SOA Adoption Trends in 2008

Jason Bloomberg
Managing Partner
ZapThink LLC
SOA = Best Practices

- You don’t have to follow them all
- There’s no rule how many you must follow before you can say you’re “doing SOA”
- Key best practice: take an iterative approach

The Right Tool for the Job

The Wrong Question!

**SOA is great. How do I sell it to the business?**

instead of...

**Here are our problems. How best to solve them?**
SOA: Paradigm Shift?

- SOA is more evolutionary than revolutionary
- Leverages many established best practices

But...

- As fundamental a change as client/server or the rise of the Internet

When *Not* to Apply SOA

- When business requirements are stable
- When the IT environment is homogeneous
- When the business has sufficient visibility based on current tools
- When a particular performance requirement calls for efficiency over flexibility

**SOA success means applying SOA where needed, but if it ain’t broke, don’t fix it!**
SOA Growing Pains

- Does the business see the value?
- Are the architects working on the right problems?
- Is IT management investing properly?
- Are you letting vendors drive?

The Problems with “VDA”

“Vendor-Driven Architecture”

- Vendor “SOA Certification” Programs
  - Always product-specific, not SOA-specific

- Vendors who design & build your SOA
  - Always start with their stack

- “One stop shopping” for SOA
  - Doesn’t give you best practices
There’s No Such Thing as a SOA Wizard!

- Click...click...click...done! You now have a SOA!
- Will never happen because...
  - SOA best practices are too general
  - Each organization has a different environment, both technical and cultural

The architect’s answer is usually “it depends”

SOA by Any Name

- “SOA” is too “techie” for the business
- SOA is a broad set of best practices
- Many SOA best practices build on existing practices

Doing it Right More Important than Calling it SOA
SOA Pitfalls

- Unclear business drivers
- Allowing a vendor to drive the initiative
- Confusing SOA and Web Services
- Too few qualified architects
- Lack of proper, early governance
- Unqualified consultants
- “Good money after bad” fallacy

Good Money after Bad...

- I spent money on a proprietary vendor solution, so now I need to make it work!
- We built inflexible EJB Services or .NET Services, so how to I make them flexible?
- We spent big money with that big consulting firm on our SOA initiative, but we don’t have anything to show for it!
Enterprise Architecture Challenges

- Many organizations have a chasm between the traditional EA crowd and the SOA team
- EA has morphed from an approach for the betterment of corporate IT to a management practice, hence resistant to change
- The person that must understand & implement SOA should be the EA in charge

EA Challenges: The Risk of SOA

- Issue: “add-not-change” approach to architecture
- Adding applications, directories, and databases to an existing architecture is easy and risk-adverse
- Changing architectures around systemic notions such as SOA is difficult and risky
- Cultures often have “you fail, you're fired” approach, vs. “let’s try new things and seek improvement”
EA Challenges: The Role of the EA

• Easier to stay high level than do actual work!
• Drawing diagrams, doing presentations, and writing reports is much easier than actually going out and making real changes with real benefits

Questions to Ask Yourself

✓ Have you compared your current architecture with best practices in your industry, looking to spot issues, such as the architecture’s inability to align with the business?
✓ Have you conducted ROI analysis of SOA and reported it to management?
✓ Do you have a complete Service, semantic, & process-level understanding of your enterprise?
✓ Do you have a common abstract model for key business elements, such as customers, sales, inventory, transactions, etc.?
✓ Are systems already well integrated and do they communicate in real time where needed?
✓ Can you change your architecture to accommodate business change at the speed required by management and the marketplace?
The Real Challenge: People, Change and Fear

- People are inherently resistant to change
- People consider job security, authority and responsibility when asked to share
- Fear is the strongest emotion of all!

The Distributed Computing Pendulum

Mainframe  Web  Client/Server  SOA

Centralized ↔ Decentralized
User Empowerment vs. IT Control

- IT charged with empowering users
  - How to avoid policy breaches?
- Centralization of IT capabilities running into roadblocks
  - How to decentralize IT responsibility without leading to redundant or incompatible capabilities?

User Empowerment → Business Empowerment

- User empowerment:
  - One person sitting in front of their computer
  - **Personalization is solitary**
- Business empowerment:
  - IT supporting business processes as the business team requires
  - **Personalization becomes collaborative**
Rethinking the Application

- SOA abstracts existing capabilities
  - Breaks down existing application silos & leverages legacy assets

- Architecture guides composition of Services

- SOA empowers the business
  - The business drives business process

Business Process the Old Way...

- People plugged into rigid processes
- Inflexible & brittle
Business Process the Service-Oriented Way...

- IT resources (among other resources) available to the business as needed
- Business users create Service-Oriented Business Applications by composing Services on the fly

Service-Oriented Process

- Business processes in the context of SOA
- SOA as business empowerment tool
- Integration becomes a side effect of implementing Service-Oriented Processes thru Service composition
The Rise of the Mashup

- Mashup = a flexible composition of Services within a rich user interface environment
- In essence, a Mashup is an interface to a Service-Oriented Business Application

Empower the Business?

- Without management and governance, mashups will never be appropriate in an enterprise environment
- How to empower the business in the spirit of the mashup, but maintain necessary control?

*Governance & SOA: keys to the “Enterprise Mashup”*
What’s New about Mashups?

- Using the application includes creating and configuring the application
“Use Case” or “Killer App” for SOA

- Enterprise Mashups are driving SOA adoption in many organizations
  - Put a visual face on SOA
  - Emphasize the business empowerment benefit
  - Driven by business process-centric motivations
Without Governance, Mashups are **Dangerous**

- Mashups enable *unpredictable* SOBAs
- **Risks:**
  - Confidentiality breaches
  - Unauthorized capabilities
  - Fraud

Without SOA, Mashups are **Toys**

- **Loose coupling of underlying Services essential for reliability & agility**
- **How can Google & Amazon update their Services?**
Governance: The Key to Business Empowerment

- Governance: creating, communicating, & enforcing policies
- SOA enables the formalization of policies as metadata

Practical SOA requires Governance
Effective Governance Requires SOA

Enterprise Mashups require Both!

Thank You!

ZapThink is an industry advisory & analysis firm focused exclusively on Service-Oriented Architecture and Enterprise Web 2.0.

Jason Bloomberg
jbloomberg@zapthink.com
GE Capital Solutions Europe
SOA Case study

April 2008
Nicolas FARGES, Technical Architect

Summary

GE Story
GE Capital Solutions Europe
Commercial Strategy
IT system overview

The SOA project
  > Integration requirements
  > Initial figures
  > Design approach
  > Technology stack
  > SOA Data approach
  > “Company ID check” architecture
  > “Deal synchronization” architecture
  > SOA instrumentation
  > Benefits for the business

Lessons learned
The GE Story

GE’s reputation and diversity equals strength. 130 years of business excellence, make us one of the most-respected and reliable companies in the world.

1878
Thomas Edison founds the Edison Electric Company

1882
Edison generates electricity from America’s first power station in New York City

1892
GE Credit Corporation founded - The Electric Bond and Share Co.

1905
GE’s finance arm founded - The Electric Bond and Share Co.

1912
GE builds the United States’ first jet engine

1927
GE brings television into American homes

1932
GE's finance arm founded - The Electric Bond and Share Co.

1938
GE scientists develop the MRI scanner

1941
GE buys SoGen Avis and launches GE Fleet Services

1951
European Equipment Finance is formed and over the next 15 years builds to become a strong European business

1983
GE acquires Transamerica's commercial distribution business

2003
GE acquires Transamerica's commercial distribution business

GE Capital Solutions Europe

Equipment Financing
Leasing and financial solutions to commercial customers across Europe including manufacturers, distributors, resellers, dealers and end users

Fleet Financing
Vehicle leasing and fleet management services provider offering solutions for the procurement, operation and sale of vehicles

Inventory Financing
Specialist financing solutions to facilitate the distribution and sale of assets for manufacturers, distributors and dealers
GE CSE Commercial Strategy

Objective: GROWTH

Strategy: Internal surveys highlighted that implementing CRM with Siebel is a key GROWTH ENABLER
  > Understand our market, our customers and our partners
  > Align resources to address
  > Manage contacts
  > Manage customer feedback
  > Manage Sales and risk business process
  > Standardize Risk and Sales process across Europe

Shift from a transaction based approach to a customer based approach

IT system overview

Main systems
  > Country based Origination systems
  > Country based Backend systems
  > Pan European decision engine
  > Pan European Document Management system

"SOA"
  > Loads of point to point interfaces resulting from rapid growth
  > No real SOA program

European IT architecture Team main focus
  > DR, HA and performance
  > MW stack consolidation
  > ...

