HYBRID APPROACH FOR EFFICIENT SOFTWARE CLONE DETECTION

Robin Sharma
M.Tech, Computer Sci. Engg,
RIMT-IET, Mandi Gobindgarh,
Punjab, India
robin1989sharma@gmail.com

Principal Dr. Sushil Grag
RIMT-MAEC, Mandigobindgarh
Punjab, India
sushilgarg70@yahoo.com

Abstract— **Software cloning is a methodology which defines that if any size of code is used apart from its original place to some another place as it is or with some modification to perform same functionality. This activity is also known as reuse of repeating code at many places instead of writing a new code. The original piece of code which is reused is called code cloning and copied form of that code is called a clone of the original. In finding of duplicated code, plagiarism detection also work pretty well but it is not applicable to the large system in finding functional clone and also it is more time consuming even at small scale which make this detection method inappropriate. After going through techniques and methods which were used in last few year papers the hybrid approach sounds more prominent approach among them to detect code clone. This survey paper describes the hybrid approach which is made up of two techniques, one is textual and another one is metric based approach for detecting code clone of all types in open source software system. This approach is used to detect functional clones from source code, which is written in language using Object Oriented Paradigm concepts. The output of the tools which developed based on this proposed approach are further compared with the existing tool in term of recall and precision parameter which shows that a hybrid approach is lightweight technique which gives accurate result being less complex.**

Keywords- metric; code cloning; Textual comparison; Hybrid approach; and code detection

I. INTRODUCTION

Code cloning–an identical or similar piece of code which is matched with other same sized code. It is harder to define what the code cloning is than detecting code clone itself, but in simple words it implies that if code A=code B or code A=code C then code B and C are the exact copies of the original part of A code. Code duplication or copying a code fragment and then reuse by pasting with or without any modifications is a well-known code smell in software maintenance [1]. It should be noted that code reuse is a standard practice in modern programming [3].While continuing to adapt this approach it also add some drawbacks in software system .It also raises the chances of bug propagation, various anomalies in maintenance/modification part, introduction of new bugs if the functionality in system need to extend so in comparison to its benefits, the harm full side of code cloning part needs more attention for detecting code clones and removes them for not becoming an obstacle in software development. Code clone could be of any type that all depend on the programmer's technique and capability of using the code which varies from copying as it is to copy the code but with some modification which would be done at different level in the method. In software system code fragments mainly shows two kind of similarities. They are said to be similar if their program text matches or they can be similar on their functionalities bases if the behaviour among them match. Mainly clones are of four types out of which first three types are under textual similarity and the last type is under functional similarity.

A. Textual Similarity: Based on the textual similarity we distinguish the following types of clones

1) Type I: Identical code fragments except for the variations in whitespace (may be also variations in layout) and comments.

2) Type II: Structurally/syntactically identical fragments except for variations in identifiers, literals, types, layout and comments.

3) Type III: Copied fragments with further modifications. Statements can be changed, added or removed in addition to variations in identifiers, literals, types, layout and comments.

B. Functional Similarity: If the functionalities of the two code fragments are identical or similar i.e., they have similar pre and post conditions, we call them semantic clone sand referred as Type IV clones.

1) Type IV: Two or more code fragments that perform the same computation but implemented through different syntactic variants.

The output of the code clone detection is returned as clone pairs/clone clusters along with their location/occurrence.

- Clone Pair (CP): identical or similar pair of code fragments.
- Clone Cluster (CC): the union of all clone pairs which have code portions in common [2].

The detection technique of codes duplication plays a central role. Researchers have so far develop so may techniques to find clones with each has merit and demerits and it remains the most interested area to research. The main advantage of using hybrid approach over others prevailing techniques is that it is made with blend of Textual comparison and Metric valued based computation technique and other advantages are:

- Approach becomes lightweight with the use of textual comparison.
- Precision value comes high because total comparison overhead is less.
- Obtain high value of Recall because textual comparison is performed only on short-listed candidates
- Output come in clone classes and clone pair
- Able to detect all clone types
- Complexity is less with respect this it takes less time in detection.

For developing a technique which proved to be a better than exist ones the following parameter will help in deciding which technique is more appropriate in detecting code clones.

• **Precision:** sound enough so that it detect less number of false positives i.e., the tool should find duplicated code with higher precision [1].

• **Recall**: capable of finding most (or even all) of the clones of a system of interest [1].

