Model-Based Testing

Jan Tretmans
Embedded Systems Institute, Eindhoven, NL
and Radboud University, Nijmegen, NL
Doing Something with Models

- Modelling: making a model reveals errors
- Simulation: go step-by-step through the model
- Model checking: go through all states of the model
- Proving: prove theorems about the model
- Code generation: executable code from the model
- Testing: test an implementation for compliance
- Model mining: generate a model from observations
Code Generation from a Model

A model is more (less) than code generation:

- views
- abstraction
- testing of aspects
- verification and validation of aspects
Model-Based Testing

Tools
MBT: Some Tools - ioco

- AETG
- Agatha
- Agedis
- All4Tec MaTeLo
- Autolink
- Axini Test Manager
- Conformiq Qtronic
- Cooper
- G∀st
- Gotcha
- JTorX
- NModel
- ParTeG
- Phact/The Kit
- QuickCheck
- Reactis
- RT-Tester
- SaMsTaG
- SeppMed MBTsuite
- Smartesting CertifyIt!
- Spec Explorer
- Statemate
- STG
- TestGen (Stirling)
- TestGen (INT)
- TestComposer
- TGV
- TorX
- TorXakis
- T-Vec
- Uppaal Cover
- Uppaal-Tron
- Tveda
- . . . . .
MBT: Some Tools - commercial

- AETG
- Agatha
- Agedis
- All4Tec MaTeLo
- Autolink
- Axini Test Manager
- Conformiq Qtronic
- Cooper
- G∀st
- Gotcha
- JTorX
- NModel
- ParTeG
- Phact/The Kit
- QuickCheck
- Reactis
- RT-Tester
- SaMsTaG
- SeppMed MBTsuite
- Smartesting CertifyIt
- Spec Explorer
- Statemate
- STG
- TestGen (Stirling)
- TestGen (INT)
- TestComposer
- TGV
- TorX
- TorXakis
- T-Vec
- Uppaal-Cover
- Uppaal-Tron
- Tveda
MBT: Off-Line - On-Line

model-based test generation

system model

test execution

SUT

pass fail
MBT: Off-Line = Batch

- Test cases
- Model-based test generation
- System model
- Test cases
- Test execution
- SUT
- Pass fail
MBT: On-Line = On-the-Fly

- **Model-based test generation**
- **System model**
- **Test execution**
- **SUT**

**Pass**

**Fail**
MBT : On Line

- On Line / on-the-fly test generation and test execution
- Examples: TorX, JTorX, TorXakis, Uppaal-Tron, ....
- In particular for dealing with non-determinism
MBT: Tool Architecture - On Line

spec. explorer test gen. test runner adapter SUT

specification text states transitions abstract actions abstract actions concrete actions
Model-Based Testing: The Process

- **ideas**
- **requirements**
- **model**
- **test generator**
- **test executor**
- **test cases**
- **adapter**
- **SUT**
- **verdict**
- **test result**
- **analysis**

- **on-line**
- **off-line**
Model-Based Testing

Electronic Passport
Electronic Passport

New Passport

- Machine Readable Passport (MRP, E-passport)
- with chip (JavaCard), contact-less
- storage of picture, fingerprints, iris scan, .......
- access to this data protected by encryption and a new protocol
- just released in EU

Our job: testing of e-passports

- emphasis on access protocol
  == exchange of request-respons messages between passport and reader (terminal)
E-Passports: Basic Model

- **INIT**
- **BAC**: Basic Access Control
  - *initial protocol*
  - read photo
  - read personal data
- **EAC**: Extended Access Control
  - requires successful BAC
  - read photo
  - read personal data
  - read fingerprints
MBT for E-Passports: Model
Model-Based Testing of E-Passports

model-based test generation

system model

test execution

SUT

pass fail
MBT for E-Passports: Test Runs
MBT for E-Passports: Results

- Tested:
  - Basic Access Control (BAC)
  - Extended Access Control (EAC)
  - Active Authentication (AA)
  - Data Reading: requires data testing

- Tests up to about 1,000,000 test events
  - complemented with manual tests

- No error found yet ......
Model-Based Testing

A Wireless Sensor Network Node
Myrianed: Wireless Sensor Network

- RF TRANSMITTER (NORDIC SEMI)
- CPU (ATMEL XMEGA128)
- RF ANTEenna
- I/O INTERFACES
Wireless Sensor Networks

warehouses: 
sense &
control

tracing
products: 
active
labels

trains: 
seat
reservation

health care: on-body networks
Myrianed: a WSN with Gossiping

communication inspired by biology:

ant colonies
cell interaction

and also by girls:

MYRIANED “GOSSIP” protocol: gMAC
WSN: Typical Test

Put WSN nodes together:
- end-to-end testing

Test all WSN nodes together,
- i.e. test the Network
  - provide inputs to network
  - observe outputs from network
WSN: Model-Based Testing

Model-based testing of a single node:

- protocol conformance test of the gMAC protocol
- according to ISO 9646
- local test method
- time is important in gMAC:

real-time model-based testing
gMAC Behaviour

Node 1

Node 2

Node 3

Node 4

Node 5

Message: 2 bytes slot nr total 32 bytes
WSN Node in Protocol Layers

node 1

Upper Tester

application interface

radio interface

Lower Tester

gMAC Layer

node 2

Application Layer

gMAC Layer

Radio Layer

node 3

Application Layer

gMAC Layer

Radio Layer

Medium Layer
Local Test Method

Approach:
• only software, on host
• simulated, discrete time

```c
if(gMacFrame.currentSlotNumber >
gMacFrame.lastFrameSlot) {
    gMacFrame.currentSlotNumber = 0;
    gMacFrame.syncState = gMacNextFrame.syncState;
    if (semaphore.appBusy == 1) {
        mcTimerSet(SLOT_TIME);
        mcPanic(MC_FAULT_APPLICATION);
        return;
    }
}
```
Test Interfaces / PCOs

implemented: software function calls in adapter

application interface:
3 function calls

modelling: actions for function call and function return

implemented: software emulation of send and receive

radio interface:
send and receive of messages via air interface

implemented: emulation of oscillator
ticks with external software trigger

timer interface:
clock ticks

modelling: clocks ticks as input actions

implemented: software function calls in adapter

GMAC layer: Software

modelling: send and receive actions in model
Test Architecture

GMAC layer:
- Software
- Adapter in Software

Model-Based Test Tools:
- Uppaal-Tron
- JTorX
- TorXakis

Application interface
Clock
Socket interface
Radio interface

Model-Based Testing

- **model-based test generation**
- **system model**
- **test execution**
- **SUT**
- **pass fail**
WSN: Model-Based Testing

MBT tool:
- Uppaal-Tron
- TorXakis
- JTorX

WSN software on PC

(vague) descriptions
guru ad-hoc model learning

pass fail
test runs
C test adapter
A First Model for MBT of a WSN Node

First Model:
- abstract, underspecified, loose model: all behaviour is allowed
- all output allowed, few inputs specified (but there must be some inputs for synchronization)
- refinement in subsequent steps
WSN: Test-Based Modeling

Make a model from observations made during testing

Uppaal-Tron
TorXakis
JTorX

C

test
adapter

pass fail

WSN software on PC

Test runs
MBT and TBM

model

improve model

refine model

no

MBT

conforming

no

yes

satisfied

no

model world

physical world

SUT

improve SUT

more tests

no
WSN: A Timed Automaton Model
Discussion
Perspective
MBT Lessons

• Model construction
  – leads to detection of design and specification errors
  – how to get a valid model?

• Adapter/test environment
  – specific for each system
  – sometimes laborious, but not specific for MBT

• Longer and more flexible tests
  – full automation: test generation + execution + analysis
  – easy to repeat for regression testing, modifications, maintainability, .....
MBT: In the V-Model

- Requirements
- Specification
- System design
- Module design
- Coding
- Model-based testing
- Unit testing
- Module testing
- Integration testing
- System testing
- Acceptance testing
MBT Who’s Done It

- Microsoft
  - EU, US anti-trust case: make Windows more open for competitors
  - producing documentation for Windows protocols
  - check documentation w.r.t. real product by model-based testing

- 75 protocols, 50 fte, 10,000 reqs, average 1000 LoM/model
- MBT tool: Spec Explorer
- Productivity gain: 42% and 34% w.r.t. traditional testing
Philips Health Care  (Niels Methorst, Bits & Chips 2011)

- Tool: Axini test Manager
- Reduction of effort: 80%
- Non-determinism: YES
- On-line testing: YES
- Different issues detected with MBT and manual testing → MBT and manual testing complementary
- MBT = fun + maintenance + defect investigation + learnability
- Future: data, performance, evolvability
MBT Who’s Done It

MBT User Conference (Berlin, Oct. 2011)

- 120 attendants, 75% industry, 45 companies, 21 countries
- telecom is furthest (mostly Conformiq)
- automotive is following, then embedded
- not so much administrative (Smartesting)
- many companies are experimenting, pilot projects
- not off-the-shelf yet
- reported gain (time, cost, coverage, ...): 10 - 40 %
MBT Who’s Done It

MBT User Conference (Berlin, Oct. 2011)

• connection to other development tools is important issue

• flexibility, maintainability is plus

• often combined with scrum, agile

• not all testing with MBT

• non-determinism: discussion

• on-line testing: discussion
Model based formal testing can improve the testing process:

- model is precise and unambiguous basis for testing
  - design errors found during validation of model
- longer, cheaper, more flexible, and provably correct tests
  - easier test maintenance and regression testing
- automatic test generation and execution
  - full automation: test generation + execution + analysis
- extra effort of modelling compensated by better tests
  - is modelling really an extra effort?
- (commercial) model-based testing tools become available
JTorX:

https://fmt.ewi.utwente.nl/redmine/projects/jtorx

→ files → 1.9.0 → your favorite version

example ‘Cat, Cabbage, ....” :  riverCrossing04.tgz