Journal of Computer Applications (JCA)
ISSN: 0974-1925, Volume V, Issue 2, 2012

A Systematic Review of Reusability Assessment Model and Related Approach for Reusable Component Mining

V. Subedha[a,*], Dr. S. Sridhar[b,1]

Abstract **- Promoting reuse at sophisticated levels is becoming an increasingly important part. Software reuse is the process of reusing the existing software components from the system rather than developing the components newly. As software reuser's include the reusable software components to improve the productivity and quality in their product, they must be able to measure the reusability degree and effective reuse strategies. This measure is done with the help of reuse metrics and models. Reusability assessment modeling is helpful in evaluating the quality of reusable software components in an environment. In this paper we review reusability assessment model and different approach in the existing literature which helps the reuser for mining the suitable component in terms of reusability.**

Index Terms - Reuse metrics, Reusable Components, Reusability metrics, Software reusability, Reuse frequency, Reusability assessment model.

I. INTRODUCTION

As software engineering matures into a true engineering discipline, there is an increasing need for a corresponding maturity in repeatability, assessment, and measurement of the artifacts [1] associated with software. Software reuse, the use of existing software artifacts or asset to create new software, is a key method for improving software quality and productivity. When an organization decide to implement systematic software reuse programs to improve productivity and quality of the system development, they must be able to measure the quality of the reusable software components and identify the most effective reuse strategies. This is done with reuse metrics and models.

Manuscript received 15/March/2012.
Manuscript received 21/May/ 2012.

V. Subedha,[a,*],
Research Scholar,
Sathyabama University, Chennai, India,
09841473130
E-mail: subedha@gmail.com
Dr. S. Sridhar,[b,1]
Research Supervisor,
Sathyabama University, Chennai, India,
E-mail: drssridhar@yahoo.com

Metrics and models of software reuse and reusability are classified into six types [2]:

- Cost-benefit models
- Maturity assessment models
- Amount of reuse metrics
- Failure modes models
- Reusability assessment models
- Reuse library metrics

A metric is a quantitative indicator of an attribute of a thing. A model specifies relationships among metrics. The aim of metrics is to predict the quality of the software products. The requirement today is to relate the reusability attributes with the metrics and to find how these metrics collectively determine the reusability of the software component.

Reuse cost-benefits models include economic cost and benefit analysis. Maturity assessment models categorize reuse programs by how advanced they are in implementing systematic reuse. Amount of reuse metrics are used to assess and monitor a reuse improvement effort by tracking percentages of reuse for life cycle objects. Failure modes analysis is used to identify and order the impediments to reuse in a given organization. Reusability metrics indicates that a component is reusable. Reuse library metrics are used to manage and track usage of a reuse repository

The identified five programs attributes for evaluating reusability [2]. The attributes used are:

- Program Size
- Program Structure
- Program Documentation
- Programming Language
- Reuse Experience

This paper presents a review on the reusability assessment models. The study throws the light on various dimensions of reusability assessment models. Categorization of models is based on different approach for evaluation. Section II defines the framework of reusability assessment model. Section III includes a brief review of software reusability assessment models. Section IV we present comparisons among reusability assessment models. At last this paper concludes in section V.

II. FRAMEWORK OF REUSABILITY ASSESSMENT MODEL

Another important reuse measurement area concerns the estimation of reusability for a component. Figure 1 shows the framework of reusability assessment model. The component reuse library is a repository for storing reusable assets, plus an interface for searching the repository. Library assets can be obtained from existing systems through reengineering, designed and built from scratch, or purchased. Components then are usually identified, a process for assuring that they have desired functionality based on the requirements of reuser. The quality of the components is another important aspect of reusability. The components are then qualified for reusability so that reuser's can effectively search for them and reuse it.