II. LITERATURE REVIEW

Kodhai. E, Kanmani. S, Kamatchi.A, Radhika. R [2] In this paper, we survey the metrics based method with simple textual analysis technique called a hybrid technique which is used to detect function clones from C language taken as source code. This approach is able to detect just type-1 and type-2 of clone from the system and a tool is developed on the basis of this approach using a Java programming language. The Textual comparison is performed to detect exact matched clone of Type-1 and Metric value method helps in detection of near miss clones of Type-2.In metrics technique a set with some of interested method level metric is defined and based on this set descriptive statistics of the metric values calculated for the various identified methods. The input to a tool is a source program called weltab project written in C, which is a vote tabulation system of approximately 11,000 lines. The computed result came in the number of found cloned methods of type-1 and type-2, which then further compared with two other existing tools. The First was phoenix-based detection method [2] who followed the abstract syntax trees (ASTs) and suffix trees technique to detect function clone and the second one was NICAD a parser-based, language specific, lightweight tool, employing TXL and simple text-line comparison to detect functional clones [2].The parameter used to compare the result of these three tools are in term of precision and recall. The proposed tool and NICAD shows 100% accurate result whereas the phoenix based method didn't match up to that level of accuracy. But the main difference which made the proposed tool better than NICAD tool even with same result was NICAD employ external parser TXL for extraction whereas proposed method employ a built-in hand-code parser which make it lightweight. Following are the critic review:

- Couldn't detect Type-3 and type-4 clone.
- Detection is valid to systems which are based on C language which make it language dependent.
- Question on working of proposed tool with more complex system

Amandeep Kaur, Mandeep Singh Sandhu [3] the researchers of this paper mainly focus on devising an algorithm which determine duplicated code piece from programs. An algorithm is based on hasltead metrics, which mainly used to determine the complexity of a program related to the number of operators and operands in the program. The objective was to design and analyse a hybrid approach which is a combination of two techniques which are metrics based method and text-comparison technique. 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. An algorithm states that read any two programs and compute the hasltaed metrics value which basically are:

- Number of unique operator n1
- Number of unique operands n2
- Total no. of operators N1
- Total no. of operands N2

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.The software metrics which used to compute and analyse are certain predictive static metrics like line of code, source line of code, number of operator, number of operands etc.[3].For analysis, a program in C language which is a sorting an array using bubble sort and sorting by using an insertion sort taken

as input for detecting line clone code whose approach is based on the metric approach and text approach (lines).The result is declared with the specification of lines of code which found to be cloned. And in the future work they stated that this designed system can be modified to run on number of platform like windows, UNIX, Linux, Solaris etc [3].Because this implementation is only able to detect code clone in C/C++ so in future with some modification it can be applicable to other programming language like C/C++, Java, COBOL, PASCAL, SQL, and HTML etc.

Critics:

- Application Program takes only two inputs as source program.
- Not mentioned which code clone types detected.

Kodhai.E, Perumal.A, and Kanmani.S [4] the main goal of this research paper is to extend the previous work. The future scope of previous mentioned paper became the problem definition of this research. The same approach is used to extend the detection procedure for code clone from open source software system. A hybrid detection method combination of textual comparison and metric value based technique become capable in finding of all the four types of clones. Semantic clones of code A and B are those redundant fragments which show the same behaviour. The code B is said to colligate code A if its pre and post conditions would be similar. 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 [4].In previous research only 7 metrics are used for computation but at this time in the Implementation stage a set of 12existing method level metrics are used. These metric values are computed for each identified methods in the system. 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[4] are declared as type-3 and method whose output was same rather than they being written completely different called as type-4.Critic point about the paper are:

- Language dependent because its detection only valid on JAVA open source project.
- Metrics value set can be even raised or a class concept as metrics may introduced.

Rubala Sivakumar, Kodhai. E [5] Code cloning is not only bound to some specific area or to some software systems which use programming language but even copy-paste code practices found in web applications. Web applications progressed from web sites by adding up commerce functionality [5].In the development of web sites a scripting language used as front end some of examples are Java Server Pages (JSP), Microsoft's Active Server Pages (ASP), and PHP[5] in which code duplication practice usually involved in making of several web pages. As web application contain webpages which are hyperlinked to each other so at first all web documents are concatenated to make it as a whole large source input file.