SOA requirements

Deal document push
- Siebel to FileNet
- near real time, ~100 Txs/day

Target Search
- Siebel to country specific decision engines
- Synchronous, ~10,000 Txs/day

Deal synchronization
- Siebel to country specific origination system
- near real time, ~50,000 Txs/day
Project figures

Siebel 7.8 CRM project

SOA Sub project
> Team size: 1 SOA Business Analyst, 2 FTE architects + 3 Off shore developers
> Duration: 6 months

Design approach

Best Practices implemented through “stealth SOA”
1. Service orientation
2. Data standardization
3. No business logic in the bus
4. Asynchronous data flows
5. Self service integration bus
   - Online audit service
   - Online routing/transformation rule service
   - Online logging service
   - Real time alert service

Some architecture tradeoffs were required
Technology stack selected

- NO “ESB” product
- XML for Data (+FLF for some legacy)
- XSD for Meta Data
- JEE as deployment platform
- Various Open source Java frameworks (Spring, Ibatis, XMLBeans, Apache commons)
- SOAP/WSDL for synchronous services
- XML/MOM for asynchronous services

SOA Data approach

Pan European Data structure design

Business objects:
- Account, Deal/Opportunity, Partner, Decision, Asset, Proposal, Reason... etc

11 systems over 7 countries in scope

14 iterations over 12 weeks to reach a consensus
SOA Data approach

Country specific “Reference data” items are transformed inside the SOA
  > Mapped to European level classification at best
  > Granularity gap issues
  > Semantic gap issues

Core “Persistent data” items transfer and matching strategy have been managed ad-hoc
  > Manual processes
  > Custom matching algorithm
  > ...

Some business Teams are pushing for Data ownership and standardization
  > Opportunity for incremental MDM

Target Search

SIEBEL 7.8

Pan European Decision Engine

Italian company ID check
Spanish Company ID check
Nordic Company ID check
Deal synchronization

SIEBEL 7.8

Integration bus

Validation service  LOV mapping service  Logging service  Alert service

Mapping rules logging

Italian origination system  Spanish origination system  Nordic origination system

XML/MQ

"Self service" integration Bus
Business benefits

Sales quotations...
> “That allows us to approach our clients with a better and more targeted approach and more efficiently”
> “It’s efficient and it’s organised and because of that I think it ensures that the right clients are getting the right contact”

CRM is gaining more adoption
The CRM culture shift will take time

The Data Quality is the visible part of SOA Iceberg
Incremental MDM is an approach

Lessons learned

Strategy
> Data Quality was a key issue
  > TX rejection at deployment: ~15%
  > TX rejection now: ~5%
> “Stealth SOA” is not a long term strategy

Business Analysis
> Skills for SOA low level business analysis are rare and precious

Technical implementation
> MOM helped to manage scalability gap
> Clean separation of business and integration logic is key
> Canonical message structure is key
> Provide technical guidance to application teams
> Instrument your SOA (alert, logging, audit, wikis…etc)
> MOM + Java EE + XML/XSD + OSS frameworks can address tactical SOA to a certain extent

Project management
> Invest in SOA Service unit testing automation
> Design phase took ~60% of total time, with frequent changes

SOA and off shoring
> The model must be carefully selected
Q/A
Achieving Enterprise Scale Service Orientation

Roy Varughese  
Chief Architect (Global Clients)  
ABN AMRO Bank

Practical SOA Conference: UK  
25th April 2008

AGENDA

Service Orientation through the Consumer’s eyes  
The SOA Blindspot  
Thinking “Enterprise-Scale SOA”  
Case Study – “Client Lifecycle Management”  
Lessons & Insights
Service Orientation through the Eyes of a Business Consumer

“Business enablement with minimum IT development”

“Time-to-market & leveraging external services”

“On the road to business process automation”

“Approach to build business solutions that are responsive to changes”

“Unbundling production for delivery across Business Unit “boundaries”

Understanding the Service-Orientation Blind-spot

Missing from the Business’ vision: “What is the business value of Data & Application services?”

Missing from Vendors’ vision: “How can business inter-operate at a data & service collaboration level”

Missing from IT vision: “How does SOA help to forge and implement a business strategy?”

Missing Conditions for SOA to Thrive:
- Reference points for “Convergence”
- Data Service Standards
- SOA Compliance/Governance “How-to”
- Business data/ information/intelligence architecture
- Mechanisms for “Corporate diffusion” of
  - Total Cost Models (related to SOA)
  - S-O Best Practice (peers & community)
  - Hybrid Skill Sets (covering the architecture continuum)
Top TEN Challenges for Enterprise Scale Service Orientation

- Articulating value of SOA against business budgets (short ROI horizons)
- Planning SOA as a multi-year transformation (lack of Return-of-Asset models)
- Typical CIO portfolios have "limited opportunities" (80% “Business-as-usual”)
- Large-scale outsourcing of IT development projects (dilutes & neuters SOA)
- Patchy support of SOA at the execution level (project & portfolio managers’ ignorance)
- Inadequate investment in SOA Infrastructure (unable to leverage past successes)
- Previous over-selling & under-delivering SOA (lowered expectations)
- Who develops and drives SOA maturity? (“continuous re-invention”)
- Lack of “pragmatic competency” in SOA (and death of internal talent development)
- Lack of enterprise-scale architecture perspective (hybrid bus-tec architects)

Elements of Strategy to Drive Enterprise Scale Service Orientation

- Reposition Service Orientation within a value proposition that forges business strategy
- Leverage (EA) process approach catalytic role in driving enterprise-scale innovation

Use “Cognitive” Feedback measurable KPI’s, providing multi-cycle financial analytics

SO in underpinning (any) corporate initiatives for a “Lean-Agile-Sustainable” Enterprise

FOCUS OF THIS PRESENTATION

- CREATING COHERENCE WITH THE BUSINESS USING CAPABILITY MAPS
- WORK THROUGH A CASE STUDY (Client Lifecycle Management Platform)
Creating a Capability Map

Applying an analysis methodology to define a hierarchical functional model (tree) of the enterprise.

Template defining key principles and operational parameters needed for all execution contexts of a business strategy.

A capability map defines **WHAT** is needed to execute an end-to-end business strategy.

**Success Factors**
- Enterprise awareness and general view & feedback access
- Joint Ownership by top CXO community as stakeholders
- Formal governance & refresh process (periodic)

**Success Factors**
- Business ownership of OM template & its maintenance
- Event driven refresh process & ownership
- IT staying in the change management loop & understanding OM’s nuances

**Success Factors:**
- Underwritten by a sponsoring CXO
- 100% buy-in from Operations, Business & IT Management
- Fit-for-use as template in Requirements and Design Lifecycle

---

From a Capability Map to “Service-Clusters”

1. **Base Capability Map**
2. **Business Operating Model Analysis**
3. **Market & Vendor Analysis**
4. **Client-type Sensitivity?**
   - YES
5. **IT Operating Model Principles**

**SERVICE CLUSTERS**
Step 5: Applying CLM Operating Model Principles := Service Clusters

Case Study:

All Client Segments
Ext. Provider Management Services

---

Applying IT Strategy & IT Operating Model Principles := Service Platforms

Corporate IT Strategy defines a number principles, such as:
1. Buy-not-build
2. Re-Use in-house business infrastructures
3. Use “approved development partners”
4. Use known SO Pattern solutions
   A. Glue
   B. Wrap
   C. Re-factor
   D. Procure (as a Service)

Service Orientation Spectrum
Oriented Enabled Capable Conformant

---
Support of Non-Functional Aspects of the CLM Business Domain

**CLM Operations**
- Global Policy Management – *different countries have compliance differences*
- Global Domain Management – corporate KPI and KRI, feedback and management
- Global Reference Data Management – “business constants” critical to compliance processes
- Inter-process Management – *integration with related processes - Operations, Credit, CRM, Legal*
- Internal Service Level Management – duration & throughput agreements with internal processes

