

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

ICSTW TAIC-PART 2016: 11th Testing: Academic
and Industrial Conference – Practice and Research
Techniques (TAIC PART)

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

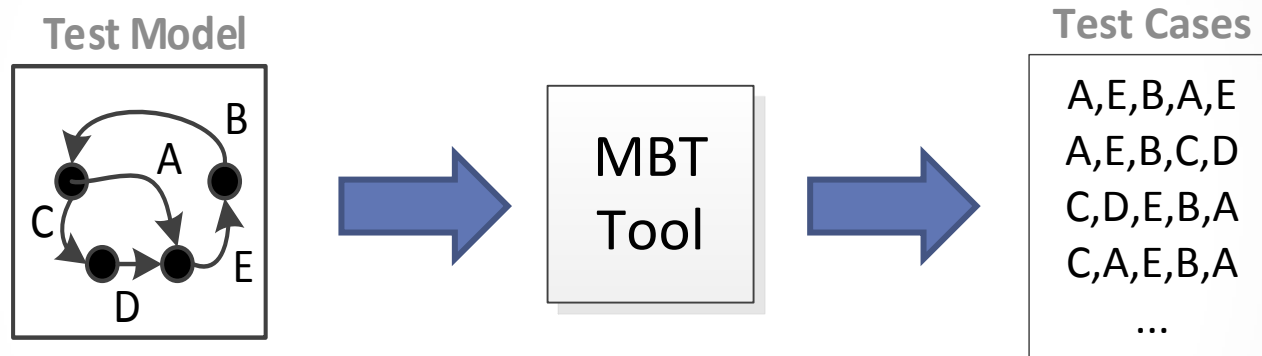
Challenges



- Short time-to-market
- Limited resources
- Large code base
 - 5M LOC in total
- Large models
 - Thousands of states and transitions
- Importance of User Perception

Model-based Testing (MBT)

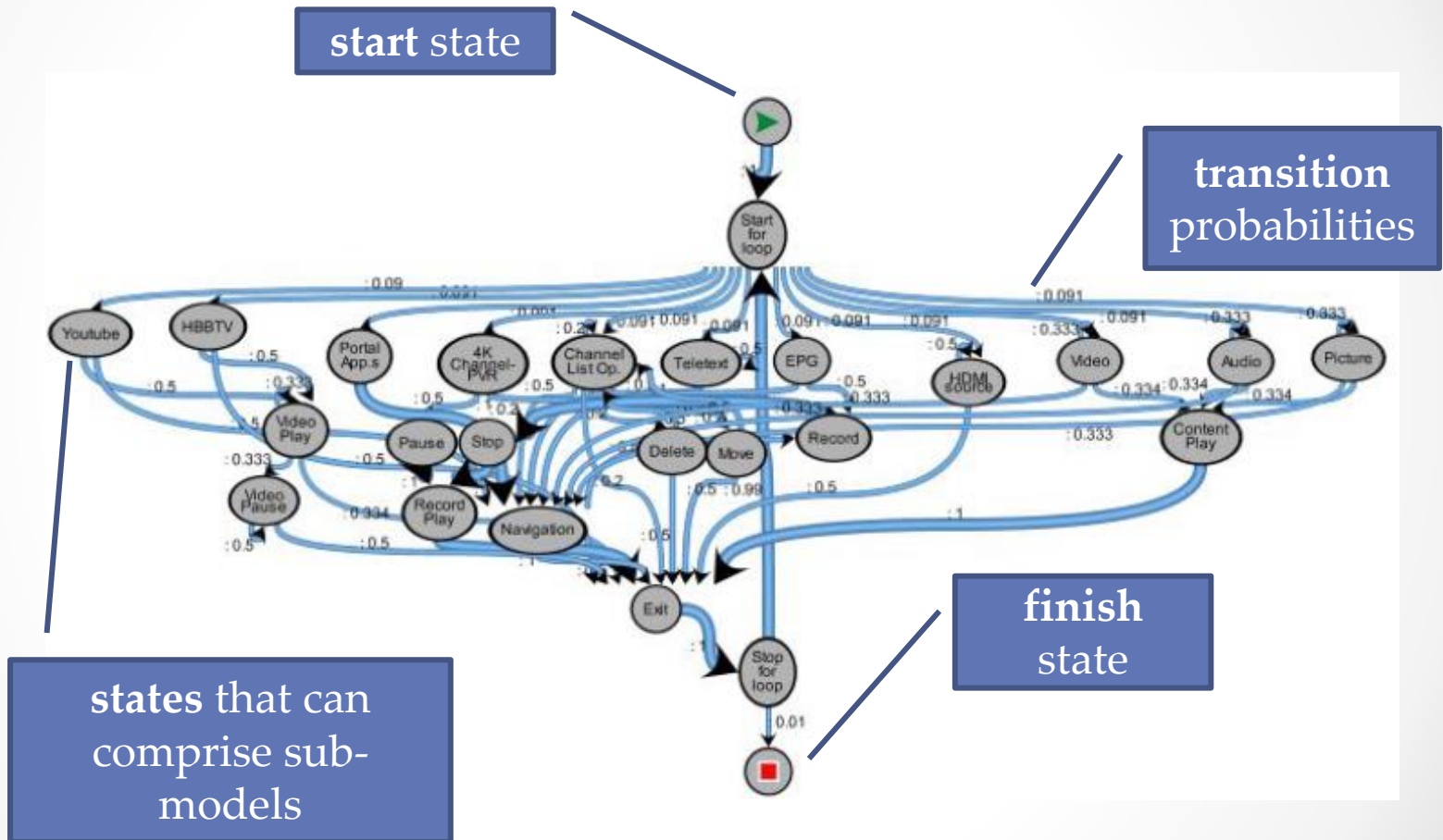
- Automated test case generation based on models that represent the desired behavior of the system under test (SUT)



