Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 8, August 2014, pg.297 – 302

RESEARCH ARTICLE

A Combined Weighted Approach to Detect Code Cloning

Himanshu[1], himanshuarya8@yahoo.com
Student, Dept. of Computer Science & Engineering, RIMT - IET, Mandi Gobindgarh, Punjab

Dr. Sushil Garg[2], sushilgarg70@yahoo.com
Professor, Dept. of Computer Science & Engineering, RIMT - IET, Mandi Gobindgarh, Punjab

Abstract -- Code clone detection is one of the useful and required approach to generate the reliable and effective code. There are number of approaches defined by earlier researchers to detect the cloning. These approaches include the textual, statistical and token based approaches. In this present work, three main categories of code clone detection approaches are combined in a weighted form to detect the cloning over the system. The work is presented as a weighted model with the exploration of all three stages under the algorithmic concept. The paper also discussed the problems and the relative solution provided by the presented model

Keywords – Weighted Approach, Model Based, Semantic, Syntactic

I. INTRODUCTION

Code Cloning actually represents the use of existing code segments by a programmer instead of writing a new code. Web is a vast repository for different kind of codes written in different languages. A new programmer when writes the code, instead of writing its own code, the preference is given to the easy means to get the existing. This kind of code can be searched over the web according the project or the module requirement. The programmer uses such code as it gets from web as well as can do the small changes over the code. This all kind of altered or non altered use of existing code comes under code cloning. According to the earlier research, in a software organization about 7% to 23% codes are cloned. The cloning not only affects the efficiency but also affects the reliability of the software system. The criticality of the code cloning increases when the copyrights of the code are reserved to some person or the firm. In such case, the firm has to face some legal action also.

To maintain the reliability of the software system, it is required to identify the code cloning at the earlier stage of software development. There are number of existing approaches defined by different researchers. These approaches are adopted by many existing tools in individual as well combined way. The categorization of these all approaches is shown in figure 1.

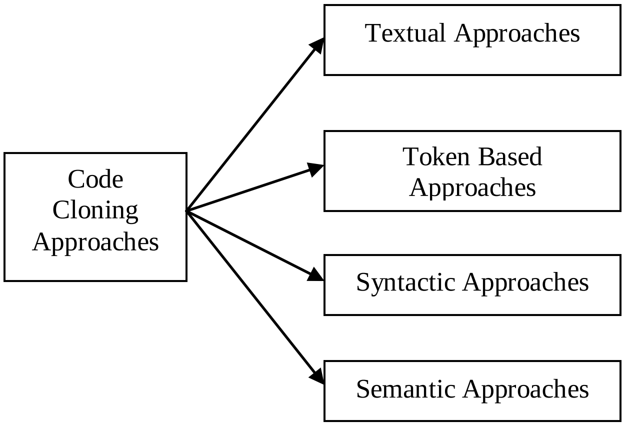

Figure 1 : Code Cloning Approaches

These approaches are mainly based on the structural analysis of the programs so that the accurate code cloning will be identified. The structure level analysis is the basic form clone detection approaches, later on lot of improvements are provided by different researchers. These improvements are performed to obtain the more accurate results from the detection system under token level analysis. The improvement is done in the syntactic and semantic analysis of the program codes. Some of these approaches are language specific and some are language independent approaches. The advantage of these approaches includes the low cost analysis and efficient code clone detection. The basic characteristic adapted by these approaches is shown in figure 2.

