Experiences from model-based GUI testing in smartphone camera and messaging development

Rupesh Dev
Prove Expertise
Finland
rupesh.dev@prove.fi
www.prove.fi

Mika Katara
Department of Software Systems
Tampere University of Technology
Finland
mika.katara@tut.fi

Special thanks to the former TEMA team at Tampere University of Technology: Henri Heiskanen, Antti Jääskeläinen, Mika Maunumaa, Mika Mäenpää, Antti Nieminen, Tuomas Pajunen, Tommi Takala, and Heikki Virtanen

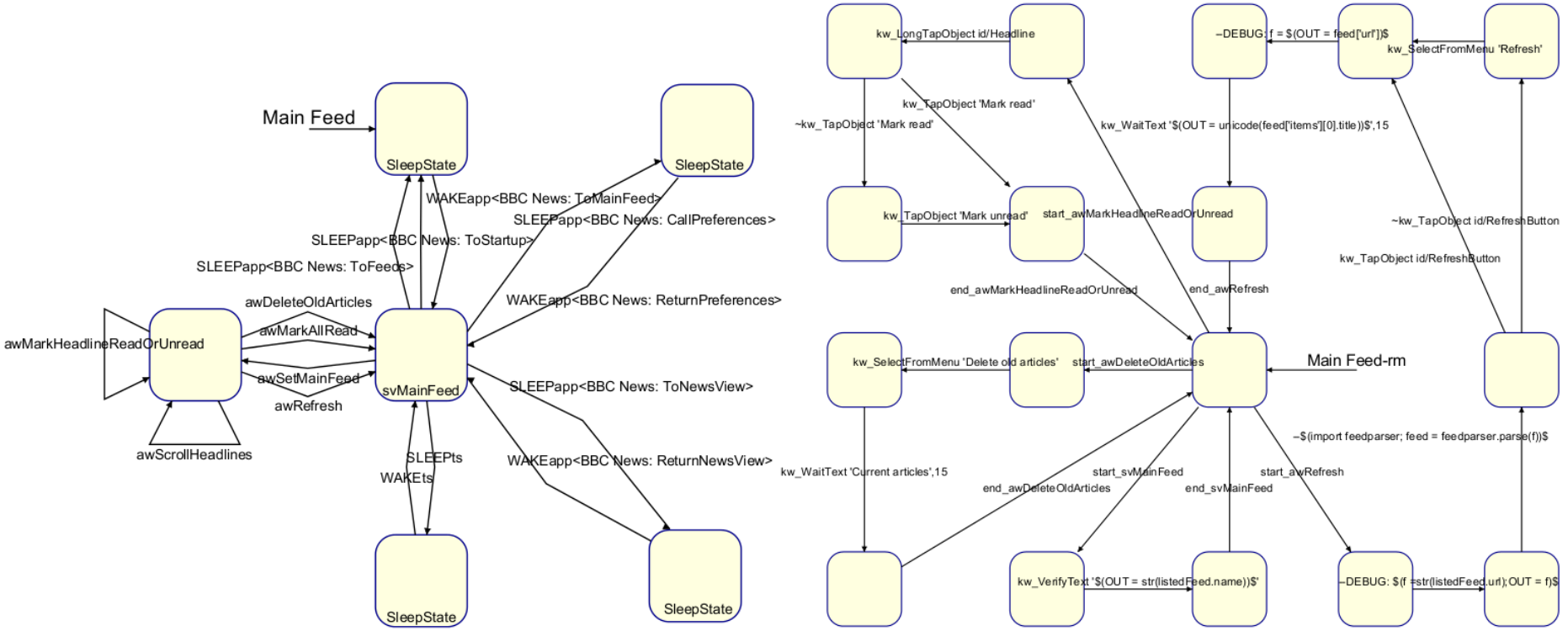
TAMPERE UNIVERSITY OF TECHNOLOGY

What are We Looking For?

