Test the test

A review and outlook of quality assurance techniques for software-intense communication-based systems

Ina Schieferdecker
TU Berlin/Fraunhofer FOKUS, Berlin, Germany
The percentage of successful IT projects is again dropping:

Severity of errors depends on the phase where they’re made

Defect correction effort
Although programmers, testers and program managers know that code must be designed and tested, many appear to be unaware that tests themselves must be designed and tested – by a process no less rigorous and no less controlled than that used for code.

Boris Beizer
Model-Driven Engineering

- Models as central artefacts
- Precise meaning
- Automation of engineering steps

Open Issues

- Heterogeneity of models
- Model management
- Models at runtime
- Model-based testing
- Model quality
- ...

Context | Test Models | Test the Test | Other Issues | Summary
Placement of My Work

- Integrated Development and Testing
- Early and continuous consideration of test aspects

Test Objectives

Analysis → Requirements
Design → Architecture
Specification → Models
Implementation → Code
Unit/Comp. Tests
Integr./Sys. Tests
Operation & Maint. → Productive System
Auditing Supervision

Today’s testing

Model-driven/Model-based testing

Online tests
The role of test models

- **System model** represents **requirements** and **Test model** represents **System**
- **System** validates **Test system** and **Test model** validates **System**
- **Requirements** realizes both **System model** and **Test model**
Model-Based Test Generation

Test models enable

- objective test procedures
- test reuse
- test quality assessment, and
- technology-independence.
Issues – the full spectrum of system engineering

- Model the tests
  - Concepts
  - Languages

- Test the tests
  - Correctness
  - Consistency
  - Coverage
  - Quality

- Generate the tests
  - Transformations
  - Test pattern

- Execute the tests
  - Platform
  - Adapters

- Deploy the tests
  - Distribution

- Optimize the tests
  - Performance

- Manage the tests
  - Processes
  - Variants

- ...

Context | Test Models | Test the Test | Other Issues | Summary
Myself

Ina Schieferdecker

1. Diploma, Mathematical Computer Science
   Humboldt University
2. PhD on Performance Enhanced
   Formal Specifications
   Technical University
3. Work at GMD FOKUS → Fraunhofer FOKUS
4. Research Stays at ICSI, CRIM, Uni Oslo

Prof. on Design and Testing of
Communication-Based Systems
Technical University Berlin
http://www.ets.tu-berlin.de

Head of Competence Centre
Modelling and Testing
Fraunhofer FOKUS
http://www.fokus.fraunhofer.de/motion

Co-Founder of Testing Technologies, Berlin
http://www.testingtech.de

Board Member of ASQF
http://www.asqf.de

Member of German Testing Board
http://www.german-testing-board.de

Lead of TTCN-3 Maintenance Team
http://www.ttcn-3.org
Issues – the full spectrum of system engineering

- Model the tests
  - Concepts
  - Languages
- Test the tests
  - Correctness
  - Consistency
  - Coverage
  - Quality
- Generate the tests
  - Transformations
  - Test pattern
- Execute the tests
  - Platform
  - Adapters
- Deploy the tests
  - Distribution
- Optimize the tests
  - Performance
- Manage the tests
  - Processes
  - Variants
- ...

Context  Test Models  Test the Test  Other Issues  Summary
TTCN-3 - The Testing and Test Control Notation

- Test specification and execution technology for different tests
  - Distributed, platform-independent testing
  - Integrated graphical test development, documentation and analysis
  - Adaptable, open test environment
- Adopted by ETSI and ITU since 2000

- The testing middleware
  - Unifying the definition and documentation of tests in different development phases
  - Unifying the tests for various systems (cockpit, powertrain, cabin, etc.)
  - Unifying the test infrastructure

- Areas of Testing
  - Regression Testing
  - Conformance/Functionality Testing
  - Interoperability/Integration Testing
  - Load/ Stress Testing
TTCN-3 Adoption

AUTOSAR

Usages
- Telecommunication
  - VoIP
  - IMS
  - WIMAX
  - LTE
  - ...
- Automotive
  - Autosar
  - cockpit applications – MOST Forum
- Avionics systems – ESA
- Medical devices – HL7
- Power transmission systems
- Smart cards
- Transport
- ...

Context Test Models Test the Test Other Issues Summary
MDA: A Model-Centric System Development

- Enable test development within UML and along MDA principles
- Proposed the UML Testing Profile
The UML Testing Profile

- Definition of a testing profile to capture all information that would be needed by different test processes
  - To allow black-box testing (i.e. at UML interfaces) of computational models in UML
- A testing profile based upon UML 2.0
  - That enables the test definition and test generation based on structural (static) and behavioral (dynamic) aspects of UML models, and
  - That is capable of inter-operation with existing test technologies for black-box testing
- Define
  - Test architecture: Test structure, test components and test configuration
  - Test data: Test data and templates used in test procedures
  - Test behavior: Dynamic aspects of test procedures
  - Time: Time quantified definition of test procedures
- OMG standard since 2005
Selected Publications


Selected Publications


Issues – the full spectrum of system engineering

- Model the tests
  - Concepts
  - Languages
- **Test the tests**
  - Correctness
  - Consistency
  - Coverage
  - Quality
- Generate the tests
  - Transformations
  - Test pattern
- Execute the tests
  - Platform
  - Adapters
- Deploy the tests
  - Distribution
- Optimize the tests
  - Performance
- Manage the tests
  - Processes
  - Variants
  - ...

