Mixsets to Implement Variability of Software Product Lines

Presentation for Consortium for Software Engineering Research (CSER)

Abdulaziz Algablan – University of Ottawa

PhD in computer science

May 14, 2021

Mixsets

- Definition: ”A mixset is a named set of code/model fragments that can be mixed into a software system to add a feature, variant, or concern”

- Goal
  - Facilitate SPL variability in both modeling and programming languages.
  - Handle variability at model level.
  - Specify relationships between variant models.
  - Apply one mechanism to both abstract model and “embedded” native code.

Advantages of Mixsets

- Combines the best of two approaches
  - Annotative
    - Variation lives within the code.
    - Like #ifdef directives of C preprocessor
  - Compositional
    - Features are separate from the base code, which is shared by all features.
    - Similar to FOP (Feature Oriented Programming).
- Unified/Uniform
  - Can be applied across all entities in a language, including composition mechanisms such as aspects
- Encapsulates variability modeling
  - Mixsets can represent feature models.
  - Direct mapping of feature models in source code.
  - Explicit management of relationships between reused variable units, or mixsets.

Required Mechanisms

Mixsets to model variability

- First-class units in the language.
- Sub-entity of other language entities.

Artifact composer

- Compositional technique to merge different pieces of software.
- Umple uses mixins to compose identical entities.

Conditional compilation

- Conditional parsing of variable fragments of the code based on supplemented parameters.

Umple

- The approach is implemented in Umple.
  - A language that generates code based on modeling abstractions
  - Textual, with real-time rendering and editing of diagrams
  - Many modeling constructs, including:
    - Class models (attributes, associations, and generalizations)
    - State machine models (events, hierarchical states, transitions)
    - Traits
  - Can incorporate and generate code from Java, C++, PHP and Ruby
  - Analyses models to find many types of problems

- Try UmpleOnline via http://try.umple.org/

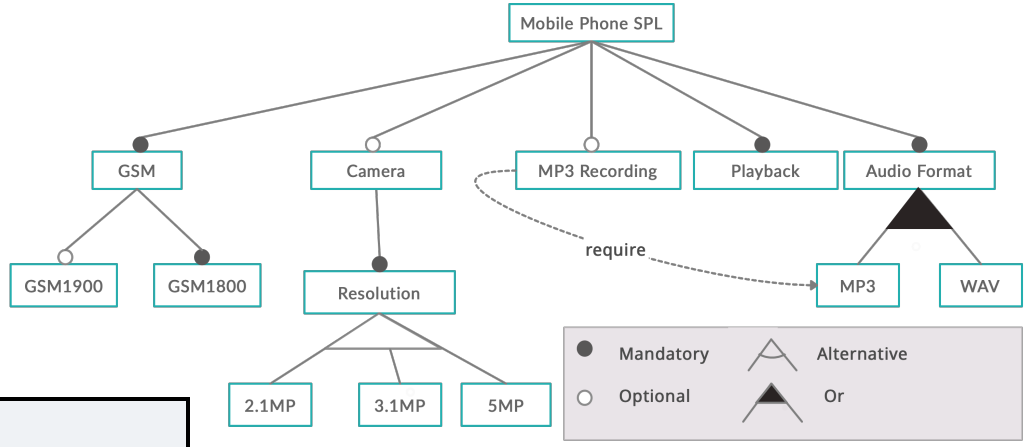
Mixsets Example

- A bank SPL.
  - Blue fragments are annotative.
  - Green fragments are compositional.
  - Compositional fragments can be in separate files.

```
1   class Bank {
2     1 -- * Account;
3     mixset Multibranch
4     {
5       1 -- 1..* Branch;
6     }
7   }
8
9   mixset Multibranch {
10    class Branch {
11      Integer id; String address;
12    }
13  }
14
15  class Account {
16    owner; Integer number; Integer balance;
17    mixset Multibranch { * -- 1 Branch;}
18  }
19
20  trait InterestBearingAccount {
21    Float interestRate;
22  }
23
24  class DepositAccount {
25    isA Account;
26    mixset OverdraftsAllowed {
27      Integer overdraftLimit;
28      isA InterestBearingAccount;
29    }
30  }
31
32  class LoanAccount {
33    isA Account, InterestBearingAccount;
34  }
```

