Introducing IBM WebSphere Service Registry and Repository, Part 1: A day in the life of WebSphere Service Registry and Repository in the SOA lifecycle

Barbara McKee
Marc-Thomas Schmidt
John Colgrave
Duncan Clark

This is the first article in a series introducing the main concepts and capabilities of IBM WebSphere Service Registry and Repository. It explains the role of Service Registry and Repository throughout the service-oriented architecture (SOA) lifecycle and provides resources to help you learn more about it.

View more content in this series

Introduction

Service-oriented architecture (SOA) offers the promise of business agility and resilience through reuse, loose coupling, flexibility, interoperability, integration and governance. These are realized by separating service descriptions from their implementations, and using this descriptive metadata across the service lifecycle. Standards-based service metadata artifacts, such as Web Service Definition Language (WSDL), XML schema, policy or Service Component Architecture (SCA) documents, capture the technical details of what a service can do, how it can be invoked, or what it expects other services to do. Semantic annotations and other metadata can be associated with these artifacts to offer insight to potential users of the service on how and when it can be used, and what purposes it serves.

Service metadata is used by analysts, architects, and developers during the model and assemble stages of the SOA lifecycle to locate services to reuse and to evaluate the impact of changes to service configurations. Service metadata is used by deployers and administrators and exploited by the SOA foundation run times in the deploy and run phases of the SOA lifecycle for dynamic selection of service endpoints and configuration of the SOA fabric. It is used in the manage phase of the lifecycle to support policy enforcement required by Service Level Agreements (SLAs) and to present a more comprehensive view of the managed service environment. For a detailed description of IBM's SOA Foundation, the SOA lifecycle and reference architecture, see IBM SOA Foundation: An architectural introduction and overview.
This article discusses the capabilities of IBM® WebSphere® Service Registry and Repository in the SOA lifecycle and describes how various users can work with it within each phase of the SOA lifecycle. The article also provides various resources you can use to get started with WebSphere Service Registry and Repository.

What is WebSphere Service Registry and Repository?

WebSphere Service Registry and Repository is the master metadata repository for service interaction endpoint descriptions. A broad definition of "service" applies here. This includes traditional Web services that implement WSDL interfaces with SOAP/HTTP bindings as well as a broad range of SOA services that can be described using WSDL, XSD and policy decorations, but might use a range of protocols and be implemented according to a variety of programming models. For more information, see SOA programming model for implementing Web services, Part 1: Introduction to the IBM SOA programming model.

As the integration point for service metadata, WebSphere Service Registry and Repository establishes a central point for finding and managing service metadata acquired from a number of sources, including service application deployments and other service metadata and endpoint registries and repositories, such as Universal Description, Discovery, and Integration (UDDI). It is where service metadata that is scattered across an enterprise is brought together to provide a single, comprehensive description of a service. Once this happens, visibility is controlled, versions are managed, proposed changes are analyzed and communicated, usage is monitored and other parts of the SOA foundation can access service metadata with the confidence that they have found the copy of record.

WebSphere Service Registry and Repository does not manage all service metadata, and it does not manage service metadata across the whole SOA lifecycle. It focuses on a minimalist set of metadata that describes capabilities, requirements, and the semantics of deployed service endpoints. It interacts and federates with other metadata stores that play a role in managing the overall lifecycle of a service.

Understanding a business service, WebSphere Service Registry and Repository, and the SOA lifecycle phases

To illustrate how to use WebSphere Service Registry and Repository across the SOA lifecycle phases, let's take a look at a day in the life of a business service -- from it's invention in the model phase of the SOA lifecycle to its translation into an executable composite application in the assemble phase to its transition into the SOA run time environment in the deploy and run phase to it being observed and controlled in the manage phase of that lifecycle, see Figure 1.
Figure 1. WebSphere Service Registry and Repository SOA tasks

Using WebSphere Service Registry and Repository in the model and assemble phases

During the development stages of the business service lifecycle, WebSphere Service Registry and Repository is used to locate the copies of record of candidate service interaction endpoints or mediating intermediaries as well as policies governing the interactions.

As in the other SOA lifecycle stages, WebSphere Service Registry and Repository is complemented by repositories that specialize in managing SOA artifacts during the development stages of the lifecycle. For example, a development artifact management system such as Rational® Clearcase takes care of service and composite application building blocks such as source code, service interface declarations, software architecture models or business process models that are under construction. A Reusable Assets Manager and Asset Repository manage bundles of artifacts describing assets according to the Reusable Asset Specification (RAS) standard. It implements governance processes controlling the promotion of artifacts to assets and the approval process associated with that. For more information, see the Object Management Group website.

