INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume2, Issue5)
Available online at www.ijarnd.com

A Survey on Software Component Restructuring

Shreya Jaiswal, Ms. Satya Verma
Computer Science & Engineering
MATS University
shrey8.jaiswal@gmail.com, *satyaverma@matsuniversity.ac.in*

ABSTRACT

Component-based software development is the reuse of the existing software. By reusing the component have cheaper cost, better quality, and improved performance. The reusable component performs better than existing software. It provides a standardized system. But finding components for efficient software reuse is one of the important problems identified by researchers as well as artifacts which constitute a software system. Many types of research have also been done in improving cohesion and reducing coupling for source code and to recover software architecture in reverse engineering process, a grouping of the component is done. The objective of our paper is to improve the design and to examine numerical clustering techniques and implement these techniques for different software applications. Different clustering techniques for component reuse and program restructuring are hierarchical clustering, density based clustering, SOM.

Keywords: Software Component Clustering, Density-Based Clustering, Hierarchical Clustering.

INTRODUCTION

Component-based software development has become very important in the field of software engineering. Restructuring of the program is a most important method for maintaining and improving the quality of ill-structured programs. Software component restructuring reduces the software development cost as due to change in requirements and technologies of software huge amount is spent on maintenance [1]. Along with this original structure of program drifts and quality degrades.

To overcome this problem [2] program restructuring or refactoring is used for better under stability and prevention from degrading of program structure [3]. By using program restructuring functions the user or client will get a desirable structure of program within less time with the best quality and less cost and the required restructured program should achieve high cohesion and low coupling [4, 5, and 6].

Cohesion indicates the relationship within the module. In software designing and restructuring, cohesion is most important part/unit. Cohesion is usually described as “high cohesion” and is often contrasted with loose/low coupling. To increase system maintainability cohesion is used.

The clustering techniques empower cohesion by grouping similar elements together and removing the dissimilar ones. Many articles of software clustering have found out that clustering technique increases potential in software restructuring field and also concluded that clustering method is the best way of demodulating the software [7].

There are many types of research on software clustering field which have been working on architecture level but have not been focusing on the source code. So here in this paper, presents the approach of program restructuring which will work and focus on source code maintainability and reusability of source code. This paper deals with restructuring at each level and to perform this, the paper uses clustering techniques which are best suited for poorly structured functions.

Clustering will make the program easy to understand and very effective in practice. Along with clustering cohesion will also be used.

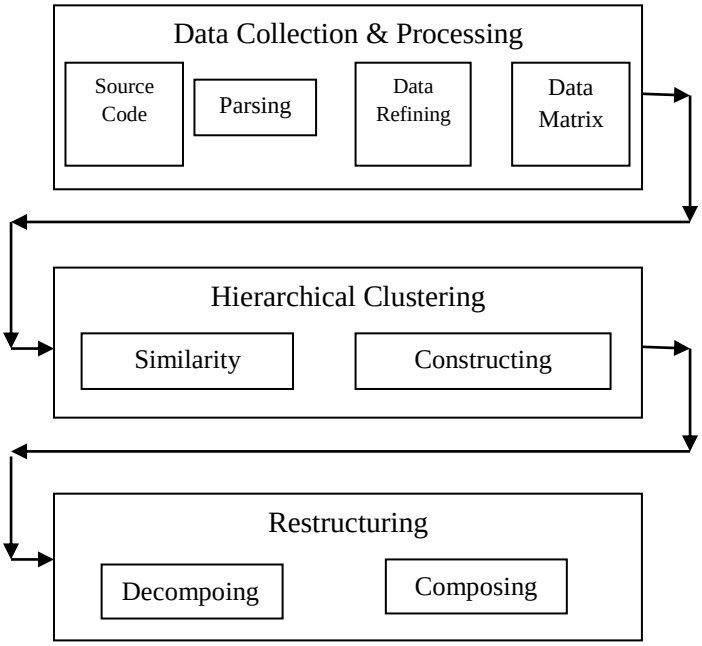
Cohesion and Clustering are most important part of the restructuring process and will also help software designers to identify ill-structured programs. The goal is to understand existing software and then to re-implement it.
This paper will discuss clustering methods. Various clustering methods have been used with different algorithms and the best algorithm of clustering will be using for program restructuring in our future work.