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

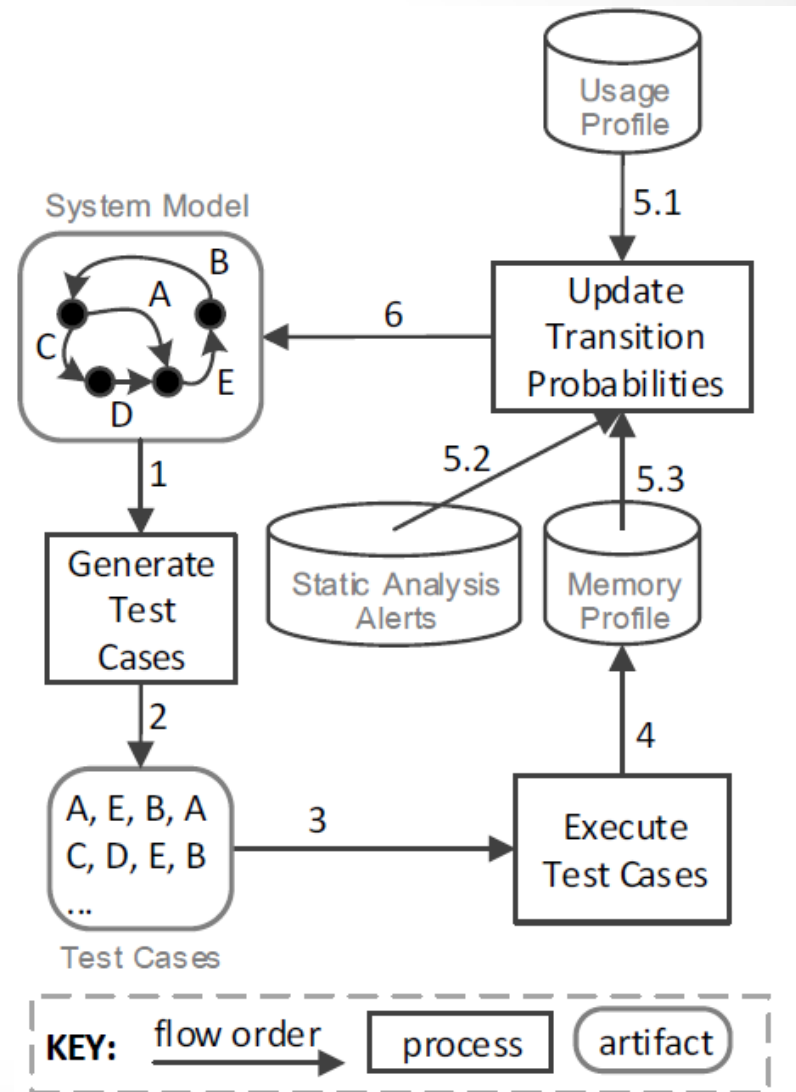
System Models used for MBT



Hierarchical Markov chains defined with the **MaTeLo** tool
(<http://www.all4tec.net>)

