Adaption domänenspezifischer Modellierungswerkzeuge am Beispiel bflow* Toolbox

Seminar Modellgetriebene Softwareentwicklung
Zwischenpräsentation

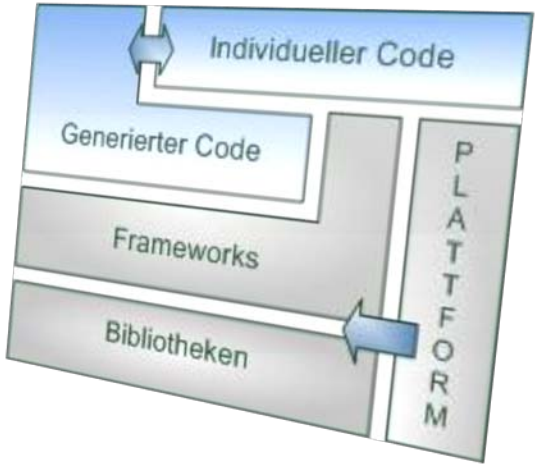
Jörg Hartmann

Gliederung

- **Grundlegendes**
  - Begriffe
  - Adaption
  - bflow* Toolbox
- **Thema**
  - Problemstellung
  - Zielsetzung und Herangehensweise
- **Arbeitsstand**
  - Techniken
  - Beispiele
  - Agenda

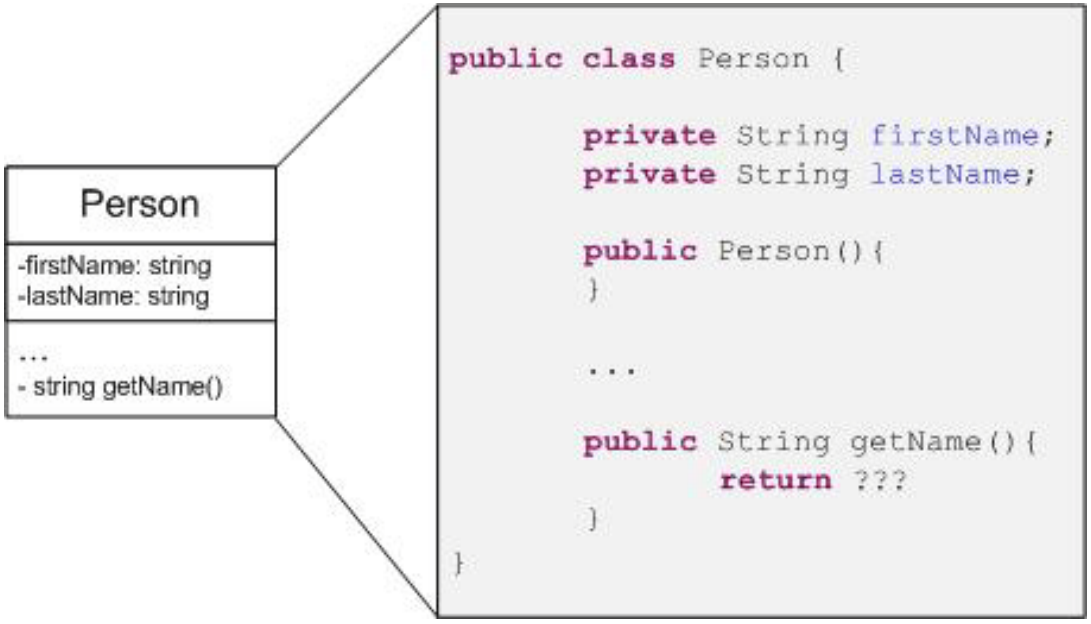
Grundbegriffe

- **DSL**
  - formale Sprache für eine Domäne
  - EPK
  - UML
- **domänenspezifisches Modellierungswerkzeug**
  - Werkzeug, zur effizienten Unterstützung der Modellierung unter einer DSL
- **Plattform**
  - Grundlage eines Werkzeuges



