

# Classification of Reusable Components Based on Clustering

**Muhammad Husnain Zafar**

Dept. of Computer Science and Information Technology, University of Sargodha, Sargodha, Punjab, Pakistan  
E-mail: husnainzafarmscs@gmail.com

**Rabia Aslam**

Dept. of Computer Science and Information Technology, University of Sargodha, Sargodha, Punjab, Pakistan  
E-mail: aizelsajday@gmail.com

**Muhammad Ilyas**

Dept. of Computer Science and Information Technology, University of Sargodha, Sargodha, Punjab, Pakistan  
E-mail: m.ilyas@uos.edu.pk

**Abstract**—Software reuse is the process of implementing or updating software systems using existing software components. A good software reuse process facilitates the increase of productivity, quality and reliability. It decreases the cost and implementation time as compared to develop new system. Despite its many benefits we cannot achieve its full benefits. The reason behind this is that software reuse is often done in an informal and haphazard way. If done systematically, then we can achieve its full benefits. This research proposes a method through which we will classify the reusable components in proper way to get the full benefits of reusability. We classify the reusable components according to their clusters. Clusters are made on the basis of parameters provided with components. We develop an algorithm for assigning clusters to the reusable components.

**Index Terms**—Software Components, Software Clustering, Software Reusability, Software Classification.

## I. INTRODUCTION

Due to the benefits of reusability like, reduction in time, cost and work as compared to develop a new system, it has turn into useful tool for software development [1, 2, 3]. If we apply reusability into software components then reusable components are maintaining easily and usually have a higher quality value [4]. In order to make reuse an effective methodology, it is necessary that the designer must know how to retrieve the suitable solution, how to adopt that solution to fulfill the requirement of new problem and how to evaluate the result of that solution [5]. That [5] is the biggest problem in many organizations. To overcome this weakness, there is a need to organize and classify collection of components so that we can identify the suitable component for potential reuse [6]. This solution provides an aid to the software developer [7].

This research proposes a method through which we classify the reusable components in proper way. We

classify the reusable components according to their clusters. 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 dissimilarity or distance [8]. A cluster analysis acts a big job in software alliance. 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 [9]. It divides data into groups (clusters) that are meaningful, useful or both [10]. The clustering approach is a key gadget in decision making and an effective inspiration method in generating ideas and obtaining solutions. Clusters combine the similar units into one cluster and clusters should be different as possible. If there exist high similarity in intra-cluster and low similarity in inter-cluster, the quality of clustering becomes high [11]. For solving classified problem, clustering is an uncertain data analysis tool. Its objective is to sort cases into groups or clusters, so that the degree of friendship is strong between members of the same cluster and weak between members of different clusters [8].

Rest of the paper compromise the following sections. Section II describes related work. Proposed framework and algorithms are given in Section III and Section IV respectively. Results and discussions are explained in Section V. Section VI compromises the conclusion and future work.

## II. RELATED WORK

Reusability becomes very effective elements of software development because it gives many benefits to the developer like less time, less effort and less cost as compared to develop new component [15]. It also reduces the risk that is related to software development. Instead of its many advantages, there is a lack of symmetric way to use the reusable components in proper way. This paper tries eliminating this flaw. M. Ilyas et al [15] gives a frame work that gives a systematic way to

use the reusable components and enact the reusability process to get good results.

J. Bhagwan and A. Oberior [8] stated that, in present age, software evolution is becoming greater in size with the passage of time however the problem is that, for handling this evolution, manpower is not increasing as evolution. This causes software failure. One of the best solutions is Reusability. To overcome the problem of software crisis, many of the professionals suggest that reuse is a powerful asset. In this research clustering of software modules is used for obtaining full benefits from the reusability. On the basis of association between modules, a good quality software module is selected automatically. Association is usually dependencies among modules. LOC and Code Clones are used for finding relationship between modules. To meet the research objective, this paper uses the HC algorithm (Agglomerative method) [12] and NHC algorithm (K-Mean method). For this purpose this research proposes an algorithm.

S. K. Soora [5] classifies the reusable component on the basis of semantic features. In his research he discussed the current methodology for classification of reusable component. According to him, currently classification is based on the use of formal specification. The author describes that there are many benefits of formal specification. But when we retrieve a component, we use theorem prover as retrieval mechanism. This is very costly that reduce the scalability of this approach. So the author proposes classification of components on the basis of semantic features. A set of semantic features are

used to classify components. These sets are generated from the formal specification of component. Theorem prover generates the feature set by using limited forward inference. This methodology is scalable because of the limited set of components over which theorem proving is applied to determine reusability.