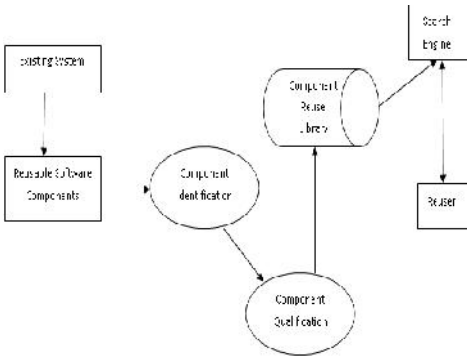

Figure 1. Framework of reusability assessment model

As said, to insert images in *Word*, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with "Float over text" unchecked).

The authors of the accepted manuscripts will be given a copyright form and the form should accompany your final submission.

III. REVIEW OF REUSABILITY ASSESSMENT MODELS

Several works has been done in this area. Some of them are as:

In 2005, Richard W. Selby [3], investigated, analyzed, and evaluated software reusability by mining software repositories from a NASA software development environment that actively reuses software. They analyzed four classes of software modules: modules reused without revision, modules reused with slight revision (< 25 percent revision), modules reused with major revision (> 25 percent revision), and newly developed modules. They identified two categories of factors that characterize successful reuse-based software development of large-scale systems: module design factors and module implementation factors. They also evaluated the fault rates of the reused, modified, and newly developed modules. They concluded that the modules reused without revision had the fewest faults, fewest faults per source line, and lowest fault correction effort. The modules reused with major revision had the highest fault correction effort and highest fault isolation effort as well as the most changes, most changes per source line, and highest change correction effort. The author achieved an average reuse of 32 percent per project, which is the average amount of software either reused or modified from previous systems.

In 2006, Parvinder S. Sandhu and Hardeep Singh [4] proposed reusability evaluation model for assessing the reusability of software components. The authors analyzed the CK metrics and remove the inconsistencies and devised the framework of metrics to evaluate the reusability. Also they proposed Neuro-fuzzy Inference engine can be used to evaluate the reusability. They proposed an algorithm in which the inputs can be given to Neuro-fuzzy system in form of structural attributes and output can be obtained in terms of reusability. The following refined CK metric suit are used in this model are : Tuned Weighted Methods per Class (TWMC), Lack of Tuned Degree of Inheritance (LTDIT), Lack of Tuned Number of Children (LTNOC), Lack of Coupling between Objects (LCBO), Lack of Cohesion in Methods (LCOM).

In 2007, Parvinder S. Sandhu Pavel Blecharz and Hardeep Singh [5] proposed Quantitative Investigation of impact of the factors contribution towards measuring the reusability of software components which helps to evaluate the quality of the components. They used Taguchi approach in analyzing the significance of different attributes in deciding the reusability level of a particular component. They inferred from the results that the complexity is the most important factor in deciding the better reusability of a function oriented software component and in case of object oriented, coupling and complexity collectively play the significant role in high reusability. The proposed metrics for function oriented paradigm are: Cyclometric Complexity Using Mc Cabe's Measure, Halstead Software Science Indicator, Regularity Metric, Reuse-Frequency Metric and Coupling Metric. The proposed metrics for object oriented paradigm are : Tuned Weighted Methods per Class (TWMC), Lack of Tuned Degree of Inheritance (LTDIT), Lack of Tuned Number of Children (LTNOC), Lack of Coupling Between Objects (LCBO) and Lack of Cohesion in Methods (LCOM).

In 2008, GUI GUI and Paul D. Scott[6] proposed new measure of coupling and cohesion to assess the reusability of components. They shown that the new measures proposed by them was consistently superior at the time of measuring the component reusability.

They used five metrics for coupling are : Weighted Transitive Coupling (WTCoup), Coupling Factor (CF), Classes Between Objects (CBO) , Response For Class (RFC), Data Abstraction coupling (DAC) and five metrics for cohesion are : Weighted Transitive Cohesion (WTCoh), RLCOM, Lack of Cohesion in Methods (LCOM), Variant Lack of Cohesion in Methods (LCOM3) and Tight Class Cohesion (TCC). Two approaches were used to evaluate the performance of the various measures in predicting reusability. They are Linear Regression and Rank Correlation. They clearly demonstrated that their proposed metrics Weighted Transitive Coupling (WTCoup) and Weighted Transitive Cohesion (WTCoh) for coupling and cohesion are very good predictors for evaluation the reusability of the component.