A lightweight approach is used to detect code clone which is made with textual and metrics value computation method called hybrid technique. The proposed method is implemented as a tool in DOT NET programming language [5].A set of 7 existing function level metrics are used for the detection of all types of clone functions in web application [5].The proposed tool gives its evaluated result in precision and recall parameter which then further compared with the other existing tool called eMetrics tool. The proposed tool was trialed with the three popular medium sized open source web applications, Quick Auction [20],Web WizForums [22] and Snitz Forums 2000 [21], where a Quick Auction is a basic auction application that can be integrated into other web sites to add simple auctions features and remained other two are web-based bulletin board engines.The eMetrics tool is based on web applications that adopt Microsoft's ASP technology which measures the size of a web application to different granularity levels (including script functions), and then selects homonym programmer-defined functions (written in JavaScript or VBScript) as potential cloned functions[5].The result of comparison showed that the value of precision and recall in term of percentage with the proposed tool using DOT NET gives more higher value with accuracy than the eMetrics tool experimenting with those three web applications. The proposed method has accomplished the desired output with full-automation [5] and detects all the 4 types of clone. But the critic review of this paper is:

- Question on its working with larger and even more complex system.

Salwa K. Abd-El-Hafiz [10] came with an idea of using data mining approach with the metric-based technique for detecting functional clone from software system. A concept of data mining approach which infuses strength in the process is an algorithm called Fractal Clustering. Fractals are used to differentiate the object with respect to its nature or properties whereas clustering work for dividing the system into smaller number of clusters so that each cluster has function within it and Fractals dimensions decide the encapsulation of functions in clusters. The main goal of Fractal clustering algorithm is to divide the software system into various varying size of

clusters contained functions that all are in system and placed according to algorithm scenario. It places all function into three categories of clusters; a unique *primary* cluster and several *intermediate* and *single* clusters [10].
In primary cluster, those all functions would be grouped whose metric values are same and shows same degree of similarity associated with chosen metrics set [10]. The primary cluster detects all the type-1 and type-2 function clones.
The Intermediate cluster contains those functions whose metric values are same or fall within some threshold defined value and it detects type-3 function clones.
The single cluster contains those singles functions which are not matched to any other functions in the system.

The use of Metric based technique is done by defining 8 commonly used metrics which were created in such a way so that each function concept can be included for detection and apart from them another two complexity metric used which relates to McCabe cyclomatic complexity and number of unique paths in functions of software system. During implementation, these 8 metric are placed in 4 different alternatives metrics set $\{m_1, m_2, \ldots, m_D\}$.
The four alternatives metrics are:

- *M*1: {Decl., Stmt., Cond., Loop}
- *M*2: {Decl., Stmt., Path, Cycl.}
- *M*3: {Decl., Stmt., Cond., Loop, Nest, Ret.}
- *M*4: {Decl., Stmt., Cond., Loop, Nest, Ret., Param., Call} [10]

The detection of clones is done by going through mainly 4 phases represented in Fig 1:

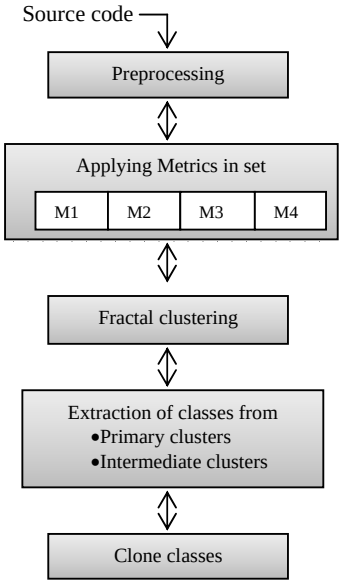

Figure 1: Phases in finding clone

This proposed idea then experimented with medium and large sized C projects which are Weltab and SNNS. First the whole system is represented in the form of fragments and for each fragments, a metrics value is computed. For Weltab software system, only M1 and M2 metric is used which sufficiently and accurately detect all type-1 and type-2 clone classes from primary cluster with high precision value. For detecting type-3 functional clone from intermediate cluster a metric set M1 and M2 with two different threshold values choose which give 92% and 100% precision value.

