A Technique for Reducing User Session Data Sets in Web Application Testing

Porfirio Tramontana
Anna Rita Fasolino
Dipartimento di Informatica e Sistemistica
University of Naples Federico II, Italy

Giuseppe A. Di Lucca
RCOST – Research Centre on Software Technology
University of Sannio, Benevento, Italy
Testing Existing Web Applications

- **Open issue:**
  - Automatic generation of a minimal test suite for exercising functionalities of an existing Web Application

- **Possible solutions:**
  - **White Box testing techniques**, based on the analysis of the source code of Web Application artifacts (i.e. server pages and components)
  - **Black Box testing techniques**, based on the analysis of the user interactions (returned client pages and user actions)
Black Box testing techniques

- Based on the analysis of interactions between user and system:
  - Collection and analysis of the traces of the interactions between the user and the Web Application:

User session data based techniques

- Recently adopted in the Web Applications testing field by:
  - S. Sampath, V. Mihaylov, A. Souter, L. Pollock: "Composing a framework to automate testing of operational Web-based software", ICSM 2004
User-session data based techniques

- A set of user sessions are collected by monitoring real Web Application executions;

- Each user session covers a subset of the execution scenarios of the Web Application;

- Any user session is associated with a candidate test case;

- A test suite is a set of user sessions.
User-session data based techniques

- The **effectiveness** of a test suite obtained by collecting user sessions depends on the ability to test any execution scenario of the application.

- Wider the user session set, greater the effectiveness of the test suite.

- **But ...**
  - Wider the test suite size, greater the testing effort!

- **So ...**
  - Reduction techniques are needed in order to have minimal test suites reproducing any execution scenario.
The proposed testing strategy

Three steps:
1. User Session collection and analysis
2. User Session reduction
3. Test Suite Generation
1. User Session Collection

- A User Session is a representation of a sequence of interactions between an user and a Web Application. It can be composed of:
  - Server pages directly called by the user;
  - Static Client pages directly called by the user;
  - Client pages built on request by server pages
Built Client Pages Analysis

- Any Built Client Page experienced by a user is assumed as an instantiation of a basic scenario.

- Similar Built Client Pages correspond to the execution of equivalent scenarios.
Built Client Pages and Use Case Scenarios

- For each Server Page, the set of Built Client Pages can be partitioned in Equivalent Built Client Pages (EBCP) classes.

- Assumption:
  - Built Client Pages belonging to the same EBCP class correspond to executions of the same basic use case scenario.

- A user session can be collapsed into the sequence of corresponding EBCPs.

- The coverage of the User Session is the set of EBCPs visited at least once during the session.

<table>
<thead>
<tr>
<th>User Session</th>
<th>Sequence of EBCPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>US4</td>
<td>1 3 4 17 18 17 17 11 15 16 11 19 20 19 20 19 19 19 1</td>
</tr>
</tbody>
</table>

Covered EBCPs: \{1, 3, 4, 11, 15, 16, 17, 18, 19, 20\}
How can be classified the Built Client Pages into EBCPs?

- By comparing HTML code of Built Client Pages
  - Clone detection techniques

or

- By Analysing the execution paths of server pages generating Built Client Pages
  - BCPs belong to the same EBCP class if they are generated by Linear Dependent Execution paths
    - This technique has been presented by Di Lucca and Di Penta in “Integrating static and dynamic analysis to improve the comprehension of existing Web applications”, WSE 2005

An expert has to validate the partitioning obtained by applying these techniques.
2. User Session Reduction Technique

The goal of the User Session reduction technique is to obtain a reduced set of user sessions, including all the EBCPs of the Web Application.

### Coverage Matrix

<table>
<thead>
<tr>
<th></th>
<th>P₁</th>
<th>P₂</th>
<th>P₃</th>
<th>P₄</th>
<th>P₅</th>
<th>P₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>US₁</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US₂</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US₃</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US₄</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US₅</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US₆</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

- Each row is a User Session;
- Each column is an EBCP class;
- Each cell is 1 if the corresponding User Session contains at least an instantiation of a BCP belonging to the corresponding EBCP class;

<table>
<thead>
<tr>
<th>User Session</th>
<th>Sequence of EBCPs</th>
<th>Covered EBCPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>US₁</td>
<td>2 4 3 4</td>
<td>{2,3,4}</td>
</tr>
<tr>
<td>US₂</td>
<td>1 4 1 3</td>
<td>{1,3,4}</td>
</tr>
<tr>
<td>US₃</td>
<td>1 2 3</td>
<td>{1,2,3}</td>
</tr>
<tr>
<td>US₄</td>
<td>6 6</td>
<td>{6}</td>
</tr>
<tr>
<td>US₅</td>
<td>6 4 6</td>
<td>{4,6}</td>
</tr>
<tr>
<td>US₆</td>
<td>5</td>
<td>{5}</td>
</tr>
</tbody>
</table>
The Reduction Technique

Three Criteria:

- **Essentiality criterion:**
  - a user session US\textsubscript{i} is essential if it is the only user session containing a certain EBCP;

- **Row dominance criterion:**
  - a user session US\textsubscript{i} is dominated by a user session US\textsubscript{j} if all EBCPs contained in US\textsubscript{i} are contained in US\textsubscript{j} too;

- **Column dominance criterion:**
  - an EBCP \text{P}_i dominates a EBCP \text{P}_j if \text{P}_i is contained in all user sessions containing \text{P}_j.

