zapthink white paper

STIRRING THE SECRET SAUCE OF SOA

PERSISTING & QUERYING METADATA

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

STIRRING THE SECRET SAUCE OF SOA

PERSISTING & QUERYING METADATA

April 2006

Analyst: Jason Bloomberg

Abstract

Service-Oriented Architecture (SOA) represents an approach to organizing and using IT resources that provides increased agility to the business. At the core of any effective SOA implementation are metadata: information about the data and processes that flow through the architecture. Effective metadata management both enables the agility that SOA promises, and also forms the basis for SOA governance. For metadata management to scale, however, companies require robust metadata persistence as well as query capabilities, like those that XML query language XQuery provides.

Table of Contents

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

I. Governance, IT Governance, and SOA Governance

The concept of *governance* is drawing substantial attention today in corporate boardrooms and technical meetings alike, as companies struggle with complex regulatory compliance pressures, increasing globalization, enhanced competition, and the maturation of their markets. *Service-Oriented Architecture* (SOA), an approach to organizing IT resources into business services that the company can build into flexible, agile business processes, is also dominating the conversations in many IT meetings across the globe. As a natural consequence, the concept of *SOA governance* is in the center of the spotlight today.

The concept of SOA governance is in the center of the IT spotlight today.

SOA governance, in fact, has two related but different definitions: first, how to provide for governance of SOA initiatives within the context of IT, and second, how the transition to service-oriented approaches affects the broader area of corporate IT governance. As a result, we must discuss the three related issues of corporate governance, IT governance, and SOA governance, in order to make sense out of SOA governance and how it impacts business today.

IT governance describes how people entrusted with the authority over some aspect of the business will consider IT in their supervision, monitoring, control and direction of that business entity. How they apply IT will have an impact on whether the company will be able to attain the vision, mission or strategic goals that the management of the company has set for it. IT governance specifies who has the rights to make decisions regarding IT, what decisions they can make, and an accountability framework that encourages the IT usage behavior corporate management seeks to exhibit. IT governance is not about making specific IT decisions (management does that), but rather determines which individuals and roles with the company systematically make and contribute to those decisions.

IT governance determines which individuals and roles with the company systematically make and contribute to those decisions.

IT governance is an integral part of enterprise governance and consists of the leadership and organizational structures and processes that ensure that the organization's IT sustains and extends the organization's strategies and objectives. Corporate governance, on the other hand, encompasses both the way people in the organization behave, as well as the rules that govern that behavior. Corporate governance impacts the way managers and shareholders, as well as employees, creditors, key customers, and communities interact with each other to form the strategy of the company.

Corporate governance and IT governance are interrelated. Unlike many other resources available to the enterprise, IT is a horizontal resource in that every department, and virtually every individual within the organization, use IT capabilities as a part of their day-to-day work. As the IT resources these users require become more flexible and generally better able to meet an increasingly broad range of business needs, IT becomes inextricably intertwined in the daily operations of the business. In such a situation, enterprise architecture becomes a critical link between corporate and IT governance, and as companies adopt SOA as enterprise architecture, SOA governance becomes the primary way that companies can establish principles for the control of their organizations.

II. Metadata – The Secret Sauce of SOA

The core challenge of SOA governance is in providing an adequate governance infrastructure while still obtaining the agility and flexibility benefits of the architecture.

The core challenge of SOA governance is in providing an adequate governance infrastructure while still obtaining the agility and flexibility benefits of the architecture. If a company is implementing SOA, but then decides to hard-code governance tools and processes, they will no longer have that agility—and their SOA efforts may backfire. It is essential, therefore, to build flexibility into the governance infrastructure itself.

The key to this flexibility is *metadata.* Metadata are information about data, or more broadly, information about the business processes and rules that the business users work with, as well as information about the systems, applications, data stores, and services in the organization. In fact, most systems produce metadata that contain valuable information. The challenge with those metadata is often how to turn them into insight into the IT environment or the business.

