

A Novel Ontology Metric Approach for Code Clone Detection Using Fusion Technique

¹Syed MohdFazalulHaque, ² Dr. V Srikanth, ³Dr. E. Sreenivasa Reddy

¹Maulana Azad National Urdu University, ²Professor, Dept of CSE, K L University,

³Professor, Dept of CSE, ANU

¹fazal.manuu@gmail.com, ²srikanth_vemuru@yahoo.com, ³esreddy67@gmail.com

Abstract—Cloning of Code is effective way of identifying the faults in the software, code duplication has been a wide research area in the field of software engineering to identify clones and analysis the clones. Clones software is a region of source which give use to identify high similarity. The region of identity is named clones.

Cloning of software is the regions in identifying the similarities in a software code, an identification of duplicate in a code are called clones, class clones, or pair clones. In our paper, we present a fusion method based on metric with a combination of text method in detection of clones and reporting it is proposed. The work of proposed is categorized into two different stages, selection of clone potential and clone potential comparing based on text comparison. The technique of proposed is detected using clones exact basis on match metric and text matches.

Keywords — code clone, Hybrid, Textual clones, Functional clone.

I. INTRODUCTION

Clone codes are studied in the past for a long period and there has been an evidence of finding majority of faults in the software. The duplication of code has been vigorously studied in the area of software engineering in find clones of software with the area of clone analysis.

Clones in software source code has become the highest similar in the area of research, the region of duplication is called clone identification.

Code cloning in software is the regions where code is found with high similarity; the region of duplicate are called clones, class clones and pair clones. There may be various ways in which the two parts of code may be similar, majority of the researcher from the literature state that copying of the code from one to another is done intentionally by the programmer for rapid development of applications which leads to problem of code, clone can also occur due to the inclusive of libraries based on the framework which is proposed in the tool.

A clone in the software aspect has created new paradigms in software evolution. If a system has been evolved, the clones related to it have to be known based on the changes made consistently. Cloning is a strategic change that has evolved in a software. In recent, various type of code cloning techniques have been developed in empirical comparison and for checking the percentage of clones occurred in a software.

Works of cloning will increase the lines of code without give the productivity of software. It also result in excessive cost of maintenance. Along with that it also increase the negative impact like increase in complexity and length, which lead to difficult in editing the code, increase in human error and also increase in maintenance cost. Forgetting and code overlook will increase the size of the code, it also re-duplicates the code error which leads to increase in error and decrease in efficiency.

Code clone identification

Code clone identification will serve major purpose like study of evolution of clones, detection of plagiarism and performance, factor of extraction procedure and refactoring, detection of faults and repair. Earlier studies have shown and stated in finding the identical clones, or clones which are similar in textual data via literals, identifiers and transformation state variable. Likewise identification of code which are not similar but no occurrence of identify in code are not often in practice such factors has much importance in finding and identifying segment of code in the program.

The classification of clones is done under two categories namely Functional clones and textual clones. We have found that number of developers have studied the techniques of finding clones and written various approaches for clone detection in software based on certain programs. Very few of them have followed approaches like Approach of Textual, Token-based approach, semantic approach and syntactic approach.

II. CLONE DETECTION PROCESS

Clone detection process involves several steps that are:

- Pre-processing
- Transformation
- Extraction
- Normalization
- Match-detection
- Formatting
- Post-processing
- Aggregation

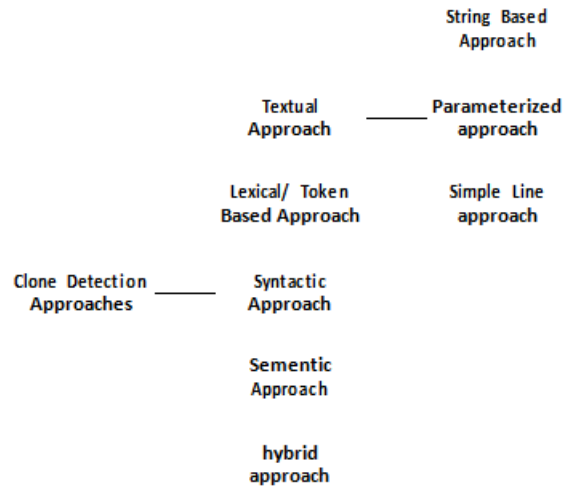


Figure 1 : Approach in Detection of Code Cloning

III. RELATED WORK

P .Batta[1] stated that detection of clones are done based on duplicate code clones in an applications. The clones in the program will create a bug in certain segments of the software when a code is duplicated. The objective of proposed work is to develop and design a fusion based technique which detects clones in web based applications on internet. This model will provide an automatic approach of clone detection.