Manhas et al. [13] stated that if we use already existing components then we can reduce the cost of software development. He uses many metrics for the classification of software component. To evaluate the result, he uses Back propagation based neural networks.

Shri et al [14] propose hybrid k-mean and decision tree approach. The author uses these approaches to achieve the reusability value of object oriented components. The authors proposed an algorithm in which the data input is specified to k-mean clustering. This input is in the form of adjusted values of the Object 7Oriented software components. A decision tree system is developed in order to achieve similar type of software entities. Metrics are used to forecast the quality of software components.

### III. PROPOSED METHODOLOGY

For the systematic clustering of reusable components we have proposed a framework named CORCs. CORCs stand for “Clustering of Reusable Components”. This framework has three process components e.g. Threshold Checking, Clustering Process and Retrieval Process. Fig. 1 shows the CORCs framework. Details of CORCs components are given as below.

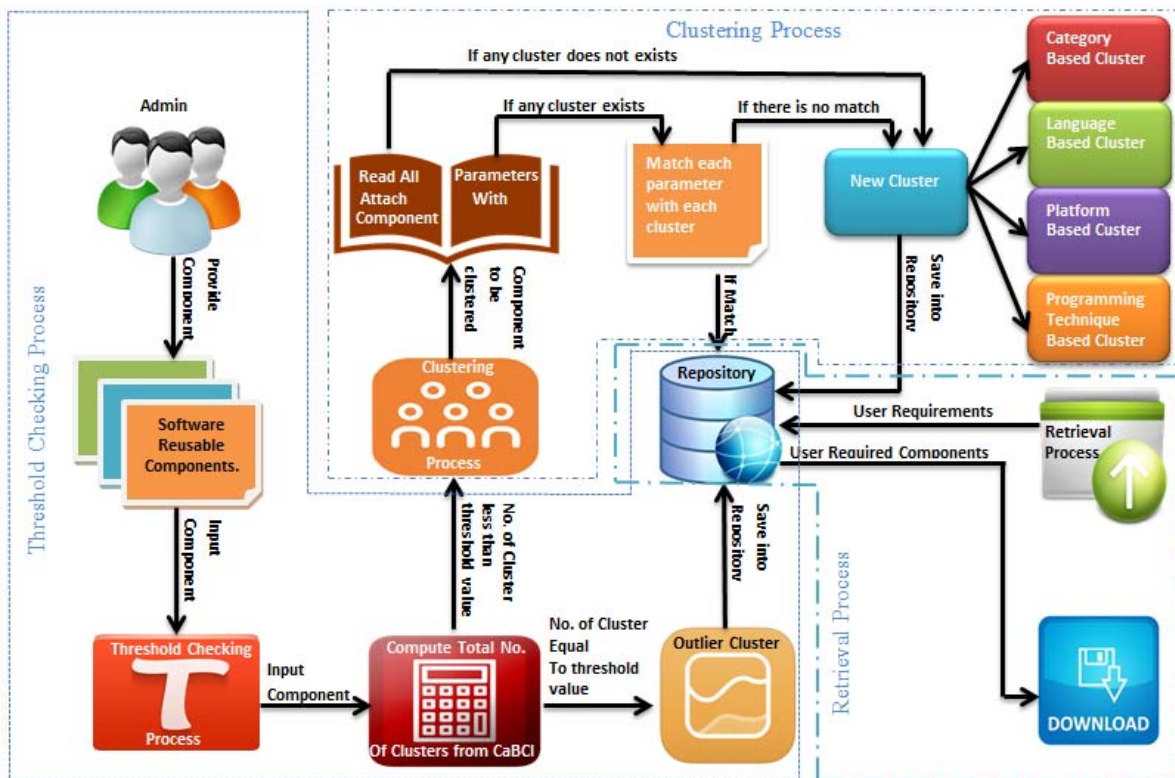


Fig.1. CORC Framework

### A. Threshold Process

Threshold checking is applied only at Category Based Cluster. In this process a value has been calculated by counting the number of already existing clusters in Category Based Cluster and then we compare this value to the given threshold value. If this value is equal to threshold value then the component goes into a cluster named outlier. Outlier is a cluster in which all those components that are not similar to any component. If calculated value is less than threshold value than the component goes into next process that is clustering process. Outlier cluster also saves into repository.