- US6 is essential because it is the unique User Session containing P5
- US4 is dominated by US5 because all the EBCPs contained in US4 are also contained in US5
- P3 dominates P1 and P2 because P3 is contained in all the User Sessions containing US1 or US2
The Reduction Technique

- Essential User Sessions belongs to the Reduced Set;
- EBCPs covered by Essential User Sessions are deleted
- Dominated User Sessions are deleted
- Dominant Pages are deleted

Reduced Set: US3, US5, US6
3. Test Suite Generation

- For each User Session belonging to the reduced set, a test case may be obtained, also taking into account:
  - Pre-conditions (Web Application state before starting the User Session)
  - Data exchanged between the user and the web pages
Case Study

- Based on the open source Web Application Bookstore
- The version under exam is realised in PHP4, supported by a Microsoft Access database.
  - 28 Server Pages, three of which just included library functions and were not able to build any client page.
- The application was hosted by the Apache web server
An incremental analysis has been carried out in order to identify the set of EBCPs.

We concluded our experimentation after 63 User Session, when 33 different EBCPs was discovered.

We experienced that no new EBCPs was discovered by increasing the User Session Set.
Applying the Reduction Technique …

- Applying the presented reduction technique, 8 User Sessions were selected.

- Any of the 33 discovered EBCPs are actually covered by at least an User Session of the reduced set.

- A test suite containing 8 test cases, that is able to replay the user sessions belonging to the reduced set is generable.

<table>
<thead>
<tr>
<th>Number of User Sessions</th>
<th>Number of included EBCPs</th>
<th>Size of Reduced Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>7</td>
</tr>
<tr>
<td>45</td>
<td>32</td>
<td>7</td>
</tr>
<tr>
<td>63</td>
<td>33</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User Session</th>
<th>Sequence of EBCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>US4</td>
<td>1 3 4 17 18 17 17 11 15 16 11 19 20 19 20 19 19 19 19 1</td>
</tr>
<tr>
<td>US14</td>
<td>1 3 4 11 24 24 25 24 25 24 11 23 26 23</td>
</tr>
<tr>
<td>US29</td>
<td>1 3 4 11 19 20 19 19 11 29 12 29 29 3 3 4 1 10</td>
</tr>
<tr>
<td>US31</td>
<td>1 7 8 2 3 9 4 1 8 2 10 5 6 5</td>
</tr>
<tr>
<td>US39</td>
<td>1 3 4 11 12 13 13 30 12 32 4 3</td>
</tr>
<tr>
<td>US43</td>
<td>1 8 2 3 9 22 1 4 1 8 2</td>
</tr>
<tr>
<td>US51</td>
<td>1 3 4 21 1 11 12 13 14 16 16 15 17 3 4</td>
</tr>
<tr>
<td>US54</td>
<td>3 4 11 27 28 28 28 33 27 27 27 11 12 13 30 12 4 3</td>
</tr>
</tbody>
</table>
Conclusions and Future Works

- A technique for the extraction of a reduced test suite from User Session data has been presented.

- This technique is based on an analysis and classification of Built Client Pages and on the reduction of the User Session set.

- Case studies showed the effectiveness of the technique in reducing the size of the set of user sessions to be considered for testing.
Future Works

- Some proposals:
  - Experimentation with larger Web Applications
  - Implementation of a complete framework that is able to collect and analyse User Sessions, to generate the reduced test suite and to be able to replay tests belonging to the reduced set
  - Combination with reverse engineering approaches that are able to identify use case scenarios on Web Applications
  - Evaluation of testing effectiveness, by checking different coverage criteria
  - Application to regression testing
  - Comparison with other reduction techniques
Time is over ... Are there any questions?
A comparison with Sampath et al. technique

- Our technique is quite similar with the technique adopted by Sampath et al., that is based on Concept Analysis.
- Anyway the two techniques cannot be compared because:
  - Our technique is based on the analysis of visited EBCPs, while the Sampath technique is based on the analysis of visited server pages.
  - Our technique does not take into account the visiting order.
Concept Lattice

Reduced Set