Cross-Platform Testing and Maintenance of Web and Mobile Applications

ICSE 2014 Doctoral Symposium
Shauvik Roy Choudhary

Advisor: Dr. Alessandro Orso
School of Computer Science, Georgia Tech
Web & Mobile Applications

Web

Mobile

Web

Mobile
Testing & Maintenance Issues for Multi-platform Applications
I. Inconsistencies
I. Inconsistencies

Apple Safari
I. Inconsistencies
II. Missing Features
III. Costly Test Migration

Test Suite

Mobile App

Platform

iOS

Android
How can we address these problems?
Proposed Thesis

**Key Insight:** Analyze cross-platform application behavior to address the issues

**Challenge:** Significant difference in cross-platform behavior. Match behavior despite legitimate differences.

**Thesis:** Approximate behavior-matching algorithms and abstractions can be used to automate cross-platform testing and maintenance.
Overall Approach

Applications on two platforms → Behavior Capture → Captured Behavior of each application → Behavior Matching

Matched Behavior:
- \( a1.b1 = a2.b1 \)
- \( a1.b3 = a2.b3 \)

Unmatched Behavior:
- \( a1.b2, a2.b2 \)

Matching Result → Inconsistency Detection → Cross-Platform Inconsistencies (e.g., XBi) → Retargeted Tests for new platform

Feature Analysis → Cross-Platform Applications → Missing Features on one platform
Overall Approach

Applications on two platforms

Captured Behavior of each application

Behavior Capture

Behavior Matching

Matched Behavior

a1.b1 = a2.b1
a1.b3 = a2.b3

Unmatched Behavior

a1.b2, a2.b2

Matching Result

Inconsistency Detection

Cross-Platform Inconsistencies (e.g., XBIs)

Retargeted Tests for new platform

Feature Analysis

Missing Features on one platform

Cross-Platform Applications
Overall Approach

Applications on two platforms

Captured Behavior of each application

Behavior Matching

Matched Behavior
a1.b1 = a2.b1
a1.b3 = a2.b3

Unmatched Behavior
a1.b2, a2.b2

Matching Result
Overall Approach

- **Matched Behavior**:
  - $a_1.b_1 = a_2.b_1$
  - $a_1.b_3 = a_2.b_3$

- **Unmatched Behavior**: $a_1.b_2$, $a_2.b_2$

- **Matching Result**:
  - Inconsistency Detection
  - Cross-Platform Inconsistencies (e.g., XBIs)
  - Test Migration
  - Retargeted Tests for new platform
  - Feature Analysis
  - Missing Features on one platform

- **Cross-Platform Applications**
Research Outline

Cross-Browser Testing
- WebDiff [ICSM’10], CrossCheck [ICST’12], X-PERT [ICSE’13]

Feature Mapping
- FMAP [ISSTA’14]

Test-suite Migration
- Remaining work in progress
Cross-Browser Incompatibility Detection (for web applications)

WebDiff [ICSM’10], CrossCheck [ICST’12], X-PERT [ICSE’13]
Approach Overview

Web Application

Model Generation

Model Comparison

Effectiveness
77% Precision and 95% Recall

Improvement over state-of-art
45%↑ Precision & 14%↑ Recall

Error Report
Feature Mapping Across Platforms
(for desktop & mobile web apps)

FMAP [ISSTA’14]
Approach Overview

Platforms

Trace Extraction

Action Recognition

http://www...

