

# Mobile testing automation @F-Secure

Testing Symbian OS application through the GUI



# Symbian OS

- Operating system for embedded devices
  - Mobile phones from Nokia, Sony-Ericsson...
- Different GUI systems available
  - S60, S80 -> Nokia
  - UIQ -> Sony-Ericsson



# Challenges of Symbian OS

- Non-deterministic system
  - No direct control to the system reduces reliability
- Automation using remote controlling
  - Connectivity reduces reliability
  - Causes delays. Timed states( ex. notes ) may be missed
- None of the different GUI platforms provide API for getting the current text on the screen
  - Text recognition from image reduces reliability



# Some requirements for automatic tests

- Tests must be easy to make
- Test maintenance must be very low
- Tests must be dynamic
  - Required by undeterministic system
  - Tests must survive from unexpected changes in the SUT
  - Unexpected change is not (always) a failure



# Dynamic tests with models and state recognition

## Model

- States and links between them.
- Links contain the operations required for transition.
- Conditional path finding

## State recognition

- Symbian OS GUIs are based on views( states )
- We use text comparison
  - Many approaches possible



# Benefits of modelling

- Speeds up manual scripting
  - Easy state transitions
    - GoToState( 'AnotherState' )
    - X \* (Go to state) -> perform required checks
  - Readable short tests -> less maintenance
- Model helps tester to understand the application
- Application changes -> Change the model
- Makes testing easier and reduces maintenance



# Benefits of modelling

- Dynamic transitions
  - Continuously find path to target state from current state
  - Undeterministic behavior can be handled
- Tests can be generated from the model
  - Less layers to maintain
- Many graph –algorithms available, for example:
  - Cheapest first – Find shortest path between states
  - Chinese postman – Find optimal path to check all states
- Supports integration with design and development
  - Model can be imported from UML state diagrams



# Implementation

- Our choice of programming language: Python
  - PC:
    - Testing framework and tests
  - Device
    - Execute commands
    - Get system info( RAM usage, etc.. )
    - Standalone test scripts for simple regression tests
- Using open-source components
  - If something doesn't work, we can fix it.



# Benefits of Python

- Fast to learn
    - Simple and clear syntax
  - Expressive and powerful
    - Huge amount of extension libraries available, for example:
      - Image analysis and text recognition
      - Model visualization
      - Test summary generation
  - Works in PC and phone
    - Same language for everything.
- Tests are easy to make and maintain



# Questions?