In 2009, Parvinder S. Sandhu, Harpreet Kaur and Amanpreet Singh [7] proposed reusability evaluation system for object oriented software components. They proposed software metrics and quality of the software components were inferred by different neural network based approaches. The proposed metrics for object oriented paradigm are : Tuned Weighted Methods per Class (TWMC), Lack of Tuned Degree of Inheritance (LTDIT), Lack of Tuned Number of Children (LTNOC), Lack of Coupling Between Objects (LCBO) and Lack of Cohesion in Methods (LCOM). The different neural network approaches are used for the modeling of the reusability data. The following fourteen neural networks algorithms were experimented by the author. They are

- Batch Gradient Descent without momentum
- Batch Gradient Descent with momentum
- Variable Learning Rate without momentum
- Variable Learning Rate training with momentum
- Resilient Back propagation
- Fletcher-Reeves version of the conjugate gradient
- Polak-Ribiere Update version of the conjugate gradient
- Powell-Beale Restarts version of the conjugate gradient
- Scaled Conjugate Gradient
- Quasi-Newton BFGS Algorithm
- Quasi-Newton One Step Secant Algorithm
- Levenberg-Marquardt Algorithm
- Generalized Regression Neural Networks

In 2010, Sonia Manhas, Rajeev Vashisht, Parvinder S. Sandhu and Nirvair Neeru [8] proposed reusability evaluation model for assessing reusability of software components. They proposed structural attributes and software metrics to evaluate the reusability of the component by experimenting with five different Neural Network based approaches by taking the metric value as input. The calculated reusability value enables to identify the good quality components automatically. The proposed metrics are: Cyclometric Complexity Using Mc Cabe's Measure, Halstead Software Science Indicator, Regularity Metric, Reuse-Frequency Metric, and Coupling Metric. The different neural network approaches are used for the modeling of the reusability data. The following five neural networks algorithms were experimented they author. They are

- Batch Gradient Descent
- Batch Gradient Descent with momentum
- Variable Learning Rate
- Variable Learning Rate training with momentum
- Resilient Back propagation

In 2011, Fazal-e-Amin, Ahmad Kamil Mahmood and Alan Oxley [9] proposed reusability attribute model for assessing reusability of software components. They proposed six attributes related to the reusability of software components. The proposed model is derived using the GQM approach and this helps to understand the factors to measure the software quality. Also they provide the metrics which are used to assess the reusability. The metrics which are used for reusability assessment are: Size Metrics, Coupling metrics, Cohesion metrics and Variability Metrics. The metric used for measuring the size is Lines of Code (LOC) and Number of Method (NOM). The metric used for measuring the coupling is Coupling Between Object Classes (CBO) and metric used for measuring the cohesion is Lack of Cohesion in Methods (LCOM). The variability metrics are based on the theory and mechanism of inheritance. The metrics used to measure variability are Depth of Inheritance Tree (DIT) and Number of Children (NOC the equations used to calculate the attributes value are as following :

Flexibility =1 - [(0.5 X Coupling) + (0.5 X Cohesion)]. Coupling = adjusted CBO, Cohesion = adjusted LCOM. Understandability = 1 - [(0.25 X Coupling) + (0.25 X Cohesion) + (0.25 X Comments) + (0.25 X Size)]. Size = (0.5 X adjusted LOC) + (0.5 X adjusted NOM) . Portability = Independence = 1 - adjusted DIT . Scope coverage = NOM ÷ Total number of methods in all classes . Variability = 0.5 X (NOC ÷ Total number of classes) + 0.5 X (NOM ÷ Total number of methods in all classes). Reusability of Class = 0.16 X Flexibility + 0.16 X Understandability + 0.16 X Portability + 0.16 X Scope coverage + 0.16 X Maintainability + 0.16 X Variability .

