fUML Refactoring with EMF

Business Informatic Group

Kristof Meixner
Sebastian Geiger

6 Mai 2014

Overview

- Refactoring Overview
- UML Models and Refactoring
- fUML Introduction
- Complex Example
- fUML Refactoring
- Semantic Preservation
- Refactoring Constraints with OCL
- Toolchain
- EMF Refactor

Refactoring Overview

- What is refactoring?
  - “defines a set of program restructuring operations” that “preserve the behavior of a program” [2]
- Why do we need it?
  - Increases software (or model) quality.
- There are catalogues with refactorings [1]
- Examples: rename class, extract superclass, encapsulate field.

UML Models and Refactoring

- Whats the difference between source code and model refactoring?
- UML defines different interconnected views:
  - Class diagrams
  - Activity diagrams
  - Sequence diagrams
- Refactoring needs to consider all views.
- Refactoring needs to preserve semantics (behavior).

fUML Introduction

- fUML = foundational UML
- fUML 1.1 is based on UML 2.4.1
- Subset of UML (Class and Activity diagrams)
- Enhanced with consise semantics
- Turing complete and allows execution or interpretation
- Existing VM to execute models
- Extended VM for testing and debugging (Moliz)

Complex Example

An insurance company example which sells insurance policies. A policy can insure cars and truck and calculate its price.

fUML Refactoring

Semantic Preservation

Refactoring Constraints with OCL

Toolchain

EMF Refactor

Questions?

References

- Fowler, M.
  Refactoring - Improving the Design of Existing Code.
  AddisonWesley, July 1999.

- Opdyke, W. F.
  Refactoring object-oriented frameworks.
  Master's thesis, University of Illinois, 1992.