SOA Test Governance:
enabling service integration testing
across organization and technology borders

Antonia Bertolino
antonia.bertolino@isti.cnr.it

Andrea Polini
andrea.polini@unicam.it

Keynote Talk @ WebTest2009: 1st Int. Wksp on Web Testing
April 4th, 2009
Co-located with ICST 2009, Denver, Colorado, USA
SOA
SOA Governance
SOA Testing Challenges
SOA Test Governance (STG)
STG Cases:
  - Audition
  - SOCT
Conclusions&discussion
SOA: a new paradigm

- Service-orientation is claimed to be the “New Computing Paradigm for the Networked World”
  - (c.f.r. the theme of the First ICSOC in 2003)
- Revolution or Evolution?
  - Service-orientation (like Rome) “was not built in a day”: it is not a revolution in IT, rather it is the natural (and relatively slow) evolution from the confluence of several recent and less recent approaches and technologies.
- The WWW era in the 90’s heralded SOA by emphasizing:
  - shift from intra-organization to inter-organization integration
  - need: “B2B” advent
  - increasingly dynamic nature of IT market: companies enter and exit the scene continuously and fastly
SOA promises

➢ IT-based companies seem infatuated of the SO wave and in different ways and with different expectations are jumping on the bandwagon.

➢ The new paradigm brings in many attracting features, among which:

- reuse
- interoperability
- loose coupling
- standardization
- flexibility
- faster time to deployment
SOA in numbers

“SOA will be used in more than 50 percent of new mission-critical operational applications and business processes designed in 2007 and in more than 80 percent by 2010”

- Press Release from Gartner Symposium/ITxpo 2007: Emerging Trends

- In its report “SOA Infrastructure Market Shares, Strategies, and Forecasts, 2008 to 2014”, Research And Markets predicts SOA worldwide market will grow from an estimated $2 billion in 2007 to more than $9 billion by 2014

- See at http://blogs.zdnet.com/service-oriented/?p=1104, or you can buy the whole report from R&M (for € 3,531.00)
Service-orientation is a design paradigm comprised of service-orientation design principles. The logic units are services yielding distinct design characteristics that support the overall goals and vision of service-oriented computing.

A software service is a software component providing a functionality to other software components.

A service defines a precise interface to be used by clients in order to access it. Precisely, each service exposes one or more access points, defining the location where a provided functionality can be accessed.

The interface and the access points can be dynamically retrieved and used by other software components.

Service development is not related to the requirements of its future users.

Service is an abstract concept that is realized through the definition of an “including” infrastructure.
Service Oriented Architecture

- SOA is a conceptual architecture supporting the idea and the characteristics of services.
- In a minimalistic scenario there exist a service provider and a service requestor or consumer.
- Publishing a Web service involves creating the software artifact and making it accessible to potential consumers.
- Service discovery is the process of finding a suitable service for a given task. Optionally a provider can explicitly register a service within a service registry or broker.
SOA Development

- Traditional SE emphasizes the importance of fully documenting system requirements and specifications, based on which thorough analyses of the system under development can be performed as early as possible.

- SOA development in its very conception is drastically different: the vision is that SOA applications are composed on the fly at runtime using existing services.

- Therefore, SOA development is a collaborative effort from several parties: service developers, service providers, service brokers, application integrators.

- SOA engineering tasks are performed by separate organizations, none of which having complete information or control over the building pieces.
Need for SOA Governance

- To achieve SOA promises we need to set the “Rules of the game” that the stakeholders have to abide by.

- “Counterintuitive as it may seem, SOA requires more organizational discipline than previous development models. Your intuition might tell you that flexibility results from fewer rules, not more, but that’s not the case.”

Governance

- The rules, policies, practices and responsibilities by which any complex system of relations and interactions (a country, a company, a social community) is controlled and administered.

- (as compared to “government” term) brings in notions of
  - societal participation and shared responsibility;
  - moral qualification, orientation to the common welfare

(e.g., Wikipedia states: assuring, on behalf of those governed, a worthy pattern of good while avoiding an undesirable pattern of bad)
Enterprise Governance

- Governance applied to a business organization aimed at:
  - providing strategic directions
  - ensuring that company objectives are achieved
  - ascertaining that risks are appropriately managed
  - verifying that the enterprise’s resources are used responsibly.

- EG’s emphasis is on business goals

IT Governance

- IT Governance is a subset of EG, of growing importance in modern enterprises, that specifically focuses on IT assets and their exploitation.

- "ITG is the strategic alignment of IT with the business such that maximum business value is achieved through the development and maintenance of effective IT control and accountability, performance management and risk management."

