Analysis of Maintainability Models for Object Oriented System

Sanjay Kumar Dubey, Amit Sharma, Prof. (Dr.) Ajay Rana
Department of Computer Science & Engineering
Amity School of Engineering and Technology
Amity University, Sec-125, NOIDA, (U.P.), India
skdubey1@amity.edu, amit.krsharma123@gmail.com, ajay_rana@amity.edu

Abstract—**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-Maintainability, Object-oriented, Model, UML

I. INTRODUCTION

As society become more and more dependent on the software system and demand for the new one, more capable software will be provided on the short cycle, but they requires or need more maintenance. There are various object-oriented paradigms in the frame work for the software. Maintainability is the way through which we can modify an object-oriented software product to overcome their performance and attributes to adapt the product in a modified environment and maintainability is the ease with which software can be corrected errors, adapted to the change of environment and enhanced the software as per the need of the customer strict engineering mostly used to bug fixing.

II. LITERATURE SURVEY

Most of the software maintainability assessment model have been proposed and compared with other models. Zhuo F. *et al.* (1993) proposed maintainability index (MI) that determine the maintainability of software system based upon the status of the source code, which show high correlation between assessments automated model and some expert evaluation. Binkley A. *et al.* (1998) collect the data of maintenance for the development of project written in any language like C, C++, COBOL etc and produce a level of interaction between modules, which show low coupling were subjected for fewer maintenance effort and fewer maintenance fault and failures. Muthana S. *et al.* (2000) 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. Polo M. *et al.* (2001) used outsourcing maintenance, which is being prepared a little bit information on the software to be maintained to estimate their maintainability. Genero M. *et al.* (2003) proposed 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. Hayer J.H. *et al.* (2004), 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. Kiewkanya M. *et al.* (2004) prescribed that object-oriented (OO) is ease of maintenance to provide better understandability 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 understandability and modifiability). Hayes J.H and Zhao L. *et al.* (2005), 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. Rizvi S.W.A. *et al.* (2010) propose a MEMOOD model, which provide an opportunity to improve the maintainability or understandability of class diagram and consequently the maintainability in final software. Gautam C. *et al.* (2011), describe that the compound MEMOOD model is better the MEMOOD model to determine the maintainability of class diagram in terms of their understandability, modifiability, scalability and level of complexity.

III. RESULT ANALYSIS

There have been various empirical studies in the maintainability models are studies in the field of object-oriented system. In this work we study the impact of object-oriented software system maintainability and their attributes that have been considered. This study will help to improve the maintainability of that software which is developed in the very short cycle. There is various maintainability models proposed in the literature. In our review, we have considered only those papers where object-oriented are used. To analyse the methods, variable and datasets from these papers the following procedure was followed in selecting the studies:

We have searched through various papers, journals and conferences such as:

- IEEE
- Journal of system and software
- Journal of information and technology
- International journal of computer science
- Journal of computing
- ACM SIGSOFT Software Engineering
- International journal of computer science and information technology.

All the previous papers since 1993 to 2011 that are concerned with software maintainability in object-oriented system. The title and the abstract of the relevant studies containing key words like object-oriented, maintainability index, line of code, unified modeling language were identified in initial search and then reviewed by assistant professor in Amity University. The review was that the study of maintainability effects in the object-oriented software system.

IV. RESULT DESCRIPTION

We have studied various papers in the relevant field of software maintainability in object-oriented system from the years 1993 to 2011. TABLE-1 shows the summary of our study. It gives the overall review of each paper in which the columns defines the reference, journal in which published or presented, year, variables, methods and datasets used in the particular paper.

- There are numbers of methods are use in the literature such as maintainability analysis tool, halstead metrics, logistic regression, class coupling and cyclomatic complexity but Polo M. (2001) described the Logistic Regression, MANTEMA a methodology for maintenance developed by atos ODS which can specified the explicitly. In some other similar cases, only the names of the method suited are specified.

- Papers have used different type of datasets which are mostly public dataset, commercial dataset, open-source or student/university datasets. We have seen mostly public datasets are used like PROMISE and NASA repositories. Some students and university data sets are Data Collection from ELLENKI research and development center, Hyderabad and some more others like ANOVA, HP-MAS (Hewett Packard- Maintainability Assessments System) by University of Idaho Software Engineering lab, AFOTEC Instrument are used.

We have seen that from year 1993 to 1999 some common datasets are used and now days there usage percentages are so much high in multiples domain. We have summarized our review in some relevant criteria there are so much evaluating criteria are validation and some others parameters.

