K-Means Clustering of Use-Cases Using MDL

Sunil Kumar[1], Rajesh Kumar Bhatia[2], and Rajesh Kumar[3]

[1] Department of Computer Science & Engineering,
Thapar University,
Patiala
sunilgautam82@gmail.com
[2] Department of Computer Science & Engineering
Deenbandhu Chhotu Ram University of
Science & Technology Murthal (Sonepat)
rbhatiapatiala@gmail.com
[3] School of Mathematics & Computer Application,
Thapar University,
Patiala
rakumar@thapar.edu

Abstract. Software architecture plays an important role during the design stage. It is one of the responsible factors for software quality. Functionalities of the software components have been distributed among the constituents of software architecture. In this paper partitional clustering of the use cases has been proposed. The components to be clustered have been extracted from the mdl file formed by default in Rational Rose. Rational Rose has been taken as the modeling tool to model the software during the design stage. MDL is the model description language describing the facts about the model components in Rational Rose. It is a textual representation of the various aspects about the design of the software. The QUID (qualified unique identifier) has been extracted from the MDL file on which k means clustering algorithm has been applied which results into non overlapping clusters. Euclidean distance measure has been used to measure the distance between the components. In order to represent the partitional clustering, graphs have been drawn.

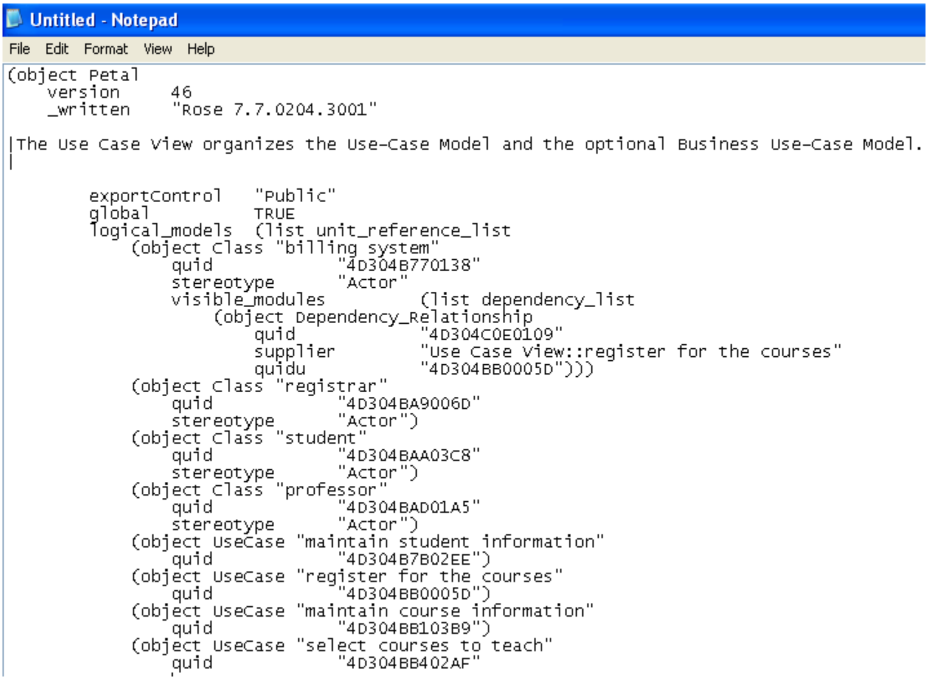
Keywords: MDL, QUID, Use- Case, Clustering.

