Analyzing the Impact of Changes to Variability Models on Products

Dr. Rick Rabiser

Johannes Kepler University Linz, Austria
rick.rabiser@jku.at
http://ase.jku.at/staff/rabiser
http://mevss.jku.at
November 2012, CREST-23, UCL, London
Product Line Research in the CD Lab

- Flexible Product Line Modeling
- Configuration Tools for End-Users
- Model Fragments
  - Product Line Evolution
  - Case Studies
- Multi Product Lines
  - Product Line Bundles
  - Distributed Configuration

Christian Doppler Laboratory on Automated Software Engineering

- Sales Support Workflow
  - Focus on End Users
  - Deployment Support

Refactoring & Maturation

- Modernization Packages
- Business Calculations

Industrial Rollout

2006 2007 2008 2009 2010 2011 2012 2013
Impact Analysis in SPLs: Context, Motivation, Goal

- Maintenance and evolution of product lines is increasingly complex

- We developed a product line engineering tool suite, the DOPLER tool suite

- We observed the evolution of this industrial product line while it was refactored for industrial use cases by Siemens VAI

→ Developers requested
  Support for Analyzing the Impact of Changes on Derived Products
DOPLER Product Line Models

Which DOPLER tools do you want?

none  CW  ...  ...  ...  ...  ...
Shall Authentication be supported?

yes  no
Which mode?

Decision model defines end-user perspective on variability

Asset model defines reusable artifacts with links to the implementation

Included assets used by generators to automate product derivation
Product Derivation

![Diagram of Product Derivation]

- **Domain Engineering**: Variability Model described in Core Assets, used by Application Engineering.
- **Application Engineering**: Product Configurations used by Generators, create Product Outputs.
Product Derivation

<?xml version="1.0" encoding="UTF-8"?>

<product name="ConfigurationWizard" ...>
  <plugins>
    <plugin id="org.eclipse.rcp"/>
    <plugin id="at.jku.ase.cw.ui"/>
    <plugin id="at.jku.ase.dopler.core"/>
  </plugins>
</product>

CW Feature selected
Authentication = "PW"
...

<?xml version="1.0" encoding="UTF-8"?>

<product name="ConfigurationWizard" ...>
  <plugins>
    <plugin id="org.eclipse.rcp"/>
    <plugin id="at.jku.ase.cw.ui"/>
    <plugin id="at.jku.ase.dopler.core"/>
  </plugins>
</product>
Problem

Changes may affect product configurations
- Features chosen for products are not available any more
- Additional features become available for products

CW Feature selected
Authentication = “PW”...

Diagram:
- Domain Engineering
  - Variability Model
  - Core Assets
- Application Engineering
  - Product Configurations
  - Generators
  - Product Outputs
Problem

Changes may affect final product outputs

- Components may need to be updated for products
- Components may be added / removed in products

<?xml version="1.0" encoding="UTF-8"?>
<product name="ConfigurationWizard" ...>
<plugins>
  <plugin id="org.eclipse.rcp"/>
  <plugin id="at.jku.ase.cw.ui"/>
  <plugin id="at.jku.ase.dopler.core"/>
</plugins>
...
Variability Models are used to derive many products…
Impact Analysis: Common Approaches
Required Change Impact Analysis

Model Verification

Unit Testing

Integration & System Testing
Our Approach
Our Approach
Our Approach

Domain Engineering

Engineers evolve variability model

Variability Model

Application Engineering

Product Configurations

trigger

Generators

create

Product Outputs

Product Archive

Earlier versions of product outputs

archived in
Our Approach

1. Application Engineering
   - Product Configurations
     - trigger
   - Generators
     - create
   - Product Outputs

2. Domain Engineering
   - Variability Model
     - Engineers evolve variability model
   - Product Archive
     - Earlier versions of product outputs

3. Update product configurations

archived in
Our Approach

1. Product Archive
   - Earlier versions of product outputs

2. Variability Model
   - Engineers evolve variability model

3. Update product configurations

4. Product Outputs
   - trigger
   - create

Domain Engineering

Application Engineering
Our Approach

1. Product Archive
   - Earlier versions of product outputs

2. Variability Model
   - Engineers evolve variability model

3. Update product configurations

4. Product Outputs
   - Domain-specific Extensions
   - create
   - archived in

5. Comparators
   - trigger
   - compare
Our Approach
Required plugin at.jku.ase cw.ui would be removed in product
Reduce Variability: Remove the choice on meta modeling.
Always include “decisionking.metamodeling”?
Or make it dependent on another decision?
One final product would include two additional plugins.
Go to "at.jku.ase.decisionking.metamodelling" in the variability model.
Example

Include "decisionking.metamodeling" depending on DK!
Product configurations for two products change. The final product outputs (required assets) do not change!
Conclusions / Lessons Learned

• Approach currently also used to analyze the impact of product line evolution for a system supporting the management of tombs (RIP)

• Impact analysis in product lines can perform well for real world scenarios

• Existing product generators can be reused

• Simple artifact representations reduce execution time

• Product updates need to be automated


Find out more on http://ase.jku.at/
Future Work / New Christian Doppler Lab

• Monitoring and Evolution of Very-Large-Scale SW Systems
  • Feb 2013-Jan 2020; Head: Paul Grünbacher; [http://mevss.jku.at](http://mevss.jku.at)
  • **Goal:** Develop and improve methods for monitoring and diagnosing VLSS to support their systematic evolution
    • Requirements-based monitoring and diagnosis
      • Ensuring compliance with requirements after evolving a large-scale component in a VLSS
  • Multi-modeling in software ecosystems
    • Determining the impact of a platform change on globally distributed customer systems in a multi-level ecosystem
  • Application performance management
    • Diagnosing performance bottlenecks in distributed and heterogeneous web applications after software changes