An Empirical Comparison of Test Suite Reduction Techniques for User-session-based Testing of Web Applications

Goal: Compare the cost effectiveness of reduction techniques for Web application test suites

Motivation

- Test Web applications for correctness
- Collect user sessions as test cases
  - Representative of application’s evolving operational profile
- Problem: expensive to collect and replay large number of user sessions
- Solution: Test Suite reduction - reduce cost of storage, replay without losing original suite’s fault detection ability
- Goal: Empirical comparison of costs effectiveness of reduction techniques

Research Questions

- How much reduction in size of the test suite is obtained by the various reduction techniques?
- How effective is the program coverage of the reduced suites as compared to the original suite?
- How effective are the reduced suites when compared to the original suite with respect to fault detection?
- What are the relative time and space costs of the reduction techniques?

Subject Web Applications

- Bookstore
  - Open-source, e-commerce application
- Course Project Manager (CPM)
  - Course management tool for instructors and students

<table>
<thead>
<tr>
<th>Application</th>
<th>Classes</th>
<th>Methods</th>
<th>NCLOC</th>
<th>Faults</th>
<th>User Sessions</th>
<th>URLs req</th>
<th>Avg. US size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookstore</td>
<td>11</td>
<td>383</td>
<td>7791</td>
<td>40</td>
<td>123</td>
<td>3640</td>
<td>29 URLs</td>
</tr>
<tr>
<td>CPM</td>
<td>72</td>
<td>174</td>
<td>9542</td>
<td>60</td>
<td>73</td>
<td>1423</td>
<td>19 URLs</td>
</tr>
</tbody>
</table>

Summary of Results

- Reducing using concept analysis is a promising, cost-effective technique
  - reduces suite to size similar to requirements-based approaches
  - reduced suites have similar coverage, fault detection without overhead of requirements mapping

Empirical Results for Bookstore Application’s Reduced Test Suites

- General Reduction Algorithm
  - Reduce

- Requirements-based
  - Variations
    - Random
    - Greedy
    - HGS [Harrold 93]
  - create requirements mapping for each test case

- Concept Analysis Approach
  - [Sampath 04]
  - create concept lattice

- Examples of properties to preserve:
  - program coverage
  - statement, method, condition, MC/DC
  - fault detection
  - URL coverage

- Reduced Test Suite Size
- Reduced Test Suite Coverage
- Reduced Test Suite Fault Detection

http://www.cis.udel.edu/~hiper