

# Software maintenance using effective code clones detection

M.Alagurathinam<sup>1</sup>

Department of Computer Science and Engineering  
Sathyabama University .  
Chennai-600 119  
alagu004@gmail.com

Ms.Kavitha Esther Rajakumari<sup>2</sup>.

Department of Computer Science and Engineering  
Sathyabama University .  
Chennai-600 119  
kavithaesther@rediff.com

**Abstract:** In this proposed paper it described about the code clone detection using the fine granularity technique called frequent pattern growth algorithm. Here it used for detecting that the type clones which are known as functional clones. In software project it mainly detect the software code .it helps to extract the methods using frequent occurrences in patterns and names from that files it extracts the common methods as libraries In addition to that it proposes maintenance To be performed such as adaptive maintenance, corrective maintenance, perfective maintenance for the sample project.

**Keywords:** Code clones, maintenance, frequent pattern.

## I. INTRODUCTION

Normally in software maintenance testing team commonly maintaining the set of software projects in which the software has been maintaining quiet a period of time .and after I will updated into current updates. In updating the existing codes are changed with newly added properties and some configurations

In the process some developer proposed a some thoughts are matches with similar concepts in already designed code .That kind of code are called code clones.This will caused the software to preform little bit slower and affects the system performance

So we look into those code fragments and rectify them by detecting them varies techniques are already proposed some of the techniques are Antonio Cuomo describe about the detection of code clone pair using calculus communicating systems for type -2 clones.

Nicolas Battenberg discuss about the impact of software code clones and its effect on the user level.and then normalize the methods and filter the basic methods such as getter and setter methods and filter out the methods then after generate the hash values for every textual representation.

We listed out the those repeated clone methods in the functional tableland finally they make hash groups from the hash groups they extract the common methods as Libraries which will helpful for the huge software files

In the proposed paper it filters out methods that Have no block statements so such functions have some error checks normally to make a library files are predefined file which was developed by most frequent usage

## II. LITERATURE SURVEY

Some of the authors are also discussed about the code clones and its common factors.

Pitts and Raoult and Guillemain these author described the equivalence of programs has been discussed in terms of operational semantics

Pits introduced a method proving contextual equivalence of ML functions. Result and Guillemain proves that two different ways of program equivalence they are fixed-point semantics and operational semantics are discussed for recursive definitions

Ivanovo introduces a technique called program schemata He discussed about the program transformation is to transform a program into a semantically equivalent one by applying only semantic-preserving transformations

Fischer describes about whether a component can be reused in a given context without any modification. The basic idea of the reuse approach is that a component satisfies a query with precondition and post condition

Podgurski and Pierce introduces a method called behavior sampling for the automated retrieval of components from a software component library for the purpose of reuse. The components are organized in a classification. The user is prompted with a choice between two tuples and its behavior. And there are another two different detection are available they are static and dynamic similarity detection

There has been significant research dedicated to detecting syntactically similar code fragments the research in the area of program based clone detection can be found in the survey from Kosher Clone detectors, such as CCFinder, Clone Detective and Deckard are effective in finding Clones created by copy paste programming.

These similarities of independent origin are also referred to as semantic clones or type-4 clones Sager another researches proposed an approach for detecting similar Java classes using tree algorithms for understanding software evolution Kim use a semantic-based static analyzer for detecting semantic clones

Kawrykow and Robillard propose a method for detecting similarity of API methods. Their approach detects client code that duplicating methods available in the API of the type's. Kuhn introduced an approach for identifying "topics" in source code. They use latent semantic indexing and analyze linguistic information in identifiers and comments for grouping source code artifacts that are semantically related. This technique is called as Semantic Clustering McMillan suggested a method for detecting similar software applications. They use the Concept of semantic anchors that used for API methods to define the semantic characteristics of an application.

KarpRabin fingerprinting is used for calculating the length of n substrings of a text first text to text transformation is performed for the dataset for eliminating uninterested characters then it removing all whitespace characters except line separators

Dynamic Similarity Detection In this detection Jiang and Su deploys random testing to find functionally similar code fragments, code fragments described under input and output behavior. A code transformer turns the code fragments into executable units. Then the code fragments are clustered by separating fragments with different outputs for the same input.

