IBM Software Demos

IBM SOA WebSphere Service Registry and Repository

IBM WebSphere Service Registry and Repository Animation

You know, you can think of services as being kind of like clothes. Yeah, It's true!

You need to have the right clothes for the right occasion.

You need to have clothes that work in multiple situations.

You need to be able to change clothes quickly, and adjust your clothes to fit you correctly. And you need to be able to combine clothes to make something different.

But even when you have all the right clothes - if you can't find what you need, when you need it – then they're useless.

Uh-oh.

It's the same with services – the trick is knowing which services you have, and which one to use when.

That's where the IBM WebSphere Service Registry and Repository becomes an invaluable part of your business.

If you are implementing a Service Oriented Architecture, or SOA, the WebSphere Service Registry and Repository makes it easy to quickly find and utilize the right service at the right time.

IBM SOA WebSphere Service Registry and Repository

Let's see how it works here at JK Enterprises Bank.

For the sake of this demo, I'll play the role of the IT Director. Every day, the Bank needs to cost-effectively process hundreds of loan applications without overloading the back end applications like CICS, SAP and DB2. We use the WebSphere Process Server to build and execute loan process flows with our Enterprise Service Bus to integrate and connect the disparate systems we have.

But even with a great solution like this in place, we could miss out on the full benefits of SOA, especially if we can't do these two things:

If we can't ensure that our business processes are agile enough to respond quickly to market forces – forces that call for changes to services by internal and external providers.

And if we can't successfully reuse our existing services to rapidly reassemble and reconfigure applications or processes - so that our projects are on time and within budget.

That's what's important to the CIO. What it means to the SOA architects is that we need to have the right tools in place to know what services are being used, how they are being used, who is using them and the business rules that apply to these services.

We need to know how the various services interact with each other and how we can create flexibility in this interaction and optimize it.

And we need to prevent unnecessary duplication of services and reuse as much as possible.

IBM SOA WebSphere Service Registry and Repository

What we need to meet these challenges is a powerful approach to managing our services—that's where the IBM WebSphere Service Registry and Repository comes in.

WSRR is an industrial strength tool that enables companies like ours to publish, find, enrich, manage, and govern our services in an SOA.

How does it work?

For starters, let's look at the Publish and Find functions. These functions deliver visibility and control of the services in an SOA. WSRR federates with other standard registries and repositories, to provide one reliable, consistent service reference.

The Enrich function enables dynamic and efficient access to services information by both run-time applications and processes that facilitate better connectivity and efficiency.

The Manage function promotes optimal interaction of services by managing information about their performance, availability, relationships and dependencies.

WSRR enables enterprise governance by handling decision rights, policies, versioning, classifications, and change notifications throughout the service lifecycle.

Here at JK Enterprises Bank, we understand that without effectively managing and governing our services, we can miss out on a lot of the benefits of SOA. Let me tell you a story to help explain.

IBM SOA WebSphere Service Registry and Repository

IBM SOA WebSphere Service Registry and Repository

Last year, our CTO presented us with a challenge: to get the necessary processes in place so that we could offer auto loans to our customers, in addition to the home loans that we already handled. And, of course, we had to make sure that we were compliant with all the policies we had in place for processing loan applications.

Here are the key players of the team that made it happen: Tom, our in-house guru on insurance business; Amy, an expert in enterprise architecture implementations; and Mark, who manages deployment of our new products.

The project began with Tom, who first created a service concept in WSRR to represent the Auto Claims Service that we envisioned. He assigned it to the "model" phase and classified it under 'claims management'. Next, he determined that a "risk policy" was needed, not unlike home loans, to distinguish between low and high risk claims. He created that policy in WSRR and attached it to the 'concept' service.

Satisfied with his analysis, Tom transitioned the concept service to the "Assemble" phase, and WSRR sent a notification to Amy, telling her to take over.

