



# **Robot assisted model-based test automation**

Matti Vuori

Tampere University of Technology  
Department of Pervasive Computing

# In this slide set

- Comparison of robot-assisted model-based testing with traditional instrumentation approach.
- Discussion of benefits and pitfalls of both approaches.
- Goal: orientation for understanding these systems and issues related to their application.



# Benefits of MBT

- Model-based testing is the most advanced form of test automation.
- Instead of static, manually designed test cases it uses models of the target system as a basis for generation tests dynamically.
- The result is an ability to cover very thoroughly the behaviour of a system during testing and an ability to alter test inputs even during long test runs.



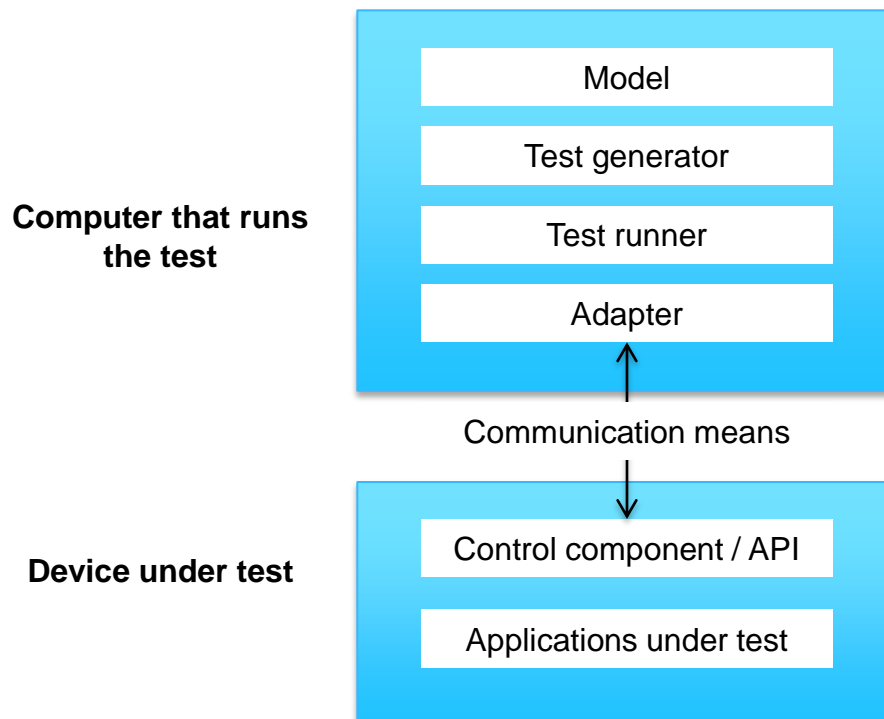
# Two ways of controlling the device under test (DUT)

1. Programmatically – a component in the device that can "press buttons", type text etc.
  2. By a robot hand – one finger (or more) that can press buttons, type text etc. based on what a camera tells about what is available.
- We'll look more into these two alternatives – both of them have advantages and weaknesses.

