Model-Based Symbian Test Automation using Keywords and Action Words

Dec 19, 2005 @ Mobiilipäivä, TUT

Mika Katara, TTY/OHJ

mika.katara@tut.fi

Mostly joint work with Antti Kervinen,
Mika Maunumaa and Tuula Pääkkönen

Background

- TEMA project
  - Planned duration: Aug 2005 – Jul 2008
  - Personnel: Mika Katara, Antti Kervinen and Mika Maunumaa, Mikko Satama
  - Funding: TEKES, Nokia, Conformiq,

    F-Secure, Plenware, Mercury Interactive
- Disclaimer: the following slides represent the views of the presenter, not necessarily the views of the above or any other parties

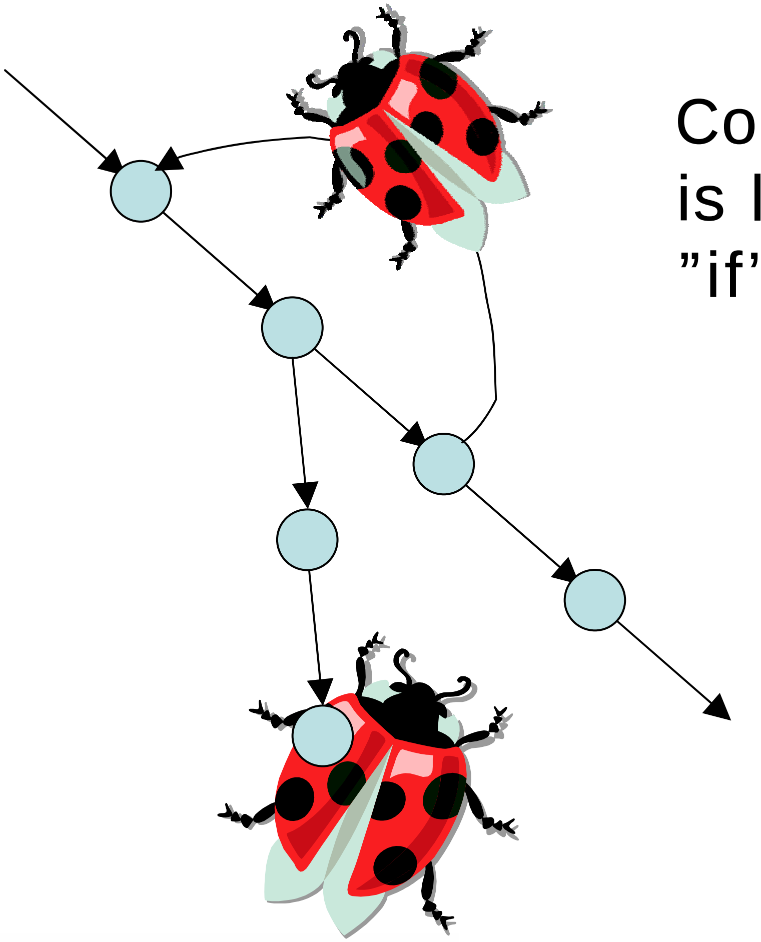
Testing is boring?

- The most boring parts of testing
    1. Executing (regression) test case
    2. Creating test cases
- – The conventional test automation solutions help you to automate the execution of test cases (1)
- – Model-based solutions help you to automate also the generation of tests (1&2)

Finding bugs with automation

- Testing is about finding bugs, isn’t it?
  - – A well known fact is that conventional automation finds fewer bugs that manual testing
    - One reason for this is in the static and linear nature of test scrips
  - – Model-based testing reveals errors already when developing the models
  - – Smart generation heuristics enable finding bugs also in the test execution phase

Finding bugs with automation…



Conventional test automation is like programming without ”if” statements and loops

TAMPERE UNIVERSITY OF TECHNOLOGY
Institute of Software Systems

http://practise.cs.tut.fi

Maintaining test suites

- Another major problem with conventional automation is the maintenance of the test suites
  - In worst case, you have to modify each test in your suite whenever something changes in the SUT (System Under Test)
  - Using models, there are no test suites, and you only have to change your model
    - or few of the component models

