The ITEA D-MINT Project; Overview, Results and Lessons Learnt

Dr. Colin Willcock
Nokia Siemens Networks
October 19th 2011
Presentation Outline

Project Overview
- Project Goals
- Project Relevance
- Project Structure
- Industrial Case Studies
- Methodologies & Tools

Project Results

Lessons Learnt
D-MINT: Project Overview
Project Goals

• To develop the methodologies, tools and industrial experience to enable European industry to test more effectively and more efficiently

• To drive the deployment of Model-based testing technology into European industry
D-MINT: Project Relevance
Project Relevance

Number of successful software projects still less than 1/3¹

Software Project Success

¹The Standish Group 2006; The Chaos Report
Project Relevance

• The importance of software in product development is increasing
• 40-60% of product development costs goes in testing
• Improving testing will directly impact competitiveness
• In addition we are facing a series of new specific testing challenges
Project Relevance

Increasing complexity of products

- GSM Specifications 1306
- 3G Specifications 2290

2G total:
3G total:
total:
Project Relevance

• Pressure to shorten time to market
  • New systems and services must be available quicker
  • How can we reduce testing time?
• Pressure to improve quality
  • SW outage average time for Network elements measured in seconds per year
  • How can we improve testing quality (and quantify it)
D-MINT: Project Structure
Project Consortium

The ITEA D-MINT project; Overview, results and lessons learnt

October 2011
D-MINT Project Structure

WP1
Industrial cases and demonstrators
- requirements
- evaluations

WP2
Model-based testing principles and methods
- concepts
- specification
- processes
- evaluations

WP3
Tools and tool chain

WP4
Exploitation

WP5
Dissemination, standardisation and training

D-MINT - Deployment of Model-Based Technologies to Industrial Testing

© Nokia Siemens Networks
Application Areas

- Milling Machines
- Industrial Engineering
- Cars
- Street Lights
- Telecoms
- Video Conferencing
D-MINT: Industrial Case Studies
Daimler Automotive Case Study

Daimler focus in D-MINT

Test cases

Architecture-based approach

Usage-based approach

Car electronics architecture

Test model

Requirements

Blinker

Simulink/Stateflow model

Daimler-internal TestSpec formalism

As target container for the test cases

Test script generation

Test execution & evaluation

dSpace tools

PROVEtech:TA

TPT

Test cases to be executed in HIL test environment

Covering

- model lines
- test stages
**NSN Telecom Case Study**

- **SUT:** Mobile Switching Server (MSS)
- MSS is responsible for establishing calls and to control the handover of mobiles among different cells
- Models are built and test cases are generated with QTronic tool

**Test environment for MSS**
ABB Production Engineering Case Study

- SUT: *soft starter* (a device to smoothly start and stop an electrical motor)
- Design models in use: UML use cases and class diagrams
- Test model in use: usage model
- Test model derived from requirements and UML models, then test cases are derived from test model and executed
ETSI Telecom Case Study

- **SUT: IP Multimedia Subsystem (IMS)**
- The case study focuses on the assessment of interoperability of basic services such as basic Voice over IP (VoIP) call and instant messaging between two distinct IMS networks
- Both functional and conformance tests
- UML state charts are used to model the SUT, test cases are derived from this
Soraluce/Ideko Production Engineering Case Study

- SUT: DIGITMILL mechatronic solution as part of a milling machine
- Focus in this case study is to get a more systematic test process based on model-based testing
- Models in use: several UML diagrams (component, architectural, sequence, state diagrams)
- Test case derivation from UML diagrams
- Demonstrator: DIGITMILL milling solution (used e.g. for manufacturing wind power towers)
Trimek/Datapixel Production Engineering Case Study

- SUT: Coordinates Measuring Machines (CMM) control software (which is called CDMS) for controlling a measuring system
- Focus: test case derivation from UML models
- Models in use: UML class, sequence, state diagrams
Eliko City Street Lights Case Study

- **SUT:** Eliko street lighting control system feeder box control unit (FBCU)
- Models for the SUT: UML state charts, produced with tool Poseidon
- Elvior test generator derives test cases from state charts
Tandberg case study

- **SUT:** Video conferencing systems with the support for multiple simultaneous calls and presentations
- **Focus:** Model-based functional, stress, and robustness testing
- **Models in use:** UML state machines, sequence diagrams, and profiles such as MARTE and UML Testing Profile
D-MINT: Methodologies and Tools
D-MINT basic technical idea  
(automotive example)

On the basis of models test cases will be derived. Test cases will be used to dynamically check the system under test.
Several technology application domains

Models
- Simulink models
- TPT model
- Class diagram models
- Interaction models
- CamelView model
- Stateflow models
- Architecture
- Use case models

SUTs

Test Cases

D-MINT Test Framework

© Nokia Siemens Networks
D-MINT Approach

Design/development models

Classical approach

Advanced approach

Requirements

Results of FMEA/Risk Analysis

Test model

Test Cases
Architecture-Driven Test Design

- **Features**
- **Functions**
- **Hardware and software components**
- **Technical layout**

**Logical Architecture**
- Function Module
  - Vehicle Reaction
  - Event/Reaction
- is realized by
  - Sensor Block
  - Fet. Block
  - Actor Block

**Technical Architecture**
- is executed by
  - Sensor
  - Comp.
  - Bus
  - Actuator

**Topology**
- is located
  - Installation location
- goes through
  - Installation location
- is located
  - Installation location

© Daimler
NSN Case Study Mapping

Functional requirements in Doors and SysML

UML structure, data and behavior models

Specific server configurations defined in product documentation (not explicitly tested)

