A Framework for Automated Test Mocking of Mobile Apps

Mattia Fazzini

Alessandra Gorla

Alessandro Orso

University of Minnesota

imdea

Georgia Tech
Mobile Applications
Testing
Testing in this Type of Software Environment

- **Flaky Tests**
  - Test 1
  - Test 2
  - Test 3
  - Test 4
  - Test 5
  - Test 6

- **Slow Tests**

- **Interference Bugs**
  - App
  - Dependency
Test Mocks
Test Mocks

Test Execution

App

Test Mock

Developer
TestMocks

Test Execution

App

Test Mock

Developer

Time-consuming

Error-prone

Too Specific
MOKA Overview

Collect mock data from test executions

Use iterative approach based on program synthesis to generate test mocks
Mock Data Collection

Mock Data

- Mocked entity
- Mock input
- Mock output
- Mock components
- Mock coverage
Iterative Mock Generation Process

- Group mock data by mocked entity
- Generate test mock through program synthesis
  - Limit synthesis components to mock components
  - Size of synthesized program ≤ size of modeled method
  - Constraint composition of AST nodes during synthesis
- Generate new mock data
  - Input generation based on concolic execution
  - Test executions from other apps
- Generated record-and-replay-based mock
- Test Mock
Test Mock Generation

Iterative Mock Generation Process

1. Group mock data by mocked entity
2. Generate test mock through program synthesis
   - Limit synthesis components to mock components
   - Size of synthesized program $\leq$ size of modeled method
   - Constraint composition of AST nodes during synthesis
3. Generate new mock data
   - Input generation based on concolic execution
   - Test executions from other apps
4. Generated record-and-replay-based mock
5. Test Mock

App
Test Mock Generation

Iterative Mock Generation Process

- Group mock data by mocked entity
- Generated record-and-replay-based mock
- Test Mock

- Generate test mock through program synthesis
  - Limit synthesis components to mock components
  - Size of synthesized program ≤ size of modeled method
  - Constraint composition of AST nodes during synthesis

- Generate new mock data
  - Input generation based on concolic execution
  - Test executions from other apps
Test Mock Generation

Iterative Mock Generation Process

- Group mock data by mocked entity
- Generated record-and-replay-based mock
- Generate test mock through program synthesis
  - Limit synthesis components to mock components
  - Size of synthesized program ≤ size of modeled method
  - Constraint composition of AST nodes during synthesis
- Generate new mock data
  - Input generation based on concolic execution
  - Test executions from other apps
- Test Mock
Test Mock Generation

Iterative Mock Generation Process

- Group mock data by mocked entity
- Generated record-and-replay-based mock
- Generate test mock through program synthesis
- Limit synthesis components to mock components
- Size of synthesized program \( \leq \) size of modeled method
- Constraint composition of AST nodes during synthesis
- Generate new mock data
- Input generation based on concolic execution
- Test executions from other apps

App

Test Mock
Test Mock Generation

Iterative Mock Generation Process

Group mock data by mocked entity
Generated record-and-replay-based mock

Generate test mock through program synthesis
- Limit synthesis components to mock components
- Size of synthesized program \( \leq \) size of modeled method
- Constraint composition of AST nodes during synthesis

Generate new mock data
- Input generation based on concolic execution
- Test executions from other apps

App
Test Mock
Test Mock Generation

Iterative Mock Generation Process

1. Group mock data by mocked entity
2. Generated record-and-replay-based mock
3. Generate test mock through program synthesis
   - Limit synthesis components to mock components
   - Size of synthesized program ≤ size of modeled method
   - Constraint composition of AST nodes during synthesis
4. Generate new mock data
   - Input generation based on concolic execution
   - Test executions from other apps

App

Test Mock
Test Mock Generation

Iterative Mock Generation Process

- Group mock data by mocked entity
- Generate test mock through program synthesis
- Generate new mock data
- Input generation based on concolic execution
- Test executions from other apps
- Limit synthesis components to mock components
- Size of synthesized program \( \leq \) size of modeled method
- Constraint composition of AST nodes during synthesis

