Andreas Wierda, Eric Dortmans, Lou Somers

Architecture Reconstruction of Industrial Object-Oriented Software: A Case Study
Contents

- Context
- Approach
- Case study: Architectural clustering
- Conclusions
Context

Architecture Reconstruction
- Reconstruct architectural view from source code
- Which methods?
- Useful for printer controllers?
- Result quality?
- Improvements?

Context: Océ printer controller
- Océ: printer & copier manufacturer
- Océ multifunctionals: printer, scanner & copier
- Object-oriented (mostly C++)
- 932 KLOC in 2661 classes
Approach

- Architecture Reconstruction methods
  - Manual methods: supported with tools
  - Automatic methods (for object-oriented software):
    - Pattern detection with a pattern library
    - Pattern detection without a pattern library
    - Architectural clustering: different kinds of information used

- Case study
  - Architectural clustering
Clustering: Introduction

- Group closely related elements
- Small example:
Case Study: Architectural Clustering

- Group closely related classes
  - Based on relations between them.

- Literature study led to two hypotheses:
  - Clustering can reconstruct “good” architectural views.
  - Using information from old versions gives a better result.

- Prototype to confirm hypotheses
Clustering: Architecture overview

- Source code
- Fact extraction
- Class graph
- Decomposition
- Visualisation
- Architectural view
- Assessment
- Similarity metric
- Expert decomposition
- Source-tree based clustering
- Decomposition editing
- Expert knowledge
- Class information
- Expert knowledge

- Class graph
- Decomposition
Clustering: Architecture Reconstruction

- Source code $\emptyset$ class graph
- Class graph $\emptyset$ cluster graph
  - Classes $\emptyset$ nodes
  - Class-relations $\emptyset$ edges
  - Edge “thickness”
  - Type weight
  - Redundant edges for dependencies
Case Study: Parameters

- Weight of relations:
  - Association ($p_{wa}$): Integer
  - Generalisation ($p_{wg}$): Integer
  - Dependency ($p_{wd}$): Integer

- Reduction of information:
  - Take instance count or just presence of relations ($p_c$): Boolean
  - Omit redundant dependencies ($p_i$): Boolean
Clustering: Using version information (1)

Versions: $V_8$  $V_1$  ...  $V_8$

Models: $M_8$  $M_1$  $M_{1,8}$

Architecture: $A_8$  $A_8$
Clustering: Using version information (2)

Reconstructed version

Other version

Class-relations-intersection
Clustering: Experiment Approach

- Find parameters
  - For weights and combine/ignore behaviour
  - Try different combinations Ø small program
  - Experiment with ten architects
  - Used Grizzly & RIP Worker subsystems

- Result
  - Different best combinations for Grizzly and RIP Worker
  - Use set of parameter-combinations
Clustering: Results

- Applied twice to printer controller (version 7e and 8a)
- Resulting MoJoQuality
  - Version 7e: 61%
  - Version 8a: 63%
- Multiple versions:
  - Class-relations-intersection
    - With first version: better MoJoQuality (74% and 78%)
    - With previous version: no improvement
  - Class-relations-union: no improvement
Clustering: Conclusions

- Clustering can reconstruct useful architectural views...
  - ...but manual refinement of result needed
- Time consuming but practical
- Information from old versions helps to get better results

Hypotheses:
- ✓ Clustering can reconstruct “good” architectural views.
- ✓ Using information from old versions gives a better result.