Agglomerative Hierarchical Approach for Clustering Components of Similar Reusability

Aman Jatain
Assistant Professor
(CSE & IT Department)
ITM UNIVERSITY, Gurgaon

Arpita Nagpal
Research Scholar
(CSE & IT Department)
ITM UNIVERSITY, Gurgaon)

Deepti Gaur, PhD.
Associate Professor
(CSE & IT Department)
ITM UNIVERSITY, Gurgaon

ABSTRACT

This paper presents a clustering approach for grouping components of similar reusability using an already worked out fuzzy data set [2]. Research has shown that, component based systems development concept benefits the object oriented software development. A Component based system achieves flexibility by clearly separating the stable parts of systems from the specification of their composition. Many software systems contain many similar or even identical components and these components are developed from scratch over and over again which require extra effort. So to minimize the extra effort in developing these components, it is more beneficial to reuse the existing components. To reuse components effectively in Component Based Software Development, it is required to quantify the reusability of components. However it is difficult to use clustering approach to predict reusability. This paper discusses a technique to cluster components of similar reusability together for the purpose of minimizing the efforts of the developer using agglomerative hierarchical clustering. Components attribute affecting the reusability are classified into rules using fuzzy system and are then taken as the inputs to the proposed clustering model.

Keywords

Component, Component based Software engineering, Fuzzy, Clustering, Hierarchical, Agglomerative, and Reusability.

1. INTRODUCTION

A cluster analysis acts a big job in software development. Cluster analysis is the proposal for sorting out data into clusters or groups in a situation where no prior information about a structure is vacant [18]. It divides data into groups (clusters) that are meaningful, useful or both. The clustering approach is a key gadget in decision making and an effective inspiration method in generating ideas and obtaining solutions [17]. Clustering is mainly used to discover patterns in the data but from literature review we have seen that it has limited application in component based system. Component based is accepted in the industry for reducing development cost and increasing the reliability of entire system. Efficient Reusability, flexibility, higher productivity and scalability are some additional advantages of CBSD [3]. Software Reusability is an attribute that refers to the expected reuse potential of a software component [3]. It is quite difficult to measure reusability directly because reusability is affected by many other factors and there is no direct method to weigh this quality attribute. The cluster analysis approach is an important tool in decision making and an effective creativity technique in generating ideas and obtaining solutions. The software module clustering problem consists of automatically finding a good quality clustering of software modules based on the relationships among the modules [20]. These relationships typically take the form of dependencies between modules. Research in past has shown that many researcher have proposed various methodologies for assessment of the reusability of components. In this paper we have implemented the agglomerative hierarchical algorithm.

2. RELATED WORK

After study of various papers, books and discussing with researcher in the field of CBSD and clustering technique, we able to define our objective. A brief of papers analysis is given under this section. Shri et al. in [4] have predicted the reusability of software system using hybrid k-means and decision tree approach. The reusability value will be Nill or Excellent to separate different components of software module. Sembiring et al. [26] discussed that there are various clustering methodology in order to mine the data. Sonia et al. [25] have proposed a framework for evaluating the reusability using metrics based approach. In [6] data mining technique is applied to extract useful knowledge from software repository using different software metrics. Washizaki [12] presented a metric suite for measuring reusability of software components. Chen and Wang [8] described the use of clustering methods on fuzzy numbers similarity measure for constructing hierarchical groups of mutually exclusive subsets on the basis of their similarity with respect to specified characteristics. Jain [5] discussed that K-means is the most widely used algorithm for clustering even after 50 years of its discovery. The other new algorithms are just the modification of K-means. It is one of the partition clustering algorithm. ROCK is one of the agglomerative hierarchical clustering algorithm. It is used for categorical data set and does not use any distance function. It is based on the number of links between two records. Lopez et al. [23] proposed a technique for analysis and clustering of reusable requirements. The analysis and clustering of requirements are automatically made with their prototype system for requirement reuse.

3. PROPOSED METHOD

Reusability evaluation system for software component is framed using fuzzy rules which are constructed for identified components attributes and then agglomerative hierarchical clustering is implemented on these fuzzy rules to show that hierarchical clustering provide the appropriate result to predict reusability. In order to find our research objective steps followed are as follows:

3.1 Identification of components attributes that affects the reusability

The effort required for reuse of existing software depends on several factors that must be well understood before a determination to reuse functionality is made [9]. Following factors have been identified, which will influence reusability of component [2]:

a) Customizability
b) Configurability
c) Interface Complexity
d) Portability
e) Compatibility