V. CONCLUSION

This paper review several significant conference paper and journal articles on software maintainability in object-oriented system. Maintenance is necessary for every software system for better efficient running and removes all the faults and failures from the software systems. We have included relevant significant paper on the software maintainability. Moreover, we have included these papers and articles where only object-oriented software systems are used. Based on these criteria, we found total 21 papers or articles relevant to these. We have not give all the detailed description of the paper but our aim to provide some methods, variables and datasets or some techniques used are described. This review will be beneficial for the both the students and the researchers to have a brief overview of the work already done in the field of software engineering. This will help them in carrying out much better and efficient research in future.

VI. LITERATURE REVIEW

S. No.	Paper	Journal name	Year	Variable Used	Methods Used	Dataset used
1	[1]	IEEE Computer Society	1993	aveLOC(Average Line of Code), ES(Executable Statement), CM(Line of Comment), NES(Number of executable Statement)	MAT(Maintainability Analysis Tool), Regression, Halsted metrics, Cyclomatic Complexity, Assessment Model, Entropy	HP-MAS(Hewett Packard-Maintainability Assessments System)by University of Idaho Software Engineering lab, AFOTEC Instrument
2	[22]	Journal of system and software	1993	DIT (Depth of inheritance tree), NOC (Number of children), CBO (Coupling between object), RFC (Response for a class), LCOM (Lack of cohesion of method), WMC (Weight method per class)	Linear, Regression Analysis	Local data sets
3	[24]	Journal of Information and technology	1994	Complexity, coupling, inheritance	Linear L.S.	12 system designed by students to depict specs.
4	[25]	Journal of system and software	1995	CBO(Coupling between object classes), LOC(Line of code)	Linear L.S.	2 ADA system(UIMS,QUES commercial)
5	[23]	Proc. 20th International conference of software engineering	1998	CDM (Coupling dependency metric), CBO (Coupling between object classes), NSSR(Number of sub system relationship), RFC (Response for a class), WMC (Weight method per class), DIT (Depth Inheritance tree),	Class coupling	C++ system (patient core management), 113cls,82KLOC,file transfer facility,29 java classes, 6 KLOC

				CHNL (Class hierarchy nested level), NCIM(Number of class inheriting a method), WIH (Width of inheritance hierarchy), HIH (Height of inheritance hierarchy)		
6	[26]	Proceeding of metrics	1999	C&K without LCOM (lack of cohesion in methods), LOC (Line of code)	Regression Analysis	3C++ subsystem from a telecom application 6kloc/20clk, 21kloc/45clk, 6kloc/27clk
7	[2]	IEEE Computer Society	2000	Impact Rate, Effort, Error Rate, Subjective Evaluation, Goodness-of-fit statistics test, Regression coefficient test, Multidimensional Assessment, Albrecht Metrics, Software Complexity Metric, Card and agresti's Complexity metric	Regression Analysis	MAT(maintainability Analysis Tool)using FLECS(a Structured Fortran preprocessor
8	[3]	International conference of Software maintenance	2001	SLA (Service level agreement), KLOC(Kilo Line of Code), TRCA(Time of resolution of critic Anomalies), MR (No. of modification request), UC (Urgent corrective)	Logistic Regression, MANTEMA a methodology for maintenance developed by atos ODS	Using C++ ,where the context is very different i.e. not available in COBOL (i.e. pointers)
9	[4]	IEEE(International Software Metrics Symposium)	2003	KA(key Abstraction), VOPC(View of Participating Classes), UML(Unified Modeling Language), PCA(Principal Component Analysis), NC(No. of Classes), NA(No. of attributes), NAGG(No. of Aggregation), NDEP(No. of Dependencies),	Linear	Finding the replicated data form from data description using ANOVA method
10	[5]	IEEE (Conference on reverse Engineering)	2003	KLOC(Kilo Line of Code), LCOM(lack of Cohesion in Method),	DTRIX parser used to assess maintainability aspects of object	Java Systems Fujaba-UML(FUML) and dynamic object

