Industry-Driven Testing: Past, Present, and Future Activities at Simula

Shaukat Ali
Research Scientist
Certus Software V & V Center
Simula Research Lab
Myself

- Affiliated with Simula since 2007
  - Have been working in collaboration with Cisco for several years
  - Exploring new testing topics in FMC technologies

- Software Quality Engineering Laboratory (SQUALL), Carleton University, Ottawa, Canada
  - Siemens Corporate Research (SCR), Princeton, New Jersey, USA.

- Verification and Testing Lab (VT), The University of Sheffield, UK

- Center for Software Dependability (CSD), Islamabad, Pakistan
Embedded Systems Shape Our World

- **Home Appliances**
- **Medical Equipment**
- **Consumer Electronics**
- **Transportation**
- **Industrial Robots**
- **Oil Industry**
Interesting Facts About Embedded Systems

- About 98% of computing devices are embedded\(^1\)
- By 2020: **40 billion** embedded devices\(^1\)
- Annual budget for embedded systems: **160 billion euros** with annual growth of **9%**\(^2\)
- In 2008: **30** embedded microprocessors per person with at least **5 million** function points of embedded software\(^2,3\)
- The growth rate of embedded software is increasing\(^2,3\)
  - On average an embedded system has **1 million** Lines Of Code (LOC)
  - A premium-class automobile has around **100 million** LOC
  - Boeing 787: **6.5 million** LOC

---

Embedded Systems Can be **Critical**

Non-critical

- Mars Climate Orbiter
- Mariner I
- Ariane 5

Mission-critical

Safety-critical

- Therac-25
- Oil rig mishap
Large-Scale Embedded Systems

- Embedded Software in A380
- Communication Intensive Video Conferencing Systems
- Cruise Ships
- Data Acquisition System in Oil Refineries
Certus Software V&V Center at Simula

[ simula . research laboratory ]
- by thinking constantly about it

Industrial Partners

[ simula . research laboratory ]
- by thinking constantly about it
Methodology for Industry-Driven Research at Certus

Automated Model-based Testing of State-driven Systems (AMOS)
D-Mint Partners
Model-based Test Case Generation: A Simplistic View

UML Class diagram

UML State machines

TRUST

Test Case Generation Tool

Test Scripts

//statements to verify state invariant for Idle state
testtarget.dial(5345,SIP,2K)

//statements to verify state invariant for Connected_1 state
Why Model-based Testing

- Facilitates **automation**
- Helps defining specialized test strategies targeting specific **test objectives**
- **Systematic** and **rigorous**
- Automatically generate many **non-repetitive** and **useful** tests
- Easier test case **maintenance**
TRUST supports configurable and extensible features

- Input models: State machine, Sequence diagram, ..
- Test models: Test tree, ..
- Coverage criteria: All transition coverage, All message coverage, ..
- Test data generation strategies: Random, Search-based, ..
- Test script languages: Python, C++
Model-based Robustness Testing - Cisco

Robustness is the degree to which a software component functions correctly in the presence of exceptional inputs or stressful environmental conditions (IEEE Std 610.12-1990)
Why is robustness important?

- Robustness is considered very critical for embedded systems, for example communication and control systems.
- Required by many quality standards:
  - IEEE Standard Dictionary of Measures of the Software Aspects of Dependability
  - ISO’s Software Quality Characteristics standard
  - Software Assurance Standard by NASA
MBRT is challenging due to the following reasons:

- Modeling robustness behavior makes modeling highly complex and redundant
- Automated generation of executable test cases from robustness models
  - Targeted to reveal robustness faults
  - Defining appropriate test strategies for robustness testing
  - Generating test data
Case Study: Video Conferencing System
How does a Video Conferencing System work in practice?
Core Functionality

Saturn

EP1

Video Channel

Incoming channel

EP3

Presentation Channel

Call

Audio Channel

EP2

Outgoing channel
Robustness Testing for Video Conferencing Systems

System Under Test

Video
Audio
Presentation

Network

Endpoint 1

Endpoint 2

Endpoint 3
Key results: 95% modeling effort reduced, improved readability, improved modeling quality, and reduced modeling errors, test data generation significantly better than existing tools
Test Case Selection

- System testing on real platform and network is expensive and time consuming.
- MBT techniques will generate large sets of test cases when applied to Industrial systems especially in case of non-functional testing.
- Selecting a small enough subset of these test cases that is realistic to be executed and analyzed during the time and resource constraints, while preserving maximum possible fault revealing power.
Similarity-based Test Case Selection

Goal
- Finding a subset of the original test suite, with a given affordable size (based on testing budget), that detects the most faults.

Procedure
- Defining a measure for similarity between test cases
- Encoding of test cases using right level of abstraction that contains only relevant information
- Defining a similarity function that assigns a similarity value per each pair of encoded test cases

Minimizing similarity among selected test cases

Key Result: by selecting 15% of the generated test suite we achieve almost 100% fault detection rate → No need for running 85% of the test cases.
VERDE: Verification-Oriented & Component-based Model Driven Engineering For Real-Time Embedded Systems
Overall Testing Framework

Key Results: Comprehensive environment modeling approach for testing, Automated environment simulator generation, and Automated test case generation for black-box system testing.
Management of product variability in test configuration generation and execution
Management of product variability in test configuration generation and execution
Model-Based functional and non-functional testing of Product Lines
Model-Based functional and non-functional testing of Product Lines

- Product Line Engineering (PLE) is expected:
  - To enhance quality and productivity
  - Speed up time-to-market
  - Decrease development effort, through reuse

- To use PLE for systematic testing and more specifically model-based testing (MBT)

- The aim is to reduce the effort required for modeling and configuring products of the product line family for MBT.
Testing Data-Intensive Systems

Arnaud Gotlieb
Head of Department
Mobile: +47 406 26 077
E-mail: arnaud@simula.no

Erik Rogstad
PhD Student
Mobile: +47 45 60 06 63
E-mail: erikrog@simula.no

[ simula . research laboratory ]
- by thinking constantly about it
Testing Data-Intensive Systems

- System and acceptance testing using large data samples
- Running of the whole system (as a black box) against test data or a complete simulation of the actual running system for purposes of testing out the adequacy of the system.
- Simulation, duplication or replication of parts of a system (whether being a sub-system, super-system or environment) is difficult
- Sample selection is of outmost importance
The Future
Projects

- Testing in real-world configurations and scenarios
  - Testing by emulating real world configurations such as different Video Conferencing Systems geographically distributed with diverse configurations and with different network configurations as deployed at customers’ sites.

- Smart Software Development Solution
  - Integrated model-based development, code generation, and testing solutions
Questions