International Journal of Computer Science and Software Engineering
Volume 1, Number 1 (2015), pp. 17-24

http://www.irphouse.com

The Role and Issue of Clustering Techniques in Designing Maintainable Object Oriented System

Dr. Manju Kaushik and Mrs. Bhawana Mathur

Associate Professor, Dept. of Computer Science & Engineering,
JECRC University, Jaipur, India
manju.kaushik@jecrcu.edu.in
Research Scholar, Dept. of Computer Science & Engineering,
JECRC University Jaipur, India
bhawanamathur19@gmail.com

Abstract

Software development is dynamic and always undergoing major change. The methods and tools will differ significantly from those currently in use. Researcher can anticipate which methods and tools are going to succeed, but Researcher cannot predict the future.
Today a vast number of tools and methodologies are available for systems development. Systems development refers to all activities that go into producing an information systems solution. There is always has been a demand to provide efficient and effective high quality software. There are various maintainability objects to provide better maintainability. The quality of good software design heavily effective the quality of software. One of the claimed advantages of object-oriented paradigm is ease of maintenance. This paper provides the review on the papers studied on the software maintainability model with object oriented system. In other words, it reviews several journal and conference paper on software maintainability in object-oriented software system. In which large number of maintainability model and different subset related to maintainability in object-oriented system are described. Thus this study focuses on the different variable, methods and datasets are used and the analysis used by various authors. This review provides the benefit in future for research as a comparative analysis.

Keywords:- Software development, methods, tools, Programs, Object Oriented Design, Maintainability, Object-oriented, Model, UMLetc.

I.INTRODUCTION

The main objective of Object Oriented Design is to define and understand the[1]

- The object oriented philosophy and why it is needed
- The unified approach, methodology used to study the object oriented concepts

Systems development activities consists of

- Systems analysis,
- Modeling,
- Design,
- Implementation,
- Testing, and
- Maintenance.

A program can be defined as

Algorithms + Data Structures = Programs

"A software system is a set of mechanisms for performing certain action on certain data."

Most of the software maintainability assessment model have been proposed and compared with other models.

II. REVIEW OF LITERATURE

MEMOOD model, which provide an opportunity to improve the maintainability or understandability of class diagram and consequently the maintainability in final software.[1]

The object-oriented (OO) is ease of maintenance to provide better understand ability and modifiability. It describes three technique discriminant techniques (correlation between maintainability and structural complexity), weighted score level technique (combination of understanding and modifiability) and weighted predicate level technology (combination of predicate understand ability and modifiability).[2]

The proposed adaptive maintenance effort model (AMEffMO) used to determine the line of code change and also describe the regression model for adaptive maintenance, which can provide the useful information for manager and maintainer.[3]

To proposed a maintainability model that categorized software module as "easy to maintain" and "not easy to maintain", which can help to identify that modules are not easy to maintenance, before integrating and also introduced an effort based metrics, mean-time to-change (MTTC) to predict maintainability.[4]

To proposed that the linear prediction model which is being evaluated by some industrial software system to estimate the maintainability of large system and to identified some fault prone models to define impact rate, effort and error rate.[5]

In outsourcing maintenance, which is being prepared a little bit information on the software to be maintained to estimate their maintainability.[6]

Some internal and external attribute used in object-oriented (OO) software to analyse metrics, such as structural complexity and size of UML can be used in early as an maintainability indicators, which is used to gather empirical data to turn in the

basis of current study and also define some measuring properties of object-oriented (OO) such as inheritance, cohesion and coupling.[7]

III. INTRODUCTION OF CLUSTERING TECHNIQUES WITH RESPECT OBJECT ORIENTED SYSTEM

3.1 Role of Clustering with respect to Object Oriented System

- To grouping of Objects.
- To minimize the number of Object.

3.2 Role of Clustering with respect to Object

- With help of object researcher draw UML, Class diagram[4].
- To minimize the number of classes.
- To minimize the number of objects at design level, which can fullfill by this Clustering Technique.

3.3 Issues of Clustering Techniques in Designing Maintainable Object Oriented System

- The value of K.It may be K=2, K=3, K=4, K=5, and so on, it may be any one. So, after that result may be differ for different K value.
- The shape may be convex and concave.
- Redundant attributes in different Cluster.

3.4 The problem of the number of clusters

- Typically, the number of clusters is known.
- When it's not, that is a hard problem called model selection. There are several ways of proceed.
- A common approach is to repeat the clustering with K=1, K=2, K=3, etc.

IV. BASICS CONCEPTS OF OBJECT-ORIENTED DESIGN

4.1 Objectives of Object-oriented Design

- To explain how a software design may be represented as a set of interacting objects that manage their own state and operations
- To describe the activities in the object oriented design process[3].
- To introduce various models that can be used to describe an object-oriented design
- To show how the UML may be used to represent these models[3].

4.2 Object-oriented development

1. Object-oriented analysis, design and programming are related but distinct.
2. OOA is concerned with developing an object model of the application domain.
3. OOD is concerned with developing an object oriented system model to implement requirements[2].
4. OOP is concerned with realizing an OOD using an OO programming language such as Java or C++[5].

