

# Functionality Based Software Packaging using Cuckoo Search

Ilamathy.V

PG Student,

Department of Computer Science And Engineering

Pondicherry Engineering College

Puducherry-605014, India

[ilamathy12@pec.edu](mailto:ilamathy12@pec.edu) 9655153066

Salini.P

Assistant Professor

Department of Computer Science and Engineering

Pondicherry Engineering College

Pudhucherry-6055014

[salini@pec.edu](mailto:salini@pec.edu)

Kavitha.K

PG Student,

Department of Computer Science And Engineering

Pondicherry Engineering College

Puducherry-605014, India

[kindkavi@pec.edu](mailto:kindkavi@pec.edu)

**Abstract**— Modularity is an approach that subdivides a system into smaller parts that can be independently created and then used in different systems. Moreover modularity allows to dealing with changing customer demands, so modularity helps the developer for reusability of packages. There have been some attempts at automatic partitioning of the object-oriented classes into modules (i.e., packages) based on functionality. However, all these attempts are based on source code, they occur late in the development process at the implementation stage, which may need more efforts and cost. Moreover software packaging from survey we understand that values are based on cohesion and coupling at coding phase. Heuristic technique such as genetic algorithm is used to find an optimal package we observed that partitioning cohesion and coupling metric alone is not enough for quality packaging, at coding phase of software development life cycle. So as a solution, extracting a class from sequence diagram is very difficult .So in proposed work software packaging based on source code and also functionality based software packaging. In the proposed system we intend to use cohesion, coupling and overall packaging metric to improve quality of software packaging. In order to find the optimal package we use search based optimization technique such as cuckoo search algorithm.

**Keywords**— Heuristic algorithm, Source code, Functionality Based Partitioning, Cuckoo search algorithm, quality packaging, metrics, sequence diagram

## INTRODUCTION

Modularity is an approach that subdivides a system into smaller parts that can be independently created and then used in different systems. Moreover, modularity allows to dealing with changing customer demands, so modularity helps the developer for reusability of packages. Thus, we propose functionality based software packaging at design phase. In the proposed system, we intend to use cohesion, coupling and overall packaging metric to improve quality of software packaging. In order to find the optimal package we use search based optimization technique such as cuckoo search algorithm.

For the software packaging either sequence diagram or source code can be used the proposed work, aims to use the source code as input and then classes are identified and they grouped using clustering techniques for which k means algorithm is used. It will select k clusters randomly as initial clustering centers and calculate the distance between each classes and initial clustering centers if any changes in the values means it will automatically updates the clustering centers otherwise the process will be stop. Cohesion and overall packaging metric values should be high .coupling values should be low based on these criteria classes will be packaging. Finally searching algorithm is applied to find a best optimal solution. Cuckoo search algorithm is used as a searching technique select random population of n host obtain cuckoo randomly by levy flight behavior select a nest among randomly from host nest. If fitness value is less than old solution, means replace with new solution. Finally, the output will be related classes.

## **RELATED WORK**

Hani Abden, RMof(2011)[1]Modularization Metrics: Assessing package Organization in Legacy Large Object-Oriented Software. proposed cohesion and coupling metrics for problem accessing modularization for not API based object oriented software system .It should not valid for real large software system and validate their utility with independent metrics

Abdeen, H. Ducasse, S., Sahraoui, H., & Allou(2009)Automatic Package Coupling and Cycle Minimization[2] developed Search based technique-Simulated Annealing for Optimizing existing modularizations by reducing the connectivity particularly the cyclic connectivity among packages some of the limitations, does not support indirect cyclic dependencies among packages Large application are usually complex Classes usually are not well distributed over packages

D.Doval, S.Manocrodis, B.S.Mitchell(2010)Automatic clustering of software system using genetic algorithm[3] to find good partition of a given software system using genetic algorithm. But It does not provide a mechanism to integrate a designer knowledge of the system.

Alkhalid, A.,Alshayeb, M., & Mahmoud,S(2011)Software refactoring at the package level using clustering technique [4] Clustering technique as a method to help software designers in refactoring activities in order to balance between software intra-package cohesion and inter package coupling Clustering algorithm to reduce the amount of computational in addition SLINK, CLINK and WPGMA AKNN compare its performance and computational complexity with those techniques. A-KNN algorithm initially considers each entity as a cluster; each entity is labeled with a unique identifier representing the cluster identity. But internal complexity is very difficult to maintain

Shouki, A. Ebad,Moartaz A.Ahmed(2015)[5]functionality software packaging using sequence diagram genetic algorithm is used to find an optimal package using cohesion , coupling and overall packaging metric to find a quality package .