Instead of hard-coding business logic into the programming code, the business logic itself can also appear in the form of metadata, which the programs must be able to deal with. The underlying software must therefore be able to process such metadata. Programmers must understand that the role of the software infrastructure is to deal with data—moving them, processing them, storing them, making them available, and not coding to specific business requirements, since, after all, metadata are data.

The proper use and management of metadata are exceptionally important to the effective operation of the SOA implementation.

Even in a siloed IT environment, in which heterogeneous systems and applications integrate poorly, the use of metadata is important for the effective operation of those disparate applications. However, with SOA, where the architecture provides application and data integration via an abstraction layer, the proper use and management of metadata are exceptionally important to the effective operation of the SOA implementation.

Managing metadata for SOA governance

Fundamental to the abstracted nature of SOA is the role metadata play. Metadata essentially describe the services as well as how consumers of those Services interact with them. Likewise, metadata describe how services abstract the underlying application functionality. Therefore, the metadata that describe the functionality within the SOA also provide both visibility and a measure of control to the people responsible for IT governance.

In fact, the traditional software development lifecycle—requirements definition, design, development, testing, deployment—transforms in the face of SOA. The lifecycle still applies, but now to services instead of siloed applications. As a result, service implementation becomes more iterative than traditional application deployment, as companies take advantage of the fundamentally agile nature of SOA.

Managing the service lifecycle is a critical part of SOA governance.

It is vital that companies incrementally implement services in order to support specific consumer application and orchestration needs, since in SOA, the focus on delivering services to the business necessarily contracts the development and delivery cycle of the underlying technology improvements. Managing the service lifecycle, therefore, is a critical part of SOA governance, and also relies upon effective metadata management for its ongoing flexibility in the face of changing business requirements.

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

Along with the heightened role of metadata within SOA, asset management is also critical to the SOA governance process. In traditional enterprise development, assets such as requirements documents and design artifacts , are of use within particular project teams in specific departments. In an SOA, however, such assets may cross applications and departments. An SOA governance platform must therefore support both asset and metadata management across the organization. It must provide visibility and control over the various elements of the SOA infrastructure to each role, including IT managers, development managers, and architects.

Managing policies

One form of metadata an SOA governance platform must manage is policy metadata. A service *policy* is a set of constraints and capabilities that governs how services and their consumers interact. Simple policies typically include rules describing who can access a service and what credentials they need, how messages should be routed to the service, and what service-level agreements (SLAs) that apply to the service.

It is important that companies codify a set of standard, reusable policies that they can associate with services. Linking services and policies in this way enables the automated validation of services as well as the enforcement of specific policies. It is important, however, for organizations to take business policies, typically in written form, and transform them into metadata-based rules that can help automate the process of validating and enforcing compliance with those policies in both design time and runtime environments.

The first goal of policy management is to detect non-conforming services before anybody puts them into production. After all, heading off such problems early in the lifecycle is less costly to correct and less disruptive to operations than dealing with such issues in a production environment. Many companies will also implement runtime policy management capabilities, including the monitoring of policy compliance and the automatic enforcement of policies as consumers access the services.

It's important for companies to manage policies through their entire lifecycle.

It's important for companies to manage policies through their entire lifecycle, from the creation and communication of policies, to finding and following them, to updating them. In general, managing the lifecycle of policies within SOA focuses on ensuring the quality, performance and applicability of available services, enabling service consumers to discover and reuse services as well as other artifacts, managing service versions, handling the security of services and other SOA artifacts, and assessing and managing the impact of change across all service consumers. Managing policies also includes providing visibility into whether people are following policies, as well as handling policy infractions. Such policy management tasks are also an inherent aspect of IT governance, as well.

Metadata persistence – essential aspect of metadata management

Because there are so many different types of metadata, effective management of those metadata can be a complex task.