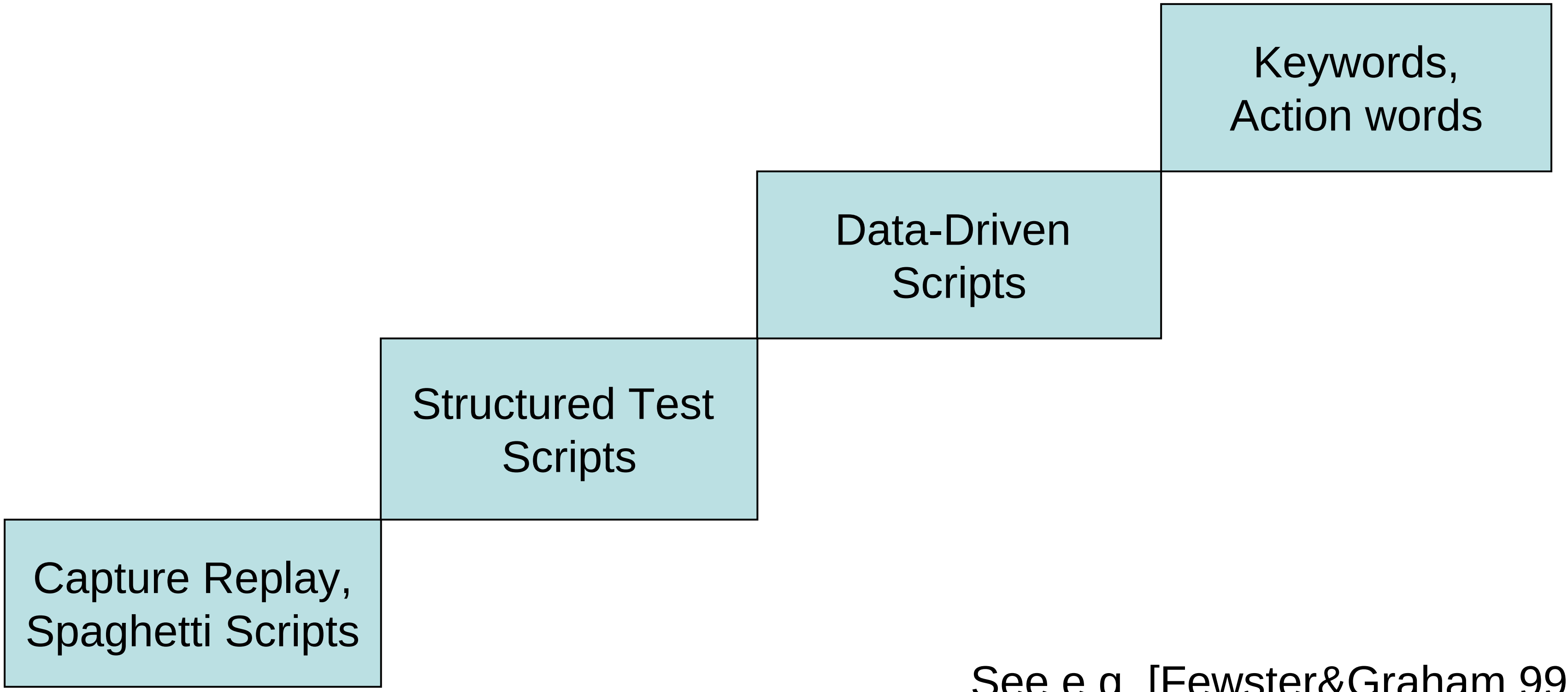
What is a model?

- Simple answer in our case: a state machine
  - More formal answer: labeled transition system (LTS) consisting of states, transitions in between them and labels attached to the transitions
- Another types of models: UML state charts, models enabling generation of different data values etc.

Testing through GUI vs. API

- The software architecture of the SUT almost never supports testing by virtue
  - Test interfaces are very rare
- For system level black-box testing, APIs are the often the second best alternative
- However, suitable APIs do not always exist
  - Testing tools need to be adapted to use an existing API, if suitable one exists
- GUI is often available and the interface is usually standard (e.g. Windows)
- Unfortunately, GUI testing is much harder than API testing
  - Finding certain bugs only succeeds when testing in a certain way, i.e., the bugs you will find will be somewhat dependent on the way you test

