On Dependability of Composite Web Services with Components Upgraded Online

Peter Popov

Centre for Software Reliability

City University, London, United Kingdom

in collaboration with

Vyacheslav Kharchenko (NAU, Kharkiv, Ukraine),

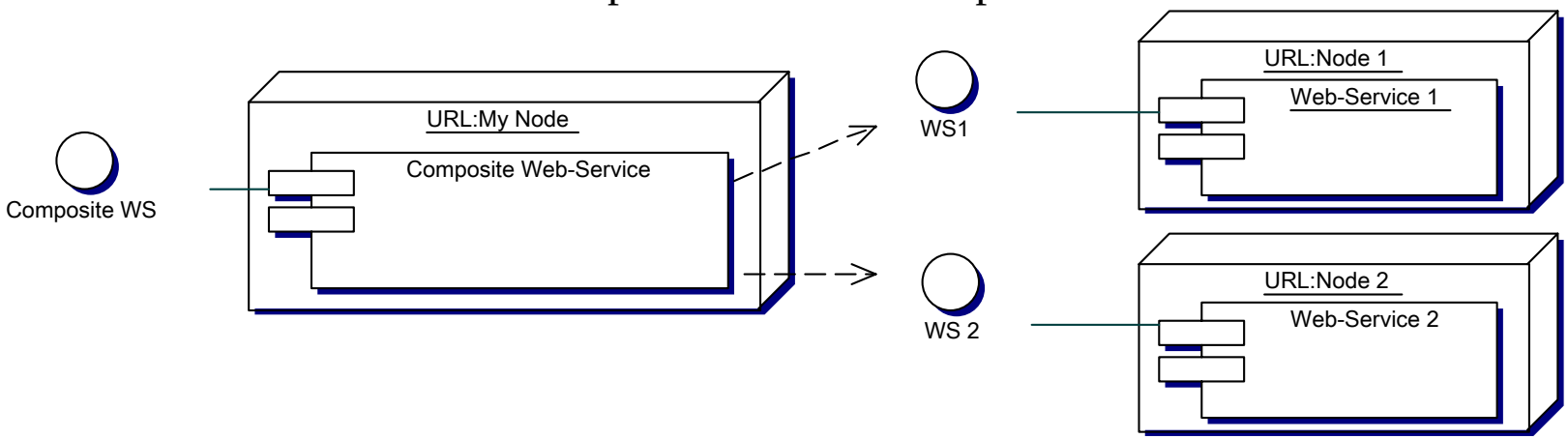
Alexander Romanovsky (CSR, Newcastle upon Tyne, UK)

Objectives

- Confidence in correctness of Web-services
  - why
  - how
- On-line upgrade of component services and its impact on composite Web-services
  - confidence in correctness of composite service
  - architectures for on-line upgrade

