An Overview of Generative Software Development
Krzysztof Czarnecki
University of Waterloo
Overview

- Introduction
- Domain vs Application Engineering
- Problem vs Solution Space
- Feature Modelling
- Model Driven Architecture vs Generative Software Development
- Conclusion
Introduction

- **System Family Engineering**
  - Focuses of creating reusable assets

- **Generative Software Development**
  - Automatic generation of system family component

- **Model Driven Development**
  - All aspects of software are described using models.
  - Models can be directly compiled into executable code
Domain vs Application Engineering

Domain engineering
- Domain analysis
- Domain design
- Domain implementation

Application engineering
- System requirements
- System derivation
- System tailoring

Reusable assets
New requirements

Management
Problem vs Solution Space

**Problem space**
- domain-specific abstractions

**Solution space**
- implementation-oriented abstractions

**Mapping**
Problem vs Solution Space

- Two views Configuration and Transformation

- Configuration
  - Problem Space: Domain specific features, default values, illegal combinations
  - Solution Space: Implementation components

- Transformation
  - Problem Space: Domain Specific Language
  - Solution Space: Implementation Language
Feature Modeling

- Method for describing the common and the varying components of a system family
  - Concrete
    - Individual components
    - e.g. data storage
  - Aspectual
    - Affect groups of components
    - e.g. logging, synchronization
  - Abstract
    - Configuration of components/aspects
  - Grouping
Feature Modeling

- e-shop
  - payment
  - taxCalculation
  - shipping
    - creditCard
    - addressResolution
      - debitCard
      - country
      - streetAddress
        - electronicCheque
        - postalCode
Feature Oriented Approach
Model Driven Architecture (MDA)

- Framework for model driven development
- Developed by the Object Management Group (OMG)
- Allows platform independent descriptions of software systems
- Platform Independent Models (PIM) are transformed into Platform Specific Models (PSM) and then into code
Transformation from PIM to PSM can be seen as a transformation from the problem to solution space.

Focus on variation in technology:
- Specifically platform independence.
Conclusion

• Generative Software Development
  - Automatic generation of reusable system components
  - Domain specific languages
  - Feature modelling for domain scoping, DSL creation and system architecture
Discussion

• Does generative software development lead to an improvement in developer efficiency

• Are feature models a good way to describe system families