Generations of System Level Test Automation



Capture Replay, Spaghetti Scripts

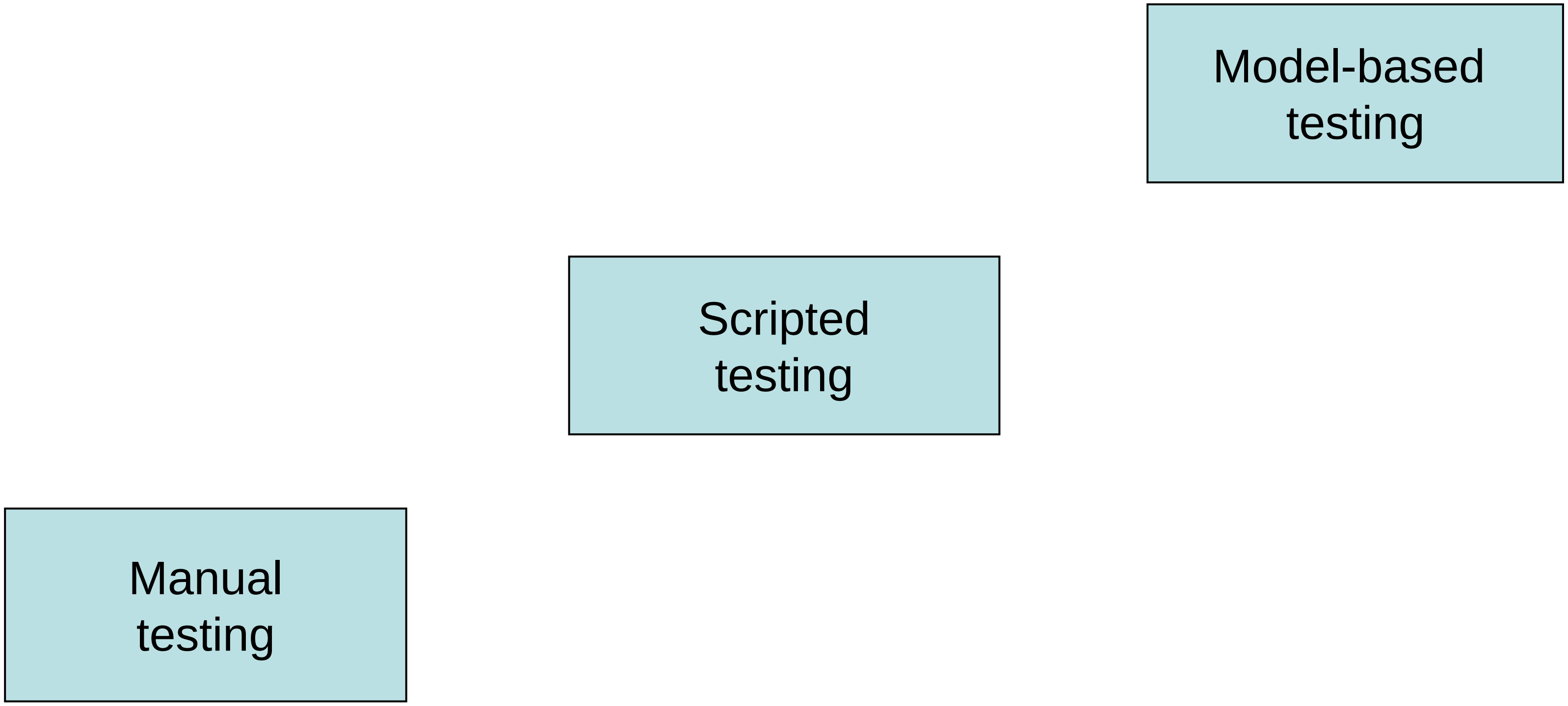
Structured Test Scripts

Data-Driven Scripts

Keywords, Action words

See e.g. [Fewster&Graham 99]