Traces

A A M
X Q Q N
B R R O
Y Q S P
C R D S

Labeled Actions

A A A U
B B Q V
C C R W
D D S
Approach Overview

Trace Extraction

Action Recognition

Trace Simplification

Clustering and Mapping across Platforms

\[ JaccardDistance(a, b) = 1 - \frac{|words(a) \cap words(b)|}{|words(a) \cup words(b)|} \]
Approach Overview

- **Trace Extraction**
- **Action Recognition**
- **Trace Set Canonicalization**

**Platforms**

- **http://www...**

**Traces**

- **Labeled Actions**

**Features**

- **Feature Matching**

**Action recognition F-Score:**
97.8% (Desktop) vs 99.6% (Mobile)

**Overall Effectiveness:**
86.3% vs 51.5% (baseline)
Cross-platform Test Migration
(for mobile apps)

Remaining Work
Problem

• **Given:** Test Suite \((TS_i)\) for App on Platform 1
• **Task:** Generate corresponding Test Suite, \((TS_a)\) for the same App on Platform 2

\[ TS = \text{Set of TC} \]
\[ TC = [a_1, a_2, \ldots, a_N, \text{Oracle.assert()}] \]

Action is \([\text{ActionType, Selector, Data}]\)
Example Test on iOS

- tap “Comments”
- tap cell[0]
- tap “icon delete”
Example Test on Android

- tap “Comments”
- tap checkBox[0]
- tap “Delete”
Challenges

No structural similarity

Independently developed
(In different languages & frameworks)

Automated Behavior Exploration

Partial

Same actions
Different Widgets
Assumptions

• **Action correspondence:** If actions are present across platforms, they have a 1-1 correspondence

• **Action ordering:** The matched actions appear in same order in matched use cases

• The test cases given to translate, have implementations on both platforms
High-level Overview

Test Trace
\[ tt = < a_1, a_2, \ldots, a_n > \]

Model Trace
\[ mt = < a_1, a_2, \ldots, a_u > \]

Trace Extraction

Model Generation

Match Actions

Generate Candidate Tests

Developer provides feedback on partially migrated tests

Model Enrichment
Action Matching as an Optimization Problem

• Given: Test traces for Platform 1
  Model for Platform 2

• Formulation:

\[
\max \sum_{a_1 \in \Sigma t i} |Map(a_1)|
\]

Such that

• \( Map : a_1 \rightarrow a_2 \quad a_2 \in \{\varepsilon \cup \Sigma\} \)
  \( a_1 \) & \( a_2 \) from platforms \( p_1 \) & \( p_2 \)

• \( |Map(a)| = 0 \) if \( a \) is mapped to \( \varepsilon \) and 1 otherwise

• \((Map(a_{1,i}) = a_{2,x}) \land (Map(a_{1,j}) = a_{2,y}) \land (a_{1,i} < a_{1,j}) \land (a_{2,x} \neq \varepsilon) \land (a_{2,y} \neq \varepsilon)\)

\[\implies (a_{2,x} < a_{2,y}) \land (\forall a_{2,z} \mid (a_{2,x} < a_{2,z} < a_{2,y})) \land \exists a_{1,k} \mid (Map(a_{1,k}) = a_{2,z}) \]

\(\rightarrow (a_{1,i} < a_{1,k} < a_{1,j})\)

\(tt = a_{1,1}, a_{1,2}, \ldots, a_{1,m}\)

\(mt = <a_1, a_2, \ldots, a_u>\)
Action Matching as an Optimization Problem

- Given: Test traces for Platform 1
  Model for Platform 2
- Formulation:

$$\max \sum_{a_1 \in \Sigma}$$

Such that

- $$Map : a_1 \rightarrow a_2 \quad a_2 \in \{\varepsilon \cup \{\text{actions}\}\}$$
- $$|Map(a)| = 0$$ if $$a$$ is mapped to $$\varepsilon$$
- $$\exists a_1, a_2$$ such that
  $$\left(\text{Map}(a_1,i) = a_2,x \wedge \text{Map}(a_1,j) = a_2,y \wedge (a_1,i < a_1,j) \wedge (a_2,x \neq \varepsilon) \wedge (a_2,y \neq \varepsilon)\right)$$
  $$\implies (a_2,x < a_2,y) \wedge (\forall a_2,z \mid (a_2,x < a_2,z < a_2,y)) \wedge \exists a_1,k \mid (\text{Map}(a_1,k) = a_2,z) \rightarrow (a_1,i < a_1,k < a_1,j)$$