- Huge test suites (>60000 LOC), e.g. for:
  - Session Initiation Protocol (SIP)
  - Internet Protocol Version 6 (IPv6)
- Suffer from quality problems like any larger software!
MDA Extended: Integrated Modelling and Testing

- Extend the MDA with a dedicated test model chain
- Define test model levels according to system model levels
- Tool-supported quality-oriented MDA extension

Requirements

Business concepts

Plattform independent IT model

Plattform dependent IT model

System platform

PSM

generate
generate
generate

test

PSM

generate
generate
generate

test

System

Test-system

CIM  Computation Independent Model
CIT  ... Test
PIM  Platform Independent Model
PIT  ... Test
PSM  Platform Specific Model
PST  ... Test

Context  Test Models  Test the Test  Other Issues  Summary

FU Berlin, Kolloquium, Febr, 2008
Used for Japanese IT applications
Integrated Modelling Infrastructure: Concept and Design

- Model-Driven Development is “orchestration” of **modelling services**
- Goal of ModelBus = **Infrastructure** for modelling service **integration** and **interoperability**

1. Developer (Tool user)
   - **getUMLModel()**
   - **checkUMLModel()**
   - **transformUML2EJB()**

**Repository tool:**
Provides version control for software artifacts

**OCL checker tool:**
check the well-formed-ness of models against OCL constraints

**QVT tool:**
Transform models between different formalisms

= uses functionalities of / shares models with
Integrated Modelling Infrastructure: ModelBus

Lifecycle tools

Requirements Tool
Analysis Tool
Design Tool
Coding Tool
Deployment Tool
Simulation Tool
Verification Tool
Unit Test Tool

Management Tool
Control Tool
Orchestration Tool

Process Enactment

ModelBus

Core Services
Service Registry
Notification Service

Modelling Services
Model Verification
Model Merging
Model Composition
Model Transformation
Model Storage
Model Configuration
Model Validation
Model Traceability

Context  Test Models  Test the Test  Other Issues  Summary

FU Berlin, Kolloqium, Febr, 2008
Quantitative and qualitative assessment

- Quantitative:
  - Metrics
  - Value ranges

- Qualitative:
  - Rules
  - Guidelines
  - Patterns
  - Anti-patterns

- Quantitative and qualitative assessment not disjoint:
  - Pattern/anti-pattern $\rightarrow$ Metric: count occurrences
  - Metric $\rightarrow$ rules: metric violates threshold
Test Model Quality Improvement

Approach:

- Assess test suites,
- Detect issues,
- Restructure test suites.

→ Metrics, Patterns/Anti-Patterns

→ Refactoring
Instantiation of ISO quality model: quality of test specifications

- **Test Effectivity**
  - Test Coverage
  - Test Correctness
  - Fault-Revealing Capability
  - Test Effectivity Compliance

- **Reliability**
  - Maturity
  - Test Repeatability
    - Fault-Tolerance
  - Security
    - Recoverability
  - Reliability Compliance

- **Usability**
  - Understandability
  - Learnability
  - Operability
  - Test Evaluability
  - Usability Compliance

- **Efficiency**
  - Time Behaviour
  - Resource Utilisation
  - Efficiency Compliance

- **Maintainability**
  - Analysability
  - Changeability
  - Stability
  - Maintainability Compliance

- **Portability**
  - Adaptability
  - Resource Utilisation
  - Portability Compliance

- **Reusability**
  - Coupling
    - Flexibility
  - Comprehensibility
  - Reusability Compliance

- **Test Effectivity**
  - Test Coverage
  - Test Correctness
  - Fault-Revealing Capability
  - Test Effectivity Compliance

- **Context**
- **Test Models**
- **Test the Test**
- **Other Issues**
- **Summary**
TTCN-3 Anti-Pattern

- TTCN-3 “code smells”
  - Duplicated Code, e.g. *Duplicate Alt Branches*
  - References, e.g. *Singular Component Variable/Const./Timer*
  - Parameters, e.g. *Constant Actual Parameter Value*
  - Complexity, e.g. *Complex Conditional*
  - Default Anomalies, e.g. *Activation Asymmetry*
  - Test Behaviour, e.g. *Missing Verdict*
  - Test Configuration, e.g. *Idle Parallel Test Component*
  - Coding Standards, e.g. *Magic Values*
  - Data Flow Anomalies, e.g. *Unused Variable Definition*
  - Miscellaneous, e.g. *Over-specific Runs On*
**TTCN-3 Metrics**

- **Generic metrics**
  - lines of code
  - min, max, average lines of code for a function
  - min, max, average number of parameters for a function
  - number of unused variables
  - min, max, average cyclomatic complexity for a function
  - min, max, average function fan-in for a test suite
  - min, max, average function fan-out for a test suite
  - ...

