A Cluster Based Reusability Model with Reference to Aspect Mining

Rajender Nath, Munishwar Rai, Jai Bhagwan
Department Of Computer Science and Application, Kurukshetra University, Kurukshetra
Mmitcbm, Maharishi Markandeswary University, Mullana
Department Of Computer Science And Engineering, Guru Jambheshwar University Of Science and Technology, Hisar

Abstract—Clustering is the process of classification of similar items at one location and dissimilar to another. According to our approach, cluster analysis of aspects or modules can play a major role in maintenance of software. Aspect oriented programming is an emerging paradigm to handle the crosscutting concerns. The features of AOP make it one of the powerful programming approaches. Once you have clustered your aspects, you can reuse these effectively. Our research is targeting a new model for system partitioning used in AOP, which will be effective in the maintenance phase of software development life cycle. Hierarchical and non-hierarchical algorithms are used for clustering, so we opt agglomerative and k-mean method to finalize our model.

Index Terms—Aspect Oriented Programming, Partitioning, Clustering, Cross cutting concerns.

I. INTRODUCTION

Although research on aspect mining is still in its infancy, yet several prototype tools have already been developed that support developers in identifying crosscutting code. These tools vary in accuracy and the level of automation that they offer [8]. Aspect Oriented Programming (AOP) is an extension to object-oriented programming by facilitating another type of modularity that deals with the implementation of a crosscutting concern into a single unit, called aspect (An Aspect is a special class to handle crosscutting concerns), hence the name aspect-oriented programming. The separation of concerns is one of the major principles in software development. AOP deals with a technique to support separation of cross-cutting concerns (SOC) like synchronization, resource sharing and security etc. which could be effectively handled by aspects. By using aspects the cohesion, robustness, adaptability and reusability features of a module are increased and consequently the quality of the software is improved. From a maintenance point of view, such a crosscutting concern is problematic [9]. The crosscutting concerns are mainly due to scattering and tangling of code. The process of code scattering means code involved in a crosscutting concern is spread across the system and code tangling means a concern is intersected with some other concern [7]. The process, which identifies the scattered and tangled code, is called aspect mining or clustering of aspects. It is a specialized process of reverse engineering that identifies crosscutting concerns such as synchronization, error management, persistence, security and logging. A number of aspect mining techniques have been proposed in the literature on the basis of different factors like formal concept analysis, metrics, clustering etc. to identify the crosscutting concerns. A crosscutting concern is a feature of a software system that is spread all over the system, and whose implementation is tangled with other features’ implementation. Logging, persistence, and connection pooling are well-known examples of crosscutting concerns [6]. Cluster analysis is a course of identification and categorization of subsets of objects. Partitioning or clustering techniques are used in many areas for a wide spectrum of problems. Among the areas in which cluster analysis is used are graph theory, business area analysis, information architecture, information retrieval, resource allocation, image processing, software testing, chip design, pattern recognition, economics, statistics, biology and many more [10]. We propose a new aspect mining technique which is structured in the rest section (proposed model).