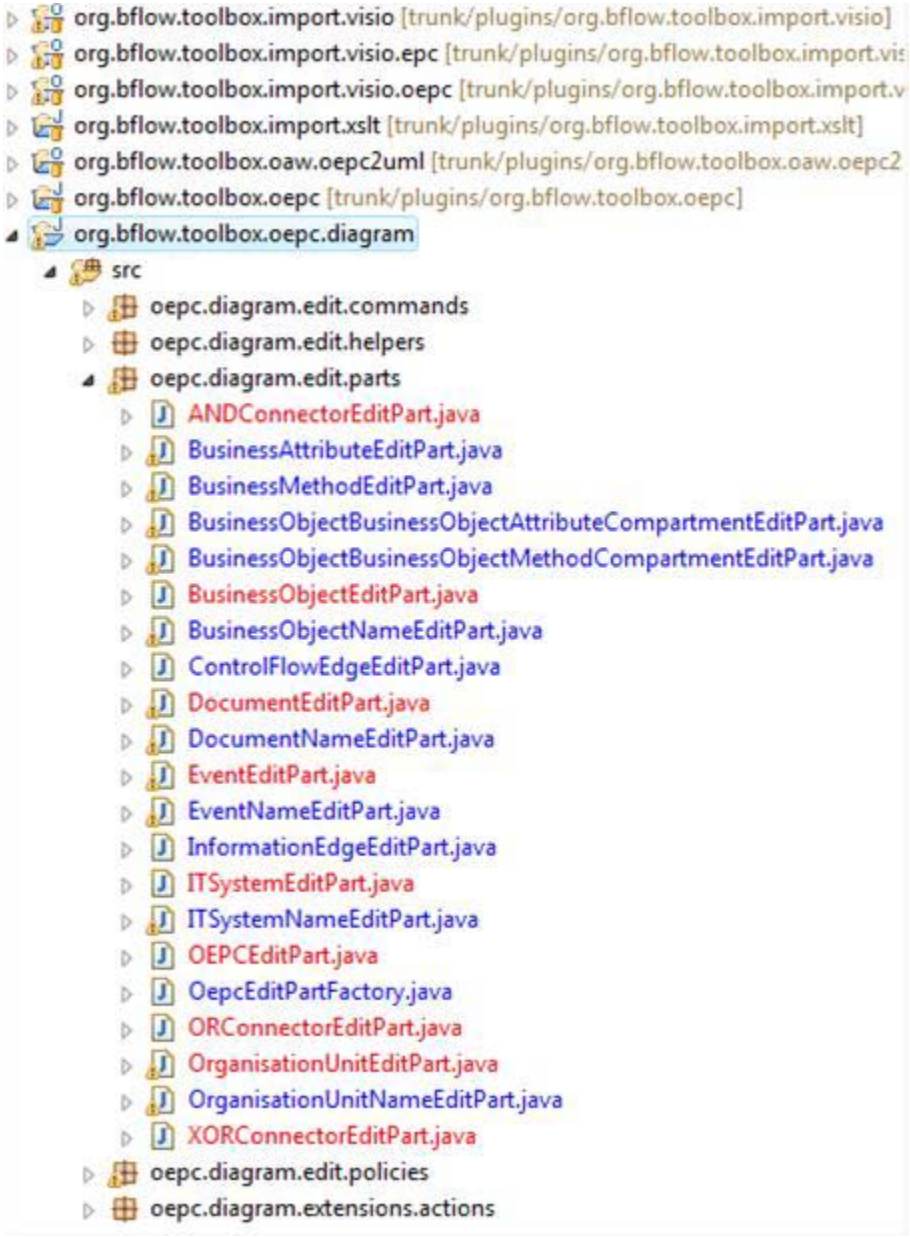
Adaption

- **Warum muss generierter Code angepasst werden?**
  - fehlende Funktionalität
  - Schwachstellen im Code
  - Adaption verursacht Probleme!



