Top 10 SOA Best Practices

Eugene Miklovich, DooBee Systems Inc.  
TechED, Austin TX  
May 4th, 2016
Agenda

• Technical Environment
• Application Architecture
• Sample Project
• SOA 10 Best Practices
• Miscellaneous Topics, Performance
Technical Environment

Running on Mainframe
•  Natural & Natural Construct
•  Adabas & VSAM
•  Adabas SQL Gateway
•  COBOL/CICS
•  67 Services

Running on Windows Servers
•  VB.Net, C#, Java
•  SQL Server
•  EntireX
•  webMethods ESB
•  22 Services
Application Software Architecture
A Typical Project

- Service-enables existing Natural programs on the mainframe

Users of newer Underwriting System ➔
- perform same functions as mainframe users
- using same underlying code (business logic)
- view/update mainframe data
- without leaving UW System
Top 10 SOA Best Practices

• Practices are NOT all technical.
  – Process is involved as well
• Top 10 are subjective based on our experiences
  – More could be added, replaced.
• These have been experienced at other sites also
1. Organization Standard

- Embrace SOA throughout the organization
  - Services are an Organizations Asset. Leverage them.
- Build a Governance Committee of high level IT managers
  - Direct reports to CIO only.
  - This is a guidance committee not technical
- Form Sub-committees to develop standards
  - This is where the technical experts are selected
- Benefits:
  - Consistent modular development - Productivity
  - Better Communication - Elimination of Bottlenecks/Delays
  - Team Approach – More Minds are better
2. N-tier Architecture

- Separate to a minimum of 3 layers
  - Presentation, Business, Data
  - Other Layers could be: Navigation, Security, …
    - Business Objects versus Data Objects, e.g. Code Tables
- Do NOT Cross layers
  - **NO** User Interface in Business Layer
  - **NO** Data Access in Presentation Layer
- EntireX is NOT a Layer
  - it is the glue between layers
3. Single Source of Truth

- Only one module maintains master set of the data
  - Multiple Browse/Query Objects are okay
- Eliminates replication of business/integrity rules.
- Expose this as a service to other layer(s).
- Replication for performance is okay.
  - But no updating, call the service to do updates
4. Replicate don’t Extract

- Replication gives real time updates
- Replication is more balanced load on the system
- Replication does not require taking things out of service
- Replication permits offload of some workload
- Replication can provide failsafe backup
- Software AG has a product ADABAS EVENT Replicator
5. Restart vs Rerun

• Rerun requires taking things out of service
  • During Restore for failure
  • During run since it may have to restore!!!!!

• Restart is not difficult
  • Restart DATA – NATURAL GET TRANSACTION DATA

• Restart is efficient – no work redone
• Restart permits 7x24 availability
• Design all processes to run in parallel (i.e. no exclusive use)
6. Standardize

- Browser
  - Thin Client
- XML Interface to Services (WEB Services)
  - Outside the Application (as XML imposes overhead)
- Wrappers around existing programs
  - Standardize message header for debug…
  - Allows for evolution as existing interface supported
7. Use ERD's

- Entities are the basic building blocks of an organization
  - They do not typically change much over time
  - They are used in different ways and process however
- Use ERD to identify Basic Data Services (or Data Model)
  - Inventory (Centrasite) to avoid duplication
- Result is a collection of services ready to go.
Entity Relationship Diagrams
8. Build Scalable Services

- Make Services Stateless
  - No memory
  - No transactions left uncompleted
- Use Optimistic Locking
  - Pass back to Client (Dialog) Restricted field(s) e.g. timestamp
  - ADABAS has new System fields incl. timestamp
  - DO NOT use NATURAL T format – insufficient granularity
- Make Services Object Oriented
  - Data Services handle CRUD functions
  - CONSTRUCT has OBJECT-MAINT model
9. Reuse Existing Services

- Avoid duplication
- Use Wrappers to avoid modification
- Refactor if necessary
  - Remove any UI from Service
  - NATURAL Engineer can help
10. TEST TEST & TEST

- TEST at many levels
- TEST Services Standalone
  - Generate Driver and test locally
- TEST communication
  - Ping from client
  - Test middleware communication
- TEST XML
  - Use SOAP Testers – Test WEB Service
- TEST from client
Summary of Top 10

1. Embrace SOA throughout the organization
2. Use an n-tier application architecture
3. Maintain data in one source only
4. Replace Extracts with Replication
5. All Processes Restartable, NOT Rerunnable
6. Standardize
7. Use ERD's to build basic DATA services
8. Build Scalable Services
9. Reuse
10. Test
Honorable Mention

• Training
  • Tools usage
  • New Paradigms (e.g. Optimistic Holds, …)

• Centre of Excellence
  • Support
  • Samples
Miscellaneous Topics/Issues

• Use Existing RPC Servers
  • E.g. NATURAL RPC Server
  • EntireX RPC Servers
  • EntireX XML Servers
• VSAM may require RPC under CICS
• Security
  • Trusted Account or Pass thru
Performance Considerations

• Use Stateless Objects
  • Promotes Scalability, No Session Locking
• Use RPC Servers
  • Reduces Server Conflict
  • Easier to initiate multiple instances
• Reduce Message Traffic
  • Browse Objects
  • However Moderate Message Sizes
• Avoid extra calls
  • Browse Objects, Load Tables in Memory
Summary

• SOA involves tech and mgmt practices
• Up front Investment Pays Back over Time
• Re-engineering may be required
• Benefits are at Organization level
  • Not for a single Project
• New Paradigm of Programming
  • Education Required
  • Architecture & Process Required
THANK YOU

Questions?

gene@doobee.ca
916-202-7047
Canton, Michigan, USA