Generated record-and-replay-based mock

App

Test Mock
Preliminary Empirical Study

Analyzed tests and test mocks in mobile apps

F-Droid

GitHub

1,220 Apps

20% of the apps have tests, for a total of 11,487 tests
## Preliminary Empirical Study - Test Mocks

<table>
<thead>
<tr>
<th>Name</th>
<th>Tests</th>
<th>TMs</th>
<th>AFTMs</th>
<th>ATMs</th>
<th>TLPTMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CINELOG</td>
<td>152</td>
<td>285</td>
<td>23</td>
<td>210</td>
<td>52</td>
</tr>
<tr>
<td>EVENTYAY</td>
<td>477</td>
<td>268</td>
<td>111</td>
<td>146</td>
<td>11</td>
</tr>
<tr>
<td>WIFIANALYZER</td>
<td>708</td>
<td>206</td>
<td>91</td>
<td>85</td>
<td>30</td>
</tr>
<tr>
<td>K-9 MAIL</td>
<td>536</td>
<td>135</td>
<td>20</td>
<td>104</td>
<td>11</td>
</tr>
<tr>
<td>MATERIALISTIC</td>
<td>312</td>
<td>97</td>
<td>24</td>
<td>49</td>
<td>24</td>
</tr>
<tr>
<td>SMS BACKUP+</td>
<td>217</td>
<td>75</td>
<td>11</td>
<td>53</td>
<td>11</td>
</tr>
<tr>
<td>DNS66</td>
<td>66</td>
<td>60</td>
<td>45</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>AnkiDroid</td>
<td>248</td>
<td>38</td>
<td>26</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>SMSSync</td>
<td>23</td>
<td>32</td>
<td>2</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>LOOP HABIT</td>
<td>277</td>
<td>32</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>COMMONS</td>
<td>21</td>
<td>32</td>
<td>30</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>OPEN KEYCHAIN</td>
<td>217</td>
<td>30</td>
<td>15</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>WIKIPEDIA</td>
<td>365</td>
<td>24</td>
<td>0</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>WEB OPAC</td>
<td>16</td>
<td>23</td>
<td>8</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>PAGETURNER</td>
<td>24</td>
<td>20</td>
<td>1</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>OPENFOODFACTS</td>
<td>155</td>
<td>20</td>
<td>2</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>FREEOTP</td>
<td>28</td>
<td>19</td>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OANDBACKUP</td>
<td>57</td>
<td>3</td>
<td>3</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>CALCULATE!</td>
<td>101</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>ANYMEMO</td>
<td>139</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>0</td>
</tr>
</tbody>
</table>

**TM**s = Test Mocks  
*AFTMs* = Android Framework Test Mocks  
*ATMs* = App Test Mocks  
*TLPTMs* = Third-party Library Test Mocks

30% of the test mocks model the Android framework

41% model Android framework+third party libraries
Future Work

- Implement MOKA for Android apps
- Evaluate MOKA on real-world apps
- Evaluate re-use across versions and with new tests
- Evaluate time savings of MOKA’s test mocks
Summary

Software Environment

Test Mocks

MOKA Overview

Preliminary Empirical Study - Test Mocks

<table>
<thead>
<tr>
<th>Name</th>
<th>Time</th>
<th>Size</th>
<th>CPU Time</th>
<th>Test Time</th>
<th>47% FPM</th>
<th>57% FPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATS</td>
<td>226</td>
<td>40.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.6</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.5</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.4</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.3</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.2</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.1</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 1.0</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.9</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.8</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.7</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.6</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.5</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.4</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.3</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.2</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.1</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STTF 0.0</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.6</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.5</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.4</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.3</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.2</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.1</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 1.0</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.9</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.8</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.7</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.6</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.5</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.4</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.3</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.2</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.1</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>STT 0.0</td>
<td>191</td>
<td>35.7</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

30% of the test mocks model the Android framework
41% model Android framework+third party libraries