

Managing SOA Metadata: Registries or Repositories?

BY CAROL WEISZMANN
AND SUSAN MESSEHEIMER

As deployments of service-oriented architectures (SOAs) survive initial pilot projects and begin to spread into more extensive use, there's a need to better organize, access and manage runtime metadata such as XML schemas, SOAP messages and WSDL interface definitions. Spreadsheets and Web page listings of services no longer suffice.

"SOA registries are playing a key role in development, specifically in assembling composite applications," said Sanjay Sarathy, vice president of marketing at Above All Software. "Here, you're trying to integrate multiple systems, and you need a place where you can store the metadata and the artifacts from those disparate pieces."

For purposes of reuse, the only way to properly store metadata is in an SOA-type repository. "And it's important not just to store the metadata," Sarathy added, "but to use the registry to consider the relationships across the different elements. Ultimately, you're creating a virtual application portfolio that cuts across your development efforts no matter what application environment you're dealing with."

Various kinds of metadata management solutions are currently in use, but

most provide only limited support for managing runtime metadata. That's because general-purpose metadata repositories haven't been designed or optimized for runtime environments and cannot manage detailed, implementation-level metadata. In contrast, registries deliver more focused management of runtime artifacts (for example, services and directories), although they generally lack the breadth and depth of repository functionality.

Gartner analysts Michael Blechar and Jess Thompson, in "When to Use Metadata Repositories, Registries or Both," offer an analogy:

"Registries play a role similar to that of a card catalog in a library. As a catalog points to the books a reader needs to access, an SOA registry points to the services a requester needs to access. Repositories manage metadata about a broader set of artifacts [such as] libraries, card catalogs, books, and processes such as checking a book out and returning it."

These days, however, emerging SOA deployments are blurring the distinctions between repositories and registries. Repository solutions have begun to incorporate what Gartner calls "registry-light capabilities," while

REPOSITORIES VERSUS REGISTRIES

Here's how Gartner analysts Michael Blechar and Jess Thompson, in "When to Use Metadata Repositories, Registries or Both," compare repositories and registries:

"Repositories primarily support the management of metadata at the contextual, conceptual, logical, and physical levels of detail. Repository parsers and bridges can access and load metadata from a variety of sources, such as business process modeling tools, database schemas, and Universal Description, Discovery and Integration (UDDI) directories. However, they extract only a subset of the metadata used by those tools. Consequently, they house limited detail about metadata at the implementation level.

"Registries primarily focus on the implementation-level information needed to support the deployment, operation, and management of metadata. Registries also include other selected types of metadata from the higher levels (for example, a service registry might include the names and descriptions of the business processes and services being used). Currently, the greatest overlap is in the area of the SOA spoke repositories and service registries."

registries are including metadata beyond what's required for runtime artifacts.

KEY FEATURES OF AN SOA REGISTRY

"A full-featured SOA registry needs to provide the ability to register, discover and access Web services as well as provide a place to associate service descriptions (WSDLs) and other associated metadata, such as XML schemas, transformations, BPEL descriptions, etc.," said Cindy Hayden, senior manager for Web services and XML gateway integration at Atlanta-based credit-card

provider CompuCredit. "It should also provide full change management capabilities."

Added Ben Moreland, assistant director of foundation services in The Hartford's eBusiness & Technology organization: "An SOA registry should include user-friendly design-time lookup, dynamic binding capabilities, governance over the submission, approval and publication of new services, and adherence to the UDDI [Universal Description, Discovery and Integration] standard." Especially useful for maintenance purposes, he noted, is an ability to track consumer-producer relationships in both directions.

UDDI also is seen as central to an SOA registry by Greg Meyer, chief technology officer at iJET International, which provides risk monitoring and management services to multinational corporations via a highly automated Intelligence Operations Center. "At the most basic level," he said, "the features, capabilities and requirements of an SOA registry are met by the existing UDDI specification: maintain standardized information about Web services, record the capabilities of those Web services, and list any location and requirements information about those Web services."

