COMPARISON AND IMPLEMENTATION OF SOFTWARE MAINTENANCE MODELS

Chhavi Gautam
R.B.C.E.N.T.W, India

Sandeep Singh Kang
C.E.C Landran, India

Abstract

Measurement of the maintainability of the software system is very important as it will help in the better maintenance of the software system. Measurement of the software maintainability early in the development life cycle, especially at the design phase is important, it may help designers to incorporate required enhancement and corrections at design phase for improving maintainability of the final software. Earlier MEMOOD model was developed which finds the maintainability of the class diagrams on the basis of Understandability and Modifiability of software system on the basis of object oriented metrics of class diagrams of software system. This paper has proposed a multivariate linear model 'Compound MEMOOD, which estimates the maintainability of class diagrams of software systems in terms of their understandability, modifiability, Scalability and Level of Complexity. The paper further developed three more multivariate models for maintainability, scalability and level of complexity. These three models use design level object-oriented metrics of software systems, to quantify maintainability, Scalability and Level of Complexity of class diagram. The models for Understandability and modifiability are taken from the previous MEMOOD model. Then, we make a comparison of MEMOOD model and Compound MEMOOD model through regression analysis and it is found that Compound MEMOOD Model gives better results with the given dataset. All these models have been validated through appropriate statistical measures.

1. Introduction

The ever-changing world makes maintainability a strong quality requirement for the majority of software systems. The maintainability measurement during the development phases of object-oriented system estimates the maintenance effort, and also evaluates the likelihood that the software product will be easy to maintain [1]. The maintainability is defined by IEEE standard glossary of Software Engineering as "the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment". Despite the fact that software maintenance is an expensive and challenging task, it is not properly managed and often ignored. As class diagrams play a key role in the design phase of object-oriented software therefore early estimation of their maintainability may help designers to incorporate required enhancements and corrections in order to improve their maintainability and consequently the maintainability of the final software to be delivered in future. Hence, there is a need of developing a maintainability estimation model, which quantifies the maintainability of object-oriented software at the design stage.

2. Literature Survey

Wide range of maintainability prediction models have been proposed in the literature within last two decades. Some of the models are predicting maintainability using the metrics from coding as well as design phase, while some are focusing only on design level metrics [3]. Anton Ellis et al. [4], proposed a method of mapping object oriented source code metrics onto the sub-characteristics of maintainability mentioned in ISO 9126. Oman and Hagemeister [5], proposed the Maintainability Index (MI) that objectively determines the maintainability of software system based upon the status of the source code. Welker and Oman [6], suggested measuring maintainability in terms of cyclomatic complexity, lines of code(LOC) and lines of comments. Hayes et al. [7], proposed a model that estimates Adaptive software maintenance effort in terms of difference lines of code (DLOC) i.e. number of added, deleted and updated lines. Polo et al. [8], used number of modification requests, mean effort per modification request and type of correction to examine maintainability. In another study Hayes and Zhao [9], proposed a maintainability model that categorizes software modules as 'easy to maintain' and 'not easy to maintain'. The model helps the developers to identify the modules those are not easy to maintain, before integrating them. From the survey of literature it has been observed that various researchers proposed several models for maintainability estimation, but in most of these studies, maintainability estimation depends on the measures taken after the coding phase. Because of this, maintainability predictions are made in the latter stages of SDLC, and it became very difficult to improve the maintainability at that stage. Muthanna et al. [10], developed a maintainability model using polynomial linear regressions. But this model could be applied only for procedural software and not suitable for object-oriented software. Genero et al. [11], developed four models that relate size and structural complexity metrics of UML class diagrams with maintainability measures like understandability time, modifiability correctness and modifiability completeness. But none of the four models quantify the maintainability of class diagrams itself. Earlier MEMOOD model was developed which finds the maintainability of the class diagrams on the basis of Understandability and Modifiability on the basis of object oriented metrics of class diagrams [12].

