A SURVEY ON CLUSTERING TECHNIQUES FOR IDENTIFICATION OF EXTRACT CLASS OPPORTUNITIES

Suchithra Chandran[1], Bright Gee Varghese.R[2]

[1]Post Graduate Student, [2]Assistant professor, Department of Computer Science and Engineering, Karunya University, Tamil Nadu, India

suchithracc@gmail.com, brightfsona@yahoo.co.in

Abstract

Refactoring is a growing research area in the field of software remodularization. Refactoring is an essential practice in software development field. Refactoring is done to clean up the code and to minimize the chance of introducing the bugs. Extract class refactoring is done to improve the design of the system thereby increasing the cohesion among the class members and reducing the coupling between two classes. Extract class refactoring is performed on large, complex and less cohesive classes, which are doing functions that should be split into two or more classes. Such large and complex classes are decomposed to several classes during refactoring. During refactoring a new class is created and the entities that perform a function are moved to it. For extract class refactoring the classes to be extracted for refactoring has to be identified first. The identified refactoring opportunities are then evaluated to check whether they preserve the original behavior of the system. Refactoring is performed even after the release of the software to improve its performance. Clustering is a technique used to group the similar entities in a class. Several clustering algorithms are used to find these refactoring opportunities. In this survey various clustering techniques to identify those classes are reviewed considering its advantages and disadvantages.

Keywords: *Clustering, Object-oriented system, Refactoring, Software remodularization*

----------***----------

1. INTRODUCTION

Main aim of Object Oriented (OO) design is to distribute the responsibilities of a software system into several classes to reduce the system complexity. To reduce the complexity the large and complex classes are decomposed to form several smaller classes. A class with more than one responsibility has low cohesion. So decomposition of these classes should be done to improve the cohesion.

In [1] (Fokaefs.M et al. 2012) says, refactoring is the process of restructuring the existing code of a system by preserving the external behavior. It is done to improve the design structure of the system. To improve the design of a system its cohesion should be increased and the coupling should be decreased. Extract Class refactoring is a technique of decomposing the large and complex classes with one or more responsibilities. Decomposition is done by splitting the classes performing more than one function or responsibility which should be done by more than one class. Before performing Extract Class refactoring, the first step is to identify the classes which should be extracted to form a separate class to improve the cohesion. Clustering technique is usually used to find the similar methods and attributes used for performing a function in a class. Clustering method includes: Hierarchical clustering algorithms, Partitional clustering algorithms, Graph theoretic clustering.

2. HIERARCHICAL CLUSTERING

Hierarchical Clustering Algorithm is one of the most widely used technique for clustering the related entities from a highly complex class to form a separate class.

In [2] (Lung.C et al. 2006) proposed a method for restructuring the program using clustering technique. The proposed approach has four phases. First phase is data collection and processing. During this phase the source code is parsed and an entity-attribute matrix is generated. The entities in a class are clustered based on the attributes they share. Second phase is clustering. During clustering the resemblance coefficient metric is calculated to measure the similarity between the entities. Then clusters are formed using the three hierarchical agglomerative clustering algorithm: SLINK (Single Linkage), CLINK (Complete Linkage), WPGMA. Then among these the best algorithm selected. Third phase is visualization and analysis. The result of clustering phase is represented in the form of tree. From these the clusters are identified. Fourth phase is restructuring. The complex classes identified are decomposed and formed several classes thereby increasing cohesion. The advantage of the technique is that it identifies low-cohesive functions and performs restructuring to improve the quality of ill-structured programs.

In [3] (Czibula.L et al. 2007) proposed a hierarchical agglomerative clustering technique for restructuring software systems(HARS), that identifies the refactorings needed to restructure the software system to improve the quality of the system. The clustering approach for refactoring determination(CARD) consists of three steps: data collection, grouping and refactorings extraction. During grouping the entities identified at previous step is re-grouped using clustering algorithm. The clustering algorithm used is hierarchical agglomerative clustering algorithm. The linkage metrics used is complete linkage, maximum distance between the members of two clusters. In HARS algorithm, each entity from the software system is put in its own cluster. Then select two most similar clusters from the current partition. Merge the two clusters into a single new cluster. The number of clusters in the partition is now decreased. The advantage of this approach is that the overall running time of HARS algorithm is reduced to 3.68 minutes when compared to partitioning clustering algorithm, which is about 5 minutes.

In [4] (Fokaefs.M et al. 2009) proposed a method for decomposition of large classes to improve coupling and cohesion using Agglomerative Clustering algorithm based on Jaccard distance between the class members to identify the attributes and methods that can be extracted to a separate class. This identification has two parts, clustering and filtering. Clustering is done using agglomerative clustering algorithm to identify the classes to be extracted. Filtering is done using a set of rules to evaluate the degree of functionality and to check whether the suggested refactorings preserve the behavior of the system. This methodology does not define a fixed threshold value for distance instead its applied for several threshold values ranging from 0.1 to 0.9. This method also produces large number of suggestions that may include duplicate suggestions.

