Reusability Calculation of Object Oriented Software Model by Analyzing CK Metric

Neha Goyal, Er. Deepali Gupta
(Department of Computer Science & Engg., Kurukshetra University/GIMT, Kurukshetra
(Department of Computer Science & Engg., Kurukshetra University/GIMT, Kurukshetra

Abstract--- **In the current world, with ever growing costs of production, software engineers in their bid to produce a software that is within the market acceptable costs and has future scalability, are bound to use techniques that ensure software quality. Production of cost effective software without cutting corners in terms of quality has become essential to stay relevant in the market.**

The research work presents a general model for component-based software. The model precisely specifies component-based software with sound basis mathematics. In the research work, a series of metrics proposed by various researchers are analyzed. Systematic analyses of the values for various metrics are carried out and several key inferences will be drawn from them. These metrics of software components can then be evaluated using design patterns. These techniques can give us useful pointers on various software features, one of them being reusability. Somehow, these have also effect on complexity, testability, modularity and stability. In the research work, we are analyzing reusability concept only that is based on Chidambaram and Kemmerer (CK) matrix. For the component based System, we've figured out various matrices that affects the performance of Component based Software Quality Model and made a try to propose a model that by selecting what matrix of component model gives greater performance as Compared to other matrices.

In this model we are proposing to use unsupervised neural network (SOM) to find out the reusability value for software. This network will take CK metric value as input and will return the extent to which software is reusable. This metric value is being shown as class diagram in Rational Rose for this model.

Keywords--- **CK Metric, Design Pattern, Object Oriented, Reusability, SOM(Self Organizing Map).**

I. INTRODUCTION

An individual software module is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data). All system processes are placed into separate classes so that all of the data and functions inside each class are semantically related (just as with the contents of classes). Developing software system from existing classes i.e. Reuse offers many advantages[1][2]:

1. Development cost is reduced.

2. Reliability is increased.

3. Less time to market.

4. Low maintenance cost.

Reusability is an important characteristic of a high-quality software classes. Software reusability is an attribute that refers to the expected reuse potential of a software component. A component can be considered an independent replaceable part of the application that provides a clear distinct function[2].

Component can be a coherent package of software that can be independently developed and delivered as a unit, and that offers interfaces by which it can be connected unchanged with other components to compose a larger system . Software reuse not only improves productivity but also has a positive impact on the quality and maintainability of software products.

The software industry is moving toward large-scale reuse, resulting in savings of time and money. To develop a new system from scratch is very costly. This has made custom software development very expensive. It is generally assumed that the reuse of existing software will enhance the reliability of a new software application. This concept is almost universally accepted because of the obvious fact that a product will work properly if it has already worked before. Here, in this paper we are considering Design patterns as object oriented classes[2].

A. Design Pattern

Design patterns make it easier to reuse successful designs and architectures[16]. Expressing proven techniques as design patterns makes them more accessible to developers of new systems. Design patterns help you choose design alternatives that make a system reusable and avoid alternatives that compromise reusability. Design patterns can even improve the documentation and maintenance of existing systems by furnishing an explicit specification of class and object interactions and their underlying intent. Put simply, design patterns help a designer get a design "right" faster. Design patterns capture solutions that have developed and evolved overtime. Hence they aren't the designs people tend to generate initially. They reflect untold redesign and recoding as developers have struggled for greater reuse and flexibility in their software. Design patterns capture these solutions in a succinct and easily applied form[22].

In this paper, we r using the artificial neural network based unsupervised learning(SOM) technique. in computer science and related fields, artificial neural networks (ANNs) are computational models inspired by an animal's central nervous systems (in particular the brain) which is capable of machine learning as well as pattern recognition. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs.

B. Unsupervised Learning

Neural networks which use unsupervised learning are most effective for describing data rather than predicting it. The neural network is not shown any outputs or answers as part of the training process-in fact, there is no concept of output fields in this type of system. A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space[4].