Policy management, of course, is only one aspect of the SOA metadata management puzzle. Metadata are information about data, or more broadly, information about the business processes and rules that the business users work with, as well as information about the systems, applications, data stores, and Services that users work with, along with the relationships and dependencies among those elements of the IT environment. Metadata include information about those processes as well. In fact, most systems produce metadata that contain valuable information. Because there are so many different types of metadata, effective management of those metadata can be a complex

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

task. It is essential, therefore, for a SOA metadata management tool to handle diverse metadata in a flexible, scalable manner.

At the center of this metadata management requirement is the need for adequate persistence mechanisms. SOA implementations need metadata persistence mechanisms to store and provide information such as the state of a business step in an application, the state of long-running business processes currently in execution, various Web services management metadata, relationships among services, and lists of available Web services and their contracts. In practice, service consumers and other parts of the SOA infrastructure frequently request and access many of these metadata, requiring scalable caching of metadata in order to alleviate the performance bottleneck multiple requests to the same information store can cause.

In order to preserve the agility of the SOA implementation as it scales, it's essential to implement an XML-based caching architecture.

In order to preserve the agility of the SOA implementation as it scales, it's essential to implement an XML-based caching architecture. Simply storing the metadata, however, is insufficient: equally important is the ability to find and use the appropriate metadata for each particular need. However, since the metadata are XML-based and so diverse, traditional query techniques like SQL aren't up to the task of accessing metadata caches. As the following sections shows, however, XQuery provides an effective tool for completing this metadata persistence picture.

III. XQuery – metadata governance tool

XQuery, therefore, forms the last link in our IT governance chain: IT governance depends on SOA governance, which in turn depends upon policy and other metadata management. Effective metadata management requires scalable metadata persistence, which relies upon effective XML-native query capabilities. Let's take a closer look at XQuery and see how it can be essential to effective SOA governance.

Overview of XQuery

XQuery is a functional language, which means that XQuery developers compose and combine expressions to create arbitrarily complex queries over one or more sets of XML data. XQuery offers both strongly-typed mechanisms using XML Schema and DTD, as well as weakly-typed mechanisms for handling raw XML data, to insure the broadest applicability and flexibility of the language.

XQuery is fundamentally a query language for data stored in XML form, primarily to get information out of XML databases, including relational databases that store XML data or present XML views of data. Some developers also use XQuery to manipulate free-standing XML documents, for example to transform messages passing between applications. In other words, XQuery competes directly with XSLT, and may eventually replace it. XQuery is not only a query language, however, but also a language that can handle general XML processing. XQuery developers can work at various levels of typing, for example, by deciding whether to import schemas or whether to use static typing.

XQuery's similarity to SQL is no accident.

XQuery's similarity to SQL is no accident. In fact, it's possible to use XQuery to create joins between two tables. Such joins are not quite the same in XML as with relational data, however, because XML data are hierarchical rather than tabular, but XQuery's queries are as similar to SQL as makes sense. Its equivalent of SQL's SELECT expression, for example, is the FLWOR expression, named after its five clauses: for, let, where, order by, and return.

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

The formal definition of XQuery centers on a data model, rather than XML text. Every input to a query is an instance of the data model, and the output of every query is likewise an instance of the data model. Every such XQuery data model represents each document as a tree of nodes. Every node has a unique node identity that distinguishes it from other nodes, even from other nodes that are the same in every other way. XQuery then uses path expressions to locate nodes in XML data. XQuery's path expressions are identical to the path expressions in XPath 2.0. The XQuery data model also allows for atomic values such as strings, Booleans, decimals, integers, floats and doubles, and dates. These simple types may occur in any document associated with an XML Schema.

XQuery and metadata persistence – putting power into the SOA registry/repository

An SOA implementation requires an effective mid-tier caching architecture that supports robust query capabilities like those XQuery offers.

