Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 125 (2018) 740–746

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India

Improving Cohesion of a Software System by Performing Usage Pattern Based Clustering

Amit Rathee[a*], Jitender Kumar Chhabra[b]

[a,b]*Department of Computer Engineering, National Institute of Technology, Kurukshetra- 136119*

Abstract

Increasing the software design quality is a key research challenge in object-oriented software development system. Cohesion is one of the key spect that helps to evaluate the quality and modularity of a software system at the design level. It helps to create software components that are directly reusable to the industry because of their less dependence on other components. In this paper, a new cohesion metric for object-oriented software, named as Usage Pattern Based Cohesion (UPBC), is proposed which is computed at the module level. This paper considers class as a module initially and subsequently group of classes (i.e. a package) is considered as a module with an aim of improving overall cohesion. This metric utilizes the Frequent Usage Patterns (FUP) extracted from different member functions interactions to capture the cohesiveness of the module. Further, the measured cohesion value is used to perform clustering of modules in order to increase cohesion and decrease coupling among modules simultaneously. The clustering is performed by using a newly proposed clustering algorithm called FUPClust (Frequent Usage Pattern based Clustering) based on FUP interactions among modules. The proposed approach is applied to two Java software systems and the results obtained show a significant improvement in the cohesiveness of the software system.

Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications.

Keywords: : Software Engineering; Clustering; Frequent Usage Patterns (FUP); Cohesion.

1. Introduction

With increasing growth of software product use in industry and our day to day life [14], the software development process has gained popularity among researchers and other practitioners [16] [20]. Since software development is a

* Corresponding author. Tel.: +91-999-177-1143
E-mail address: [a]amit1983_rathee@rediffmail.com, [b]jitenderchhabra@gmail.com

1877-0509
Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications
10.1016/j.procs.2017.12.095

human-centric activity, so, it is prone to undesirable performance and design defects [10]. So, software development process needs to be continuously assessed and evolved over time in order to fulfill customer's requirements and remove other identified defects [10]. This helps in improving the software design and hence the quality of a software system.

Cohesion and Coupling being the two important metrics that denotes the quality at structural design level of a software system [15]. The term cohesion is originated from structural design [21] and it refers to how much the various elements of a given modules are related to each other. It is an important indicator of software design quality and the modularity. A higher cohesion value of a module indicates that the given module is providing near single functionality, whereas, a lower value hinders the reuse of a software module. So, a module with higher cohesion is always desirable [7]. Numerous cohesion metrics have been proposed already [3] [5] [6] [9] [11] [17]. These proposed metrics are based on measuring the method to method interaction and member variable references made by them. These metrics do not consider member variable references to outside modules and member variable references made due to nested member function calls, which in our idea is a research gap in accurately measuring cohesion of a module.

In this paper, an approach to measure cohesion at module level is proposed. The proposed metric, Usage Pattern Based Cohesion (UPBC), measures the usage pattern of member variables among different member functions of a module. Later, based on the measured cohesion metric value, different modules are clustered by using the proposed clustering algorithm called FUP based Clustering (FUPClust). The research contributions of this paper include:

1. To propose a cohesion metric UPBC, that measures the cohesion at module level by utilizing the FUP's identified for the given module. The FUP measures the extent to which a given module references the member variables inside and outside to it. The FUP is calculated at method level and overall FUP for the module is calculated based on FUP calculated for the member functions defined inside the module.
2. To propose a clustering algorithm FUPClust that regroup the modules by doing clustering based on the measured cohesion values of different modules.

The rest of this paper is structured as follows: Section 2 gives description regarding the literature survey, section 3 gives detailed description of the proposed. Section 4 describes the experimentation and results and finally section 5 describes the concluding remarks and future works.

2. Literature Survey

As the popularity of object-oriented software development is increasing, there is a greater need for software design metrics which are capable of measuring the software design quality. Cohesion is one such key design principle in software engineering and in this direction, numerous cohesion metrics have been already proposed [3] [5] [6] [9] [11] [17]. Yourdon et al. define the coupling for an object-oriented software as the degree to which different modules are interdependent on each other [22]. Briand et al. [7] propose a structural based unified framework to measure cohesion in an object-oriented software system and proposed a cohesion metric *Coh* that counts attribute references and sharing among the methods of a class. Bansiya [2] defines cohesion in terms of coupling by proposing a coupling metric Direct Class Coupling (DCC) which counts the total number of classes that are directly related to a given class. Chidamber et al. [8] propose a metric suite that also measures cohesion as LCOM (Lack of Cohesion among Methods) metric which measures the sharing of member variables among different pairs of methods of a class. Li and Henry [18] proposes a cohesion metric LCOM3 by extending the work of [8] and representing the system as an undirected graph. They represented each class method as a node in the graph and member variables sharing as an edge in the graph. They measured class cohesion as the total number of strongly connected components in MDG (Module Dependency Graph). Hitz and Montazeri [13] proposes another cohesion metric LCOM4 by representing the system as a graph in which the nodes represents the methods and edge between any vertices denote that they are accessing the same attribute. Henderson et al. [12] give the latest proposed metric LCOM5 in LCOM metric series. This metric gives cohesion value of zero (0) if methods use only member variables of the class and it gives a value of one (1) if every method uses only one member variable of the class. Bieman and Kang's [4] also proposes two sets of cohesion metrics known as tight class cohesion (TCC) and loose class cohesion (LCC). They calculated TCC as the ratio of a total number of pairs of member functions with no sharing of member variables to a total number of pair of direct member functions which share at least one member variable among