

Domain-Specific and Model-Based Testing

Software Testing Days

August 15, 2006

Antti Kervinen

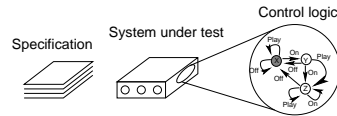
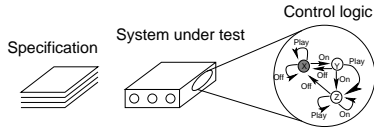
Tampere University of Technology

Domain-Specific and Model-Based Testing

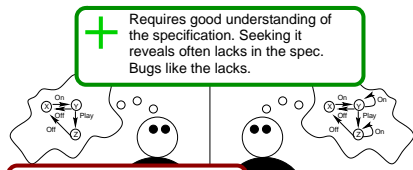
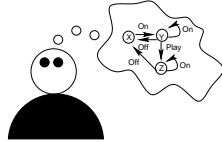
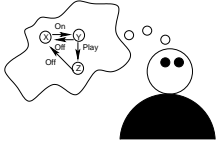
- What is model-based testing
- Pros and cons of model-based testing
- A domain-specific modelling solution

Traditional testing vs. Model-based testing

+ Pros and **-** cons of model-based testing

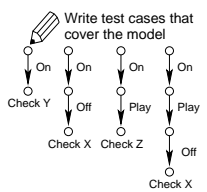
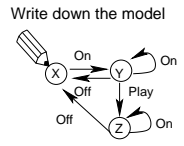
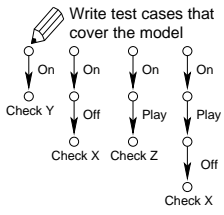


TRADITIONAL | MODEL-BASED
Understand how the SUT should work



+ Requires good understanding of the specification. Seeking it reveals often lacks in the spec. Bugs like the lacks.

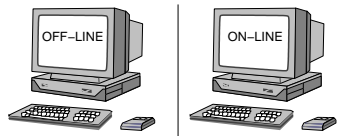
- Modelling is more difficult and time consuming than writing test cases.



+ Lots of bugs can be found already during the modelling.

! Don't be scared off by academic toy examples! Model-based testing is most beneficial with big systems. Yet modelling the whole system might be infeasible, even incomplete models give better coverage than test cases.

Model-based testing:



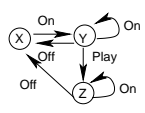
+ Coverage & bug detection: Computer generates test cases which the tester would not have thought of.

- Debugging and analysis: What caused the bug after 1000 steps? Which features are broken?

- Analysis of the coverage: which features were tested and how carefully.

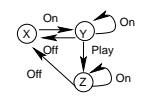
Generate test cases

1. Check X
2. On
3. Check Y
4. On
5. Check Y
6. Play
7. Check Z
8. On
9. Check Z



1. Check X
2. On
3. Check Y
4. On
5. Check Y
6. Play
7. Check Z
8. On
9. Check Z

+ Nondeterminism of the SUT can be lived with.



TEST RUN: execute the test cases

TEST RUN: execute the test cases

TEST RUN: walk in the model

TEST RUN: execute the test cases

TEST RUN: execute the test cases

TEST RUN: walk in the model

+ Maintenance of testware: The model is a compact representation of an infinite number of test cases.

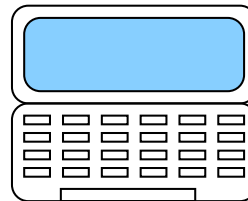
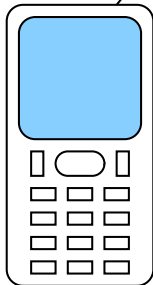
3