The classic UDDI phone book model isn't enough, though. "Effective registries provide tools, or a starting point, for governance activities, such as reporting and asset control, as well

► continued on page 34



Managing SOA Metadata:

< continued from page 33

as providing mechanisms to map business taxonomies to the registry,” iJET’s Meyer observed.

Beyond the basic UDDI

specification, an SOA registry’s most valuable function, he believes, lies in providing different views for different consuming populations. “An effective SOA registry should

include a user interface that bridges technology and business: Views should be able to be customized to a wide variety of consuming groups.”

IBM WebSphere product

manager Sunil Murthy spelled it out:

SOA repository features: “Storing a fine-grained model of service metadata artifacts—beyond handling WSDL, XSD,

etc., as BLOBs, discovering metadata instead of relying on users to manually populate the repository.”

SOA registry features: “A rich set of capabilities for semantic annotations of service metadata to support service advertisement (i.e., explain what my service does so that people can understand it without crawling through the WSDL), rich queries using semantic annotations, and rich classification models beyond simple taxonomies.”

SOA governance features: “Life-cycle definition and support for managed transitions between life-cycle states (including validation), change notification, impact analysis and a fine-grained access control model.”

DESIGN-TIME VS. RUNTIME

Brent Carlson, co-founder and vice president of technology at LogicLibrary, added another kink: “It’s important to differentiate between a design-time registry and a runtime registry.”

A runtime registry, he explained, provides dynamic lookup capabilities for SOA-based applications to retrieve deployed service instances. A runtime registry must be capable of responding in real time to operational application loads with a limited set of information, and its interface is primarily programmatic.

“Runtime registries such as LDAP have been used successfully for service lookup,” Carlson reported. “In general we’re seeing either integrated ESB/fabric registry capabilities or, to a lesser degree, UDDI registries take on this responsibility.” A design-time registry, meanwhile, provides contextual information about candidate services for use by application developers. A design-time registry must provide a much richer set of information to developers working within their SDLC environment, and its interface is primarily GUI (and ideally IDE-integrated) in nature.

“Legacy mainframe repositories are not well-suited to support design-time service registry functionality,” said Carlson. “Their metamodels tend to be more aligned with data element metadata management, and their user interfaces are not well-integrated with modern SDLC tools.”

Introducing Mindreef Coral™
A collaborative platform for Web services/SOA teams

Corporate Architect
Codify corporate standards, best practices, and governance policy around WSDL and schema

Project Architect
Leverage rules from corporate architects while creating shared workspaces that add documentation, example scripts, and simulation data to enhance the WSDL contract

Service Developer
Use shared workspaces from Project Architects to fully understand requirements. Create unit tests automatically during development

Client Developer
Enter the development process earlier with Mindreef Coral's auto documentation and simulation features

Support Staff
Capture all associated problem artifacts into a Mindreef Shared Workspace®, allowing any team member to reproduce a problem at the push of a button

Testing
Leverage architect's example scripts and developer's unit tests to quickly and easily create robust regression tests

Community Area
Shared Workspace: WSDL Contract, Message, Reusable Script, Simulation
Governance, Testing, Diagnostics, Support
Service Team
MINDREEF CORAL

The Mindreef Coral platform provides collaborative tools for the XML software layer that increase productivity and quality throughout the entire SOA lifecycle.

Our lifecycle approach—from well-adopted best practices early in the process to efficient problem resolution post-launch—drives down the cost of developing and maintaining Web services, leading to more efficient, cost-effective SOA. We designed Mindreef Coral to leverage your most valuable assets—your service team members—to achieve sustainable business results.

Pay as you grow pricing—from your first Web service project to large SOA initiatives. Mindreef Coral starts at just \$999 per year for a server supporting two concurrent users. Additional licenses are just \$499.