1 Introduction

Component based development (CBD) is now a day's comes out to be an effective field in the development and maintenance of information system [1]. A component is a block of software that can be designed and if necessary can be integrated with the other components [2]. During the design phase the software components have been determined by use case model, object model and dynamic or collaboration diagrams [3]. To enhance the reusability one approach is to cluster the software components. Our approach will cluster the use cases extracted in the form of numerical or coded values from the mdl file [6]. Clustering has to be performed by evaluating the distance between the software components. The aim of clustering is to improve the technical

P.V. Krishna, M.R. Babu, and E. Ariwa (Eds.): ObCom 2011, Part II, CCIS 270, pp. 57–67, 2012.

assessment criteria proposed by [4]. There are many clustering methods like hierarchical and partitional. Hierarchical clustering algorithms can be further categorized into two types [5], Agglomerative and Divisive. Agglomerative clustering initializes with N clusters and each of them includes only one object. A series of merge operations then follow which finally had all objects to the same group. Based upon the distance between the clusters agglomerative clustering can be further categorized. But we are concerned only with the partitional clustering only. In this method of clustering data has to be partitioned into different clusters. One such technique is the K- means [18] algorithm. k- means clustering is a partitional clustering technique to find the k non-overlapping clusters. The clusters are the representation of cluster centers i.e centroids. A centroid is the mean of the data points in a cluster. In this technique initially k initial centroids are selected randomly and clusters are evaluated. In this evaluation every data point in the data is assigned to the closest cluster center. Thus collection of the data points assigned to a cluster center forms a cluster. There may be several iterations done in order to find the non overlapping clusters. This process is repeated until there is no change in the cluster data points. The data to be clustered has been extracted from the mdl file. An instance of the mdl file is shown below in figure 1. Quid refers to the qualified unique identifier contains 12 hexadecimal digits. Name of the stereotype has been mentioned after the name of the object that is either the object is an actor or a use case.



```
(object Petal
    version     46
    _written    "Rose 7.7.0204.3001"

|The Use Case View organizes the Use-Case Model and the optional Business Use-Case Model.
|

        exportControl   "Public"
        global          TRUE
        logical_models  (list unit_reference_list
            (object Class "billing system"
                quid                "4D304B770138"
                stereotype          "Actor"
                visible_modules          (list dependency_list
                    (object Dependency_Relationship
                        quid                "4D304C0E0109"
                        supplier            "Use Case View::register for the courses"
                        quidu               "4D304BB0005D")))
            (object Class "registrar"
                quid                "4D304BA9006D"
                stereotype          "Actor")
            (object Class "student"
                quid                "4D304BAA03C8"
                stereotype          "Actor")
            (object Class "professor"
                quid                "4D304BAD01A5"
                stereotype          "Actor")
            (object UseCase "maintain student information"
                quid                "4D304B7B02EE")
            (object UseCase "register for the courses"
                quid                "4D304BB0005D")
            (object UseCase "maintain course information"
                quid                "4D304BB103B9")
            (object UseCase "select courses to teach"
                quid                "4D304BB402AF"
```

Fig. 1. MDL File

Distance measure is an important decision for clustering. This will evaluate the similarities or dissimilarities between two components. Also the shape of the clusters is to be decided by the distance measure. Distance measures are available for different dimensions. There are so many distance measures available. In our approach we are using Euclidean distance measure.

A distance between the components is to be measured by using Euclidean Formulae. This formula can be applied from one to N dimension datasets.

For one dimension: $\sqrt{(x-y)^2} = |x-y|$

For two dimensions: $d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2}$

For three dimensions: $d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + (p_3-q_3)^2}$

For n dimensions: $d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + (p_3-q_3)^2 \ldots\ldots + (p_i-q_i)^2 + \ldots\ldots + (p_n-q_n)^2}$

2 Literature Survey

In [7], the static and dynamic relationships between the classes have been analyzed. Static relationship measures the relationship strength and dynamic measures the frequency of messages exchange at run time. To evaluate the overall strength of the relationship between the classes, the static and dynamic results have been combined.

In [8], a decomposition method of requirements has been proposed. Hierarchical clustering has been performed to partition the system.