3 Metrics Selection

Metric Selection is very objective in nature. There are several ways in which these metrics can be picked up based on which a particular model can be developed. The goal of metric selection is to select such metrics which are statistically significant, relevant in coherent context of Object Oriented Programming for developing following models:

- **Maintainability Model**
- **Understandability Model**
- **Modifiability Model**
- **Scalability Model**
- **Complexity Model**

For each model, we have to do extensive research to find out which metric or which measure of attribute of software will be highly relevant for the above said models. For this purpose, recent literature survey as well as cross and within company data set was chosen with help of experts and their performance, measurable expects for developing such model were studied. Here is list of matrices for each model.

Table 1 Modifiability Model

Metric Name	Metric Definition
Number of classes	The total number of classes
Number of Generalizations (NGen)	The total number of generalization relationship within a class diagram
Number Of Generalizations Hierarchies (NGenH)	The total number of generalization hierarchies within a class diagram
Maximum DIT	It is the maximum DIT value obtained for each class diagram. The DIT value for class is the longest path from the class to the root of the tree. value obtained for each class diagram. The DIT value for class is the longest path from the class to the root of the tree.
Number Of Aggregation Hierarchies (NAggH)	The total number of aggregation hierarchies within a class diagram

Table 2.Understandability Model

Metric Name	Metric Definition
Number of classes	The total number of classes
Number Of Aggregation Hierarchies (NAggH)	The total number of aggregation hierarchies within a class diagram

Table 3. Scalability Model

Metric Name	Metric Definition
Afferent Coupling (Ca)	The number of types inside this assembly that depends upon types inside this assembly.
Effernt Coupling (Ce)	The number of types inside this assembly that depends upon types outside this assembly.
% Coverage(PC)	The percentage of code coverage by tests.
Distance From Main sequence	It is the predictor of balance between abstractness and stability.
Nesting Depth	It is the maximum number of encapsulated scopes inside the body of the method.

Table 4. Level of Complexity Model

Metric Name	Metric Definition
Coupling	The number of types inside this assembly that depends upon types inside this assembly.
Cohesion	The number of types inside this assembly that depends upon types outside this assembly.
Cyclomatic Complexity	Number of decisions that can be taken in procedure.
ILCC	It is the predictor of balance between abstractness and stability.

4. Models Development

Quantification of class diagram's understandability andmodifiability is prerequisite for the maintainability estimation model. Therefore before developing MEMOOD, the paper has developed two models for understandability and modifiability. In order to establish all the three models following multivariate linear model (1) has been selected.

$$Y=\mu+\beta1*X1+\beta2*X2+\ldots\ldots\ldots+ \beta n*Xn+ \varepsilon \qquad (1)$$

4.1 Modifiability Model

In order to establish a multivariate model for modifiability of class diagram, metrics listed in Table 1, will play the role of independent variables while modifiability will be taken as dependent variable. To identify metrics those are effectively contributing in the prediction of modifiability, the technique of backward stepwise multiple regression has been used. This procedure starts with a model, which initially includes all the independent variables and gradually eliminates those, one after another, that does not explain much of the variation in the dependent variable, until it ends with an optimal set of independent variables. Now applying backward stepwise regression, on the available data has resulted into the following modifiability model (2). This model has been taken from MEMOOD Model [13].

$$\textbf{Modifiability}=0.629+0.471*NC–0.173*NGen –0.616*NAggH–0.696*NGenH+0.396*MaxDIT \qquad (2)$$

Where, NC is the 'Number of Classes', NGen is 'Number of Generalizations', NAggH is 'Number of Aggregation Hierarchies', NGenH is 'Number of Generalization Hierarchies' andMaxDIT is Maximum DIT. From the model it can be interpreted that modifiability of class diagram is DIT', while 'NGen' and 'Number of directly proportional to 'Number of Classes' and 'Maximum Generalization and Aggregation Hierarchies' are inversely proportional to modifiability of class diagram.

4.2 Undersatandability Model

After establishing a model for modifiability the next task is to build a similar model for understandability also. Applying the same technique of stepwise backward multiple regressions on the available data resulted into the following understandability model (3).

$$\textbf{Understandability}= 1.166+0.256*NC– .0.394*NGenH \qquad (3)$$

Where, NC is the 'Number of Classes' and NGenH is 'Number of Generalization Hierarchies'. From (3) it could be interpreted that understandability of class diagram is directly proportional to 'NC', while 'NGenH' is inversely proportional to the understandability of class diagram.

