-
ranty of any kind, express or implied. In addition,
this information is based on IBM's current product
plans and strategy, which are subject to change by
IBM without notice. IBM shall not be responsible
for any damages arising out of the use of, or oth-
erwise related to, this documentation or any other
documentation. Nothing contained in this docu-
mentation is intended to, nor shall have the effect
of, creating any warranties or representations from
IBM (or its suppliers or licensors), or altering the
terms and conditions of the applicable license
agreement governing the use of IBM software.

-
- 1 en.wikipedia.org/wiki/Soa_governance, as sub-
mitted by IBM
 - 2 [ibm.com/webapp/iwm/web/preLogin.
do?lang=en_US&source=esoagov](http://ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=esoagov)
 - 3 [ibm.com/developerworks/webservices/library/
ws-soa-whitepaper](http://ibm.com/developerworks/webservices/library/ws-soa-whitepaper)