Towards Reasoning about Product Lines with Design Choices

Navpreet Kaur, Michalis Famelis
MoDeVVa @ MODELS 2019
Munich, DE
Software Product Line Basics

Feature model, Domain model, Feature mapping

Different feature **configurations** result in different variants

PL models a **set** of related, but different products
Washing Machine Product Line

Domain model

Feature model

Presence conditions
Washing Machine Product Line

Variant

Feature model

Locking → Waiting → Washing → Entry/TempCheck() → Unlocking

Washing Machine

- Wash
- Heat
- Delay
- Dry

All feature checks are marked as completed except for Dry.
# Two Different Kinds of Choices

<table>
<thead>
<tr>
<th></th>
<th>Variability</th>
<th>Design Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reason</strong></td>
<td>Market demands for product variants</td>
<td>Incomplete information, design alternatives, stakeholder conflicts, etc.</td>
</tr>
<tr>
<td><strong>Granularity</strong></td>
<td>Features</td>
<td>Decisions</td>
</tr>
<tr>
<td><strong>Expression</strong></td>
<td>Product line (PL) models</td>
<td>Partial models</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>Set of artifacts produced by combinations of features</td>
<td>Set of artifacts produced by combinations of decisions</td>
</tr>
<tr>
<td><strong>Horizon</strong></td>
<td>Long term</td>
<td>Short term</td>
</tr>
</tbody>
</table>
Uncertainty in the design of the feature model
Design 2: Incremental Heat?

Uncertainty in the design of the domain model
Design choices can affect the Feature model, Domain model, Feature mapping

They define a design space of product lines

Given a space of product lines, which one should be selected (and why)?

Feature combinations produce possible products

Decision combinations produce possible PLs
Software Product Lines with Design Choices

Motivation

How to model this design space?

What are relevant properties for exploring it?
How to check them?

What to do when properties are violated?

In what follows, signifies new contribution.
Tyson: an SPLDC language

Textual Syntax

Supports Class Diagrams, Statecharts

Feature and Choice submodels (FM, CM)

Emphasis on expressing mappings

Alloy semantics

```
CM { Mutex; IncrementalHeat; }
FM { Wash; Delay; Dry; Heat;
    FMConst: [ Mandatory (Wash) ] }

StateChart {
    State Locking; State Waiting;
    State Washing; State Drying;
    State UnLocking;
    Transition T1: Locking to Waiting
    Transition T2: Waiting to Washing
    Transition T3: Locking to Washing
    Transition T4: Washing to UnLocking
    Transition T5: Washing to Drying
    Transition T6: Drying to UnLocking
    Transition T7: Waiting to Waiting
}

Mappings {
    FMap {
        F1: { (Wash IN ) =>
            AND (Transition T3 IN, Transition T4 IN) }
        F2: { OR (Heat IN, Delay IN) =>
            AND (Transition T1 IN, Transition T2 IN) }
        F3: { (Dry IN) =>
            AND (Transition T5 IN, Transition T6 IN) }
    }
    DMap {
        D1: { Mutex IN =>
            XOR (Feature Heat IN, Feature Delay IN) }
        D2: { IncrementalHeat IN => Transition T7 IN }
    }
```
Currently: Limited Expressiveness

Domain model

Choice model

Feature model

Looks an awful lot like a feature model!

But not for long
Tyson: an SPLDC language

Textual Syntax

Supports Class Diagrams, Statecharts

Feature and Choice submodels (FM, CM)

Emphasis on expressing mappings

Alloy semantics

```
CM { Mutex; IncrementalHeat; }

FM { Wash; Delay; Dry; Heat;
   FMConst: [ Mandatory (Wash) ] }

StateChart {
   State Locking; State Waiting;
   State Washing; State Drying;
   State UnLocking;
   Transition T1: Locking to Waiting
   Transition T2: Waiting to Washing
   Transition T3: Locking to Washing
   Transition T4: Washing to UnLocking
   Transition T5: Washing to Drying
   Transition T6: Drying to UnLocking
   Transition T7: Waiting to Waiting
}

Mappings {
   FMap {
      F1: { (Wash IN ) => AND (Transition T3 IN, Transition T4 IN)}
      F2: { OR (Heat IN, Delay IN) => AND (Transition T1 IN, Transition T2 IN) }
      F3: { (Dry IN) => AND (Transition T5 IN, Transition T6 IN) }
   }
   DMap {
      D1: { Mutex IN => XOR (Feature Heat IN, Feature Delay IN))}
      D2: {IncrementalHeat IN => Transition T7 IN}
   }
```
Software Product Lines with Design Choices

Motivation

How to model this design space?