**Programme Execution**
- Multi-Projects Management – “Real time Radar” on Business & IT projects across 52 countries
- Global Platforms Migration – *Enforced and Planned migration of CLM process platforms ~150*
- Global Organizational Management – *Enforced organizational transformation*

**Domain Management Tools**
- Workflow Design & Modelling
- CLM Services Administration
- RAF Design & Modelling
- Project Dashboards
- CLM Process Automation
- CLM Ad Hoc Reporting
- Data & Document QA

**Convergence Tools**
- Data & Application Integration
- Watch List Feeds Reconciliation
- Data & Document Migration

**Information Interchange**
- CLM Data Services

**Country-level Security & Policy**

**Inter-process Workflow**
- Client, Product & CLM Static Data

**Case Study:***

**A Service Platform – “S-O-ness”, Service Classes & Use Case Scenarios**

Service Classes mapped to Business Use Case Scenarios

### Commercial Segment Platform

**CLM Core Service Classes:**
1. User Services
2. Admin Services
3. RB Services
4. Papering Services
5. Reporting Services

### Service-Oriented-ness Index
- Oriented
- Enabled
- Capable
- Conformant

### Service Platforms:

#### 1. User Services
- **Function:**
- **Application:**
- **Technology:**

#### 2. Admin Services
- **Function:**
- **Application:**
- **Technology:**

#### 3. RB Services
- **Function:**
- **Application:**
- **Technology:**

#### 4. Papering Services
- **Function:**
- **Application:**
- **Technology:**

#### 5. Reporting Services
- **Function:**
- **Application:**
- **Technology:**
```
“Technical SOA” – Service Bus, Interfaces, Information Interchange

- Instance(s) of CLM Core & Value Added Service Platforms
- Consumer Segment (US)
- Consumer Segment (NL)
- Information Interchange Services
- Doc Services
- BI Services
- Client, Product & CLM Static Services
- Inter-process Workflow Services
- Ext. Services

Case Study:

Lessons & Insights (from Case Study)

- “One SOA” is not viable in large, politically complex environments & programmes. Instead work to
  - Maximize uplift along a “service oriented-ness” spectrum.
  - Leverage business consumers’ perspective of “service orientation”.

- In complex programmes it is impossible (or too costly) to map out all requirements before a project starts. Instead work on multiple “business trajectories” and a roadmap employing service orientation to remain agile as business changes.

- GO BEYOND “business – IT alignment”: TO implementing principles that forge together business and IT strategy (in the whole lifecycle from design, build and roll-out):
  - Understand the nuances of the business strategy (keep up as it changes)
  - Employ the simplest technology (make choices based on evidence & maths)
  - Consolidate-and-Refine over multiple cycles of delivery (ring-fence some budget)
  - Make the solution modular from a business sense (deep understanding of business OM)
  - Invest in Usability design for system to sell itself to users (think commercial)
  - Design to empower users to influence & drive evolution (think 99% adoption)
```
Other Takeaways

- Enterprise Architecture, as a lifecycle and process discipline, is key to sustaining SOA success on any scale (i.e. across projects, across boundaries, across budget-periods)

- A CXO supported capability model is a major architectural asset and key to keep SOA anchored in the business mindset

- Building value propositions for SO around technology benefits has limited scope – instead think of how SOA is part of an eco-system that enables
  - Data services/ information flow/exchange possible
  - Event flow and event-oriented architecture
  - Web-oriented architecture (lowered cost of delivery)
  - Process oriented architecture (business automation)
Achieving Enterprise Scale Service Orientation

Roy Varughese
*Chief Architect (Global Clients)*
*ABN AMRO Bank*

Practical SOA Conference: UK
25th April 2008

AGENDA

Service Orientation through the Consumer’s eyes
The SOA Blindspot
Thinking “Enterprise-Scale SOA”
Case Study – “Client Lifecycle Management”
Lessons & Insights
Service Orientation through the Eyes of a Business Consumer

“Business enablement with minimum IT development”

“Time-to-market & leveraging external services”

“On the road to business process automation”

“Approach to build business solutions that are responsive to changes”

“Unbundling production for delivery across Business Unit “boundaries”

Understanding the Service-Orientation Blind-spot

Missing from the Business’ vision: “What is the business value of Data & Application services?”

Missing from Vendors’ vision: “How can business inter-operate at a data & service collaboration level”

Missing Conditions for SOA to Thrive:
• Reference points for “Convergence”
• Data Service Standards
• SOA Compliance/Governance “How-to”
• Business data/information/intelligence architecture
• Mechanisms for “Corporate diffusion” of
  • Total Cost Models (related to SOA)
  • S-O Best Practice (peers & community)
  • Hybrid Skill Sets (covering the architecture continuum)

Missing from IT vision: “How does SOA help to forge and implement a business strategy?”
Top TEN Challenges for Enterprise Scale Service Orientation

- Articulating value of SOA against business budgets (short ROI horizons)
- Planning SOA as a multi-year transformation (lack of Return-of-Asset models)
- Typical CIO portfolios have "limited opportunities" (80% “Business-as-usual”)
- Large-scale outsourcing of IT development projects (dilutes & neuters SOA)
- Patchy support of SOA at the execution level (project & portfolio managers' ignorance)
- Inadequate investment in SOA Infrastructure (unable to leverage past successes)
- Previous over-selling & under-delivering SOA (lowered expectations)
- Who develops and drives SOA maturity? (“continuous re-invention”)
- Lack of “pragmatic competency” in SOA (and death of internal talent development)
- Lack of enterprise-scale architecture perspective (hybrid bus-tec architects)

Elements of Strategy to Drive Enterprise Scale Service Orientation

- Reposition Service Orientation within a value proposition that forges business strategy
- Leverage (EA) process approach catalytic role in driving enterprise-scale innovation
- Use “Cognitive” Feedback measurable KPI’s, providing multi-cycle financial analytics
- SO in underpinning (any) corporate initiatives for a “Lean-Agile-Sustainable” Enterprise

FOCUS OF THIS PRESENTATION
- CREATING COHERENCE WITH THE BUSINESS USING CAPABILITY MAPS
- WORK THROUGH A CASE STUDY (Client Lifecycle Management Platform)
Creating a Capability Map

Applying an analysis methodology to define a hierarchical functional model (tree) of the enterprise.

Template defining key principles and operational parameters needed for all execution contexts of a business strategy.

A capability map defines **WHAT** is needed to execute an end-to-end business strategy.

**Success Factors**
- Enterprise awareness and general view & feedback access
- Joint Ownership by top CXO community as stakeholders
- Formal governance & refresh process (periodic)

**Success Factors**
- Business ownership of OM template & its maintenance
- Event driven refresh process & ownership
- IT staying in the change management loop & understanding OM’s nuances

Success Factors:
- Underwritten by a sponsoring CXO
- 100% buy-in from Operations, Business & IT Management
- Fit-for-use as template in Requirements and Design Lifecycle

---

From a Capability Map to “Service-Clusters”

1. Base Capability Map
2. Business Operating Model Analysis
3. Market & Vendor Analysis
5. IT Operating Model Principles

**SERVICE CLUSTERS**
Step 5: Applying CLM Operating Model Principles := Service Clusters

- Commercial Segment
  - CLM Core & Value Added Services
  - Anti-Money Laundering Services
- Consumer Segment
  - CLM Core & Value Added Services
  - Commercial & Consumer Client Segments
  - Business Intelligence Services
  - Commercial & Consumer Client Segments
  - Client Documentation Services

Case Study:
- All Client Segments
- Ext. Provider Management Services

Applying IT Strategy & IT Operating Model Principles := Service Platforms

Corporate IT Strategy defines a number of principles, such as:
1. Buy-not-build
2. Re-Use in-house business infrastructures
3. Use "approved development partners"
4. Use known SO Pattern solutions
   A. Glue
   B. Wrap
   C. Re-factor
   D. Procure (as a Service)

Service Orientation Spectrum
- Oriented
- Enabled
- Capable
- Conformant

Case Study:
- All Client Segments
- Business Intelligence Services
- All Client Segments
- Client Documentation Services
- Ext. Provider Management Services
A Service Platform – “S-O-ness”, Service Classes & Use Case Scenarios

Service- Oriented-ness Index

Oriented  Enabled  Capable  Conformant

Service Classes mapped to Business Use Case Scenarios

Commercial Segment Platform

CLM Core Service Classes:
1. User Services
2. Admin Services
3. RB Services
4. Parning Services
5. Reporting Services