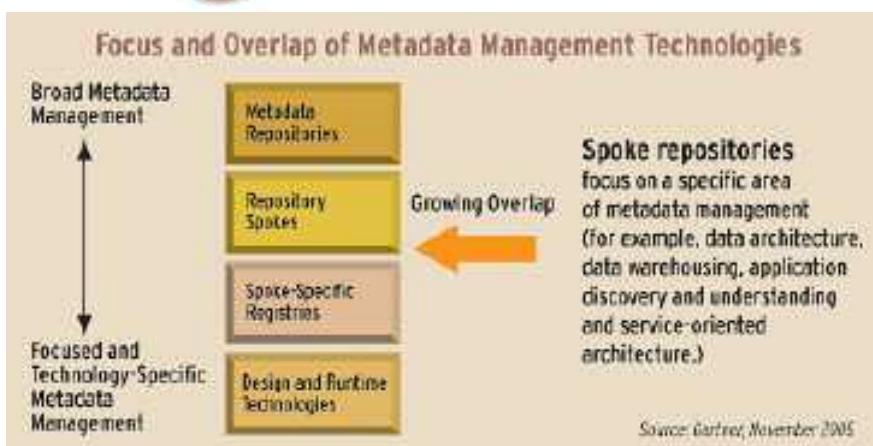
To download a free trial, visit www.mindreef.com, or to speak with a Mindreef representative, call 603-465-2204 x510.

- Governance
- Testing
- Diagnostics
- Support

Mindreef®
People Make SOA Work™

*patent pending
©2005 Mindreef, Inc. Mindreef is a registered trademark, and Mindreef Coral is a trademark of Mindreef, Inc. in the United States and other countries. All rights reserved.

Registries or Repositories?



Chris Warner, director of technical marketing at Software AG, asserted that a complete “registry/repository” solution is one that “combines SOA-specific Web services administration features—i.e., a registry—with features that are common to many developer-centric repositories, such as organization-specific metadata and management of reusable code fragments.”

While non-SOA entities like warehouse metadata dictionaries and enterprise metadata repositories include such capabilities as life-cycle management, metadata management and security, it’s the focus on standards-based interfaces for consumers, producers and administrators that characterizes SOA—and the standards-based nature of the SOA objects managed by an SOA registry/repository.

“This standardization,” said Warner, “allows development/operations teams to utilize a loosely coupled development tool set, much like the loosely coupled SOA they are building.”

Charles Stack, CEO of Flashline, sees SOA as a very different way of thinking of IT functionality. “Issues like centralized Web services, service-level agreements, systematized software asset reuse, in-place versioning, dynamic provisioning, cross-enterprise functionality, project governance, service life-cycle management and common taxonomies must all be addressed.”

Among key benefits of an SOA registry/repository, he noted, are being able to reduce duplication, reuse functionality and remix Web services to compose and recompose business processes.

“Also,” Stack said, “governance lies at the core of an SOA registry/repository, ensuring that Web services and other software assets deliver expected results and align with the architecture and business goals. It’s important to look at governance throughout the entire service life cycle, from planning through runtime. Applying governance during the early stages of planning and design can head off runtime service interruption.”

Jason Bloomberg, senior analyst at consultancy ZapThink, offered a heli-

copter’s-eye view: “As services in the enterprise proliferate and companies build out their SOA implementations, the structure of corporate technology teams must change to enable flexible, agile service development, testing and support.”

Such efforts will stretch far beyond simply using the registry, which is ultimately just one of many elements in the service life cycle, and companies will rely on what Bloomberg calls “service life-cycle teams,” composed of diverse personnel from both IT and lines of business, including business analysts, architects, developers, testers and support staff.

“Fundamentally, SOA cannot scale without effective collaboration among the members of these teams,” asserted Bloomberg. “Furthermore, as enterprise SOA implementations grow, consistent communication among such SOA implementation teams must allow participants to collaborate regardless of each individual’s role, skill set or development environment.”

LEVERAGING NON-SOA METADATA

The SOA registry features and capabilities now emerging are a definite improvement over existing non-SOA registries/repositories that development and operations teams have been using, said The Hartford’s Moreland.

“Current non-SOA registries are not UDDI-compliant and do not provide governance around service submission and publication,” he pointed out. “They rarely provide easy-to-manage taxonomies or make for easy consumer-producer relationships.”

