



FAKULTÄT FÜR
INFORMATIK

FeatureIDE: Background

Thomas Thüm, Jens Meinicke

November 27, 2011

Feature-Oriented Programming (FOP)

- ▶ Introduced 1997 by Christian Prehofer
- ▶ Based on Object-Oriented Programming
- ▶ Features realize functionalities
- ▶ Features are cross-cutting to objects
- ▶ Features modularize fragments from certain classes
- ▶ Fragment contains some methods/fields of a class belonging to one functionality
- ▶ Goals: code traceability, software customization

FOP Example

```
package util;
class Calc {
  void add() {
    e0 = e1 + e0;
    e1 = e2;
  }
}
```

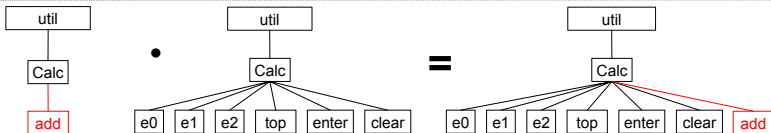
feature: Add

```
package util;
class Calc {
  int e0 = 0, e1 = 0,
      e2 = 0;
  void enter(int val) {
    e2 = e1; e1 = e0;
    e0 = val;
  }
  void clear() {
    e0 = e1 = e2 = 0;
  }
  String top() {
    return String.
      valueOf(e0);
  }
}
```

feature: CalcBase

feature: CalcAdd

```
package util;
class Calc {
  int e0 = 0, e1 = 0,
      e2 = 0;
  void enter(int val) {
    e2 = e1; e1 = e0;
    e0 = val;
  }
  void clear() {
    e0 = e1 = e2 = 0;
  }
  String top() {
    //...
  }
  void add() {
    e0 = e1 + e0;
    e1 = e2;
  }
}
```



http://wwiti.cs.uni-magdeburg.de/iti_db/lehre/epmd/2009/slides/06_FOP.pdf

Composition Engines

Feature Orientated Command-line tools used to compose files within FeatureIDE:

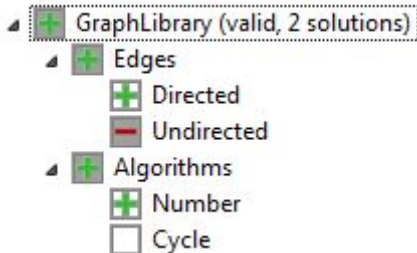
- ▶ AHEAD (mixin): .jak (Java 1.4)
<http://userweb.cs.utexas.edu/~schwartz/ATS.html>
- ▶ FeatureC++: .cpp (C++)
<http://www.fosd.de/fcpp>
- ▶ FeatureHouse: .java (Java 1.5), .cs (C#), .c/.h (C), .hs (Haskell), .jj (JavaCC), .als (Alloy), .xmi (UML)
<http://www.fosd.de/fh>

Composition Engines

Non Feature Orientated Command-line tools used to compose files within FeatureIDE:

- ▶ AspectJ(Aspectorientated Java): .aj(Aspect File), .java
<http://www.eclipse.org/aspectj/>
- ▶ DeltaJ(Deltaorientated Java): .dj(Core and Delta Modules)
<http://sourceforge.net/projects/deltaj/>
- ▶ Munge: Preprocessor
<http://sonatype.github.com/munge-maven-plugin/>
- ▶ Antenna: Propcessor
<http://antenna.sourceforge.net/>

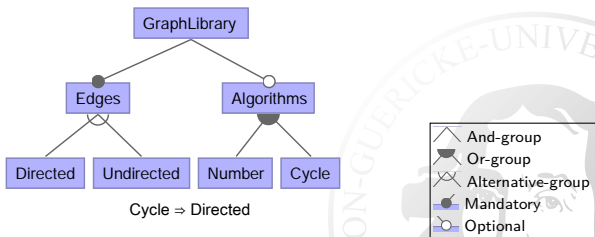
Configuration



- ▶ Selection of features
- ▶ Composition of features results in a program variant
- ▶ Not all combinations are useful