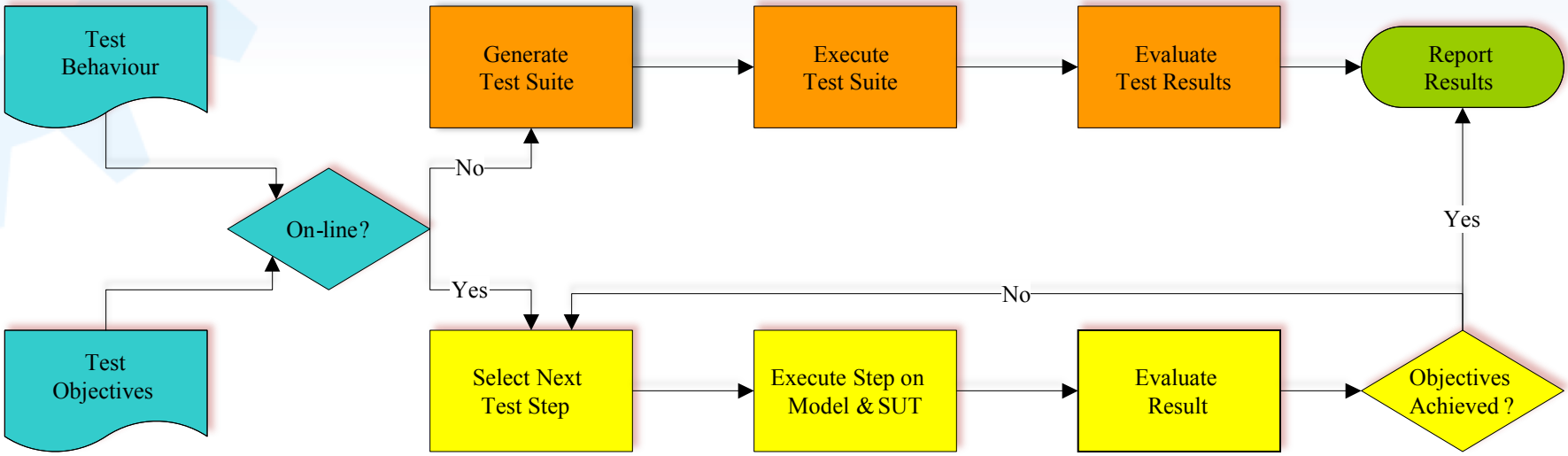
Bugs that affect smartphone users, i.e. almost everybody

How?

On-line model based testing using models describing what the user can do with the GUI and how the apps interact



Online vs. Off-line Testing



Adapted from: Alan Hartman, Mika Katara, and Sergey Olvovsky. Choosing a Test Modeling Language: a Survey. In Proceedings of the Haifa Verification Conference 2006, IBM Haifa Labs, Haifa, Israel, October 2006. Number 4383 in Lecture Notes in Computer Science, pages 204-218. Springer 2007.