Support of Non-Functional Aspects of the CLM Business Domain

CLM Operations
Global Policy Management – different countries have compliance differences
Global Domain Management – corporate KPI and KRI, feedback and management
Global Reference Data Management – “business constants” critical to compliance processes
Inter-process Management – integration with related processes - Operations, Credit, CRM, Legal
Internal Service Level Management – duration & throughput agreements with internal processes

Programme Execution
Multi-Projects Management – “Real time Radar” on Business & IT projects across 52 countries
Global Platforms Migration – Enforced and Planned migration of CLM process platforms ~150
Global Organizational Management – Enforced organizational transformation

Domain Management Tools
Workflow Design & Modelling
CLM Services Administration
RAF Design & Modelling
Project Dashboards
CLM Process Automation
CLM Ad Hoc Reporting
Data & Document QA

Convergence Tools
Data & Application Integration
Watch List Feeds Reconciliation
Data & Document Migration
“Technical SOA” – Service Bus, Interfaces, Information Interchange

Lessons & Insights (from Case Study)

- “One SOA” is not viable in large, politically complex environments & programmes. Instead work to
  - Maximize uplift along a “service oriented-ness” spectrum.
  - Leverage business consumers’ perspective of “service orientation”.

- In complex programmes it is impossible (or too costly) to map out all requirements before a project starts. Instead work on multiple “business trajectories” and a roadmap employing service orientation to remain agile as business changes.

- GO BEYOND “business – IT alignment”: TO implementing principles that forge together business and IT strategy (in the whole lifecycle from design, build and roll-out):
  - Understand the nuances of the business strategy (keep up as it changes)
  - Employ the simplest technology (make choices based on evidence & maths)
  - Consolidate-and-Refine over multiple cycles of delivery (ring-fence some budget)
  - Make the solution modular from a business sense (deep understanding of business OM)
  - Invest in Usability design for system to sell itself to users (think commercial)
  - Design to empower users to influence & drive evolution (think 99% adoption)
Other Takeaways

- Enterprise Architecture, as a lifecycle and process discipline, is key to sustaining SOA success on any scale (i.e. across projects, across boundaries, across budget-periods)

- A CXO supported capability model is a major architectural asset and key to keep SOA anchored in the business mindset

- Building value propositions for SO around technology benefits has limited scope – instead think of how SOA is part of an eco-system that enables
  - Data services/ information flow/exchange possible
  - Event flow and event-oriented architecture
  - Web-oriented architecture (lowered cost of delivery)
  - Process oriented architecture (business automation)
Enterprise SOA Patterns

Francois Lascelles, P. Eng.
Director Client Solutions, Europe
http://www.layer7tech.com

March 2008

Contents

- XML Appliances Explained
- Runtime Governance
- Layer 7 Product
- ESOA Goals
- Loose Coupling
- Enterprise SOA Patterns

March 2008
Enterprise SOA Elements

- Identity, LDAP, MSAD
- UDDI
- PKI, STS
- Proprietary
- Applications
- Management/monitoring
- Services, ESB
- Message broker
- Virus scanning

What is an XML appliance?

- accelerator/co-processor deployment
- gateway/firewall deployment
Why an XML appliance?

Delegate common or expensive XML related tasks

- Validation as a service
- Security as a service
- Transformation as a service

Message level intermediary between services and requesters

External Application Consumers

Internal Application Consumers

Services
Policy Enforcement Point

Runtime Governance - Policy Enforcement Point

PEP validates policy compliance, applies security decorations, transformations, records statistics, etc.

Policies defined by an administrator.

Policies become effective independently of the actual services.

Services

Delegation of security

PEP allows for sophisticated, dynamic and flexible security requirements.

No server-side agent

"Last mile" security to protected service is kept simple and static.
Policy Enforcement Point

Early detection of problematic messages

- Access control rules
- Schema validation
- Threat protection
- SLA
- Routing rules
- Transformation
- Other

Delegate validation and logic to PEP in order to minimize downstream errors

SecureSpan Gateway Policy Capabilities

- Security enforcement (message/transport level)
- PKI, token issuing, XML signature, XML encryption
- Content validation (XSD, pattern evaluation)
- Service level enforcement (SLA, throttling)
- Transformation, service virtualization
- Sophisticated routing rules (content based, identity based, etc)
- Identity management, access control, identity federation
- Transport mapping (HTTP, JMS, FTP)
- Much more...
Layer 7 SecureSpan XML VPN

Application

XML VPN downloads
WS-Policy document applicable to service being invoked and
decorates outgoing messages on behalf of requester.

PEP

Services

Enterprise SOA Goals

Agility
Reusability
Governance
Security
Performance

Loose coupling
Interoperability
Standard compliance
Loose coupling by example

• Dynamic Binding Based on Content

One service exposed to the requester can be bound to several different endpoints based on arbitrary content queried at runtime.

New endpoints can be added without service interruption.

1. Request content

2. Predefined xpath
   \[s:Body/tvs:browse/tvs:provider\]

3. Choose endpoint based on xpath result

4. Transform request to comply with particular provider (XSLT)

Transparent aggregation of provider channels

Channel provider connectors

Loose coupling by example

• Dynamic Routing Based on Attribute (contract)

Routing based on requester’s contract looked up remotely or managed at the XML appliance through group memberships.

1. Request content

2. Lookup ‘foocontract’ attribute for authenticated requester

3. Route requests to appropriate version

high throughput, low latency, full version for subscribers

limited demo version for non-subscribers
• **Service Virtualization**

Same service is virtualized differently for provisioning and for consumption purposes. Each virtual version has its own WSDL subset and only certain operations are enabled based on requester.

**Requests and responses can be transformed to accommodate older versions of clients.**

**Virtual Services**

**Newer Version**

---

**Enterprise SOA Patterns**

**Access control using external directory**

**LDAP based identity service**

**Access control rules in PEP**

**Separation of authentication and authorization.**

**LDAP bind for shared password based authentication or user certificate lookup for proof of possession based authentication.**

**Credential extracted from message or from transport layer**

**Proof of authentication or identity mapping to next step**
Using LDAP attributes for SAML attribute statements

LDAP attribute values are used to produce SAML attributes statements.

SAML issuer embedded in PEP intermediary.

SAML authentication, attributes statements.

SSO involving an XML Gateway and an external identity service

If identity service supports it, session issuing can be done there. Ex Netegrity.

Session could be issued by PEP.

PEP validates session at runtime. This may involve identity service but doesn’t have to.
Enterprise SOA Patterns

SSO mapping

Browser client has session. Ex. Windows integrated, cookie, Kerberos, …

Web app needing to make WS calls

Mapping between two different SSO mechanisms

XML Gateway uses a message level SSO mechanism on behalf of users when consuming web services. Ex. WS Secure Conversation

Enterprise SOA Patterns

Transport mapping

Administrator defines mapping between different transport mechanisms

(no development or deployment task required)

Virtual HTTP binding exposed by policy enforcement point

ESB services only consumable through message queues
UDDI / PEP interactions – design time

PEP management interface enables design time discovery of services published in the UDDI registry as well as loading metadata such as WSDL, XSD, XSLT, etc.

WS-Policies are associated to services using WS Policy Attachment – UDDI Attachment standard.

WS-Policies can also be discovered from the UDDI directory.

UDDI / PEP interactions – runtime

At runtime, the PEP will monitor metadata source for updated and load new versions. PEP administrator must have previously authorized this behavior.

The PEP subscribes for new service notifications and loads them based on rules previously defined (roadmap item).

The PEP dynamically discovers service using UDDI and routes to appropriate endpoint upon message reception (roadmap item).
Identity federation, trust management

- **PEP zone A** authenticates its own identities and issues a proof of authentication to zone B.
- **SAML assertion**
- **Policy administrator** manages trust relationships.
- **PEP zone B** validates signature and authorizes proof of authentication based on rules defined by its administrator.

Management, monitoring, reporting

- WSDM, WSMF, WS-Management, SNMP
- Management system receives runtime notifications and alerts per policies.
- Specialized SOA Management Tool
- Customized traffic data recorded to syslog or file for reporting purposes
- Reporting tool
For more information

http://www.layer7tech.com
A Complete Strategy for Testing WebServices...
euh... SOA

Rix Groenboom
rixg@parasoft.com

Agenda

- Little Quiz about Testing en QA
- QA requirements for SOA
- Practical approach
- Case studies
- Lessons learned
Quiz (1): What is the main Testing Tool?