Background: Web-services

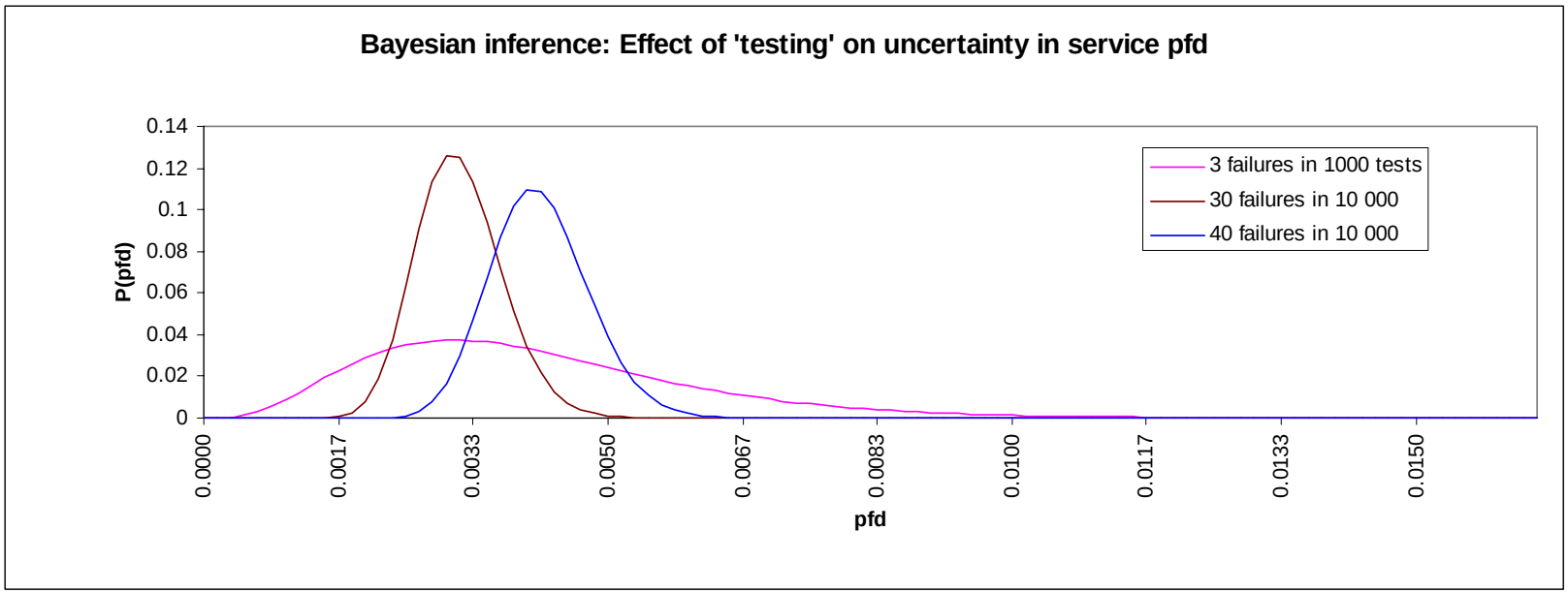
- A series of standards which made interoperability ‘easy’:
  - extendable WS descriptions kept and updated online in a registry
  - flexible binding, etc.
- Composite Web-service may depend on existing component services and on off-the-shelf (OTS) software
  - Composite web-services vs. integration of OTS software
    - similar in the absence of detailed information about their development
    - ‘locked-in’ with the provider of the component web-service



Background: Confidence in correctness

- Used in probabilistic assessment of safety-critical software:
  - done off-line (pre-deployment and periodic reviews)
  - with WS the same idea can be taken further and the confidence can be published and **updated continuously**.
- Bayesian inference (a way of calculating confidence)
  - a mathematically sound mechanism which allows one to combine the *a priori* knowledge and *new empirical evidence* (new observations)
  - detailed modelling of systems with many components leads to computational difficulties but feasible under plausible simplifications
- Confidence in system correctness can be calculated continuously
  - knowledge about the components only is NOT enough
  - solutions to this problem exist (previous work for safety-critical applications)

Bayesian inference: illustration



- pfd - probability of failure on demand.
- Probability of correct execution is a real number:

  P(correct execution on demand)=1-E[pfd]

- Slightly more complicated with several components.

Example: Confidence in correctness of WS

WSDL without confidence:

```
<types>
  <s:schema … >
    <s:element name=”Operation1Request”>
      <s:complexType>
        <s:sequence>
          <s:element minOccurs=”0” maxOccurs=”1”
           name=”param1” type=”s:int”>
          <s:element minOccurs=”0” maxOccurs=”1”
           name=”param2” type=”s:string”>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name=”Operation1Response”>
      <s:complexType>
        <s:sequence>
          <s:element minOccurs=”0” maxOccurs=”1”
           name=”Op1Result” type=”s:string”>
        </s:sequence>
      </s:complexType>
    </s:element>
  …
</types>
```

Confidence in correctness (2)

WSDL with confidence (1) redefining the old response:

```
<s:element name="Operation1Response">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
       name="Op1Result" type="s:string">
          <s:element minOccurs="0" maxOccurs="1"
       name="Op1Conf" type="s:double">
        </s:sequence>
      </s:complexType>
    </s:element>
```

- This may ‘break’ the existing composite services.