Incorporated in test system design – no explicit model

Addressed in the case study
# D-MINT Tools

<table>
<thead>
<tr>
<th>Modelling</th>
<th>Test Generation</th>
<th>Test Execution</th>
<th>Planning &amp; Mng</th>
<th>Test Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conformiq Qtronic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TTmodeler</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TT-Workbench</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Piketec TPT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>iXtronics CAMeL-View</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pattern-Driven Test Modeler (FOKUS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test Generator Simulink/Stateflow (FOKUS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Statistical Testing Framework (IESE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Elvior Test Generator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Elvior MessageMagic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NetHawk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SQS TestWorkflow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Statistical Testing Framework (IESE)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prototype: Elvior Test Generator
Tool: Pattern-Driven Test Modeler (FOKUS)
Test Data Quality Analyser (FOKUS)
D-MINT: Project Results
Demonstrators

ITEA 2 SYMPOSIUM 29-30 OCTOBER 2009, MADRID, SPAIN

EXHIBITION AWARD 2009

D-MINT
D-MINT Innovations

1. System-architecture-driven testing

2. Model-based integrated system and test development

3. Automated test-case refinement in sync with system model refinements

4. Automated consistency checks of requirements, models and test cases

5. Statistically controlled model-based test processes
## Competitive Comparison Criteria

<table>
<thead>
<tr>
<th>Comparison Criteria</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Design Tool</td>
<td><strong>Ease of use</strong> For e.g. a simpler notation such as OCL vs. fully blown language such as Java &amp; UML etc.</td>
</tr>
<tr>
<td></td>
<td><strong>Execution</strong> Ability to execute generated test cases automatically against real SUTs</td>
</tr>
<tr>
<td></td>
<td><strong>Scalability</strong> Scalability of the tool for large and complex models</td>
</tr>
<tr>
<td></td>
<td><strong>Reporting</strong> Reporting on coverage, test cases and their linkage to the model</td>
</tr>
<tr>
<td></td>
<td><strong>Debugging</strong> Ability to debug models and test cases, simulate</td>
</tr>
<tr>
<td>Test Case Formats</td>
<td><strong>Test Case Formats</strong> Test cases in many formats from human readable documents to standard languages such as C, C++, Java, TTCN-3 etc.</td>
</tr>
<tr>
<td>Integration</td>
<td><strong>Requirement Mgmt. Tools</strong> Work with requirement management tools such as DOORS or ability to integrate with such in-house tools</td>
</tr>
<tr>
<td></td>
<td><strong>Test Management Tools</strong> Work with test management tools such as HP Quality Centre or ability to integrate with such in-house tools</td>
</tr>
<tr>
<td></td>
<td><strong>System Modelling</strong> Work modelling tools such as Tau, Rhapsody, Enterprise Architect or ability to integrate with other tools</td>
</tr>
<tr>
<td></td>
<td><strong>Eclipse Support</strong> Support for Eclipse for seamless working with other testing and development tools such as modelling, requirement management or test management tools</td>
</tr>
</tbody>
</table>
Competitive Position

- Ease of Use
- Execution
- Scalability
- Reporting
- Debugging
- Test Case Formats
- System Modelling Integration
- Test Management Tools Integration
- Requirement Mgmt. Tools Integration
- Eclipse Support

Graph showing the competitive position of various tools:
- Reactis Suite
- Test Designer
- T-VEC RAVE
- D-MINT
Test process improvement measuring

Based on metrics and numbers
Overall Case Studies Evaluation Results

The D-MINT case studies have shown that

- adaptation efforts for MBT (initial investments in training, integration in existing test processes,…) are high, but this effort is seen as an one-time activity
- MBT is very beneficial for testing activities with several iterations

D-MINT technologies enable

- **test cost reduction** of at least 15% by automation and efficient test case generation (reduction of time to locate defects as well as test case generation & execution efforts)
- a **higher test coverage** of at least 10%
- **improved handling of increased test complexity** by means of systematic applying (reusable) models (here, the vision of test models instead of test cases is supported)
Usage of D-MINT technology by the Transfer Partners

- ELIKO has used D-MINT tools and technology for products that are already delivered and in use in the field
- ABB decided to use D-MINT results for the next version of the Softstarter
- Daimler has decided to use D-MINT technologies for ECU-Software Testing

Tool Vendors who already put their Products on the Market

- iXtronics Toolbox
- Testing Tech TTmodeler
- Conformiq Qtronic
D-MINT: Lessons Learnt
Lessons Learnt

1. MBT can be successfully applied in many industrial domains
   1a. The initial effort is high, but the payback is worth it

2. Be aware and take into account the industrial realities
   2a. Simple examples are not relevant
   2b. Do not try and change the world overnight
   2c. Usability is important

3. Don’t try and reinvent the wheel
   2a. Use standardized interfaces
   2b. Use open source frameworks
Be realistic when trying to develop and deploy new technologies. Whole-scale replacement of toolchains overnight is not the industrial reality.
In an industrial setting usability and stability are not just, nice to have, but deciding factors in any possible future technology selection.
D-MINT Tools in Eclipse

Modelling | Test Generation | Test Execution | Planning & Mng | Test Analysis
---|---|---|---|---
Abo MATERA | QTRONIC™ | TestingTech TTworkbench | Abo MATERA
TTmodeler | FOKUS MDTester | Elvior MsgMagic | FOKUS Test Quality
FOKUS MDTester | MOTES |
Conclusion

D-MINT has shown 8 case studies in 5 quite different application domains using a common methodological framework

By means of the case studies it could be shown that cost reduction, higher coverage and better test complexity management is possible by using D-MINT test technologies

Several D-MINT technologies are already in industrial use (i.e. fast exploitation was already accomplished during project run (ABB, Daimler, Eliko))

Experience Package Asset Box and Learning Package published on D-MINT website www.D-MINT.org