What are relevant properties for exploring it? How to check them?

What to do when properties are violated?
Constraining the Design Space using Properties

For a product-level property P, we define four SPLDC-level properties using the modalities:

- Use **All** for critical properties and **Some** for desirable properties.
- Use **Necessary** when you are sure it is needed and **Possible** when unsure but don’t want to exclude the possibility.

<table>
<thead>
<tr>
<th>Necessary for the product line</th>
<th>Possible for the product line</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All products have P</strong></td>
<td><strong>All products in All product lines</strong></td>
</tr>
<tr>
<td><strong>Some products have P</strong></td>
<td><strong>Some product in All product lines</strong></td>
</tr>
</tbody>
</table>
Necessary-Some (NS)

All product lines

Some product
Possible-Some (PS)

Some product line

Some product
Possible-All (PA)

Some product line

All products
Necessary-All (NA)

All product lines

All product

...
SPLDC Reasoning in Tyson

Tyson model → Alloy encoding of Tyson model → Alloy Analyzer → Property holds or Counterexample
Evaluation

Does property checking Tyson SPLDCs scale?

Reasonably

30 Feature models
30 Choice models
30 Domain models

Realistic product lines

a) Random combination
b) Random mappings

Alloy Analyzer

Property check result
Runtime

Fixed scope (after pilots)

http://splot-research.org/

S.P.L.O.T.
Software Product Lines Online Tools

Metamodel zoo

3 Properties
Inspired from
Van Der Straeten et al. UML’03

30 Feature models
30 Choice models
30 Domain models

Evaluation

Does property checking Tyson SPLDCs scale?

Reasonably

http://splot-research.org/

S.P.L.O.T.
Software Product Lines Online Tools

Metamodel zoo

3 Properties
Inspired from
Van Der Straeten et al. UML’03
Software Product Lines with Design Choices

Motivation

How to model this design space?

What are relevant properties for exploring it?
How to check them?

What to do when properties are violated?
Next Steps

1. Improve expressiveness of choice models
   Go beyond the closedness of feature modelling
   Model the evolution/elicitation of uncertainty

2. Modularize Tyson
   Who needs yet another DSL? UML Papyrus profile? MPS module?

3. Implement the property-based design space exploration vision of [MODELS’17]
   Some very interesting properties of the lattice structure

4. Go beyond Alloy
   Second order reasoners, Alloy*, QBF, ...

5. Apply to reuse of pull-based software development
Towards Reasoning about Product Lines with Design Choices

**Tyson**: [https://bitbucket.org/Navpreet15/dsl/](https://bitbucket.org/Navpreet15/dsl/)

---

**Software Product Lines with Design Choices** (SPLC)
- Design choices can affect the Feature model, Domain model, Feature mapping
- They define a design space of product lines
- Given a space of product lines, which one should be selected (and why)?

---

**Tyson: an SPLDC language**

- **Textual Syntax**
- Supports Class Diagrams, Statecharts
- Feature and Choice submodels (FM, CM)
- Emphasis on expressing mappings
- Alloy semantics

---

**Constraining the Design Space using Properties**

For a product-level property $P$, we define four SPLDC-level properties using the modalities: Use *All* for critical properties and *Some* for desirable properties

<table>
<thead>
<tr>
<th>Necessary for the product line</th>
<th>Possible for the product line</th>
</tr>
</thead>
<tbody>
<tr>
<td>All products have $P$</td>
<td>All products in Some product line</td>
</tr>
<tr>
<td>Some products have $P$</td>
<td>Some product in Some product line</td>
</tr>
</tbody>
</table>

---

**SPLDC Reasoning in Tyson**

- Tyson model
- Alloy encoding of Tyson model
- Alloy Analyzer
- Property holds
- Counterexample

---

Navpreet Kaur

Michalis Famelis @MFAmelis

Université de Montréal