- **TTCN-3 specific metrics**
  - number of verdict assignments
  - min, max, average execution time for a test case
  - min, max, average number of parallel test components employed for the execution of a test case
  - fan-in/fan-out for test control, test cases, altsteps
  - min, max, average number of parameters for a module, test case, altstep, template
  - ...

---

FU Berlin, Kolloquium, Febr, 2008
Test Variance Metrics

1. SUT is represented by TSI
   - set of ports of different port types allowing various data types
   - “system” clause in test case definition

2. TSI coverage – data input space
   - Quantitative similarity
     → distance measures
   - Qualitative similarity
     → partitioning method

3. Assumptions
   - all TSI ports are message-based port
Metrics computation
- Compute metrics with OCL, also with metric packages
- Based on meta models
- Generic metrics for different models

Results presentation
- Numerical reports
- Graphical presentation
- Trigger warnings to project manager
Quality Assessment: Guideline Checks

**Requirements**
- Six Sigma
- ISO 9000 Family
- Capability Maturity Model Integration (CMMI)
- Software Process Improvement and Capability Determination (SPICE)
- Failure Mode And Effects Analysis (FMEA)

**Development**
- Requirements Engineering
  - DOORS
- Model-based development
  - MATLAB/Simulink/Stateflow

**Quality Assurance**
- Test Specification
  - CTE ES / MTest

**Automotive Artefact Guideline-Checker**
Selected Publications


**Selected Publications**


Issues – the full spectrum of system engineering

- Model the tests
  - Concepts
  - Languages
- Test the tests
  - Correctness
  - Consistency
  - Coverage
  - Quality
- Generate the tests
  - Transformations
  - Test pattern
- Execute the tests
  - Platform
  - Adapters
- Deploy the tests
  - Distribution
- Optimize the tests
  - Performance
- Manage the tests
  - Processes
  - Variants
- ...
Selected Publications


Issues – the full spectrum of system engineering

- Model the tests
  - Concepts
  - Languages
- Test the tests
  - Correctness
  - Consistency
  - Coverage
  - Quality
- Generate the tests
  - Transformations
  - Test pattern
- Execute the tests
  - Platform
  - Adapters
- Deploy the tests
  - Distribution
- Optimize the tests
  - Performance
- Manage the tests
  - Processes
  - Variants
  - ...

Context  Test Models  Test the Test  Other Issues  Summary
Selected Publications


Issues – the full spectrum of system engineering

- Model the tests
  - Concepts
  - Languages
- Test the tests
  - Correctness
  - Consistency
  - Coverage
  - Quality
- Generate the tests
  - Transformations
  - Test pattern
- Execute the tests
  - Platform
  - Adapters
- Deploy the tests
  - Distribution
- Optimize the tests
  - Performance
- Manage the tests
  - Processes
  - Variants
  - ...

FU Berlin, Kolloquium, Febr, 2008
Selected Publications


# Issues – the full spectrum of system engineering

<table>
<thead>
<tr>
<th>Context</th>
<th>Test Models</th>
<th>Test the Test</th>
<th>Other Issues</th>
<th>Summary</th>
</tr>
</thead>
</table>

- Model the tests
  - Concepts
  - Languages
- Test the tests
  - Correctness
  - Consistency
  - Coverage
  - Quality
- Generate the tests
  - Transformations
  - Test pattern
- Execute the tests
  - Platform
  - Adapters
- Deploy the tests
  - Distribution
- Optimize the tests
  - Performance
- Manage the tests
  - Processes
  - Variants
- ...

Context Test Models Test the Test Other Issues Summary
Selected Publications


Summary

- Goal
  - Test model/test system as integral part of system engineering
  - For early and continuous quality considerations
- Test modelling techniques
- Test the test
  - Generate correct tests
  - Verify test correctness
  - Assess test quality
- Other aspects of quality engineering
MOTION’s Technologies

- Modeling plugins (UML, U2TP, ...)
- Metamodelling infrastructure (MOF) → medini via
- Testing infrastructure (TTCN-3) → TTworkbench via
- Invariant checking (OCL) → OSLO via
- QoS enabled Middleware (CCM) → Qedo via
- Tool coupling → ModelBus via
- … and various concrete solutions
Outlook

- Model quality is generic
  - Approach should be extended to various kinds of models
  - Generic quality aspects vs. specific quality aspects
  - Appropriateness of metrics and of rules

- Need for a model-oriented quality engineering approach
  - Model-based system management
  - Model metrics, patterns and anti-patterns
  - Model-based process integration
  - ...

Context  Test Models  Test the Test  Other Issues  Summary
Thank You for Your Attention!

Any Questions?