Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Simon Baars
University of Amsterdam
Amsterdam, the Netherlands
simon.mailadres@gmail.com

2nd Ana Oprescu
University of Amsterdam
Amsterdam, the Netherlands
ana.oprescu@uva.nl

***Abstract*—Coming up!**

***Index Terms*—code clones, refactoring, static code analysis, object-oriented programming**

I. Introduction

This document is a model and instructions for LaTeX. Please observe the conference page limits.

II. Background

A. Code clone terminology

Clone class collection, clone class, clone instance, cloned node, token.

B. Refactoring techniques

Extract method, move method.

III. Defining refactorable clones

Instead of defining clone types, I just define how we can ensure that clones can be refactored. I do not plan to differentiate between T1R, T2R and T3R here, because I think it makes the paper confusing.

To be honest, I'm not even sure how relevant it is to name the literature clone types. Maybe not focus on their shortcomings, but just start from the core (sourcecode) and define how we can get to refactorable clones.

A. Ensuring Equality

Explain type 1, why it is not always refactorable, solution

B. Allowing variability in a controlled set of expressions

Explain type 2, why it is not always refactorable, solution

C. Gapped clones

Explain type 3, why it is not always refactorable, solution

IV. CloneRefactor

The tool: Detect Clones -¿ Map Context -¿ Refactor.

A. Clone Detection

Detecting refactorable clones, (clone graph?? do I want to explain my exact methods?)

Identify applicable funding agency here. If none, delete this.

B. Context Mapping

1) Relation:

2) Location:

3) Contents:

C. Refactoring

1) Extract Method: Show my categories to show what clones can be dealt with by method extraction.

2) AST Transformation: Explain what AST transformations I do to apply the refactorings.

3) The cyclic nature of refactoring: Explain how refactoring code clones might open up new refactoring opportunities.

V. Results

A. Clone context

How many clones are there in certain contexts?

B. Clone refactorability

To what extent can found clones be refactored through method extraction, without requiring additional transformations.

C. Thresholds

I think the ultimate goal with this thesis is to experiment the refactored source code

VI. Discussion

VII. Conclusion

Acknowledgment

:-)

References