4.3 Characteristics of OOD

1. Objects are abstractions of real-world or system entities and manage themselves.
2. Objects are independent and encapsulate state and representation information.
3. System functionality is expressed in terms of object services.
4. Shared data areas are eliminated. Objects communicate by message passing.
5. Objects may be distributed and may execute sequentially or in parallel.

4.4 Advantages of OOD

1. Easier maintenance. Objects may be understood as stand-alone entities.
2. Objects are potentially reusable components.
3. For some systems, there may be an obvious mapping from real world entities to system objects.

4.5 Objects and object classes

1. Objects are entities in a software system which represents instances of real-world and system entities.
2. Object classes are templates for objects. They may be used to create objects.
3. Object classes may inherit attributes and services from other object classes.
4. OOD is an approach to design so that design components have their own private state and operations.
5. Objects should have constructor and inspection operations. They provide services to other objects.
6. Objects may be implemented sequentially or concurrently.
7. The Unified Modeling Language provides different notations for defining different object models.

4.6 Key points of OOD

1. A range of different models may be produced during an object-oriented design process. These include static and dynamic system models.
2. Object interfaces should be defined precisely using e.g. a programming language like Java[5].
3. Object-oriented design potentially simplifies system evolution.

4.7 Objects and object classes

An **object** is an entity that has a state and a defined set of operations which operate on that state. The state is represented as a set of object attributes. The operations associated with the object provide services to other objects (clients) which request these services when some computation is required. Objects are created according to some **object class** definition. An object class definition serves as a template for objects. It includes declarations of all the attributes and services which should be associated with an object of that class.

4.8 Definition of Unified Modeling Language (**UML**)

The **Unified Modeling Language** (**UML**) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to

visualize the design of a system[3].

4.9 Overview of the Unified Approach

The unified modeling language (UML) is a set of notations and conventions used to describe and model an application[3]. But, the UML does not specify a methodology or what steps to follow to develop an application; that would be the task of the unified approach(UA).

4.10 Advantage of an Object-Oriented System

The main advantage of an object-oriented system is that the class tree is dynamic and can grow. Function as a developer in an object-oriented environment is to foster the growth of the class tree by defining new, more specialized classes to perform the tasks your applications require.

4.11 Generalisation and inheritance

1. Objects are members of classes that define attribute types and operations.
2. Classes may be arranged in a class hierarchy where one class (a super class) is a generalization of one or more other classes (sub-classes).
3. A sub-class inherits the attributes and operations from its super class and may add new methods or attributes of its own.
4. Generalisation in the UML is implemented as inheritance in OO programming languages[3].

4.12 Advantages of inheritance

1. It is an abstraction mechanism which may be used to classify entities.
2. It is a reuse mechanism at both the design and the programming level.
3. The inheritance graph is a source of organizational knowledge about domains and systems.

4.13 Problems with inheritance

1. Object classes are not self-contained. They cannot be understood without reference to their super-classes.
2. Designers have a tendency to reuse the inheritance graph created during analysis. Can lead to significant inefficiency.
3. The inheritance graphs of analysis, design and implementation have different functions and should be separately maintained.

4.14 UML associations

1. Objects and object classes participate in relationships with other objects and object classes.
2. In the UML, a generalized relationship is indicated by an association[4].
3. Associations may be annotated with information that describes the association.
4. Associations are general but may indicate that an attribute of an object is an associated object or that a method relies on an associated object.

V.CLUSTERING TECHIQUES WITH DESIGNING MAINATANABLE OOS

5.1 Definition of Clustering

Clustering is defined as the process with the help of which Researcher can group the data objects which are similar to each other as one cluster. Dissimilarity is observed in the data objects belonging to different clusters. Apart from the use of clustering technique in general applications such as image processing, data analysis, marketing, WWW, etc; it is also used to evaluate and improve the maintainability of the software system. For this purpose Researcher have various clustering algorithms[7].

5.2 Definition of Clustering Analysis

Clustering analysis is the process of finding the groups of objects such that the objects in a group will be similar to one another and dissimilar from the objects in other groups[8-11]. These groups are known as clusters. Researcher can also say that the inter cluster distance should be maximized and the intra cluster distance should be minimized. In other words, data points of a dataset in same cluster are highly related to each other whereas the data points in different clusters are not related to each other. Clustering is sometimes also known as data segmentation as it is used to partition huge data in segments called as clusters[1]. Clustering is applicable for issues such as data analysis where very little information is available and the certain assumptions need to be made, image processing, unsupervised learning, pattern recognition, biology, etc. But, there are some typical requirements of clustering in data mining such as ability to deal with noisy data, scalability, ability to deal with different type of attributes, discovery of clusters with arbitrary shapes, etc[12-13]. With the help of clustering approach researcher can identify co-relation among the attributes of the dataset and the distribution pattern of the particular dataset.

5.3 Advantages of Clustering

The advantages of clustering based approach to improve maintainability is that these processes are adaptable to change and they also tell us about the features which helped in distinguishing the groups or clusters[12-13].