In 2011, Nasib S. Gill and Sunil Sikka [10] investigated four existing metrics Depth of Inheritance (DIT), Number of Children (NOC), Method Inheritance Factor (MIF) and Attribute Inheritance Factor (AIF) and also proposed five new metrics- Breadth of Inheritance Tree (BIT), Method Reuse Per Inheritance Relation (MRPIR), Attribute Reuse Per Inheritance Relation (ARPIR), Generality of Class (GC) and Reuse Probability (RP). All inheritance based metrics are classified into two

categories: Reuse Based Metrics (RBM) and Reusability Prediction Metrics (RPM). RBM are further classified into two categories- Reuse Indicator Metrics (RIM) and Reuse Estimation Metrics (REM). Analysis of these metric shows that they can be helpful for assessing reuse and reusability.

In 2012, Ajay Kumar [11] proposed a model for classification of the reusability of software components using support vector machine. The metrics used for identification of reusable software modules are Cyclometric Complexity Using Mc Cabe's Measure, Halstead Software Science Indicator, Regularity Metric, Reuse-Frequency Metric, and Coupling Metric.

IV. COMPARITIVE ANALYSIS AMONG THE MODELS

In this section, we present comparisons among some reusability assessment model based on some factors and reuse metrics. The results of the comparisons are presented in the below table, Table I.

Table I : Comparisons of reusability assessment model and related approach

Models	Authors	Year & Reference	Approach	Attributes	Metrics
Evaluation software reuse empirically by mining software repositories	Richard W. Selby	2005 [3]	Goal Question Metric (GQM)	Module design factors, Module implementation factors, Module fault factors	Percentage of modules reused without revision, Modules reused with slight revision, Modules reused with major revision, Module development effort, Module fault rate, Source lines of code, Cyclomatic complexity, Fault correction rate.
Reusability Evaluation Model	Parvinder S. Sandhu and Hardeep Singh	2006 [4]	Neuro-Fuzzy Inference System		TWMC, LTDIT, LTNOC,LCBO, LCOM
Quantitative Investigation model	Parvinder S. Sandhu, Pavel Blecharz and Hardeep Singh	2007 [5]	Taguchi Approach		Cyclometric Complexity, Volume, Regularity, Reuse-Frequency, Coupling, TWMC, LTDIT, LTNOC,LCBO,LCOM
Evaluation of software component reusability model	Gui Gui and Paul D. Scott	2008[6]	Linear Regression and Rank Correlation	Newly derived Coupling Measures and Cohesion Measures	WTCoup, CF, CBO, RFC, DAC, WTCoh, RLCOM, LCOM, LCOM3, TCC
Reusability Evaluation system	Parvinder S. Sandhu, Harpreet Kaur and Amanpreet Singh	2009[7]	Neural Network Approaches		TWMC, LTDIT, LTNOC,LCBO, LCOM
Reusability Evaluation Model	Sonia Manhas, Rajeev Vashisht, Parvinder S. Sandhu and Nirvair Neeru	2010 [8]	Neural Network Algorithms		Cyclometric Complexity, Volume, Regularity, Reuse-Frequency, Coupling
Inheritance hierarchy Based model	Nasib S. Gill and Sunil Sikka	2011 [9]	Metrics based approach		DIT, NOC, MIF, AIF, BIT, MRPIR, ARPIR, GC, RP
Reusability Attribute Model	Fazal-e-Amin, Ahmad Kamil Mahmood and Alan Oxley	2011 [10]	Goal Question Metric (GQM)	Maintainability, Portability, Flexibility, Understandability, Scope coverage, Independence	LOC, CBO, LCOM, DIT, NOM, NOC
Reusability classification model	Ajay Kumar	2012[11]	SVM classifier		Cyclometric Complexity, Volume, Regularity, Reuse-Frequency, Coupling

V. CONCLUSION

Though significant progress has been made on software reuse, many important problem remains. One important issue is how to make best reusable components for system. A literature review of the works in the area of software reusability assessment model is conducted and the results of the review are presented in this paper. This classification scheme serves as a framework for future research to differentiate between different reusability assessment enabling researchers to test more model in practice. We have consolidated array of existing software reusability model based on the metrics and approach for qualifying the components for reuse. The Table I classify the model established by this study.

