Volume 4, Issue 1, January 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Study on Metrics Based Approach for Detecting Software Code Clones

Amandeep Kaur, Balraj Singh
Department of Computer Science & Engineering,
Lovely Professional University, India

Abstract- Software engineering is an about development, design operation and maintenance of software. But there are some factors that make software maintenance difficult. Code clone is one of the factors that increase software maintenance and also cause code bloating. A code clone is nothing a similar or duplicate code in a source code or created either by replication or some modifications. Various clone detection techniques are proposed, that are quite efficient in providing clones. Many plagiarism detection also work in finding the duplicated code , but not applicable on large systems. This paper describes the metrics based approach used to detect the code clones. Rather than working on source code directly, metrics are used directly to detect clones. This approach will used different software metrics to measure clones in code.

Keywords-- Software, Clone detection, metrics based approach.

I. INTRODUCTION

Software engineering is an about development, design operation and maintenance of software. Consequently in software engineering main focus is on to assure the quality in the product, detect the bugs and prevent system from bugs by testing or analysis. While developing any software for saving time and effort, software developer copy paste program code again and again. So if any bug found in one module is reproduced in every copy. There are many copies of code present and no record of such copies is present. This will make hard to fix such bugs and maintenance of existing software. Code clone is one of the factors making software maintenance more difficult.

A code clone is nothing a similar or duplicate code in a source code or created either by replication or some modifications. These cloned code add to high maintenance cost of software and also cause the code bloating[9]. This is because when changes perform on one clone, then the same action is performed on respected clone, this will increase the maintenance. These clones can also increase risk of faults in system by increasing the risk of making inconsistent changes to the code. Previous studies have shown that around 7% to 23% of the source code in a software system contains code clone[9]. There are number of tools to detect the code clones, but it is not effective to remove the clones. Because code clones are needed for software to function properly. So we can apply the principal of refactoring or modularity to improve the reusability and maintainability of software from clone code. Code Fragments are sequence of code line with some similarity. These code fragments are identified by starting and ending lines numbers of that code with file name. In software system code clones shows mainly two kind of similarities.

They can be similar if their program text matches or on their functionalities basis if the behavior among them match.
Basically there are four types of clones:

1.) Type I: These code clones are identical code clones that only allow modification in white space and comments. These are the exact copies of original code.
2.) Type II: These code clones are structurally and syntactically identical copies. These are also known as Renamed Clones. These clones have variation in layouts and changes in literals, identifiers.
3.) Type III: These code clones are copied fragments with further modifications by changing, adding or removing statements. These Clones are also known as Modified Clones. They modified type 2 clones by changing, adding or deleting some line and become type 3 clones.
4.) Type IV: These code clones are similar in functionality and logic but having different syntax. These are also called semantic clones.
From four types of clones Type 1 Clone is easy to detect but Type 4 clone id difficult. There is many techniques and tools are developed to detect clones but all are good in detecting Type 1-3 clones, Type 4 is hard for them. This division of Type 1-4 shows the level complexity of detecting clones.

II. CLONE DETECTION APPROACHES

Metric based approach calculates metrics from source code and uses these metrics to measure clones in software[6]. Metrics are calculated from names, layout, and expression and (simple) control flow of functions, and a clones is defined as a pair of whole function bodies with similar metrics values.

TABLE I
CLONE DETECTION APPROACHES

Type of Comparison	Type of Clones	Efficiency	Other
Text Based	Type 1	High	**No Transformation**
Token Based	Type 1,Type 2	Low	**Transformation into tokens**
Abstract Tree Based	Type 3	High	**Convert code into abstract syntax tree**
Program Dependence Graph Based	Semantic & syntactic Clones	High	**Shows Control Flow and data dependence**

III. LITERATURE SURVEY

(Abdul-El-Hafiz,2012) [1]et.al proposed a metric based data mining approach for software clone detection. They are using Fractal Clustering Algorithm. This approach has basically four phases. First of all a source file is given as input for preprocessing. In second phase all the fragments that are analyzed and related metrics are extracted. Third phase will use fractal clustering to partition the set of fragments into small number of clusters. This technique places functions into three different clusters: Primary Intermediate, Single clusters. In Primary cluster type 1 and type 2 clones grouped, whose metric value are same and shows same degree of similarity associated with chosen metric set. In intermediate cluster type 3 clones are grouped, who having same metric value and fall under some threshold value. In single cluster those functions are clustered, which are single and not matched with other functions in system. This technique is done by using 8 metrics which detect each function concept. Critics of this technique is that, type 4 clones are not detected by this approach

(Rupinder Kaur,2012) [8] et.al evaluated a token based tools, CCfinder and PMD copy/paste detector on three types of metrics- Line metrics, clone set metrics, file metrics to evaluate code clones from source files. From this paper result shows that no tool can find all type of clones properly. According to this paper clones detected by CCFinder are 6 and PMD copy/paste detector identified 4 clones .Output of this paper is shown in graphical form which summaries in details about clone detected around and within files.

(Kanika Raheja,2013) [6] et.al proposed a metric based technique. MCD finder is proposed for calculating metric of java program. This approach is using byte code of java program instead of any transformation of source code to calculate metric of source code. This technique is platform independent and at some extent it will found the semantic clones also. There are mainly three phases:- First is adaptation in which it will adapt code according to tool requirements. Tool will take the byte code. Second phase is computation phase, which will compute the code with number of calls, number of public, private, local and non-local variables. This approach used 9 metrics. Third phase is measurement in which it will mapped metrics into excel sheet and comparison is performed on the basis of metric values. Calculated metrics will be stored into database also. In output same metric values are resulted as clones. This approach is not able to detect all type of clones. Because it will declare clones to same metric values and can detect some non-cloned declaration as clone.

