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" [4]
- Why do we need it?
  - Increases software and/or model quality
  - Ensures reusability of components
  - Supports change management in software lifecycle
- Examples: rename class, extract superclass, encapsulate field.
- Detailed catalogues with refactorings exist (e.g. [1])

Recall - UML

- Unified Modeling Language (v2.4.1) standardized by Object Management Group [3]
- General-purpose modeling language in the field of software engineering (Wikipedia)
- Includes different diagram types for architecture structure & behavior

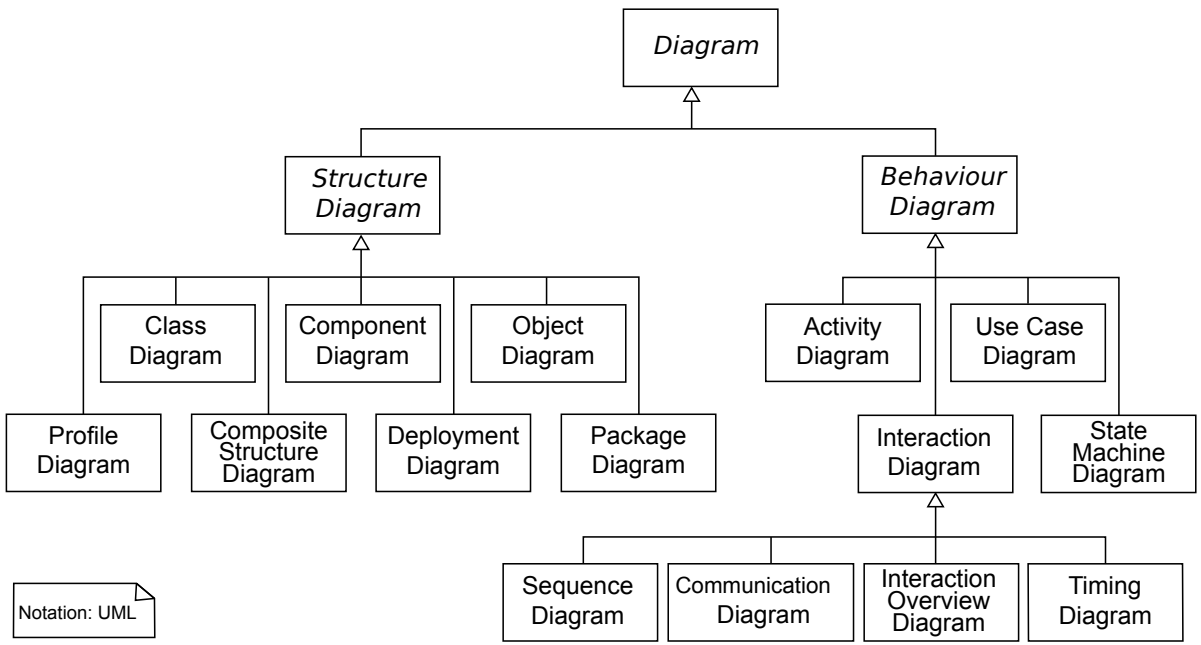

Figure : *UML* diagram type hierarchy (Derfel73, PMerson)

UML Models and Refactoring

- Whats the difference between source code and model refactoring?
  - Consider all interconnected views/diagrams
  - Consider model constraints
  - Consider different accuracy levels
- How can we preserve behavior/semantics?
  - Static analysis of models (e.g. "code smells" like complexity or dependencies)
  - Dynamic analysis of models (e.g. via behavior during execution)

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) [2]