4.3 Scalability Model

In order to establish a multivariate model for Scalability of class diagram, metrics listed in Table 3, will play the role of independent variables while Scalability will be taken as dependent variable.

$$\textbf{Scalability}=2.182+.099*AC+.100*EC+.O97*ND- .036*PC+.068*DMS \qquad (4)$$

Where AC is 'Afferent Coupling, EC is efferent Coupling, ND is 'Nesting depth, PC is % Coverage and DMS is Distance from main sequence.

4.3.1 Statistical Significance of the Model

Observing the significance for the F-test in the last column of Analysis of Variance (Table 5), it can be concluded that the Scalability model (4) is statistically significant at a confidence level of more than 99%.

Table 5. Anova for Scalability Model

	Sum Of Squares	DF	Mean Square	F	Significance
Regression	211.772	5	42.354	5.005E5	.000[a]
Residual	.002	22	.000		
Total	211.774	27			

Predictors: (Constant), DMS, PC, AC, EC, ND

Also the value of R2 (Coefficient of Determination) and Adjusted R2 in the Table 6, is also very encouraging. As, it refers to the percentage or proportion of the total variance in modifiability by all the five metrics (independent variables) participating in the model (4).

Table 6. Model Summary For Scalability Model

R	R Square	Adjusted R Square	Std. Error of the Estimate
1.000[a]	1.000	1.000	.009

4.3.2 Statistical significance of the Independent Variables

As long as statistical significance and relevance of individual independent variables in the Scalability model (4) is concern. It can be noticed from the last column of Table 7, that each of the five metrics participating in the model is statistically significant at a significance level of 0.05 (equivalent to a confidence level of 95%).

Table 7. Coefficients and Statistical Significance of Independent variables

	B	Std. Error	Beta	Significance
(Constant)	2.182	.030	73.089	.000
AC	.099	.001	76.290	.000
EC	.100	.001	84.038	.000
ND	.097	.004	25.379	.000
PC	-.036	.032	-1.138	.267
DMS	.068	.009	7.733	.000

Dependent Variable: Scalability

4.4 Level of Complexity Model

In order to establish a multivariate model for Complexity of class diagram, metrics listed in Table 3, will play the role of independent variables while Scalability will be taken as dependent variable.

LOC=.269+.008*Coupling+.181*cohesion+.119*CC +.084*ILCC (5)

Where ILCC is IL Cyclomatic Complexity and CC

iscyclomatic Complexity.

4.4.1 Statistical Significance of the Model

Observing the significance for the F-test in the last column of Analysis of Variance (Table 5), it can be concluded that the modifiability model (5) is statistically significant at a confidence level of more than 99%.

Table 8.Anova For Loc Model

	Sum Of Squares	DF	Mean square	F	Sig
Regression	147.022	4	36.755	1.296	.000
Residual	.027	23	.001		
Total	147.049	27			

Also the value of R2 (Coefficient of Determination) and Adjusted R2 in the Table 9, is also very encouraging. As, it refers to the percentage or proportion of the total variancein modifiability by all the five metrics (independent variables) participating in the model (5).

Table 9. Model Summary

	R Square	Adjusted R Square	Std. Error of the Estimate
1.000[a]	1.000	1.000	.0340099

4.4.2 Statistical Significance of Independent Variables

As long as statistical significance and relevance of individual independent variables in the Level Of Complexity model (5) is concern. It can be noticed from the last column of Table 10, that each of the five metrics participating in the model is statistically significant at a significance level of 0.05 (equivalent to a confidence level of 95%).

Table 10. Coefficients and Statistical Significance Of Independent variables

	B	Std. Error	Beta	Significance
(Constant)	.269	.025	10.744	.000
Coupling	.008	.032	.251	.804
Cohesion	.181	.032	5.725	.000
CC	.119	.029	4.169	.000
ILCC	.084	.026	3.268	.003

Dependent variable: Level Of complexity

4.5 Compound MEMOOD Model

In order to establish a multivariate model for Maintainability of class diagram, Scalability, Level Of Complexity, Understandability and Modifiability of class diagrams becomes in dependent variables while maintainability will be taken as dependent variable.