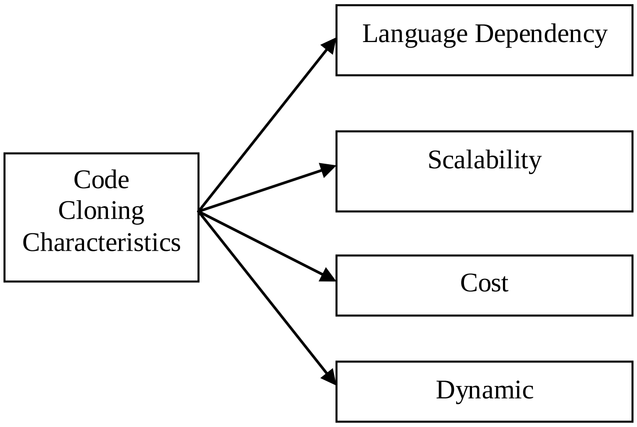

Figure 2 : Characteristics of Code Cloning Approaches

The scalability defines the capability of the code cloning system to work with small software system as well as large software system. The approach must be able to process on individual software module as well as to work on the complete software system as a whole unit. The approach must be cost effective. The cost is here defined in terms of complexity or the efficiency or the robustness of the clone detection system. These approaches must have dynamic nature in terms of inclusion of new constructs or tokens associated with the languages.

In this paper, a multi parameters based weighted approach is defined to perform the code detection. This approach includes the textual analysis, token based analysis and the statistical measures to identify the code clone for a software system. In this section, the explorations to the different approaches available for code clone detection are explored. The section also defined the properties associated with these approaches. In section II, the work defined by the earlier researchers is discussed. In section III, the proposed model for code clone detection is explained along with algorithmic approach. In section IV, the conclusion obtained from the work is defined.

II. RELATED WORK

Girija Gupta[1] has defined a work on code clone detection and designing approach for software system. Author defined the analysis on code fragments under different factors to obtain the software maintenance analysis and reusability analysis. Author defined the modification to the different adapted approaches to check the pasted code. Author defined an activity analysis based approach for risk estimation in code cloning. Author defined the refactor analysis as the major concern to the system to increase the reliability to the system with effective problem analysis. Author defined the better understanding to the system under tree based approaches. The work is defined in the form of a software tool to analyze the software system under cloning analysis. Paul Baker[2] has defined a sematic analysis based on UML sequences based communicating system. Author defined the sequence diagram based analysis so that effective clone detection over the system will be performed. The paper is defined as the concern to perform the detection of cloning at the earlier stage. The work is defined in the form of a framework to perform the semantic check under defect anlaysis so that the effective detection of cloning will be done. Huiqing Li[3] has defined the work in the form of code fragment analysis under the design problem with the code size, maintenance chances and the unification of the system. Author defined a mechanism to detect the code under refactoring and support over the system. The similar code detection algorithm is defined under interactive anlaysis so that the comparative analysis will be obtained over the system for effective clone detection. Bakr Al-Batran[4] has defined a semantic based model for code clone detection for embedded system. Author defined the engineering model analysis to analyze the software development process. In this work, the structure analysis is performed under different form for the simulink programs or the model. Author defined a generalized concept for the behaviour analysis for the structure of the model. Author defined the normal form analysis approach for the implementation of system and to obtain the syntactic and semantic aspects of the program code. Yang Yuan[5] has defined a token based scalable approach for code clone detection. Author defined an accurate model under novel counting approach to analyze the characteristics of the system under segmentation approach. Author defined the experimental model for accuracy analysis with syntactic mapping of the model with existing system. Author defined the clone detection approach under segmented approach so that effective accurate results are obtained over the system.

