SOA Testing As A Service
Are services architectures compliant with service contracts?

Riccardo Fontanelli (Dedalus S.p.A.)
riccardo.fontanelli@dedalus.eu - http://www.dedalus.eu/

Libero Maesano (Simple Engineering France s.a.r.l.)
libero.maesano@simple-eng.com – http://simple-eng.com

Internet of Services 2012 – 16 October 2012 - Diamant Center Brussels
Problem statement

SOA testing:
Why?
What?
How?
Who?
Where?
When?
How much?
MIDAS answers

• **Why?** – with the spreading of the *digital economy - digital actors* (systems and devices) collaborating *without human intermediation* and allowing the *automation of business processes* that support *daily activities* (e.g. e-Healthcare) – **strong dependability** and **security** of even ordinary tasks become **critical issues**

• **What?** – *testing* is a **must** - *formal checking* and *peer reviews* are not enough - but the reasons for which it is necessary to test SOAs are the same that make SOA testing hard
  – lack of observability, trust, control of the services architecture participants
  – fundamental uncertainty, organizational complexity, cost, scalability

• **How?** - **automated, elastic, intelligent model-based** and **usage-based** testing through **self-provisioning**

• **Who?** – each participant’s owner can test the *services architecture regions* s/he is concerned with (eventually the entire architecture)

• **When?** – if the Services Architecture Under Test (SAUT) is deployed, *whenever* s/he is ready to go

• **Where?** – if the Services Architecture Under Test (SAUT) is deployed, *wherever* s/he is situated

• **How much?** – *pay for use*
The MIDAS solution

Automated Elastic Intelligent SOA Testing self-service pay-as-you-go
MIDAS objectives

- Framework for SOA testing automation
  1. automated test suite generation *(model-based, usage-based)*
  2. automated test run execution on a TTCN-3 engine
  3. automated test evaluation, planning and dynamic scheduling based on probabilistic reasoning

- Functional, interactional, security, usage-based testing
  - of individual digital actors (systems, devices...) – *black-box testing*
  - of large services architectures (choreographies, end-to-end transactions) – *grey-box testing*

- Available as a Platform As A Service on cloud
  - self-service – pay-as-you-go
  - elastic, parallel, concurrent testing

- Suitable for different organizational contexts and goals
  - acceptance, non regression, reliability, stress, debugging support ... testing
  - failure detection, error identification, fault diagnosis

- To be evaluated in two strategic domains
  - E-Healthcare standard services architecture (HSSP, OMG, HSSP\(^1\)) – *interoperability testing*
  - Supply Chain Management services (also on mobile devices) architectures

\(^1\) Healthcare Services Specification Program (HSSP) - [http://hssp.wikispaces.com/](http://hssp.wikispaces.com/)
Model-based testing

- **Testing is always model-based testing** – at worst, the model is only in the head of the tester
- **Weak model-based SOA testing**: testing services architecture participants’ behaviors with the help of *test models + service usage models*
- **Strong model-based SOA testing**: testing services architecture participants’ behaviors with the help of *service models + test models + service usage models*
- **Service models = service contracts** – not only *interfaces*, but also *functions* and *external behaviors*:
  - *operations* – *what* the provider *does* for the consumer (not *how* it does it!)
  - *interactions* - *how* the provider and the consumer *interact in order to coordinate* the service invocation, deliberation, delivery (or refusal), reporting
  - *security* - *authentication*, *authorization*, *confidentiality*, *integrity*, *accountability*, *non repudiation*
  - *performance* – *volume*, *throughput*, *speed*, *scalability*, *availability*, *reliability*, *integrity*, *maintainability*
- Service models are *not* implementation models!
- **Strong model-based testing** is *testing-by-contract*!
- **Interoperability testing** is testing against *standard service contracts*!
- **Service usage models** to be confronted with service contracts
- **Test models** are SAUT models + test purpose models
- **Service models** (contracts) + *service usage models* + *test models* allow *automatic generation of* focused functional, interactional, security, and (tomorrow) performance *test suites*
- **Recommendations for testability of contracts and implementations** – not any SOA can be submitted to automated, elastic, intelligent testing – strong MIDAS concern
Cloud cooperation issues

- **Test and test data privacy** - for a lot of potential customers, tests and test data are sensitive
  - MIDAS Framework and Platform must run on **secure** clouds
- **Legislation issues** – e.g. dilemmas of European Branches of US Companies and of US Branches of European Companies
  - MIDAS Framework and Platform must be **portable** on different clouds
- **Interoperability** – MIDAS platform interacts with large SAUTs eventually implemented on several clouds
  - MIDAS Platform cloud should be able to **interoperate** with “any” cloud – on interoperability platforms such as SOAP and REST
  - On request, MIDAS Platform cloud should be able to host the SAUT – the SAUT should be **ported** easily on the MIDAS Platform cloud
- **Scalability management** - future development: testing SAUT against S(calability)LAs
  - SAUT and MIDAS Platform clouds should manage **automated scalability**
- **Other issues** – we will see ...
MIDAS
Model and Inference Driven - Automated testing of Services architectures
www.midas-project.eu

THANK YOU FOR YOUR ATTENTION

Internet of Services 2012 – 16 October 2012 - Diamant Center Brussels