Maintenance=2.399+.493*Modifiability+.474*Understandability+.524*scalability+.507*LOC (6)

4.5.1 Statistical significance of the Model

Observing the significance for the F-test in the last column of Analysis of Variance (Table 11), it can be concluded that the compound MEMOOD model (6) is statistically significant at a confidence level of more than 99%.

Table 11. Anova For Compound MEMOOD Model

	Sum Of Squares	DF	Mean square	F	Sig
Regression	356.331	4	89.083	1.296E	.0000
Residual	.158	23	.007		
Total	356.489	27			

Also the value of R2 (Coefficient of Determination) and Adjusted R2 in the Table 12, is also very encouraging. As, it refers to the percentage or proportion of the total variance in modifiability by all the five metrics (independent variables) participating in the model (6).

Table 12. Model Summary For Compound MEMOOD Model

R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1.000	1.000	.999	.08291	1.570

4.5.2 Statistical Significance of Independent Variables

As long as statistical significance and relevance of individual independent variables in the compound MEMOOD model (6) is concern. It can be noticed from the last column of Table13, that each of the five metrics participating in the model is statistically significant at a significance level of 0.05 (equivalent to a confidence level of 95%)

Table 13. Coefficients and statistical Significance of independent variables

	B	Std. Error	Beta	Significance
(Constant)	2.399	.048		.000
Modifiability	.493	.018	.250	.000
Understandability	.474	.047	.136	.000
Scalability	.524	.014	.404	.000
LOC	.507	.012	.325	.000

4.6 MEMOOD MODEL

In order to establish a multivariate model for Maintainability of class diagram, Scalability, Level of Complexity, Understandability and Modifiability of class diagrams becomes independent variables while maintainability will be taken as dependent variable

Maintainability = -0.126 + 0.645* Understandability +0.502*Modifiability (7)

TABLE 14: Model Summary For MEMOOD Model

R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
.982	.965	.962	.2947681	1.895

Also the value of R2 (Coefficient of Determination) and Adjusted R2 in the Table, is also very encouraging. As, it refers to the percentage or proportion of the total variance in modifiability by all the five metrics (independent variables) participating in the model (7).

5. Results And Discussion

With Comparison of MEMOOD Model and Compound MEMOOD Model, it is found that Compound MEMOOD Model have R Square value equals to 1 which states that it best fits the data. MEMOOD Model doesn't have R Square value equals 1. Also the value of Durbin Watson variable of compound MEMOOD has value less than that of MEMOOD Model which is considered good. Thus Compound MEMOOD Model has better results than MEMOOD Model. Hence it is better.

Table 15. Performance Results of MEMOOD Model and Compound MEMOOD Model

	COMPOUND MEMOOD MODEL	MEMOOD MODEL
Maintenance value of R Square	1.00	.962
Durbin Watson	1.570	1.895

