2011 International Conference on Information and Intelligent Computing
IPCSIT vol.18 (2011) © (2011) IACSIT Press, Singapore

Object Use- Cases Clustering using PFT

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

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

Abstract. There are many sources of software components from where they can be extracted. In this paper MDL (model description language) has been taken as the source of components for extraction and clustering. MDL is an unstructured text file for the representation of software components. Each component has been assigned a unique identity in the form of a hexadecimal number i.e quid (qualified unique identifier). For clustering this number is converted to its decimal equivalent. Prime factorization theorem has been applied on these decimal numbers and a graph has been drawn. The graph visualizes the clusters. The technique is UML (unified modeling language) based.

Keywords: UML, MDL, QUID.

1. Introduction

Clustering is the process to represent objects and phenomena, their features into groups [1] on the basis of similarity and dissimilarity measures [2]. Data set required for clustering consists of an object attribute matrix in which objects are grouped on the basis of similarity. A similarity coefficient measure for any pair of objects is the degree of similarity between these two objects [1]. Our aim of this paper is to illustrate a visualization technique of clustering. The components taken are design components. Component clustering plays a vital role in component based development (CBD). It has been identified as one of the effective solution for software development and maintenance [3].

The components have been extracted from the mdl file. MDL file is an unstructured text file containing the description of all the components used in the model. The format of a mdl file is shown here in figure 1. It is not possible to show the whole file here hence an instance of it is shown here. The approach is UML based and the UML notation has been used to model the software. The UML model comprises of four views namely use case view, logical view, component view and the deployment view [5]. Use case view has further two perspectives. One is use case model and other is business use case model. Use case model has been discussed here. Essential components of use case model are the actor. Use case and the relationships.

In arithmetic prime factorization technique has been used to obtain the prime factors [6]. This technique is to find which prime numbers multiply together to make the original number. It is a fundamental theorem of arithmetic which states that any integer greater than 1 can be written as a unique product of prime numbers [7]. Let us understand it with the help of an example:

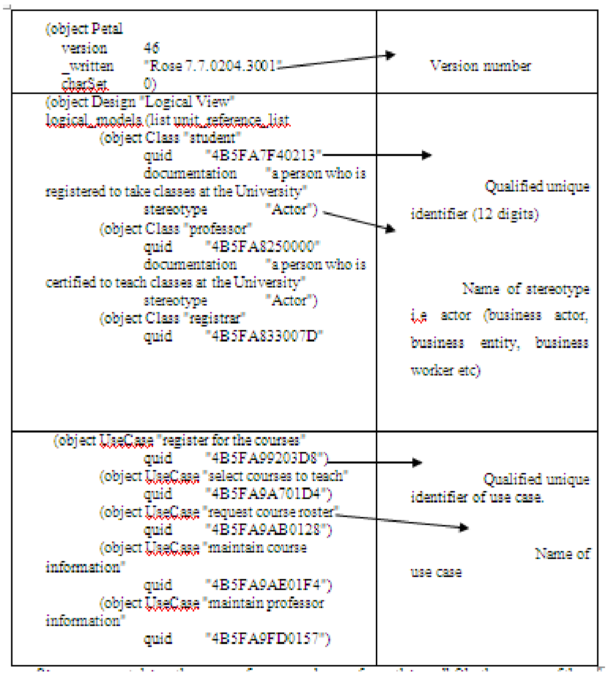
Calculate the prime factorization of 147.it will be

$$147= 3x49 =3x7x7.$$

Hence the factors are 3, 7, 7.

Now to represent the factors a two dimensional graph

G(*x*,*y*) has been drawn. The clusters has been clearly visualized from the graph.



```
(object Petal
   version       46
   _written      "Rose 7.7.0204.3001"
   charSet       0)
```

Version number

```
(object Design "Logical View"
logical_models (list unit_reference_list
          (object Class "student"
                quid        "4B5FA7F40213"
                documentation       "a person who is
registered to take classes at the University"
                stereotype              "Actor")
          (object Class "professor"
                quid        "4B5FA8250000"
                documentation       "a person who is
certified to teach classes at the University"
                stereotype              "Actor")
          (object Class "registrar"
                quid        "4B5FA833007D"
```

Qualified unique identifier (12 digits)

Name of stereotype i.e actor (business actor, business entity, business worker etc)

```
 (object UseCase "register for the courses"
                quid        "4B5FA99203D8")
          (object UseCase "select courses to teach"
                quid        "4B5FA9A701D4")
          (object UseCase "request course roster"
                quid        "4B5FA9AB0128")
          (object UseCase "maintain course
information"
                quid        "4B5FA9AE01F4")
          (object UseCase "maintain professor
information"
                quid        "4B5FA9FD0157")
```

Qualified unique identifier of use case.