Radhika D. Venkatasubramanyam[6] has defined a prioritization based approach for code clone detection and management. Author defined the work in the form of tool so that the informed decision can be taken regarding the code clone detection. Author defined the detection approaches under large data sets so that the effective clone detection process will be defined. Author considered the factor analysis under the maintenance overhead so that the software quality of the system will be identified under the quality standard. Author also provided the systematic approach for analyzing and prioritizing the cloning over the system so that the code fixing for the program code will be obtained. Hiroaki Murakami[7] has defined a folding repeated instruction to improve the token based system so that the token based code cloning. Author defined the scalability analysis under the detection tool and approaches. Author defined the token based detection system under the cloning analysis. The proposed approach is based on the speed analysis so that the effective estimation and quantitative evaluation of the system will be obtained. Author defined the effective estimation of usefulness of the tool as well as the approach. Ian J. Davis[8] has defined a source analysis based code clone analysis under the assembler utility. Author defined a complementary technique for the code analysis. Author defined the change detection in the compilation process so that the normalization to the system will be obtained. Author defined the complementary approach for semantic and syntactic analysis of code so that the detection process will be improved. Miryung Kim[9] has defined a study on code clone detection. Author presented the tool as the evolution to the clone method so that the reliable and effective code clone detection will be performed. Author defined the study under different contradiction analysis for the traditional system so that the refactoring of the system will be analyzed. Author defined the system under the limitation analysis. Author defined the code consistency analysis under different approaches. D. Gayathri Devi[10] has defined a comparative study on different approaches for code cloning approaches under metrics based analysis. Author defined the speed quality and cost based analysis. Author defined the metrics based study so that the software reusability will be improved. Chanchal K. Roy[11] has defined a comparision and evaluation approach under clone detection system with qualitative methods. Author provided the tool based analysis over the software system. Author presented the work as a framework that can be applied to any software system independent to the language and other factors. Author performed the categorization of code cloning under different techniques so that effective analysis will be performed over the system. Deepak Sethi[12] defined the work on cloning process on different dataset. Author defined the reliability analysis over the system so that the fragment analysis will be performed over the system. Author develop a project using ASP.net and C# and then Author se a clone detection method, called Solid SDD that can detect clones in XML formats and then display the results diagrammatically. Kanika Raheja[13] has presented a metric based approach for code detection over the byte code.

III. RESEARCH METHODOLOGY

The code cloning the problem defined in most of the software system. Today most of the programmer tries to identify the code online instead of writing the code from the scratch. Such kind of code theft is called code plagiarism or the code cloning. While using this code, the programmer does the smart changes over the code in terms of change variables, splition or merge of statements etc. But, legally any kind of code cloning is an offence. Because of this it is required to identify the cloning over the code. The presented work is about to perform the code clone analysis over two different codes.

- In this work, instead of performing the one to one literal matching, we have defined a flow path analysis to identify the code cloning. The presented approach is expected to provide the solution of detecting the code clone.
- The approach does not require any specific data, we can use any two C codes written in different ways for the analysis. We can test on self generated programs.
- The problem as well the solution is significant. The problem will be identified in terms of cloning is present or not over the code.
- The main principle involved in this work is the detection of duplicate code. If some programmer do the claim for his code. It can be verified from the presented work.
- The work includes but the structural as well as the procedural analysis so that we have combined the existing approaches in a hybrid analysis and environment.
- Significant problems are (i) detection of change variable name in the code (ii)identification of unreachable statements etc.
- Problem is not new, but it is one of the critical problem that is not yet effectively resolved.
- Some existing solutions are available but there is the requirement to provide more effectiveness so that we have defined a hybrid approach for the analysis.
- The problem is feasible because the algorithmic concept is involved here.

In this present work, to overcome the drawbacks of existing code clone detection approaches, a combined weighted approach is suggested to detect the code cloning. This approach is divided in number of sub approaches that will process on input code in parallel. The basic approach model used in this proposed work is shown in figure 3.

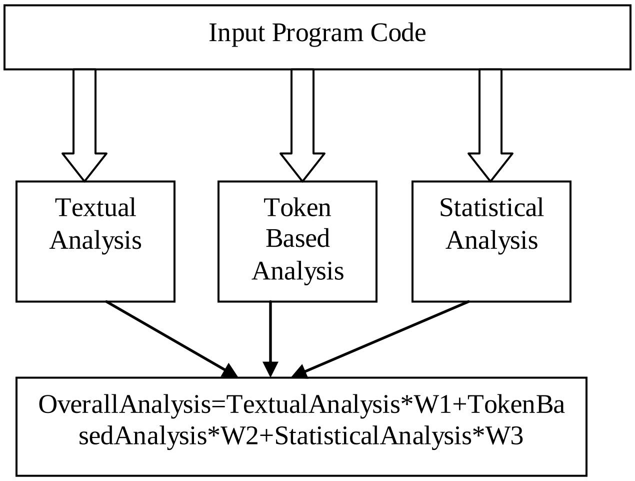
Input Program Code
Textual Analysis
Token Based Analysis
Statistical Analysis
OverallAnalysis=TextualAnalysis*W1+TokenBasedAnalysis*W2+StatisticalAnalysis*W3

Figure 3: Proposed Model

As shown in the figure, the presented model has combined three main approaches called textual analysis, token based analysis and the statistical analysis. The textual analysis is defined as the line based matching of the program. In this work, the textual matching is provided in two different ways. In first way, each line of the source program is compared with the object program in same sequence. It means the exact line by line matching is performed. In second textual analysis approach, each line of source program will be identified over the object program code. The matching in this method is about the content not based on the sequence.

