MCS Workshop

Variability Modelling in DREAMS

Øystein Haugen, Franck Chauvel

SINTEF ICT

2 July 2014
Outline of the talk

1. What is Variability Modeling?
2. Variability Modelling with BVR / CVL
3. What can Variability Modeling be used for in our context?
4. Enhanced BVR tool bundle
1. What is Variability Modeling

- Focus on what varies
- and the decisions associated with the variability

- Making what varies explicit and "first class citizens"

- In particular our approach to variability modeling makes explicit the discrete options'
  - different from values of a parameter
# Traditional ways to model variability

<table>
<thead>
<tr>
<th>Constructs</th>
<th>Framework/Parameters</th>
<th>Union-of-all-systems</th>
<th>Domain Specific Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How?</strong></td>
<td>By mechanisms of a general language</td>
<td>As annotations to a language</td>
<td>By the specific language mechanisms</td>
</tr>
<tr>
<td><strong>Unforeseen modeling needs</strong></td>
<td>Function, Parameters, Type, Inheritance, Template, Plugin</td>
<td>Enhance the product line model</td>
<td>If not expressible, enhance the language</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Variability modeling with BVR/CVL

- **Generic**
  - Specification in BVR of base model variabilities
- **BVR/CVL**
  - Variability model
- **DSL**
  - Focused on a domain
  - Product line model in any MOF-compliant language
- **Base model**
- **Execute CVL**
  - Product models fully described in the base language.
  - All regular base language tools can be applied to these models
- **Resolved models**
  - Selection of a set of choices in the variation model
  - resolution models
CVL (Common Variability Language)
BVR (Base Variability Resolution)

- BVR is based on the CVL legacy
- BVR will try and satisfy the new demands
  - from VARIES
  - from DREAMS
  - from other interested parties
- BVR is not a subset or a superset of CVL
- BVR has a new metamodel
- BVR is in the first place only proprietary in the sense that no standardization effort will be attempted a priori
BVR Example from DREAMS domain

Diagram:

- **Platform**: Configurable Unit
  - Mandatory
- **Choice**
  - Hardware
    - Memory
    - CPU
    - IO
    - MultiCore
    - SingleCore
    - Xen
    - VirtualBox
    - VMWare
    - Hyper-V
  - OS Virtualization
    - Hardware Virtualization
      - x86
      - SPARC
      - ARM
    - XtratuM
    - KVM
  - Hypervisor
    - Select min 1 max 1

A BVR Example with more constructs

- **Constraint**: Parkassist implies (not FWD)
- **Type**:
  - Gear
  - Parkassist
  - type Engine
- **Optional**: type AWD "Front wheels must have the same Engine kind"
- **VClass Occurrence**:
  - Manual
  - FWD
  - Automatic
  - awdmanual:AWD
  - awdautomatic:AWD
  - wfront:Engine : [2, 2]
  - wrear:Engine : [2, 2]
  - kw300
  - kw500
3. What can Variability Modeling be used for in our context?

- **Configurations Generation**
  - BVR Tool Bundle handles explicit variability in a generic way
  - BVR could derive compliant platform specific model for any legal variant of a DREAMS system
  - BVR can be adapted to working at runtime as well as at designtime

- **Configuration Testing**
  - Validating/Testing an effective selection of configurations

- **Configurations Exploration**
  - Compute near optimal subsets of products/configurations
  - To be used offline and preparing online adaptation
From the DREAMS submission
4. Enhanced BVR tool bundle

- Analysis
  - ICPL tool

- Realization
  - Fragment editor
  - BVR Engine

- VSpec & Constraints
  - GMF editor
  - MVC editor

- Resolution
  - Graphical resolution editor
  - Table resolution editor

Diagram:
- BVR GMF graphics
- BVR EMF model
- EMF Base model
- EMF Product models
The Variability Editors

Purpose:
Create and modify the features and their constraints

The GMF editor applies direct graphic editing which is stored in a graphical resource

The MVC (Model View Controller) editor visualizes the feature diagrams from EMF

VSpec & Constraints
- GMF editor
- MVC editor

BVR GMF graphics
BVR EMF model
The GMF editor applies direct graphic editing which is stored in a graphical resource.

The MVC (Model View Controller) editor visualizes the feature diagrams from EMF.
The Resolution Editors

Purpose: Resolve the variability of the features

The resolution editors both work only on the EMF resource
Screenshots of the Resolution Editors

GMF edit

MVC edit

Table resolution

Graphic resolution
Purpose: Map the variability to the effects on the products

The fragment editors work seamlessly with different base editors

The BVR engine takes the BVR model and combines it with the base model to get product models
Screenshots of the Fragment Editor

Legend:
- **orange**: elem outside, but referred from inside
- **red**: elem inside plcmnt
- **violet**: elem outside, but referring to inside
- **turquoise**: orange+violet
- **blue**: elem inside replcmnt

Papyrus UML Tool colored by BVR Tool Bundle
Analysis tools

**ICPL tool**
- **Purpose:** Make different kinds of analysis tools based on explicit variability
- The original ICPL tool takes a variability model and creates an optimal set of resolutions.
- ICPL is based on interaction coverage arrays. Its results can be used for selecting best configurations to test.
- ICPL can also work on other representations of variability

**BVR EMF model**
Screenshot of running ICPL

Variability model

Resolutions generated by ICPL with 2-wise coverage arrays (11 configs)
Thank You
Questions, Remarks, Comments?

oystein.haugen@sintef.no
franck.chauvel@sintef.no