A Maintainability Estimation Model and Tool

Alisara Hincheeranan and Wanchai Rivepiboon

Abstract—**A measuring maintainability of software system in the design phase, may help a software designer must improves the maintainability of software system before deliver to a customer. In this paper presents a multivariate linear regression for establish the maintainability estimation model in terms of flexibility and extendibility are sub-characteristics of maintainability as criteria to evaluate maintainability model of class diagram and develop the Maintainability Estimation Tool (MET) for a maintainability estimation of class diagram. This tool help a software designer for improves the maintainability of class diagram in design phase and help reduces the increasing high cost of software maintenance phase.**

Index Terms—**Software metrics, software maintainability, maintainability model, object-Oriented design, object-oriented qality attribute.**

I. Introduction

In demand of software quality, efficiency and reliability increased in recent years. The quality of a software system can be measured by using a quality attribute of software system. The maintainability is a quality factor with influence in the software maintenance phase. Many researchers reported that 50-70% of the total life cycle is spent on software maintenance phase can provided earlier feedback to help a software designer improved the quality of software systems and reduced the increasing high cost of software maintenance phase [1], [3]. 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*". Many researchers [8] presented sub-characteristics: reusability, usability, reliability, flexibility, testability, extendibility are affect with a maintainability factor. This paper proposes flexibility and extendibility are sub-characteristics of maintainability as criteria to evaluate maintainability model of class diagram. While flexibility and extendibility calculate from object-oriented design metrics.

Flexibility is a characteristic allow the incorporation of changes in a design. The ability of a design to be adapted to provide functionally related capabilities [5].

Extendibility refers to the presence and usage of properties in an existing design that allow for the incorporation of new requirements in the design [2].

The Unified Modeling Language (UML) has been proposed as a standard language for expressing object oriented software designs and used in the development of software systems. It provides a range of object oriented diagrammatic notations for expressing the structural and behavioral aspects of software systems. The class diagram has important in the software design phase for early estimation of maintainability may help a software designer for improves a software design and corrections [4].

This paper is organized as follows: Section 2 describes the related works for establish the maintainability estimation model. Section 3 explains the research methodology. Section 4 some conclusion of the maintainability estimation model and tool. Section 5.References.

Manuscript received January 16, 2012; revised March 8, 2012.

Alisara Hincheeranan is with the Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand (e-mail: Alisara184@gmail.com).

Wanchai Rivepiboon is with the Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand (e-mail: Wanchai.R@chula.ac.th).

II. Related Works

Kiewkanya *et al.*[3], presented a maintainability model of class diagram using three techniques viz. Discriminate technique, Weighted-Score-Level technique, Weighted-Predicted-Level technique and two sub-characteristics of maintainability: understandability and modifiability for evaluate maintainability model. Rizvi et al. [6], developed a multivariate linear model ‘Maintainability Estimation Model for Object-Oriented software in design phase (MEMOOD)’ for estimate the maintainability of UML class diagram in terms of understandability and modifiability. Gautam *et al.*[9], developed a multivariate linear model ‘Compound Maintainability Estimation Model for Object-Oriented software in design phase (Compound MEMOOD)’ for estimate the maintainability of class diagram in terms of understandability, modifiability, scalability and level of complexity. Tong Yi [7], compared the advantages and disadvantage of class diagram complexity metrics based on statistics and entropy-distance in term of understandability, analyzability and maintainability.

III. Research Methodology

A. The Maintainability Estimation Model

In this section selects quality model for object-oriented design. Bansiya et al. [2], Extended the Dromey’s generic quality model to propose a hierarchical Quality Model for Object-Oriented Design (QMOOD). As show in Fig. 1, there are four levels of three mapping between levels in QMOOD. The definition of levels in QMOOD involves identify design quality attributes, object oriented design properties, object design metrics, and object oriented design components. The QMOOD is useful to assess the object oriented design quality. The QMOOD has identified a six quality factors that are functionality, effectiveness, understandability, reusability, flexibility and extendibility following the set of ISO 9126 quality attributes and proposed as a set of eleven design properties in the Table 1, show a design property definition that are Design Size, Hierarchies, Abstraction, Encapsulation, Coupling, Cohesion, Composition, Inheritance, Polymorphism, Messaging, Complexity, and a mathematical formulas in the Table 2, show a design metrics for maintainability estimation model.