Domain

Applications

Telephone Alarm clock Calendar Camera WWW browser

SUT

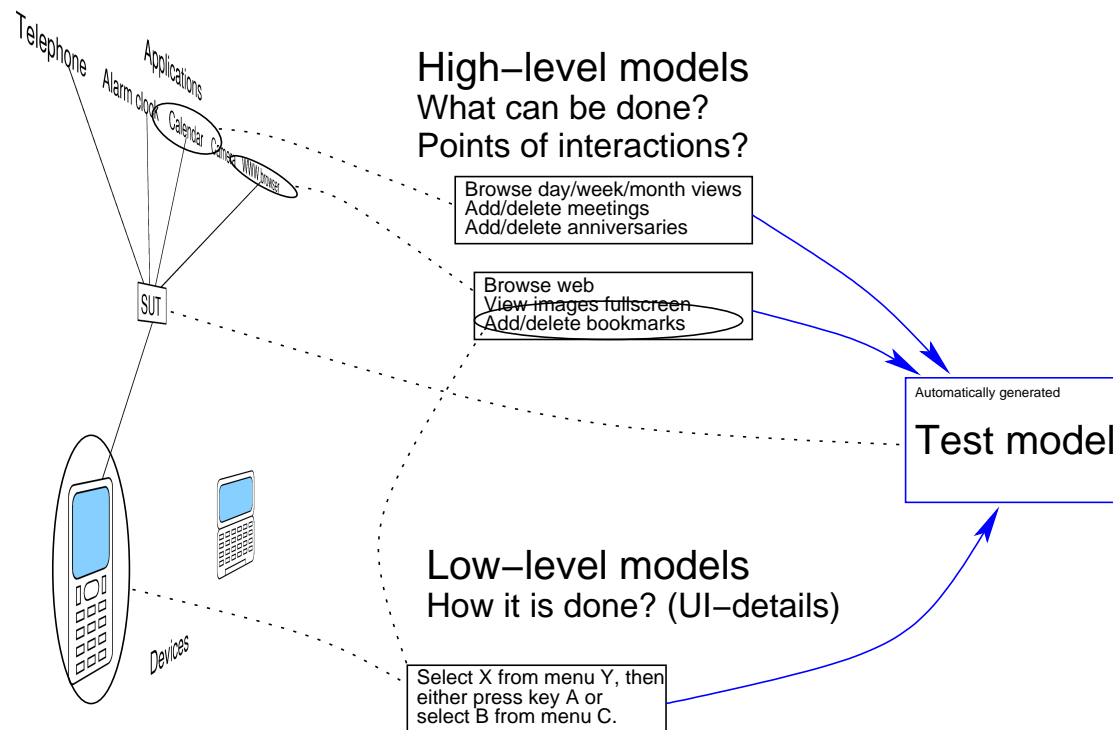


Devices

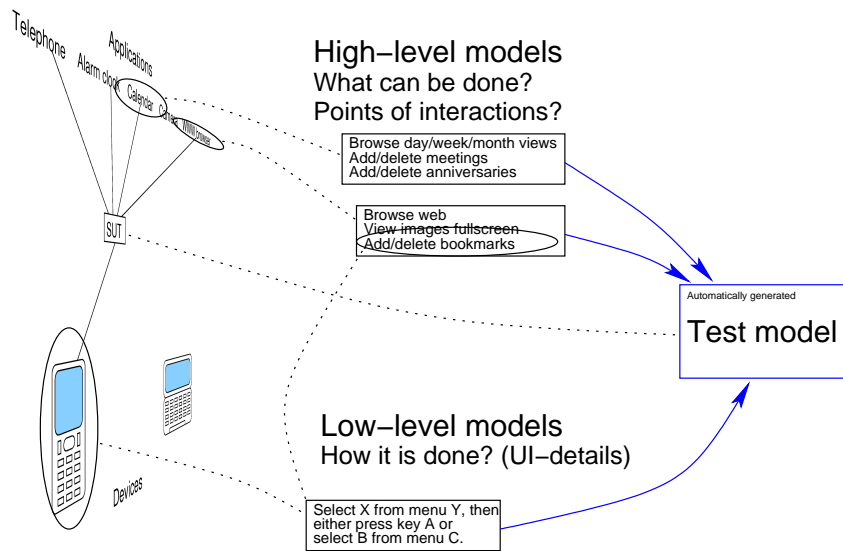
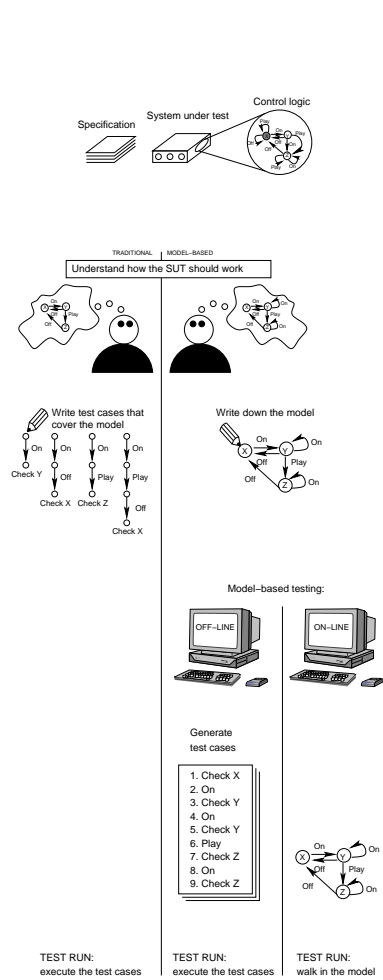
Goals for the domain-specific modelling practice

- Good *coverage* of the system under test
(application interactions, interleaved executions)
- *Reusability* of model components when UI changes
(testing different UIs but similar applications)
- *Reusability* of model components when configuration changes
(testing devices with different set of applications)
- *Reusability* of model components when test objective changes
(The same test models can be used in smoke, requirement coverage and long period tests.)

Domain-specific modelling solution



Traditional testing vs. Model-based testing



Thank you!

antti.kervinen@tut.fi

<http://practise.cs.tut.fi/>