SOFTWARE COMPONENT RESTRUCTURING

Software component restructuring improves and modifies quality attributes of existing software products by the method of reorganizing the structure of software system [8]. It improves the style of coding, editing of documents and transforming of program components and also enhance software's functional structures. Software with poor structure results in the immature and incomplete process, inappropriate prototyping etc [8]. Even the well-designed software decline due to overtime maintenance. The software component restructuring can make the software system easy to understand and reusable. Restructuring of software also reduces cost at maintenance levels and also extends the lifetime of the software. Software restructuring technique is applied to designing and implementations as restructuring is done during designing of a new software system as well to the existing software system to recover integrity and improvement from deterioration of software.

METHODOLOGY

A. Hierarchical Agglomerative Clustering: For assigning a cluster to each component this algorithm is used. Two similar clusters are also merged with the help of this technique. This process goes on until there is one single cluster left. Usually, 3 HAC algorithms used- Single Linkage (SLINK), Complete Linkage (CLINK), and Weighted Pair Group Method Algorithms (WPGMA). Although there are various problems with hierarchical algorithms as when there are many data to make clusters then it performs very slow and also makes mistakes while merging valuable clusters into a single cluster and doesn't scale well. Moreover, previous work done cannot be undone using this algorithm. It also requires repetition of calculations to find similar clusters.
B. Fuzzy Logic: Deals with incomplete data, it can work with little data or with no data. It has many advantages as compared to other software computing methods. It does not rely on old values. This algorithm provides portability, better understandability, the lesser complexity of the output component.
C. Neural Network: NN represents input-output relationships and functions. It helps to predict the level of reusability of software by maintaining the relation between input and output. It also produces correct output from training or incomplete data. This method gives stability to software more than fuzzy logic. It represents software reusability from functionality.
D. Clustering analysis with cohesion is the approach which provides information about existing structure for a function. Fig.1 shows use of clustering techniques for program restructuring and also deals with challenging and fundamental issues of clustering technique like the definition of attributes and also entities inside the function and used an algorithm to calculate similar coefficients, selecting best method of clustering. The approach has three main phases. Phase 1 is collecting data and processing it. In this phase, the source code is parsed automatically by using parser tool and generating raw data from the matrix that contain entity- attributes. But this data may contain 'unwanted data' (noises) which can be eliminated during data refining. Entities are components that are clustered. Every entity has more than one attribute. On the basis of attributes, entities are grouped. Two entities can be closely related on the basis of how many attributes those two entities have in common. For applying clustering techniques defining entities and attributes are necessary.
The input data for the second phase is the matrix generated of entity attribute after data refining. The second phase is clustering and the most important step of clustering is the similarity measure. After defining entities and attributes a resemblance is calculated to find similarity among two entities. Several algorithms have proposed to compute resemblance coefficient for different applications [9, 10, 11]. After calculating resemblance coefficient, clusters are constructed using clustering algorithm. SLINK, WPGMA AND CLINK are three hierarchical agglomerative algorithms used. Different algorithms are used for different applications. The tools perform clustering automatically in this phase.
Phase three is a restructuring of the program. Program slicing technique is used in restructuring. Efforts of restructuring used to focus on making control flow of program easier as low-cohesive functions are decomposed into many code fragments and new functions are also formed. This phase operates manually.



RESULT

This graph shows the clustering technique used for particular data set entities and their attribute [14]. This survey is based on the development of time/effort taken before and after clustering. It can be seen that survey results can vary using different clustering techniques. Using Density Based Clustering allows software user/developer to visualize and view software component using clustering technique with the help of program restructuring with ease of code reusability and reduction of development time, effort, and cost.

This clustering technique is more flexible to various clusters as compared to other techniques. This clustering technique can easily group data using the cohesive method and also enhance the functional structure of software by relocating functional software components into new modulus.

CONCLUSION AND FUTURE SCOPE

We review four major techniques and after deep analysis, it is found that the performance of variants of hierarchical clustering may benefit the proposed software component clustering as it is more flexible and can handle a number of clusters. As this technique doesn't require high maintenance and is worth being further investigated. Also, this technique has the capability to demonstrate varied components with respect to size and cohesion criteria. We also surveyed about the agglomerative

