

Contracts, Policies & Governance with SOA

Jason Bloomberg and Ron Schmelzer
Senior Analysts
ZapThink, LLC

Take Credit Code: CPGSOA

Copyright © 2005, ZapThink, LLC

zapthink



Level Set – What is SOA?

- SOA is *architecture* – a set of best practices for the organization and use of IT
- Abstracts software functionality as loosely-coupled, business-oriented *Services*
- Services can be composed into *business processes* (which are also *Services*) in a declarative manner



Copyright © 2005, ZapThink, LLC



zapthink

Service-Oriented Architecture

- The Definition:
 - An approach to building and managing distributed computing infrastructures that considers **IT resources as Services available and discoverable on a network.**
- The Implication:
 - Rather than dealing with isolated systems that must be *integrated after the fact*, Service Orientation provides business users with understandable Services they can call upon and compose into business processes as needed – building **systems that can adapt as the business changes.**
- The Benefit:
 - The Service Orientation vision is therefore one of providing the **business values of agility and flexibility** for users of technology, coupled with an *abstraction layer* that simplifies the complexity of today's heterogeneous IT environments from those users.

Copyright © 2005, ZapThink, LLC



zapthink

Characteristics of SOA

- ✓ Services are *loosely coupled* – making a change to a Service provider does not mandate changing any Service consumers
- ✓ Business processes are composed of Services, and are in turn exposed as Services
- ✓ Services are *policy-driven* – business users can change how a Service behaves
- ✓ Systems are *inherently integrated* by virtue of composable services – not through layers of middleware
- ✓ Services leverage legacy systems – SOA does not mandate replacement of runtime infrastructure
- ✓ In SOA, *metadata* control how the system behaves, not code. Business logic trumps application logic
- ✓ In SOA, it's the *contracted interface* that matters, not the underlying runtime environment.

Copyright © 2005, ZapThink, LLC



zapthink

The Challenges with doing SOA right

- **Architectural issues**
 - Enterprise Architecture capability maturity
 - Building SOA the *right* way
 - XML Performance issues
- **Organizational issues**
 - Pulling together the SOA team
 - Who's in charge of Services?
 - How will you pay for Services?
 - How will SOA affect your organization moving forward?
- **New tool & infrastructure requirements**
 - Messaging
 - Contract development
 - Composite application development
 - Metadata management
 - Policy infrastructure

Copyright © 2005, ZapThink, LLC



zapthink

SOA is Architecture

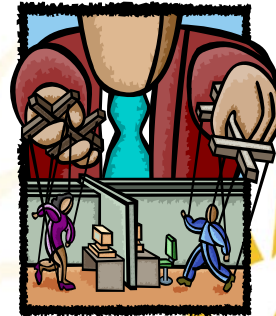
- Remember...SOA is architecture – in particular, *Enterprise Architecture*, including:
 - An aggregated architecture of all the individual IT systems within an organization
 - The human element within the enterprise
 - Systems, people, and organizational constructs at other companies that have relationships with the enterprise
 - Individual consumers who are that enterprise's customers
 - Corporate governance

Copyright © 2005, ZapThink, LLC



Corporate Governance

- 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



From project to program to sustainable process

Copyright © 2005, ZapThink, LLC



SOA 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

***Not just governance of SOA...
governance in the context of SOA***

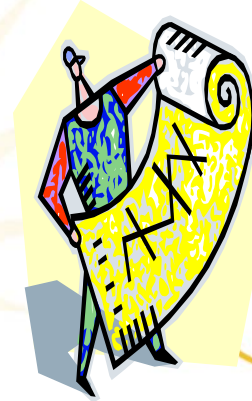
Copyright © 2005, ZapThink, LLC



zapthink

Building SOA the *Right* Way

- **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



Copyright © 2005, ZapThink, LLC



zapthink

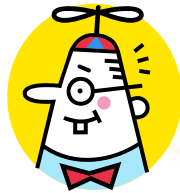
Top-Down & Bottom-Up Approaches

- Top-down only: have the plan, may not be able to execute
- Bottom-up only: build Services, may not be reusable
- SOA planning *must* be both
 - Develop the vision (but not the details) ahead of time
 - Service development should be iterative

Copyright © 2005, ZapThink, LLC



Bring Together Different Mindsets



- Developer Mindset: “Bottom-Up”
 - Everything is a Service or an Interface
 - Goal: connect Services
 - Method: Use objects and App Servers
 - Problem: Too many things to connect!