In [9], a tool called as UML analyzer has been proposed. Tool will abstract the class diagrams and object diagrams in UML at higher level. It can extract the classifiers, relations and semantics. It is an automated abstraction technique and had in built abstraction rules for class and object diagrams. Rational Corporation has also adopted this tool and has implemented it on the Rational Rose [17].

In [10], a systematic UML based method to identify the components has been discussed. It is based upon the assumption that the object oriented model for the target domain is available. This includes the use case model, object model and dynamic model. The method utilises these artifacts and transforms them into components [9].

In [11] a method to measure the interclass relationships in terms of create, retrieve update, and delete CRUD has been proposed. Clustering algorithm for shifting rows and columns was implemented to make appropriate clusters. Based upon the data dependency and interclass relationships among the classes, clusters were formed [10].

In [12], an assessment system for UML class diagram called as UML class diagram assessor UCDA has been proposed. The tool gets an input in the form of Rational Rose petal files. The tool will evaluate the class diagram on the basis of three aspects: structure, correctness and language used. The output of the tool is a list of comments on the diagram that are to be used in understanding the requirements. The naming convention for the class and its attributes were based upon the Malay language. The author had also proposed an extraction technique that extracted the notation information from the Rational Rose Petal files and were kept in the tables for further assessment.

In [13], OMG has clearly specified the representation of a software asset. An asset comprises of profile name, description, classification, solution, usage, and related asset. These are the reusable asset specification of an asset. Profile describes the particular type of asset being described. Description provides the summary of the asset. Classification contains the description which classifies the characteristics and behavior of the asset. Solution contains the location of the specific artifact that comprises the asset. Usage defines how to use the asset. Related asset describes the relationship between the assets.

[14] has developed a tool to check the syntax, rules and notations imposed by the UML specifications similar to the [9] called as UMLST, unified modeling language specification tool. Many tools are available to develop the UML specification like Visio [15], Cadifra [16], and Rational Rose [17]. [9] uses the java programming and deals with the architecture and design mismatches in the UML models, where as this tool uses C++ programming and deals with the UML diagram abstract syntax, its well formed, semantics and notations used in the UML specifications. UMLST first checks the diagrams against each other for any mismatch word and then check the diagram abstract syntax, its well formed, semantics and notations. It has been implemented to check the compliance between the class diagrams, activity diagram, interaction and use case diagram.

3 Proposed Approach

Extraction of design components from the mdl file followed by clustering has been used. The approach can be better understood by the following block diagram in fig 2. The starting point of the process is the requirement analysis. On this basis the use case view has to be modeled. The requirements of the software have been mapped to use case diagram. The model has been saved by some name e.g abc in the same directory in which the rational files have been saved. The file is then reopened in the notepad. It will appear as an unstructured text file containing all the information contained in the model drawn. From this file the quid of the components of use case view has been extracted and converted to their decimal equivalent. Prime factorization theorem has been applied on these decimal numbers and a graph has been formed. The graph visualizes the clusters.

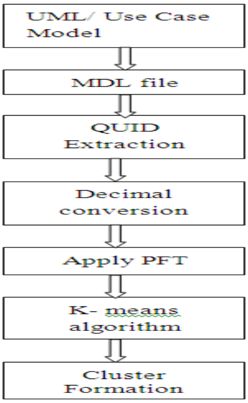
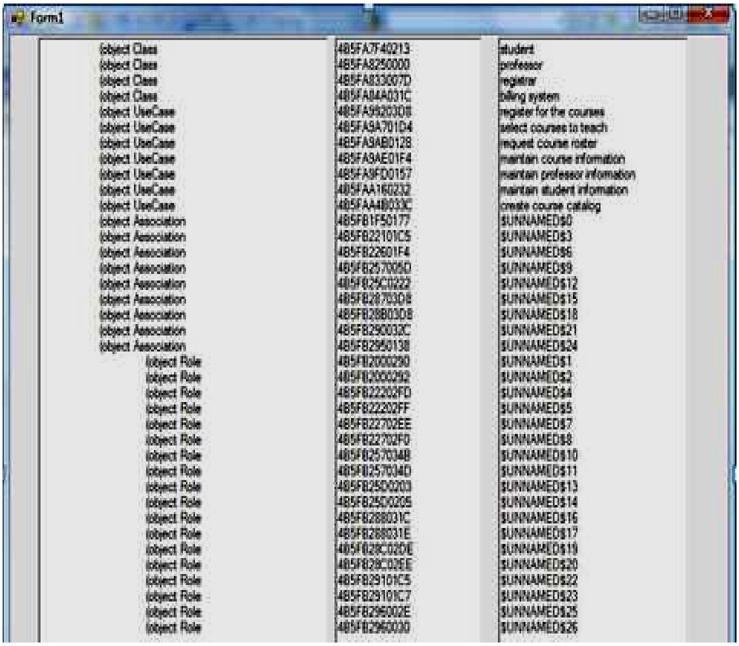