Amandeep Kaur [2] et.al mainly focus on defining an algorithm which determine duplicated code piece from programs. An algorithm is based on hasltead metrics, which mainly used to find the complexity of a program related to the number of operators and operands in the program. In post processing phase i.e. in textual comparison a line by line of code is compared rather than by taking token or word. A user friendly interfaced has been prepared with the Visual Basic 6.0 programming language for detecting code clone in an application. Algorithm states that, read any two programs and compute the halstead metrics values. And if these value computed at each line of program would be find equal for their each respective programs then they said to be "clone detected" and location is computed otherwise "Mismatch of All Computer Metrics" would be printed. For analysis , a program in C language has been taken. The result is declared with the specification of lines of code which found to be cloned. But this approach has not declared which type of clones are detected and it can take only two inputs as a source program.

Kodhai.E, Perumal.A, and Kanmani.S [7] et.al proposed a light weight technique to detect functional clones with the computation of metrics based technique with the textual analysis technique. Semantic clones of code A and B are those redundant fragments which show the same behavior. This approach has also been implemented as a tool using Java. The tool efficiently and accurately detects type-1,type-2, type-3 and type-4 clones found in source codes at method level in JAVA open source code projects. In this approach 12 existing method level metrics are used. These metric values are computed for each identified methods in the system. They are as follows:

1) No. of effective lines of code in each method.
2) No. of arguments passed to the method.
3) No. of function calls in each method.
4) No. of local variables declared in each method.
5) No. of conditional statements in each method.
6) No. of looping statements in each method.
7) No. of return statements in each method.
8) No. of function calling in each method
9) No. of inheritance in each method
10) No. of virtual functions in each method

11) No. of overloading constructor in each method
12) No. of overriding functions in each method

By implementing these metrics in a process results in improving the precision and recall values.Type-1 clone method found from line by line comparison whereas type-2 clone found with the comparison of templates. Matching template with exact code are declared as type-3 and method whose output was same rather than they being written completely different called as type-4. But this approach is language dependent that only implemented in Java Open Source.

Jean Mayrand, Claude Leblanc, Ettore M. Merloa [5]et.al proposed a technique to automatically identify duplicate and near duplicate functions in a large software system. The identification technique is based on metrics extracted from the source code using the tool Datrix. This technique uses 21 function metric grouped into four points of comparison. Each point compare functions and their cloning level. Four points of comparison are

1. Name of Function
2. Layout of function
3. Expression
4. Control Flow

An ordinal scale is defined with cloning level. The levels range from an exact copy to distinct functions. The metrics, the thresholds and the process used are fully described. Project A and Project B involves testing for finding each pair of function at each level from level 1 to level 8 and came to conclusion that Level 8, Distinct Control Flow captured all the pairs of function that were not considered clones which means that in both projects, more than 96% of the relations between functions are not clone relations .Level 1 clones are reliable and the case studies showed a negligible level of false accusation. The level of false accusation increased substantially at level 3. These experiments leave many avenues to explore. The first would be to reproduce the experiment and work on the sensitivity of delta definition

(D.Gayathri Devi, 2013) [3] et.al proposed an technique to detect software clones using Distance Clustering. Firstly source code is tokenize and converted into tokens. All the white spaces, etc. are removed from the sequence of tokens and these tokens are concatenated into a sequence. In second step tokens are given as input to distance algorithm and similar statements are identified. First of all a matrix is created .Similar statements are detected using Euclidean distance and they are partitioned into clusters. Each token will be cluster and most similar cluster are detected and are merged. If distance between values is less than threshold than they are clone.

IV. CONCLUSION

This paper suggests the study of metrics based approaches used to detect code clones .This approach can detect all type of clones. Metrics based technique is very efficient. The metric provides to assess quality. The Metric based approach was found more suitable for combination to make a hybrid approach because it can detect other clone types with good recall value and outcome so far obtained by this approach usually shows positive result.

REFERENCES

[1] Abd-El-Hafiz, S. K., " A Metrics-Based Data Mining Approach for Software Clone Detection," IEEE 36th International Conference on Computer Software and Applications,2012.
[2] Amandeep Kaur, Mandeep Singh Sandhu, "Software code clone detection model using hybrid approach ,"in IJCT, Volume 3 No.2, OCT, 2012.
[3] Dr .Gayathri Devi, Dr. M.Punithavalli , "An Effective Software Clone Detection Using Distance Clustring" International Journal of Engineering and Technology,2013.
[4] Dr. Gayathri Devi, Dr. M.Punithavalli, "Comparison and Evaluation on Metric based approach for detecting code clone", Indian Journal of Computer Science and Engineering, Vol. 2 No. 5Oct-Nov, 2011
[5] Jean Mayrand, Claude Leblanc, Ettore M. Merlo, "Experiment on the Automatic Detection of Function Clones in a Software System Using Metrics." in ICSM , pp. 244-253, IEEE 1996.
[6] Kanika Raheja, R. T., "An Emerging Approach towards Code CloneDetection:Metrics based approach on byte code,"IJARCSSE vol 3,issue,2013 .
[7] Kodhai. E, Perumal. A, and Kanmani. S, "Clone Detection using Textual and Metric Analysis to figure out all Types of Clones," in IJCCIS, Vol2. No1. ISSN: 0976–1349 July – Dec 2010.
[8] Rupinder Kaur, H. K, "Evaluation of Token Based Tools On The Basis Of Clone Metrics" International Journal of Advanced Research in Computer Science and Electronics Engineering, 2012.
[9] Samuel A. Ajila, A. S, "Reusing and Converting Code Clones To Aspects-An Algorithm Approach" IEEE IRI,Las Vegas, Nevada,USA,2012.