5.4 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages[2].

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

VI. Scope of Object Oriented Maintenance

Maintenance can often be performed without significantly changing the structure of

an existing application. Modification can be done to a group of objects without affecting other objects. Additional objects may be added without needing to know the implementation details of existing objects. Object orientation have several advantages such as lower development effort, lower development time and better maintainability. It is best upon object and constitute a system. Hence, it is more close to real world. The OO approach is supported by some very useful design principles like maintainability, extensibility, reusability, interoperability and quality; its popularity is increasing.

Conclusion

Class maintainability is the likelihood that a class can be easily modified. Before releasing an object-oriented software system, it is impossible to know with certainty when, where, how, and how often a class will be modified. At that stage, this likelihood can be estimated using the internal quality attributes of a class, which include cohesion, coupling, and size. To reduce the future class maintenance efforts and cost, developers are encouraged to carefully test and well document low maintainability classes before releasing the object-oriented system.

Researcher empirically studies the relationship between internal class quality attributes (size, cohesion, and coupling) and an external quality attributes (class maintainability). Using statistical techniques, Researcher also construct models based on the selected internal attributes to predict class maintainability.

Researcher considers classes of three open-source systems. For each class, Researcher account for two actual maintainability indicators, the number of revised lines of code and the number of revisions in which the class was involved. Using many internal quality measures, Researcher empirically explore the impact of size, cohesion, and coupling on class maintainability. Researcher also empirically investigates the abilities of the measures, considered both individually and combined, to estimate class maintainability. Statistically based prediction models are constructed and validated.

References

1. Rizvi S. W. A, Khan R. A., "Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)", Journal Of Computing, **2(4)**, 26-32, (2010).
2. Kiewkanya M., Jindasawat N., Muenchaisri P., "A Methodology for Constructing Maintainability Model of Object-Oriented Design, " 4th International Conference on Quality Software, IEEE Computer Society, Washington, DC, USA, 206, (2004).
3. Hayes J.H., Patel S.C., Zhao L., "A Metrics-Based Software Maintenance Effort Model, " 8th European Conference on Software Maintenance and Reengineering, IEEE Computer Society, Los Alamitos CA, 254, (2004).
4. Hayes J.H., Zhao L., "Maintainability Prediction: a Regression Analysis of

Measures of Evolving Systems, " 21st IEEE International Conference on Software Maintenance, IEEE Computer Society Washington, DC, USA, 601, (2005).

5. Muthana S., Kontogiannis k., Ponnambalam K., Stacey B., "A Maintainable Model for Industrial Software Systems Using Design Level Metrics", IEEE Software, **7(2),** 248-256, 2000.
6. Polo M., Piattini M., Ruiz F., "Using Code Metrics to Predict Maintenance of Legacy Programs: a Case Study, " International Conference on Software Maintenance, IEEE Computer Society, Florence Italy, 202, (2001).
7. Genero M., Manso E., Cantone G., "Building UML Class Diagram Maintainability Prediction Models Based on Early Metrics, " 9th International Symposium on Software Metrics, San Diego, USA, 263, (2003).
8. Malviya, A.K., Dutta, M., "Maintenance Activities in Object Oriented Software Systems Using K-means Clustering Technique: A Review", Software Engineering CSI Sixth International Conference, Indore, 1, (2012)
9. Dubey S.K, Mehra A., "Maintainability Evaluation of Object-Oriented Software System Using Clustering Techniques", Council for Innovative Research International Journal of Computers & Technology, **5(2)**, 136-143, (2013).
10. Beyer D., Noack A., "Clustering Software Artifacts based on Frequent Common Changes", 13 International workshop on Program Comprehension, Switzerland, 259, (2005)
11. Kanellopoulos Y., Others, "K-attractors: A clustering Algorithm for Software Measurement Data Analysis", 19th IEEE International Conference on Tools with A.I., **25(2)**, 2011.
12. Oca C.M., Carver D.L., "Identification of Data Cohesive Subsystem Using Data Mining Techniques", International conference on IEEE, Bethesda, MD, 16, (1998).
13. Lawrence K.D., Kudyba S., KlimbergR.K., "Data mining methods and applications", 2nd Edition, CRC Press, Auerbach Publications, New Delhi, 83, (2008).
14. Kaushik M., Mathur B., **" An** Experimental analysis of Parent Teacher Scale Involvement with help of K Mean Clustering Technique using Matlab", in International Journal of Advance Research in Computer Science and Management Studies(IJARCSMS), **2(9)**, 117-123, (2014)
15. Kaushik M., Mathur B.**,** "Comparative Study of K-Means and Hierarchical Clustering Techniques", International Journal of Software & Hardware Research in Engineering (IJSHRE), **2(6),** (2014)**.**
16. Kaushik M., Mathur B., "Data Analysis of Students Marks with Descriptive Statistics"**,** International Journal on Recent and Innovation Trends in Computingand Communication(IJRITCC), **2(5),** 1188-1191. (2014).