Extensive Test of Heavy-Machinery ECU on a NI VeriStand HiL using TestWeaver

Dr. Thomas Neubert, Hydrive Engineering GmbH
Dr. Mugur Tatar, QTronic GmbH
1 Introduction

2 Software Test Strategies

3 New Test Generation Strategy

4 Test Application

5 Summary
Software Complexity in Heavy Machinery

- single solutions - yesterday - integration of single systems - today - networking of data and functions to assistance systems - tomorrow -

- engine, transmission, hydraulic -
- damping systems -
- tracking systems -
- error detection, diagnosis -
- ground surface system -
- telemetric -
- virtual tow bar -
- teach-in-systems -
- workspace monitoring -
- anti-roll-over-system -
- fleet management -

Time

Complexity of software
14.09.1993: Accident of flight LH 2904, Airbus A320-211 in Warsaw after landing

Runway distance too short for overshooting or braking

**Consequence:** 2 dead persons, 54 injured persons

Source: http://de.wikipedia.org/wiki/Lufthansa-Flug_2904
Causes and Consequences of a Software Failures

- **Environmental Conditions**
  - Ground type (street, sand)
  - Street (dry, wet, ice)
  - Sand (soft, hard)
  - Summer / winter conditions

- **Driver - Use Cases**
  - Non-predictable use cases and applications
  - Non-logical operating sequences
  - Interaction with other systems

- **Systematic Failures**
  - Concept faults
  - Rounding faults, division by zero, data type conversions
  - Hardware faults
  - Tolerances
  - Aging

Faults happen...

Important: Find all faults in due time...

Test and validation must be done in a large space of situations
V-Model for Software Development
Manually written test scripts

Test #31
start_car();
shift_lever = D;
while (gear<2)
  accelPedal = 20;
if (time>2)
  error("no shift")
...

Problems
- High costs for development and maintenance of scripts
- Low coverage of tests
Idea
- Intelligent generation of ... 1000s of differing test scenarios
- Active attempt to:
  - maximize state coverage
  - drive the system in “difficult” situations

Benefit
- High coverage
- Low efforts for test specification

Testing = playing against (simulated) system
1 Introduction

2 Software Test Strategies

3 New Test Generation Strategy

4 Test Application

5 Summary
TestWeaver – Test Generation Strategy

1. Change sub-optimal scenarios to generate worst-cases
2. Drive the system in states that were not covered before

Reactive scenario generation: each scenario depends on history of generated scenarios. All cases can be reproduced.
TestWeaver – Test Generation Strategy

Th. Neubert / M. Tatar
1 Introduction
2 Software Test Strategies
3 New Test Generation Strategy
4 Test Application
5 Summary
Test Application - Excavator

- **Excavator Control Software**
  - developed and approved 5 years ago
  - undesirable error messages in field
  - not reproducible
  - high warranty costs
  - customers dissatisfied

Software should be driven into situations where undesirable errors will be detected.
HiL Test Bench

- Test bench frame
- ECU
- Signal conditioning
- Host PC
- Real-time PC
- RT Model control
Test Bench Architecture – Hand Coded Tests

Test Cases -> Test Automation
- manually created test cases -> Test Reports

Driver signals

.net API

Realtime Simulation
NI VeriStand

Realtime Machine Model
SIMULATION X

Signal Conditioning

Failure Insertion Unit

Sensor Signals

Valve Currents

ECU Control Software

Error codes reading and clearing
Results of Hand Coded Tests

- 1st step:
  - 50 manually created test cases
  - different load types (fixed, shuttling)
  - different load and load positions

- 2nd step:
  - more than 200 further test cases manually created
  - sensor tolerances

- Test effort: 3 month

No error reproduced.
Test Bench Architecture – Generated Tests

Test Space
- Coverage Goals
- Test Weave
- Automatic Generated Test Scenarios
- Test Reports

Test Reports
- Error codes reading and clearing
- Test Bench Architecture – Generated Tests
- Realtime Simulation
- ECU Control Software
- .net API

Realtime Simulation
- NI VeriStand

Realtime Machine Model
- Simulation X

Signal Conditioning
- Sensor Signals
- Valve Currents

Failure Insertion Unit
- Automatic Generated Test Scenarios
- .net connector API

NI VeriStand .net connector API

Pressure Relief Valve A
- Check Valve A
- Check Valve B
- Supply
- Tank 1
- Tank 2
- Engine

Th. Neubert / M. Tatar
14th ITI Symposium Dresden 30.11./1.12.2011
Results of Generated Tests

- Implementing of TestWeaver Set-up: 3 days
- test effort: 7 days
  - more than 2700 test scenarios automatically created
  - slope
  - load at bucket
  - attachment position
  - speed

After 2700 test scenarios error has been reproduced!
Summary

1. Introduction
2. Software Test Strategies
3. New Test Generation Strategy
4. Test Application
5. Summary
## Summary

### Hand Coded Software Tests
- Definition of test cases and quality criterias
- Verification of software functionality (functions, electrical faults, safety)
- Limited test coverage (100…1.000 test cases)
- High test effort

### Generated Software Tests
- Definition of input space and not allowed system states
- Automatic generation of test scenarios
- Verification of software robustness
- High test coverage (10.000…100.000)
- Low test effort
THANK YOU FOR ATTENTION

www.qtronic.de  www.hydrive-engineering.de