				LCC(Loose Class Cohesion), TNOS(Total No. of statement), DIT(Depth of Inheritance Tree), ICAIC(Inheritance class-attribute import coupling), NICAIC(Non-Inheritance class-attribute import coupling), ICAEC(Inheritance Class-attribute export coupling), NICMIC(Non Inheritance Class method import coupling), NIMMIC(Non Inheritance method-method import coupling), IIC(inheritance Import Coupling), IEC(inheritance export Coupling)	oriented software	browser(dobs)
11	[6]	IEEE Computer Society(Eight european conference in Software Maintenance and Reengineering)	2004	MP(maintainability Products), CF(Coupling factor), CR(Comment Ratio), Hdiff(Halstead Difficulty), LCOM(Lack of Cohesion in Methods), AC(Attribute Complexity), CC(Cyclomatic Complexity), TCR(True Comment ratio), PM(Perceived Maintainability), LOC(Line of Code)	COCOMO constructive cost estimation Model), SLIM, AMEffMo(Adaptive Maintenance Effort Model), Regression Analysis	Validation dataset (the residue even increase as DLOC increase)
12	[7]	IEEE Computer Society	2004	understandability, modifiability, UML(Unified modeling Language	Association, Aggregation, Generalization, Classification	Multilayer perceptron and decision trees (applied to construct the maintainability)
13	[8]	IEEE Computer Society (International Conference on Software Maintenance)	2005	CR(Comment Ratio), AC(Attribute Complexity), LOC(Line of Code), TCR(True Comment Ratio), MI(Maintainability Index), LOC(Line of Codes)	Regression Analysis DC Ratio	Spathic Project Data from source code a test generation tool

14	[21]	Journal of system maintenance and evolution	2005	Number of methods, Number of association	Linear model	Local dataset
15	[9]	IAENG International Journal Of Computer Science	2007	UML(Unified Modeling language), ODMG(Object data Management Group), OMG(object Management group), OCL(Object Constraint Language), RDF(Resource Description framework)	CORBA, Ontology	Externalizing and organizing meta-data from data base using W3C technology
16	[10]	IEEE Conference on software maintenance and Reengineering	2007	MA(Multi-criteria Analysis), WMC(Weighted Methods Per Class), NPM(Number of Public Methods), CBO(Coupling Between Objects), NOP(No. of polymorphic Methods), DIT(Depth of Inheritance Tree), LCOM(Lack of Cohesion in methods)	Data Extraction Process, Clustering Maintainability	Methodology to analyze 1440 classes of APACHE Geronima
17	[11]	IEEE Computer Society	2007	K-Attractors, Code-level metrics	Data Extraction Process, Multimedia and GIS(Geographic Information Services)	
18	[12]	Journal OF Computing	2010	Class Diagram, LOC(Line of Code), MI(Maintainability Index) NC (Number of Class) NA(Number of Attributes) NM(Number of Methods)	MEMOOD, Regression	Data from ISO 9126 for reliability and testability
19	[20]	Journal of system and software	2010	LOC(Line of code), CP (change pattern), ROC(Receiver operating characteristics area)	Linear	Local data sets
20	[14]	International Journal Of Software Engineering	2010	class diagram, LOC(Line of Code), DLOC(Difference Line of Code), NC(Number of classes), NA(Number of Attributes), NM(Number of	Linear Regression model	Data collection from multivariate maintainability and modifiability models

				Methods)		
21	[15]	International Journal of Computer Science and Information Technology	2011	Test Cases, Classic life Cycle Quality, Effort ratio effectiveness, Weighted defect density, XP(Extreme programming), TDD(Test-Driven Development)	Regression CR tool Architecture	Data Collection from ELLENKI research and development center, Hyderabad
22	[17]	International Journal of Computer Science and Information Technology	2011	LOC(Line Of Code), DLOC(Difference Line of Code), MI(maintainability Index), CC(Cyclomatic Complexity),	Multivariate Linear model	F-Test for multivariate analysis
23	[18]	ACM SIGSOFT Software Engineering	2011	LOC(line of code), RFC (Response for a class), LCOM (Lack of cohesion of method), DIT (Depth of inheritance tree)	Logistic Regression	Data Sets of PROMISE and NASA repositories

REFERENCES

[1] Zhuo F., Lowther B., Oman P. and Hagemeister Jack., “Constructing and testing software maintainability assesement models”, IEEE Computer Society, 1993, pp 61-70.

[2] Muthanna S., Kontigiannis K., Ponnambalam K. and Stacey B.,”A Maintainability Model for industrial Software System Using Design Level Metrics”, IEEE Computer Society, 2000, pp 248-256.

[3] Polo M., Piattini M. and Ruiz F., “Using code metrics to predict maintenance of legacy programs: a case study”, 2001.

[4] Genero M., Manso E. and Cantone Giovanni., “Building UML class diagram maintainability prediction model based on early metrics”, 2003, pp 1530-1535.

