Analyzing Clusters of Web Application User Sessions

Sreedevi Sampath, Sara Sprenkle, Emily Gibson, Lori Pollock
University of Delaware

Amie Souter
Drexel University

User-session-based Testing

- User sessions as test cases
  - Real usage of Web applications
  - Beta/maintenance testing phases

Motivation

Study dynamic behavior of user sessions

- Correlate user sessions to application behavior
  - program code covered by executing user session
  - faults detected when user session executed

Focus: Test Suite Reduction

- Use correlation to guide testing process
- Key Question: Is there adequate correlation between application behavior and user session clustering characteristics to warrant test suite reduction based on user session attributes only?
Program-based Test Suite Reduction

1. Get coverage, create requirements mapping for each test case
2. \( T \rightarrow \{ t_0, t_2, \ldots, t_j \} \)
3. Create concept lattice
4. Reduce test suite \( T \)

Concept Analysis

- Mathematical technique for clustering objects by their discrete, common attributes based on a binary relation
- Creates a partial ordering of concepts

Example: Modes of Transportation

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Airplane</th>
<th>Boat</th>
<th>Roller Skates</th>
<th>Unicycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>passengers</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>wheels</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>&gt;80mph</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Concept Lattice:

- \((\text{airplane}, \text{boat}, \text{roller skates}, \text{unicycle}), \text{null})\)
- \((\text{airplane}), (\text{boat}, \text{airplane}), (\text{roller skates}), (\text{unicycle}), \text{null})\)
- \((\text{airplane}, \text{>80mph}, \text{engine}, \text{passengers}), (\text{roller skates}, \text{wheels}), (\text{unicycle}, \text{worn}))\)
- \((\text{null}, \text{wheels}, \text{>80mph}, \text{passengers}, \text{engine}, \text{worn}))\)

Clustering User Sessions via Concept Analysis

Cluster user sessions \( s \) by their single URLs

Relation: \( s \) requests \( a \)

Attribute similarity increases
**Test Suite Reduction Heuristic**

Choose one representative user session from each next-to-bottom node
- Union of requested URLs

1. $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, \text{null})$
2. $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, \text{null})$
3. $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, \text{null})$
4. $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, \text{null})$
5. $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, \text{null})$
6. $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, \text{null})$

**Clusters and Use Cases**

- **Characteristics of clusters**
- **Shared URLs → use case**

**User Session B<sub>1</sub>**
- Search?author=knuth
- ViewItem?id=book023

**User Session B<sub>2</sub>**
- Y
- Search?sub=C+programming
- ViewItem?id=book576

**Cluster B: Browsing**
- Search
- ViewItem

**Cluster P: Purchasing**
- Search
- ViewItem
- Login
- Purchase

---

**Analyzing Overlap within Clusters**

Program characteristics, e.g., program coverage and fault detection, of a cluster’s user sessions

Cluster:
- $s_1$
- $s_2$
- $s_3$

**Overlap among all sessions**

Cluster:
- Overlap among all sessions
Intracluster Overlap Hypotheses

1. High commonality in coverage and fault detection among sessions within a cluster
2. Program-characteristic commonality increases with more common attributes

Analyzing Overlap Across Clusters

Clusters may share common program characteristics

Intercluster Overlap Hypothesis

Coverage, detected faults overlap across clusters is small

Case Study Subject Applications

- **Bookstore**
  - JSPs, MySQL backend
- **Course Project Manager (CPM)**
  - Java servlets, data store backend

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Bookstore</th>
<th>CPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td>11</td>
<td>75</td>
</tr>
<tr>
<td>Methods</td>
<td>385</td>
<td>172</td>
</tr>
<tr>
<td>NCLOC</td>
<td>7791</td>
<td>9298</td>
</tr>
<tr>
<td>Faults</td>
<td>40</td>
<td>86</td>
</tr>
<tr>
<td>Sessions</td>
<td>125</td>
<td>251</td>
</tr>
</tbody>
</table>
**Hypothesis 1**

- As attribute set size increases, coverage increases

**Validation of Intracluster Overlap Hypotheses**

- High commonality in coverage and fault detection among sessions within a cluster
- Program-characteristic commonality increases with more common attributes

Validates choosing representative user session from cluster

**Hypothesis 2**

- Intercluster overlap is only slightly less (<30 fewer methods) than intraclass overlap
Hypothesis 2

90% of pairs have less than 30 common methods out of 172 total methods.

Validation of Intercluster Overlap Hypothesis

Program-characteristic commonality small between next-to-bottom nodes for CPM.

Summary of Results

- **URL similarity** implies similarity in program coverage and detected faults:
  - Use URLs to describe test case behavior
  - Avoid cost of program-based techniques

- **Validates next-to-bottom heuristic**:
  - A user session is representative of its cluster's program characteristics
  - Next-to-bottom nodes represent different use cases

Future Work

- Additional reduction heuristics
- Alternative clustering techniques

- **Discussion**: How would clustering by other attributes, e.g., URLs with name-value pairs, change the results for intracluster and intercluster overlap?
  - Conclusions about use cases?