Although our review has explored the field of assessment model, further studies are needed to confirm the obtained results. Future work includes the extension of this review by including more sources. Also the future direction is the inclusion of other dimensions to categorize the approach.

In short, to summarize it, some key findings in our review were:

- Reuser's reuse the code but they do not test it to the extent.
- Tools are lacking when it comes to creation and analysis of reusability.
- Lack of methods for qualifying the high quality potential reuse component.

REFERENCES

[1] Z.M. Zhang, Y.T. Zhuang, Y.H. Pan, "Object-Oriented Software Reverse Engineering," *J. of Computer Research and Development*, vol.40, no.7, pp.1062–1068, 2003.

[2] William Frakes, Carol Terry, "Software Reuse : Metrics and Models", ACM Computing Surveys, Vol.28, No.2, pp.416-435, Jun 1996.

[3] Richard W. Selby, "Enabling Reuse-Based Software Development of Large-Scale Systems," IEEE Transaction of Software Engineering, Vol. 31, No. 6, PP. 495-510, Jun 2005.

[4] Parvinder Singh Sandhu and Hardeep Singh, "Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach", International Journal Of Information Technology, vol. 3, no. 3, 2006, pp. 209-214.

[5] Parvinder S. Sandhu Pavel Blecharz and Hardeep Singh, "A Taguchi Approach to Investigate Impact of Factors for Reusability of Software Components", World Academy of Science, Engineering and Technology, pp.135-140, Sep 2007.

[6] Gui Gui and Paul D. Scott, "New coupling and cohesion Metrics for Evaluation of Software Component Reusability", in Proc. ICYCS, 2008, pp.1181-1186.

[7] Parvinder S. Sandhu, Harpreet Kaur and Amanpreet Singh, "Modeling of Reusability of Object Oriented Software System", World Academy of Science, Engineering and Technology Issue. 30, pp. 162-165, Aug 2009.

[8] Sonia Manhas, Rajeev Vashisht, Parvinder S. Sandhu and Nirvair Neeru," Reusability Evaluation Model for Procedure Based Software Systems", International Journal of Computer and Electrical Engineering, Vol.2, No.6, pp. 1107-1110, Dec 2010.

[9] Fazal-e-Amin, Ahmad Kamil Mahmood and Alan Oxley, "Reusability Assessment of Open Source Components for Software Product Lines", International Journal on New Computer Architectures and Their Applications (IJNCAA), 1(3), pp. 519-533, 2011.

[10] Nasib S. Gill and Sunil Sikka, "Inheritance Hierarchy Based Reuse & Reusability Metrics in OOSD", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 6, pp. 2300-2309, June 2011.

[11] Ajay Kumar, "Measuring Software Reusability using SVM based Classifier Approach", International Journal of Information Technology and Knowledge Management.", Vol. 5, No. 1, pp.205-209, Jan 2012.

BIOGRAPHY

V. Subedha, Research Scholar, Sathyabama University, Chennai (SBU), has completed her post graduation in Computer Science and Engineering. Her career spans over 15 years in the field of teaching and in-depth experience in academics. She is currently working as Associate Professor in the Department of CSE, Panimalar Engineering College, Chennai. She has published various papers in national/international conferences and journals.

Dr. S. Sridhar, is a Ph.D. in Computer Science from JNU, Delhi (1984) with 20 years of experience in ONGC, Sharjah Airport UAE, and over 15 years as VC, Director, Principal, Dean, Professor in Engineering institutions. He has guided 12 Ph. D scholars and currently is guiding 10 under various Universities. He has carried out 42 MNC Projects and 30 R&D Projects. He has authored 80 publications in Journals and 2 books.