Complex Example

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

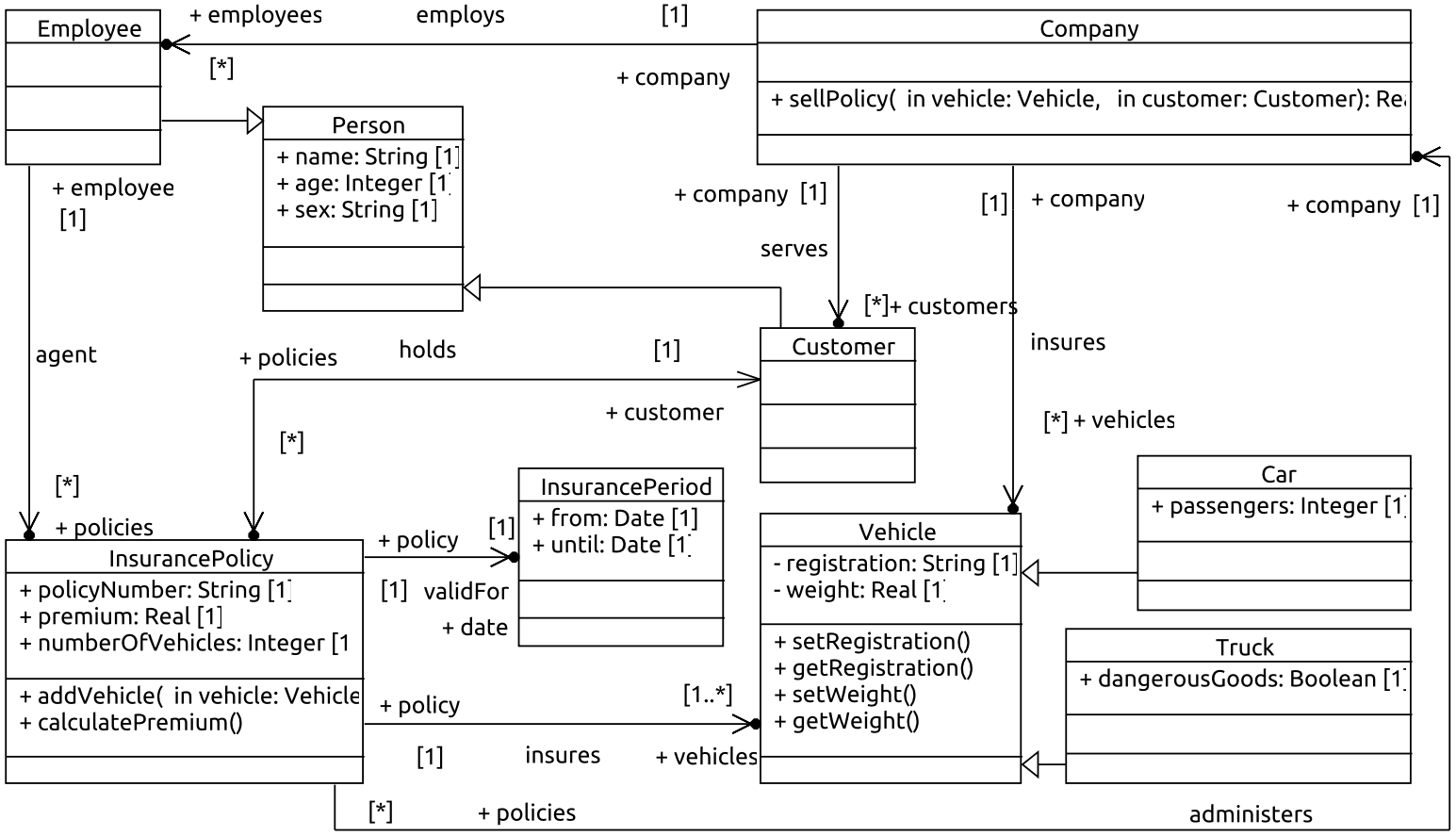

Figure : Insurance class diagram with refactorings

fUML Refactoring

- Show example of encapsulate field.

Semantic Preservation

- What means semantic preservation?
  - Same execution trace?
  - Same output?
  - Same state?
- Depends on refactoring!
- How to preserve semantics?
  - Specify pre- and postconditions with OCL constraints
  - Validate refactored models.
  - Execute models and analyse execution properties (trace).

Refactoring Constraints with OCL

Toolchain

EMF Refactor

Questions?

References

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

Mayerhofer, T., Langer, P., and Kappel, G.
A runtime model for fuml.
In *Models@run.time* (2012), pp. 53–58.

OMG.
OMG Unified Modeling Language, 2.4.1 ed.
OMG, http://www.omg.org/spec/UML/2.4.1/, 05 2011.

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