N. selahmat Ali [2] he suggest and proposed a mapping technique which is used to identify the number of source code clones occur in webpages as the pages get duplicated. He first designed an ontology and map technique which are used to identify and detect clones in various web based files which are available in various types of systems.

Deissen beck .F [3] specified the reasons and harm that is caused by code cloning and the solution which are used to detect those clones. He also suggest the challenges and present problems are occur in real time code clones. He developed a tool which evaluated the code cloning and provides a percentage of occurs of duplicated code and gives the percentage assurance of quality.

Roy chancel .K [4] developed a framework which coherently detects clones in software and also provides the art of detection of clones based on comparison qualitative methods which show the range of percentage of duplication occurred in code and its dangerous effect on the software.

Kawaguchent. S [5] proposed a reliability study on the duplication of code and suggested that code cloning in software is the major drawback which causes the failure of software dynamically and also increase the cost of software maintenance.

Nguyen .H [6] suggested a approach of structure programming in detection of clones, which have become more popular in model base detection of clones and code based detection. He compared these method with structured based clone method and computed that his method suggest best in detection of duplicate code.

ONTOLOGY METRIC FOR CODE CLONING

Ontology Metric in Software code cloning - The ontology for code cloning metric mainly depends on metric values

Source code , Count measure , Normalized code, Calculate metric, Compare metric , Value match , Clone detection , No clone detection , MetCout , MetStmt , MetMCC , MetNLUD , MerVar , Depth , countInput , CountOutput , CountPath , CountStmtDecl , CountStmtExe

The ontology develop will measure the quality of the code cloning algorithm base on the metric. the reason for developing this ontology is to measure the efficiency of the code cloning algorithm the ontology of metric specify the combination are calculated using tools. there are certain tools available for metric calculation. This ontology metric will provide the efficiency of the software, which is based on detecting library candidate, software compacting size, understanding code, malicious software and usage patterns.

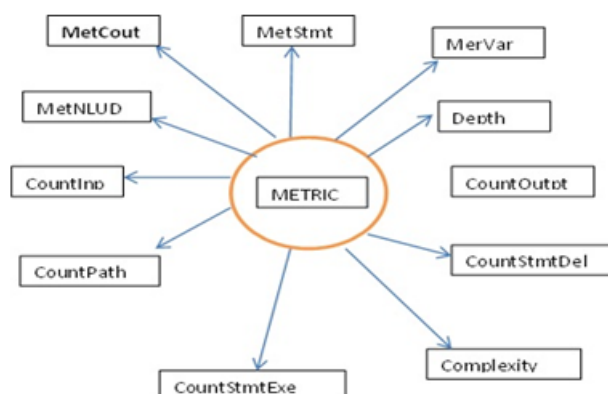


FIGURE 2 Ontology Metric for Code Clone Detection

IV. OBJECTIVE AND METHODOLOGY OF THE WORK

The major objective of study is to develop a software clone detection method which is very much compatible to various languages code and also to web based objection oriented platform too i.e web based applications and Java base like html, php, asp and Jsp.

The proposed method has certain objectives to be achieved, with the inclusive of fusion approach in finding the clones in various programming languages of code and also find the level of clones that are occurred using software measure metric like number of lines of code, complexity cycle, procedure calls, operators, operands number, number of variables and etc. these are verified within the two applications and are compared with the clone metric for performance evaluation. If the match is found apply comparison of textual i.e the source file which are opened are checked line by line for the detection of duplicate code; if there is an occurrence of match generate a report for each line stating the line which is cloned, number of same occurrences and frequency, percentage of clone and reflection to the software code.

Performance Metrics

The metric of performance will give and test the occurred clones in both the files. The comparison is done based on the clone potential and the occurrence

Performance metrics gives the identification of potential clone in both the testing files. If there is existence of some potential clone then only we can do the textual line by line comparison of the two files. If there is no common performance metrics that means; there is no potential clone and we don't go for the textual comparison. This will save the time and gives efficient results.