Generations of System Level Test Automation…



Model-based testing

Scripted testing

Manual testing

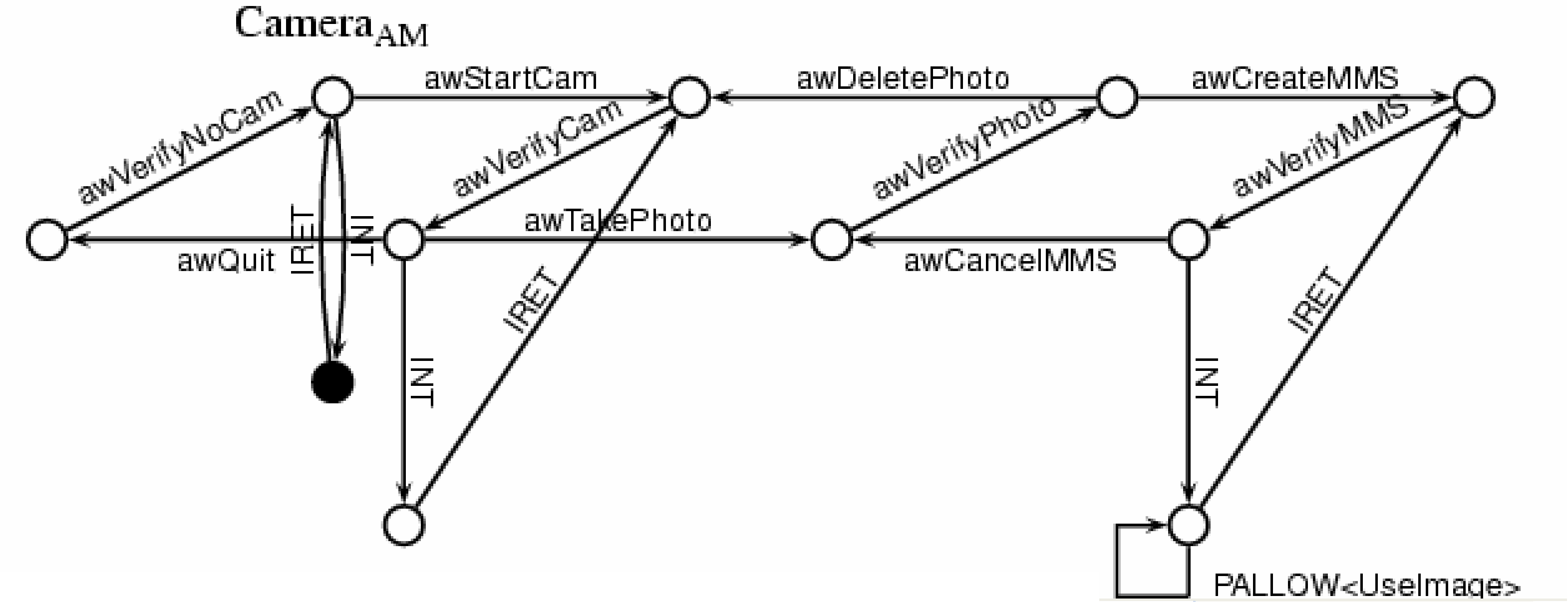
Keywords and Action Words

- Consider a mobile phone with a keypad and a graphical display with menus etc.
- Action words are “user stories” that describe user’s actions at a high level of abstraction:
  - Send an SMS, answer a call, add a new contact, browse the calendar etc.
- Keywords map each action word to a series of key strokes (menu navigation, inputting text etc.)

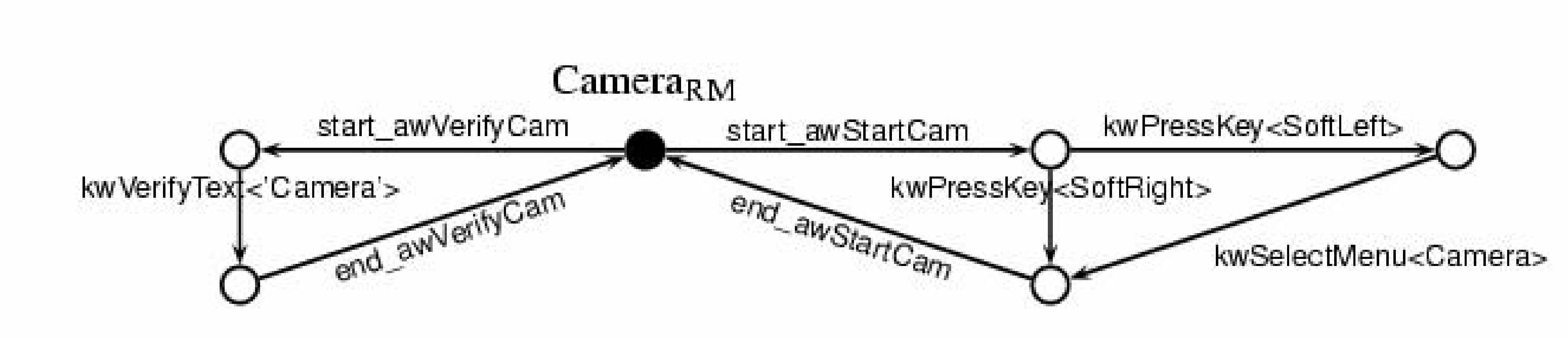
- To achieve a good separation of concerns, we use action words and keywords in separate models at different levels of abstraction
- *Action models* containing action words are composed with *refinement models* containing key words
  - The actions available define a domain-specific language
- The resulting composite model is input to the tools executing the model, i.e. generating the test cases