II. LITERATURE REVIEW

A brief review of the work done in the field is described here:

Abhikriti Narwal [1] explained with a closer look about the usage of Complexity metrics for component-based Software Systems in Software Industry. This metric is based on the complexities involved in interface methods and properties of these interfaces in components. Most of the findings are based on Survey and Interviews with Software developers, Testers, Research Engineers of many major Software Companies. Various suggestions and recommendations are made that if implemented properly may improve the quality of Software in Software Industry. They show that complex components take much time to execute than simple components and are very difficult to maintain. Also, complex components are not easily understandable. The case study shows that the methods presented are effective by comparing the complexity metrics for each component.

Alexandre Alvaro et. al. [2] proposed a component quality model which describes consistent and well-defined characteristics, sub-characteristics, quality attributes and related metrics for the components evaluation. A preliminary evaluation is proposed to analyze the results of using the component quality model.

Amr Rekaby and Ayat Osama [3] discuss the reusability concept in component-based development (CBD). The roles and responsibilities are mentioned. It studies CBD model, its lifecycle, component based development guidelines. A survey of component-based development and reuse driven development life cycles is presented. The proposed model contains all the needed activities towards a complete component-based development lifecycle. A comparison between ICBD, normal component-based development, and non-component based development is provided. The case study proves that ICBD could decrease the effort of the projects by 40% after few months of application.

Brij Mohan Goel et. al. [7] gives the design of CK suit of metrics and evaluation to these metrics so that these metrics should reflect accurate and precise results for object oriented based systems. Moreover, a set of new metrics are proposed that can find the impact on reusability of a class by using the combination of one CK metric with another metric.

G. Shanmugasundaram et. al. [10] discusses a study of the reuse metrics of three systems i.e. object oriented systems, component based systems and service oriented systems is made and proposed a model to bring out the relationship between them.

.

III. PROBLEM FORMULATION

The goal of this paper is to study the Object Oriented Software Engineering with quality factors and propose a model for improving the performance that would use an Artificial Neural Network as the backend to solve the classification problem. In order to accomplish these objectives, a comprehensive study of design patterns has been carried out.

IV. OBJECTIVE

1. First, analyse ck matrix values.
2. Second, calculate reusability of object oriented software model using neural network based SOM technique.
3. Comparing the obtained reusability value using SOM technique with reusability value based on ck matrix analysis.

V. METHODOLOGY/PLANNING OF WORK

In the research work, a detailed analysis of Object Oriented Software Development, Quality criteria for the internal design of a software component and a detailed analysis to calculate reusability of Object Oriented Mode e.g. Design patterns using CK Metric analysis is done[7]. A popular reuse technique in the object-oriented programming community is design patterns. Design patterns represent a recurring solution to a software development problem within a particular context. They have frequently been used to guide the creation of abstractions in the software design phase, necessary to accommodate future changes and yet maintain architectural integrity. These abstractions help us de-couple the major components of the system so that each component may vary independently. Firstly, Object Oriented Software Engineering model is discussed in detail. Quality factors like reusability i.e. our main parameter and some other factors like maintainability, complexity etc. will be discussed.

A. CK Metric Analysis

For the analysis purpose of Object Oriented Design, design patterns may be used that are worldwide accepted. Through these design patterns, a CK metric analysis has been made. Based on the analysis, some evaluations are made regarding reusability and some methods are proposed to increase reusability based on this outcome[15].

1) Calculating various matrices like WMC (Weighted Methods per Class), DIT (Depth of Inheritance Tree), NOC (No. of Children), RFC (Response For Class), CBO (Coupling Between Objects) are computed for every class of design patterns[18].

2) Summation of matrices for each class is done for every design pattern. Also another matrix NC (No. of classes used in each pattern) is calculated at run time.