Fig. 2. Process

The above process has been implemented on the use case view. Microsoft visual studio .net and SQL server 2005 has been used. The screenshots of the extracted information has been shown below in fig 3.



```
(object Class              485FA7F40213   student
(object Class              485FA8250000   professor
(object Class              485FA833007D   registrar
(object Class              485FA84A031C   billing system
(object UseCase            485FA99203D8   register for the courses
(object UseCase            485FA9A701D4   select courses to teach
(object UseCase            485FA9AB0128   request course roster
(object UseCase            485FA9AE01F4   maintain course information
(object UseCase            485FA9FD0157   maintain professor information
(object UseCase            485FAA160232   maintain student information
(object UseCase            485FAA4B033C   create course catalog
(object Association        485FB1F50177   $UNNAMED$0
(object Association        485FB22101C5   $UNNAMED$3
(object Association        485FB22601F4   $UNNAMED$6
(object Association        485FB257005D   $UNNAMED$9
(object Association        485FB25C0222   $UNNAMED$12
(object Association        485FB28703D8   $UNNAMED$15
(object Association        485FB28B03D8   $UNNAMED$18
(object Association        485FB290032C   $UNNAMED$21
(object Association        485FB2950138   $UNNAMED$24
       (object Role        485FB2000290   $UNNAMED$1
       (object Role        485FB2000292   $UNNAMED$2
       (object Role        485FB22202FD   $UNNAMED$4
       (object Role        485FB22202FF   $UNNAMED$5
       (object Role        485FB22702EE   $UNNAMED$7
       (object Role        485FB22702F0   $UNNAMED$8
       (object Role        485FB257034B   $UNNAMED$10
       (object Role        485FB257034D   $UNNAMED$11
       (object Role        485FB25D0203   $UNNAMED$13
       (object Role        485FB25D0205   $UNNAMED$14
       (object Role        485FB288031C   $UNNAMED$16
       (object Role        485FB288031E   $UNNAMED$17
       (object Role        485FB28C02DE   $UNNAMED$19
       (object Role        485FB28C02EE   $UNNAMED$20
       (object Role        485FB29101C5   $UNNAMED$22
       (object Role        485FB29101C7   $UNNAMED$23
       (object Role        485FB296002E   $UNNAMED$25
       (object Role        485FB2960030   $UNNAMED$26
```

Fig. 3. Extracted Information

The quids extracted from the mdl file are converted to their decimal equivalents. Analysis of the hexadecimal quid and the decimal equivalent of the actor and use cases have been done that resulted into the observations listed in section 3.

3.1 Hexadecimal to Decimal Conversion

Mathematically, let

$$X = h_1h_2h_3h_4h_5h_6h_7h_8h_9h_{10}h_{11}h_{12}$$

be the quid of any actor or use case.

