

On the use of developers' context for automatic refactoring of software anti-patterns

Rodrigo Morales¹, Z ephyrin Soh¹, Foutse Khomh¹, Giuliano Antoniol² y Francisco Chicano³

¹ SWAT Lab., Polytechnique Montr al, Canada
rodrigomoraless2@acm.org, {zephyrin.soh, foutse.khomh}@polymtl.ca

² Soccer Lab., Polytechnique Montr al, Canada
giuliano.antonio@polymtl.ca

³ Universidad de M laga
chicano@lcc.uma.es

Abstract: Anti-patterns are poor solutions to design problems that make software systems hard to understand and extend. Entities involved in anti-patterns are reported to be consistently related to high change and fault rates. Refactorings, which are behavior preserving changes are often performed to remove anti-patterns from software systems. Developers are advised to interleave refactoring activities with their regular coding tasks to remove anti-patterns, and consequently improve software design quality. However, because the number of anti-patterns in a software system can be very large, and their interactions can require a solution in a set of conflicting objectives, the process of manual refactoring can be overwhelming. To automate this process, previous works have modeled anti-patterns refactoring as a batch process where a program provides a solution for the total number of classes in a system, and the developer has to examine a long list of refactorings, which is not feasible in most situations. Moreover, these proposed solutions often require that developers modify classes on which they never worked before (i.e., classes on which they have little or no knowledge). To improve on these limitations, this paper proposes an automated refactoring approach, ReCon (Refactoring approach based on task Context), that leverages information about a developer's task (i.e., the list of code entities relevant to the developer's task) and metaheuristics techniques to compute the best sequence of refactorings that affects only entities in the developer's context. We mine 1705 task contexts (collected using the Eclipse plug-in Mylyn) and 1013 code snapshots from three open-source software projects (Mylyn, PDE, Eclipse Platform) to assess the performance of our proposed approach. Results show that ReCon can remove more than 50% of anti-patterns in a software system, using fewer resources than the traditional approaches from the literature.

Keywords: Software maintenance, Automatic refactoring, Task context, Interaction traces, Anti-patterns Metaheuristics

Reference: Morales, R., Soh, Z., Khomh, F., Antoniol, G., Chicano, F.: On the use of developers' context for automatic refactoring of software anti-patterns. *Journal of Systems and Software* 128, 236 – 251 (2017), <http://dx.doi.org/10.1016/j.jss.2016.05.042>