However, the skills required to create non-SOA registries are transferable.

“Given that some non-SOA registries are very sophisticated and require wide-ranging skill sets,” said iJET’s Meyer, “an argument can be made that if one has the ability to create a non-SOA registry, existing skill sets are sufficient for creating an SOA registry.”

CompuCredit’s Hayden agreed. “Teams that are building an SOA registry need to have a full understanding of SOA concepts and be skilled in Web service creation and deployment. Ideally

they should have experience in application integration and have a complete understanding of remote object access concepts.”

Added Moreland: “Companies looking within their organization for skilled employees to create an SOA registry should first turn to their enterprise architects, then their database support people and finally their data modelers.”

iJET’s Meyer distinguished between development and operational issues. On the development side, success depends on something that those building non-SOA registries may not possess: an intimate understanding of UDDI. Operationally, he said, “the skill sets should focus on the ability to render registries for a wide variety of consuming groups, so business and stakeholder understanding is important.”

Meyer also pointed out that “some of the existing [and emerging] SOA registries and registry tools are very sophisticated, user-friendly and powerful. This provides not only the granularity and detail necessary for the developer, but also the consolidated, nontechnical interface for the business user.”

The ability to rapidly assemble and create relationships among different services is fundamental to SOA-based development, noted Above All Software’s Sarathy. “In an SOA world, that requires a different registry than the one you use to manage governance and service policy—but the two need to be integrated.”

Using registries and repositories appropriately takes organizational discipline, observed Randy Heffner, vice president of the application develop-

ment and infrastructure research group at consultancy Forrester Research.

“This,” he said, “means things like correctly populating services, correctly describing services, providing quality services that people can trust, having the discipline to look for existing services before building new ones, etc. Without these disciplines, you will waste money putting a registry or repository in place.”

What, then, are the best ways for development and operations teams to leverage their existing skills in creating an SOA registry?

“First,” advised Sandy Carter, vice president of SOA and Web services at IBM, “step back and identify the core business processes that will be leveraged in the SOA.”

The team that does this work should include representatives from all parts of the organization. “Once business process redundancies are streamlined,” Carter continued, “the team can collectively share and determine which business processes, best practices and Web services will be applicable to other parts of the organization for reuse. The true value of an SOA is that it leverages existing skills and doesn’t require development or operations teams to have to learn new skills in order to create an SOA registry.”

Flashline’s Stack advised focusing on the “A” in SOA. “The most important thing is to take advantage of the architecture group within the organization,” he said. “If you don’t have an architecture group, be sure to leverage the experience and skills of architects, and approach SOA as an architectural initiative with a business imperative.” ■

WHAT ROLE UDDI?

“One could argue that non-SOA registries provide all of the discovery, description and location capabilities of today’s SOA-based registries,” said Greg Meyer, chief technology officer at iJET International.

The primary difference between the two, he said, is the widely embraced industry standard specification, UDDI (Universal Description, Discovery and Integration).

More Than a Registry: Randy Heffner, vice president of the application development and infrastructure research group at consultancy Forrester Research, explained. “Before v3.02, UDDI was only a registry,” he said. “With v3.02, UDDI provides some very basic raw materials for it to be used as more of a repository—but no standards for how to do so.”

UDDI, noted Heffner, operates more as part of the development life cycle, much as a traditional repository would. “As UDDI becomes used more and more as a repository,” he said, “then the difference has more to do with the use of emerging WS standards for policy, reliable messaging, transactions, security, etc.—each of these creates metadata to be controlled within the repository.”

Interoperability and Federation. With UDDI as the center post, Meyer advised, “it’s now possible to realize two things that were simply not possible in the past: interoperability and federation. With a standard specification like UDDI, disparate and large communities can create registries, and the services described in these registries can be easily consumed by a wide variety of consumers.”

And it does not stop there. “Taking this interoperability concept to the next level,” Meyer noted, “a standard specification allows the creation of federated registries, or multiple registries that can be viewed as a single entity.”

—Carol Weiszmann and Susan Messenheimer