3.2 Formation of fuzzy rules using mamdani style fuzzy model

There are five inputs to this fuzzy process, namely Customizability, Configurability, Interface Complexity, Portability, and Compatibility. Figure 1 shows the fuzzy system:

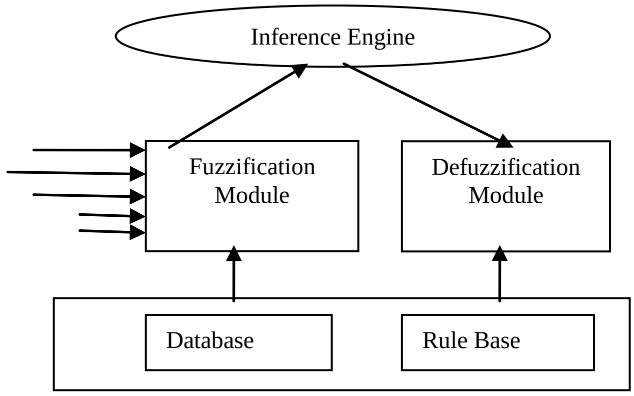

Fig 1: **Fuzzy System**

This system considers all five inputs and provides a crisp value of Reusability using the Rule base. All inputs can be classified into fuzzy sets viz Low, Medium and high. The output estimated Reusability is classified as Very High, High, Medium, Low and very low. In order to fuzzify the inputs, the following membership functions are chosen namely: low, Medium and High.

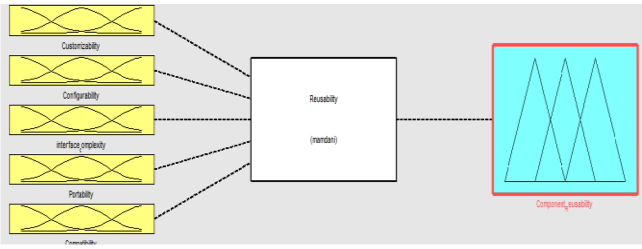

Fig 2: **Inputs and Outputs in Fuzzy System**

Similarly the output variable i.e estimated reusability has five membership functions. All the inputs and outputs are fuzzified as shown in figure3. All possible combinations of inputs are considered which leads to 3^5 i.e 243 sets. Reusability estimated in case of all 243 combinations is classified as Very high, high, medium, Low, Very Low by expert opinion. This lead to the formation of 243 rules for the fuzzy system and some of them are shown below:

1. If Customizability is low, Configurability is low, Interface complexity is low, Portability is low, and Compatibility is low then Reusability is low.
2. If Customizability is high, Configurability is low, Interface complexity is low, Portability is medium and Compatibility is low then Reusability is medium.
3. If Customizability is medium, Configurability is low, Interface complexity is low, Portability is high and Compatibility is low then Reusability is medium.
4. If Customizability is low, Configurability is low, Interface complexity is medium, Portability is low and Compatibility is low then Reusability is low.
5. If Customizability is high, Configurability is high, Interface complexity is medium, Portability is high and Compatibility is medium then Reusability is high.
6. If Customizability is high, Configurability is high, Interface complexity is low, Portability is high and Compatibility is high then Reusability is very high.

All 243 rules are entered and Rule Base is created, a rule will be fired depending on a particular set of inputs. Mamdani style of inference is used. Some of the rules entered are shown in rule viewer. All the data of the rule viewer is not shown to limit the no. of pages. Table 1 show the system configuration and values of various parameters set of inputs.

Table1. **Parameter values for Inputs**

```
  [Input1]
  Name='Customizability'
  Range=[0 1]
  NumMFs=3
  MF1='low':'trimf',[0 0.2 0.35]
  MF2='medium':'trimf',[0.3 0.5 0.68]
  MF3='high':'trimf',[0.65 0.8 1]

 [Input2]
  Name='Configurability'
  Range=[0 1]
 NumMFs=3
 MF1='low':'trimf',[0 0.2 0.35]
 MF2='medium':'trimf',[0.32 0.5 0.68]
 MF3='high':'trimf',[0.62 0.8 1]

 [Input3]
 Name='interface_complexity'
 Range=[0 1]
 NumMFs=3
 MF1='low':'trimf',[0 0.16 0.35]
 MF2='medium':'trimf',[0.3 0.5 0.68]
 MF3='high':'trimf',[0.62 0.85 1]

 [Input4] Name='Portability'
 Range=[0 1]
 NumMFs=3
 MF1='low':'trimf',[0 0.16 0.35]
 MF2='medium':'trimf',[0.3 0.53 0.68]
 MF3='high':'trimf',[0.63 0.8 1]

 [Input5] Name='Compatibility'
 Range=[0 1]
 NumMFs=3
 MF1='low':'trimf',[0 0.15 0.35]
 MF2='medium':'trimf',[0.320.55 0.66]
 MF3='high':'trimf',[0.63 0.76 1]
```

Out of 243 rules, six rules mentioned above are taken as data set to hierarchical clustering algorithm. Linguistic terms in the rules are replaced by the numerical values in table1.