### III. PROPOSED WORK.

In this proposed paper it handled set of software projects for software maintenance. In this paper it reads the application file and scan The file using frequent pattern growth Algorithm

#### A. Extract methods

In frequent pattern it reads the file using method name and it compare the method with the another file it checks the method name and inside the it check the block statements and some logic that occurred in parent program

#### B. Compare with sample

It performing comparative code it with the child program it keeps on checking the total lines related to the application and concludes them in the functional table after

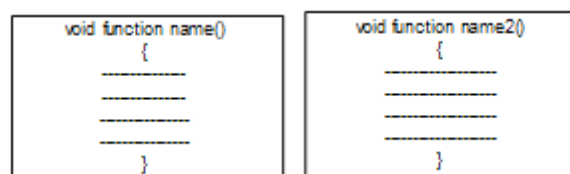


Fig 1. method detection

#### C: Create functional table

Drawing into functional table it checks the threshold value of most repeated method with default threshold

#### D: Enable as library files:

If any method exceeds the threshold value it was converted into library file

#### E: Maintenance analysis

In adaptive maintenance modification of software product performed after delivery to keep a software product usable in a changed or changing environment

In corrective maintenance it performs diagnosing and fixing errors possibly ones found by users

In perfective maintenance it performs implementing new or changed user requirements which concern functional enhancement to the software.

In preventive maintenance it performs increasing software maintainability or reliability to prevent future problems are concerned. Another state of the token based clone detection technique is CP-Miner where a

frequent subsequence mining technique is used for identifying a similar sequence of tokenized strings in the sample project. Normally token-based techniques are used to assist plagiarism detection.

M. Kim proposed a model of clone genealogy on clone evolution. According to their study, refactoring of clones may not always improve software quality based on the revisions of the performance and maintainability.

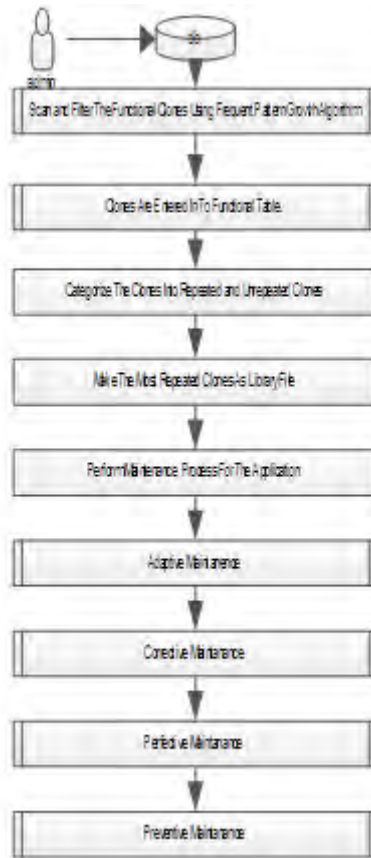


Fig .2. Architecture Diagram

#### IV.ALGORITHM USED

Frequent patterns are patterns that appear frequently in a dataset.it performs the function as to repeatedly scan the whole database and check a large set of candidates by pattern matching.

For designing a method that mines the complete set of frequent item sets it compresses the database representing frequent items in to fptree, then divides the compressed database into a set of conditional databases each associated with one frequent item or pattern fragment and mines each database separately.

Fragment 1:

```

int i, j=1;
for (i=1; i<=VALUE; i++)
j=j*i;
  
```

Now consider the following code fragment 2, which is actually a recursive function that calculates the factorial of its argument *n*.

Fragment 2:

```

int factorial(int n) {
if (n == 0) return 1 ;
else return n * factorial(n-1) ;
}
  
```

This approach may substantially reduce the size of the datasets to be searched .in frequent pattern mining method uses tree to generate conditional pattern basses using a bottom –up projection technique.