- Business Mindset: “Top-Down”
 - Everything is a Process
 - Goal: Run business efficiently: manage processes
 - Method: Use diagrams and flowcharts
 - Problem: How can you turn “shelf-ware” into software?

Copyright © 2005, ZapThink, LLC



Challenge: Service Granularity

- The trick of building composable Services is building at the right level of granularity
- Challenges:
 - Engraining business logic into code
 - Decomposing legacy services that are not fine-grained enough
- Method
 - Top-down process decomposition, vs. bottom-up Service development
 - Must be iterative

Copyright © 2005, ZapThink, LLC



zapthink

Challenge: Contracts & Policies

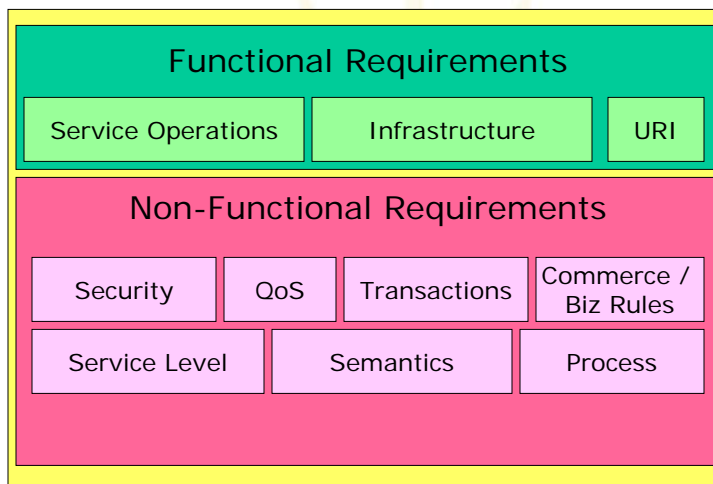
- **Step One: Service Description**
 - What does the service provide for prospective consumers?
 - Description of what is accomplished by the service
 - Quality of service or applicability constraints
 - Constraints on types of consumers or other consume requirements
- **Step Two: Service Usage Description**
 - How is service used?
 - Message and semantic formats for requests
 - Conditions for particular outcomes / behaviors
 - Process leading to outcomes
- **Step Three: Service Interaction Description**
 - How can we communicate with Service?
 - Description of acceptable communication protocols
 - Invocation style (request/response, one-way, event-driven)
 - Expectations of latency

Copyright © 2005, ZapThink, LLC



zapthink

What's in a Contract?



Copyright © 2005, ZapThink, LLC



zapthink

What's NOT in the Contract

A "Contract" is an expression of **visible** aspects of Service behavior.

It does NOT specify:

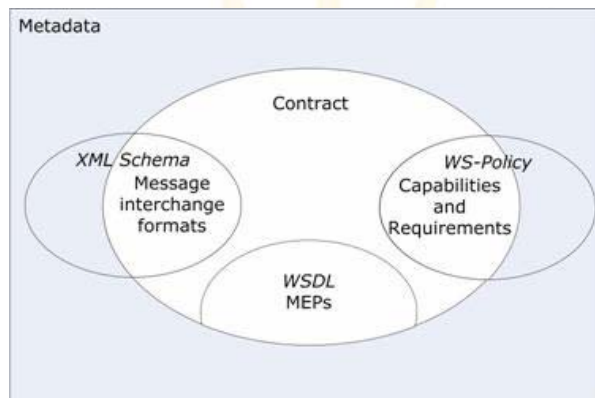
- Service Implementation Details
 - Programming model
 - Object references
 - In-memory representations
- Examples
 - Exposing Java Classes
 - Serialization of Java Objects