Convert this no to its decimal equivalent and add. Let it be D.

$$D = d_1+d_2+d_3+d_4+d_5+d_6+d_7+d_8+d_9+d_{10}+d_{11}+d_{12}$$

Where d_1.......d_{12} is the decimal equivalent of h_1……….h_{12}. The use case and class diagram of online marks analysis system is shown in fig 4 and fig 5 below:

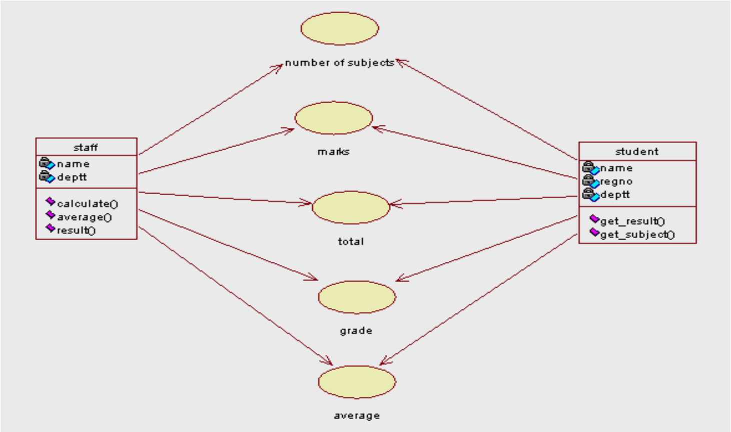


Fig. 4. Use- Case Diagram

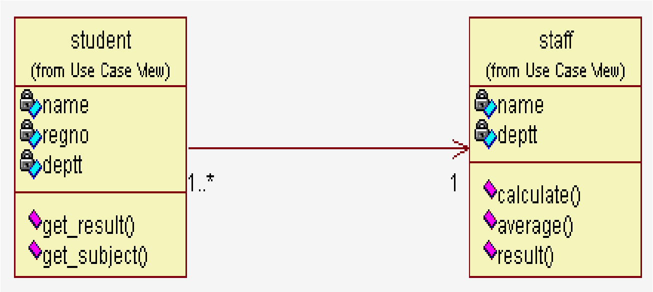


Fig. 5. Class Diagram

Consider the use case diagram only:

Now look at the MDL file of the above diagrams. The components in the above model can be clustered by using the following process:

1. Analyse the MDL file of the above model.
2. Extract the quids of the components contained in the above model.
3. Convert the hexadecimal quid into decimal no.
4. Now calculate the difference between the decimal equivalents of the actor and use cases in which a relationship exists like the actor staff and use case no. of subjects.
5. Now implement prime factorization theorem on the differences obtained from the quids and draw a graph. From the above facts the differences obtained are as under:
   1, 5, 3, 21, 10, 15, 21, 13, 5, 26. Obtain the factors of the above numbers:

   No = factor1 x factor2
   1= 1x1
   5=1x5

3=1x3
21=3x7
10=2x5
15=3x5
21=3x7
13=1x13
5=1x5
26=2x13

Now factor 1 will be on the x axis and factor 2 will be on the y axis. This form a table as below:

Table 1.

A	(1, 1)
B	(1, 5)
C	(1, 3)
D	(3, 7)
E	(2, 5)
F	(3, 5)
G	(3, 7)
H	(1, 13)
I	(2, 13)
J	(1, 5)

3.2 K Means Clustering

k means clustering is a method of data mining whose aim is to partition n data items into k clusers. In this every data item belongs to the nearest mean or the cluster center. The algorithm is composed of following steps:

1. Place K points into the space represented by the data points which are to be clustered.
2. Initially assume the value of k i.e no of clusters.
3. Assign each data item to the group that has the closest centroid.
4. When all data items have been assigned, recalculate the position of the K centroids.
5. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups.

Let us apply k means clustering on the data set in Table 1. The duplicate values in the above table must be omitted. Therefore the new data set will be shown in table below. There are eight data items in the data set.

Table 2.