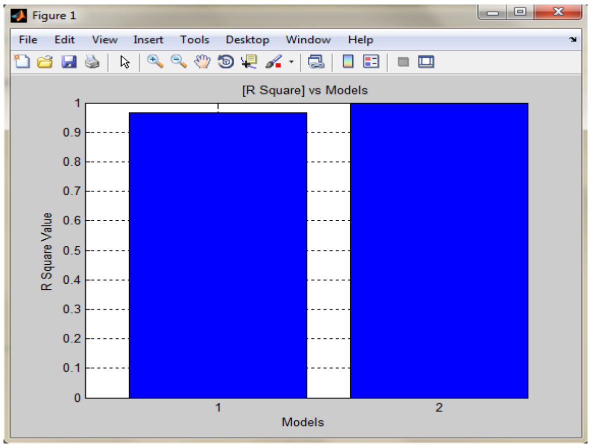

Fig. 5.1. Comparison of models on the basis of R-Square value

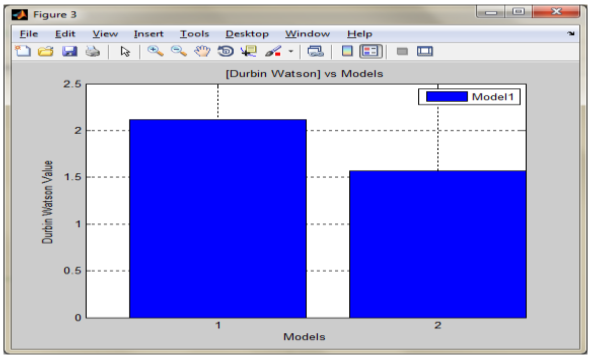

Fig. 5.2. Comparison of models on the basis of Durbin-Watson value

5.1 Comparison of Models on attribute basis

A comparison between models is done on the basis of attributes of models. A comparison is made on the basis of modifiability, Understandability, Scalability and Level of Complexity which shows that compound MEMOOD Model gives better results.

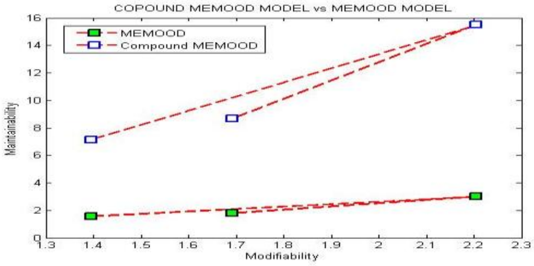

Fig. 5.3 Comparison of MEMOOD Model and compound MEMOOD Model on modifiability

The above Fig. 3 shows the comparison of MEMOOD MODEL and Compound MEMOOD Model on the basis of modifiability which shows that Compound MEMOOD Model has higher chances of modifiability. Thus it is easy to modify and there are more chances of maintenance.

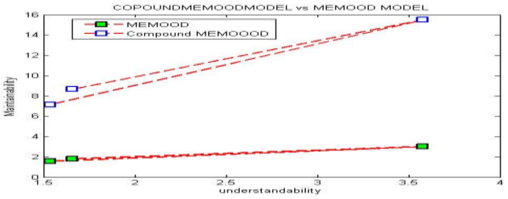

Fig. 5.3 Comparison of MEMOOD Model and compound MEMOOD Model on understandability

The above Fig. 2 shows the comparison of MEMOOD MODEL and Compound MEMOOD Model on the basis of understandability which shows that Compound MEMOOD Model has higher chances of understandability. Thus it is easy to modify and there are more chances of maintenance.

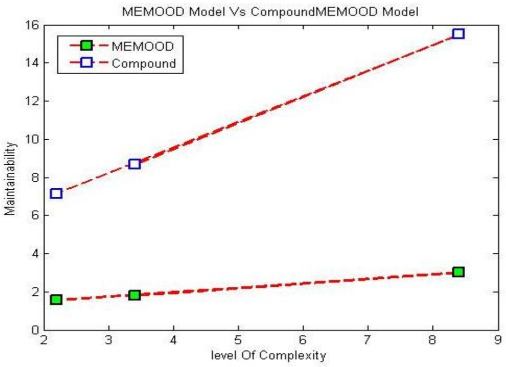

Fig. 5.4 Comparison of MEMOOD Model and compound MEMOOD Model on Level of Complexity

The above Fig. 3 shows the comparison of MEMOOD MODEL and Compound MEMOOD Model on the basis of level Of Complexity.

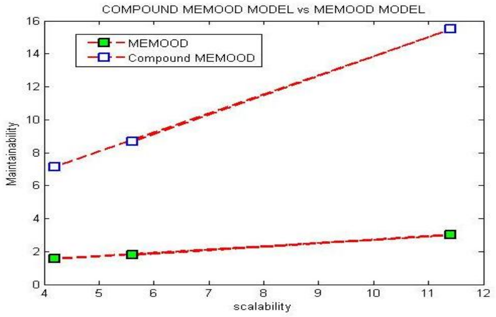

Fig.5.5 Comparison of MEMOOD Model and compound MEMOOD Model on Scalability

The above Fig. 4 shows the comparison of MEMOOD MODEL and Compound MEMOOD Model on the basis of scalability.