The Practice with large sized software system SNNS provides the result which clearly stated that for the detection of type-1 and type-2 clone, if the number of metric in set would increase then precision value also increased and number of false positives decrease.M4 with metrics=8 give high precision than M3 whose metrics=6 and similarly M3 give high precision than M2 which have 4 number of metrics in its set. For the detection of type-3 clone, the intermediate clusters are further investigated that satisfy the threshold criteria given in [10] which clearly state that M4 with higher number of metrics in set gives high precision value.
Critics of the above approach:

- Couldn't detect Type-4 functional clone.
- Result can be varying with the change of threshold value.
- Post–Processing phase was manually evaluated

Jean Mayrand, Claude Leblanc, Ettore M. Merlo [11] The main goal of this paper was to detect functional clones automatically from source code of any language using the Metric-based technique. This can be accomplished by using a DATRIX tool framework which is a source code analyser tool set [11] whose main purpose is to change the source code into some Intermediate Representation Language. Out of which only control flow metrics and data flow metrics were selected because they provide internal characteristics information about functions. For automatic detection, this proposed approach experimented on two telecommunication monitoring system in which for finding function clone, 4 points of comparison for clone is performed which are:

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

Basis on these points of comparison, any two functions can be considered as equal, similar or distinct. Under each point of comparison except function name ,there are number of metrics whose description states that what need has to extract as given in [11] and for all these metrics a certain delta or threshold value was assigned which decide whether that function lie in that category or not. For less false accusation, the delta number must be as low as possible. Eight levels on ordinal scale marked up for identifying clone in projects which helps in determining the programming style to was good or bad

adopted in system, as level increase from level 1 to level 8 it indicates that clone in system are less and in worst case at level 1 the copies of function in system are more.
Experimentation on 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 [11].This paper also enlighten the method which can be used in controlling the level of clone in software system. Steps include are:

1. Measurement program: implementation of multi-version source code measurement program.
2. Design principles: setting of guidelines or policies regarding reuse practices of code.
3. Clone monitoring: setup of organisation which monitor.
4. Clone reduction: removal of targeted clone at each level which might vary from creating common libraries to use parameterization**.**

III. DISCUSSION

The techniques which were chosen to make a combination must be opted in such a way so that the limitation of one technique overcomes with the use of other. The simplicity of Textual comparison helps in find of Type 1 clone with high precision value but become inapplicable in detecting of remaining types of clone and to overcome this limitation the second technique should be opt in accordance without being more complex. 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. Furthermore, this approach can be used for experimentation on various other languages.

IV. CONCLUSION

After going through all the papers, it has been found that code smell is a part of code, it is not an error which is visible to the user, but somehow it is affecting the performance of the system or application software which we are using, in terms of memory blockage. It has been also concluded that till now, there is no such platform or language driven software which can implement the code smells of different languages or patterns. The future work of the current scenario is to convert a source code of any language to a generic and flexible code for detecting code clone from systems.

REFERENCES

[1] Chanchal Kumar Roy and James R. Cordy, "*A Survey on Software Clone Detection Research"*, Technical Report No. 2007-541, September 2007.

[2] Kodhai. E, Kanmani. S, Kamatchi. A, Radhika. R, "*Detection ofType-1 and Type-2 Clone Using Textual Analysis and Metrics"*,in ITC, 2010 IEEE.

[3] Amandeep Kaur, Mandeep Singh Sandhu**, "***Software code clone detection model using hybrid approach"*, in IJCT, Volume 3 No. 2, OCT, 2012.

[4] 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.

[5] Rubala Sivakumar, Kodhai. E,"*Code Clones Detection in Websites using Hybrid Approach", in* IJCA (0975 – 888) Volume 48– No.13, June 2012.

[6] Hassan Raza Bhatti, "*Automatic Measurement of Source Code Complexity"*, Master's Thesis submitted at Luleå University of Technology, 2011

[7] Yogita Sharma, "Hybrid technique for object oriented software clone detection", M.E Thesis submitted at Thapar University, Patiala, 2011.

[8] 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. 5 Oct-Nov 2011.

[9] Mohammed Abdul Bari, Dr. Shahanawaj Ahamad, "*Code Cloning: The Analysis, Detection and* Removal", in International Journal of Computer Applications (0975 – 8887), Volume 20– No.7, April 2011.

[10] Salwa K. Abd-El-Hafiz, "*A Metrics-Based Data Mining Approach for Software Clone Detection"*, 2012 IEEE 36th International Conference on Computer Software and Application.

[11] 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, 1996. 1996 IEEE

[12] Chanchal K. Roy and James R. Cordy**, "***Scenario-Based Comparison of Clone Detection Techniques"*, Proc. 16th IEEE Int. Conf. on Program Comprehension, 2008, pp.153-162.

AUTHORS PROFILE

Robin Sharma is persuing her Master's in Engineering in Computer Science from RIMT-IET, Mandi Gobindgarh, Punjab Technical University, Kapurthala. She is currently working on the project of Code Clone Detection for her research work. She has interests in subject areas like data mining, software engineering, web development etc.
Email: robin1989sharma@gmail.com