[5] Dagpinar Melis. and Jahnke Jens H., “Predicting maintainability with object oriented metrics-an empirical comparison”, proceeding of the 10th working conference on reverse engineering (WCRE’03), 13-17 nov.2003, 2003, pp 155-164.

[6] Hayes J.H., Patel S.C. and Zhao L., “A Metrics-Based Software Maintenance Effort Model,” Proc. 8th European Conference on Software Maintenance and Reengineering (CSMR'04), 24 – 26 Mar. 2004, pp. 254 – 258, IEEE Computer Society.

[7] Kiewkanya M., Jindaswat N. and Muenchaisri P., “A Methodology for constructing Maintainability Model of Object Oriented Design”, Proc. 4th International Conferences on Quality Software”, 8-9 Sept, 2004, pp.206-213. IEEE Computer Society.

[8] Hayes J.H. and Zaho L., “Maintainability Prediction a Regression Analysis of Measures of Evolving Systems”, Proc.21st IEEE International Conference on Software Maintenance, 26-29 Sept.2005, pp.601-604.

[9] Dr. Waralak. and Sricharoen V., “Ontologies and object models in object oriented software engineering”, IAENG International Journal of Computer Science, 13, Feb, 2007.

[10] Antonellis P., Antonious D., Kanellopoulous Y., Makris C., Theosoridis E., Tjortjis C. and Sirakis T., “A Data Mining Methodology Maintainability According to ISO/ IEC-9126 Software Engineering Product Quality Standard,” Proc.11th IEEE Conference on software maintenance and Reengineering (CSMR2007), 21- 26 Mar.2007, Amsterdam, Netherlands.

[11] Antonellis P., Antonoiu D., Kanellopoulos Y. and Makris Christos., “Code4 Thought Project: employing the ISO/IEC-9126 standard for software engineering-Product Quality Assessment”, European Conferences on software maintenance and reengineering, 2009, IEEE Computer Society.

[12] Rizvi S.W.A. and Khan R.A., “Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)”, Journal of Computing, Volume 2, Issue 4, April 2010,

[13] Dubey S.K. and Rana A., “A comprehensive assessment of object oriented software system using metrics approach”, International journal of computer science and engineering (IJCSE), 2010, pp 2726-2730.

[14] M. Babita, “Software maintenance using estimation model and object oriented design”, International journal of software engineering, 2010.

[15] Shaik A., Bhadriraju H., Vikram K., Shaik N. and Rao S.V.A., “ A Suggestive evaluation of system test cases in OO system trough carving and replaying differential unit test cases: A Metric Approach”, International journal of Computer Science and Technology, 2011, pp 1345-1353.

[16] Saini R., Dubey S.K., Rana A., “Analytical study of maintainability models for quality evaluation”, International journal of computer science and engineering, pp 449-454, 2011.

[17] Gautam C., kang S.S., “Comparison and Implementation of Compound MEMOOD MODEL and MEMOOD MODEL”, International journal of computer science and information technologies, pp 2394-2398, 2011.

[18] Malhotra R. and Jain A., “Software fault prediction for object oriented System: A Literature Review”, ACM SIGSOFT Software Engineering Notes, 2011.

[19] Pressman R.S., SE: A Practice approach (Mc. Graw Hill).

[20] Arisholm E., Briand L. and Johanvessen E., “A systematic and comprehensive investigation of methods to build and evaluate fault prediction model”, Journal of system and software, vol. 83, pp. 2-17.

[21] Bocco M., Moody D. and Piattini M., “Assesing the capability of internal metrics as early indicators of maintenance effort through experimentation”, Journal of system maintenance and evolution, vol.17, pp. 225-246, 2005.

[22] Li W. and Henry S., “Object-oriented metics that predict maintainability”, Journal of system and software 23, pp. 111-122, 1993.

[23] Binkley A. and Schach S., “Validation of the coupling dependency metrics as a predictor of run time failures and maintainability measures”, Proc. 20th International conference of software engineering, pp. 452-455, 1998.

[24] Chan J.Y., Lu J.F., “A new metric for object oriented design”, information Technology vol.35, NO.5, 232-240.

[25] Henry W.L.S., kafura S., Schulman R., “object oriented metrices that predict maintainability”, Journal of System and Software, 23 (2), 111-122, 1995.

[26] Tang, Kao., Chen., “An empirical study in object-oriented metrics”, Proceeding of metrics, 242-249, 1999.