3.3 Implementation of Hierarchical Clustering on fuzzy Data.

In this section hierarchical clustering approach is used on the fuzzy data in which five linguistic terms are to be grouped on basis of reusability. All these 5 components which influence reusability (Customizability, Configurability, Interface, Complexity, Portability, and Compatibility) are described by 243 rules in fuzzy system. Here six of these rules are clustered using an agglomerative hierarchical clustering technique.

Step 1: The input data for 6 rules is:

1. [0; 0.2; 0.35; 0.2; 0.2]
2. [0.65; 0.2; 0.8; 0.68; 0.2]
3. [0.65; 0.2; 0.8; 1; 0.35]
4. [0; 0.2; 0.5; 0.35; 0.2]
5. [1; 0.65; 0.8; 0.65; 0.5]
6. [0.65; 0.8; 0.2; 0.65; 0.65]

Step 2: Find the similarity between every pair of objects in the data set that is you calculate the Euclidean distance between object. The dissimilarity matrix is given in table 2.

Table 2.Dissimilarity matrix of the input data

	1	**2**	**3**	**4**	**5**	**6**
1	0	0.9249	1.1347	0.2121	1.3029	1.1000
2	0.9249	0	0.3534	0.7883	0.6449	0.9609
3	1.1347	0.3534	0	0.9785	0.6856	0.9657
4	0.2121	0.7883	0.9785	0	1.2135	1.0794
5	1.3029	0.6449	0.6856	1.2135	0	0.7263
6	1.1000	0.9609	0.9657	1.0794	0.7263	0

Step 3: Here group the pair of objects that are in close proximity using distance information in step2. Two individual points or clusters are grouped into one individual cluster using the linkage function. This linkage function can be one of the following:

Single Linkage: In this it tends to choose minimum distance of the point to group distance.

Complete Linkage: It considers the longest distance from any member of one cluster to any member of other cluster.

Average Linkage: It chooses average distance within the cluster to some other point outside the cluster.

In this single linkage method is used to group the data from the dissimilarity matrix in table 2 and its result is shown in table 3(a):

Table 3 (a)

	1,4	**2**	**3**	**5**	**6**
1,4	0				
2	0.788	0			
3	0.978	0.3534	0		
5	1.213	0.644	0.685	0	
6	1.079	0.960	0.965	0.726	0

Table 3(b)

	1,4	**2,3**	**5**	**6**
1,4	0			
2,3	0.788	0		
5	1.213	0.644	0	
6	1.079	0.960	0.726	0

Table 3 (c)

	1,4	**2,3,5**	**6**
1,4	0		
2,3,5	0.788	0	
6	1.079	0.726	0

Table 3(d)

	1,4	**2,3,5,6**
1,4	0	
2,3,5,6	0.788	0

Step4: The final output in the form of dendogram is shown below:

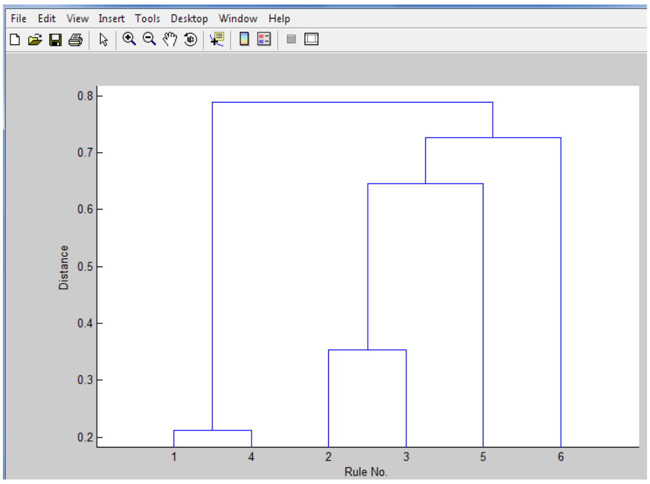

Fig. 3: **Result of dissimilarity matrix in the form of Dendogram**

Step 5: Now, clusters are extracted from this dendrogram on fuzzy data. To do this, inconsistency coefficient has been used. In fig. 4 each dendrogram is labeled by its inconsistency value. The higher the value of each label, least similar is its reusability value is with the other clusters below that link. The inconsistency value is calculated for each cluster by the length of that link in dendogram with mean length of other clusters at the same level of dendogram. To calculate inconsistency coefficient, linkage matrix is needed. This matrix is generated in matlab along with Dendogram. Linkage matrix for data in table 2 is given in table 4.