This paper selects a quality factors there are flexibility and extendibility are sub-characteristics of maintainability as a criteria for establish a maintainability estimation model. In Fig. 2, describes the quantification process of the maintainability estimation model. The flexibility and extensibility calculate from Computation Formulas for Quality Attribute in the Table 3, show a computation formula for calculating a flexibility and extendibility proposed by Bansiya et al. [2]. In order to establish a maintainability estimation model following multivariate linear regression (1) has selected.

$$Y = \mu + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + B_n * X_n + \epsilon \quad (1)$$

Y is dependent variable.
$X_1, X_2, \ldots, X_n$ are independent variables.
$\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients of independent variables.
ϵ is error term.
μ is the intercept.

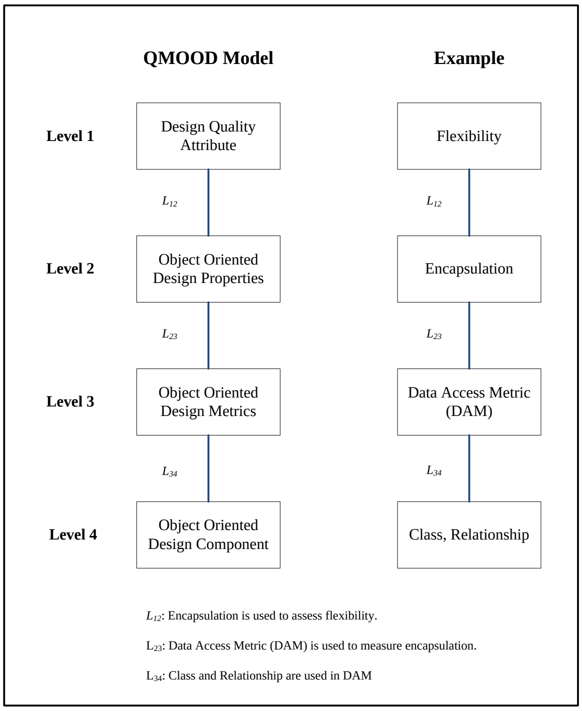

Fig. 1. The QMOOD Model and an Example.

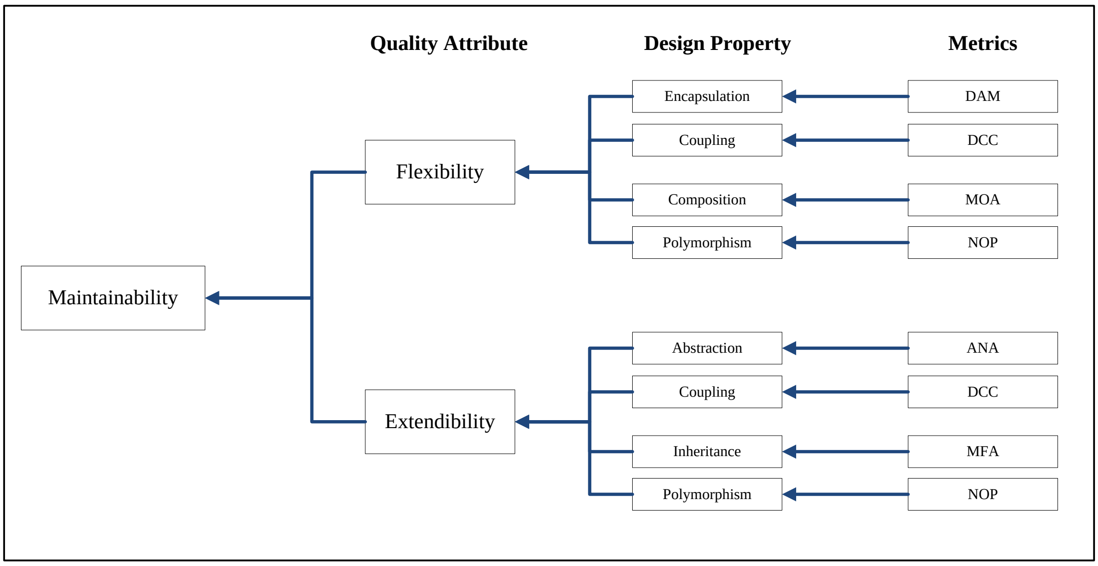

Fig. 2. The structure of the Maintainability Estimation Model.

TABLE 1: DESIGN PROPERTY DEFINITIONS [2].