In [5] (Rao.A et al. 2011) proposed another method using metrics supplemented agglomerative clustering technique to identify the low cohesive classes to extract them to a separate class thereby improving cohesion. The proposed approach has two steps. First identifying low cohesive classes using the metrics LCOM (Lack of Cohesion in Methods), TCC (Tight Class Cohesion) values. Secondly, the metrics supplemented agglomerative clustering technique is applied to the low cohesive classes. During this step the similarity between the class members are calculated using Jaccard similarity metric value and then agglomerative clustering algorithm is used to find the cluster of class members at a specified threshold to be extracted to separate class. The advantage of this approach is that it can handle a situation where agglomerative clustering alone cannot find proper clusters.

In [6] (Cassell.K et al. 2011) proposed a dual clustering approach for extract class refactoring. The approach uses a divisive clustering based on structural information, followed by agglomerative clustering based on semantic information. The divisive structural clustering algorithm known as betweenness clustering separates the class members into groups based on the dependencies between the class members. Betweenness clustering is a graph-based technique where the nodes represent the class members and edges represent the dependency among them. The betweenness value is calculated to group the class members. Then an agglomerative clustering technique based on semantics is used to combine the smaller groups with the larger ones. The technique starts with seed clusters and then it adds closely related clusters to them until the stopping criterion is reached. Betweenness clustering uses extended structural information in addition to local information. The disadvantage is that the technique is complex as it uses two complementary algorithms.

3. PARTITIONAL CLUSTERING

In partitional clustering methods, an initial cluster is given and then the instances are moved from one cluster to other to obtain better cluster.

In [7] (Czibula.L et al. 2006) has proposed a partitional clustering technique called k-means for refactoring determination (kRED). The clustering approach has three steps: data collection, grouping and refactorings extraction. During grouping the set of entities identified at the previous step is re-grouped to clusters using k-means algorithm. For the kRED algorithm, the initial number of clusters is given and the initial centroid is defined. The clusters are recalculated when each object is assigned to the closest cluster. This is repeated until two consecutive iterations remain unchanged or if the number of iterations performed exceeds the maximum number of iterations allowed. The disadvantage is that the performance depends on the initial centroid, so no guarantee of optimal solution.

In [8] (Czibula.L et al. 2008) proposed a partitional clustering algorithm for improving the structure of the object-oriented software systems. The clustering approach used here has three steps: Initially the system is analysed and relevant entities are extracted, the entities extracted are re-grouped using partitional clustering algorithm for refactoring determination (PARED), finally the new system is compared with the original to obtain the list of refactorings. In PARED technique, the initial number of clusters and the initial medoids are determined. Then the clusters and medoids are recalculated when each object is assigned to the closest medoid. This is repeated until there is no change in the partition. The approach has an overall running time of 1.2 minutes which is less when compared with the kRED algorithm. The evolutionary algorithms has to be executed 10 times to obtain a stable result, while PARED algorithm has to be executed just once. The disadvantage of this technique is that the result is not finite as the initial partition can change.

4. GRAPH THEORETIC CLUSTERING

In graph-theoretic clustering methods, clusters are formed from graphs. In [9] (Mancoridis.S et al. 1998) proposed an approach to develop high-level system code by restructuring the source code. In the proposed approach, initially the source code is parsed to obtain the system modules and module-level dependency and is represented as a Module Dependency Graph(MDG). Then several automatic software modularization algorithms are applied to automatically partition the graph into clusters in a way so as to minimize inter-connectivity and maximize intra-connectivity. Modularization Algorithms used might be Optimal Clustering Algorithm, Sub-Optimal Clustering Algorithm, Genetic Algorithm. Then builds a hierarchy of clusters using hierarchical clustering algorithm. For systems with large number of modules, hierarchical clustering and genetic clustering algorithm produces optimal solution. Automatic modularization techniques are useful to programmers who lack familiarity with a system. The disadvantage is that the modularization quality does not consider the Interconnection Strength (IS) of the relationships that exist between the modules in the system.

In [10] (Xanthos.S 2006) proposed a spectral graph partitioning technique to identify the large classes in the object-oriented system. In this approach the class diagram is analysed and a graph is formed where the vertices are the classes and the edges are the messages they exchange. Then this graph is iteratively bipartitioned. In each iteration the graph is partitioned to two sub-graphs. The iteration stops when a less cohesive class is formed during the partition. These sub-graphs forms separate classes thereby increasing cohesion. This approach minimizes the communication between the modules of the system thereby reducing coupling. The disadvantage is that it's hard to find an optimal partition.

In [11] (Bavota.G et al. 2010) proposed a technique for identifying extract class opportunities exploiting structural and semantic relationships between the methods of the classes. The proposed approach starts from a class with low cohesion, the input class is parsed in order to extract a weighted graph representing it, each node represents a method of the class and weight of an edge represents a measure reflecting structural relationship such as attribute references, method calls and semantic relationship of two connected methods. Once the graph is computed a MaxFlow-MinCut algorithm is used to partition the original graph into two subgraphs, cutting a minimum number of edges with low weight. This two sub graphs form two new classes that have higher cohesion than original class without increasing coupling. Better refactoring opportunities are identified by using the combination of structural and semantic cohesion measures. The disadvantage is that the MaxFlow-MinCut algorithm is able to split the original class into only two sub-graphs, so only two new classes can be extracted.