Lionel C. Briand, Senior Member, IEEE, Yvan Labiche, Member, IEEE, and Johanne Leduc(2006)[6] Toward the Reverse Engineering of UML Sequence Diagrams for Distributed Java Software work is the way we specified our reverse-engineering process by using meta models (as UML class diagrams) andtransformation rules (as constraints in the Object Constraint Language).

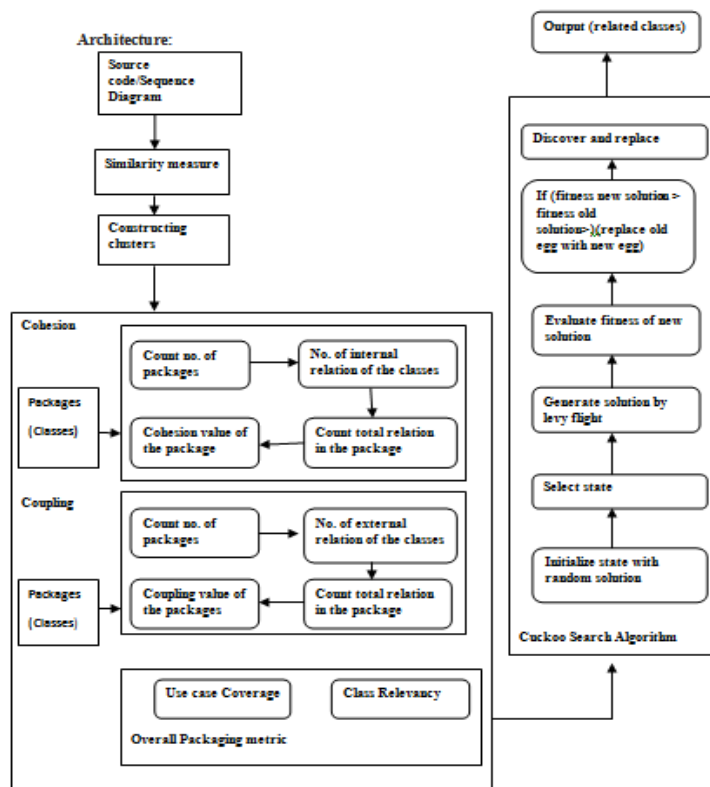
Debasish Kundu<sup>1</sup>, Debasis Samanta<sup>1</sup>, Rajib Mall<sup>2</sup>(2012)Automatic code generation from unified modeling language sequence diagrams[7] was an approach to generate the code within class methods from UML 2.x SDs. For this, we have built a novel graph model called SIG. In contrast to the conventional graph model (i.e. control flow graph), the SIG subsumes control flow graph and

additionally contains method scope information of the interactions. The explicit method scope information present in our graph model helps to identify different class methods for which the code is to be generated.

Chiricota, Y., Jourdan, F., & Melancon, G. (2003)[9] Software components capture using graph clustering. We have presented a simple, fast computing and easy to implement method aiming at the capture of software components from a logical and physical point of view. Our method exploits a metric based clustering of graphs.

### PROPOSED FRAMEWORK

Proposed system architecture consists of three modules such collecting a data set as source code or sequence diagram .Clustering technique is used for similarity measure. Cohesion, coupling and overall packaging metric is used for calculating quality packaging. Cuckoo searching algorithm is used to find an optimal solution.



### MODULES

#### Clustering Technique:

K-means algorithm is used for clustering technique input for k-means is number of classes for clustering .Select k- classes randomly as initial clustering centers. Calculate the distance between each classes with initial clustering centers. If any changes in centers then update the clustering centers there is no changes means the process will get stoped.like that we have calculate distance for all classes until the classes package.

**K-means clustering** is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *K-means* clustering aims to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean, serving as prototype of the clusters.

#### Steps for K-means Algorithm:

- Randomly select 'c' cluster centers.
- Calculate the distance between each data point and cluster centres.
- Assign the data point to the cluster center whose distance from the cluster centre is minimum of all the cluster centers.

#### Metric Calculation

##### Cohesion metric

Cohesion measures the degree of interaction and relationships among modules, such as classes, methods, attributes, and packages within classes.