Example Action Model

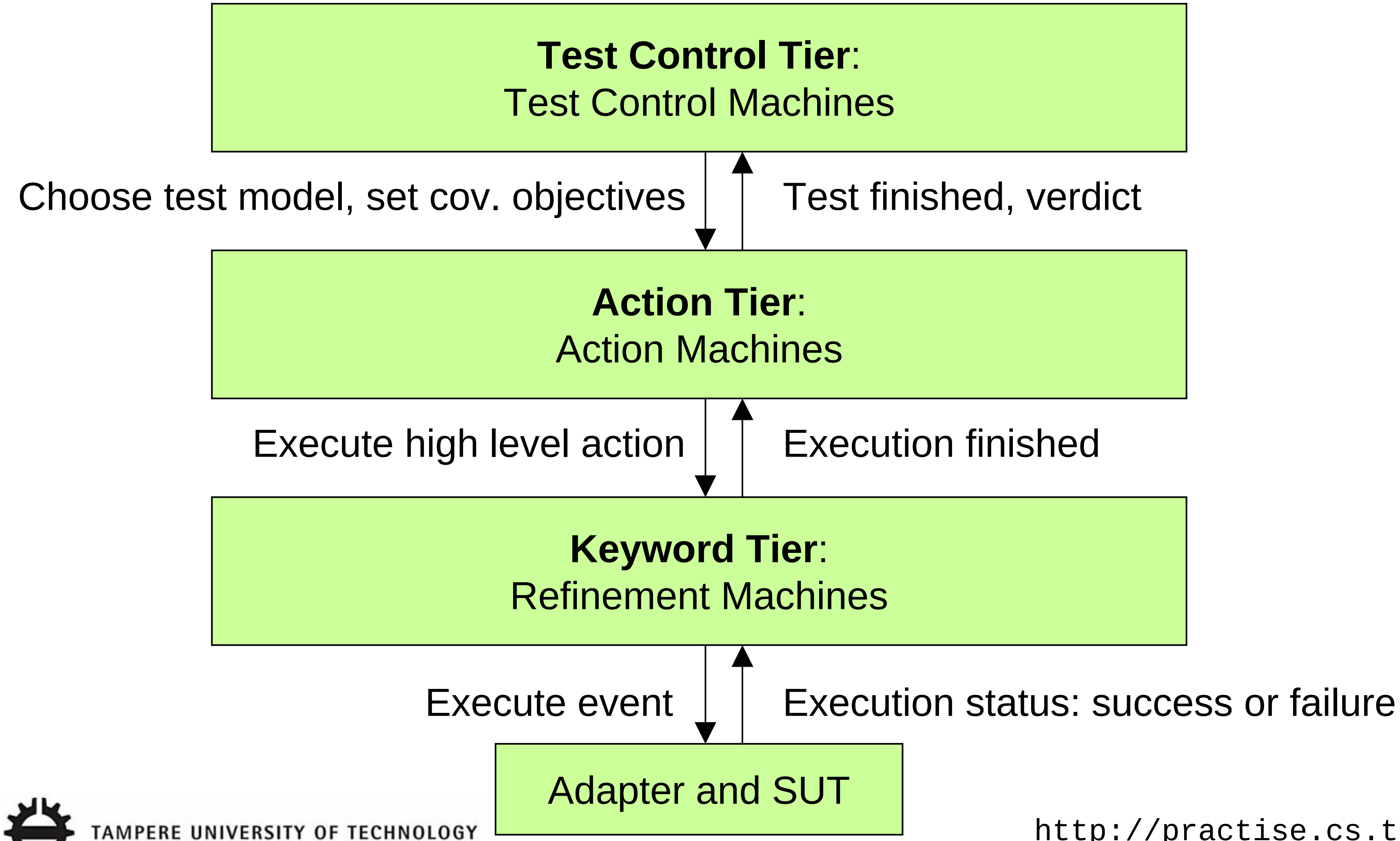
Series 60 Camera application



Example Refinement Model



TEMA 3-Tier Test Model Architecture



Test Control Tier:
Test Control Machines

Choose test model, set cov. objectives

Test finished, verdict

Action Tier:
Action Machines

Execute high level action

Execution finished

Keyword Tier:
Refinement Machines

Execute event

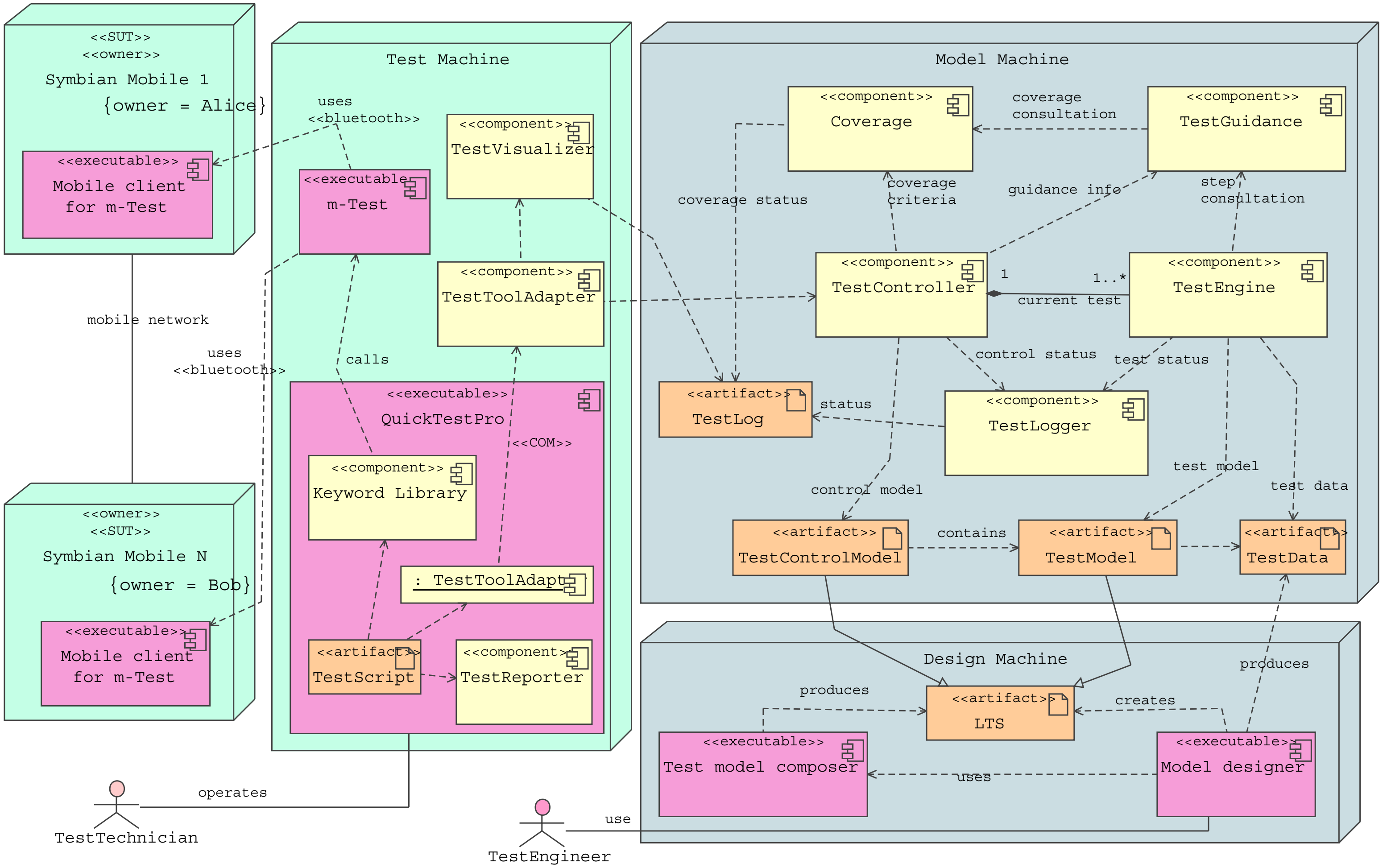
Execution status: success or failure

Adapter and SUT

TAMPERE UNIVERSITY OF TECHNOLOGY
Institute of Software Systems

`http://practise.cs.tut.fi`

TEMA Tool Software architecture



TAMPERE UNIVERSITY OF TECHNOLOGY
Institute of Software Systems

http://practise.cs.tut.fi

Conclusions

- Model-based testing is technically superior to conventional test automation
- The tools are not necessarily more expensive; you can start bug hunting with a piece of paper and a pen
    - For automatic generation of tests from models our first prototype needed also a few hundred lines of code
- Deployment of new and ground-breaking development technologies is always hard: management buy-in, training, how to reorganize work, how to get trusted commercial support etc.

Field testing S60 camera application