Test Requirements-Driven Testing:
Model-Based Test Automation with Fokus!MBT and ModelBus

Max Bureck, Marc-Florian Wendland | ETSI MBT User Conference 2011 | 10-20-2011 | Berlin |
Agenda

- Introduction
- Technologies
  - FOKUSIMBT
  - ModelBus
- Test Requirement Driven Approach
- Use Case
- Summary
- Ideas For Future Work
Introduction

What To Archive?

- General goals of MBT
  - Reduce time-/resource consumption, cope with ...
    - Repetitive work
    - Lack of documentation
    - Unstructured test design
  - Early start of testing

- Additional goals of FOKUS MBT
  - Flexible, integrative tool chain
  - Adapt to testing processes
  - Built upon established and well-known standards (UML, UTP, SysML)
• There is one ready-to-go process, that can be used out of the box
• One can choose to use the suggested process or adapt tool to own process
• We can develop support for other tools on request
• Similar tools exist, but don’t have a flexible service orchestration support
• Tracing from requirement to test executions
Technology Overview

ModelBus – Tool Integration Done Service Oriented & Model Driven

- Service-oriented communication infrastructure, based on WebServices
  - Deals with tool heterogeneity (model-driven tool integration)
  - Allows effective process automation

- Support for modeling services (e.g., transformations)

- Model aware repository service

- Easy integration of own (modeling) services

- Process automation is controlled by service orchestration
  - Business Process Modeling Notation (BPMN)
  - Business Process Execution Language (BPEL)

- Tool integration done over tool adapters, model repository and transformations between tool specific metamodels
- Repository service: like SVN for models
- Other model repositories exist, but no orchestrated modeling services
Methodology and Scenario

**Test requirements-driven approach**

- Testing efforts are often underestimated and starts too late
  - Testing should take place as early as possible
  - Depends on the level of testing

- The consolidated requirements specification is the earliest point in time were testing activities make sense and are purposeful for acceptance and system testing

- Based on the system requirements dedicated test requirements are defined
  - An item or event that can be verified by one or more test cases
  - Semi-informal specification of what a test case should verify

- Allows prioritization and assessment of testing activities before detailed system design is specified

This is one of many possible methodologies, that can be implemented using FOKUSIMBT
Methodology and Scenario
Process And Scenario Domain

- General steps in process
  - Requirements modeling -> create SysML model
  - System & Test modeling -> generate test cases
  - Execute test cases -> store results in model
  - Analyze test results in FOKUS/MBT editor

- Most activities in the process automated through ModelBus and run in background

- Scenario use case: Modeling and testing of a microwave oven
This slide will be shown in parallel to the tool screenshots
Process Orchestration on ModelBus
Definition of Requirements in ProR
Checking in Requirements with ProR
Checking Out Model With Papyrus
System Modeling in Papyrus
Test Modeling in Papyrus (Test Requirements)
Test Modeling in Papyrus (Test Configuration)
Test Modeling in Papyrus (SUT State Chart, Tester's Perspective)
Checking in System & Test Model in Papyrus
Check Out Generated Testscripts in TTWorkbench
Execution of Generated Test Cases With TTWorkbench
Check in Test Results in TTWorkbench
Check Out Results in FOKUSIMBT
Analyze Results in FOKUSIMBT
View Test Results in Word File

Summary
- Total Number of Test Runs
- Total Number of Test Cases
- Total Number of Test Cases Passed
- Total Number of Test Cases Failed
- Total Number of Test Cases Reported Errors
- Total Number of Test Cases Tested/Untested

System Requirements
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PR_1</td>
<td>The user can start the HID by pressing the &quot;on&quot; button.</td>
</tr>
<tr>
<td>2</td>
<td>PR_2</td>
<td>The user can stop the HID by programmed with an...</td>
</tr>
</tbody>
</table>
Summary
Model-based test automation

- Modeling and test modeling based on well-known, proven and established standards
- Early testing based on (test) requirements
- Requirements traceability from initial elicitation all the way down to test code
- Several autarkic services worked together to realize a tool chain for a particular methodology
- Orchestration was done via BPMN
- Generation of fully executable test cases
- Result analysis and report generation with Fokus!MBT Core Editor and MS Word
Open architecture allows addition of services to the tool chain

Ideas For Future Work
Possible Future Developments

- Rich requirements modeling
- Better support for iterative (test) development
- Model based testing of legacy code (without system model)
- Integration into continuous integration systems
- ...

28
Thanks for your attention

QUESTIONS?