1	A	(1, 5)
2	B	(1, 3)
3	C	(3, 7)
4	D	(2, 5)
5	E	(3, 5)
6	F	(1, 13)
7	G	(2, 13)

Assume K= 2 and the initial cluster centers as B and D i.e (1, 5) and (3, 7).

First we list all points in the first column in the table drawn below. The initial cluster centers means are (1, 5) and (3, 7) chosen randomly. Next we will calculate the distance from the first point (1, 5) to each of the two means by using the distance measure.

Table 3.

		(1, 5)	(3, 7)	
	Point	Mean 1	Mean 2	Cluster
A	(1, 5)	0	4	CC1
B	(1, 3)	4	6	CC1
C	(3, 7)	4	0	CC2
D	(2, 5)	1	3	CC1
E	(3, 5)	2	2	CC1 & CC2
F	(1, 13)	8	8	CC1 & CC2
G	(2, 13)	9	7	CC2

By carefully analyzing the above table it is clear that points F and G belongs to both the cluster centers. It means that the cluster centers should be re-calculated. This can be clearly visualized from the graph in fig 6 that both the clusters are overlapping.

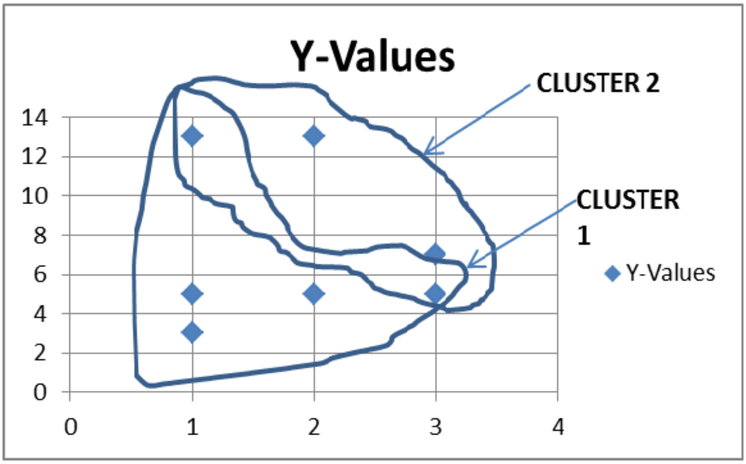


Fig. 6. Graph showing Overlapping of Clusters

As shown in the graph cluster1 and cluster 2 are overlapping. So let us re-calculate the cluster centers. For this calculate the mean for both the above clusters.

For cluster 1
NCC= (1.5, 5.3)
For cluster 2
NCC= (2.2, 9.5)

Now calculate the mean for new cluster centers as in the table 4 below:

Table 4.

		(1.5, 5.3)	(2.2, 9.5)	
	Point	Mean 1	Mean 2	Cluster
A	(1, 5)	0.8	5.7	CC 1
B	(1, 3)	2.8	7.7	CC 1
C	(3, 7)	3.2	3.3	CC 1
D	(2, 5)	0.8	4.7	CC 1
E	(3, 5)	1.8	5.3	CC 1
F	(1, 13)	8.2	4.7	CC 2
G	(2, 13)	8.2	3.7	CC 2

Two separate clusters are shown in the graph (fig 7) below on the basis of above table.