Mixsets for Feature Modeling

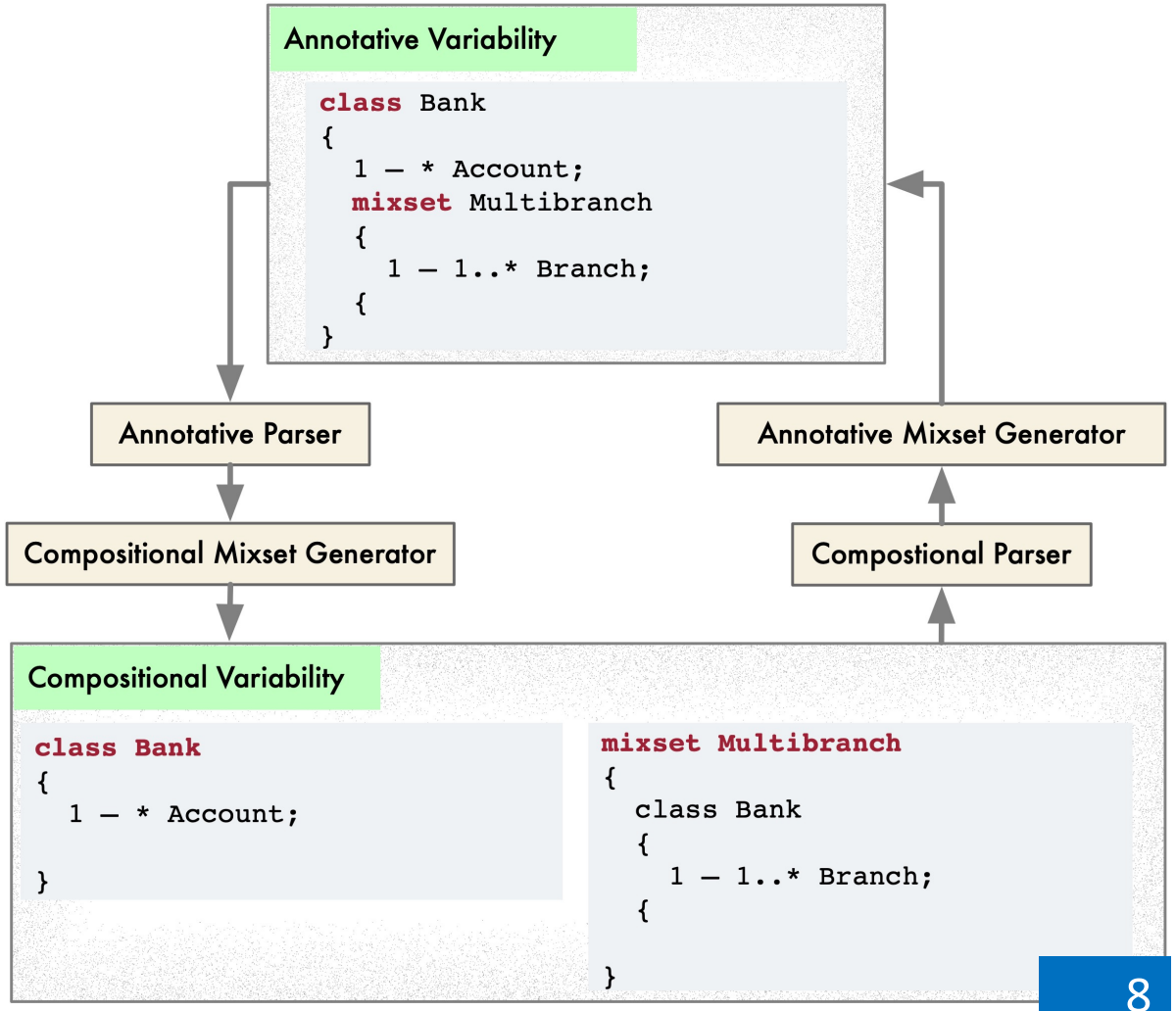
- A syntax to specify mixsets as features.
- Mixsets do not always map to features
- They can contain reusable (shared) mixsets.




```
1   require subfeature [GSMProtocol opt Mp3Recording
2     and Playback and AudioFormat opt Camera];
3   mixset GSMProtocol {
4     require subfeature [GSM1800 opt GSM1900]; }
5   mixset AudioFormat {
6     require subfeature [1..2 of {Mp3,Wav}]; }
7   mixset Mp3Recording { require [Mp3]; }
8   mixset Camera { require subfeature [Resolution]; }
9   mixset Resolution{
10    require subfeature [0..1 of {Res21MP, Res31MP, Res50MP}];}
11  use GSMProtocol; use GSM1900; use Playback; use AudioFormat;
```

Refactor between Annotation/Compositional

- Annotate fragments are treated as compositional fragments.
- Refactoring from compositional fragments to annotative fragments is possible.
- We call this “rewriting” of mixsets.
- Uses the abstract syntax tree (AST) to refactor mixsets .



Fine-grained Variability in Composition

- Extending aspect injection with labels.
- The code in green box results from generating Java code from the code in blue box.

```
1   class Main {
2    public static void main(String[] args) {
3     Hello_label:
4     Beautiful_label:
5     Wonderful_label:
6     mixset World {
7      System.out.print (" world !");
8     }
9    }
10  }
11
12  class Main {
13   mixset Hello {
14    after Hello_label: main(String ) {
15     System.out.print(" Hello ");
16    }
17   }
18   mixset Beautiful {
19    after Beautiful_label:
20      main(String) {
21     System.out.print(" beautiful ");
22    }
23   }
24   mixset Wonderful {
25    after Wonderful_label:
26      main(String) {
27     System.out.print (" wonderful ");
28    }
29   }
30  }
31
32  use Hello;
33  use Wonderful;
34  use World;
35
36  require subfeature [ opt Hello];
37  require subfeature [ opt Beautiful];
38  require subfeature [ opt Wonderful];
39  require subfeature [ opt World];
```

```
1   public class Main {
2    public static void main(String[] args) {
3     Hello_label:
4      System.out.print (" Hello ");
5     Wonderful_label:
6      System.out.print ("wonderful ");
7     System.out.print (" world !");
8    }
9   }
```

* The example is modified from: Krüger, J., Schroter, I., Kenner, A., Kruczek, C. & Leich, T. (2016). FeatureCoPP: Compositional Annotations. *Proceedings of the 7th FOSD*, 74–84. https://doi.org/10.1145/3001867.3001876

Case Studies

- Berkeley DB JE
  - Reduce the code size and hook methods.
- Refactoring Umple into a feature-driven software system
  - It is still in progress.
  - Usefulness of annotative fragments to identify variable elements.
  - Automate refactoring to compositional mixsets.

Limitations & Open Problems

- Lack of SPL variability awareness
  - Native support of SPL variability in the modeling language.
  - Formal analysis of SPL variability
- Modularity of fine-grained variability
- Expression-level variability

Demo

Conclusion

- Mixsets offer a combined variability mechanism
  - Seek smooth transformation between annotative and compositional fragments.
- Unified to work on both models and native code
- Offer mechanism to model variability as feature models

Questions?