Testing Service-Oriented Architecture (SOA) Solutions: Best Practices and Tools

2009 AF SOA Developer’s Conference

Rick Goodwin, SAIC
21 May, 2009
Common Objectives for SOA Solutions

• **Create greater agility** allowing IT to better and more economically support business objectives
• **Reduce time-to-delivery** for new services, because the implementation of **business processes** will be **highly configurable**
• **Better utilize development resources** through **reuse of existing assets**

• **Service-oriented architecture (SOA)** solutions must be **flexible** enough to support changing business strategies and **disciplined** enough to enforce policies and maximize re-use of services
• **SOA Solutions** are based on heterogeneous technologies and consist of loosely coupled, distributed services and components

• **SOA Solutions** introduce unique testing challenges
• **Test the unexpected**… multiple network conditions; system under attack, …
SOA Solutions Introduce Testing Challenges

- Service-oriented architecture (SOA) solutions require a more robust test methodology to test areas such as reuse, agility, vulnerability, and composability.

- Web services introduce unique testing challenges:
  - Web services are intrinsically distributed and are platform- and language-agnostic.
  - Web services ownership is shared across various stakeholders.
  - Testing that SOA governance (quality of service, regulatory policies, business policies, audit policies, infrastructure policies) is being enforced.
  - Dependencies on external Web Services can change without notice.
  - Federating identity information between services that reside in different domains while composing business processes or composite applications may be required.
  - Vulnerability testing is required – a good understanding of network security and the enterprise security architecture and policies is required.
  - Predicting future usage of services to assist with performance, load, and scalability.
  - Testing service behaviors in disconnected intermittent or limited (DIL) networks.
  - Testing SOA solutions when you do not have control over all of the Web services interacting with your solution.
  - Test teams require a blend of domain and technology expertise.

- Broad technical and domain experience is needed, because SOA testing involves not just functionality checks, but also reuse, agility, interoperability, security, and business process testing.
SOA Solutions Require New Test Methods

Composite Application Test Areas
- Unit Testing
- Integration Testing
- System Testing
- Scalability Testing
- Performance Testing
- Acceptance Testing
- Regression Testing
- Governance Testing

Service Specific Test Areas
- Reuse Testing
- Composability Testing
- Consumability Testing
- Interoperability Testing
- Reuse Existing Assets
- Agility Testing
- Security Testing
- Vulnerability Testing
- Federation Testing
- BPEL Testing

Create Greater Agility
- Policies
- Business Rules
- Configurations

Highly Configurable Business Processes
- Transaction Safety
- Service Unavailability
- Workflow Testing
- Events Testing

SOA = service-oriented architecture
BPEL = Business Process Execution Language
Best Practices Related to SOA Solutions

- Test service agility
- Test service vulnerability
- Test service performance over DIL networks
- Continuous integration and automated builds
- Continuous testing
- UML diagrams help identify Service usage patterns
- Virtual services to simulate real-world services
- Perform smoke tests
- Exercise concurrent code to test for race conditions
- Test from a variety of different client setups (different browser installations, different Java™ installations, etc.)

Java is a trademark of Sun Microsystems, Inc. in the United States and/or other countries.
DIL = disconnected intermittent or limited
UML = Unified Modeling Language
Test Service Agility

• Creating agile services is about identifying volatility and externalizing the configuration parameters, business rules, and policies that influence these behaviors
  – Configuration parameters influence system functions
  – Business rules influence business functions
  – Policies influence security, monitoring, performance, etc. … of a service

• Service agility testing involves testing the ability of a service to adapt to change — changing configurations, business rules and policies

• Another important factor in agility testing is measuring how quickly a service can be provisioned to accept a new consumer
Test Service Vulnerability

• Vulnerability testing is an emerging area of service-oriented architecture (SOA)-testing
  – Measure the vulnerability profiles of each Web service
  – Ensure that Web service vulnerabilities, such as buffer overflows, deeply nested nodes, recursive payloads, schema poisoning and malware traveling over Simple Object Access Protocol (SOAP) messages, do not affect critical Web services
  – Assess areas of exposure, determine severity levels, provide vulnerability diagnosis, and publish remediation techniques
Test Service Performance Over DIL Networks

- Use network simulation tools to mimic network conditions, such as network latency, network delay variation (jitter), bandwidth, congestion, packet errors, bit errors and other link impairments.
- Use network analysis tools to analyze the TCP/IP traffic on the network and record high-precision timings of services, processes and composite applications.