### B. Clustering Process

This is the main process of CORCs framework. In this process matching criteria is applied to the component. Each component goes into four types of clusters. These are given below.

#### 1) Category Based Cluster

It is the main objective of our research to cluster the components on the basis of their functionalities. Here the category means functionality that a specific component performs. Firstly we tried to determine the functionality by reading the component, components mean a source code that performs a specific functionality, but it is almost impossible to determine the functionality of a component by reading the source code. It is due to the development of one component in multiple ways. Every programmer has its own way of development and has its own logic. So it is very difficult to govern the functionality by reading source code and hence it is decided that this feature has been provided with the source code and by reading this feature, component goes into a clusters that is called Category Based Cluster. Abstract data types, Artificial Intelligence, Algorithms, Games, Graphics and Utilities are some of the example of Category Based Cluster. Because there are many categories that can be build, so we apply a threshold to lemmatize the maximum number of categories. This threshold is provided by the user according to their needs.

#### 2) Language Based Cluster

It is another type of cluster. Formation of this cluster is based on the programming language in which a source code has been developed. Programming Language of component has been provided with source code as another feature or parameter of source code, although it is not so much difficult to determine the language by reading the source code. Language of source code has been detected by reading the syntax of the language. Now a day, there are limited languages in which programming is done, so there is no need to put a threshold on this type of cluster.

#### 3) Platform Based Cluster

Another type of cluster is Platform Based Cluster. Platform means the operating system for which the component has been developed. This feature has also been provided with the source code. There is no need to

put threshold value as there is limited number of platforms. Windows, UNIX, Linux, Androids are the some examples of platform.

#### 4) Programming Technique Based Cluster

It is the last type of cluster that has been formed on the basis of programming technique in which component has been developed. Here we decide three types of techniques. These are Simple, Structured and OOP. This feature also provided with the source code.

##### a) Simple Technique

In this technique, the component is neither developed in structured approach nor by using OOP concepts. In the type components are developed by using simple functions, loops or by using switch statements.

##### b) Structured Technique

In this type component has been developed by using structured approach. This gives many benefits like progress in quality, clarity and development time of component. In this block structure, subroutines, for and while loops are used extensively instead of using simply jump statements or simple tests that leads to a complex structure and complicated control.

##### c) OOP Technique

In this type component has been developed by using the concepts of Object Oriented Programming. Inheritance, Polymorphism and Encapsulation are included in Object Oriented Programming. There are lots of benefits of this technique. These benefits include reusability (easily modify already existing components), data hiding, better design, easy maintenance of software, proper program structure and provide a very suitable memory management. Here it very important to announce that each component goes into these four types of clusters.

#### 5) Matching Criteria

For making these four types of clusters four equations are given. Equation 1, 2, 3 and 4 shows the matching criteria for Category, Language, Platform and Programming Technique Based Clusters respectively.

$$C_aBC_{l_{new}} = \begin{cases} 0 & \text{when } (C_aIC_o) = \sum_{i=1}^n (C_{a_{C_aBC_l}})_i \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

$C_aBC_l$  is Category Based Cluster.  $C_aIC_o$  is category parameter of input component.  $n$  is the total number of clusters that exists in the type of Category Based Cluster.  $C_{a_{C_aBC_l}}$  is category parameter of components that already exists and belongs to Category Based Cluster. Here the category parameter of input component matches with one component of all clusters of Category Based Cluster. If any match exists then this equation return 0 and input component goes into that cluster with which category parameter of input component matches and if there is no

match then this equation return 1 and a new cluster is formed that is Category Based Cluster and input component goes into this new cluster.

$$LBC_{l_{new}} = \begin{cases} 0 & \text{when } (LIC_o == \sum_{i=1}^n (L_{LBC_l})i) \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

$LBC_l$  is Language Based Cluster.  $LIC_o$  is language parameter of input component.  $n$  is the total number of clusters that exists in the type of Language Based Cluster.  $L_{LBC_l}$  is language parameter of components that already exists and belongs to Language Based Cluster. Here the Language parameter of input component matches with one component of all clusters of Language Based Cluster. If any match exists then this equation return 0 and input component goes into that cluster with which language parameter of input component matches and if there is no match then this equation return 1 and a new cluster is formed that is Language Based Cluster and input component belongs to this new cluster.

