(Automated) Refactoring - A Taxonomy

Seminar in Software Engineering

Peter Feichtinger
Lisa Kritzinger

Institute for Software Systems Engineering
Johannes Kepler University Linz

25. April 2017

Refactoring

- Restructuring code without change in semantics
  - Importance in software evolution is obvious
  - Tool support is important
- Topics:
  - Performance impact of refactoring
  - Modernizing code
  - Automated refactoring

Refactoring

- Restructuring code without change in semantics
  - Importance in software evolution is obvious
  - Tool support is important
- Topics:
  - Performance impact of refactoring
  - Modernizing code
  - Automated refactoring

Maintainability versus Performance: What's the Effect of Introducing Polymorphism?

Serge Demeyer (2002)

- Case Study Paper
- Comparison of the performance of two programms
  - one which contains large conditionals
  - one where the conditionals are implemented using polymorphism

Co-evolution of Object-Oriented Software Design and Implementation

Theo D'Hondt, Kris De Volder, Kim Mens, Roel Wuyts (2002)

- Experiments that use logic meta-programming (LMP)
  - LMP = an instance of hybrid language symbiosis merging a declarative (logic) meta-level language with a standard object-oriented base language
- Codify design information as
  - constraints
  - a process for code generation

Co-evolution of Object-Oriented Software Design and Implementation

Theo D'Hondt, Kris De Volder, Kim Mens, Roel Wuyts (2002)

- Experiments that use logic meta-programming (LMP)
  - LMP = an instance of hybrid language symbiosis merging a declarative (logic) meta-level language with a standard object-oriented base language
- Codify design information as
  - constraints
  - a process for code generation

Automated Refactoring using Design Differencing

Iman Moghadam, Mel Ó Cinnéide (2012)

- Novel refactoring approach that refactors a program based on
  - desired design
  - source code
- Using desired design as target, based on
  - current software design and
  - understanding of how it may be required to evolve

Automated Refactoring using Design Differencing

Iman Moghadam, Mel Ó Cinnéide (2012)

- Novel refactoring approach that refactors a program based on
  - desired design
  - source code
- Using desired design as target, based on
  - current software design and
  - understanding of how it may be required to evolve

Restructuring Legacy C Code into C++

Richard Fanta, Václav Rajlich (1999)

- Case study on Mosaic browser code
- Combination of refactorings to create classes
  - From C `structs`
  - From related variables

Restructuring Legacy C Code into C++

Richard Fanta, Václav Rajlich (1999)

- Case study on Mosaic browser code
- Combination of refactorings to create classes
  - From C `structs`
  - From related variables

The Spartanizer: Massive Automatic Refactoring

Yossi Gil, Matteo Orrù (2017)

- Tool demo paper
- Eclipse plugin for automatic refactoring to make code more compact
- Shows that automatic refactoring can be used effectively

The Spartanizer: Massive Automatic Refactoring

Yossi Gil, Matteo Orrù (2017)

- Tool demo paper
- Eclipse plugin for automatic refactoring to make code more compact
- Shows that automatic refactoring can be used effectively

Slicing Object-Oriented Software

Loren Larsen, Mary Jean Harrold (1996)

Mostly unrelated to our topic $\rightarrow$ canned

Papers

Demeyer

Maintainability versus Performance: What's the Effect of Introducing Polymorphism?

Technical Report, Lab. on Reengineering, Universiteit Antwerpe, 2002

D'Hondt, De Volder, Mens, Wuyts

Co-evolution of Object-Oriented Software Design and Implementation

Software Architectures and Component Technology, 2002

Moghadam, Ó Cinnéide

Automated Refactoring using Design Differencing

16th European Conference on Software Maintenance and Reengineering (CSMR), 2012

Papers

Fanta, Rajlich

Restructuring Legacy C Code into C++

IEEE International Conference on Software Maintenance (ICSM), 1999

Gil, Orrù

The Spartanizer: Massive Automatic Refactoring

24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2017

Larsen, Harrold

Slicing Object-Oriented Software

18th International Conference on Software Engineering (ICSE), 1996

ISSE