- In other words, what is the hammer of testing?

Excel!
Quiz (2): What is the size of Software?

- How long is the listing in this (old) phone?

- 3 MLOC of SW
- 50 lines = 25 cm
- 100 = 50 cm
- 200 = 1 m
- 1,000 = 5 m
- 10 kloc = 50 m
- 100 kloc = 500 m
- 1 Mloc = 5 km
- **3 Mloc = 15 km**
- 8 Mloc = MARATHON
Quiz (2): What is the size of Software?

- No wonder we have problems...

Quiz (3): XML Bomb?

bomb.xml
Quiz (3): XML Bomb?

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SOAP-ENV:Envelope [ 
<!ELEMENT SOAP-ENV:Envelope ANY>
<!ATTLIST SOAP-ENV:Envelope entityReference CDATA #IMPLIED>
<!ENTITY x0 "Bomb!"> 
<!ENTITY x1 "&x0;&x0;"> 
<!ENTITY x2 "&x1;&x1;"> 
...
<!ENTITY x20 "&x19;&x19;"> 
<!ENTITY x21 "&x20;&x20;"> 
<!ENTITY x22 "&x21;&x21;"> 
]>
```

Quiz (4): How to test this thing?
Why SOA?

- Business agility
- Speed
- Lower integration costs
- Alignment between business and IT

Business Effectiveness
- Agility, responsiveness to market/competitive dynamics
- Greater process efficiencies
- Deploy resources based on business needs

Cost Efficiency
- Reduced maintenance costs
- Reduce integration costs
- Reduced skills and effort to support business change
- Reduce application redundancy

Reduced Risk
- Higher level of IT quality
- Incremental deployment
- Improved payback times
SOA What’s Different...

- Modular
- Client-less server modules
- External access to modules
- Loose coupling (black box)
- Interoperable
- Intermediary rich environment
- Designed to be useful and usable by other applications
- Useful and usable by other enterprises
- Centrally-managed repository and registry
- Walls between partners blur

SOA Impact to Projects

Currently

People
- New Relationships
  - Business and technology
  - Providers and consumers

Process
- Constant Change
- Business Service Lifecycle
  - Independent dev. cycles
  - Producers and Consumers

Technology
- Interoperability
- Logic abstracted
- Complex

The Challenge

People
- Business Analyst
- Trust
- Partners

Process
- Change based testing
- A new lifecycle
- Continuous quality process

Technology
- Consistency
- Tests need to follow
- Isolation and emulation
SOA Challenges for QA

1. Process Cadence
2. Significant risk
3. Challenge of reuse
4. Properly Addressing Security
5. Organizational impact
6. Unconscious Migration

Process Cadence

- Organizations evolve the development process without QA
- Agility is lost due to lack of education
- Complexity not well understood
- Three aspects to consider:
  - New Service
  - Versioning
  - Process changes
Managing Risk

Consolidation of application or services for mission critical processes increases the risk of failure. More users are impacted.

**Impact of Downtime (Risk)**

- Distributed Applications
- Reuse of Services

---

Promoting Reuse

- Creating an asset that is reusable is easy, promoting reuse is a much different challenge.
- Aside from granularity, reuse is all about trust.
- There is no such thing as a “used car.”

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Point Inspection</th>
<th>Special Financing</th>
<th>Certified Warranty Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrysler</td>
<td>125</td>
<td>Yes</td>
<td>8 years / 80,000 mile Powertrain Limited Warranty, measured from original vehicle in-service date.</td>
</tr>
<tr>
<td>Ford</td>
<td>115</td>
<td>Yes</td>
<td>6 years / 75,000 miles from the In-Service date</td>
</tr>
<tr>
<td>GMC</td>
<td>110+</td>
<td>Yes</td>
<td>3 months / 3,000 miles from the Purchase date</td>
</tr>
<tr>
<td>Lexus</td>
<td>161</td>
<td>Yes</td>
<td>3 years from the Purchase date / 100,000 miles from the In-Service date</td>
</tr>
<tr>
<td>Mercedes-Benz</td>
<td>130+</td>
<td>Yes</td>
<td>12 months from Purchase date / 100,000 miles from the In-Service date</td>
</tr>
<tr>
<td>Toyota</td>
<td>160</td>
<td>Yes</td>
<td>7 years / 100,000 miles Limited Power Train Warranty from date when first sold as new.</td>
</tr>
</tbody>
</table>
Properly Addressing Security

- There is a gap in how WS security is addressed
- “Security is not my problem it’s coming from somewhere else”
- There hasn’t been a big scandal, yet!
- Security is usually bolted-on
- Audits are usually performed too late

Develop
Test
Monitor

Architect
Assumptions

Need to be able to detect vulnerabilities as early as possible.

SOA Impacts IT Roles

Trend 1
Project durations are shorter with higher levels of integration.

Trend 2
"Quality" and the quality process is being promoted higher in the organization

Trend 3
Silos are being broken down into smaller cross-functional teams. Those teams have more distributed team members.

Trend 4
The onus of quality is being distributed in the process. QAs role is split.
With success comes demand for more services.

Every new consumer or provider adds exponential potential for complexity:
- Identity credentials
- Standards
- Message format
- Transport protocol

New versions of a provider or consumer adds complexity.

With SOA there is too much at risk we cannot have a “Save it for Later” quality process.

Futile to stage a “real” SOA environment:
- Systems and services might not be controlled by a single entity
- Environment is complex
- Application layer is a “software unit” that performs a service
- Service must interact with other systems and services
- Assumptions about how components will operate are critical for quality
- Our process is only as good as its weakest link

We need a different process.
Step By Step Approach

Achieving secure, reliable, compliant services requires visibility, trust and control

1. Provide visibility
2. Supply an infrastructure for reuse
3. Promote bottom-up quality
4. Leverage the infrastructure for top-down quality
5. Assist to manage complexity
6. Concentrate on quality process improvement

Visibility

- Promoting Trust must begin early in the process
- When an asset is created the quality process begins
- Consider internal and external, consumers and providers
Infrastructure for Reuse

Bottom-Up Quality

- Code Analysis
- Automated Unit/Regression Testing
- Security - Reliability
- Performance - Maintainability
- Component Unit/Regression Testing
- Verify Scalability and Performance
Top-Down Quality

- Verify Service Description
- Verify Policies
- Test Web Services Infrastructure
- Unit test Service Layer
- Business Process Test
- Scenario Test
- Functional Security Test / Penetration Test
- Regression Test
- Verify Scalability and Performance

Message Layer

- Code Analysis
  - Security / Reliability
  - Performance / Scalability
- Component / Unit / Regression Testing
- Cross-system / Cross-platform testing
- Verify Scalability and Performance

Java

.C/++

Manage Complexity of Environment

- Stub out the service consumer (client) to test the service provider (server)
- Stub out the service provider (server) to test the service consumer (client)
- Stub out both tiers to test a proxy or an intermediary
Improve the Process of Quality

What is needed?
- SOA Quality Visibility
- SOA testing framework
- SOA aware to reduce complexity
- Automated policy enforcement
- Automated business process testing
- Automated scenario testing
- Scriptless load and performance testing

Generic SOA Architecture

- Security Gateway
  - Test gateway policies by driving positive and negative traffic. Security POCs.

- Registry

- Orchestration
  - Automated BPEL testing. Graphical construction of scenarios.
- Test multiple protocols with scenarios to automate test coverage. Emulate endpoints.
- Test cases can leverage QoS data from WSM. Create test cases for SLA violations.

- ESB

- WSM

- Java / .NET App Servers

- Mainframe

- Legacy Adapters

What is needed?
- SOA Quality Visibility
- SOA testing framework
- SOA aware to reduce complexity
- Automated policy enforcement
- Automated business process testing
- Automated scenario testing
- Scriptless load and performance testing

Test via emulations.
OKAY, and now the theory in action!

SOA Testing Summary

- Quality Concerns
  - Security
  - Interoperability
  - Reliability
  - Availability
  - Performance
  - Evolving Standards

- Testing services requires:
  - Client implementation is error-free
  - Server implementation is error-free
  - Client and server interact correctly
  - Business processes execute successfully
Phase 1: Implementation layer testing

- Testing the components that offer the services
  - Security, Reliability, Performance, Maintainability