Using WSRR, Amy could clearly see what Tom had envisioned. She designed the Auto Claims service, and the associated risk policy. Using WSRR she found that an existing policy validation service could be reused - but needed updates. So she used WSRR to see what would be impacted by that change. She learned that the applications which processed Home Loans would be affected by the change. Thanks to WSRR, however, she was able to notify all the users of that service and make sure everything kept running smoothly.

Amy then worked with her team to complete the new components, carefully scrutinizing all the changes. Then, she transitioned the service to the "deploy" phase using WSRR – comfortable in the knowledge that, even if she HAD

IBM SOA WebSphere Service Registry and Repository

missed something, the validations run by WSRR would make sure all the Quality Assurance activities were completed.

Finally, the service went to Mark, who was confident about the quality of the new changes. Plus, it was reassuring for him to know that throughout the process, WSRR's access control rules validated user rights in making those changes. He associated the service with service endpoints, and ran a suite of tests. He promoted the service to the production environment, and assigned it to the "manage" phase.

And just that easily, the Auto Claims Service sees the light of day. Tom is happy, Amy is happy, Mark is happy, and the CTO is happy. And most importantly, our customers are happy.

WSRR enables SOA governance in several ways. First, it supports role-based access to services, to control visibility and improve reuse. Second, it manages changes to services and the related metadata. Third, it organizes and classifies services using your business terminology. Fourth, it helps implement best practices throughout the service lifecycle. Finally, it provides Governance Profile, a customizable service tool to get you started -quickly and easily.

Now that's pretty a high level overview of what it can do, but let's take a deeper look.

Here's how WSRR optimizes the dynamic interaction of a client application or a process with a service provider.

The Enterprise Service Bus receives a message from a client application or a business process that requests a service which meets the criteria encoded in the message.

IBM SOA WebSphere Service Registry and Repository

The Enterprise Service Bus sends the message to a mediation flow. Based on the request, the mediation looks up the candidate services in WSRR. For each of these services, WSRR provides information about the service in terms of its availability, usage criteria and performance.

The mediation then selects the optimal service provider that matches the request and routes the request to that provider.

And here is how we quickly build a new business process using existing services discovered by WSRR.

Using WebSphere Integration Developer, a tool for WebSphere Process Server, an Integration Developer can search for the existing services in WSRR; import them into the workspace and then simply drag-and-drop them on the assembly diagram to build a new business process.

Once built, the new business process can be exposed as a web service. The Integration Developer can easily publish the new process web service to WSRR by using the "Publish Document" option.

One of the great things about WSRR is how well it integrates with other products to maximize reuse and connectivity.

For example, WSRR can seamlessly integrate with the IBM ESB products.

WebSphere Enterprise Service Bus can dynamically route client requests to services managed by WSRR by way of a mediation that enforces the business rules to select service providers.

IBM SOA WebSphere Service Registry and Repository

IBM SOA WebSphere Service Registry and Repository

WebSphere Message Broker can also use service information from WSRR to resolve and select service endpoints dynamically at runtime. This is enabled by predefined WebSphere Message Broker nodes that perform the WSRR look-up.

WebSphere DataPower Integration Appliance XI50 retrieves and binds to services that are managed by WSRR. It also enforces runtime policies and security that are associated with the services.

Also, WSRR tightly integrates with IBM CICS Transaction Server for z/OS version 3.

CICS can be both a provider and requester of web services. When it is a provider, it can publish CICS applications exposed as web services to WSRR, and as a requester it can use WSRR as a catalog of available web services.

As you can see, WSRR increases runtime flexibility of our applications by providing intelligent service selection for the IBM ESB products.

WSRR promotes reuse and eliminates redundancies by increasing visibility of services across the organization, which enables us to realize new business processes using existing services.

WSRR enables SOA governance for better control of our SOA maximizing its business value.

The IBM WebSphere Service Registry and Repository will help you publish, find, enrich, manage and govern your services. So you'll never be without the outfit you need.

And once you have all the right clothes, you can go anywhere.

IBM SOA WebSphere Service Registry and Repository

IBM Software Demos
IBM SOA WebSphere Service Registry and Repository

IBM SOA WebSphere Service Registry and Repository