The Dark Side of SOA Testing: Towards Testing Contemporary SOAs Based on Criticality Metrics

Philipp Leitner¹,

Stefan Schulte¹, Schahram Dustdar¹, Ingo Pill², Marco Schulz², Franz Wotawa²

¹Distributed Systems Group, Vienna University of Technology, Austria
²Institute for Software Technology, Graz University of Technology, Austria
Introduction

Service-based systems promise many advantages …

- … loosely coupled
- … interoperable
- … reusable
- … aligned with business goals
- … etc …

However, this generally comes with a price:

- Complexity / Error-proneness
SOA Testing – General Overview

1. SOA Model
2. Test Criteria Specification
3. User Model

- Test Suite
- Test Bed

Runtime Testing

- Test Result

Fault(s) Detected?

Fault Diagnosis

Fault(s) Detected?

YES

NO

YES

NO

Fault(s) Isolated?

End

Active Diagnosis
State of the Art

- There is some related work on testing

- But mostly:
  - Testing of single services, e.g.
  - Testing of service compositions, e.g.,

- Very little related work on how to test SOAs in its entirety
Contemporary SOA

Service Registry

CRM Service
Billing Service
Compute Service

Services

Service Consumers
SOA Testbeds

- What is currently missing?

Approaches to test not only ONE service or ONE process, but an entire service ecosystem (including complex dependencies and cross-references)

- In the end, you require complex testbeds to ‘test run’ your SOA components before deployment

- However, how do you build such testbeds?
Criticality Metrics

Our idea:

- Use **criticality metrics** to identify which components of your SOA are most relevant (and hence need to be part of your testbed)

- Based on the concept of complexity metrics from software engineering (e.g., cyclomatic complexity)

- Those metrics are **heuristics** to help human users identify critical SOA components
## Some Initial Example Metrics (1)

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Name</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF</td>
<td>Service Statefulness</td>
<td>{stateful, stateless}</td>
</tr>
<tr>
<td>NOSV</td>
<td>Number of Service Versions</td>
<td>(N_1)</td>
</tr>
<tr>
<td>NOPU</td>
<td>Number of Process Usages</td>
<td>(N_0)</td>
</tr>
<tr>
<td>IMR</td>
<td>Interface Mediation Required</td>
<td>{yes,no}</td>
</tr>
</tbody>
</table>
Some Initial Example Metrics (2)

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Name</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPC</td>
<td>Cyclomatic Process Complexity</td>
<td>$N_1$</td>
</tr>
<tr>
<td>CMEP</td>
<td>Complexity of Message Exchange Pattern</td>
<td>$N_1$</td>
</tr>
<tr>
<td>PLC</td>
<td>Process Landscape Categorization</td>
<td>{core, support}</td>
</tr>
<tr>
<td>DIR</td>
<td>Dynamic Invocation Ratio</td>
<td>0; 100]</td>
</tr>
</tbody>
</table>
Next Steps

- Clearly, our initial metrics (some more are in the paper) are only a rather simplified first step

- Next steps:
  - Survey more metrics / validate existing ones
  - Define a more concrete process for using those criticality metrics to build better testbeds
Acknowledgements

P23313-N23 (Audit 4 SOAs)
P22959-N23 (MoDiaForTed)
Current research is somewhat awkward:
  - A lot of research is done w.r.t. VM placement, energy efficiency, multi-tenant clouds, …
    - Issues that only interest cloud providers

More important (I think):
  - Consumer side issues

How to build applications for the cloud?
How to migrate between clouds?
How to ensure privacy in the cloud?
Current problems for SE in the cloud:

- Lack of understanding what the cloud is and does
  - Solution: ?? Educating people better / more truthfully ??

- Lack of good tooling to build cloud services
  - Solution: build better tools 😊
  - For instance CloudScale (which we will hear about in a later talk)