In order for an SOA repository to enable increased performance, reliability, functionality, and usability of SOA artifacts, an SOA implementation requires an effective mid-tier caching architecture that supports robust query capabilities like those XQuery offers. Specifically, XQuery-enabled metadata persistence enables a policy-based caching service. This caching service in turn enables XQuery-based policies to cache result sets of poorly performing services.

It's also possible to construct such policies to include the time-to-live before the cache needs refreshing. Furthermore, it's possible to base such policies on time-of-day requests for determining the validity of cached data. Policies based on service availability can then ensure that if the service is not available, the consumer obtains the desired data from the cache, because policies can trigger the XML persistence mechanism.

XQuery-based persistence also enables a data repurposing service to further improve performance. Such a service enables filtering and search criteria on content that a service returns. In addition, XQuery drives the transformations behind the repurposing of the content, as well as providing analytics and reporting capabilities. Taking data repurposing one step further, it's possible to use XQuery to create a data abstraction service, which can eliminate the need for services to be aware of individual data sources.

XQuery-based metadata persistence enables a SOA registry/repository to scale by supporting the federation of services.

Most importantly, however, XQuery-based metadata persistence enables a SOA registry/repository to scale by supporting the federation of services within a SOA implementation. A SOA repository can thus serve as a persistence layer in the middle tier, storing transactional information for many purposes, including analysis and integrity management issues. SOA-enabling technologies such as enterprise service buses and orchestration engines can then employ such an SOA repository for state management, workflow persistence, and message persistence.

It is important for IT users within any SOA implementation to understand the technology available to them, and business users must understand the services and processes they can access and manage. However, it is not sufficient simply to be able to find large quantities of disparate metadata; users must be able to discover just the metadata they need. Therefore, it is essential that users have access to sophisticated metadata search capabilities. However, because of the diversity of metadata within the organization, such search capabilities must leverage the power and complexity of XML. Since most SOA content is XML, XQuery is the obvious choice for enabling such searches.

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

IV. The ZapThink Take

Companies traditionally store metadata in relational databases and file systems, but these tools are not particularly well-suited to handling SOA-related metadata. Being fundamentally XML and hence hierarchical, SOA metadata does not fit well in relational databases. Furthermore, the inflexibility of relational database schemas does not lend itself well to the always-evolving, often *ad hoc* schemas in an SOA implementation, especially in the face of changing business requirements. On the other hand, file systems do not provide the advanced querying and management capabilities that SOA implementations require.

Metadata are becoming the lifeblood of SOA implementations.

In fact, metadata are becoming the lifeblood of SOA implementations, because of their importance to corporate visibility, policy management, and governance. In fact, the more dynamic the business environment, the more important metadata become, because the alternative to building SOA implementations based on metadata is to create tightly coupled links to static information. Such brittle solutions simply do not pass muster in today's turbulent business environment.

Metadata also serve an important unifying role within SOA, as SOA touches upon many different parts of the company, including heterogeneous systems environments, multiple lines of business, and processes that involve all aspects of the business. Many different people in many different roles must work in different ways within the SOA, and metadata helps to provide the governance infrastructure that such diverse teams require.

ZapThink, LLC • 108 Woodlawn Road • Baltimore, MD 21210 • info@zapthink.com • www.zapthink.com

Copyright, Trademark Notice, and Statement of Opinion

About ZapThink, LLC

ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

ZapThink helps its customers in three ways: by helping companies understand IT products and services in the context of Service-Oriented Architecture (SOA) and the vision of Service Orientation, by providing guidance into emerging best practices for Web Services and SOA adoption, and by bringing together all our audiences into a network that provides business value and expertise to each member of the network.

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

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

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

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOA by vendors, end-users, and the press. Respected for their candid, insightful opinions, they are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry. ZapThink was founded in November 2000 and is headquartered in Baltimore, Maryland.

ZAPTHINK CONTACT:
ZapThink, LLC
108 Woodlawn Road
Baltimore, MD 21210
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com