1) Observe service behavior in limited or disconnected networks.
2) Service performance is captured by looking at the TCP/IP packets.
3) Clocks on the clients and servers do not perturb test results.
4) TCP/IP timing is accurate to the micro-second.
5) No log correlation is required.
6) Network or TCP/IP issues are easily seen.

TCIP/IP = Transmission Control Protocol/Internet Protocol
Network Logging Setup and Analysis

- Cisco™ switch mirrors all network traffic on virtual local area network (VLAN) to NetVCR® machine
- NetVCR captures all traffic and uses virtual recording data sets to separate traffic for each system
- NetVCR exports to packet capture (PCAP) and ENC (Wireshark®) format for later analysis
- Custom Perl programs count different types of messages via scan of NetVCR log files
- Counts are fed into Excel® for graphical presentation and comparison with test event journal

Network analysis toolset
- NIKSUN NetVCR
  - Capture real-time network packet information from a router
- Wireshark
  - Use network protocol analyzer to perform packet analysis
- GL Communications IPNetSim™
  - Simulate network conditions such as network latency, network delay variation (jitter), bandwidth, congestion, packet errors, bit errors and other link impairments
  - Shape the bandwidth to perform simulated remote connectivity
- JFreeChart
  - Chart thousands to millions of data points
- Custom programs
  - Some tests may generate 4GB of data in a few minutes. Custom programs to post process/filter very large data results
Continuous Integration and Automated Builds

- Continuous integration – members of a development team integrate their work frequently, daily, and sometimes multiple integrations per day.
- Fully automated build system (using Python® and Ant [The Apache Software Foundation Apache Ant]) tags software and performs complete build of all components from configuration management in minutes.
- Fully automated deployment scripts push build out to all test systems for deployment of all components within an hour.
- Rapid turnaround of builds and deployments allows for multiple collaborative tests within a day.

Python is a registered trademark of Python Software Foundation in the U.S. and/or other countries.
Continuous Testing

- Iterative development models like SCRUM, XP, and TDD are pushing design engineers to be more involved in testing and often result in continuous test approaches
- Testing for service-oriented architecture (SOA) solutions never stops
- Test strategy should support testing across all phases of the service and solution life cycle, including runtime
- New testing tools go beyond application testing to include governance (HP® Service Test), such as policy enforcement, as well as life-cycle management. These tools provide process automation across the SOA governance and quality areas
- Continuous testing and monitoring services:
  - Functionality
  - Scalability
  - Performance
  - Compliance/governance
  - Interoperability/standards
  - Vulnerability
  - Agility

HP is a registered trademark of Hewlett-Packard Development Company, LP in the U.S. and/or other countries.

XP = Extreme Programming  TDD = test-driven development
Generate Service Tests From UML Diagrams

- A model-driven architecture (MDA) is a good way to model Web services in terms of interface definitions, use cases, sequence and activity diagrams, protocol state machines, communications diagrams, and business process diagrams.

- UML diagrams can model how a “user process” will interact with the service and what pattern of usage a service might expect. This is a step or two earlier in the design process than writing the specific business process rules, but it supports that step.

- Test programs can be modeled and code-generated, directly related to the data and service definitions in the model.

- By designing tests in parallel of designing the services, we have a better chance of testing the right things – the way the services will actually be deployed and used.

UML = Unified Modeling Language
Benefits of Diagrams/Models of SOA Solutions

- UML diagrams can help identify service usage patterns that can be improved, and address questions like
  - How often will the service be accessed and under what circumstances?
  - Are service interface definitions at the right level to support the overall business process flow?
  - What data translations need to occur between underlying data sources and the output of the service?
  - If a service is slow or fails, what downstream processes are affected?
  - What will the service payloads be? How big do we expect them to be?
  - Should the service be broken into a set of smaller services?
  - Is this service redundant or covered by another service or combination of services?