  - 3GL & 4GL verification: Standard Techniques

- Static analysis and best coding practices
  - Find errors before running dynamic tests
  - Adhere to standards (e.g. security, reliability, testability, maintainability)

- Unit testing
  - Make sure smallest pieces of code behave correctly
    - Check for unexpected inputs
    - Increase test coverage
    - Black box test cases to validate component
Phase 2: Message layer testing

- General architecture overview:

  ![Diagram](image)

  - Message, description, discovery
    - WSDL, XML/SOAP, UDDI
  - Transports/Messaging API
    - HTTP
    - JMS
    - EJB
    - TIBCO Rendezvouz
    - IBM MQ
    - SMTP
    - RMI
### Test Strategy

- 1. Create and manage tests:
  - A) WSDL tests (validation / interoperability)
  - B) Unit tests (technical tests)
  - C) Functional tests (functional / use case / scenario)
  - D) Security tests (non functional)
  - E) Load tests (non functional)

- 2. Automate with regression testing throughout the Web Services lifecycle
  - Version control

- 3. Report to management

### Some real life questions

- **Airline:**
  - Only 1 day a week all the systems in the “SOA” are online at the same time
  - This requires stubbing of the components to have a working test-environment

- **MOT registration application:**
  - End-to-end call took 12 and not 4 seconds
  - Unpeal the onion: Problem in XML translations
Case Study SwissLife

Background

- Pension scheme application
- In house development team (Java)
- Back end services exposed via ESB
- Business process using BPM (JBMP)

- Example of Unit & Scenario testing
SwissLife architecture

Development started with Unit testing

- Similar to unit tests for programming:
  - Test each operation in isolation to ensure the validity of the XML payloads and that it returns the expected response per request
    - Positive conditions (positive tests)
    - Error and faulty conditions (negative tests)
    - Standards compliancy
Functional testers: Scenario based testing

- Functional test to ensure business process scenarios
  - Combine individual services function to realistic business use cases
  - Use real customer data
- Create tests which are representing
  - Expected usage patterns with real data (positive tests)
  - Unexpected usage patterns (negative tests)

Quiz (4): How to test this thing?

- Answer: Bottom up
Case Study UWV

Background

- UWV is the Dutch government organisation for social security benefits
  - 20,000 employees, 3 million customers
- New system for state disability assurance
  - Wet Inkomen en Arbeid (WIA)
- Highly complex (technical and political)
- Outsourced development (tender)
- Example of vendor management and regression testing
Generic Software Stack WIA

Vendor management

- Logical and Functional test cases development iterative process
- WIA responsible for LTC and test coverage
- Vendor responsible for FTC and testdata
- Vendor performs preshipment test to demonstrate quality of delivery
- If A / B type issues, delivery rejected, C & D is accepted
- Then WIA will perform the formal acceptance test
Test Environment WIA

- Analyse
  - Testframes
  - UI / Web
  - Testframe
  - Testframe Editor

- Navigate
  - UWI / WIA
  - Testframe
  - Testframe Editor

Testframe excelsheet

- Contains Logical and Physical testcases in one spreadsheet based on "action words".
- Action word specifies test action and parameter
- Execution of test-cases with SOAtest
Test execution in SOATest

Issue tracking in JIRA and GRS
Main goal: regression testing

- Test execution is automated to run tests on a regular basis, in this case every shipment
- Time and effort on the creating tests, not running them.
- Regression tests are reused in:
  - Maintenance projects
  - Troubleshooting services in production

Case Study: Lufthansa Cargo
Background

- Cargo planning application (shipment DB)
- Offshore development (Java)
- Non SOAP / XML messaging, JMS

- Example of Functional and Load testing

Cargo application and test environment

- Test Topology
### Test management with XLS

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Test Set Name</th>
<th>Test Set Description</th>
<th>Test Case Name</th>
<th>Test Case Structure</th>
<th>Build</th>
<th>Build Result</th>
<th>Test Result</th>
<th>Test Result Comments</th>
<th>Test Result Date</th>
<th>Test Result Time</th>
<th>Test Result User</th>
<th>Test Time</th>
<th>Test Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST Smoke Test</td>
<td>ST Smoke Test</td>
<td>Test a service call with a single root entity and a small number of attributes.</td>
<td>ST Smoke Test</td>
<td>Hierarchical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SG Service Store</td>
<td>SG Service Store</td>
<td>Store a service call with a single root entity and a small number of attributes.</td>
<td>SG Service Store</td>
<td>Hierarchical</td>
<td>Build v2.12</td>
<td>Passed</td>
<td>Passed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SG Service Store</td>
<td>SG Service Store</td>
<td>Store a service call with a single root entity and a small number of attributes.</td>
<td>SG Service Store</td>
<td>Hierarchical</td>
<td>Build v2.13</td>
<td>Passed</td>
<td>Passed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Test design: High test coverage

- **Test Cases**
  - 281 functional test cases
  - Up to 23 service calls per test case
  - Up to 1914 data elements per service call

- **Test Case Structure**
  - Grouped in test suites
  - Test cases structured hierarchically
Implementation and maintenance of tests

Test Execution

- Manual execution of functional tests
  - Randomized AWB Number allows repetitive execution of test cases without DB clean-up
  - HTML-based test reports used to document the test results (attached to ECQM-item)
  - All test cases can be performed on
    - TEST Environment
    - INT Environment
    - PROD Environment
Issue description for reporting to outsourcer

Regression testing

- Automated re-execution (regression):
  - Hierarchical test case structure
  - Can be executed on a test suite basis
  - Automated regression test within 10 min.
  - Keep up with Short build cycles (up to 1 week)
10 minute regression test

- Load and Performance Tests
  - Re use functional test cases
    - Have been validated in previous step
  - Test Scenarios
    - Measurement of throughput per service
    - Measurement of response times per service
    - Measurement of combinations
Load and Performance Tests

- Test Execution
  - Throughput
    - “Virtual Users” used to control message flow
  - Response Times
    - “Hits per Second” used to control message flow
  - Export of HTML-based reports
Load and Performance Tests

- Test Report (Throughput)

![Throughput Graph]

- Test Report (Response Times)

![Response Times Graph]
Load testing results

- Tuning of system components
  - Application
  - Application Server
  - Database
  - RAC Cluster
  - Operating System
  - EAI
- Performance improvements
  - 1,3 msgs/sec vs. 8.1 msgs/sec

Additional usage

- System Monitoring
  - Monitors availability of all SDB services
  - Calls SOAtest in command-line mode
  - Executed every 15 minutes
  - Automated export of reports
    - response messages
    - documents availability over time in Excel document
Lessons Learned

Summary

- SOA testing requires new approach
  - Based on experience with “classic” testing
  - However, requires a new mind set, skills, tools
  - Requires proper testing infrastructure
  - Make use of the glass-box structure of SOA
Summary

- Personal views:
  - Classic problems, but more prominent
  - Due to B2B communication, no humans in the business process anymore
    - Less robustness in the overall process flow
  - Opportunity for interaction between dev & QA
    - Like SOA brings together Business and IT
  - Separation of concerns for testing
    - View testing as a business process
    - Express Test Patterns in BPEL
  - And, please, less Excel...
Redefining the Economics of Mainframe SOA

DataDirect Technologies
Dave Little
April 25, 2008

Delivers Full Value in the Multi-Vendor World
(Legacy) Mainframe Systems

- Historically, information technology has been a major driver in the growth and success of almost all Blue Chip companies

- Since 1950s mainframe technology has been the foundation technology of some of the largest organisations

- Early adopters of state-of-the-art information technology continue to rely on mainframe technology as their system of record
