A Tool Supported Approach to Perform Efficient Regression Testing of Web Services

Animesh Chaturvedi
animesh@iiitdmj.ac.in

Atul Gupta
atul@iiitdmj.ac.in
A Tool Supported Approach to Perform Efficient Regression Testing of Web Services

1. ABSTRACT
2. INTRODUCTION
3. DIFFERENCE, UNIT, REDUCE & COMBINED WSDL
4. REGRESSION TESTING OF WEB SERVICE WITH AWSCM
5. COST REDUCTION ESTIMATION METRICS
6. CASE STUDIES
7. CONCLUSION AND FUTURE WORK
8. RELATED WORK
Problem: Web service regression testing needs
  – changes in WSDL
  – changes in code
  – selective re-testing of web service operations.
ABSTRACT

**Fact:** of performing WS testing and Regression Testing

- Functional and non-functional WS testing is done with WSDL parsing [1, 2]
- Regression testing is performed by identifying the changes made [4, 6, 3, 17].

**Solution:** Automatic Web Service Change Management (AWSCM): Proposed three intermediate forms of WSDL for efficient regression testing of the web service by selecting the relevant test cases to constructing reduced test suite from the old test suite file of Soap-UI.

- Difference WSDL (DWSDL)
- Unit WSDL (UWSDL)
- Reduced WSDL (RWSDL), respectively.
- Combined WSDL (CWSDL)

**Efficiency:** Two proposed cost metrics for Reduction in the effort.

**Applications:** Three case studies: for the real world projects.
INTRODUCTION

• Regression Testing of Web Service (RTWS) by doing selective regression testing of modified and inserted portion only with the guaranteed that new and old code would conform to the changes in requirements.

• AWSCM which generates CWSDL whose operations are further mapped to their respective test case to do efficient RTWS by facilitating the standard WSDL parsing techniques i.e. generates smaller WSDL interface descriptions to creates Reduce Regression Test Suite (RRTS) for testing purposes.
DIFFERENCE, UNIT, REDUCE & COMBINED WSDL

Scope of changes handling by utilities

<table>
<thead>
<tr>
<th>Utilities</th>
<th>Capture changes WSDL or at code level</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWSDL</td>
<td>WSDL, operation deletion, insertion or I/O modification</td>
</tr>
<tr>
<td>RWSDL</td>
<td>WSDL, select operations on WSDL to test</td>
</tr>
<tr>
<td>UWSDL</td>
<td>Code of operation changes are captured</td>
</tr>
</tbody>
</table>

Diagram showing changes in WSDL structure with green and red highlights for inserted and deleted operations respectively.
REGRESSION TESTING OF WEB SERVICE WITH AWSCM

• Using (D/U/R/C) WSDL
  – to generate testing templates to write new test cases
  – reduced test case from the old test suite file on the basis of (D/U/R/C) WSDL.
• Deleted operation: operation will not be further in the next version of WS.
• Insertion of a new operation
• Modification occur in two levels at the WSDL level and at the code level.
  – at WSDL level means change in XSD
  – at the code level: JDiff, Java.Util.Regex: Change impact analysis using control and data flow analysis: UWSDL and RRTS according to the changes inside the operation.
• Using Operation in CWSDL: Combined RRTS (CRRTS) which contains all the unique test cases
REGRESSION TESTING OF WEB SERVICE WITH AWSCM
# Cost Reduction Estimation Metrics

Based on the change ratio between number of operations or lines in new version WSDL and (D/U/R/C) WSDL

## Operation Count Cost Metric

<table>
<thead>
<tr>
<th>Variables</th>
<th>Quantity</th>
<th>Unit</th>
<th>SaaS</th>
<th>Eucalyptus CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of operations in WSDL_1</td>
<td>X</td>
<td>Op</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>No. of operations in WSDL_2</td>
<td>Y</td>
<td>Op</td>
<td>4</td>
<td>28</td>
</tr>
<tr>
<td>No. of operations in DWSDL</td>
<td>Z</td>
<td>Op</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>No. of operations in DWSDL</td>
<td>Y-Z</td>
<td>Op</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

### Percentage reduction in operation by DWSDL

\[
((Y-Z)/Y)*100 \text{ Op} \]

<table>
<thead>
<tr>
<th>No. of Line of Code (to be tested) V'</th>
<th>C * Xav * Yav</th>
<th>LoC</th>
<th>38<em>24.5</em>48</th>
<th>117<em>61</em>253.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort required</td>
<td>V' / V_2</td>
<td>LoC</td>
<td>44688 / 148120</td>
<td>1824059 / 30389712</td>
</tr>
<tr>
<td>Percentage effort required</td>
<td>(V' / V_2) * 100</td>
<td>%</td>
<td>0.3017 * 100</td>
<td>0.0600 *100</td>
</tr>
<tr>
<td>Percentage effort reduction</td>
<td>100 – (V’ / V_2) * 100</td>
<td>%</td>
<td>100 – 30.17</td>
<td>100 – 6.00</td>
</tr>
</tbody>
</table>
CASE STUDIES