Copyright © 2005, ZapThink, LLC



zapthink

Contract Metadata



Source: W3 - <http://www.w3.org/2004/08/ws-csdlms-20040903>

Copyright © 2005, ZapThink, LLC



zapthink

Contract-First Development



- Service contracts specify required functionality to IT and provided functionality to the business
- Service model represents the clearinghouse for information about IT environment
- Contracts go beyond WSDL:
 - Usage policies
 - Security policies
 - Consumer delivery contracts
 - Service-level agreements, etc.

Copyright © 2005, ZapThink, LLC



zapthink

Identifying Policies

- **Security**
 - What Security is required and how?
- **Quality of Service**
 - What guarantees of quality will Service provide to Consumer?
- **Transactional requirements**
 - Transaction support? ACID? Compensating? None?
- **Commercial Policy**
 - Does someone need to pay for this Service?
 - Auditing and logging of Services
- **SLA policy**
 - How long should this service be available?
 - How reliable is it / should it be?
- **Matching versions to specific invocations**
 - Not all contracts are possible for all consumers!
- **Semantic / Payload policies**
 - UTF-8 or other version support?
 - Which languages?
 - Which document types?



Copyright © 2005, ZapThink, LLC



Steps for Creating Policies

- **Policy inventory**
- **Decide on level of granularity**
 - Message-level policies/Data-driven policies
 - Corporate-level policies/Governance & Security policies
 - Service-level policies/Performance, Security, Mediation, Transports, Response time / QoS / SLA
- **Encode metadata in human-readable format**
- **Translate policy into system-understandable format**
 - Pick an XML-based standard method
 - Develop your own XML-based spec
 - Encode directly into policy enforcement system
- **Figure out how to enforce policies in practice**
 - Policy mediation system
 - Registry-based system
- **Identify long-term policy maintenance**
 - Federated or centralized policy management?
 - How is policy going to be controlled?



Current Contract Metadata Spec List

Table 1. Some Existing Languages for the Specification of Contracts for Web Services

Language	Contract Category	Contract Type	Functionality			Quality		Infrastructure	
			Identity	Behavioral	Compositional	QoS	Pricing	Security	Management
WSDL			+					+	
BPEL4WS					+	+			
WS-CDL					+	+			
WS-Policy				+					+
WSLA							+	+	+
WSOL							+	+	+
OWL-S			+	+	+	+			

Source: Tosic & Pagurek – Technical Report 635, Department of Computer Science, The University of Western Ontario, October 20, 2004

Copyright © 2005, ZapThink, LLC



zapthink

Policy as Organizational Issue

- Remember, policies are *business* concepts
- Governance is the way the *business* communicates & manages policies
- SOA helps *automate* policy activities, but...

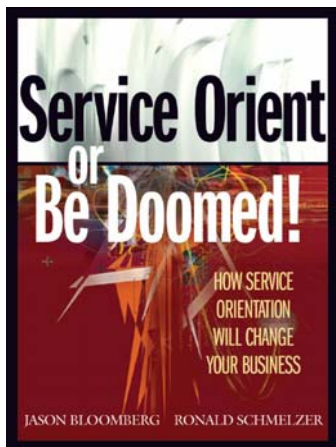
SOA means changing how people work – and such change is far more difficult than technology change

Copyright © 2005, ZapThink, LLC



zapthink

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!

Copyright © 2005, ZapThink, LLC

Thank You!



ZapThink is an industry analysis firm focused exclusively on XML, Web Services, and Service-Oriented Architecture.



Jason Bloomberg
jbloomberg@zapthink.com



Ronald Schmelzer
rschmelzer@zapthink.com

Copyright © 2005, ZapThink, LLC



Photos © Lisa Polucci