eDeltaMBT
Delta-oriented Model-based Software Product Line Testing

Sascha Lity, Malte Lochau, Ina Schaefer
Eclipse Testing Day 2012, 05.09.2012, Darmstadt
Outline

Introduction

Delta-oriented Model-based SPL Testing

Tool Support

Case Study Body Comfort System

Conclusion
Software Product Lines (1/2)

1980’s
1980’s 2012

- Mass customization [Dav87, Car11] of complex (software) systems
- **Software Product Lines**
  
  [...] explicit specification of commonality and variability between variants in a family of similar [software] products by means of features [PBL05].
SPL Engineering [PBL05]

- Domain engineering: design for reuse
- Application engineering: design with reuse
Software Product Lines (2/2)

SPL Engineering [PBL05]
- Domain engineering: design for reuse
- Application engineering: design with reuse

SPL Philosophy
- Features denote explicit product configuration parameters
- Common core platform to derive product variants
- Systematic reuse of engineering artifacts among product variants
Testing Software Product Lines

Product-by-product testing is infeasible

- High number of potential product variants
- Limited resources
- Redundancies due to commonality/similarity
Testing Software Product Lines

Product-by-product testing is infeasible

- High number of potential product variants
- Limited resources
- Redundancies due to commonality/similarity

*Model-based testing is well-suited for testing software product lines [cf. Olimpiew, 2008]*
Model-based Testing

Model-based testing is the automation of the design of black-box tests [UL06].

Test Model

\[ i_k, \ldots, i_2, i_1 \]

\[ o_1, o_2, \ldots, o_k \]

conforms?

IUT
Model-based Software Product Line Testing

Regard commonality and variability
Adapt reuse principles to SPL testing

Reusable test model
Reusable test cases
Reusable test results

Lity et al. | eDeltaMBT | 7
Model-based Software Product Line Testing

- Regard commonality and variability
- Adapt reuse principles to SPL testing (?)
  - Reusable test model
  - Reusable test cases
  - Reusable test results
Delta-oriented Model-based SPL Testing

Delta-oriented Test Modeling

- Adaption of delta modeling [Sch10] to state machines
- Definition of reusable test models

Delta-oriented Test Artifact Evolution

- Adaption of principles of regression testing
- Incremental evolution based on changes of test models
- Reuse of test cases and test results


[Lity et al.: Delta-oriented Model-Based SPL Regression Testing, PLEASE 2012]
Delta-oriented Test Modeling

Feature Configuration

Feature Model

Application Condition

Core Product

T1: e1/e2
T2: e3/
T3: e2/
T4: e5/e6
T5: e4/e6

Delta 1 Add

S1
T7: e1/

S2
T6: e3/

S3
T5: e4/e6

Core + Delta 1

Core + Delta 1 + Delta 2

Product

T7: e1/

T6: e3/

T4: e5/e6

T5: e4/e6

Delta 2 Rem

S1
T7: e1/

S4
T6: e3/

S2

Core + Delta 1 + Delta 2 + Delta 3

Delta 3 Mod

S3
T5: e5/e2;e3;e4

S2

T4: e5/e6

T3: e2/

T7: e1/

T5: e4/e6
Delta-oriented Test Artifact Evolution

\[ \delta \Delta m_i, \Delta m_{i+1} \rightarrow \Delta m_{i+1} \]

\[ \delta \Delta g_i, \Delta g_{i+1} \rightarrow \Delta g_{i+1} \]

\[ \delta \Delta p_i, \Delta p_{i+1} \rightarrow \Delta p_{i+1} \]

\[ \delta \Delta s_i, \Delta s_{i+1} \rightarrow \Delta s_{i+1} \]
Tool Support

- Crucial for efficient SPL testing
  - Automated test model generation
  - Automated test case generation
  - Automated test artifact evolution
  - Automated test case execution
  - ...

- Using existing (well-established) frameworks/tools
  - Eclipse Modeling Framework
  - EMF Validation Framework
  - Eclipse/RCP extendable
  - IBM Rational Rhapsody (Eclipse Plug-in)
Tool Chain

- Feature Modeling
- Configuration Management
- Delta Repository
- Visualization
- ATG

Import Test Models

Export Test Cases
eDeltaMBT – GUI
Example – Feature Model

eDeltaMBT – Feature Modeling
Example – Configurations

Feature Configuration Core
Example – Configurations

Feature Configuration Core

Feature Configuration P1
Example – Configurations

Feature Configuration Core

Feature Configuration P1

Configurations

- 8 features combinable to 6 possible product variants
eDeltaMBT – Configuration Management
eDeltaMBT – Configuration Management
eDeltaMBT – Configuration Management
Example – Delta Repository

Core Test Model
Example – Delta Repository

Core Test Model

DeltaAddTea
Example – Delta Repository

Core Test Model
eDeltaMBT – Delta Repository
eDeltaMBT – Delta Repository
eDeltaMBT – Test Model Import

[Image of a dialog box for generating a test model]

- Choose a Configuration Repository: platform:/resource/EclipseTestingDay/VMC
- Choose a Configuration: P1
- Choose Delta Repository: platform:/resource/EclipseTestingDay/VMD
- Choose Product Core: Core
- Testmodel Name: P1

Buttons: Delta Selection, Generate, < Back, Next >, Finish, Cancel
eDeltaMBT – Test Model Import
Rhapsody – Test Case Generation

![Coverage Summary](image)

- Total Coverage: 27
- State Coverage: 12
- Transition Coverage: 15

Stop | Hide Details | Close
Objectives:

- Application of the delta-oriented SPL testing approach
  - Application of delta test modeling
  - Validation of tool support
- Comparison of results with an existing SPL testing approach (MoSo-PoLiTe [OZLG11])
Body Comfort System

#Features: 27
#Products: 11616
Results (1/3)
Results (2/3)

[Image of a software testing tool and a graphical model]

Lity et al. | eDeltaMBT | 25
Results (3/3)
Conclusion

- SPL testing is challenging
  - Efficient test approaches are needed
  - Open field in research
- Combination of model-based and regression testing
  - Reuse of test artifacts
  - Incremental evolution of test artifacts
- Tool support based on well-established frameworks/tools
- Application of testing approach and tool support had positive results
Future Work

- Modeling with graphical editors
- Automated test artifact evolution
- (Semi-)Automated reasoning about test case and test result reuse
- RCP extension with further research results
- Tool chain extension

...
Thank You for Your Attention! Questions?

TU Braunschweig, Institute for Programming and Reactive Systems
Sascha Lity
Email: lity@ips.cs.tu-bs.de
References I


Stanley M. Davis. 
*Future Perfect.*

A. Fantechi and S. Gnesi.
Formal modeling for product families engineering.

Delta-oriented model-based spl regression testing.
References II


