SIMILARITY AND DISSIMILARITY MEASURE FOR CLASS CLUSTERING

Bharti Chhabra *

Ashish Oberoi **

Sunil Kumar ***

ABSTRACT

Clustering provides a way to easily retrieve the software components. There are many clustering techniques available but choosing the right kind of clustering depends upon the type of data.

We have developed a class clustering technique which focuses on the clustering of the classes on the basis of similarity and dissimilarity measures. This measure can be further analyzed from class attributes and methods. The classes are taken from the logical view of the software Rational ROSE which is an IBM modeling tool. Further a binary matrix has been developed which emphasizes on the presence and absence of class attributes and methods. Our approach resulted in the clustering of the classes on the basis of similarity and dissimilarity measures.

Keywords: *Clustering, UML, Binary Matrix*.

* Maharishi Markandeshwar Engineering College, Mullana, Haryana.

** Assistant Professor, Maharishi Markandeshwar Engineering College, Mullana.

*** Assistant Professor, Haryana Engineering College, Jagadhri.

INTRODUCTION

Software Design is one of the crucial part in software development. It needs in-depth analysis so that further delays can be avoided. Software development can be enhanced if it is integrated with the concept of reusability [5]. Reusability enhances the development in terms of time cost and effort. There are some factors on which reusability depends one such factor is the availability of the component to be reused. This availability can be further enhanced if the components are grouped or clustered on the basis of some parameters like similarity or dissimilarity measures. In our approach we are clustering the classes on the basis of attributes, methods and relationships. The class diagram we are using is taken from the logical view in the Rational ROSE. Rational ROSE is a modeling tool of IBM. There are other views like use-case view, component view and deployment view[10] in Rational ROSE[9]. The logical view or static view has been mapped from the use-case view. After analyzing the logical view in terms of class name, attributes, methods and relationship between classes, a binary matrix has been formed. This matrix indicates the absence and presence of attributes, methods and relationship amongst the various classes. Further a dissimilarity measure has been calculated justifying the degree of dissimilarity among classes.

LITERATURE SURVEY

R. Ibba et al [8] has emphasized on the design based reuse of software components. The approach has used a set of metrics to create clusters of existing components on the basis of their similar internal structure and also performed functional similarity checks.

Yves Chiricota et al [14] has described a method for determining clusters of software systems. The author has applied a straight forward metric which is used to find out the weak edges. The deletion of weak edges resulted into several components. The quality metric MQ has also been used.

Xie Binhong et al [13] has used Grade strategy to assign a grade weight to each facet and has proposed a component clustering algorithm. The comparison of Vector space model and latent semantic analysis for component clustering is shown. The technique defines that the quality of component classification can be improved using Grade strategy.

Chung-Hong Lung et al **[**2] has defined a technique based on cohesion and coupling information of a software system and has applied it to real time software system in telecom and computer networks.

Shi Zhong et al [12]has described a technique for unifying bipartite graph view of probabilistic model based clustering and has also analyzed the model based partitional clustering mathematically from a deterministic annealing perspective. The approach has also discussed two new variations balanced model clustering and hybrid model based clustering.

Istvan Gergely Czibula et al [6] has defined an approach for clustering using refactoring which helps in improving the internal structure without affecting the external structure. A new k-means based clustering approach (CARD) is also proposed.

METHODOLOGY

Our approach follows the process view given below

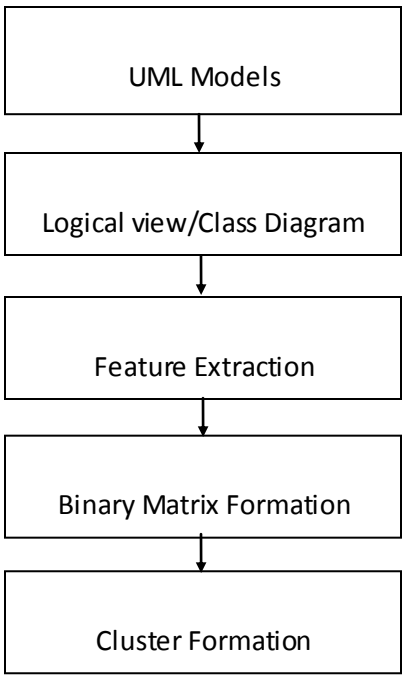

Figure 1: Process View

Modeling a software requires various views to be modeled into a modeling tool. We have Rational ROSE[9] as a modeling tool and logical view as the origin from which the data is to be clustered. Logical view consists of a class diagram which further includes class name, its attributes, methods and relationship amongst the various classes. On the basis of presence and absence of attributes, methods and relationship a binary matrix has been formed. We have tried to justify our technique with the help of an example as below:

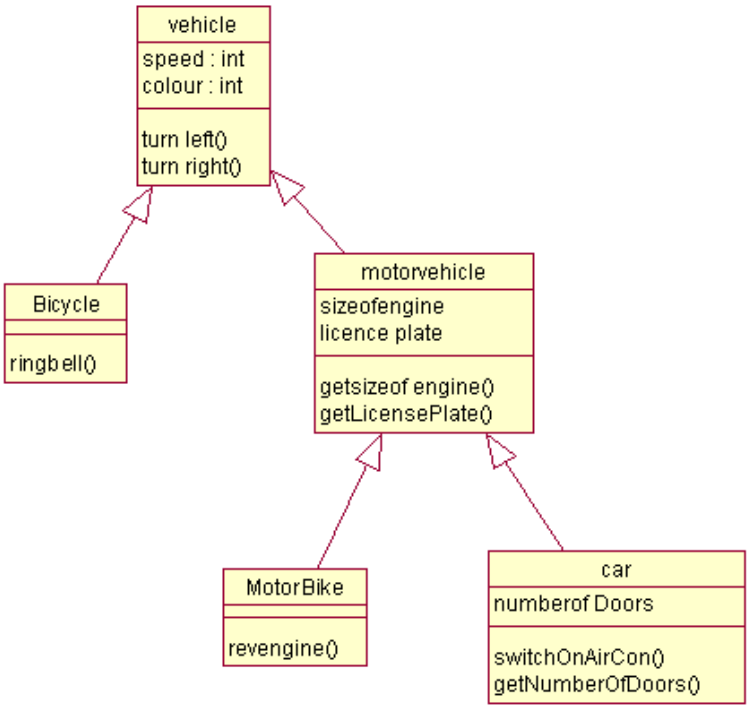

Figure 2: Class Diagram

After analyzing the above diagram the binary matrix is shown below:

Binary Matrix

	AT1	AT2	AT3	AT4	AT5	M1	M2	M3	M4	M5	M6	M7	M8	M9
C1	1	1	0	0	0	1	1	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	1	0	0	0	0	0	1
C3	0	0	1	1	0	0	0	0	1	1	0	0	0	1
C4	0	0	0	0	0	0	0	0	0	0	1	0	0	1
C5	0	0	0	0	1	0	0	0	0	0	0	1	1	1

Table 1:Binary Matrix

In the fig:

AT1…AT5 denote the attributes of classes C1…C5

M1…M9 denote the methods of classes C1…C5

Now apply the similarity measure [7]

$$S_{mm} = \frac{J_{ij\ i=j=1}}{J_{ij\ i=0,j=1} + J_{ij\ i=1,j=0} + J_{ij\ i=j=1}}$$

where S_{mm} is the similarity measure

J_{ij} is a binary variable.

After calculation and analysis we get the following similarity matrix

Similarity Matrix

	(C1)	(C2)	(C3)	(C4)	(C5)
(C1)	1	0	0	0	0
(C2)	0	1	.16	.33	.2
(C3)	0	.16	1	.16	.12
(C4)	0	.33	.16	1	.2
(C5)	0	.2	.12	.2	1

Table 2: Similarity Matrix

Analysis from the above matrix result into

1) The similarity between the same classes comes out to be 1, it means if two classes are equally similar in terms of attributes and methods then their value comes out to be 1.
2) The dissimilarity can be calculated by subtracting the similarity from 1.
3) The relationship between the classes like association, aggregation and generalization does not play any significant role.

Dissimilarity Matrix

	(C1)	(C2)	(C3)	(C4)	(C5)
(C1)	0	1	1	1	1
(C2)	1	0	.84	.67	.8
(C3)	1	.84	0	.84	.88
(C4)	1	.67	.84	0	.8
(C5)	1	.8	.88	.8	0

Table 3: Dissimilarity Matrix

CONCLUSION

From the above analysis of similarity and dissimilarity matrix it has been concluded that clustering of classes depends upon the similarity and dissimilarity of attributes and methods. The relationship between the classes contributes very less towards similarity and dissimilarity evaluation. Our approach describes a way to cluster the components having attributes or characteristics.

FUTURE SCOPE

Our approach has provided a way to cluster the class components. However there are other components also which have attributes. Hence this approach can be used for other UML components.

REFERENCES

1. Alexender Egyed "UML/Analyzer "A Tool for Instant Consistency Checking of UML Models". Precedings of the 29th International Conference on Software Engineering (ICSE-2007), Minneapolis, USA, May 2007.
2. Chung-Horng Lung, Amit Nandi, Marzia Zaman "Applications of Clustering to Early Software Life Cycle phases" Proc. of Int'l Conf. on Software Eng. Research and Practice 2002.
3. Chuangxin Yang ,Hong Peng ,Jiabing Wang "A clustering Algorithm for Weighted Graph Based on minimum Cut" First International Conference on Intelligent Networks and Intelligent Systems IEEE 2008
4. Grady Booch, Ivar Jacobson & Jim Rumbaugh (2000) OMG Unified Modeling Language Specification, Version l.3 First Edition: March 2000
5. Haikuan Li , Jan van Katwijk , A.M.Levy "The Reuse of Software Design and Software Architecture" IEEE 1992
6. Istvan Gergely Czibula , Gabriela Serban "Improving Systems Design Using a Clustering Approach" IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.12, December 2006
7. Jan Schulz , "Jacard similarity" http://code10.net/index.php?option=com_content&view=article&id=60:articlejaccard-similarity&catid=38:cat_coding_algorithms_data-similarity&Itemid=57
8. R.Ibba , D. Natale , P.Benedusi and R.Naddei "Structure-based Clustering of Components for Software Reuse" IEEE 1993
9. Rational Rose, 2008, http://www.rational.com.

10. Rosziati Ibrahim and Noraini Ibrahim "A tool for checking the conformance of UML specification", www.waset.org/journals/waset/v51 /v51 -45.pdf.
11. Reuse based Software Engineering, Book by Hafedh Mili, Ali Mili, Sherif Yacoub, Edward Eddy published by John Wiley & Sons, Inc.,2002.
12. Shi Zhong , Joydeep Ghosh "A Unified Framework for Model-based Clustering" Journal of Machine Learning Research 4 (2003) 1001-1037
13. Xie Binhong, Ren Yaopeng, Zhang Yingjun ,Chen Lichao "Research on the Clustering Algorithm of Component Based on the Grade Strategy" 20IO International Conference on Computer Application and System Modeling (ICCASM 2010)
14. Yves Chiricota , Fabien Jourdan, Guy Melancon "Software components capture using graph clustering" Proceedings of the 11 th IEEE International Workshop on Program Comprehension (IWPC'03)