Collecting the data set as a source code group the classes based on functionality.

1. Cohesion metric
2. Coupling metric

Cohesion metrics is used to find the average number of packages in internal dependency per class and it measure the connectivity of central classes of a given package.

$$CohesionQ(p) = \frac{|P_{Int. D}|}{|PD|}$$

Where,

$P_{Int}$  = no of internal dependencies within classes

$PD$  = no of dependencies of classes

##### Coupling metric:

Efferent coupling ( $C_e$ ): Coupling metrics provide coupling degree of the package.

Afferent Coupling ( $C_a$ ): It provide degree package between client packages.

$$CoupligQ(p) = 1 - \frac{|P_{Pro. P} \cup P_{Cl. P}|}{|PD|}$$

Where,

$P_{Pro}$  = no of provider packages

$P_{Cl}$  = no of client packages

$PD$  = no of dependencies of classes

### Overall Packaging Metric:

#### Use Case Coverage

The least number of packages offers use case. Preferably only one package. We divide the interactions into two type's direct and indirect interactions. The indirect interaction relation is the transitive closure of the direct interaction relation.

$$Ij(Pi) = WD * D + Wt * \frac{T}{wd}Td + wt * TT$$

WD and WT are the weights for the direct and indirect interactions

#### Class Relevancy:

Class relevancy is a combination of class functionality (CF) and class utilization (Util). CF is the extension of the method similarity principle at the class level. Class utilization measures the ratio of class method and interacts with other classes .

$$CF(C1, C2) = \frac{|UC1 \cap UC2|}{UC1} \cup UC2$$

Overall packaging quality calculated by packaging quality and average packaging quality.

$$\text{Overall packaging} = \text{average (packaging quality (pj))}$$

It is used to maximize the packaging effort to find optimal solution

#### Algorithm used:

- ✓ **Cuckoo Search Algorithm:** Cuckoo search algorithm is used to find new and potentially better optimal solution.
- ✓ It will search randomly high quality of solution will carry over to next generation

### EVALUATION METRICS

To compare existing and proposed system using some metrics for software packaging they are as follows.

#### 1. Function oriented metrics

##### Function Oriented Metrics:

Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Since 'functionality' cannot be measured directly, it must be derived indirectly using other direct measures. Function-oriented metrics measure called the function point. Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and assessments of software complexity.

$$FP = \text{raw FP} * [0.65 + 0.01 * CAV]$$

**Number of user inputs.** Each user input that provides distinct application oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately.

**Number of user outputs.** Each user output that provides application oriented information to the user is counted. In this context output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.

**Number of user inquiries.** An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

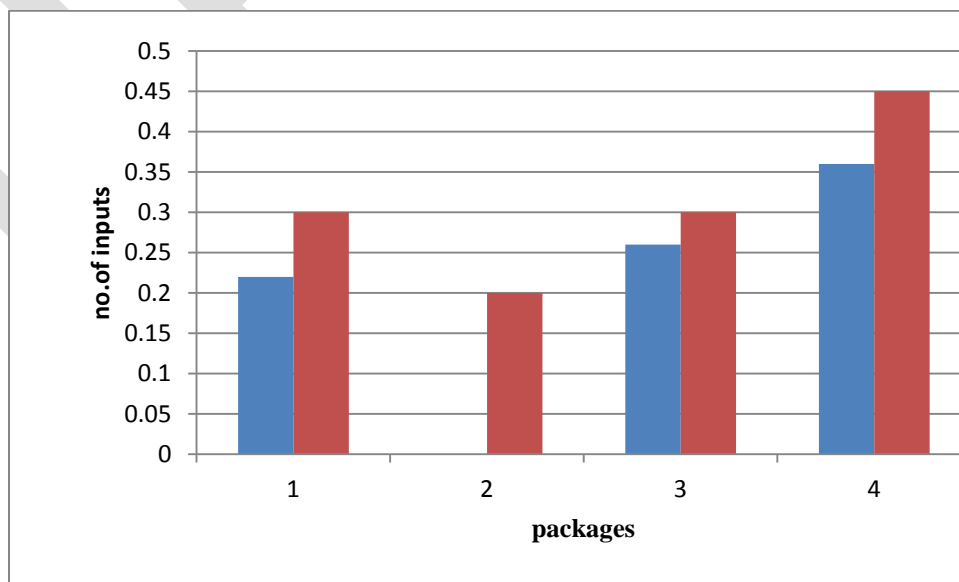
**Number of files.** Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted.