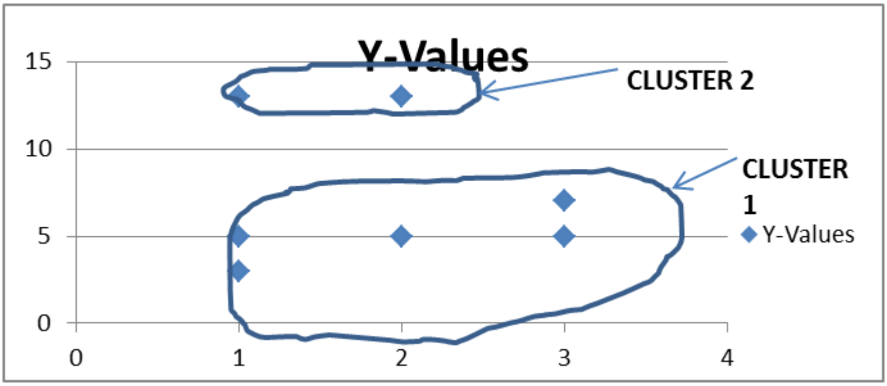


Fig. 7. Graph Showing Cluster Formation

4 Results

The graph shown above indicates two clusters, which are located at far distances from each other. Now let us map the clustered points to actual components.

Cluster 1includes the data points 5, 3, 21, 10, 15. These are the difference values obtained from the decimal equivalents of the quids in the use case diagram. This includes the components staff (actor), marks (use case), total (use case), no of subjects (use case), marks (use case), average (use case), grade (use case).

Cluster 2 includes the data points 13, 26. This includes the components student (actor), no of subjects (use case), average (use case). Our approach deals with the clustering of use- cases. The novelty in our method is the extraction of components from the mdl file in the form of quids. We applied clustering on the values obtained from quids. We consider only the use- case diagram. Clustering can be applied for other UML components also. Since each diagram or view in a model comprises of various components.

References

1. Peng, L., Tong, Z., Zhang, Y.: Design of Business Component Identification method with graph segmentation. In: 3rd Int. Conf. on Intelligent System and Knowledge Engineering, pp. 296–301 (2008)
2. Wu, R.: Componentization and semantic mediation. In: 33th Annual Conference of the IEEE Industrial Electronic Society, Taiwan, pp. 111–116 (2007)
3. Kim, S., Chang, S.: A Systematic method to identify Software Components. In: Proc. of 11th Software Engineering Conference, pp. 538–545 (2004)
4. Mili, A., Mili, R., Mitterweir, R.T.: A survey of Software Reuse Libraries. Annals of Software Engineering 5, 349–414 (1998)
5. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
6. Dahm, M.: Grammar and API for Rational Rose petal files (July 2001)

7. Jain, H., Chalimeda, N.: Business Component Identification- A Formal Approach. In: Proc. of the 5th IEEE Int. Conf. on Enterprise Distributed Object Computing, p. 183 (2001)
8. Lung, C.H., Zaman, M., Nandi, A.: Application of Clustering Technique to Software Partitioning, Recovery and Restructuring. Journal of System and Software, 227–244 (2004)
9. Egyed, A.: Semantic abstraction rules for class diagrams. In: Proceedings of 15th International Conference on Automated Software Engineering, ASE 2000 (2000)
10. Kim, S.D., Chang, S.H.: A systematic method to identify software components. In: Proceedings of 11th Asia Pacific Software Engineering Conference, APSEC 2004 (2004)
11. Lee, S., Yang, Y., Cho, E., Kim, S., Rhew, S.: COMO: A UML based component based methodology. In: Proceedings of the IEEE Sixth Asia Pacific Software Engineering Conference (December 1999)
12. Ali, N.H., Shukur, Z., Idris, S.: A Design of an Assessment System for UML Class Diagram. In: Fifth International Conference on Computational Science and Applications (2007)
13. OMG. OMG unified modeling language, UML (2004), `http://www.omg.org/`
14. Ibrahim, R., Ibrahim, N.: A tool for checking the conformance of UML specification, `http://www.waset.org/journals/waset/v51/v51-45.pdf`
15. Microsoft Visio Toolbox (2008), `http://www.visiotoolbox.com`
16. Cadifra UML Editor (2008), `http://www.cadifra.com/`
17. Rational Rose (2008), `http://www.rational.com`
18. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. 5th Berkeley Symp. Math. Stat. Probab., vol. I, Univ. California Press, Berkeley (1967)