SOA Design, Development and Deployment
Methods, Processes and Tools

Hard Problems in SOA Workshop
January 30, 2008
Pittsburgh, PA

Participants

- 43 attended
- Lively discussion!
Main Issues Identified

- Methodology for system lifecycle with high rate of change and complexity
  - How to understand business process and strategic intent
  - How to design, build, test, deploy in that world
  - How to discover reusable services, how to customize to instance of need, what is deployment architecture

- How to understand and evaluate an Architecture
  - What are attributes that are most important, and particular in a SOA environment
  - Way to model and measure agility, rate of change, resiliency of architecture, ability to evolve
  - What are the generations of the architecture

- Assessment
  - Assurance, testing and problem determination
  - Suggestion: Build a maturity model (should be able to assess ourselves)

Methodology for system lifecycle with high rate of change and complexity – Rationale

- Rapid evolution of requirements
- Switch to focus on connecting components vs building single system
- Enabling decision making
- Prescribed methodology will help interoperability
- Shared & 3rd-party provided vs owned components
- Evolution of services in parallel with evolution of application using the services
- Separation of producer and consumer of services, need for “contract”
Methodology for system lifecycle with high rate of change and complexity – Current Efforts

- BPMN
- Agile development
- SOMA
- Ad-hoc assessments of legacy transformation
- SCA
- Formal analysis of BPEL models (Kramer, Magee)
- SEI – Smart legacy transformation
- OWL-S

Methodology for system lifecycle with high rate of change and complexity – Challenges and Gaps

- Methodology that bridges between business requirements, service and system design, and deployment architectures
  - Closer integration of business analysts with IT architects – using common methods, conceptual models and formalisms
  - Ability to model and simulate third party service levels to assess overall behavior

- Regulatory compliance issues
  - SOA methodology needs to evolve to address compliance issues while maintaining agility and speed
  - Across global set of regulations

- Asset management and reuse
  - Versioning and compatibility
  - Platform dependencies
  - Customization and configurability of assets
  - Change impact analysis
  - Methodology to modularize and SOA-enable legacy components

- Defining the details of the “contract”, SLA, specifications
  - How do you specify the service and service level you are providing?
  - What about open-source, 3rd party services?
Methodology for system lifecycle with high rate of change and complexity – Ideas to Address Challenges and Gaps

- Explore new methods and formalisms to better bridge business analysis with SW architecture
- Study effectiveness of different methodologies for rapidly evolving systems under set of regulatory constraints
- Address need for management of changes and versions at many levels of abstraction
- Investigate formalism for specification of Services and Service Level
- Define a common terminology for service repository accessible at design and run time

How to understand and evaluate an Architecture – Rationale

- Need to communicate clearly to identify and manage value and risk across an organization
  - Exacerbated by distributed ownership of components
  - Additional integration complexity
  - Much more emphasis on interfaces

- Need for interoperability, reuse
  - Avoid building SOA silos and duplicate efforts

- Need to measure agility, rate of change, resiliency of architecture, ability to evolve

- Frequent change of platforms and protocols
How to understand and evaluate an Architecture – Current Efforts

- SCA and SDO
- DODAF v2
- Vendor tools
- Various industry models
- Existing models, like ATAM
- OASIS reference model
- NIST’s SOA metrics

How to understand and evaluate an Architecture – Challenges and Gaps

- Building a common model/taxonomy of services
  - Common terminology across business and service providers
- Common metadata for describing and finding services
- Distributed data services not well known yet
- Acquisitions (self and technology suppliers)
- Education and training skills
How to understand and evaluate an Architecture – Ideas to Address Challenges and Gaps

- Develop method to incrementally build large systems quickly enough
- Training on current technologies and methodologies
  - Including getting adoption by open source community and students
- Better analysis techniques for understanding implications of models