WebSphere Service Registry and Repository complements those repositories and federates the data they manage; for example, for a given WSDL service declaration, it can have a reference to a reusable asset declaration that WebSphere Service Registry and Repository users can follow to find more information about intended usage and function of that service, or Reusable Asset managers can use to analyze usage of the asset (for example, how many deployed services are based on it).

Business analysts, solution architects, and developers create service metadata and make use of existing service metadata as they perform their specific tasks. They use development artifact management systems to take care of your intermediate work products and contribute to the definition of reusable assets that are hosted by a RAS manager. Business analysts, solution architects, and developers use WebSphere Service Registry and Repository to explore the set of already deployed services as building blocks for the new things they are building, and they produce service metadata that end up in WebSphere Service Registry and Repository once the underlying service makes it out of development into a deployed state (test, staging, and...
production). Their interest in and contribution to WebSphere Service Registry and Repository content differs according to their specific tasks.

Here’s a brief summary:

**Analysts** model a new business processes and browse WebSphere Service Registry and Repository to better understand the existing environment. As an analyst, you are less interested in IT-level technical details of services like their WSDL interfaces, than in a business view on service capabilities established by WebSphere Service Registry and Repository via semantic annotation of the underlying IT-level artifacts. Analysts may work with the **asset manager** to define and apply classification systems used to describe business semantics of services.

**Solution architects** use the Service Registry and Repository to find existing services you can use as building blocks for assembly of new composite services. As a solution architect, you can also use service descriptions you find in WebSphere Service Registry and Repository to populate the set of building blocks analysts use when modelling new business processes, abstracting the IT specifics into process modelling artifacts.

**Component developers** build new service components and can browse in WebSphere Service Registry and Repository to find services you can invoke as part of your implementation or to scope queries against WebSphere Service Registry and Repository that you might want to use as part of your service implementation. Component developers can use WebSphere Service Registry and Repository to analyse the impact an envisioned change on a service might have on other services.

**Integration developers** assemble solutions from new or existing components. As an integration developer, you use WebSphere Service Registry and Repository to find building blocks that you can bind references in your composite applications to. Integration developers model the mediations necessary to facilitate interactions between service interaction endpoints, for example, using WebSphere Service Registry and Repository lookup to select a service endpoint to be used in a dynamic routing scenario. In coordination with the asset manager role integration developers also can discover existing service endpoints and publish them in WebSphere Service Registry and Repository.

For more information about the user roles discussed in this article, see [An SOA programming model for implementing Web services, Part 10: SOA user roles](https://ibm.com/developerWorks/).

**Using WebSphere Service Registry and Repository in the deploy and run phases**

WebSphere Service Registry and Repository provides the system of record for metadata describing service interaction endpoints. It is populated with metadata as part of SOA solution deployment or via discovery of existing endpoints; and it is used by SOA run times as well as **Deployer** and **administrator** roles when detailed information about service endpoints is required to drive operations of the deployed composite applications.

Once the development team has finished their work and testing is complete, Deployers further augment the service metadata, providing binding information for service endpoints used in
composite applications and managing deployment of the metadata from the development environment to the staging or production Service Registry and Repository as part of the service deployment process. Governance over the service metadata takes place, as metadata is promoted from test to staging to production environments that may have separate WebSphere Service Registry and Repository installations. In the production environment, Service Registry and Repository is made available to a broader audience and shared. It is available to the runtime systems and those user roles that are responsible for the management of the IT systems.

Here again, new service metadata, or more often a change in existing service metadata, can be discovered in other service endpoint registries and repositories, published in WebSphere Service Registry and Repository, and can be used as input for the application configuration and binding tasks performed by the Deployers and Solution Administrators. Discovered service metadata is usually incomplete and not yet suitable for broader visibility and consumption. Deployers work with asset managers to ensure the metadata is augmented with the necessary semantics, permissions and scoping constraints.

Automation of service metadata publication during service application deployment integrates the management of service metadata with the overall SOA management and governance processes in a first class way.

