Test-Driven Modelling of Embedded Systems

Allan Munck: allmun@dtu.dk or amunck@gnresound.com.

EMSIG2015
GN ReSound – Smart Hearing Systems

Control and Relief apps also for Android
Case study

- Analysis of impact of new architecture.
- More than 30 highly interacting use cases.
- Protocol constraints.
- Timing requirements.
- Power constraints/requirements.

Diagram:
- Main System
  - bus
    - Memory System
    - Radio System
Methodology

**TD-MBSE**

1. Obtain and specify behavioural or architectural requirement
2. Refine requirement
3. Model behavioural scenario or architectural test case
4. Run new test case
5. Interactive simulation
6. Create or modify system behaviour / system architecture

**TD-DSE**

1. TD-MBSE → Base Design
2. Base Design → Alternative Designs
3. Alternative Designs → Obtaining Estimates
4. Obtaining Estimates → Formal Verification
5. Formal Verification → Property Estimation
6. Property Estimation → Property Simulation
7. Property Simulation → Functionality Impact
8. Functionality Impact → Performance Impact

**Figure 1:** Test-Driven Model-Based Systems Engineering

**Figure 2:** Test-Driven Design Space Exploration
Practical modelling and verification

• Collecting system knowledge in SysML.

• TD-MBSE with UPPAL (figure 3).

• Formal model checking queries, e.g.:
  – A[] not deadlock
  – Streaming.Receive --> Streaming.Done
  – etc.

  – Base design: “Efficient” solution.

  – E[<=100;2] (max:Streaming.delay)
  – simulate 1[<=200]{Streaming.curr}
  – Using UPPAAL plot composer + Matlab

Figure 3: Test case scenario modelling
Related work

• Test-Driven Development (TDD) [e.g. Cordemans et al.]
  – Used heavily in the software industry to increase code quality.
  – Not directly applicable to embedded systems that include hardware.

• Model-based test-driven development [Mou et al.]
  – Method proposed but not tested or verified (no experimental data reported).
  – Has limited scope and is probably not feasible for embedded systems.

• Test-driven software modelling [Zhang]
  – Method based on simulations of message sequence charts.
  – Experimental data demonstrates increased productivity and quality on large projects.

• Test-driven UML modelling of software systems [Hayashi et al.]
  – Test-first methodology include both unit and scenario testing.
  – A tool to facilitate the method is provided – with somewhat poor usability.
Problem and solutions

• Previous work
  – Special tools required.
  – Too cumbersome to use.
  – Only unit or scenario testing.
  – Cannot find unintended emergent errors.
  – Cannot guarantee design.
  – Limited scope.
  – No widespread acceptance.

• Our solution
  – Utilizing existing tools as is.
  – Easy to use – modelling becomes a simple mechanical process.
  – Formal and statistical verification.
  – Captures unintended behaviour.
  – This work is limited to modelling of architecture and behaviour of embedded systems.
  – Method easy to adopt.
Discussion

• Traces generated by UPPAAL made modelling easy.

• Problem with memory exhaustion.
  – Very long traces for low level modelling.
  – State explosions for large models.
  – No simple long term solution.
  – Formal model checking may have to be abandoned.

• Only behaviour and architectural parameters considered in this work.
  – Architecture: Number of interfaces, coupling, etc. may also benefit from TDA.
  – User interfaces and even stakeholders, use cases, requirements, etc. may also benefit.
Industrial future

• Smart Products:
  – CPS : Cyber Physical (sub) Systems.
  – BDS : Big-Data Systems.
  – Millions/Billions of users.
  – Connected via (multiple) clouds.

• No tools available to analyse complete system.

• Proposal:
  – Model and analyse single-user scenarios with UPPAAL.
  – Model and simulate system with thousands of users using SimJava, SystemC, VDM or equivalent language.
  – Vary parameters to obtain simulation results for subsequent estimation/extrapolation of realistic usage.
Summary

• TD-MBSE and TD-DSE proposed to handle increasing complexity.
• Includes test-driven modelling of behaviour and architecture.
• Methodology tested on an actual industrial case with success.

• Future research to overcome memory exhaustion problems.
• Future research to expand the methodology to include other aspects of modelling.
• Methodology for future industrial cases proposed.

Thank you for listening!
Acknowledgements

• The authors would like to thank all involved stakeholders at GN Resound and at Technical University of Denmark for their interest and involvement in this project.

• We will also like to thank the UPPAAL team for excellent support.

• Finally we wish to thank the Confederation of Danish Industry for sponsoring this project.