Design Property	Metrics
Design Size	A measure of the number of class used in a design.
Hierarchies	Hierarchies are used to represent different generalization-specialization concepts in a design. It is a count of the number of non-inherited class that has children in a design.
Abstraction	A measure of the generalization-specialization aspect of the design. Classes in a design which have one or more descendants exhibit this property of abstraction.
Encapsulation	Defined as the enclosing of data and behavior within a single construct. In object-oriented designs the property specifically refers to designing classes that prevent access to attribute declarations by defining them to be private, thus protecting the internal representation of the objects.
Coupling	Defines the interdependency of an object on other objects in a design. It is a measure of the number of other object that would have to be accessed by an object in order for that object to function correctly.

Cohesion	Assesses the relatedness of methods and attributes in a class. Strong overlap in the method parameters and attributes types is an indication of strong cohesion.
Composition	Measures the "part-of","has","consists-of" or "part-whole" relationships, which are aggregation relationships in an object-oriented design.
Inheritance	A measure of the "is-a" relationship between classes. This relationship is related to the level of nesting of classes in an inheritance hierarchy.
Polymorphism	The ability to substitute objects whose interface match for one another at run-time. It is a measure of service that is dynamically determined at run-time in an object.
Messaging	A count of the number of public methods that is available as services to other classes. This is a measure of the services that a class provides
Complexity	A measure of the degree of difficulty in understanding and comprehending the internal and external structure of classes and their relationships.

TABLE 2: DESIGN METRICS FOR DESIGN PROPERTIES [2].

Design Property	Metrics
Design Size	Design Size in Class (DSC)
Hierarchies	Number of Hierarchies(NOH)
Abstraction	Average Number of Ancestors (ANA)
Encapsulation	Data Access Metric (DAM)
Coupling	Direct Class Coupling (DCC)
Cohesion	Cohesion Among Methods in Class (CAM)
Composition	Measure of Aggregation (MOA)
Inheritance	Measure of Functional Abstraction (MFA)
Polymorphism	Number of Polymorphic Methods (NOP)
Messaging	Class Interface Size (CIS)
Complexity	Number of Methods (NOM)

TABLE 3: COMPUTATION FORMULAS FOR QUALITY ATTRIBUTE [2].

Quality Attribute	Index Computation
Flexibility	0.25 * Encapsulation – 0.25 * Coupling +0.5 * Composition + 0.5 * Polymorphism
Extendibility	0.5 * Abstraction – 0.5 * Coupling + 0.5 * Inheritance + 0.5 Polymorphism

B. *The Maintainability Estimation Tool (MET).*

In this section present a structures of the Maintainability Estimation Tool (MET) for evaluate maintainability of class diagram in terms of flexibility and extendibility. This tool automates the collection of data from class diagram by parsing the XMI file format of class diagram and uses these data to calculate the object- oriented designs metrics there are Design Size, Hierarchies, Abstraction, Encapsulation, Coupling, and Cohesion etc. Fig. 3, show a structure of the Maintainability Estimation Tool (MET) consists of the four components.

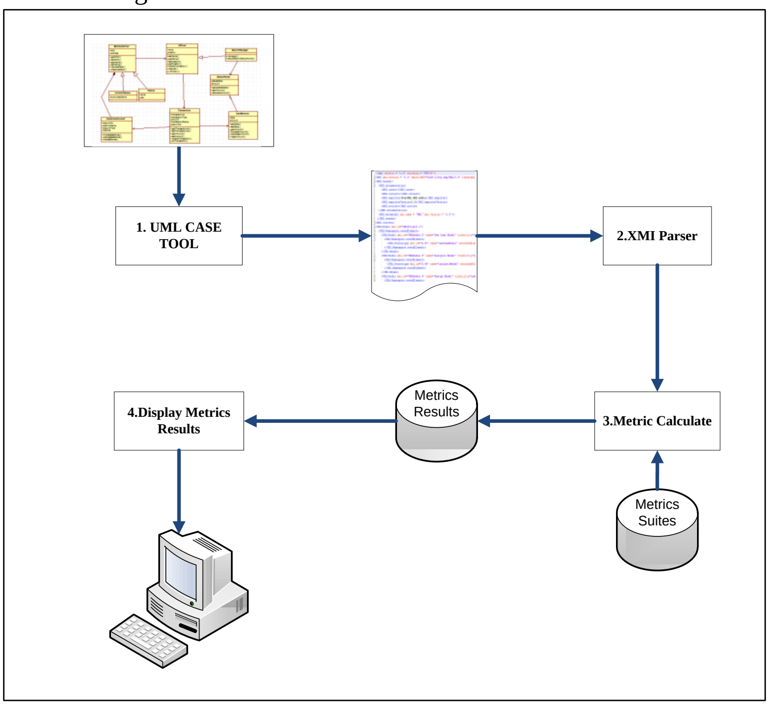

Fig. 3. A structure of the Maintainability Estimation Tool (MET).