Table4. Linkage Matrix

1.0000	4.0000	0.2121
2.0000	3.0000	0.3534
2.0000, 3.0000	5.0000	0.6449
2.0000, 3.0000 , 5.0000	6.0000	0.7263
1.0000, 4.0000	2.0000, 3.0000 , 5.0000 ,6.0000	0.7883

The Inconsistency matrix is given in table5. Column 1 gives mean of the lengths of all links till its level. Column 2 gives the standard deviation of all the links till its level. Column 3 shows the number of links. 4th column gives inconsistency coefficient, which is calculated using the formula:

$$W(k,4)=(Z(k,3)-W(k,1))/W(k,2)$$

Where K specifies the level number, W is inconsistency matrix, Z is the linkage matrix in table 5

Table5. Inconsistency Matrix

Mean	Standard Deviation	Number of links	Inconsistency Coefficient
0.2121	0	1	0
0.3534	0	1	0
0.4991	0.2061	2	0.7074
0.5749	0.1961	3	0.7720
0.5449	0.2497	5	0.9736

To understand table 5, values of 3rd row are explained: no. of links are two. Mean is taken from linkage matrix. It is the average of all links till we reach to the 3rd level in the dendogram.

$$\text{Mean} = (0.3534+0.6449)/2$$
$$= 0.4991$$

Also, standard deviation is calculated which is 0.2061. For computing inconsistency coefficient , following formula is used.

$$W(3,4) = (Z(3,3)-W(3,1))/W(3,2)$$
$$= (0.6449-0.4991)/0.2061$$
$$= 0.7074$$

Similarly others values of inconsistency coefficient of column 4 of table 5 are calculated.

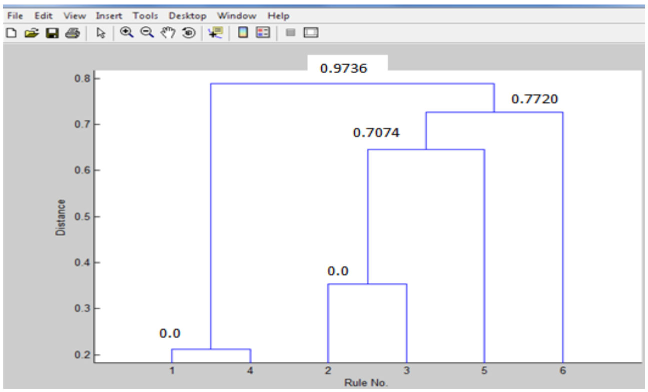

Fig4: Inconsistency coefficient related to each link

In figure 4 , it can be observed that rule 1 and 4 are most similar because of lower value of inconsistency. Rule 1 and 4 with rule 6 is the least similar cluster as it has the highest value. From the 6 fuzzy rules used to predict reusability described in step 2, it can be verified from histogram that rule 1 and 4 are clustered together as both have low reusability. These inconsistency coefficients can be used to define clusters. Suppose if no of clusters defined by the user is three, then three clusters formed out of these six rules used in the dendogram are:

$$C1 = \{1,4\}, C2= \{2,3,5\}, C3=\{6\}$$

Therefore clusters are defined via inconsistency matrix. As shown in section 3.2, 243 rules have been entered in the fuzzy system and reusability value is calculated in three categories (low ,medium and high), but it becomes very difficult if the user want to see which factors affect the reusability (say according to which rule reusability falls in low category) , then he has to scan through the all the rules. But by using clustering approach, all data set can be categorized into clusters of low, medium, high reusability, and users can easily predict the reusability category just by examining the clusters.

4. CONCLUSION

For component-based development, efforts are mainly invested in deciding upto what extent the component can be reused and then integrating it in the application. In the paper [3] a fuzzy rule based approach was proposed for estimating component Reusability. The proposed approach here is used to predict the Reusability using agglomerative hierarchical clustering. By using clustering technique same category data points are clustered together. Experimental results in this paper shows, how rules can be grouped together into clusters of same range of reusability. However, the work further requires validation. For this purpose, validation formulas are being studied and will be implemented on fuzzy data. Hierarchical clustering is a very suitable data because the nature of this algorithm is illustrative and clear.

5. REFERENCES

[1] A. Sharma: “Design and Analysis of Metrics for Component- Based Software Systems”, Ph.D thesis, 2009.