II. LITERATURE REVIEW

The literature review is the backbone of a research, so we have studied the following research papers in order to develop a better model rather than our previous research [10]. C. Gabriela et al. [1] proposed that clustering is a division of data into groups of similar objects. Aspect mining is a process that tries to identify crosscutting concerns in existing software systems. The goal is to re-factor the existing systems to use aspect oriented programming, in order to make them easier to maintain and to evolve. The aim of this paper is to present a new hierarchical clustering based approach in aspect mining. For this purpose the authors propose HAC algorithm (Hierarchical Agglomerative Clustering in aspect mining). Clustering is used in order to identify crosscutting concerns. The authors evaluate the obtained results from the aspect mining point of view, based on two quality measures that they have previously introduced and a newly defined one. The proposed approach is compared with other similar existing approaches in aspect mining and two case studies have also been reported. Aspect oriented Programming (AOP) is a relatively new paradigm that is used to design and implement crosscutting concerns. Aspect mining is a relatively new research direction that tries to identify cross-cutting concerns in already developed software systems. The goal is to identify them and then to re-factor them to aspects, to achieve a system that can easily
be understood, maintained and modified. In this paper the authors have presented a new hierarchical agglomerative clustering approach in aspect mining that uses a newly defined algorithm, HAC. The authors have evaluated the obtained results from the aspect mining point of view based on three quality measures. The authors have given a definition in order to compare two partitions from the aspect mining points of view. Based on this definition, they have shown that HAC algorithm provides better partitions. Gabriela et al. [2] have proposed a new graph-based approach in aspect mining. The authors define the problem of identifying the crosscutting concerns as a search problem in a graph and they introduce GAAM algorithm (Graph Algorithm in Aspect Mining) for solving the problem. The authors evaluate the results obtained by applying GAAM algorithm from the aspect mining point of view, based on a set of quality measure that the authors have studied earlier. The proposed approach is compared with a clustering approach in aspect mining and a case study is also reported. In order to evaluate the obtained results from the aspect mining point of view, the authors have used a set of quality measures. They have given a definition in order to compare two partitions from the aspect mining point of view. Based on this definition, they showed that GAAM algorithm provides better partitions than kAM algorithm. M. S. Grigoreta et al [3] proposed Aspect Oriented Programming is a programming paradigm that addresses the issue of crosscutting concerns. Aspect mining is a process that tries to identify crosscutting concerns in existing software systems. The goal is to re-factor the existing systems to use aspect oriented programming to make them easier to maintain and to evolve. This paper presents a new approach in aspect mining that uses clustering and proposes two techniques: a k-means based clustering technique and a hierarchical agglomerative based clustering technique. The authors are trying to identify the methods that have the code scattering symptom. For the method, the authors consider as indication of code scattering a big number of calling methods and, also, a big number of calling classes. In order to group the best methods (candidates), the authors use in their approach, the vector-space model for defining the similarity between methods. For testing the efficiency of the proposed techniques, a number of Java applications are being used. These try to identify the methods used to implement crosscutting concerns that have the scattered symptom. For that the authors compute the fan-in metric of each method that is called inside the application classes, and the number of classes that call this method. The obtained results are divided in clusters using two clustering algorithms: HACA and k-means. Some of the obtained clusters are then manually analyzed to determine if they contain methods used to implement crosscutting concerns. The authors have studied on JHotDraw open source project. Cojocar and Serban [4] proposed that clustering is a division of data into groups of similar objects. Aspect mining is a process that tries to identify crosscutting concerns in existing software systems. The goal is to re-factor the existing systems to use aspect oriented programming in order to make them easier to maintain and to evolve. The aim of this paper was to analyze and evaluate comparatively the results obtained by different clustering algorithms in aspect mining. The evaluation was performed on an open source case study (JHotDraw) using four measures. In this paper the authors have analyzed and evaluated comparatively the results obtained by different clustering algorithms KM, FCM, HAC, GAM, kAM, and HAM) for crosscutting concerns identification. The evaluation was performed using the obtained results show that clustering can be useful in aspect mining, but some improvements are needed. If a clustering based algorithm succeeds in obtaining an optimal partition of a software system, then the results can be integrated with tools that (automatically) perform aspect oriented based re-factorings, as a first step to re-factor. B. Silvia et al [5] have proposed aspect mining identifies cross-cutting concerns in a program to help and migrate it to an aspect-oriented design. Such concerns may not exist from the beginning, but emerge over time. By analysing where developers add code to a program, the author’s history-based aspect mining (HAM) identifies and ranks cross-cutting concerns. The authors evaluated the effectiveness of their approach with the history of three open-source projects. HAM scales up to industrial-sized projects: for example, they were able to identify a locking concern that cross-cuts 1284 methods in Eclipse. Additionally, the precision of HAM increases with project size and history; for Eclipse, it reaches 90% for the top-10 candidates. Here PhotoDraw is used for case study as well. B. Jai et al. [10] proposed that cluster analysis is a system used for cataloging of data in which data elements are screened into groups called clusters that represent collections of data elements that are based on a distance or dissimilarity. The cluster analysis approach is an important tool in decision making and an effective creativity technique in generating ideas and obtaining solutions. This research covers different types of methods such as Hierarchical Clustering (HC) and Non-Hierarchical Clustering (NHC) techniques for software modules classification. The authors used JB clone scanner to find out the clones from 15 C++ programs.

