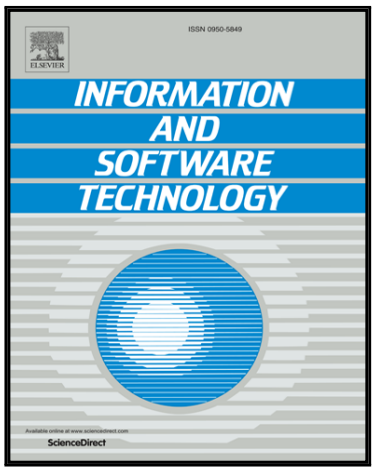

Improving Modular Structure of Software System using Structural and Lexical Dependency

Amarjeeta Prajapati , Jitender Kumar Chhabrab

PII: S0950-5849(16)30195-1
DOI: 10.1016/j.infsof.2016.09.011
Reference: INFSOF 5769

To appear in: *Information and Software Technology*

Received date: 19 June 2015
Revised date: 26 September 2016
Accepted date: 30 September 2016

Please cite this article as: Amarjeeta Prajapati , Jitender Kumar Chhabrab , Improving Modular Structure of Software System using Structural and Lexical Dependency, *Information and Software Technology* (2016), doi: 10.1016/j.infsof.2016.09.011

Improving Modular Structure of Software System using Structural and Lexical Dependency

Amarjeeta, Prajapati, Jitender Kumar Chhabrab

a,bDepartment of Computer Engineering,National Institute of Technology, Kurukshetra-136119, Haryana INDIA

Email:
aamarjeetnitkkr@gmail.com<mailto:amarjeetnitkkr@gmail.com>,
bjitenderchhabra@gmail.com<mailto:jitenderchhabra@gmail.com>

Corresponding author: Amarjeet,

Abstract

Context: A software system's structure often degrades due to repetitive maintenance. To make a sustainable evolution of such systems, it becomes mandatory to improve their modular structure after a certain time. Many remodularization approaches were proposed to improve the modular structure of software systems. Most of the existing approaches rely on structural or lexical dependencies. However, there is a lack of research that distinguishes different types of structural (e.g., inheritance, method calls, references, etc.) or lexical (Name of classes, methods, variables, etc.) dependencies, but assumes that they are equivalent, which is illogical from a software developer's point of view.

Objective: In this paper, we propose an approach that considers various types of structural as well as lexical dependencies along with their relative importance to remodularize the Object-Oriented (OO) systems. The main goal of the paper is to generate remodularization solutions that can reflect the developers' perspective (as visible in the well-modularized software system) of remodularization, which is highly desirable in software evolution.

Method: The paper computes coupling strength among classes using different weights (computed on basis of well-modularized software system) in terms of various mechanisms of structural and lexical dependencies. Software remodularization problem is formulated as a single and multi-objective optimization problem and solved using Genetic Algorithms (GA). Based on the different types of structural and lexical dependencies and as per their un-weighted/weighted variants, we have designed following 24 coupling schemes: structural-based (i.e., SBUW, SBW, SAUW, SAW, STFUW, STFW, STFIDFUW, and STFIDFW), lexical-based (i.e., LBUW, LBW, LAUW, LAW, LTFUW, LTFW, LTFIDFUW, and LTFIDFW), and combined structural-lexical based (i.e., SLBUW, SLBW, SLAUW, SLAW, SLTFUW, SLTFW, SLTFIDFUW, and SLTFIDFW). Values obtained through these coupling schemes are used in coupling and cohesion objective function of the GA. Along with this objective, some supportive objective functions such as MCI and MSI have been used to drive the optimization process towards a good quality modularization solution.