Automated Test Design in MSS Protocol Testing

June 10th 2010

Ohjelmistotestauksen teemapäivä, TUT

Szilard Szell – Nokia Siemens Networks
Jani Koivulainen – Conformiq
Nokia Siemens Networks: strong tradition in innovation

Siemens since 1847

- The first telephone exchange using the digital electronic switching system EWSD goes into operation (1980)
- Built first UMTS network in Europe (2001)
- Completion of the Rhineland telephone cable (1921)
- Construction of Telefunken television set (1935)
- First automatic Telephone exchange (1905)
- First telephone exchange (1855)

Founded 1847

Nokia since 1865

- World's first Multimedia Messaging Service Center (2001)
- World's first triple-mode (GSM, EDGE, WCDMA) base station (1999)
- World's first GSM call made in Radiolinja's network, supplied by Nokia (1991)
- Europe's first digital exchange, the DX 200 (1982)
- Nokia's first generation of manual radio telephone systems (1967)
- Roots in paper, rubber, and cables, in just over 100 years Nokia becomes a powerful industrial conglomerate... (2001)

Founded 1865

Industry first live demonstration of Long Term Evolution (2006)
The R&D center production chain in Budapest

1,000,000,000 subscriber connected world wide

From requirements... to software deliveries!
Challanges in Testing

• Huge amount of existing functionality
  – More regression testing need
• Shorter release cycle
  – Less time for testing
• More customization
  – More specific functionality for smaller targets
• Economy
  – Pressure on cost effectiveness
• Quality goals
  – No quality sacrifice
So make it faster, let’s automate

1. Automate Test Execution
   - Make executable test case scripts
   - Schedule test run
   - Collect result

2. Automate Test result analysis
   - Compare test outcome with expected result
   - Report test result
   - Store/Archive result

3. Automate Test Design
   - Test design specification
   - Selecting best test techniques
   - Find optimal coverage
   - Document test cases
CONFORMIQ AUTOMATED TEST DESIGN
Challenges of Manual Test Design

- Missed tests
  - Can result in product defects
- Incorrect tests
  - Cause additional test development work
- Redundant tests
  - Cause extra development and maintenance costs
- Unknown requirements coverage
  - Can result in untested features
- Frequent changes to specification
  - Cause high cost for test suite maintenance
Automated Test Design

- **Model Based Testing (MBT)**
  - An "umbrella" of approaches that can be used to generate tests from models

- **Automated Test Design (ATD)**
  - An approach that uses system model driven MBT to design, document, and implement tests

- **Enables**
  - Faster test development
  - Improved test quality
  - Wider test coverage & guaranteed requirement coverage
  - Cost-effective test maintenance
  - Earlier test validation & detection of specification defects
  - Independence from test execution environment
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
SINAP CHARGING PROJECT
Project Scope and Goals

• **Scope of the project**
  − SUT: MSC Server (MSS)
  − Testing objective: SINAP Charging

• **Goals of the project**
  − Introduce the concept of Automated Test Design to NSN IN testing team.
  − Demonstrate that automatically designed test suite can be integrated with existing test execution environment.
  − Create reusable assets (models and model components for use in deployment) of Automated Test Design.
Test Architecture

Model -> Script templates

Test Harness (HIT) -> TEL Scripts (.lst, .tel)

Traffic Generator (TEL, MML) -> SCF simulator (IPSL)

BICC/SIP/ISUP MAP

CCF SSF

MSS (SUT) SRF

VLR

Expected CDRs

Charging and basic call analyzer (Perl)

IPSL Scripts (.ctl .scr)

Perl scripts (.pm .pl)

Execution logs

CDRs

SINAP analyzer

Test Architecture
Results

• Goals reached
  1. Introduce Qtronic
  2. Integrate Qtronic with HIT
  3. Create reusable assets/model architecture

• Test suite
  – Basic call + basic call party handling + User Interactons at different call phases
  – 15 test cases/~120 test steps (~20 functional requirements)

• 4 issues found

• Resources spent
  – Obtain domain knowledge 128 h
  – Work with model 45 h
  – Analyze system + logs 32h
  – Backend work 41 h
Conclusions

• ATD has a great potential in providing a "glue" i.e. single point of test case generation.
• ATD enforces understanding and documenting correct system behavior in the form of models.
• ATD with proper model architecture likely to increase test design efficiency.
• In ATD test plans and scripts generated with traceability mapping in one go.
Contact Information

Szilard Szell
szilard.szell@nsn.com

Jani Koivulainen
jani.koivulainen@conformiq.com