Mainframe Demand Continues to Grow

- **Mainframe Relevance**
  - 14000 z/Series machines worldwide (50% are z/OS)
  - Greater than 60% of the world "system of record" data on mainframe
  - More than 450 of Fortune 500 rely upon mainframes
  - More commercial transactions processed on mainframe than any other platform

- **IBM z/Series 2097/E64 (aka z10)**
  - +/- 27000 MIPS across multiple z/OS images
  - 1520GB main memory
  - 1024 I/O Channel Processors
  - 10gb/s Data Transfer rate
  - 64 Processor Units (PU)
  - Java Co-Processor (zAAP)
  - Information Management Co-Processor (zIIP)
  - Crypto Co-Processor

- **Performance**
  - Scalable
  - Secure
  - Reliable

Mainframe Architectural Evolution

- **Transaction Monitors**
  - CICS, IMS, Natural, IDMS

- **Databases**
  - DB2, IMS, IDMS, Adabas, VSAM

- **Intellectual Property**
  - Business Logic / Rules
    - Commarea programs
  - Screen Logic
  - Data Logic/Data Access

- **Process Constraints**
  - Security
  - Visibility/Monitoring
  - Skills
  - Scalability
Primary Challenges to Mainframe in SOA

**Mainframe SOA Enablement**

- **Simplicity** – Reduce Complexity
- **Accelerate** – Speed Development
- **Industry Standards** – SOAP, SQL, BPEL
- **Quality of Service**

**Tough ROI for mainframe SOA**

- **Mainframe Processing Costs**
- **SOA on Mainframe is Expensive**
- **SOA Can Trigger Unplanned Costs**
- **Technology Needed to Reduce Mainframe Costs**

**Shadow Simplifies Mainframe SOA Enablement**
DataDirect Shadow Platform Details

- Multiple Industry Standard Interfaces
  - Web Services (SOAP)
  - Direct SQL Access
  - Real Time Events
  - Web Enablement
  - BPEL 2.0

- Multiple Mainframe Assets
  - Data: DB2, VSAM, IMS/DB, Adabas
  - Applications: CICS, IMS/TM, IDMS, Natural

- Universal Shadow Studio
  - Eclipse based window to the mainframe

- Enterprise Class Features
  - Shadow Instrumentation Server
  - Security Optimization & Management
  - Workload Manager integration
  - Industry standard BPEL for service orchestration
  - Full XA 2PC support with RRS
  - Multi-tasking and multi-threaded

Tools To Simplify Development & Reuse

- Data as a Service
- From SOAP to BPEL
- From SQL to SOAP
Maintaining Mainframe Quality of Service

- Data and transactional integrity are essential
- Fundamental to SOA are layers of abstraction
- Diagnostics challenge increases with addition of legacy services
- Enterprise class management needed for integration

SOA and Mainframe Security Management

- Each logon (RACROUTE) consumes 6 ms
- 1 million web service invocations per day
- 11 transactions per second
- 1.6 CPU hours savings per day
- 50 CPU hours per month (greater than 2 days)
- Typical SOM savings > 99%
Real World Processes Are Fragmented

- Applications deployed in different departments and business units become silos of data and process

- True process optimization requires more sophisticated tools

Simple orchestration using proprietary modeling tools cannot support real-world business process optimization

BPEL Orchestration Fits SOA Model

BPEL is the SOA standard for orchestration

Shadow allows organizations to leverage their mainframe to orchestrate platform independent, heterogeneous Web services using BPEL
BPEL Tools for Orchestrating Web Services with BPEL

Drag & Drop WSDL for Orchestration

Industry Standard BPEL Workflow

Shadow Quality of Service

- Real-time in-production visibility
  - Reduced problem determination & resolution cycle time

- Single Console to reduce operational complexity
  - IXF specialty engine support

- Security Optimization & Management
  - Support for stateless SOAP protocol
  - Eliminate repetitive authentication/authorization requests
  - Maintain security integrity
Primary Challenges to Mainframe in SOA

Mainframe SOA Enablement
- SIMPLICITY – REDUCE COMPLEXITY
- ACCELERATE – SPEED DEVELOPMENT
- INDUSTRY STANDARDS – SOAP, SQL, BPEL
- QUALITY OF SERVICE

Reducing Mainframe ROI for SOA
- MAINFRAME PROCESSING COSTS
- SOA ON MAINFRAME IS EXPENSIVE
- SOA CAN TRIGGER UNPLANNED COSTS
- TECHNOLOGY NEEDED TO REDUCE MAINFRAME COSTS

Factoring Mainframe Costs in SOA

Graph showing the relationship between MIPS of system and monthly license charge (MLC). The graph includes a trend line that indicates an incremental cost per MIPS of $512.
A Word About IBM Specialty Engines

- Internal Coupling Facility (ICF) 1997
  Centralized data sharing across mainframes
- Integrated Facility for Linux (IFL) 2001
  Support for new workloads and open standards
- IBM System z9 Integrated Information Processor (IBM zIIP)
  Designed to help improve resource optimization for eligible data workloads within the enterprise
- System z Application Assist Processor (zAAP) 2004
  Incorporation of Java™ into existing mainframe solutions

Exploiting zIIP for Lower Mainframe TCO

- Shadow 7.1 can exploit the zIIP for SOA and data related workloads
- Delivers dramatic improvement in SOA performance and TCO
Capacity Based Scenario – Typical Customer Capacity Growth

- This scenario depicts:
  - 20% ACGR in MSU Consumption (sited by IBM as average rate)
  - Shadow runs completely on the GPP
  - End result is no impact on when z/OS upgrade must occur
**Capacity Based Scenario – Impact of Shadow zIIP Exploitation**

- This scenario depicts:
  - 20% ACGR in MSU Consumption (sited by IBM as average rate)
  - Shadow’s zIIP eligible processing runs off the GPP
  - End result is that Shadow’s usage of the zIIP helps defer the upgrade

**Capacity Based Scenario – zIIP Savings v. Upgrade Deferral**

<table>
<thead>
<tr>
<th>Total MSU Savings</th>
<th>Current Utilization</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
<th>Year 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.0%</td>
<td>96.0%</td>
<td>115.2%</td>
<td>138.3%</td>
<td>165.9%</td>
<td>199.1%</td>
<td></td>
</tr>
<tr>
<td>14%</td>
<td>68.6%</td>
<td>82.3%</td>
<td>98.8%</td>
<td>118.5%</td>
<td>142.2%</td>
<td>170.7%</td>
</tr>
<tr>
<td>30%</td>
<td>55.9%</td>
<td>67.0%</td>
<td>80.5%</td>
<td>96.5%</td>
<td>115.9%</td>
<td>139.0%</td>
</tr>
<tr>
<td>40%</td>
<td>47.9%</td>
<td>57.5%</td>
<td>69.0%</td>
<td>82.8%</td>
<td>99.3%</td>
<td>119.2%</td>
</tr>
<tr>
<td>51%</td>
<td>39.0%</td>
<td>46.8%</td>
<td>56.2%</td>
<td>67.4%</td>
<td>80.9%</td>
<td>97.1%</td>
</tr>
</tbody>
</table>

- Shadow savings will impact the capacity growth of the customer mainframe
Capacity Based Scenario –
zIIP Savings v. Upgrade Deferral

<table>
<thead>
<tr>
<th>Total MSU Savings</th>
<th>Current Utilization</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
<th>Year 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>14%</td>
<td>68.6%</td>
<td>82.3%</td>
<td><strong>98.8%</strong></td>
<td>118.5%</td>
<td>142.2%</td>
<td>170.7%</td>
</tr>
<tr>
<td>30%</td>
<td>55.9%</td>
<td>67.0%</td>
<td>80.5%</td>
<td><strong>96.5%</strong></td>
<td>115.9%</td>
<td>139.0%</td>
</tr>
<tr>
<td>40%</td>
<td>47.9%</td>
<td>57.5%</td>
<td>69.0%</td>
<td>82.8%</td>
<td><strong>99.3%</strong></td>
<td>119.2%</td>
</tr>
<tr>
<td>51%</td>
<td>39.0%</td>
<td>46.8%</td>
<td>56.2%</td>
<td>67.4%</td>
<td>80.9%</td>
<td><strong>97.1%</strong></td>
</tr>
</tbody>
</table>

30% savings with Shadow & 50% workload saves **15%** total MSU consumption
(large z/Direct)

90% savings with Shadow & 20% workload saves **18%** total MSU consumption
(small z/Services)

90% savings with Shadow & 40% workload saves **36%** total MSU consumption
(large z/Services)

Challenges for Mainframe SOA

- **SOA Enablement for Mainframe**
  - Simplicity
  - Accelerated Development
  - Industry Standards
  - Quality of Service

- **Focus on Mainframe ROI for SOA**
  - Mainframe processing is expensive
  - SOA on mainframe is expensive
  - SOA can trigger additional costs
  - New technology now available to reduce mainframe ROI for SOA
Shadow Return on Investment

- Makes mainframe SOA enablement less complex
  - Single unified platform reduces integration points

- Increased speed to value
  - Industry standard tools
  - Eclipse GUI for accelerated developer productivity

- Maintains Quality of Service for mainframe SOA
  - Enterprise systems management to ensure integration integrity
  - Security Optimization for mainframe Web services

- Only mainframe middleware with patented technology for reducing mainframe SOA costs
  - Unique zIIP specialty exploitation for increased SOA performance and lower mainframe TCO

Questions and Answers
Recession-Proofing your Company with SOA