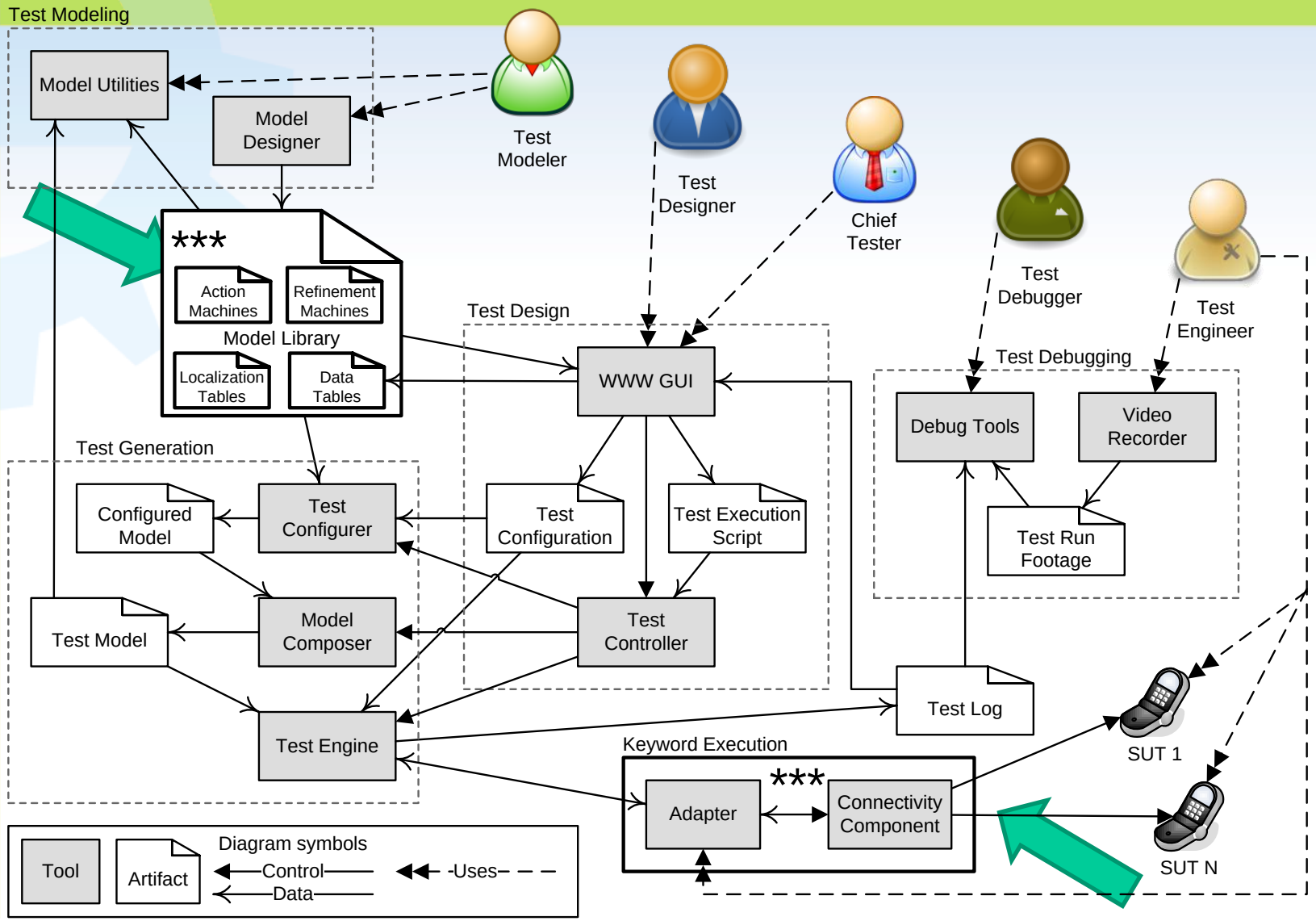
Obstacles and Opportunities for MBT

- Practitioners are willing to try out new tools that might help them
  - Wide variety of open-source testing tools already used (agile unit testing, continuous integration, etc.)
- Practitioners are not willing to invest heavily on modeling or specification in general
- When quality is not a prime consideration, conventional testing methods seem to work reasonably well
- There are areas that are very hard to test using conventional methods (static and linear test cases)
  - Many applications running concurrently and sharing resources may suggest concurrency problems
- Protecting the brand: End users who experience application hang-up/crashing problems etc. may post their bad experiences to the Internet

TEMA Toolset – Hiding Innate MBT Complexity

- Since testers don't want to deal directly with models or test generation algorithms, we have abstracted the algorithms out in our web GUI
- TEMA web GUI is testers' interface with the test server, used for designing and managing test configurations, running and tracking actual tests, and managing test model packages
- This boils down to allowing testers to just choose what they want to test, what physical device they want to run their tests on, etc.
- Organizational impact:
  - Need for test design has diminished, only test configurations (that may involve use cases) have to be created
  - Modeling is imperative
    - High-level models can be reused, but SUT-specific refinements must be created case by case for each product in the product family, for instance

TEMA Tool Architecture



Test Suite Maintenance

A major problem with conventional test automation, especially in the GUI context, is the maintenance of the test suites

In the worst case, you have to modify each test in your suite whenever something changes in the SUT (System Under Test)

Using models, test suites are generated automatically, and you only have to change your model

Or few of the component models

Keywords and Action Words

- Action words describe the user's actions at a high level of abstraction
  - Send an SMS, answer a call, add a new contact etc.
  - Used in high-level models (action machines)
- An action word is translated to a sequence of keywords (keystrokes) for menu navigation, text inputting etc.
  - Some action words can have multiple keyword sequences implementing them
  - Keywords are used in low-level models (refinement machines)

To achieve a good separation of concerns, we use action words and keywords in separate models at different levels of abstraction

Action machines containing action words are composed with refinement machines containing keywords

The resulting composite model is input to the tools executing the model i.e., generating the test cases

