Variability Implementation: Preprocessors

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Preprocessors

- A preprocessor is a tool that manipulates source code before compilation
  - A popular one is CPP for C and C++
- It defines directives to remove code fragments based on conditions
  - So, it supports conditional compilation

Conditional Compilation

- In conditional compilation, marked code fragments are removed before compilation
  - Conditional compilation directives, such as *#ifdef*, are annotations
- This is the most common mechanism for implementing variability in industry

Examples of Directives

- Preprocessor usually supports several conditional compilation directives
  - *#define F // it means a feature is enable*
  - *#ifdef // conditional for defined*
  - *#ifndef // conditional for not defined*
  - *#else // alternative*
  - *#endif // end of condicional block*

Example of Code (1)

```
public class PhotoListScreen extends List {

  public static final Command viewCommand;
  public static final Command addCommand;
  public static final Command deleteCommand;
  public static final Command editLabelCommand;
  // #ifdef includeSorting
  public static final Command sortCommand;
  // #endif
  // #ifdef includeFavourites
  public static final Command favoriteCommand;
  public static final Command viewFavoritesCommand;
  // #endif
...
}
```

S

F

Example of Code (2)

```
public class PhotoListScreen extends List {
  ...
  public void initMenu() {
    this.addCommand(viewCommand);
    this.addCommand(addCommand);
    this.addCommand(deleteCommand);
    this.addCommand(editLabelCommand);
    // #ifdef includeSorting
    this.addCommand(sortCommand);
    // #endif
    // #ifdef includeFavourites
    this.addCommand(favoriteCommand);
    this.addCommand(viewFavoritesCommand);
    // #endif
  }
}
```

S

F

Annotated code can be inside a method or anywhere in a class.

Preprocessor Options for Java

- Several tools are available for languages without build in preprocessor, like Java
  - Antenna: http://antenna.sourceforge.net/
  - JavaPP: http://www.slashdev.ca/javapp/
- Java options often mimic the syntax of the C preprocessor
  - Directives are usually written in comments
  - Instead of removing, they sometimes only comment code fragments

IDE and Non-Code Artifacts

- Most preprocessors can be integrated with IDE, such as NetBeans and Eclipse
- They can also be applied to different kinds of textual artifacts (not only code)
  - For instance, some tools work with documents and graphical models

Easy to Lean and Use

- Preprocessors are widely used in practice
  - They have a simple programming model
- They are easy to learn
  - C programmers are already familiar with directives
- Lightweight tools can be used

Easy to Annotate Features

- Preprocessors can be applied to different kinds of artifacts (including non-code)
  - Fine-grained variability allows precise control
- Annotations can refer directly to features
  - They allow searching and tracing feature code
- Well-suited for extractive adoption
  - Easy to add directives to an existing product

Drawbacks

- Academics often criticize preprocessors
  - Low code quality and maintainability
- Poor separation of concerns
  - Feature code is scattered and tangled
- Heavy use of annotations makes the code hard to understand
  - It is worse than Parameters when used inside statements or expressions
- They are considered error-prone

Example of Code

```
public class PhotoViewController extends AbstractController {
  …
  // #if includeSmsFeature || capturePhoto
  if (imgByte.length > 0)
    getAlbumData().addImageData(photoname, imgByte, albumname);
  // #endif

  // #if includeCopyPhoto && (includeSmsFeature || capturePhoto)
  if (imgByte.length == 0)
  //#endif

  // #if includeCopyPhoto
  getAlbumData().addMediaData(imageData, albumname);
  // #endif
…
```

Example of Code

```
public class PhotoViewController extends AbstractController {
  …
  // #if includeSmsFeature || capturePhoto
  if (imgByte.length > 0)
    getAlbumData().addImageData(photoname, imgByte, albumname);
  // #endif

  // #if includeCopyPhoto && (includeSmsFeature || capturePhoto)
  if (imgByte.length == 0)
  //#endif

  // #if includeCopyPhoto
  getAlbumData().addMediaData(imageData, albumname);
  // #endif
…
```

Complex dependency between features.

Example of Code

```
public class PhotoViewController extends AbstractController {
  …
  // #if includeSmsFeature || capturePhoto
  if (imgByte.length > 0)
    getAlbumData().addImageData(photoname, imgByte, albumname);
  // #endif

  // #if includeCopyPhoto && (includeSmsFeature || capturePhoto)
  if (imgByte.length == 0)
  //#endif

  // #if includeCopyPhoto
  getAlbumData().addMediaData(imageData, albumname);
  // #endif
…
```

Error-prone code.

Bibliography

- S. Apel, D. Batory, C. Kastner, G. Saake. **Feature-Oriented Software Product Lines: Concepts and Implementation**. Springer; 2013.
  - Section 5.3