B. Neural Network Analysis

The research work establishes a mechanism through which the reusability of object oriented models can be measured, which is then applied to find best metric values for which maximum reusability can be achieved. *Secondly*, a Self organizing map is used for creating clusters for the metric values. Self-organizing in networks is one of the most fascinating topics in the neural network field. Such networks can learn to detect regularities and correlations in their input and adapt their future responses to that input accordingly. The neurons of competitive networks learn to recognize groups of similar input vectors. Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors. Self-organizing feature maps (SOFM) learn to classify input vectors according to how they are grouped in the input space. They differ from competitive layers in that neighboring neurons in the self-organizing map learn to recognize neighboring sections of the input space[4][8].

The best values are find out using neural network scheme i.e. Self Organizing Map. This is divided into 2 stages:

- Developing a Self Organizing Map[9] Neural Network and train the network.
- Based on the computed weights, select the maximum values of matrices so that proposed model may give optimized value of the metric to achieve optimum reusability.

VI. RESULTS & DISCUSSION

I. CK Metric Values

Calculation of CK metric values are done and are calculated as[20]:

1) Calculating various matrices like WMC (Weighted Methods per Class), DIT (Depth of Inheritance Tree), NOC (No. of Children), RFC (Response For Class), CBO (Coupling Between Objects) are computed for every class of design patterns[20].

2) Summation of matrices for each class is done for every design pattern. Also another matrix NC (No. of classes used in each pattern) is calculated at run time.

Table 1 shows the CK metric value analysis for the figure 1. In the same way, CK metric analysis for all the design patterns can be done[16].

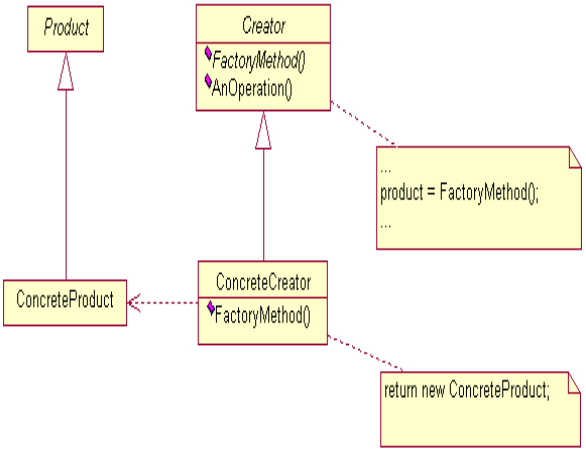

Figure1. Factory Method Design

Table 1: CK Matrix Analysis for Factory Method Design[21]

Class/ Matrix	WMC	DIT	NOC	CBO	RFC
Product	0	0	2	0	0
Concrete Product	0	1	0	0	0
Creator	2	0	1	0	2
Concrete Creator	1	1	0	1	1

II. Neural Network Values

Self Organize Map Neural Network uses number of training data of 21 design patterns with 6 metric values each i.e. 210 elements. The capability of neural network to generalize and insensitive to the missing data would be very beneficial. For training purpose all 21 design patterns and 6 matrices are taken. Number of epochs taken is 200 to achieve high accuracy.

VII. CONCLUSION

The proposed system is capable of finding the reusability value for object oriented software model. We have applied CK metric analysis on the design patterns for each and every class of various types of design patterns. While adding up the metric values for each pattern, number of class is also calculated at run time and concatenated with the final metric value. This in total acts as input for the neural network which is trained as self organizing map (SOM) neural network and Reusability is calculated with the proposed method. By using the example of design patterns and unsupervised neural network, we have proposed a model that provides better reusability for object oriented software.

REFERENCES

[1]. Abhikriti Narwal, "Empirical Evaluation of Metrics for Component Based Software Systems", International Journal of Latest Research in Science and Technology, Vol. 1, Issue 4, pp.373-378, Nov.- Dec. 2012.

[2]. Alexandre Alvaro, Eduardo Santana de Almeida, Silvio Lemos Meira, "A Software Component Quality Model: A Preliminary Evaluation", IEEE Proc. of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, 2006.

[3]. Amr Rekaby and Ayat Osama, "Introducing Integrated Component Based Development (ICBD) Life Cycle and Model ", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.6, Nov. 2012.