There are several parameters that are used in the code clone detection techniques. Some of these parameters are LOC: No of lines in code, Public Variables, Private Variables, Protected Variables, Public Functions, Private Functions, Protected functions, If statements, Loop Statements, Redirect statements

The proposed Clone detection algorithm:

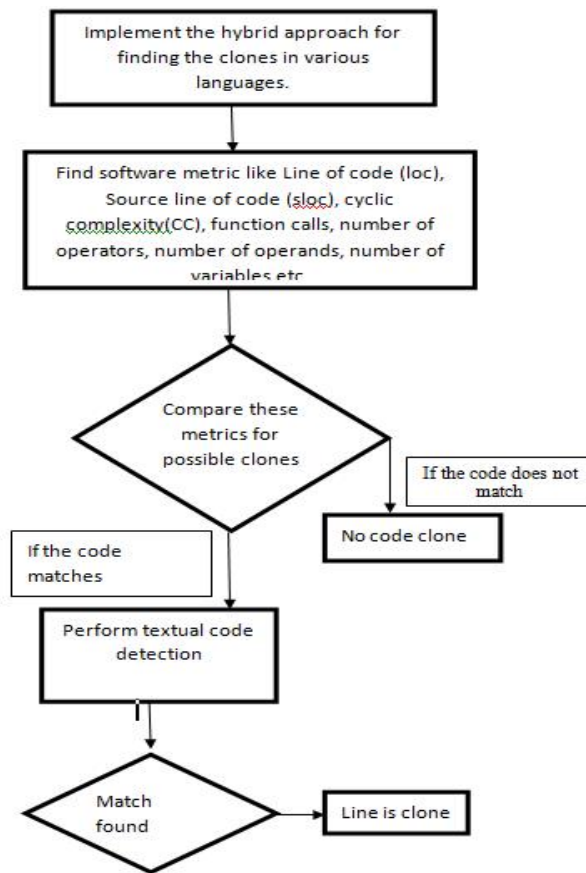


Figure 3 : Flow chart for Detection of Possible Clone

Software description:

The tool is developed using Java Beans. Net Beans provides an open source for development and dedication of software product. i.eNetBeans , this product is used to develop applications very rapidly by the users, developers based on the need of the business customers. It also provides an effective an easy way of developing internet based application on java platform which strength the product and also relevel the standards of the industry.

V. RESULTS

The web applications files for clone detection are chosen as follows

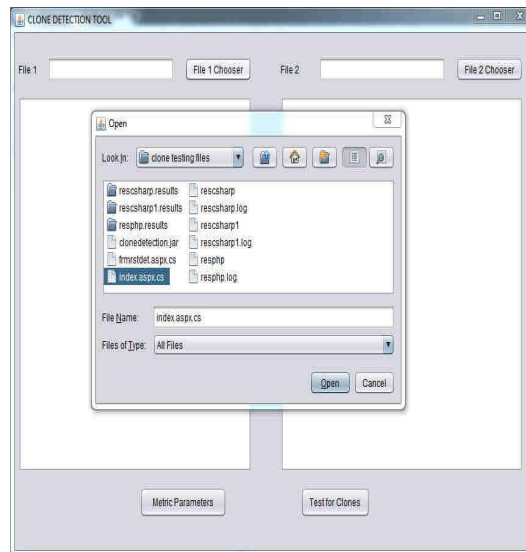


Figure 3 : Application for Web Application Source

In figure 3 the web applications are first chosen for identification of clones in software, then the parameter metric of both the files are considered and are calculated with the inclusive of : total number of lines in code, variables of private, variable of public, variable of protected, loops, if statements, private functions, public functions, protected functions, keywords, input and outputs statements, expression and operators.

	FILE 1	FILE 2
LOC	35	25
Public Variables	0	0
Private Variables	3	1
Protected Variables	0	0
Public Functions	0	0
Private Functions	0	0
Protected Functions	3	1
If statements	2	0
Loop control statements	0	0
Redirect Statements	1	0

Figure 4 shows the parameter metric of two application of both the files in identification of clones