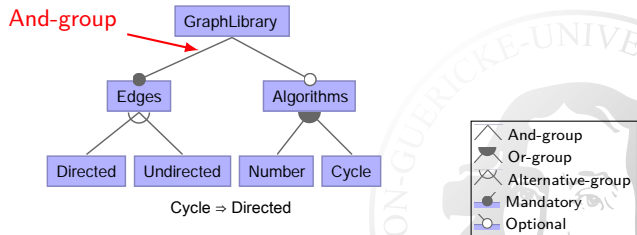
Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)



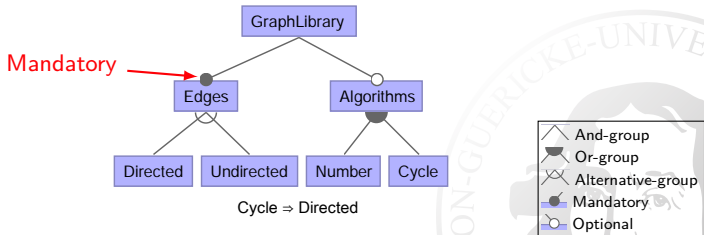
Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)



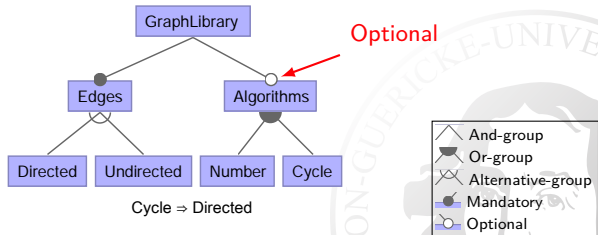
Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)



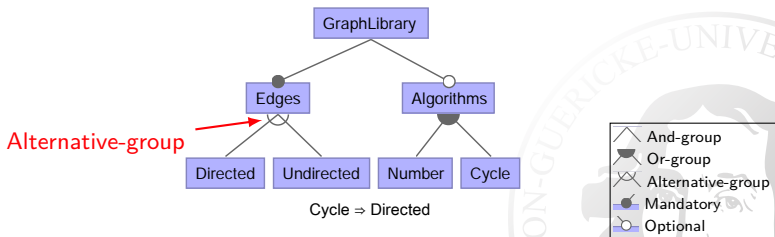
Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)



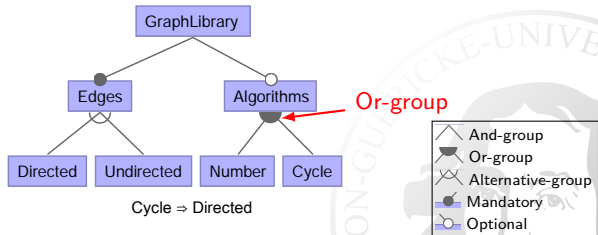
Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)



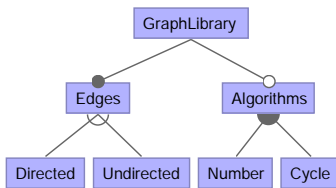
Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)

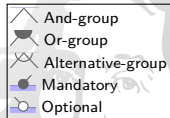


Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)

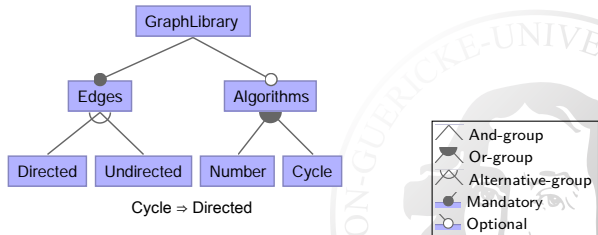


Cross-tree constraints → Cycle ⇒ Directed



Feature Model

- ▶ Specifies valid combinations of features
- ▶ Graphically represented by a feature diagram
- ▶ Created for a particular domain
- ▶ Describes a software product line (SPL)



Feature-Oriented Software Development (FOSD)

