Model-Based Testing of the NoTA Interconnect

Antti Huima
CTO
Conformiq Inc.
Model-Based Testing and Automated Test Design™
Company

• Founded in 1998
• Three locations
  – Saratoga, CA, USA (HQ)
  – Helsinki, Finland (R&D)
  – Stockholm, Sweden
• Privately held
• 20–30 employees
Setting

- Companies that develop software need to test it also
- The **testing process** consumes time and resources
- It “produces” quality in terms of reducing the number of “customer found defects” (CFDs), i.e. bugs that users find
Testing Processes

- Read documentation
- Design tests
- Produce a model
- Experiment and test manually
- Encode test cases as test scripts
- Derive test cases and test scripts automatically
- Collect test cases
- Encode test cases as test scripts
- Run scripts for initial testing
- Report
- Run scripts later for regression
Test Automation Overview

- **Device or System Under Test** (stuff you get from developers)
- **Test Harness** (stuff you use to talk with the system you test, e.g. TCL libraries)
- **Test Execution** Subsystem (stuff that runs your scripts, e.g. TCL interpreter)
- **Test Scripts** (your testing commands e.g. in TCL)

The Beef

Test Automation

System Under Test (SUT)
Risks and Costs of Test Case Design

- **Missed tests**
  - Can result in defects on field
  - Missed tests do not advertise themselves

- **Incorrect tests**
  - Can result in defects on field
  - If detected, trigger rework

- **Redundant tests**
  - Cause unnecessary implementation and maintenance costs

- **Maintenance**
Conformiq Qtronic—ATD Tool

• Reads in a system model expressed in Java syntax and optionally state charts

• Automatically designs test inputs and expected results, with data and timings

• Produces test cases and test documentation in user-defined formats, e.g. web pages, TCL and XML

• Runs in Eclipse
System Model...

- Is computer-readable
- Describes the correct operation of the system
- Is small and high-level
- Is in practice an object-oriented computer program
- Can be described also using diagrams (e.g. state machine diagrams)
Conformiq Qtronic™ Flow

1. Analyze Requirements
2. Update Model
3. Execute Tests
4. Generate Tests
5. Troubleshoot and Report
Productivity Improvement

1X
Manual test design

5X
Automated Test Design™ in initial deployment

10–20X
Automated Test Design™ in subsequent tested product iterations

Source: average results from customer benchmarks
The Core Productivity Improvement

• The average core productivity improvement in test case design from customer benchmarks: above $5X = 80\%$ cost reduction when
  – benchmarked against already automated testing process (e.g. TTCN-3 based)
  – test coverage remains constant
  – model is created from scratch

• The results are higher for subsequent product iterations because maintaining the model is more cost-efficient than maintaining hand-written test cases (typically >10X)
Conformiq Qtronic Application Areas

- Functional (black box) testing
- Conformance testing
- Component testing
- System-level testing
- Integration and acceptance testing
- Error testing
- Initial and regression testing
## Product Features

<table>
<thead>
<tr>
<th>Mathematically Generates</th>
<th>Other Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Test inputs</td>
<td>• Modeling in UML and Java-compatible notation</td>
</tr>
<tr>
<td>• Expected test outputs</td>
<td>• Fully customizable output formats</td>
</tr>
<tr>
<td>• Test timings</td>
<td>• Imports diagrams from 3&lt;sup&gt;rd&lt;/sup&gt; party tools</td>
</tr>
<tr>
<td>• Test plans</td>
<td>• Interactive workbench</td>
</tr>
<tr>
<td>• Executable test cases</td>
<td>• Based on Eclipse&lt;sup&gt;®&lt;/sup&gt;</td>
</tr>
<tr>
<td>• Traceability matrices</td>
<td></td>
</tr>
<tr>
<td>• Test dependency matrices</td>
<td></td>
</tr>
</tbody>
</table>
Based on the publication “Testing NoTA Interconnect R3 via H_IN interface” available at http://www.notaworld.org/files/NoTA%20Interconnect%20R3%20Testing.pdf

Model-Based Testing of a NoTA Interconnect Implementation
Project Overview

- Carried out in Q2 2008
- Target system NoTA interconnect release R3 alpha and beta versions on Linux
- Tested via the BSD socket type H_IN interface only
- All tests computer-generated using automated test design
- Test execution via C-based concurrent, multi-threaded test harness
Workflow