To avoid state space explosion, this has been implemented using an on-the-fly algorithm

Example Test Models

Symbian Camera application, action machine

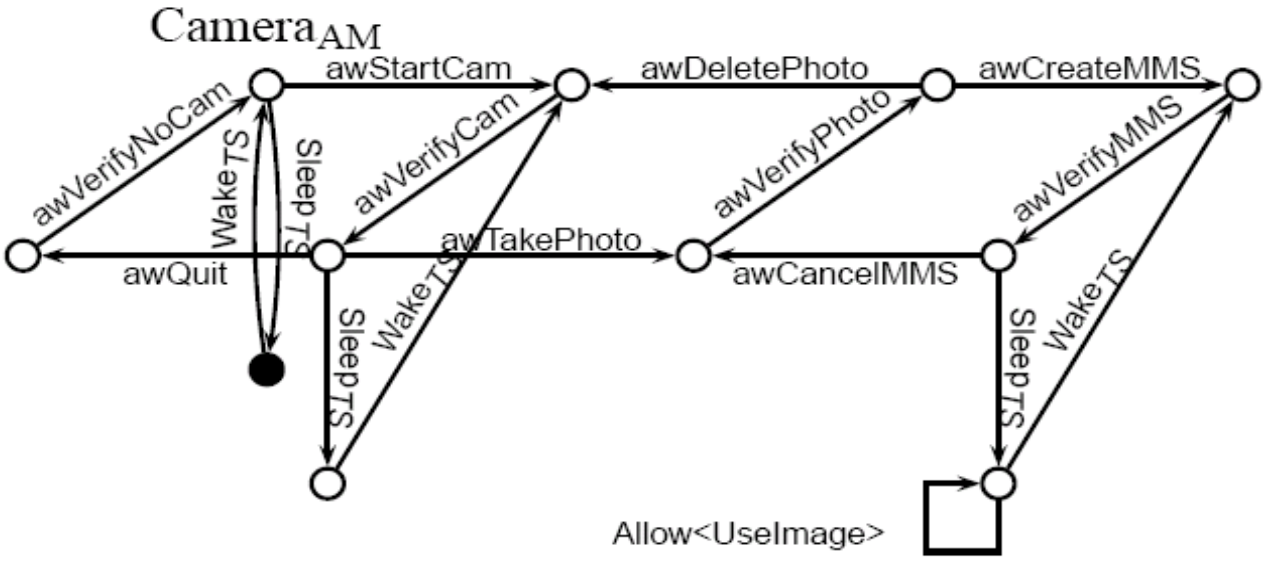


Illustration: Antti Kervinen/TUT

Symbian Camera application, refinement machine

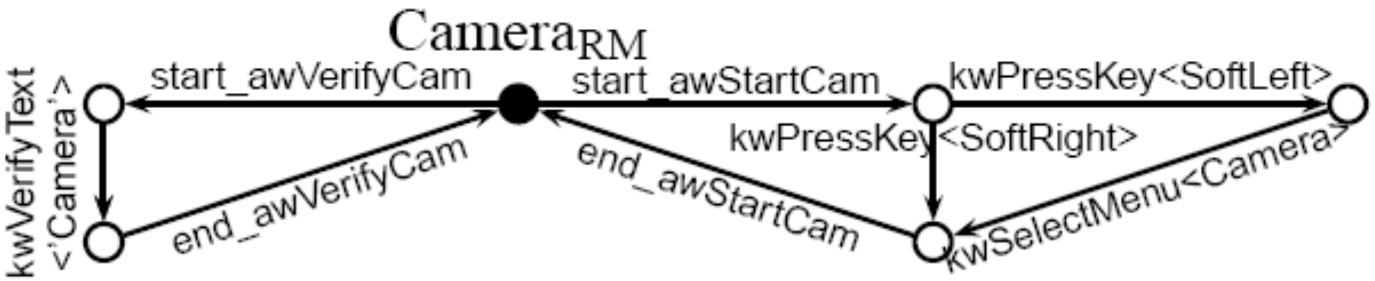


Illustration: Antti Kervinen/TUT

Case: Symbian (TEMA Starting Point)