[2] Aman & Dr. Deepti , “Estimation of Component Reusability by Identifying Quality Attributes of Component – A Fuzzy Approach”,

[3] A. Sharma, P.S. Grover, R. Kumar: “Investigation of Reusability, complexity and Customizability Metrics for Component Based Systems”, ICFAI Journal of Information Technology, 2006.

[4] Anju, Parvinder S. Sandhu, Vikas Gupta, Sanyam Anand, “Prediction of reusability of object oriented Software systems using clustering Approach ”, World academy of science, Engineering and Technology , 2013.

[5] Anil K. Jain, “Data clustering: 50 years beyond K-means”, 19th International Conference in Pattern Recognition,2009.

[6] B V Ajay Prakash, D V Ashoka, V N Manjunath Aradhya, “Application of data mining techniques for Software Reuse”, Procedia Technology pg. 384-389, 2012.

[7] Capers Jones, “Software Estimating Rules of Thumb”, http://www.ieeexplore.ieee.org/iell/2/20412/00485905 pdf.

[8] Shi-Jay Chen, Zhi-Yong Wang, “A hierarchical Clustering Method based on Fuzzy number Similarity Measure Applied to aproblem of Grouping Profiles” , International conference on Innovation in Bio- Inspired Computing and Applications, 2012.

[9] Dan Galorath President Galorath Incorporated, “Software Reuse and Commercial Off-the-Shelf Software”, El Segundo, CA.

[10] Fokaefs M., Tsantalis N., Chatzigeorgiou A., Sander J. (2009), “Decomposing Object-Oriented Class Modules Using an Agglomerative Clustering Technique”, IEEE, Proc. ICSM, Canada.

[11] Hafed Mili, Ali Mili and Edward Addy, "Reuse based Software Engineering".

[12] Hironori Washizaki1, Hirokazu Yamamoto2 and Yoshiaki Fukazawa, "A Metrics Suite for Measuring Reusability of Software Components", Department of Computer Science, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan f washi, Fukazawa @fuka.info.waseda.ac.jp 2Matsushita Electric Industrial Co., Ltd. 1006 Kadoma, Kadoma City, Osaka 571-8501, Japan.

[13] Ian Sommerville: "Software Engineering", 7th edition, 2004.

[14] J.G. Schneider: "Component Scripts and Glue: A Conceptual framework for software composition", Ph.D. thesis, Institute for Inforrmatik (IAM), University Bern, Berne, Switzerland 2003.

[15] Jai Bhagwan and Ashish Oberoi, "Software Modules Clustering: An Effective Approach for Reusability", Journal of Information engineering and Applications, Vol-1, No.4, 2011.

[16] Johannes Sametinger, "Software Engineering with Reusable Components", Springer verlag, Berlin Heidelberg, NewYork , London , Paris ,Tokyo ,Hong Kong ,Barcelona.

[17] K.K. Aggarwal, Y.Singh, P.Chandra, M. Puri, " Measurement of Software Maintainability Using a Fuzzy Model", Journal of Computer Sciences, Vol.1,Issue 3, pp:538-542, 2005.

[18] Kanungo T., Mount D. M., Netanyahu N. S., Piatko C. D., Silverman Wu A. Y. (2002), "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, Issue 7.

[19] M. Sparling: "Lessons Learned through Six Years of Component Based Development", Communications of the ACM, 2003.

[20] Mahdavi K., Harman M., Hierons R. M. (2003), "A Multiple Hill Climbing Approach to Software Module Clustering", Proceedings of the International Conference on Software Maintenance, DISC Brunel University

[21] N.J.Piscataway, IEEE 1517, "Introduction to IEEE Std. 1517 –Software reuse Processes", IEEE, 1999.

[22] N.S. Gill: "Importance of Software Component Characterization for Better Software Reusability", ACM SIGSOFT SEN, Vol. 31, Issue 1, pp:1-3, 2006.

[23] Oscar L´opez, Miguel A. Laguna, and Francisco J. Garc´ıa, "Reuse based Analysis and Clustering of Requirements Diagrams", 2004.

[24] Roger Jang and Ned Gulley, "Fuzzy Logic Toolbox for MATLAB. User's Guide", The Math Works Inc, USA, 1995.

[25] Sonia Manhas, Rajeev Vashisht, Reeta Bhardwaj, "Framework for Evaluating Reusability of Procedure Oriented System using Metrics based Approach", Internationall Journal of Computer Application 2010.

[26] Sembiring R. W., Zain J. M., Embong A. (2010), "A Comparative Agglomerative Hierarchical Clustering Method to Cluster Implemented Course", Journal of Computing, Vol. 2, Issue 12, PP. 1-4.