Assessment – Rationale

- Assurance, testing and problem determination
  - Required for enterprise scale production
  - Required for ensuring appropriate ROI
  - Required for ensuring the ability to build on time and in budget
    - Need to know cost of services and what to charge people
  - Required for meeting SLAs
  - Required to mitigate risk
  - Required to measure quality throughout the lifecycle
  - Required to maintain operationally viability
- Regression testing in the face of dynamically changing services
  - Especially for SOA because SOA represents unique testing challenges
Assessment – Current Efforts

- Conventional testing tools
  - Many niche tools
- Building enterprise standards for accountability by individual companies
- Vendor maturity models
- Light-weight currency protocol

Assessment – Challenges and Gaps

- Impedance mismatch between specification and implementation of services
- Workflow complexities – services in development may need access to services in production
- May rely on services that you don’t control
- Many different parts that need to be part of test
Assessment – Ideas to Address Challenges and Gaps

- Build a maturity model (should be able to assess ourselves)
- Tie to methodology work
- Develop standard test beds
- Define ways to measure quality of an architecture
- TRL level for systems comprised of services
- Ability to provide certified tests by service providers to service consumers
  - Testing as a service itself

Backup
Hard Problems in SOA Workshop
Notes on Working Session 4: Design – Methods and Tools
Jan 30, 2008
CMU SEI, Pittsburgh, PA

Agenda

- 10.15 Position Statements
- 10.45 Selection of Topics
- 11.15 Topic 1
- 12 Working Lunch
- 12.15 Topic 2
- 1 Topic 3
- 2 Prepare Report
1. Requirements
   • SOA Specific questions
   • Standards/Configuration mgmt

2. Engineering
   • XForm non-SOA to SOA
   • Understand Risks/Cost
   • Modularizing, “SOAisizing” legacy

3. Maintain–Enhance
   • Evolution

Topics
   ◦ Support System Evolution
   ◦ Integration (heter, svs.)
   ◦ Assurance

Assurance/Testing
   ◦ Fragility of systems
     • Latency
     • Memory leaks
   ◦ Testing Methodology
     • Distributed, 3rd party
     • SW Engineering for composite apps

Requirements
   ◦ Need modeling, simulation to validate
   ◦ Derivation of lower level reqs for design & estimation
   ◦ Usability of architecture
Req’s process suboptimal for highly dynamic systems
Need language to specify policies (e.g., security) that can generate service impl.

System Partitioning
Information as a service
  ◦ Data still embedded
  ◦ How do you abstract?

Performance scale-up
  ◦ Each abstraction layer adds computations
SOA as a Science vs
- New paradigm
- Starting from a clean slate

SOA Technology
- E.g., Formal models
- Analysis
- SW Eng. Methods/Tools
- Still valid for SOA?

How do you design for flexibility? (Future needs)
- Design for reuse
- Role of Agile

Formal methods
- How do they work in different contexts
- What could we make simpler today?
- We struggle to integrate heterogeneous systems
- Skills

“Usability”
- Cleanliness of interfaces
- Predictability through modeling
- APIs: rethink
  - Identify criteria for trade-offs
- What are the component boundaries
- Think about specs, not code
Information architecture in SOA

Quality management across all lifecycles
- System of components
- Focus on whole/interactions
- Impact of loss of control–shared backbone services in evolution
- Stress discovery/reuse/interaction – not new development

Instrument services to enable measurements
How do you mitigate problem of not owning data def.
Why more important for SOA?
- You don’t know who is going to use service
- You want to use other people’s services
Proposed Topics

- Methodology for system lifecycle with high rate of change and complexity
  - How to design, build, test, deploy in that world
  - How to discover reusable services, how to customize to instance of need, what is deployment architecture
  - How to understand business process and strategic intent

- How to understand and evaluate an Architecture
  - What are attributes that are most important, and particular in a SOA environment
  - Way to model and measure agility, rate of change, resiliency of architecture, ability to evolve
  - What are the generations of the architecture

- Assessment
  - Assurance, testing and problem determination
  - Suggestion: Build a maturity model (should be able to assess ourselves)