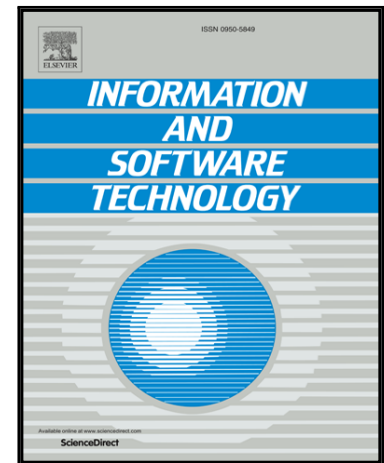


Accepted Manuscript

Performance-driven Software Model Refactoring

Davide Arcelli, Vittorio Cortellessa, Daniele Di Pompeo

PII: S0950-5849(17)30178-7
DOI: [10.1016/j.infsof.2017.09.006](https://doi.org/10.1016/j.infsof.2017.09.006)
Reference: INFSO 5879



To appear in: *Information and Software Technology*

Received date: 1 March 2017
Revised date: 6 September 2017
Accepted date: 14 September 2017

Please cite this article as: Davide Arcelli, Vittorio Cortellessa, Daniele Di Pompeo, Performance-driven Software Model Refactoring, *Information and Software Technology* (2017), doi: [10.1016/j.infsof.2017.09.006](https://doi.org/10.1016/j.infsof.2017.09.006)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Performance-driven Software Model Refactoring

Davide Arcelli^a, Vittorio Cortellessa^a, Daniele Di Pompeo^a

^a *University of L'Aquila, L'Aquila, Italy*

Abstract

Context: Software refactoring is a common practice aimed at addressing requirements or fixing bugs during the software development. While refactoring related to functional requirements has been widely studied in the last few years, non-functional-driven refactoring is still critical, mostly because non-functional characteristics of software are hard to assess and appropriate refactoring actions can be difficult to identify. In the context of performance, which is the focus of this paper, antipatterns represent effective instruments to tackle this issue, because they document common mistakes leading to performance problems as well as their solutions.

Objective: In order to effectively reuse the knowledge beyond performance antipatterns, automation is required to detect and remove them. In this paper we introduce a framework that enables, in a unique tool context, the refactoring of software models driven by performance antipattern detection and removal.

Method: We have implemented, within the EPSILON platform, detection rules and refactoring actions on UML models for a set of well-known performance antipatterns. By exploiting the EPSILON languages to check properties and apply refactoring on models, we enable three types of refactoring sessions.

Results: We experiment our framework on a Botanical Garden Management System to show, on one side, that antipatterns can effectively drive software refactoring towards models that satisfy performance requirements and, on the other side, that the automation introduced by EPSILON-based sessions enables to inspect multiple paths and to propose a variety of solu-

Email addresses: davide.arcelli@univaq.it (Davide Arcelli),
vittorio.cortellessa@univaq.it (Vittorio Cortellessa),
daniele.dipompeo@graduate.univaq.it (Daniele Di Pompeo)

Download English Version:

<https://daneshyari.com/en/article/6948141>

Download Persian Version:

<https://daneshyari.com/article/6948141>

[Daneshyari.com](https://daneshyari.com)