**Number of external interfaces.** All machine readable interfaces (e.g., data files on storage media) that are used to transmit information to another system are counted.

**RESULT AND ANALYSIS**

The experiments are conducted to compare both existing and proposed system result using evaluation metrics. In this, evaluation metrics has been taken as packaging quality will be analyze by some metrics here iam going to funtion point .Below table5.1 is mentioned for comparison work.

EXISTING SYSTEM		PROPOSED SYSTEM	
Technique Used	No.of inputs	Technique Used	No.of inputs
Genetic Alogirthm	0.62	Cuckoo Search Algorithm	0.70
	0.56		0.63
	0.63		0.66
	0.76		0.77



## CONCLUSION

In our work it will be functionality based software packaging in design phase. It will provide effective modularization compared to previous method. Cuckoo Search Algorithm will be used to search for optimized solution. In future work it will be package using some other packaging metric to improve packaging quality and also to extract classes from some other clustering technique and also apply some other variation of heuristic technique for searching algorithm and also it will be used real world open source system.

## REFERENCES:

- [1].Hani Abden, RMof, INRIA-Lille, France (2011) Modularization Metrics-Assessing Package Organization in Legacy Large Level Object Oriented Software. Working conference.
- [2]. Abdeen, H. Ducasse, S., Sahraouiy, H., & Alloui, I. (2009). Automatic Package Coupling and Cycle Minimization. Working Conference on Reverse Engineering (WCRE), pp. 103–122 USA
- [3].D.Doval, S.Manocrodis, B.S.Mitchell(2010) Automatic Clustering Of Software System Using Genetic Algorithm (Dept of mathematics and computer science Drexel University Philadelphia,PA 19104.
- [4]. Alkhalid, A.,Alshayeb, M., & Mahmoud, S. (2011). Software Refactoring at the Package Level Using Clustering Techniques.IET Software, 5(3), 274-286
- [5].Shouki, A. Ebad,Moartaz A.Ahmed(2015) Functionality – Based Software Packaging Using Sequence Diagram Springer science business media New York.
- [6]. Lionel C. Briand(2006) Toward the Reverse Engineering of UML Sequence Diagrams for Distributed Java Software Member, IEEE, Yvan Labiche, Member, IEEE, and Johanne Leduc IWPC USA.
- [7]. Debasish Kundu<sup>1</sup>, Debasis Samanta<sup>1</sup>, Rajib Mall<sup>2</sup>(2012) Automatic code generation from unified modeling language sequence diagrams <sup>1</sup>School of Information Technology, Indian Institute of Technology, Kharagpur 721302, India<sup>2</sup> Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India
- [8]. Ahmed, M. (2011). Towards the development of integrated reuse environments for UML artifacts. In The 6th international conference on software engineering advances (ICSEA). Spain.  
Alkhalid, A., Alshayeb, M., & Mahmoud, S. (2011)
- [9].Chiricota, Y., Jourdan, F., & Melancon, G. (2003). Software components capture using graph clustering. In 11th IEEE international workshop on program comprehension (IWPC). USA.
- [10].Clarke, J., Dolado, J., Harman, M., Jones, B., Lumkin, M., Mitchell, B., et al. (2003). Reformulating software engineering as a search problem. IEEE Proceedings-Software, 150(3), 161–175.

- [11]. Doval, D., Mancoridis, S., & Mitchell, B. (1999). Automatic clustering of software systems using a genetic algorithm. In Software technology and engineering practice (STEP'99). USA.
- [12]. Ebad, S. (2012). Functionality-based software packaging approach for higher architecture stability. Ph.D. Dissertation. Department of Computer Science and Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia.
- [13]. Ebad, S., & Ahmed, M. (2011a). Software packaging approaches—A comparison framework. In The 5<sup>th</sup> European conference on software architecture (ECSA 2011). LNCS 6903, pp. 438–446. Essen, Germany.
- [14]. Ebad, S., & Ahmed, M. (2011b). An evaluation framework for package-level cohesion metrics. In Proceedings of the 2nd international conference on Future Information Technology (ICFIT). Singapore  
Software Qual J (2015) 23:453–481 479
- [15]. Seng, O. Bauer, M., Biehl, M., & Pache, G. (2005). Search-based improvement of subsystem decompositions. In Proceedings of the genetic and evolutionary computation conference (GECCO)