Name of use case

Figure 1. MDL file

In the above file every component has been represented by a hexadecimal number known as quid (qualified unique identifier). It is a unique identification given to every component by Rational Rose. The extraction and clustering of the design components from the mdl file has been proposed in this paper.

2. Literature Review

Alexander Egyed developed a tool called as UML analyzer. 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 build abstraction rules for class and object diagrams. Rational Corporation has also adopted this tool and has implemented it on the Rational Rose [8].

Soo Dong Kim and Soo Ho Chang proposed a systematic UML based method to identify the components. 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].

Lee et al proposed a method to measure the interclass relationships in terms of create, retrieve update, and delete CRUD. 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].

Noraida Haji Ali et al proposed an assessment system for UML class diagram called as UML class diagram assessor UCDA. 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 [11]. 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.

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 he 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 [15].

Rosziati Ibrahim and Noraini Ibrahim has developed a tool to check the syntax, rules and notations imposed by the UML specifications similar to the [8] called as UMLST, unified modeling language specification tool [12]. Many tools are available to develop the UML specification like Visio [13], Cadifra [14], and Rational Rose [4]. [8] 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 formedness, 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 formedness, 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 figure 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.

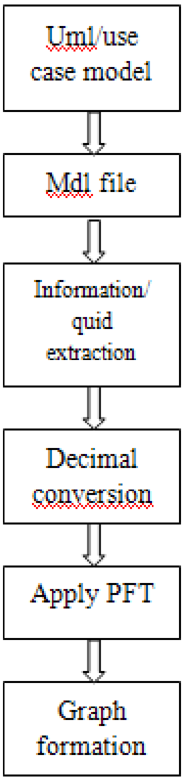
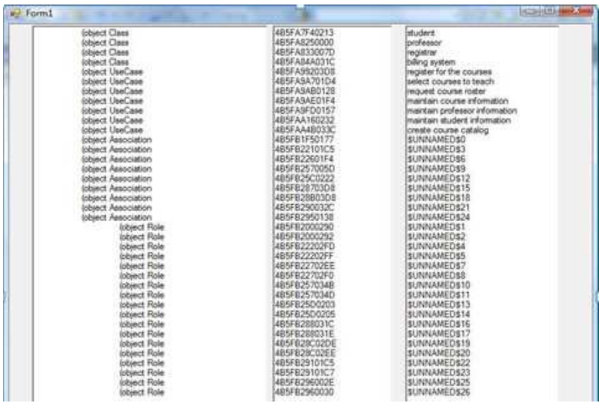