[Phyl Webb, Carol Pollard, and Gail Ridley. Attempting to define IT governance: Wisdom or folly? Hawaii Int. Conf. on System Sciences, 8:194a, 2006]
Different key components make SOA governance at design-time and at run-time.

At design time it manages and deploys services throughout their lifecycle in order to ensure high levels of control and monitorability.

At run-time policies and procedures are embedded into the service management infrastructure.

- e.g. the service registry.
Governance intra-organization

- Assigns responsibilities, authorities and tasks (structural component of governance)
- Establishes measurement and enforcement mechanisms of assigned roles and responsibilities (dynamic or measurement component of governance)
Governance intra-organization (ctd)

Taken from [S. Chulani, C. Williams and A. Yaeli, Software Development Governance and Its Concerns, SDG’08, May 12, 2008, Leipzig, Germany]
Inter-organization level

- Governance is listed among the “grand challenges” for Service-oriented computing in reference 1. below, because of the “cross-organizational nature of end-to-end business processes”
- Indeed, service compositions across different enterprises must establish ownership and governance responsibilities.

There are similarities between CB development and Service-oriented life-cycle. Such similarities can be traced back to the separation of roles and concerns between those who develop a piece of software (a component or a service) and those who later use such piece of software for invocation or for integration.
Component/Service Developer

decides (on business grounds)

develops

Custom piece or reusable Component?

Component Spec, Code and Validate

Component Documentation For ext. use

Public Comp. Repository

packages

requires

Specific

Design

Code

Testing

Maintenance

For use

For external use
Component/Service User

System Architecture

- Component A
- Component B
- Component C
- Component D

Based on

in the context of

Public Comp. Repository

Candidate component

System Assembly

Discovers and selects

Analyzes and tests

Integrates

WebTest2009: SOA Test Governance ©2009 A.Bertolino, A. Polini
Communication Between Developer and User

Seen altogether

Custom piece or reusable Comp.?

? decides (on business grounds)

? develops

? augments

packages

Comp.Spec, Code and Validate

Component Documentation For ext. use

Public Comp. Repository

Candidate component

System Assembly

Discovers and selects

Analyzes and tests

Integrates
Concerning testing

- Testing in the traditional lifecycle relies on planning ahead based on development artifacts, and testing before release:

The classical V-model presupposes that the software tester yields full control and observability.
How is SOA testing different?

- We get all “traditional” testing challenges (no discount!)

  **PLUS new ones:**

- Global protocols of service interactions
- Adaptability and dynamicity
- Application “ilities”, e.g., availability, robustness, reliability, ...
- Distributedness, heterogeneity, mobility, ..., aspects of the implementation → first-class citizen
Challenges in SOA testing

- **Reduced Control:**
  - control all and only through the provided interface
  - no control over the services lifecycle (un-alerted evolution)

- **Reduced Observability:**
  - inter-organization distributedness
  - no access to code neither to its environment

- **Reduced Information**

- **Runtime integration:**
  - dynamic discovery and binding
  - context-awareness

- **Increased relevance of QoS characteristics**
...and opportunities

- Declarative and formal specification publicly available both for functional and non-functional characteristics
- New “control” point in the platform to which testing activities can be associated
  - E.g., directory services
- Open and standardized protocols and formats
  - SOAP, WSDL, XML
  - Orchestration provides a baseline for integration testing
  - Choreography provide an open specification to which cooperating services must conform
- Need of trust among organizations
Need for testing research

- From the EU report: “The service engineering area: An overview of its current status and a vision of its future” - July 2005

  “The area of testing services and their specifications has until now received limited attention from the research community. ....”
Introducing SOA Test Governance

Enterprise Governance

IT Governance

SOA Governance

SOA Test Governance

GOVERNANCE
STG is a subset of SOA governance that specifically concerns the establishment and enforcement of policies, procedures, notations and tools that are required to enable distributed testing of service integrations.
STG: an artificial idea?

- Analysis of interacting BPEL web services
  X Fu, T Bultan, J Su - Proc. WWW 2004
  - ...We assume that there is a virtual watcher which records the sequence of messages as they are sent by the peers. ...

- Model-based verification of web service compositions
  H Foster, S Uchitel, J Magee, J Kramer - Proc. ASE 2003
  - ... We need a specification that can closely resemble workflow specification languages, and a tool to simulate from the specification created in this language. ...

- Automatic conformance testing of web services
  R Heckel, L Mariani - Proc. FASE 2005
  - ...we foresee the introduction of High-Quality Service Discovery agencies .... automatically tests a Web Service with respect to a provided model consisting of GT rules that specify the individual operations of the service... ...