• (D/U/R/C) WSDL construction depends upon the availability of Web Service WSDL and code

<table>
<thead>
<tr>
<th>Case Study WS Projects</th>
<th>CWSDL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DWSDL</td>
</tr>
<tr>
<td>Eucalyptus</td>
<td>Y</td>
</tr>
<tr>
<td>SaaS</td>
<td>Y</td>
</tr>
<tr>
<td>Amazon WS</td>
<td>Different versions of WSDL is not available</td>
</tr>
<tr>
<td>Bible WS</td>
<td></td>
</tr>
<tr>
<td>Currency Conversion WS</td>
<td></td>
</tr>
<tr>
<td>Weather WS</td>
<td></td>
</tr>
</tbody>
</table>
CASE STUDIES

• AWSCM constructs reduced test cases from old test suite of SOAPUI
CASE STUDIES

• AWSCM constructs reduced test cases from old test suite of SOAPUI
CASE STUDIES

<table>
<thead>
<tr>
<th>WSDL</th>
<th>Test Suite</th>
<th>Reduce WSDL</th>
<th>Test Suite</th>
<th>Reduce Test Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- KeywordSearchRequest
- PowerSearchRequest
- BrowseNodesSearchRequest
- AddSearchRequest
- SelectSearchRequest
- AuthorSearchRequest
- ActorSearchRequest
- ManufacturerSearchRequest
- DirectorSearchRequest
- WishlistSearchRequest
- ExchangeSearchRequest
- MarketplaceSearchRequest
- SellingListRequest
- SellRequest
- SimilarToSearchRequest
- GetShoppingCartRequest
- ClearShoppingCartRequest
- AddShoppingCartItemRequest
- RemoveShoppingCartItemRequest
- ModifyShoppingCartItemRequest

```
- targetNamespace = http://soap
- name = AmazonSearch
- header
  - operationType = Name
- operation
  - input
  - output
- binding
  - name = AmazonSearchBinding
    - type = types:AmazonSearchP
    - soap:binding
      - name = AmazonSearchP
      - transport = http
- service
```

```
- name = AmazonSearchBindingTest
  - name = AmazonSearch
  - input
  - output
```

```
- name = AmazonSearch
  - input
  - output
```
CONCLUSION AND FUTURE WORK

AWSCM: approach to find out exactly which operations are changed and these operations for reduced regression testing.

Intermediate WSDL: (D, R and U) WSDL to generate three RRTS.

CWSDL contains unique operations in (D/U/R) WSDL; CRRTS generated from CWSDL.

Effort reduction: cost estimation metrics based on the change ratio between number of operations or lines in new version WSDL and (D/U/R/C) WSDL.

Further, extend UWSDL to capture changes in complex interactions between WS operations.
RELATED WORK

• David Binkley generated reduced test cases by identifies components that need tests after getting differences and similarities between the old and new programs [4].
• Rothermel and Harrold used CFG of program and its modified version to select tests cases that execute changed code from the original test suite [6].
• Romano and Pinzger proposed a tool called WSDL Diff to extract fine-grained changes from subsequent versions of WSDL’s of WS [8], similar work in reducing cost in RTWS is in [5].
• W. T. Tsai et al. described four ways in which the WSDL can be extended to facilitate WS testing in [2].
• X. Bai et al. proposed a WSDL based automatic test case generation approach for WS testing [1] is done by parsing WSDL [1] and the same forms the basis of automate WS testing tools like SoapUI and JMeter.
• A. Pasala et al. A tool for RTS selection based on analyzing the dynamic behavior of the application [3].
• J. Zheng et al. presented I-BACCI for change identification and then performing regression test selection, conducted in scenario of black-box [17].
• Thirumaran. M et al. in [18] proposed a dynamic business logic metric and algorithms for change factors evaluation helps in reducing the change management issue in WS.
REFERENCES


REFERENCES


REFERENCES


Thank You

- Italian: Grazie
- Hebrew: תודה רבה
- Japanese: ありがとうございます
- Korean: 감사합니다
- Portuguese: Obrigado
- Spanish: Gracias
- French: Merci
- German: Danke
- Russian: Спасибо
- Thai: ขอบคุณ
- Arabic: شكراً
- Traditional Chinese: 多謝
- Simplified Chinese: 多谢
- Hindi: धन्यवाद
- Tamil: வணக்கம்