6. Conclusion

The paper has developed three models to quantify understandability, modifiability, scalability, Level of Complexity and maintainability of the class diagrams. Maintainability model (Compound MEMOOD) estimates the maintainability of class diagrams in terms of their understandability, modifiability, scalability and Level of Complexity. In [12], MEMOOD Model is introduced. With Comparison of MEMOOD Model and Compound MEMOOD Model, it is found that Compound MEMOOD Model have R Square value equals to 1 which states that it best fits the data. Also the value of Durbin Watson variable of compound MEMOOD has value less than that of MEMOOD Model which is considered good. From the diagrammatical analysis it is found that all the attributes of maintainability has better results for Compound MEMOOD Model than MEMOOD Model. Thus Compound MEMOOD Model has better results than MEMOOD Model. Hence it is better.

6. References

[1] V.S. Alagar, L.Qiaoyum, and O.S. Ormandjieva, “Assessment of Maintainability in Object-Oriented Software,” *Proc. 39th International Conference and Exhibition on Technology of Object-Oriented Languages andSystems (TOOLS39)*, 29 July -03 Aug. 2001, vol. 19, pp. 194 – 204, Santa Barbara, Callifornia, USA, 2001.

[2] M. Dagpinar and J. Jahnke, “Predicting Maintainability with Object-Oriented Metrics – an Empirical Comparison,” *Proc10th Working Conference on Reverse Engineering (WCRE’03),*13 - 17 Nov. 2003, pp. 155-164, 2003.

[3] S.W.A. Rizvi and R.A. Khan, “A Critical Review on Software Maintainability Models,” *Proc. of the National Conference on Cutting Edge Computer and Electronics Technologies*, 14 - 16 Feb. 2009, pp. 144 – 148,Pantnagar, India, 2009.

[4] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis, and N. Tsirakis, “A Data Mining Methodology for Evaluating Maintainability According toISO/IEC-9126 Software Engineering Product Quality.Standard,” *Proc. 11th IEEE Conference onSoftware Maintenance and Reengineering (CSMR2007)*, 21 – 23 Mar.2007, Amsterdam, Netherlands, 2007.

[5] P.W. Oman and J.R. Hagemeister, “Construction and Testing of Polynomials Predicting Software Maintainability,” *Journal of Systems and Software*, vol. 24, no. 3, pp. 251- 266, 1994.

[6] K.D. Welker and P.W. Oman, “Software Maintainability Metrics Models in Practice,” *Journal of Defense Software Engineering*, vol. 8, no. 11, pp. 19 - 23, 1995.

[7] J.H. Hayes, S.C. Patel, and L. Zhao, “A Metrics-Based Software Maintenance Effort Model,” *Proc. 8th European Conference on Software Maintenanceand Reengineering*

(CSMR'04), 24 – 26 Mar. 2004, pp. 254 – 258,IEEE Computer Society, 2004.

[8] M. Polo, M. Piattini, and F. Ruiz, "Using Code Metrics to Predict Maintenance of Legacy Programs: a Case Study," *Proc. of InternationalConference on Software Maintenance, ICSM 2001*, pp. 202-208, IEEE Computer Society, Florence Italy, 2001.

[9] J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," *Proc. 21st IEEE InternationalConference on Software Maintenance*, 26 - 29 Sept. 2005, pp. 601 -604, 2005.

[10] S. Muthanna, K. Kontogiannis, K. Ponnambalam, and B. Stacey, "A Maintainability Model for Industrial Software Systems Using Design Level Metrics," *Proc. 7th Working Conference on Reverse Engineering(WCRE'00)*, 23 - 25 Nov., 2000, pp. 248 – 256, Brisbane, Australia, 2000.

11] M. Genero, E. Manso, and G. Cantone, "Building UML Class Diagram Maintainability Prediction Models Based on Early Metrics," *Proc. 9th International Symposium on Software Metrics (METRICS'03)*, 3 - 5 Sept., 2003, pp. 263 - 275, 2003.

[12] S.W.A. Rizvi and R.A. Khan "Maintainability Estimation Model for Object- Oriented Software in Design Phase (MEMOOD)".