- **UML Case Tool:** This component for develop class diagram is a design of software system (In this paper use the StarUML is an open source project to develop class diagram [10].) and export into XMI file format is an input file for calculate a maintainability estimation of class diagram.
- **XMI Parser:** This component for prepare a data for establish a maintainability estimation of class diagram by read element and attribute of XMI (XMI: XML Metadata Interchange (XMI) is an open standard file format that enables **the** interchange of model information between models and tools [11].)
- **Metric Calculate:** This component for calculate a maintainability estimation of class diagram by using the object- oriented design metric suites there are Design Size **in** class (DSC), Number of Hierarchies (NOH), Average Number of Ancestors (ANA), Data Access Metric (DAM), Direct Class Coupling (DCC), Cohesion Among Methods in Class (CAM), Measure of Aggregation (MOA), Measure of Functional Methods (NOP), Class Interface Size (CIS), Number of Methods (NOM) and computation formulas for quality attributes of flexibility and extendibility.
- **Display Metrics Results:** This component for display data as a set of eleven design properties, set of two quality **attributes** there are flexibility and extendibility and a maintainability estimation of class diagram. It consists of the three components.
  1) **Design Properties Results:** This component for display data of the Design properties eleven elements design size, hierarchies, abstraction, encapsulation, coupling, cohesion etc.
  2) **Quality Attribute Results:** This component for display data of flexibility and extendibility by using a **computation** formula for Quality Attributes.
  3) **Maintainability Estimation Results:** This component for display data of the maintainability **estimation** for a software designer improves a software system before deliver to a customer.

IV. Conclusion

In this paper presents a multivariate linear regression for establish the maintainability estimation model and develop the Maintainability Estimation Tool (MET) for a maintainability estimation of class diagram in design phase. This tool for help a software designer for evaluates maintainability of software system early in software development life cycles (SDLC), improves the maintainability of class diagram in design phase and reduces the increasing high cost of software maintenance phase.

References

[1] P. Oman and J. Hagemeister, "Metrics for assessing a software system's maintainability," *Software Maintenance*, 1992, pp. 337 - 344.

[2] J. Bansiya and C. G. Davis, "A hierarchical model for object-oriented design quality assessment," *IEEE Transaction on software engineering*, vol. 28, pp. 4-17, 2002.

[3] M. Kiewkanya, N. Jindasawat, *et al*., "A methodology for constructing maintainability model of object-oriented design," *Quality Software QSIC 2004 Proceedings Fourth International*, 2004, pp. 206 - 213.

[4] T. Yi, F. Wu, *et al.*, "A comparison of metrics for UML class diagrams," *SIGSOFT Softw. Eng. Notes* vol. 29, no. 5, pp. 1-6.

[5] S. R. Ragab and H. H. Ammar, "Object oriented design metrics and tools a survey," *Informatics and Systems (INFOS)*, 2010 The 7th International, pp. 1-7.

[6] S. W. A. Rizvi and R. A. Khan, "Maintainability Estimation Model for Object- Oriented Software in Design Phase (MEMOOD)," *COMPUTING*, vol. 2, no. 4, pp. 26-32.

[7] T. Yi, "Comparison Research of Two Typical UML-Class-Diagram Metrics: Experimental Software Engineering," *International Conference on Computer Application and System Modeling* , 2012, Taiyuan, vol. 12, pp. 86-90.

[8] S. Ghosh, S. K. Dubey, *et al.*, "Comparative Study of the Factors that Affect Maintainability," *International Journal on Computer Science and Engineering*, vol. 3, no. 12, pp. 3763-3769, 2011.

[9] C. Gautam and S. S. Kang, "Comparison and implementation of compound memood model and memood model," *International Journal of Computer Science and Information Technologies* vol. 2, no. 5, pp. 2394-2398, 2011.

[10] GPL (2011). StarUML. [Online]. Available: http://staruml.sourceforge.net/en/index.php.

[11] Sparxsystems. (2012). XMI Import and Export. [Online]. Available: http://www.sparxsystems.com/enterprise_architect_user_guide/projects_and_teams/importexport.html.