What are Metadata?

- Literally, *data about data*
- More broadly, *data about the workings of a system*, as opposed to the data the system works with

This is a purchase order
- It has an address field
- It has an item field
- It has a dollar field

...
Metadata for SOA

• Contract metadata
  – What a Service should do (functional)
  – How a Service should behave (non-functional)
• Process metadata
  – How processes are configured
  – How processes are composed
• Policy metadata
  – What rules apply in various situations
• Schemas
• ...

Goals of Metadata Management

• Design time storage and discovery of all kinds of metadata
• Service versioning/lifecycle management
• Runtime Service discovery
• Governance
• Composite application/business process management

SOA is not about the code... it’s about the metadata
Supporting the Move from Code to Composition

- Fundamental shift from code-centric business logic to metadata-centric
- It’s not the Service that makes the application, but the composition and the policy
- Goal: users build applications out of Services

Registries vs. Repositories

- Repository: Where assets are stored (metadata and other assets)
- Registry: How to find and work with the assets you need – wherever they are (metadata about those assets)
- Products have aspects of both
- People confuse the terms because both data and metadata must be managed in an SOA
What is a Registry?

- **Metadata System of Record** for SOA
  - Interfaces
    - UDDI – *Web Services*
    - LDAP
    - Microsoft Word
    - Word of Mouth

- What do you want from your registry?
  - Design time capabilities to discover Services
  - Interoperability across runtime infrastructure
  - Ability to use at runtime for binding...

What is a Repository?

- **Asset Store** for SOA
  - Artifacts of design
    - Conformance Documentation
    - Contract versions
    - Sample Code
    - Profiles
    - Schema
  - Runtime operational store
    - Messages
    - Policies
    - Logs, Security Certificates, Keys, SAML artifacts
    - Transformations
    - Contracts

- Interfaces:
  - ebXML
  - CVS / Source Code systems
  - WebDAV
  - File Systems
  - SQL
  - XQuery
The Business Benefits

- **Provide Governance**
  - Single point of control
  - Enable distributed development, but centralized management
  - Simplify policy design and management

- **Facilitate Reuse**
  - The more you know, the more you can reuse
  - Reduces redundancy
  - Enables discovery

- **Increase Agility**
  - Runtime usage of registries and repositories enables dynamic change
  - Metadata controls it all
  - Contract negotiation?

Standards vs. Products

- STANDARDS provide *interoperability* while PRODUCTS provide *functionality*
- UDDI: A standard for Registry Interoperability; ebXML & JAXR other registry-related standards (may be more?)
- UDDI and the other standards are NOT products!
- There are many other ways to interact with registries
- Registries do more than just UDDI
- Repositories have a UDDI role, too!
How do you manage change?

- SOA is all about continuous and sometimes unpredictable change
- Development issues
  - How to handle versioning?
  - How to handle metadata management?
  - How to develop changing policies?
- Runtime issues
  - Service availability
  - Policy enforcement
  - Guarantee service-level agreement
  - Maintain low TCO

Handling Service Versioning

- New requirements may involve only process configuration changes
- Services may support multiple contracts
- New requirement may require new contract
- Policy drives version selection & deprecation
Service Lifecycle Considerations

- Lines of business should work with metadata — no IT involvement
- Services go thru individual lifecycles – development, test, production, revision
- Service development driven by Service contracts

The Services Team

- Business analysts/business process architects
  - Adjust Service specs depending on business requirements
  - Reconfigure Services & composite applications
- Enterprise/SO architects
  - Codify policy and best practices
- Technical/project architects
  - Specify implementation
- Service provider/consumer developers
  - Implement requirements and policy
- Testers
  - Insure conformance, simulate Service providers & consumers
- Support personnel
  - Respond to user issues
Handling The Process of Change: Governance

- How do you make sure *people* are doing what they should?

- How to handle the process of changing metadata, services, processes, and practices?

- How to establish repeatable, predictable, and auditable methodologies and practices for SOA?

- How do you make sure changes don’t result in runaway problems?
SOA and IT Governance

- **Policy management**
  - SOA configured & controlled via metadata, including policy
- **Visibility**
  - Services abstract heterogeneous data sources, providing necessary business intelligence
- **Flexibility**
  - Ability to build Services that address compliance issues and adjust them as regulations or business needs change

IT Governance

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Registry/Repository</th>
<th>IT Governance</th>
</tr>
</thead>
</table>

- Establishing and communicating the policies that employees must follow
- Giving employees the tools they need to be compliant with those policies
- Providing visibility into the levels of compliance in the organization
- Mitigating any deviations from established policy
- Registries/repositories provide control points for SOA governance at design time as well as runtime
In Summary...

- Registries: Common control, trust, reference, & understanding of service metadata
- Repositories: Common storage & control of service assets
- Registries & repositories used at both design time & runtime
- Service lifecycle requires collaboration on Services team
- Governance depends on registries & repositories
- Governance is just as important at design time as in runtime

ZapThink’s New Book

- This presentation is based on our upcoming book, *Service Orient or Be Doomed! How Service Orientation will Change your Business*
- Published by Wiley, available in the spring
- Pre-order now on Amazon!
ZapThink is an industry analysis firm focused exclusively on XML, Web Services, and Service-Oriented Architecture.

Thank You!

Jason Bloomberg
jbloomberg@zapthink.com

Ronald Schmelzer
rschmelzer@zapthink.com