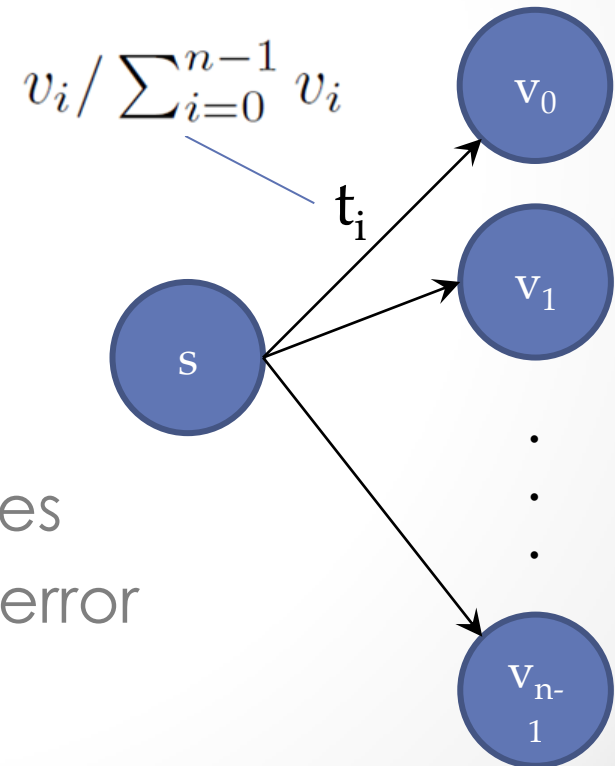
Overall Approach

- 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

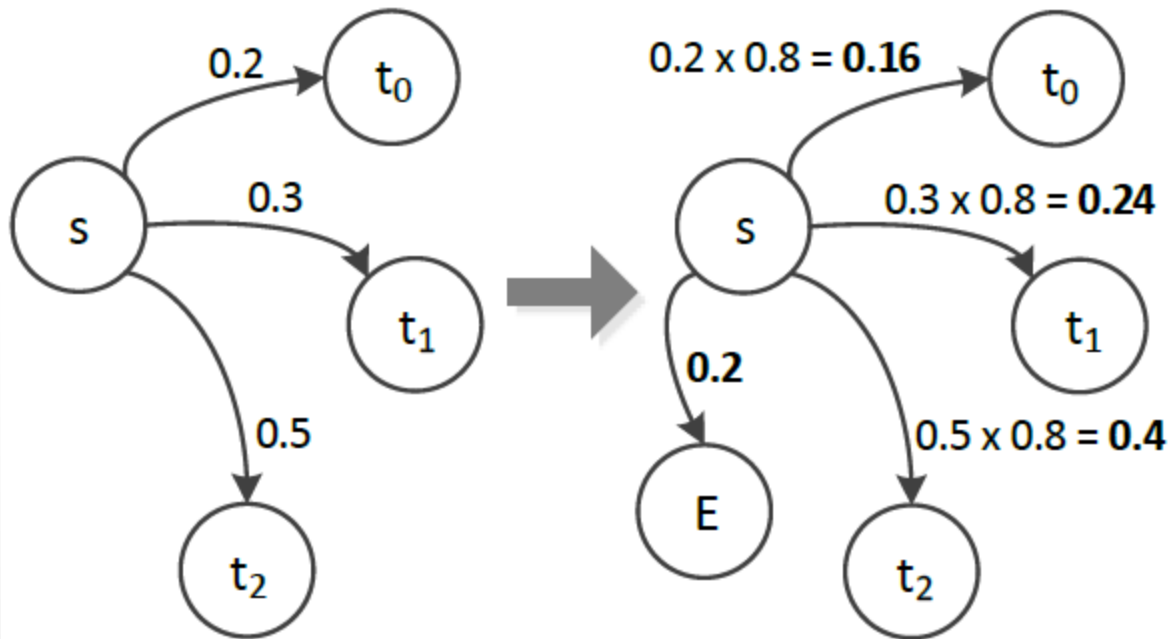
- First assignments of transition probabilities based on number of visits recorded in the usage profile



- 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.