- ......

All above and many others are assuming a STG
Risks of Failure in SOA Projects

A 2007 press release from Gartner warns against risks of SOA project failure, which are both technical and organizational.

Gartner’s ‘hit list’ of the most common organisational errors includes:

1. **Overlooking governance**
2. Thinking an SOA project should be organised just like any other application development project
3. Not anticipating service number explosions in a maturing SOA
4. Giving up on an integration competency center or SOA center of excellence
5. Outsourcing architects (or not having them at all)

[taken from http://www.gartner.com/it/page.jsp?id=508397]
A 2007 press release from Gartner warns against risks of SOA project failure, which are both technical and organizational. Gartner’s ‘hit list’ of the most common organizational errors includes:

1. **Overlooking governance**
2. Thinking an SOA project should be organised just like any other application development project
3. Not anticipating service number explosions in a maturing SOA
4. Giving up on an integration competency center or SOA center of excellence
5. Outsourcing architects (or not having them at all)

“Through 2010, the biggest barriers to SOA adoption will be non-technical issues related to inadequate governance”
Gartner Research Vice Presidents, 2006
The ‘hit list’ of the most common technological errors includes:

1. Underestimating the technical complexity of a large-scale SOA
2. Bad selection of application infrastructure components (ESB, orchestration and adapters)
3. Insufficient validation of the SOA enabling technical infrastructure implementation (for example, no proof of concept and no stress tests)
4. SOA infrastructure, services and consumer applications are insufficiently instrumented for security/management/troubleshooting
5. Too-coarse/too-fine service granularity
6. Insufficient/not up-to-date documentation

[taken from http://www.gartner.com/it/page.jsp?id=508397]
The 'hit list' of the most common technological errors includes:

1. Underestimating the technical complexity of a large-scale SOA
2. Bad selection of application infrastructure components (ESB, orchestration and adapters)
3. Insufficient validation of the SOA enabling technical infrastructure implementation (for example, no proof of concept and no stress tests)
4. SOA infrastructure, services and consumer applications are insufficiently instrumented for security/management/troubleshooting
5. Too-coarse/too-fine service granularity
6. Insufficient/not up-to-date documentation

“testing is critical and at least 25 percent of the effort in a SOA project should be dedicated to this activity”
Gartner Vice President, 2005
Building a STG framework

STG Foundation Time

STG Design Time

STG Run-time
STG Foundation Time

Governance Board

- Establishes
  - SOA Test Standards
  - SOA Test Policies
  - SOA Test Rules
STG design

STG Design Time

Service Developer

Develops

Service Integrator

Composes

Service Composition

Service Provider
STG run-time

Service Provider

STG Run-time

Service User

Registry

On-line testing
Governance Board

this is the organism that is deputed to define the rules. Typically will be constituted by experts from organizations willing to cooperate or interested in establishing an environment to be used for cooperation. At the same time the governance board will define the various roles and will assign specific tasks to the various roles.
To ensure interoperability, Web services must comply with standards purposely created to enable communications across heterogeneous environments, both inside and across enterprise firewalls.

Web Services Interoperability Organization (WS-I) is an open industry consortium chartered to promote Web Services interoperability across platforms, operating systems, and programming languages.

http://www.ws-i.org/

WS-I does not itself define standards for web services; rather, in cooperation with standards development organizations such as OASIS / W3C, it creates guidelines and tests for interoperability.
STG stakeholders

- **Service developer**: develops a service, may be as consequence of a request made by a service provider
  - governance within each given context could impose development standards, implementation of specific interfaces and so on.
    - See, e.g., the role of service developer in SOCT case

- **Service provider**: deploys the service over a server and makes it available to external organizations (for pay-per-use services it is the provider who gets money)
  - Also in this case the governance could impose rules on the way in which a service is provided, for instance requiring that specific information is published
  - Provider can play the role of user when one of its services need to interact with a service deployed by another provider.
STG stakeholders

- **Service integrator**: derive new functionality integrating existing services.
  - is influenced by the governance that can define the rules and policies to be followed in the integration
  - code for the composition is available, invoked services instead are not generally under the control of the integrator

- **Service user**: accesses and uses a service
  - Has to abide by the established rules when using a service.
STG stakeholders

- **Monitor and enforcer**
  - These are specific organizations that can use mechanisms within a SOA infrastructure to check that what is happening is in-line with what specified by the governance. The governance will have to previously define which are the information of relevance for monitoring and which are the specific powers of the “monitor and enforcer” organization.