- These questions need to be addressed early in the system’s development life cycle

UML=Unified Modeling Language
Virtual Services Simulate Real-World Services

- Simulation techniques enable developers and testers to perform their tasks in parallel and can be used to enforce policies that service consumers must meet prior to deployment.
- iTKO® LISA® Virtual Service Environment (VSE) can capture, then model as a virtual service, the expected functionality of any service, whether a Web application user interface, a Web service (Web Services Description Language, WSDL), an integration layer such as an enterprise service bus, or an implementation layer such as an Enterprise JavaBeans® (EJB) or database.
- Crosscheck Networks, Inc.’s SOAPSImulator can mimic Web services before they are implemented.
- Green Hat Tester 5.0 provides automated capabilities for creating simulation assets for functional and performance testing.

iTKO and LISA are registered trademarks of Interactive TKO, Inc. in the U.S. and/or other countries. Enterprise JavaBeans is a registered trademark of Sun Microsystems, Inc. in the U.S. and/or other countries.
Net-Centric CBRNE Mission Thread

WebCentric GeoSpatial Collaboration is a registered trademark of Science Applications International Corporation in the U.S. and/or other countries.

PureActiv is a registered trademark of PureTech Systems Inc. in the U.S. and/or other countries.

CBRNE = chemical, biological, radiological, nuclear, and explosives
NORTHCOM = U.S. Northern Command
SCC-WMD = U.S. Strategic Command Center for Combating Weapons of Mass Destruction
EMF = enterprise management framework
IWM = Integrated Weapons of Mass Destruction Toolset
RSS = Really Simple Syndication
ASOCC = area securiaty operations command and control
CAP = Common Alerting Protocol
ESB = enterprise service bus
SLDS = senior leadership decision support
JMS = Java Message Service
JUM = joint user messaging

© 2009 Science Applications International Corporation. All rights reserved. SAIC, the SAIC logo and “From Science to Solutions” are registered trademarks of Science Applications International Corporation in the U.S. and/or other countries.
CBRNE Mission Thread - Incident Workflow

1. EMF picks up incident traffic from feeds
2. EMF publishes CBRNE incident assessment to JUM
3. PureActiv detects motion and publishes Possible Threat alert
4. Watch Officer receives alerts
5. Watch Officer reviews alerts & sends to DTRA and First Responders
6. DTRA Reachback Reviews Alert and Prepares Analysis
7. IWMĐT provides incident analysis with hazard area and casualties
8. First Responders Review DTRA analysis and plan their course of action

CBRNE = chemical, biological, radiological, nuclear, and explosives
NORTHCOM = U.S. Northern Command
EMF = enterprise management framework
JUM = joint user messaging
SBOC = service-oriented architecture-based operations center
DTRA = Defense Threat Reduction Agency
IWMĐT = Integrated Weapons of Mass Destruction Toolset
COCOM = Combatant Command

WebCentric GeoSpatial Collaboration is a registered trademark of Science Applications International Corporation in the U.S. and/or other countries.
PureActiv is a registered trademark of PureTech Systems Inc. in the U.S. and/or other countries.
Assembly of End-to-end Test Scenarios

Reuse and extend existing Unit Tests for functional, regression and performance testing

Assemble end-to-end test and validation workflows

“Invoke and Verify” each individual component in the workflow
Capture and Model Virtual Test Environments

Automatically produce realistic and stateful virtual test environment models

Capture and record live “conversations” with constrained IT resources

Further optimize the model by adjusting transaction responses and performance parameters
In-depth Visibility to Reveal Problem Areas

View detailed test results for each step in the workflow

Provide visibility and traceability across multi-tier, heterogeneous technologies

Drill-down into detailed performance results to resolve problems and optimize further
Monitor and Validate Mission-level Outcomes

- Customize the automated validation parameters for each test
- Trigger events and alerts to respond to issues in real time
- Monitor results based on real-time intervals, summary statistics, or pass/fail thresholds
Final Thoughts About Testing

• People, working together, are the most important testing practice that is necessary for a successful test program that results in quality solutions
• Testers exist to provide testing-related services. They do not run the development project; they serve the project
• Testing is done on behalf of the customer in the service of developing, qualifying, debugging, investigating, and understanding the current state of a system or service
• A “best” practice in the service of one mission might be irrelevant or counter-productive in the service of another
• The essential value of any test case lies in its ability to provide information (that is, to reduce uncertainty)
• Tests should become more challenging and should focus on different risks as a program matures
• Test artifacts are worthwhile to the degree that they satisfy the customer’s relevant requirements
• Start testing as early as possible in the system’s development life cycle