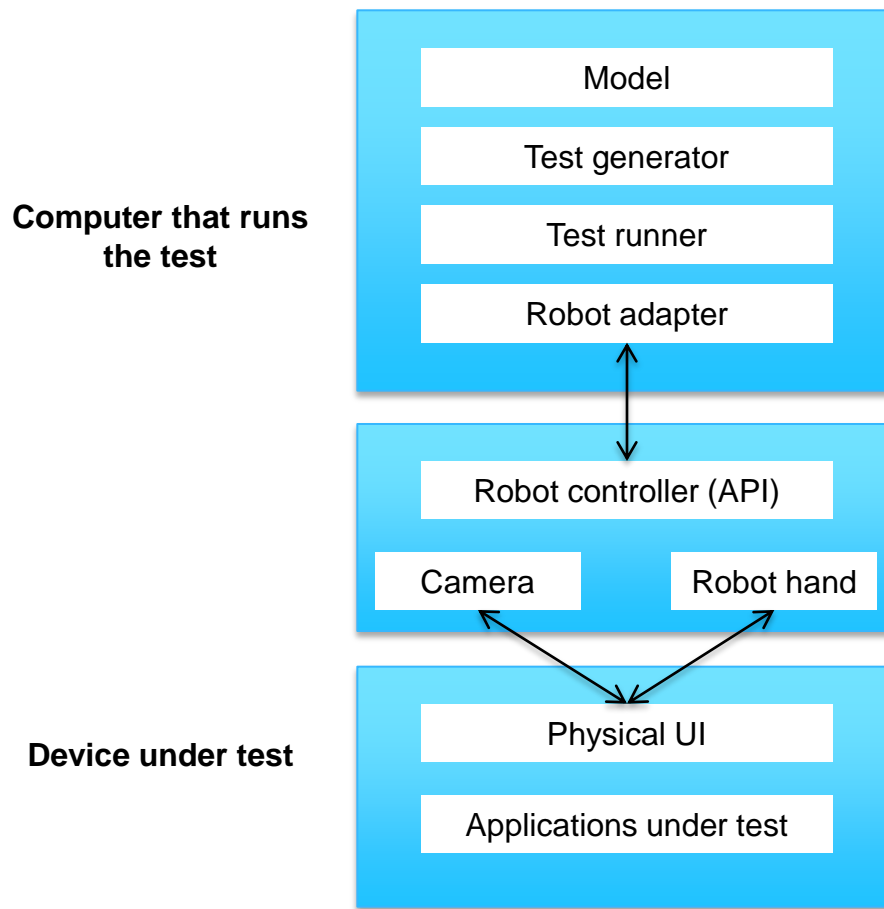


# Test system architecture for programmatic control

**SIMPLIFIED**



# Test system architecture for robot control



**SIMPLIFIED**





# Instrumentation plusses

- Allows API level testing and checking of the internal state of the device/software.
- Easier problem detection and debugging.
- Detection of UI elements more reliable and fast than by OCR or image matching.
- More tolerant to UI changes (fonts, colours).



# Problems of instrumentation



- Traditionally test automation is based on instrumenting the test target with extra software that feeds inputs to it and checks how the system responds.
- Or there might be a test interface that is not used on production use.
- Both create an extra level of control that can compromise the integrity of the system and alter its behaviour.





# Instrumentation minuses in detail



- In-device adapter can be hard to build – a program development task in itself.
- Requires a good API to find elements.
- Invalidates the configuration.
  - Extra software.
  - Slowing down, possible timing changes.
  - Can we trust the test results?
- API level control does not guarantee that the same happens visibly to the user.



# Benefits of robots



- When using a robot to control an application through its interface, such control software is not required and thus test results are more reliable.
- Furthermore, with a robot we can simulate user's behaviour more accurately.



# Various levels of robot functionality

- One finger robot. Tapping, dragging.
- Two finger robot. Two finger gestures.
- Platform that can rotate to change device orientation.
- Etc...
- => More functionality – more cost.



# Robot plusses in detail



- No extra components in the DUT.
- Can be done with any device, no need to access to OS or APIs.
- "What the test sees is what you get".
- Simulates the actual user experience.
- => Very reliable test results.
- => Test arrangements easy to understand, to trust.



# Robot minuses in detail 1/2



- Robots cost a lot of money.
- Availability of test environments limited.
  - Robot reserved for long lasting tests.
- OCR and image matching can be slow and unreliable.
- Can not see below what is visible in the UI.
- Debugging more difficult.
- Debugging tests requires access to robot.  
(Emulation difficult.)



# Robot minuses in detail 2/2



- Changing device orientation difficult with simple robots.
- Mechanical robots can cause hazards.
  - Harm to the device.
  - Hazards to the operator, passers-by.
  - Need to ensure physical safety just as with manufacturing robots.
- Need of supervision, maintenance.
- Complexities to remote testing.



# Why not use real humans instead of a robot?

- So, why not use real humans?
- The kind of testing that robots are used for are
  - Repetitive in nature.
  - Can be executed for long duration.
- => Not suitable for humans.



# Some vendors

- [OptoFidelity](#) provides robot-based test systems that can be used for testing mobile devices.
  - They also have a YouTube channel that show examples of robot-based testing.