Confidence in correctness (3)

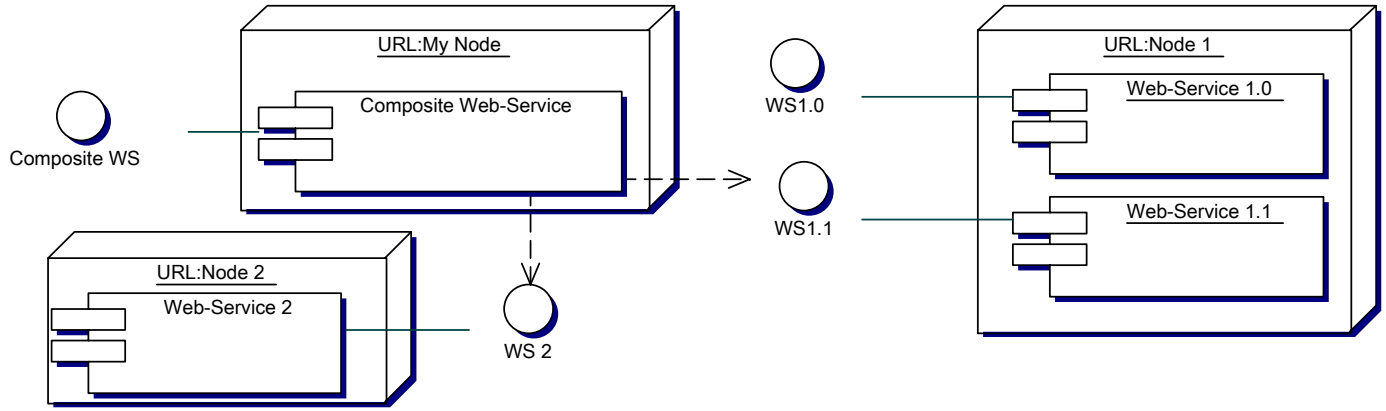
WSDL with confidence (2) a **new operation** defined:

```
<s:element name="OperationConfRequest">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
     name="operation" type="s:string">
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="OperationConfResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
     name="Op1Conf" type="s:double">
      </s:sequence>
    </s:complexType>
  </s:element>
```

Architectural implications of calculating Confidence in WS

- Confidence (**`Op1Conf`** or **`OperationConfResponse`**) 'continuously updated':
  - the confidence of a particular user A will be **affected by the other users** of the same WS (cost of 'testing' shared between the users)
- Some performance penalty (due to continuous Bayesian inference):
  - with every invocation of the operation (may be prohibitively expensive, especially under heavy load)
  - predefined intervals to make it 'cheaper'
  - **adaptive scheme (supported by system architecture)** of changing the intervals of calculating confidence depending on the load on WS.

On-line upgrade of a component WS



- A new release of WS1 is available. Confidence in the Composite WS can be calculated after the upgrade
- Possible architectural solutions for the service upgrade:
  - Both the old and the new releases, WS1.0 and WS1.1, are available:
    - continue using WS1.0 until the confidence in WS1.1 reaches ‘acceptable’ level. The architecture of the Composite WS should cater for the ‘transitional’ period
  - The new release WS1.1 replaces WS1.0. Some drop of confidence is possible. If not acceptable switch to using a **different component** WS.

Conclusions

- On-line assessment of confidence in correctness of Web-services proposed:
  - Various ways of publishing this confidence discussed
- Architectures of Composite WSs needed that allow for management of:
  - the cost of keeping the confidence in the correctness of the service up to date (changing the frequency of updates of the confidence to minimise the performance penalty)
  - the transitional period between the releases of the component services:
    - whether to switch from the previous release of a WS to a new one or not
    - whether to switch to an alternative component service if the new release ‘not good enough’
- Questions