- Built-in applications in Symbian smartphones, such as Gallery, Music Player, Flash Player, Notes, Voice Recorder, Contacts and Messaging
- Keyword execution using proprietary and commercial test automation tools
  - Optical character recognition was used for verifications, which caused some reliability and maintenance issues
- 21 defects of different severities and priorities were found
  - Some of these defects existed in more than one smartphone model
  - The most severe of the defects caused the phone to hang with “System error” message on the display
  - About two thirds of the defects were discovered while modeling (reverse engineering), and the remaining third by execution (dynamic testing)
  - Most of the defects had already been previously found in traditional testing (both manual and automatic test execution), but they had not been fixed for some reason
    - However, there were also some that were totally new
  - Some of the defects were related to **concurrency issues**: performing some multimedia-related functionality in one application and then switching to another application caused unexpected behavior in some circumstances
  - In addition to defects found in applications, some were found in test automation tools, which was considered rather surprising, as these tools were quite mature

Case Symbian Messaging and Camera Testing (Nokia E7 & N8)

Project Starting Point

Implemented by Mr. Rupesh Dev as his Master's Thesis work

Goal: to show benefits of model-based testing over existing keyword-based automation practices

Practical limitations:

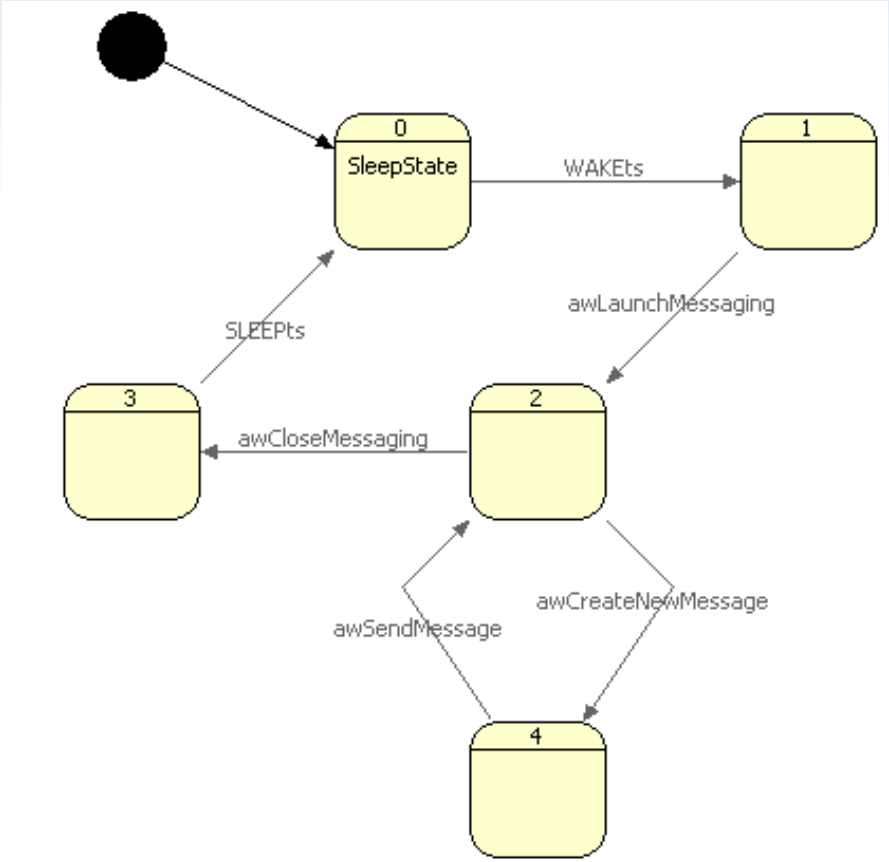
Access the SUTs using TDriver (Testability Driver)

- http://wiki.meego.com/Quality/QA-tools/TDriver
- There were previous experiences on using TEMA with TDriver only on Linux-based SUTs