Ordering Constraint

On matched actions
Action Matching as an Optimization Problem

- Given: Test traces for iOS
- Model for Android

Formulation:

\[
\max \sum_{a_1, a_2} \text{such that}
\]

- \(\left|\text{Map}(a)\right| = 0\) if \(a\) is mapped to \(\epsilon\) and 1 otherwise

- \(a_{1,i} \neq a_{1,j}\) and \(a_{2,x} \neq a_{2,y}\)

\[
(Map(a_{1,i}) = a_{2,x}) \land (Map(a_{1,j}) = a_{2,y}) \land (a_{1,i} < a_{1,j}) \land (a_{2,x} \neq \epsilon) \land (a_{2,y} \neq \epsilon)
\]

\[
\Rightarrow (a_{2,x} < a_{2,y}) \land (\forall a_{2,z} \left( a_{2,x} < a_{2,z} < a_{2,y} \right)) \land (\exists a_{1,k} \left( Map(a_{1,k}) = a_{2,z} \right)) \rightarrow (a_{1,i} < a_{1,k} < a_{1,j})
\]
Branch & Bound Strategy

INITIAL STATE

Platform 1 actions = \{a_{11}, a_{12}, a_{13}, a_{14}\}
Platform 2 actions = \{a_{21}, a_{22}, a_{23}, a_{24}\}

Assign $a_{11}$

Assign $a_{12}$

Assign $a_{13}$

Assign $a_{14}$
Branch & Bound Strategy

- Test traces: \(<a_{11}, a_{12}, a_{13}>, \langle a_{11}, a_{12}, a_{14}\rangle>\)
- Model:

Profit = 4
Branch & Bound Strategy

- Test traces: \(<a_{11}, a_{12}, a_{13}> \quad <a_{11}, a_{12}, a_{14}>

- Model:

  - Initial State
  - Assign \(a_{11}\)
  - Assign \(a_{12}\)

Profit = 1
Branch & Bound Strategy

- Test traces: \(<a_{11}, a_{12}, a_{13}\> <a_{11}, a_{12}, a_{14}\>
- Model:

**INITIAL STATE**

- Assign \(a_{11}\)
- Assign \(a_{12}\)
- Assign \(a_{13}\)

Profit = 3
Evaluation (TBD)

**Tool:** MigraTest - Implementation of the technique

**Subjects:** Apps with iOS and Android versions

**Source Test-suite:** Recruit humans to develop tests

**Research Questions:**

**RQ1 (Effectiveness):** Can MigraTest effectively migrate test cases from one platform to another?

**RQ2 (Quality):** Do migrated tests hide or reveal any issues in the app on the target platform?
<table>
<thead>
<tr>
<th>Semester</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summer 2014</td>
<td>Attend ICSE 2014 Doctoral Symposium</td>
</tr>
<tr>
<td></td>
<td>Conduct Evaluation for <em>Test Migration</em></td>
</tr>
<tr>
<td>Fall 2014</td>
<td>Submit <em>Test Migration</em> to top conferences</td>
</tr>
<tr>
<td></td>
<td>Write the dissertation</td>
</tr>
<tr>
<td>Spring 2015</td>
<td>Graduate</td>
</tr>
</tbody>
</table>
Summary

**Web & Mobile Applications**

- Web
- Mobile Web
- Mobile

**Problem Space**

- Cross-Browser Testing
- Feature Mapping
- Test-suite Migration

**Overall Approach**

- Applications on two platforms
- Captured Behavior of each application
- Matching Behavior
  - Matched Behavior:
    - a1.b1 = a2.b1
    - a1.b3 = a2.b3
  - Unmatched Behavior:
    - a1.b2, a2.b2

**Progress**

- Cross Browser Testing
  - Published: WebDiff [ICSM’10], CrossCheck [ICST’12], X-PERT [ICSE’13]
  - Accepted: X-PERT Tool paper [ISSTA’14]

- Feature Mapping
  - Accepted: FMAP [ISSTA’14]

- Test Migration
  - In Progress: Problem formulation, Evaluation