[4]. Anupama Kaur, Himanshu Monga, Mnupreet Kaur and Parvinder S. Sandhu, "Identification and Performance Evaluation of Reusable Software Components Based Neural Network", Int. Journal of Research in Engineering and Technology (IJRET), Vol. 1, No. 2, pp. 100-104, March 2012.

[5]. Alvaro, E.S. de Almeida, S. L. Meira, "A Software Component Quality Model: A Preliminary Evaluation", IEEE 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 28-37, Aug.29-Sep.01, 2006.

[6]. Anju Shri, Parvinder S. Sandhu, Vikas Gupta, Sanyam Anand, "Prediction of Reusability of Object Oriented Software Systems using Clustering Approach", World Academy of Science, Engineering and Technology, Vol. 43, pp. 853-856, 2010.

[7]. Brij Mohan Goel, Pradeep Kumar Bhatia, "Analysis of Reusability of Object-Oriented System using CK Metrics", International Journal of Computer Applications, Volume 60, No.10, Dec. 2012.

[8]. en.wikipedia.org/wiki/Artificial_neural_network.

[9]. en.wikipedia.org/wiki/Self-organizing_map.

[10]. G. Shanmugasundaram, V. Prasanna Venkatesan and C. Punitha Devi, "Reusability metrics - An Evolution based Study on Object Oriented System, Component based System and Service

Oriented System", Journal of computing, Vol. 3, Issue 9, pp. 30-38, Sept. 2011.

[11]. Gomathi. S, Edith Linda. P, "An overview of Object Oriented Metrics A complete Survey", International Journal of Computer Science & Engineering Technology, Vol. 4, No. 09, Sep. 2013.

[12]. K. Kaur Chahal, H. Singh, A Metrics Based Approach to Evaluate Design of Software Components, IEEE International Conference on Global Software Engineering (ICGSE), pp. 269-272, 2008.

[13]. Kilsup Lee, Sung Jong Lee, "A quantitative software quality evaluation model for the artifacts of component based development", IEEE 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 20-25, May 2005.

[14]. Margaretha W. Price, Donald M. Needham, Steven A. Demurjian, "Producing Reusable Object-Oriented Components: A Domain-and-Organization-Specific Perspective", ACM Proc. of the symposium on Software reusability: putting software reuse in context, pp. 41-50, May 18-20, 2001,

[15]. Neha Sadana, Surender Dhaiya, Manjot Singh Ahuja, "A Metric for Assessing Reusability of Software Components", International Journal of Computer Application, R S. Publication, Vol. 1, Issue 4, pp. 98-108, Feb. 2014.

[16]. Net Objective Press, "Design Patterns: From Analysis to Implementation", Manuals for design patterns explained: A New perspective for Object Oriented Design, 2007.

[17]. Parvinder S. Sandhu and Hardeep Singh, "A Reusability Evaluation Model for OO-Based Software Components", World Academy of Science, Engineering and Technology, Vol.15, pp. 850-855, 2008

[18]. Parul Gandhi & Pradeep Kumar Bhatia, "Reusability Metrics for Object-Oriented System: An Alternative Approach", International Journal of Software Engineering (IJSE), Vol. 1, Issue 4, pp. 63-72, 2010.

[19]. Pradeep Kumar Bhatia, Rajbeer Mann, "An Approach to Measure Software Reusability of OO Design", Proc. of 2nd National Conference on Challenges & Opportunities in Information Technology (COIT-2008), pp. 26-30, March 2008.

[20]. Ramanath Subramanyam and M.S. Krishnan. "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", IEEE Transactions on Software Engineering, Vol. 29, No. 4, pp. 297-310, April 2003.

[21]. Sandeep Srivastava, "Software metrics and Maintainability Relationship with CK Metrics", Int. Journal of Innovations in Engineering and Technology (IJIET), Vol. 1, Issue 2, pp. 76-82, Aug. 2012.

[22]. www.uml.org.cn/c++/pdf/DesignPatterns.pdf.