$$PBC_{l_{new}} = \begin{cases} 0 & \text{when } (PIC_o == \sum_{i=1}^n (P_{PBC_l})i) \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

$PBC_l$  is Platform Based Cluster.  $PIC_o$  is platform parameter of input component.  $n$  is the total number of clusters that exists in the type of Platform Based Cluster.  $P_{PBC_l}$  is platform parameter of components that already exists and belongs to Platform Based Cluster. Here the platform parameter of input component matches with one component of all clusters of Platform Based Cluster. If any match exists then this equation return 0 and input component goes into that cluster with which platform parameter of input component matches and if there is no match then this equation return 1 and a new cluster is formed that is Platform Based Cluster. In this new cluster, the input component will go.

$$P_tBC_{l_{new}} = \begin{cases} 0 & \text{when } (P_tIC_o == \sum_{i=1}^n (P_{tP_tBC_l})i) \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

$P_tBC_l$  is Programming Technique Based Cluster.  $P_tIC_o$  is programming technique parameter of input component.  $n$  is the total number of clusters that exists in the type of Programming Technique Based Cluster.  $P_{tP_tBC_l}$  is programming technique parameter of components that already exists and belongs to Programming Technique Based Cluster. Here the programming technique parameter of input component matches with one component of all clusters of Programming Technique Based Cluster. If any match exists then the value of this equation is 0 and input component goes into that cluster with which programming technique parameter of input component matches and if there is no match then 1 is the value of this equation and a new cluster is formed that is Programming Technique Based Cluster. The input component becomes the member of this new cluster.

### C. Retrieval Process

This is the main process at the user end. In this process the user input his/her requirements and according to their requirements one or more components, that meet the criteria of user, is available to download. These user's requirements are according to the parameters that are attached with source code. These parameters are Category, Programming Language, Platform and Programming Technique of components. Here we give the five options to the user to retrieve components in five different ways by choosing the parameters in five different ways.

#### 1) Algorithms for Retrieval Process

Five different algorithms are proposed for retrieval process. These are describes in the following section. In these algorithms  $C_a$  is for Category.  $L$  is used for language;  $P$  is used for Platform and  $P_t$  is used for programing technique.

```
Clustering_algorithm_Retrieval ( $C_a, L, P, P_t$ )
{
     $C_s = \text{Components}$ 
     $N = \text{Number of Clusters}$ 
    while( $N \neq 0$ )
    {
```

```
         $C_s = \text{Match } C_a, L, P, P_t \text{ with existing Components}$ 
        Decrement value of  $N$ 
    }
    Display  $C_s$ 
    Download required component/components
}
```

In the above algorithm user provides all four parameters according to their needs. Our algorithm as a result provides one or more components that are according to the requirements of user. Here user can download his required component.

```
Clustering_algorithm_Retrieval ( $C_a$ )
{
     $C_s = \text{Components}$ 
     $N = \text{Number of Clusters}$ 
    while( $N \neq 0$ )
    {
         $C_s = \text{Match } C_a \text{ with existing Components}$ 
        Decrement value of  $N$ 
    }
    Display  $C_s$ 
    Download required component/components
}
```

In the above algorithm user only give one parameter that is category. User queries that which category of component he/she required. This algorithm provides all the components those belong to that category.

```
Clustering_algorithm_Retrieval ( $L$ )
```

```

{
  Cs = Components
  N = Number of Clusters
  while(N != 0)
  {
    Cs = Match L with existing Components
    Decrement value of N
  }
  Display Cs
  Download required component/components
}

```

In the above algorithm user only give one parameter that is language. Components developed in different languages are available so a query has been made from the user that in which programming language he/she required the components. After querying this, algorithm provides all the components those belong to that programming language.

```

Clustering_algorithm_Retrieval (P)
{
  Cs = Components
  N = Number of Clusters
  while(N != 0)
  {
    Cs = Match P with existing Components
    Decrement value of N
  }
  Display Cs
  Download required component/components
}

```

In the above algorithm user only give one parameter that is platform at which component will run. User queries that he/she required the components that runs on windows or other platform. After querying this, algorithm provides all the components those belong to that platform.