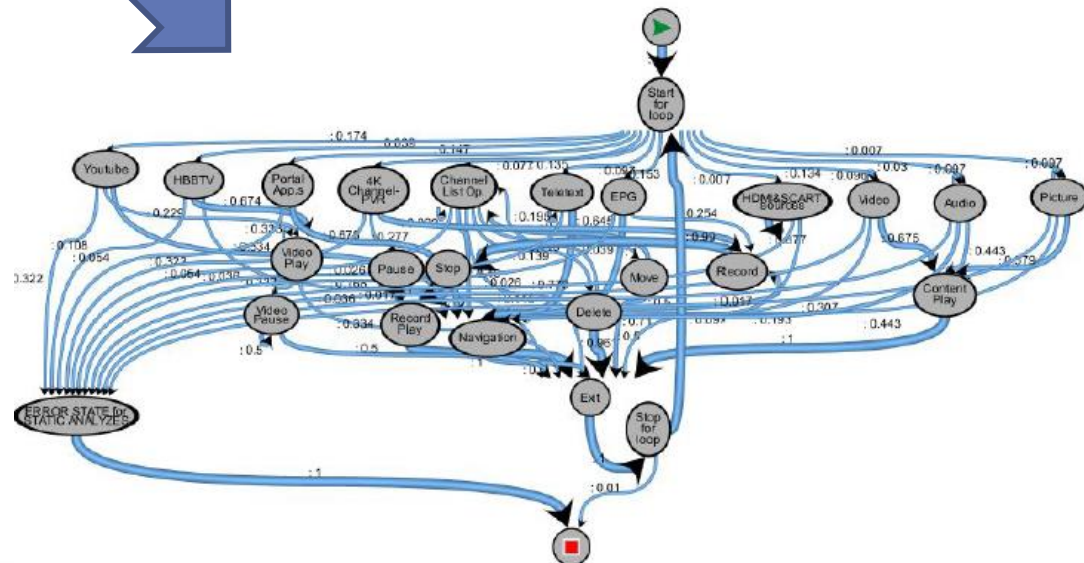
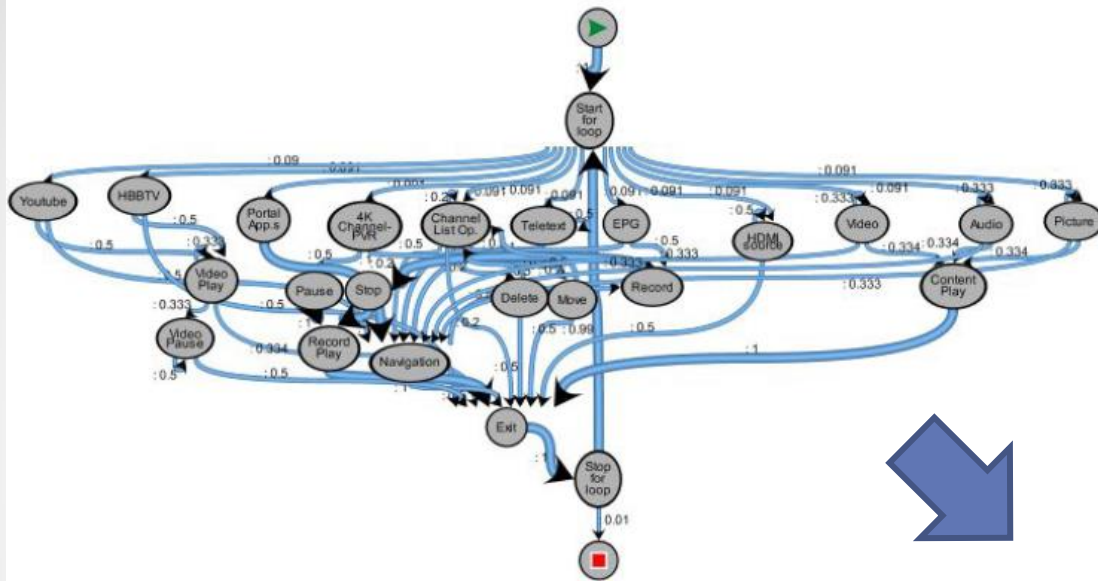
Data Collection and Estimations;

- Usage Profile
- Static Analysis*
- Memory Profile



* Performed with the **Klockwork tool** (<http://www.klocwork.com/>)

Model Updates



Iterations

Software Module	Iteration 1		Iteration 2		Iteration 3	
	# of Visits	Calculated Prob.	# of Warnings	Calculated Prob.	Memory leak (MB)	Calculated Prob.
Portal	1900	0.146	18	0.322	40.855	0.218
Youtube	2250	0.173	18	0.322	89.380	0.477
HBBTV	500	0.038	6	0.108	8.846	0.047
MBR Video	1750	0.134	2	0.036	22.375	0.119
MBR Audio	400	0.03	1	0.017	4.167	0.022
MBR Picture	100	0.007	1	0.017	3.980	0.021
PVR	1000	0.076	3	0.054	9.351	0.05
Channel List	1750	0.134	3	0.054	2.516	0.013
EPG	2000	0.153	2	0.036	3.094	0.017
Teletext	1250	0.096	1	0.017	1.675	0.009
HDMI-SCART	100	0.007	1	0.017	1.002	0.005

Results and Future Work

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

Iteration #	# of Test Cases	Test Execution Time (hr)	# of Faults Detected	# of New Faults Detected
0	847	4	7	2
1	809	4	9	2
2	136	1.5	3	1
3	117	1.5	3	2

- 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..