**Component developers and integration developers.** Execution time interactions with the Service Registry and Repository can be implemented in a service endpoint by a component developer or by a mediating intermediary that acts on behalf of the service requester or the provider and is configured by an integration developer. Dynamic endpoint selection is a key use case that involves querying the Service Registry and Repository for the service provider or a set of candidate service providers, and binding to the endpoint that is the best choice. Endpoint selection rules can be encoded in ESB-managed mediations. For more details, see SOA programming model for implementing Web services, Part 4: An introduction to the IBM Enterprise Service Bus.

Another use case is dynamic retrieval and enforcement of the policies that are in effect for a service interaction in the areas of logging, filtering, data transformation, or routing. WebSphere Service Registry and Repository stores the policy definitions as well as information about which service interaction endpoints they apply to; policy enforcement points can be configured using that information and can be update when WebSphere Service Registry and Repository-managed metadata changes.

SOA run time elements will often maintain configuration information beyond the minimalist set of service metadata managed by WebSphere Service Registry and Repository; they will cache WebSphere Service Registry and Repository managed metadata and relate it to the additional configuration information they maintain.

**Solution administrators** use WebSphere Service Registry and Repository content to better understand the solution artifacts they are administering, and in the future may be able to dynamically affect changes in the configuration of deployed applications by updating WebSphere Service Registry and Repository content (for example, replacing documents describing endpoint selection rules or applicable policies). Clearly those types of administrator-driven updates must
be controlled and that's achieved using WebSphere Service Registry and Repository governance support for the artifacts in question which controls visibility of artifacts as well as who can perform which actions on specific governed entities.

Using WebSphere Service Registry and Repository in the manage phase

Operational management and resilience within the SOA is enhanced by sharing the service metadata that exists in the Service Registry and Repository with operational data stores, allowing management and monitoring dashboards to present a more comprehensive view of the managed service environment. Summary information about service performance can be fed back into WebSphere Service Registry and Repository and used by the execution environment to affect the selection of the best-fit provider.

As in the other lifecycle facets, WebSphere Service Registry and Repository only manages a minimalist set of service metadata and federates with repositories specializing on managing information about services in this lifecycle stage. For example a Configuration Management Database (CMDB) like ITSM Change and Configuration Management Database will acquire and manage detailed information about the environment and topology in which service endpoints execute. Service management products like IBM’s ITSM suite will use and update that information and drive governance of processes that provision and configure the underlying infrastructure.

WebSphere Service Registry and Repository and CMDB federation enables WebSphere Service Registry and Repository users to drill down into information about environment and runtime status of a service while CMDB exploiters can obtained detailed descriptions of shape and semantics of service endpoints from WebSphere Service Registry and Repository. Service monitoring and management products like ITCAM for SOA provide instrumentation of service interactions that allow monitoring of service interactions and service endpoint behaviour. Summary information about service behaviour can be pushed into WebSphere Service Registry and Repository to decorate service endpoint metadata with execution statistics. In the future service management products can use WebSphere Service Registry and Repository managed policy information to configure policy enforcement points that implement the SLAs users want to enforce in service interactions.

Incident analysts and business operations managers. In the operator user role category, the incident analyst investigates unexpected incidents when they occur in the IT system and manages service endpoints. You consult the metadata and summary statistics about service endpoints found in the Service Registry and Repository to understand the behaviour of the IT systems and to assess the impact of an underlying failure. The business operations manager analyses performance of SOA applications from a business value perspective and uses metadata provided from WebSphere Service Registry and Repository to understand application semantics and to compare expected with observed behaviour of those applications.

Conclusion

This article explained the main concepts and capabilities of WebSphere Service Registry and Repository and its place in the SOA lifecycle. It describes WebSphere Service Registry and
Repository perspective for each lifecycle phase and explains the user roles and WebSphere Service Registry and Repository interactions. The article also provides additional resources to help you get started with WebSphere Service Registry and Repository.

Acknowledgements

Special thanks to the following for their contributions in shaping WebSphere Service Registry and Repository: Mandy Chessell, John J. Choi, Raymond Ellis, Andrew Hately, Beth Hutchison, Ed Kahan, Susan Malaika, Sunil K. Murthy, Birgit Schmidt-Wesche, Mike Starkey, Willi Urban, and Dan Wolfson.
Related topics

- WebSphere Service Registry and Repository product documentation in the IBM Knowledge Center
- SOA programming model for implementing Web services article series
- WebSphere Service Registry and Repository developer center

© Copyright IBM Corporation 2006
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)