For examining type-3 clones other equivalences are considered in software product. Easy to identify and refactor and formulate

TABLE I .TAXONOMY FOR CODE CLONE TECHNIQUES

Language Paradigm	Only Procedural
	Only Object-oriented
	Both procedural and OO
	Byte code
	Assembly code
	Extreme Programming
Clone Relation	Directly ClonePair
	Directly CloneClass
	CC in post processing
Level of similarity	Textual
	Lexical
	Syntactical
	Semantical
	Hybrid
Clone granularity	Free
	Fixed
Clone similarity	Exact Match
	Parameterized match
	Near miss
	High level clone
	Design level structural clone
Comparison granularity	Line
	P-Line
	Substring
	Identifiers and comments
	Tokens
	Statements,subtree,sub graph
	Begin-End Blocks Methods,classes,Files

A. In CMMI level-3 service level software maintenance as for as concerned it with other level of maintenance level it provides software product a more effective one .by providing this prototype it helps for the user related to maintenance In previous code clones method detection are done by only some efficient tools in this method we are detecting and performing template creation for the software product and code refactoring also done to eliminate the dead codeIn common software maintenance project companies provide annotations and type of work spaces are given to user separately about what are the specific functions repeated are identified and helpful to the user to work on

## V. RESULTS AND DISCUSSIONS

In this proposed paper it discussed about the processing of code clones and it will displayed in the table It will filter out the method and its repetitive ones are extracted by algorithm are shown in the table.

TABLE II WORK COMPLETED.

Code Clone Table		
Sl.No.	Code Analysis	Depth measures
1	Total no of projects	10
2	Total no of code examined	1033
3	Total no of methods retrieved	42
4	Total no of similar methods retrieved	12
5	Library files converted	2

## VI. CONCLUSION AND FUTURE WORK

This paper proposed a method-based clone detection technique is used to creation of libraries. Here we examined the some sample application in java as software projects. Our study proves that the proposed technique could detect the clone detection from the application set.

We inspected and retrieved detected clones and transferred as library file. In future it will implement the second part of frequent pattern algorithm, that could be helpful for the automatic and even more efficient detection of code clone it will helpful for the easy manipulation handheld executable files for future software projects.

## REFERENCES

- [1] Hiroshi Igaki Keisuke Hotta, Tomoya Ishihara, Shinji Kusumoto ,Yoshiki Higo:“Inter-Project Function a Clone Detection toward Building Libraries Survey” An Empirical Study on 13,000 Projects in year,pp-17.
- [2] AkitoMonden, DaikaiNakae, Ken-ichi Toshihiro Kamiya, Matsumoto, Shin-ichiSato“Software Quality Analysis by Code Clones in Industrial Legacy Software”.
- [3] Jonathan I. Maletic ,Michael L. Collard, NouhAlhindawi Omar Meqdadi“Towards Understanding Large-Scale Adaptive Changes From Version Histories”in the year 2009,pp-1-4.
- [4] Scott Grant James, R. Cordy David B.Skillicorn“Reverse Engineering Co-maintenance Relationships Using Conceptual Analysis of Source Code”in the year 2008, vol-, pp-4-7.
- [5] Nelly Maneva“Software Quality Assurance and Maintenance for Outsourced Software Development”.
- [6] Scott Grant, James R. Cordy, David B. Skillicorn “Using Topic Models toSupport Software Maintenance”.
- [7] Antonio Cuomo, AntonellaSantone, UmbertoVillano “A Novel Approach Based on Formal Methods for Clone Detection”.
- [8] Nicolas Battenberg, Weiyi Shang, Walid M. Ibrahim, Bram Adams, Ying Zou, Ahmed E. Hassan “An empirical study on inconsistent changes to Code clones at the Release level”.
- [9] Lars Heinemann “Effective and Efficient Reusewith Software Libraries”.
- [10] JiaweiHan,JianPei,MichelineKamber, “Data mining concepts and techniques” published by Elsevier, a division of Reed Elsevier India Private Limited Chapter 6. Mining frequent patterns Associations and correlations, Basic concepts and methods, 6.2 Frequent Item set Mining Methods, 6.2.4 A Pattern Growth Approach for Mining Frequent Itemsets pp 243-245,257-259.
- [11] [http://Data\\_Mining\\_Algorithms\\_In\\_R/Frequent\\_Pattern\\_Mining/The\\_FP-Growth\\_Algorithm](http://Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm).
- [12] [http://www.cs.uiuc.edu/~hanj/pdf/dami04\\_fptree.Pdf](http://www.cs.uiuc.edu/~hanj/pdf/dami04_fptree.Pdf) pp- 73-79.