III. PROPOSED MODEL

Our previous research [10] is not fit for a live project as we have tested it. So to improve this we are giving a new model with few tools. The model is structured below:

1. Identify similar behavior modules in a project.
2. Find out the number of Clusters to be made using agglomerative method as follows.
   a) Start with N clusters, each containing a single entity and a N X N symmetric matrix of distances (or similarities).
      Let
      \[ \text{Dij} = \text{distance between item i and j items} \]
   b) Search the distance matrix for the nearest pair clusters (i.e., the two clusters that are separated by the smallest distance).
Denote the distance between these most similar clusters \( u \) and \( v \) by \( D_{uv} \).

(c) Merge clusters \( u \) and \( v \) into a new cluster, labeled \( T \).

Update the entries in the distance matrix by

(i) Deleting the rows and column corresponding to clusters \( u \) and \( v \) and

(ii) Adding a row and column giving the distances between the new cluster \( T \) and all the remaining clusters.

(d) Repeat steps b and c for a total of N-1 times.

3. Merging of similar clusters using K-mean method as follows:

The k-means algorithm is a method for partitioning data points into clusters. Let \( X = \{x_1, x_2, \ldots, x_n\} \) be a set of points in Rd. After being seeded with a set of k centers \( c_1, c_2, \ldots, c_k \) in \( R_d \), the algorithm partitions these points into clusters as follows,

(a) For each \( i \in \{1, \ldots, k\} \), set the cluster \( C_i \) to be the set of points in \( X \) that are closer to \( C_i \) than they are to \( C_j \) for all \( j \neq i \).

(b) For each \( i \in \{1, \ldots, k\} \), set \( C_i \) to be the center of mass of all points in

\[
C_i := \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j
\]

(c) Repeat steps 1 and 2 until \( c_i \) and \( C_i \) no longer change, at which point return the clusters \( C_i \).

4. Calculate and analyze the development time and efforts before and after partition, by which the maintenance can be optimized.

IV. CONCLUSION

In this paper, in order to obtain the perfect clustering, we have proposed a model (method) based upon LOC and similar elements (relation between functions or modules). The number of clusters will be determined by using agglomerative method and then merging of clusters via K-mean method. As we have discussed in literature review, our last approach is not fit for a big sized live project. The buffer problem is fired during the test of a live project. That’s why we will use an additional tool with JB Clone Scanner. The future work can be done in the direction of the implementation of our approach for live projects. A case study should be done on a live project to enhance the reusability of the programs in the comparison of efforts, development time; so that the maintenance can be done easily in the field of Software Engineering.

REFERENCES


[5] B. Silvia and Z. Thomas, “Mining Aspects from Version History”, Computer Laboratory, University of Cambridge, UK, Department of Computer Science, Saarland University, Germany.


AUTHOR’S PROFILE

Prof. Rajender Nath is working as a Professor at DCSA, Kurukshetra University, Kurukshetra. He has more than 24 years of teaching experience. His research area is Computer Architecture & Parallel Processing, Object Oriented Modeling, Bio-informatics and Aspect Oriented Programming. He has published more than 30 research papers in both international and national journals. He is involved in various academic activities at Kurukshetra University, Kurukshetra.

Munishwar Rai is currently working as Associate Professor at MICT&BM (MCA department) of MM University, Mullana, India. He has around 15 years of teaching experience. He has received his MCA degree from M.D. University, Rohtak. His area of interest is software Reusability, Data mining, Computer Organization and computer networks. He is a life member of CSI. He is involved in various academic and automobiles administrative activities at MM University, Mullana.

Jai Bhagwan is currently working as Assistant Professor in the department of Computer Science and Engineering, GIU S&T, Hisar, India. He has 2 plus year of teaching experience. He has been worked as Lecturer in IT Deptt, MM University, Mullana, India during 2011-12. He received his M.TECH (CSE) degree from Maharishi Markandeshwar University, Mullana in 2011. He has developed as well as maintained several
projects like Campus Suite, Bijli Suvidha Kendra,
DHBVN Billing System, JB Clone Scanner etc.
His current research area is Software Engineering
and Data Minding.