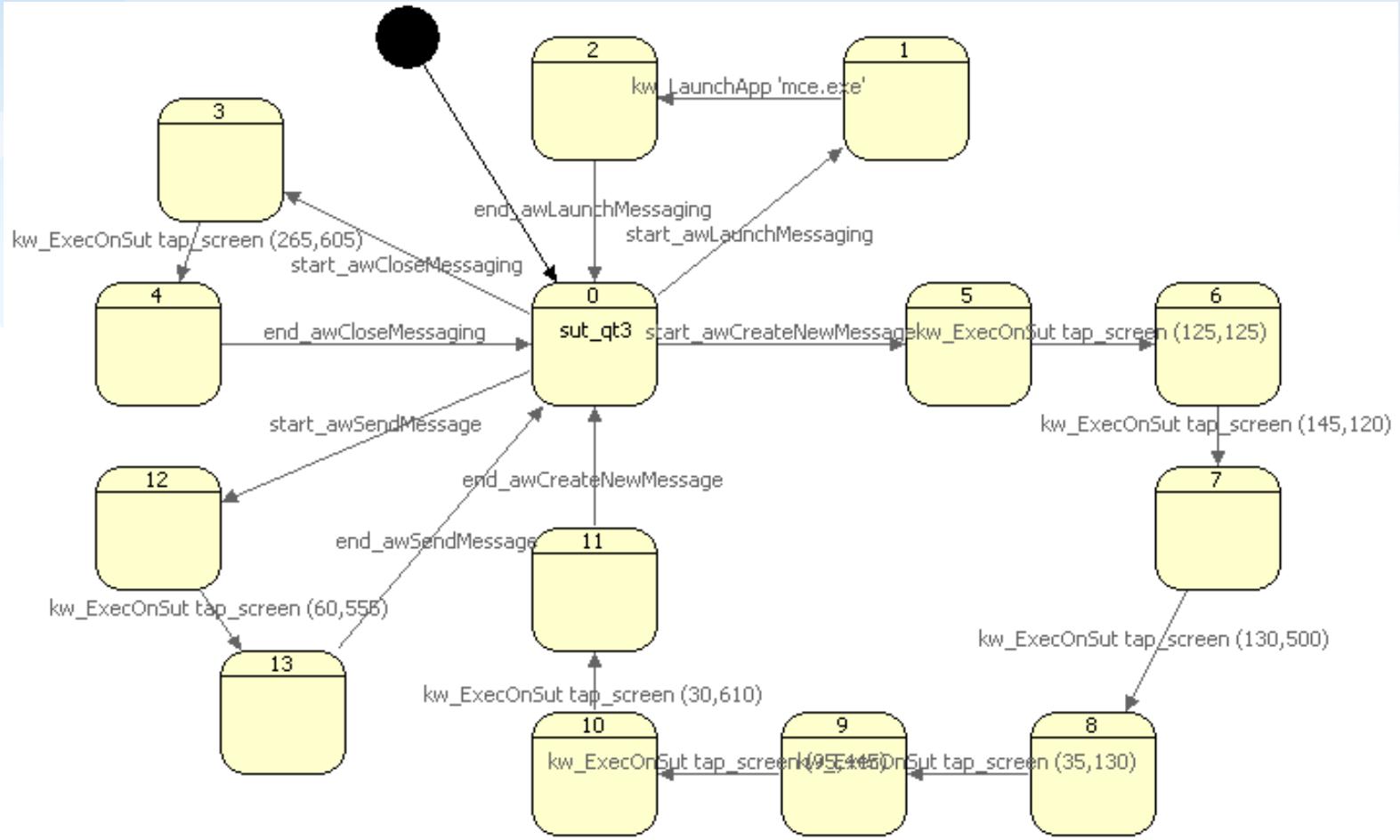
WLAN connection preferred over Bluetooth or USB cables

- Reliable connection, multiple phones, greater distance

Action Machine for the Sender



Refinement Machine for the Sender



Lessons Learned

- The entire test run performed were divided in three different combinations
  - The first combination included only testing of camera based actions
  - The second combination included only messaging related tasks
  - In the third one both combinations were tested jointly
- Successful execution of a single model to multiple SUTs at the same time
  - In other words, we were able to execute one or more use cases on two different phones simultaneously
- Camera based test run automated the actions like image capturing and video recording in a loop
  - The test run successfully captured 1000 still images and around 800 videos in three hours