Test cases as MSCs

Automatic test generation

NoTA system model

Executable test cases in C

TEST_CASE(qtronic, 7)
{
    TEST_CASE_PREAMBLE;
    TestInput (ss_master_in, Hsocket(0));
    TestInput (ss_master_out, HsocketRetvalue(0, 0, "OK"));
    TestInput (ss_master_in, Hbind(0, 0, 1001, 0));
    TestInput (ss_master_out, HbindRetvalue(0, "OK"));
    TestInput (ss_master_in, Hlisten(0, 0));
    TestInput (ss_master_out, HlistenRetvalue(0, "OK"));
    TestInput (ss1_in, Hsocket(0));
    TestInput (ss1_in, Hconnect(0, 0, 1001, 0));
    TestInput (ss_master_in, Hlisten(0, 0));
    TestInput (ss_master_out, HlistenRetvalue(0, "OK"));
    TestInput (ss_master_in, Hsocket(0));
    TestInput (ss_master_out, HsocketRetvalue(0, 0, "OK"));
    TestInput (ss_master_out, HsocketRetvalue(0, 1, "OK"));
    TestInput (ss_master_out, HsocketRetvalue(0, 2, "OK"));
    TEST_CASE_POSTAMBLE;
}
TEST_CASE(qtronic, 7) {
    TEST_CASE_PREAMBLE;
    TestInput(ss_master_in, Hsocket(0));
    TestOutput(ss_master_out, HsocketRetvalue(0, 0, "OK"));
    TestInput(ss_master_in, Hbind(0, 0, 1001, 0));
    TestOutput(ss_master_out, HbindRetvalue(0, "OK"));
    TestInput(ss_master_in, Hlisten(0, 0));
    TestOutput(ss_master_out, HlistenRetvalue(0, "OK"));
    TestInput(ss1_in, Hsocket(0));
    TestOutput(ss1_out, HsocketRetvalue(0, 0, "OK"));
    TestInput(ss1_in, Hconnect(0, 0, 1001, 0));
    TestInput(ss_master_in, Hlisten(0, 0));
    TestOutput(ss_master_out, HlistenRetvalue(0, "OK"));
    TestInput(ss_master_in, Hsocket(0));
    TestOutput(ss_master_out, HsocketRetvalue(0, 1, "OK"));
    TestInput(ss_master_in, Haccept(0, 0));
    TestOutput(ss_master_out, HacceptRetvalue(0, 2, "OK"));
    TestOutput(ss1_out, HconnectRetvalue(0, "OK"));

    TEST_CASE_POSTAMBLE;
}
notatester Architecture

- Test cases
- Test master
- Shared memory area
- Control
- Log thread
- Tester
- Peer protocol (not directly under test)
- Stdout
- Stderr
- RM nota-ind
- Nota-ind
- Nota-ind
## Some Incident Highlights

<table>
<thead>
<tr>
<th>Incident</th>
<th>Symptom</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory corruption in data transfer routine in socket.c</td>
<td>Data transfer tests crash NoTA, or data transfer integrity check fails (received wrong data)</td>
</tr>
<tr>
<td>bidirectional data transfer deadlocks NoTA</td>
<td>Attempt to do reading and writing via same socket in two different threads causes deadlock</td>
</tr>
<tr>
<td>ind daemon crashes during start-up</td>
<td>ind daemon segfault or assertion failure</td>
</tr>
<tr>
<td>ind daemon does not find IA when connecting via non-RM</td>
<td>Hconnect returns unexpectedly with error</td>
</tr>
<tr>
<td>connection attempt hangs and causes tester timeout</td>
<td>either Haccept or Hconnect does not return within tester timeout (observed in multiple test cases)</td>
</tr>
<tr>
<td>Hlisten call deadlocks when called for connecting socket</td>
<td>Hlisten does not return with error code if called for a socket for which another thread is calling Hconnect</td>
</tr>
<tr>
<td>multiple connections to same service not accepted</td>
<td>Hconnect returns error code when connecting to a service that has already accepted one client</td>
</tr>
<tr>
<td>Hrecv returns corrupted return value</td>
<td>Hrecv return value -107 observed</td>
</tr>
</tbody>
</table>
Early Test Execution Report

- Release: 20080527A (0527 with socket.c bug fix)

- Tests: 92 tests generated with Conformiq Qtronic
  - 25 pass
  - 64 fail
    - 4 high-priority problems
    - 9 bad error code situations
    - 4 other obvious anomalies
  - 3 erroneous, fail after fixing by hand
Final Test Execution Report

- Release: 20080605-284
- Tests: 191 tests
  - 158 pass
  - 17 fail
    - 3 different Hselect related
    - 4 distinct error code scenarios
    - Infrequent L_IN failures
  - 16 erroneous tests; all but one related to Hselect
Model Architecture

- One class for a NoTA subsystem
- One class for socket
- 3 instances of NoTA subsystem (corresponds to test setup)
- variable number of socket instances (depending on number of Hsocket() and Haccept() calls generated)
- Peer protocol modeled, shows up in MSCs
  - Not necessarily according to the NoTA standard (documentation lacking)
  - Does not show up in or affect the testing interface
Beyond
Extending the Approach

<table>
<thead>
<tr>
<th>Current</th>
<th>Next: Stage 1</th>
<th>Next: Stage 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="current_diagram.png" alt="Diagram" /></td>
<td><img src="next_stage1_diagram.png" alt="Diagram" /></td>
<td><img src="next_stage2_diagram.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
| - Test transport (not application) level functionality via $\text{H}\_\text{IN}$  
- Test system has three components to ensure $\text{L}\_\text{IN}$ is being exercised | - Test infrastructure via both $\text{L}\_\text{IN}$ and $\text{H}\_\text{IN}$  
- Because subsystem is fully enclosed in the harness, one subsystem is enough for testing | - Test applications via $\text{L}\_\text{IN}$  
- “WSDL-level” codec used in test harness for app. testing  
- NoTA protocol may have been implemented by 3rd party, and its conformance is assessed automatically |
| - Model describes the behavior at $\text{H}\_\text{IN}$ normatively, i.e. used to assess test results  
- $\text{L}\_\text{IN}$ modeled in non-normatively | - Model describes the behavior of the subsystem between the two interfaces normatively | - Model describes both the $\text{L}\_\text{IN}$ level subsystem (common off-the-shelf component) features as well as the application specific features |
Conclusions
Summary

- Automated test design based on system models can be used to improve test design capability significantly
- In the NoTA context the approach has been successfully used to verify one interconnect implementation
- The approach can be extended towards application-level interface testing
Contact Information

Antti Huima
CTO
Conformiq Inc.

antti.huima@conformiq.com

+1 408 759 1108