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.
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/](http://try.umple.org/)
Mixsets Example

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

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

```plaintext
require subfeature [GSMProtocol opt Mp3Recording
and Playback and AudioFormat opt Camera];
mixset GSMProtocol {
  require subfeature [GSM1800 opt GSM1900]; }
mixset AudioFormat {
  require subfeature [1..2 of {Mp3,Wav}]; }
mixset Mp3Recording { require [Mp3]; }
mixset Camera { require subfeature [Resolution]; }
mixset Resolution{
  require subfeature [0..1 of {Res21MP, Res31MP, Res50MP}];}
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.

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?