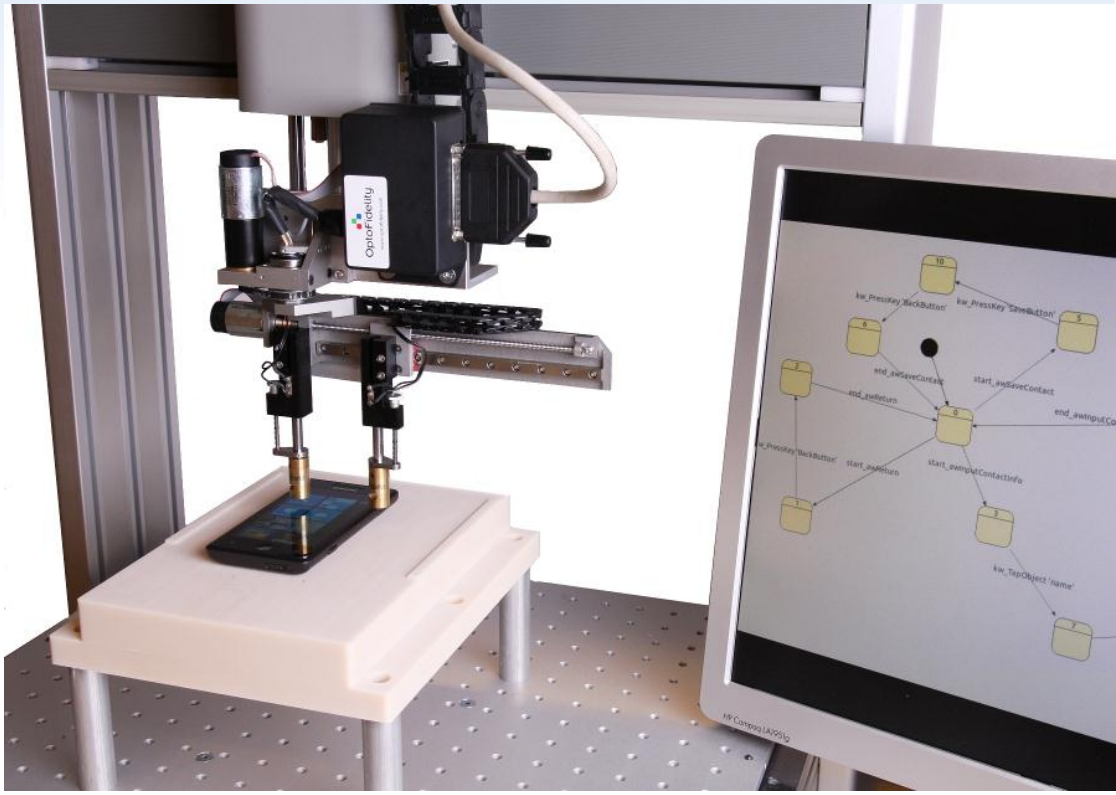
- Similarly, messaging related test run automated the text message sending procedure
  - One of the SUT composed the text message, and sent to the other SUT
  - The other SUT checked message and sent back the received confirmation
- Multi-phone MBT really works
- Suggestions for improving TEMA toolset in the future
  - Model management should be improved in TEMA Model Designer
  - Easier installation
  - Better documentation needed

Conclusions

- Model-based GUI testing of smartphone applications is starting to be an attractive option compared to existing keyword-based tools where test cases are designed manually
- An online tool enables robustness testing that can explore the model and gain a lot coverage
- Once the adaptation and connectivity issues are solved, the problem is in creating effective models that are easy to maintain
- TEMA is an academic prototype, but has many features targeted towards industrial-size problems
  - Previous case studies have showed good performance regarding scalability (huge models)
- Rupesh's thesis available at
  - http://urn.fi/URN:NBN:fi:tty-2011062014725

Example of Another Type of Adaption: Keyword Execution with a Robot

- Solution for the automated testing of touch display devices
- Simulates real human user interaction with SUT
- The applications are tested in actual devices
- Different sets of robot fingers for device actuation
- Visual verification of the results with a camera and OCV (Optical Character Verification)
- Easy integration with TEMA Toolset



For more information, visit
http://www.optofidelity.com

Acknowledgements for financiers of MBT research at TUT:
ATAC ITEA2 project (2011-2014): Tekes
MBT-MOSE project (2008-2010): Tekes, Nokia, Ixonos, Symbio, Cybercom Plenware, F-Secure, Qentinel, Prove Expertise
AMOEBA-TESTING project (2008-2010): Academy of Finland (grant #121012)