Algorithm that is given below show that user only give one parameter that is programming technique in which component is developed. User queries that he/she required the components that are developed in either structured approach or using OOP concepts or developed by using simple technique. After querying this, algorithm provides all the components those belong to that programming technique.

```

Clustering_algorithm_Retrieval (Pt)
{
  Cs = Components
  N = Number of Clusters
  while(N != 0)
  {
    Cs = Match Pt with existing Components
    Decrement value of N
  }
  Display Cs
  Download required component/components
}

```

#### IV. PROPOSED ALGORITHM

Before discussing proposed algorithm first we like to discuss input for proposed algorithm.

##### A. Input for Proposed Algorithm

Input for our algorithm is a component. The extension of input is txt. In this file both source code and parameters associated with source code are present. Internal structure of input file is given below. Beginning of this file contains parameters and rest of the file contains source code.

##### Parameters

Language:	C++
Name:	Overloadingexample
Category:	C++ Features
Sub Category (If applicable):	NA
Technique:	Simple
Platform:	Windows
Operating System Independence:	No
Other Information:	Demonstrates function overloading

##### Source Code

```

#include<iostream>
using namespace std;
void func1(char* str);
void func2(int val);
int main()
{
    func1("character string passed\n");
    func1(123);
    return(system("pause"));
}
void func1(char* str)
{
    cout<<"string value:="<<str<<endl;
}
void func1(int val)
{
    cout<<"integer value :="<<val<<endl;
}

```

##### B. Clustering Algorithm

Clustering algorithm is given below.

1. Numbers of already existing Category Based Clusters are computed and compare this value with threshold value.
2. If this value is equal to threshold value than component will goes into outlier cluster?
3. If this value is less than threshold value than component will ready for clustering process
4. In clustering process if any cluster don't exist then first component make four types of clusters by reading four parameters (Category, Language, Platform, Programming Technique). These clusters are Category Based, Language Based, Platform Based and Programming Technique Based Cluster.

5. If any cluster exists then read four parameters (Category, Language, Platform, and Programming Technique) that are attached with source code and match these parameters with each cluster of four types of clusters that mentioned in step 4.
6. If match to any cluster then component will goes into that component's cluster
7. If not match to any component then a new cluster will form and component will goes into that cluster.

Pseudo code of proposed algorithm is given below.  
Begin

```

j = total number of clusters in category based cluster
k = total number of clusters in language based cluster
p = total number of clusters in platform based cluster
m = total number of clusters in programming
technique based cluster
t = threshold value defined by user
if j <= t
    Store input component into outlier cluster
else
    for 1 to j
        if  $C_aIC_o == C_{aC_aBC_l}$ 
            Input component goes into that  $C_aBC_l$ 
        else
            new cluster is formed in  $C_aBC_l$ 
    for 1 to k
        if  $LIC_o == L_{LBC_l}$ 
            Input component goes into that  $LBC_l$ 
        else
            new cluster is formed in  $LBC_l$ 
    for 1 to p
        if  $PIC_o == P_{PBC_l}$ 
            Input component goes into that  $PBC_l$ 
        else
            new cluster is formed in  $PBC_l$ 
    for 1 to m
        if  $P_tIC_o == P_{tP_tBC_l}$ 
            Input component goes into that  $P_tBC_l$ 
        else
            new cluster is formed in  $P_tBC_l$ 
End

```

## V. RESULTS AND DISCUSSION

For evaluating algorithm and obtaining results we developed an application named as CORCs (developed in java). We collected 36 software components from different websites. After getting results from prototype, we transformed these results into graph for better understandings of results. Bar chart was used to exhibits the results in the form of graph.d

### A. Results Visualization through Graph

As our framework shows that we made four types of

clusters. So we discuss the results of these four types of cluster one by one. For all these types of cluster total number of software components are 36.

#### 1) Category Based Cluster

In (section III.B.1) we described that only in category based cluster we apply threshold value to lemmatize total number of categories. So here we set threshold value 6. Fig. 2 shows the results of clustering. In this graph black color demonstrates the result of Category based cluster. As we set our threshold value to 6, so we got 6 clusters. These were Utilities, C++ Features, Full Project, Algorithm, Graphics and Game. The numbers shows how many components exist in a specific cluster. For example number 5 in a black column above utilities means, there were five components in cluster of Utilities.

In this type, as we discussed earlier that we apply threshold value, we have some components that did not belong to any category. So they put into an outlier cluster. In Fig. 2, red column denotes outlier cluster which shows that one component of ADT, one component of Security and one component of AI belonged to outlier clusters.