Figure 2. Clustering Process

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 figure 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
```

Figure 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 has been done that resulted into the observations listed in Section III.

A. *Hexadecimal to Decimal Conversion*

Mathematically, let

$$X = h1h2h3h4h5h6h7h8h9h10h11h12$$

be the quid of any actor or use case.

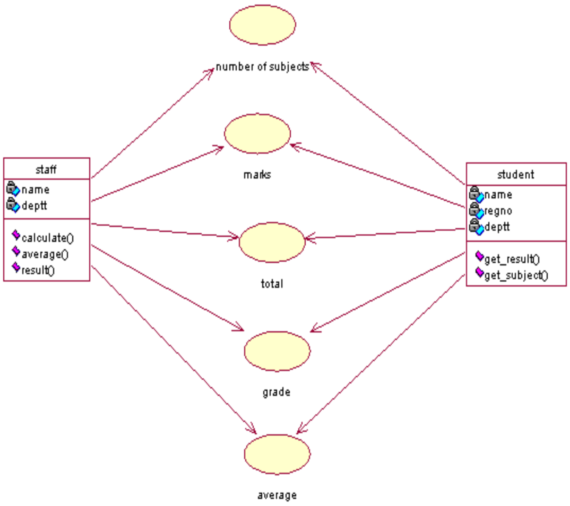
Convert this no to its decimal equivalent and add. Let it be D.

$$D = d1+d2+d3+d4+d5+d6+d7+d_8+d9+d10+d11+d12$$

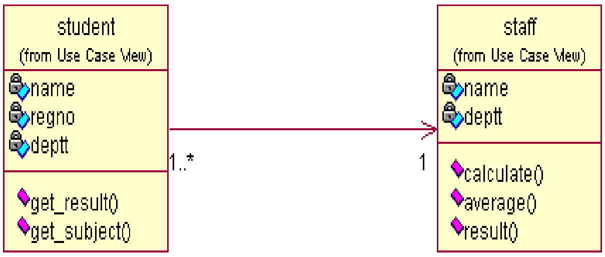
Where $d_1.......d_{12}$ is the decimal equivalent of $h_1.........h_{12}$.

The use case and class diagram of online marks analysis system is as follows:

Use case diagram:



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:

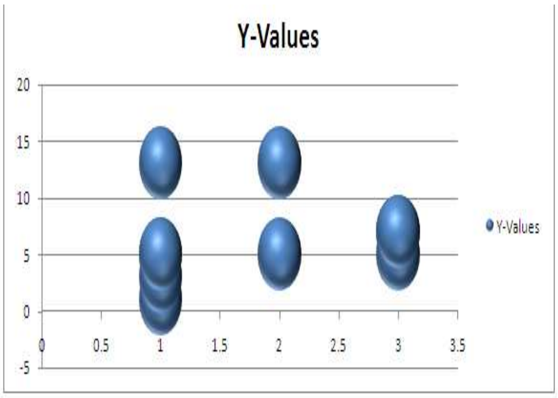
1. Analyse the MDL file of the above model.
2. Extract the quids of the components contained in the above model.
3. Convert the hexadecimal quid into decimal no.
4. Now calculate the difference between the decimal equivalents of the actor and use cases in which a relationship exists like the actor staff and use case no. of subjects.
5. Now implement the **prime factorization theorem** on the differences obtained from the quids and draw a graph.
6. 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. Now draw the graph:

The graph will look like:



4. Results

As shown in the graph we have five clusters. Now let us validate the results:

Biggest cluster as visualize from the graph consists of difference values 1,3, 5, 5.

These constitute the components

1. Staff (actor)
2. Number of subjects (use case)
3. Marks (use case)
4. Total (use case)
5. Student (actor)
6. Grade (use case)

The second cluster consists of difference values 15, 21. These constitute the components:

1. Student (actor)
2. No of subjects (use case)
3. Marks (use case)
4. Staff (actor)
5. Marks (use case) Duplicates may be deleted.

The left differences are 10, 13, 26.

The components associated with the 10 are

1. Staff (actor)
2. Grade (use case)

The components associated with the 13 are

1. Student (actor)
2. Total (use case)

The components associated with the 21 are

1. Student (actor)
2. Marks (use case)

Now analyze the above clusters. These make an independent functionality. Using the prime factorization theorem helps us in visualizing the clusters. This is an easy to implement technique to form and visualize the clusters.

5. Future Scope

In the proposed approach extraction and clustering has been performed on the use case view. Since the mdl file represents the other views also hence the same approach can be implemented to cluster the other components. This will further enhance the reusability characteristics.

6. References

[1] Gholam Reza Shahmohammadi, Saees Jalili and Seyed Mohammad Hossein Hasheminejad “Identification of System Software components using Clustering Approach,” Journal of Object Technology- 2010.

[2] R. Xu and D. Wunsch “Survey of Clustering Algorithms,” IEEE Transactions on Neural Networks, vol 16, No 3, May- 2005, PP. 645-678.

[3] L. Pung, Z. Tong and Y. Zhang “Design of Buissness component identification method with graph segmentation,” 3rd International Conference on Intellegent System and Knoledge Engineering, PP. 296-301, 2008.

[4] Rational Rose, 2008, http://www.rational.com.

[5] James Rumbaugh, Ivar Jacobson and Grady Booch “The Unified Modeling language Reference Manual,” Pearson Education , 2009.

[6] www.mathsisfun.com/prime-factorization.html

[7] www.en.wikipedia.org/wiki/fundamental_theorem_of_airth metic

[8] Alexander Egyed “ Semantic abstraction rules for class diagrams,” Proceedings of 15th international conference on Automated software engineering ASE-2000.

[9] Soo Dong Kim and Soo Ho Chang “A systematic method to identify software components,”., proceedings of 11th Asia Pacific software Engineering Conference APSEC-2004.

[10] Lee, S., Yang, Y., Cho, E., Kim, S., and Rhew, S ., “ COMO: A UML based component based methodology,”, Proceedings of the IEEE Sixth Asia Pacific Software Engineering Conference, Dec. 1999.

[11] Noraida Haji Ali, Zarina Shukur and Sufian Idris “A Design of an Assessment System for UML Class Diagram,” Fifth International Conference on Computational Science and Applications, 2007.

[12] Rosziati Ibrahim and Noraini Ibrahim “A tool for checking the conformance of UML specification,” www.waset.org/journals/waset/v51/v51-45.pdf.

[13] Microsoft Visio Toolbox, 2008,http://www.visiotoolbox.com

[14] Cadifra UML Editor, 2008, http://www.cadifra.com/

[15] OMG. 2004 OMG unified modeling language, UML. http://www.omg.org/.