Successive Refinement of Models for Model-Based Testing to Increase System Test Effectiveness

Ceren Şahin Gebizli  
Vestel Electronics, R&D  
ceren.sahin@vestel.com.tr

Hasan Sözer  
Özyeğin University  
hasan.sozer@ozyegin.edu.tr

Ali Özer Ercan  
Özyeğin University  
ali.ercan@ozyegin.edu.tr

11 April 2016, Chicago, IL, USA
Outline

• Testing Challenges in Consumer Electronics Domain
• Model-based Testing and System Models
• Overall Approach
• Model Updates and Case Study
• Results and Future Work
Challenges

• Short time-to-market
• Limited resources
• Large code base
  o 5M LOC in total
• Large models
  o Thousands of states and transitions
• Importance of User Perception
Model-based Testing (MBT)

- Automated test case generation based on models that represent the desired behavior of the system under test (SUT)

Effective test case generation:
- Focus on features that are **mostly used**
- Focus on scenarios that are mostly **error-prone**
- Focus on scenarios that reveal **different** failures
Hierarchical Markov chains defined with the **MaTeLo tool**
(http://www.all4tec.net)
Overall Approach

- Update system models based on:
  - Frequency of usage by the end-users
  - Estimated risk of failure based on static analysis
  - Estimated risk of failure based on dynamic analysis

- (Re)generate and execute test cases
System Model Updates

- First assignments of transition probabilities based on number of visits recorded in the usage profile

\[
v_i / \sum_{i=0}^{n-1} v_i
\]

- Next: second & third updates based on estimated risk of error
Update based on Risk of Error

- Risk estimations:
  - Static analysis: Ratio of static code analysis alerts
  - Dynamic analysis: Ratio of memory leaks
- Example: Update of the system model after the probability of error for state $s$ is calculated as 0.2
Industrial Case Study

• Initial model was previously developed by the software test group in the company.

Data Collection and Estimations;
• Usage Profile
• Static Analysis*
• Memory Profile

* Performed with the Klockwork tool (http://www.klocwork.com/)
Model Updates
## Iterations

<table>
<thead>
<tr>
<th>Software Module</th>
<th>Iteration 1</th>
<th></th>
<th>Iteration 2</th>
<th></th>
<th>Iteration 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of Visits</td>
<td></td>
<td># of Warnings</td>
<td></td>
<td>Memory leak (MB)</td>
<td></td>
</tr>
<tr>
<td>Portal</td>
<td>1900</td>
<td></td>
<td>18</td>
<td></td>
<td>40.855</td>
<td>0.218</td>
</tr>
<tr>
<td>Youtube</td>
<td>2250</td>
<td></td>
<td>18</td>
<td></td>
<td>89.380</td>
<td>0.477</td>
</tr>
<tr>
<td>HBBTV</td>
<td>500</td>
<td></td>
<td>6</td>
<td></td>
<td>8.846</td>
<td>0.047</td>
</tr>
<tr>
<td>MBR Video</td>
<td>1750</td>
<td></td>
<td>2</td>
<td></td>
<td>22.375</td>
<td>0.119</td>
</tr>
<tr>
<td>MBR Audio</td>
<td>400</td>
<td></td>
<td>1</td>
<td></td>
<td>4.167</td>
<td>0.022</td>
</tr>
<tr>
<td>MBR Picture</td>
<td>100</td>
<td></td>
<td>1</td>
<td></td>
<td>3.980</td>
<td>0.021</td>
</tr>
<tr>
<td>PVR</td>
<td>1000</td>
<td></td>
<td>3</td>
<td></td>
<td>9.351</td>
<td>0.05</td>
</tr>
<tr>
<td>Channel List</td>
<td>1750</td>
<td></td>
<td>3</td>
<td></td>
<td>2.516</td>
<td>0.013</td>
</tr>
<tr>
<td>EPG</td>
<td>2000</td>
<td></td>
<td>2</td>
<td></td>
<td>3.094</td>
<td>0.017</td>
</tr>
<tr>
<td>Teletext</td>
<td>1250</td>
<td></td>
<td>1</td>
<td></td>
<td>1.675</td>
<td>0.009</td>
</tr>
<tr>
<td>HDMI-SCART</td>
<td>100</td>
<td></td>
<td>1</td>
<td></td>
<td>1.002</td>
<td>0.005</td>
</tr>
</tbody>
</table>
Results and Future Work

- Reduction in the number of test cases
- Detection of new faults

<table>
<thead>
<tr>
<th>Iteration #</th>
<th># of Test Cases</th>
<th>Test Execution Time (hr)</th>
<th># of Faults Detected</th>
<th># of New Faults Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>847</td>
<td>4</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>809</td>
<td>4</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>136</td>
<td>1.5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>117</td>
<td>1.5</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

- Different types of fault/error states
- Test execution time vs. fault detection trade-off
  - Eliminating iterations
  - Updating only once by aggregating estimations
Conclusions

• Challenging context of the consumer electronics domain

• Testing focus on mostly used, error-prone scenarios

• An iterative model refinement approach

• Detection of new faults in each iteration
Thanks! Questions welcome..