- **Certification Authorities** (not shown in Figure)
  - As standard architectures and policies are established, CAs will likely appear.
  - The certification authority can provide formal “stamp” of approval.
STG: two cases

- **STG in choreography testing: Audition**
  - Antonia Bertolino and Andrea Polini. The audition framework for testing web services interoperability. In Proc. EUROMICRO ’05

- **STG in orchestration testing: SOCT**
  - Cesare Bartolini, Antonia Bertolino, and Eda Marchetti. Introducing service-oriented coverage testing. In ASE Wksp ARAMIS 2008
The Audition testing framework proposed to add a testing phase before the service goes “on the scene”

The objective is to create a discovery service with **trustable entries**: registration is guaranteed only to services carefully tested.
The service wants to be registered within a UDDI server. To do this it specifies a binding to a specific WSDL.

The behavioural description for the service under registration is retrieved from the database. The service is inserted in a pending list.

The DB contains all the information concerning services, among them operational models describing expected service behaviour.

The testing engine starts to make invocations on the service under registration. Now under test (IBC).

Behavioural and WSDL descriptions are passed to the testing engine.

As consequence of the incoming invocations the service under test asks for references to required services.

Info on required services are retrieved from the DB.

LDDI activates the monitoring functionalities on the container of the required services (IBC).

The result of the testing phase is reported to LDDI.

Monitored invocation on the required service (IBC).

Final verdict reported to the WS.
On a pure SOA-based scenario the framework is not applicable.

STG involves:

- Service provider must be available and equipped for on-line testing
- Registry takes on an enhanced role
- The testing service is an additional component
- A choreography specifier publishes their specifications (according to a standardized format)
- Specific enforcement mechanisms must be implemented
STG: two cases

- **STG in choreography testing: Audition**
  - Antonia Bertolino and Andrea Polini. The audition framework for testing web services interoperability. In Proc. EUROMICRO ’05

- **STG in orchestration testing: SOCT**
  - Cesare Bartolini, Antonia Bertolino, and Eda Marchetti. Introducing service-oriented coverage testing. In ASE Wksp ARAMIS 2008
Similarly to testing of Component-based systems, the testing process of service-oriented systems is decomposed between two distinct stakeholders:

- **Service Developer**
  - Full access to code and specs
  - Does not know (future) use context

- **Service User/Integrator**
  - Only BB testing
  - Needs accompanying documentation for test design

Alice: Service Developer

Bob: Service User
WB testing for SOA?

- Services need to ensure “implementation neutrality”, i.e., the internal implementation details of a service must be totally uninfluential to its usage:
  - Services must be used as **black boxes**

- “[...] to users and systems integrators, services are just interfaces. This hinders white-box testing methods based on code structure and data flow knowledge.[...]”

White-box testing of SOA?

It is a pity that service integrators cannot use well-established and effective testing methods like coverage measures.

we exploit the very feature of service-oriented technology implement coverage testing, realizing what we call

Service-oriented Coverage Testing (SOCT)
SOCT scenario

1: Instruments services with TCOV probes

2: Invokes testable services

3: Logs coverage information

4: Gets test reports

TCOV Provider

Coverage Report Services

Testable Services

Alice:
Service Developer

Bob:
Service User

©2009 A. Bertolino, A. Polini
The basic assumption behind SOCT is an STG agreed among involved stakeholders:

- **Service Developer** must be willing to incorporate the probes into the services.
- **Service Integrator** performs WB testing, by analysing the log reports exported by the “testing Services”.
- Appropriate “Testing Services” are deployed by (external or internal) service providers, Testing Houses, Regulatory bodies …
STG behind SOCT

- Service developer must be willing to instrument service with the standardized probes
- TCov is a new stakeholder
  - Logs test runs
  - Provides test reports
- Service integrator interacts with TCov to obtain coverage measures
The framework presupposes a complex scenario, difficult to realize

Nevertheless, all WS* technology assumes standard processes and documentation

Can be applied with varying thoroughness and at different levels

Governance bodies must regulate the process
Conclusions

- We have introduced STG concept
- We have discussed challenges of WS testing and pointed at STG as a potential way to address them
- We have outlined a generic minimal STG framework
- We have overviewed two previous proposed approaches, in relation with STG:
  - Audition
  - SOCT
- We believe STG is a sensible important concept for SOA testing
...to be done

- Survey literature and classify STG implied approaches
- Raise abstraction level and also look at process/organization issues
- Embrace and spread STG awareness
- Build concepts and processes for STG
- A consensus-based community effort
WebTest 2009

SOA Test Governance:
enabling service integration testing
across organization and technology borders

Discussion time