#### 2) Language Based Cluster

It is the second type of cluster that we made. In this type clusters were made on the basis of programming languages in which components had been developed. In Fig. 2 green color is used to represent Language Based Cluster. According to this Fig., we had four clusters in which those components exist that had been developed in four different languages. These clusters were C, Java, Julia and C++. The numerical value in the green column indicated the number of components in each cluster.

#### 3) Platform Based Cluster

Third type of cluster is platform based cluster. These are the clusters that were made on the basis of platform at which components had to be executed. In Fig. 2, purple color belongs to Platform Based Cluster. Fig. 2 demonstrates that we had three clusters in this type. One cluster was that in which those components exists that were compatible with windows platform. In second clusters those components were present that were executable at windows, Linux and UNIX. In third cluster one component was present that was compatible with windows, OS X and Linux. The numbers in purple columns are used to represent number of components in a specific cluster.

#### 4) Programming Technique Based Cluster

It is the last type of cluster. In section (III.B.4), we shared that we categorized programming techniques into three types. So in this type we had maximum three clusters. These were simple, structured and OOP. In Fig. 2 blue color exhibits this type of cluster. The function of numeric values in blue columns are the same as in other columns that is to represents number of components in a specific cluster.

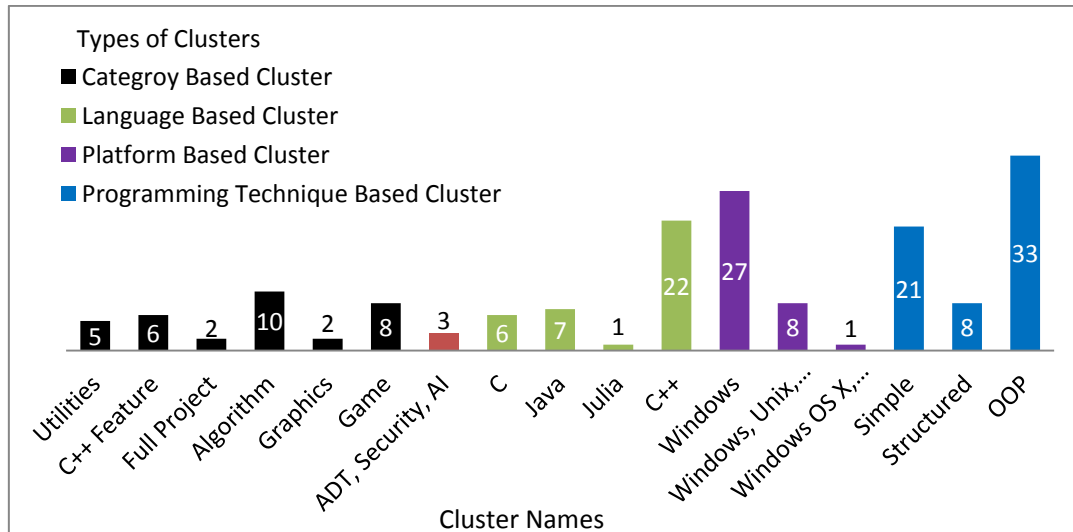


Fig.2. Graphical Results of Clustering

## VI. CONCLUSION AND FUTURE WORK

Reusability has a great importance in software engineering and its usage in software engineering is increasing day by day. But the problem is that there is no proper way to extract reusable components or objects. To solve this issue component must be classified in an efficient way. Classification is done in many ways but clustering is very effective technique for classification. So we propose a framework to understand the process of clustering. This framework consists of three processes that are threshold checking, clustering and retrieval process. In threshold checking process threshold value is applied on at category based cluster to lemmatize the maximum number of clusters in category based cluster. By following this framework we propose a clustering algorithm and for implementing and evaluating proposed algorithm we develop prototype named as CORCs Application. This application is developed in java. We get 36 software components from different website and get our results in the form of graphs.

Our future aimed to generalize proposed clustering process. In proposed framework software components are clustered with specific format so later on we want to cluster software components without specific format and parameters for clustering software components extract at runtime. Another aim to analyze proposed algorithm is by making simulation in MATLAB and do statistical analysis.