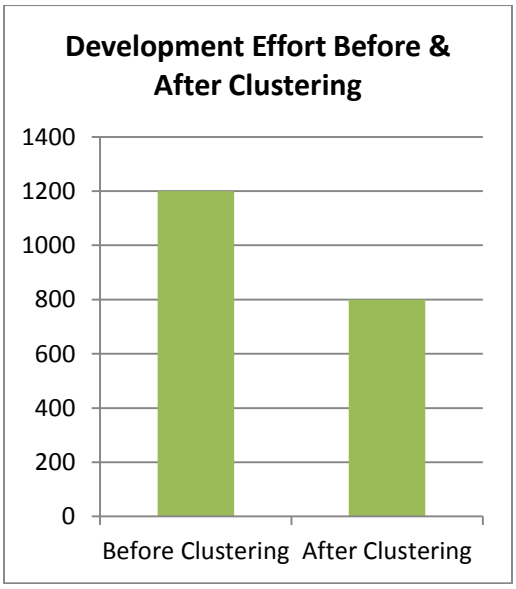


Fig. 1 Comparison of Development Effort before & after clustering

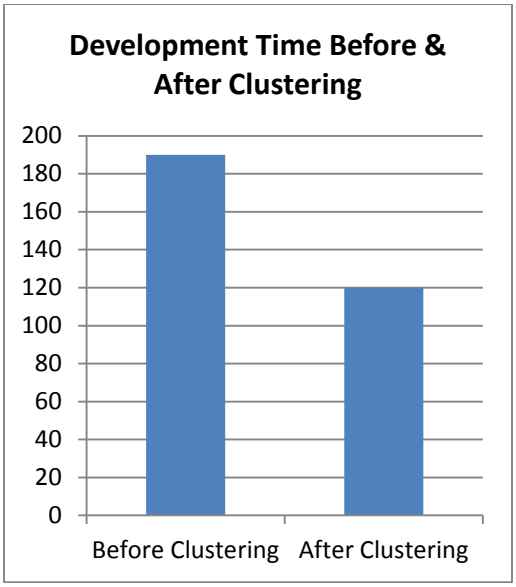


Fig. 2 Comparison of Development Time before & after clustering and divisive techniques. These approach of hierarchical techniques is basically is used for poorly designed software. The parameter tuning of this clustering method can raise the performance bar. In future, a Hierarchical method of clustering will be using for higher code reusability, better understandability, and maintainability of software components.

REFERENCES

[1] Sommerville. "Software Engineering". 5th ed. Addison –Wesely. England. 1996.
[2] Chikofsky. E.J... Cross. "Reverse Engineering and Design Recovery: A taxonomy". IEEE Software..1990.
[3] Fowler.M.. "Refactoring: Improving the design of existing code". Addison-Wesely.1999.
[4] Briand. L...Morasca. S... Basili. "Property Based software engineering measurement. IEEE Trans. Software Engineering. 1996.
[5] Munson. C.J... "Software engineering measurements". Auerbach Publications. ACRC Press Company. 2003.
[6] Pressman. R.S... "Software Engineering: A Practitioner's Approach". 4th edition McGraww-Hill.Inc. 1997.
[7] Wiggerets. T.A... "Using Clustering Algorithms in Legacy Systems Modularization" Fourth Working Conference on Reverse Engineering. 1997.
[8] Arnold. R.S. "Software Restructuring". Proc. IEEE. 1989.
[9] Everitt. B. "Cluster Analysis". Heinemann Educational Books. London.
[10] Romesburg. H.C. "Cluster Analysis for Researchers". Krieger Publishing Company. Malbar.FL. 1990.
[11] Sneath. P.H.A. Sokal.R.R. "Numerical Taxonomy: The Principles and practice of Numerical Classification". W.H. Freeman and Company, San Francisco. 1973.
[12] Duo Liu. Chung-Horng Lung. Samuel A. Ajila, "Adaptive Clustering Techniques for Software Components and Architecture". IEEE, 39th Annual International Computers, Software & Applications Conference.2015.
[13] Chung-Horng Lung. Xia Xu. Marzia Zaman and Anand Srinivasan. "Program Restructuring Using Clustering Techniques". Journal of systems and software. 2006.
[14] Matrika Sinha. Shreya Jain. "Program Restructuring And Component Reuse With Data Mining Techniques". International Journal of Engineering Research and General Science Volume-4, Issue-3. 2016.
[15] Ronaldo C. Veras and Silvio R.L. Meria. Adriano L.I. Oliveiria and Bruno J.M. Melo. "Comparative Study of Clustering Techniques for the Organisation of Software Repositories". 19th IEEE International Conference on Tools with Artificial Intelligence. 2007.