5. COMPARISON OF VARIOUS ALGORITHMS

The following table shows the comparison of various clustering algorithms used for identifying the extract class opportunities.

Table -1: Comparison of various clustering algorithms

Parameter	Hierarchical Clustering	Partitional clustering	Graph partitioning
Running Time	Less	Higher compared to hierarchical clustering.	Depends on the partitioning algorithm.
Solution	Produces optimal solution.	No guarantee of optimal solution.	No guarantee of optimal solution.
Performance	Good Performance.	Performance depends on initial cluster.	Performance depends on the partitioning algorithm.
Deterministic or Non-deterministic output	Deterministic	Non-Deterministic	Deterministic
Finite or Non-finite output	Finite	Non-finite	Non-finite

6. CONCLUSION

This paper gives a survey on various clustering techniques for identifying the extract class opportunities. The survey showed that there are several clustering approaches for the identification. Among the techniques reviewed, hierarchical clustering technique identifies better extract class opportunities for performing extract class refactoring than partitioned or any other clustering algorithms.

ACKNOWLEDGEMENTS

I am highly indebted to Mr. Bright Gee Varghese, Assistant Professor, Department of Computer Science and Engineering for his guidance and I would like to thank all the authors of the references which were very helpful for this survey.

REFERENCES

[1]. Fokaefs,M., Tsantalis,N., Stroulia,E., Chatzigeorgiou,A.,2012. "Identification and Application of Extract Class refactorings in object-oriented systems," Journal of Systems and Software, vol.85, issue 10, pp. 2241-2260.

[2]. Chung-Horng Lung, Xia Xu, MarziaZaman, and Anand Srinivasan, 2006."Program Restructuring through Clustering Techniques," SCAM '06: Proceedings of the Source Code Analysis and Manipulation, Fourth IEEE International Workshop, pp. 75- 84.

[3]. Czibula,I. G. and Serban, G., 2007. "Hierarchical clustering for software systems restructuring," INFOCOMP Journal of Computer Science, Brasil6 , no. 4, 43 - 51.

[4]. Fokaefs,M., Tsantalis,N., Stroulia,E., Chatzigeorgiou,A., 2009. "Decomposing object-oriented class modules using an agglomerative clustering technique," In: 25th IEEE International Conference on Software Maintenance (ICSM'2009), Edmonton, AB, Canada.

[5] AnandaRao.A, and K. Narendar Reddy, 2011. "Identifying Clusters of Concepts in a Low Cohesive Class for Extract Class Refactoring Using Metrics Supplemented Agglomerative Clustering Technique," International Journal of Computer Science Issues, vol. 8, issue 5, no. 2, pp. 185-194.

[6] Keith Cassell, Peter Andreae, and Lindsay Groves, 2011."A Dual Clustering Approach to the Extract Class Refactoring," 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE'11), Miami, FL, USA.

[7] Czibula, I. G. and Serban, G., 2006. "Improving Systems Design Using a Clustering Approach," International Journal of Computer Science and NetworkSecurity (IJCSNS), 6(12):40–49.

[8] Czibula.I.G, G.Serban, 2008."A partitional clustering algorithm for improving the structure of object-oriented software systems," StudiaUniversitatis Babes-Bolyai, Informatica 53 (2), 105-114.

[9] Mancoridis, S., Mitchell, B.S., Rorres, C., Chen, Y., Gansner, E.R.,1998. "Using automaticclustering to produce high-level system organizations of source code," In: 6thInternational Workshop on Program Comprehension. IEEE Computer SocietyPress, pp. 45–52.

[10 Xanthos, S., 2006. "Clustering Object-oriented Software Systems using Spectral GraphPartitioning," ACM Student Research Competition.

[11] Bavota,G., De Lucia, A., Oliveto, R., 2010. "Identifying extract class refactoring opportunities using structural and semantic cohesion measures," Journal of Systemsand Software 84, 397–41.

BIOGRAPHIES

Suchithra Chandran is pursuing MTech (Computer Science and Engineering) degree from Karunya University, Tamil Nadu, India. She received her B.E (Computer Science and Engineering) degree from AMS Engineering College, Tamil Nadu, India.

Bright Gee Varghese. R completed his B.E (Computer Science and Engineering) from Sun College of Engineering and Technology, Nagercoil, Tamil Nadu, India in 2003. He started his career as Lecturer in Sun College of Engineering and Technology, Nagercoil, from June 2003. He completed his M.E from Vinayaka Mission university, Salem, Tamil Nadu in 2011.Currently he is working as an Assistant Professor in Computer Science Department in Karunya University and pursuing part time Ph.D in Karunya University, Tamil Nadu, India. His main research area is Software Engineering.