Clones

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
    }
}
                
```

Cloned Parameters

Clones Percentage

51.428574 %age

Cloned Parameters

Figure 5 :Choosing the second web application file to test for clone

Showing the detected clones from the 2 uploaded files and the clone parameters

Detection Parameters	
Value	
Similarity Threshold	99%
Maximum parameter count	6
Minimum Miss (Lines)	60
Character per node	16
Starting length	2

Clone Detection Statistics	
Statistic	Value
File Count	2
Total Source Lines of Code (SLOC)	248
Estimated SLOC before preprocessing	238
Expanded SLOC after preprocessing	238
Total Clonesets	1
Exact-match Clonesets	1
Near-miss Clonesets	0
Number of cloned SLOC	238
SLOC as clones %	100.0%
Estimated reasonable SLOC	117
Possible SLOC reduction %	49.2%
Possible SLOC reduction in expanded file %	49.2%

Figure 6 : Metric Value Measure in Code Cloning using Ontology

The above setting is based on clone's detection in uploading 2 files and percentage of clone identification using detector clone tool.

Results are based on comparing technique of proposed with that of existing detector clone tool, which is used for testing asp.net which is based on web files. Detection of clones is done based on metric performance and the ability of potential occurred in clones. Our approach will give only the detect the lines of dependency in code and LOC line and will compare with the textual and generate the percentage of clones in the software and also give the time taken for detection of clones. This approach is very much effective in comparison of large files of software in finding the percentage of clones effectively.

VI. CONCLUSION

Code duplication is a process of phenomenon in large system software. These are caused many software engineering who copy and paste the functionality of the code from one software to another.

The main reason for clone existence in programming code is easy to copy and fragment the code easier, simpler and fast process in development of code instead of building from scratch. A time it is difficult to write the program based on the logic, so it is very easier to implement the existing or available code functionality. So duplication of code is increase based on size and form, so it is very difficult for maintenance of code and comprehension difficult in execution process.

In fusion approach based on metric technique along with combination with text technique in detection and reporting done on clones has been proposed. This work has been categorized into 2 stages, first the clone potential selection and its comparison based on textual information. This technique uses exact detection of clones based on the match of metric and has been compared with text matching

REFERENCES

- [1] BattaPriyanka, "Hybrid technique for software code clone detection" International Journal of Computer and technology, Volume 2, Issue 2, April 2012
- [2] Selamatali&wahidnorfaradilla, "code clone detection using string based matching technique" International symposium on empirical software engineering" volume 2, Issue 4 April, 2005
- [3] Deissenboeck Florian, Hummel Benjamin Juergens Elmar, Pfaehler Michael, "Model clone detection in Practice" ICSE Vol. 2, Issue 3, September 2008
- [4] Roy K Chanchal, Cordy R James, Koschke Rainer " Comparison and evaluation of code detection techniques and tools" volume 2 issue 24 Febuary, 2009
- [5] Kawaguchi Shinji, Yamashinay Takanobu, Uwanoz Hidetake, Fushida Kyhohei, Kamei Yasutaka , Nagura Masataka and Lida Hajimu "Shinobi: A Tool for Automatic Code Clone Detection in the IDE" ISSN : 2229-3345 Vol. 4 No. 06 Jun 2013
- [6] Nguyen Anh Hoan, Nguyen Thanh Tung, Pham . H Nam, and Nguyen N Tien "Accurate and Efficient Structural Characteristic Feature Extraction for Clone Detection Volume 2 issue 8, September 2010.
- [7] Puri Amit, "software code clone detection model" Vol. No.1, Issue 3, October 2012
- [8] Chanchal K. Roy and James R. Cordy, "An Empirical Study of Function Clones in Open Source Software", 1095-1350/08 \$25.00 © 2008 IEEE
- [9] Mark Gabel Lingxiao Jiang Zhendong Su, "Scalable Detection of Semantic Clones", ICSE'08, May 10– 18, 2008, Leipzig, Germany. Copyright 2008 ACM
- [10] Chanchal K. Roy, "Detection and Analysis of Near-Miss Software Clones", 978-1-4244-4828-9/09/\$25.00 2009 IEEE
- [11] Yoshiki Higo, and Shinji Kusumoto, "Enhancing Quality of Code Clone Detection with Program Dependency Graph", 2009 IEEE

- [12] ImanKeivanloo, JuergenRilling, Philippe Charland, "SeClone - A Hybrid Approach to Internet-scale Real-time Code Clone Search", 1063-6897/11 \$26.00 © 2011 IEEE
- [13] Randy Smith and Susan Harwitz-detecting and meausingsimilaties in code clones.
- [14] Chanchal K. Roy ^a, James R. Cordy^a, Rainer Koschke^b-Comparison and Evaluation of Code Clone Detection Techniquesand Tools: A Qualitative Approach
- [15] K. Kontogiannis, R. DeMori, E. Merlo, M. Galler, and M. Bernstein, Pattern Matching for Clone and Concept Detection, Journal of Automated Software Engineering, 3(1-2):77-108 (1996).