## REFERENCES

- [1] B. AL-Badareen, M. H. Selamat and M. A. Jabar, "Reusable Software Component Life Cycle" *Int. J. of Computers*, Vol.5, pp. 191-199, 2011.
- [2] G. Singaravel, V. Palanisamy and A. krishnan, "Overview Analysis of Reusability Metrics in Software Development for Risk Reduction" *Int. Conf. on Innovative Computing Technologies (ICICTI)*, February 2010, Tamil Nadu, India. DOI: 10.1109/ICINNOVCT.2010.5440081.
- [3] R. Kamalraj, B. G. Geetha and G. Singaravel, "Reducing Efforts on Software Project Management using Software Package Reusability" *IEEE Int. Conf. on Advance Computing*, March 2009, pp.1624-1627. DOI: 10.1109/IADCC.2009.4809260
- [4] R. D. Kuhns, 1998. "Strategies for Designing and Building Reusable GIS Application Components" <http://proceedings.esri.com/library/userconf/proc98/proceed/to600/pap557/p557.htm>
- [5] S. K. Soora, "Feature Based Classification & Retrieval of Reusable Components". *Int. J. on Advanced Computer Theory and Engineering (IJACTE)*. Vol. 1, 2012.
- [6] G. Jones and R. Prieto-Diaz, "Building and Managing Software Libraries", *IEEE Int. Conf. on Computer Software and Applications*. October 1998, pp.228-236. DOI: 10.1109/CMPSAC.1988.17177
- [7] P. Niranjana, Dr. C.V.Guru Rao, "A MOCK- UP TOOL FOR SOFTWARE COMPONENT REUSE REPOSITORY" *Int. J. on Advanced Computer Theory and Engineering*, Vol. 1, April 2010. DOI: 10.5121/ijsea.2010.1201
- [8] J. Bhagwan and A. Oberoi, "Software Modules Clustering: An Effective Approach for Reusability", *J. of Information Engineering and Applications*, Vol 1, No.4, 2011.
- [9] K. Tapas., D.M. mount, N.S. Netanyahu, .C.D. Piatko, R. Silvermand and A.Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, Issue 7, pp. 881-892, 2002. DOI: 10.1109/TPAMI.2002.1017616
- [10] M. Fokaefs, N. Tsantalis, A. Chatzigeorgiou, J. Sander, "Decomposing Object-Oriented Class Modules Using an Agglomerative Clustering Technique", *IEEE Int. Conf. on Software Maintenance*, September 2009, pp. 93-101. DOI: 10.1109/ICSM.2009.5306332.
- [11] M. Abubaker and W. Ashour, "Efficient Data Clustering Algorithms: Improvements over Kmeans," *Int. J. of Intelligent Systems and Applications*, Vol. 3, pp. 37-49, 2013. DOI: 10.5815/ijisa.2013.03.04
- [12] R. Kamalraj, A. R. Kannan and P. Ranjani, "Stability-based Component Clustering for Designing Software Reuse Repository," *Int. J. of Computer Applications*, vol. 27, no. 3, pp. 33-36, Aug. 2011.
- [13] M. Sonia., S.S. Parvinder, C. Vinay and N. Nirvair "Identification of Reusable Software Modules in Function Oriented Software System using Neural Network Based Technique", *World Academy of Science, Engineering and Technology*, Vol. 67, pp. 823-827, 2010.

- [14] A. Shri., S.S. Parvinder, G. Vikas and A. Sanyam, "Prediction of Reusability of Objected Oriented Software System using Clustering Approach", *World Academy of Science, Engineering and Technology*, Vol. 67, PP. 853-856, 2010.
- [15] M. Ilyas, M. Abbas and K. Saleem, "A Metric Based Approach to Extract, Store and Deploy Software Reusable Components Effectively", *Int. J. of Computer Science*, vol. 10, issue 4, no. 2, July 2013.

### Authors' Profiles



**Muhammad Husnain Zafar**, has completed his Master of Science in Computer Sciences at the University of Sargodha. He has already completed his BS 4 year degree in computer sciences from the same university. His research interests include Software Engineering, Software Reusability and other

topics.

**Rabia Aslam** is the student of Bachelor of Science in Software Engineering at University of Sargodha. She is also a web designer and developer. Her research interests include Software Reusability, clustering of dynamic content, semantic based clustering, and machine learning algorithms.



**Muhammad Ilyas**, received a Master degree in Software Project Management in 2004 from National University of Computer and Emerging Sciences, Lahore and a Doctor of Informatics from Johannes Kepler University, Linz Austria in 2010. His research interests include Software

Engineering, Design Pattern and knowledge base systems. He is currently an assistant professor in the Department of Computer Science and Information Technology at the University of Sargodha, Pakistan.