bflow* Toolbox

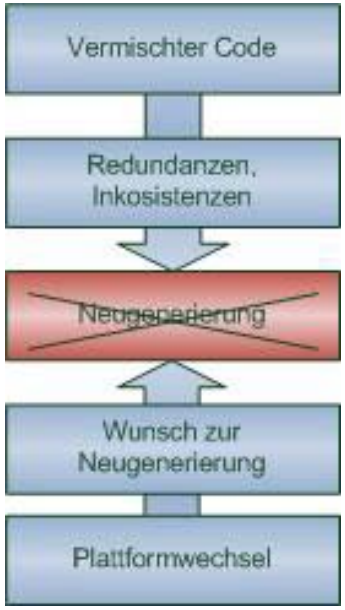
- **generiertes Werkzeug**
  - EMF 2.4.2
  - GEF 3.4.2,
  - GMF 2.1.3
- **Sprachen und Tools**
  - EPK, oEPK, VC
  - Microsoft Visio Schnittstelle
  - Aris EPK Import



Problemstellung

- **Sekundäre Probleme**
  - keine Trennung von generierten und individuellen Code
  - Code-Redundanz erhöht Wartungs-/Implementierungsaufwand
  - Code-Redundanz erhöht Fehlerwahrscheinlichkeit
  - Neugenerierung unter Umständen nicht mehr möglich

- **Primäre Probleme**
  - Plattformwechsel



Zielsetzung und Herangehensweise

- **Wie lassen sich Modellierungswerkzeuge effektiv adaptieren?**
  - Wie lässt sich generierter und individueller Code trennen?
  - Wie kann man Code-Redundanz verringern?
  - Welche Techniken gibt es?
  - Nach welchen Kriterien werden diese bewertet?
  - Wie kann der Aufwand zum Plattformwechsel gering gehalten werden?
- **Herangehensweise**
  - individuellen Code in bflow* identifizieren
  - Plattformwechsel – Techniken anwenden
  - bewerten

Arbeitsstand

- **Adaptionstechniken**
  - am Generator, Templates
  - im Code (Protected Regions, …)
  - Code Separation (OO, Vererbung)
  - Aspektorientierte Programmierung (Object Teams)
  - Aspektorientierte Templates

  - verwandtes Thema: Softwareadaption – Code-Transformationssprachen (inject/j)

Beispiele

- **Adaption am Generator**
  - Neugenerierung jederzeit möglich
  - weiterer Vorteil: im Vergleich zur Code-Anpassung müssen über 20 Klassen nicht geändert werden!
  - neue Elemente können einfach hinzugefügt werden
  - Nachteil?

```
DEFINE getManager(diagram : gmfgen::GenDiagram) FOR gmfgen::GenCommonBase
      EXPAND xpt::Common::generatedMemberComment¬
            protected org.eclipse.gef.tools.DirectEditManager getManager(){
              if (manager == null) {
                setManager(new TextDirectEditManager(this, WrapTextCellEditor.class,
                          ¤diagram.getEditPartFactoryQualifiedClassName()¬.
                             getTextCellEditorLocator(this)));
              }
              return manager;
            }
¤ENDDEFINE¬
```

Beispiele

- **Trennung mit AOP**
  - Object Teams
  - verwendet Rollen, die von Klassen gespielt werden
  - flexibel, Rolle muss nur hinzugefügt werden
  - komplette Trennung von individuellen und generierten Code

```
public class OepcViewFactoryRole playedBy OepcViewProvider{

            callin Node createNode(IAdaptable semanticAdapter, View containerView,
                        String semanticHint, int index, boolean persisted,
                        PreferencesHint preferencesHint){
                  Node node = base.createNode(semanticAdapter, containerView,
                              semanticHint, index, persisted, preferencesHint);

                  node.getStyles().removeAll(node.getStyles());
                  new StoreColorsFromSchemaCommand(node,
                              TransactionContext.Write).execute();
                  new StoreTextAlignmentCommand(node, TextAlignment.CENTER_LITERAL,
                              TransactionContext.Write).execute();

                  return node;
            }
            createNode <- replace createNode;
}
```

Agenda

- Literaturarbeit 
- Techniken 
- Implementierung 
- Bewertungskriterien 
- Schreiben…