Jason Bloomberg
Managing Partner
ZapThink LLC

Copyright © 2008, ZapThink, LLC

Seeking a More Perfect Regulator: From Lincoln to Bush
A brief history of U.S. banking regulation

1863: Congress establishes national bank charter

1913: Federal Reserve System established

1933: In response to the Great Depression, Congress passes the Securities Act requiring that investors receive financial information about securities being offered for sale, and Glass-Steagall Act (President Roosevelt shown above signing it into law), which calls for the separation of commercial and investment banking and requires use of government securities as collateral for Federal Reserve notes.

1934: The Securities and Exchange Commission is created

1935: In response to shortcomings of Fed policy in Great Depression, the Banking Act of 1935 shifts power to make monetary policy towards Washington, creates the Federal Open Market Committee to do so, and separates Fed from administration by removing Treasury secretary and Comptroller of Currency from its governing board.

March 31, 2008: Treasury Secretary Henry Paulson announces plans for sweeping changes to the way the government monitors financial markets and to revamp the system of regulatory oversight

Sources: Treasury Department; SEC

1991: In response to S&L debacle, Congress passes Federal Deposit Insurance Corporation Improvement Act (FDICIA), which reduces regulators discretion to keep ailing banks alive

1999: The Gramm-Leach-Bliley Act effectively overturns the Glass-Steagall Act of 1933, allowing banks to offer financial services, such as investment banking and insurance sales

2002: Sarbanes-Oxley Act is passed, mandating reforms in corporate responsibility and enhanced financial disclosure, among other things
I coulda had SOA!!

Business Constant: Change

Changing Marketplace
Competition
Mergers & Acquisitions
Business Partners

Customer Demands
Optimizing Processes
New Technologies

A Business is Never STATIC
The Business Inflexibility Trap

- Inflexibility is the Mother of All Business Problems
  - If you’re flexible enough, you can solve all the other problems
- In tough times, inflexibility can lead to extinction!

Business Agility

- Companies require Business Agility...
  »Responding quickly and efficiently to change, and
  »Leveraging change for competitive advantage

Agility is the key to innovation
Business Agility in Tough Times

- Important to *identify* and *quantify* agility requirements
  - Time to value?
  - Efficient responses to regulatory change?
  - More successful mergers/acquisitions?
- Combine agility and cost savings efforts
- Remember to *measure* resulting agility!

You Can’t Afford *Not* to Innovate

- The tougher times get, the more essential agility is to survival
- Innovation requires bringing change to the market
  - Requires effective execution – on a budget!
How does SOA help Build Agility?

- Supporting flexible Services and processes over time
- Formalizing the approach to governance
- Providing better visibility into process change

Appropriate in good times & bad

Business Drivers for SOA

- Reduction in integration expense
  - Middleware replacement/Legacy rejuvenation
- Increase in reuse
  - Lower redundancy, better customer visibility
- Greater visibility
  - Enablement of governance & compliance, improved efficiency
- Business empowerment
  - Business control over flexible processes
- Increase in business agility
  - Improved competitiveness, faster innovation/time to market
Motivations in Tough Times

- Cost savings #1, but not the only priority
  - Regulatory compliance
  - Streamlining over slashing
  - Maintaining customer value
  - Strategic positioning for the recovery
  - Efficient M&A

*Focusing solely on cost reduction will weaken your organization*

How Do You Eat an Elephant?

- *One bite at a time!*
- Key best practice: take an *iterative* approach
- Don’t expect to have all the answers on day one
- Show business value at each step
Iterative: More than Step-by-Step

- Each iteration bounded by time, money, or scope
- Each iteration is a full project
- Improvements made to any part of earlier iterations in future iterations
- Lowers risk and delivers short-term value

Source: Pidex

SOA on the Roadmap?

- Focus on causes of excess IT spend
  - Redundant capabilities?
  - Inefficient IT processes, like funding, planning or governance?
  - Disconnect between IT and business?
- Position SOA as best-practice approach to tactical value
- Target quick-win projects iteratively
SOA in Progress?

• Focus on delivering short-term value
• Support your architecture team/SOA Center of Excellence
• Combine cost savings efforts with strategic efforts
  – Customer self-service
  – Time-to-value for new product/service
• Monitor business processes closely

Key Takeaways

✓ Business agility essential in good times & bad
✓ Iterative approach to SOA lowers risk & shows rapid value
✓ Cost savings important but not only motivation
✓ SOA can help position you for success, not just survival
The Problems of IT are
The Problems of Business

Module 5 Case Study:
SOA Organizational Change & Funding @ Novartis

Presented at Practical SOA: Frankfurt, Jan. 15, 2008

Licensed ZapThink Architect Bootcamp
Novartis
Company Overview

- International Healthcare Company
  - World’s 4th largest healthcare company
  - 90’000+ employees
  - 140+ countries
  - Headquartered in Basel/CH
- Balance sheet facts (first 9 months 2007 in USD):
  - Novartis Group
    - $28.1B net sales (increased by 13%)
    - $5.6B net profit (increased by 7%)
  - Pharma
    - $17.9B net sales (↑ 8%)
  - Consumer Health
    - $4.0B net sales (↑ 10%)

Why SOA?

Globalization
- Need to expand capabilities world-wide

Changing Populations
- Need to retarget and repurpose applications based on changing needs.

New Regulations
- Need to support enterprise-wide policy enforcement that may only be regionally relevant.

Increasing Cost
- Need cost-efficient lifecycle management and operations.

Consumerism
- Greater need for personalization and customization of capabilities to meet diversified/demanding individuals’ expectations.

Support Innovation
- IT systems need to meet future needs - even beyond what can be anticipated today (flexibility and agility).
Challenge: 3500 Data Centers!

- Data Centres

1 Terra Bytes External Traffic / Day

3,500 Data Center Servers

350 Supported Sites Worldwide

Reining in Complexity: Novartis Enterprise Architecture Council

The Enterprise Architecture Council hosts a chair for each IT department's representative – including Global Infrastructure Services (GIS).
Further refining of roles

- Enterprise Architecture provides holistic view
- Business Architects have to drive Business Standards
- IT Architects have to drive IT-Standards
- EA Architects have to combine IT- and Business Standards
A Top-down, Process-Driven EA Model @ Novartis

SOA Infrastructure
- SOA Services
  - SOA Processes
    - Application Frameworks
      - Processes
        - Governance FWs
          - Repository
  - SOA Services
    - Application Frameworks
      - Service Consumer
      - Service Provider
      - Process as Service
    - BPM & Orchestr.
      - BAM
    - ESB
      - Registry
        - T, MOM, Transform, Route, Secure, etc.

Applying SOA to the EA Model
Addressing the Funding Challenge

- Different business domains require different EA strategies
- Returns from IT investments depend on level of digitalization.
- Costs:
  - Enabling decentralized human decision making.
  - Automating centralized automatic decision making.
  - Optimizing process performance
  - Achieving consistency
  - Staff selection, placement, training.
- **Idea:** Focus on process-based funding. Seek money for the process, not the project.
- Focus on the hot spots for SOA... all else is wasted.

Lessons Learned

- Establish Business Architecture to move top-down (Business)
- Establish SOA Platform to move bottom-up (IT)
- Establish Enterprise Architecture program for holistic view
- Respect corporate culture
- Understand and apply architecture principles
- Focus SOA on the **process**, not on the **project**.
- Don’t just follow the crowd – understand your business’ reqs.
- Education has the best ROI in all architecture fields
Thank You!

ZapThink is an industry advisory & analysis firm focused on Service-Oriented Architecture, Enterprise Architecture, and Enterprise 2.0.

Jason Bloomberg
jbloomberg@zapthink.com