The second approach combined here is token based analysis. In this approach, the complete source and object program is divided in terms of tokens. These tokens are identify as the works excluded the general works or the text used in the program code. The token includes all the keywords and identifiers used in the program. This work flow of this stage is shown in figure 4

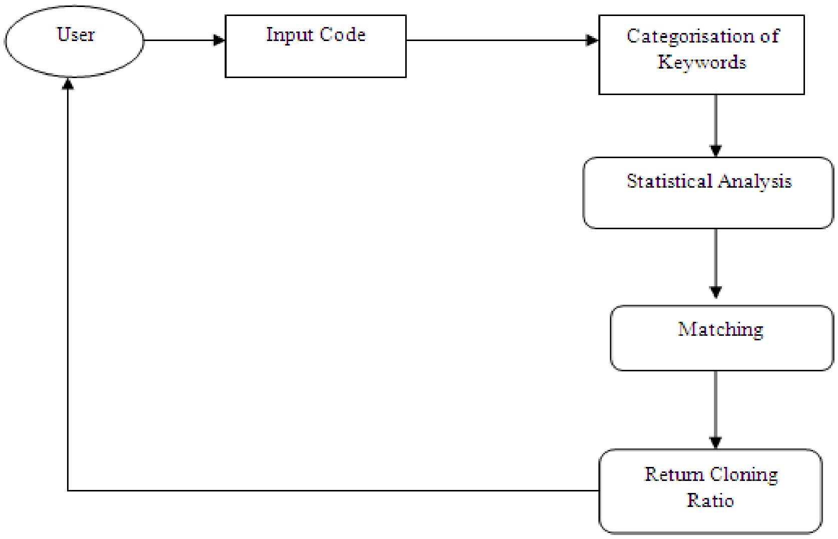

Figure 4: Token based Code Cloning

In the third stage, the statistical analysis over the code cloning is performed. According to this approach the different kind of associated keywords are obtained from the code. These keywords include the identifiers, data types, conditions, loops etc. Once the statistical information is collected, the next work is to perform the statistical matching over the code. The algorithmic approach adapted by the statistical matching in shown in table 1.

Table 1 : Statistical Matching

```
1.    StatisticalMatchingAlgorithm
2.    {
3.    Input the Program Filename called FName
4.     While s = GetLine(FName) <> null
      [Repeat Steps 3 to 12]
3     if Contains(s, "conditions")
4     {
5     Conditions=conditions+1
6     }
7     if Contains(s, "loop")
8     {
9     loop=loop+1
10    }
11    if Contains(s, "datatypes")
12    {
13    datatypes=datatype+1
14    }
15
16    Return (Frequency of Different Literature)
17    }
```

Once the analysis on these all approaches will be done, finally the outcome of these approaches will be combined to generate the overall weighted result. The work has combined different approaches so that the accuracy of the result will be improved. The work also having the flexibility to change the weightage so that the importance to the particular approach can be given based on the software system requirement.

IV. CONCLUSION

In this paper, a new combined weighted approach is defined to detect the code cloning for a software system. The work has combined the textual, token based and statistical approach. The paper has defined the system model as well as explored each algorithmic stage of the work. The work is defined as a generalized language independent model to improve the accuracy of code clone detection.

REFERENCES

[1] Girija Gupta," A Novel Approach Towards Code Clone Detection and Redesigning", International Journal of Advanced Research in Computer Science and Software Engineering 2013 ISSN: 2277 128X

[2] Paul Baker," Detecting and Resolving Semantic Pathologies in UML Sequence Diagrams", ESECFSE’ 05, September 5–9, 2005, Lisbon, Portugal. ACM 1-59593-901-4-0/05/0009

[3] Huiqing Li," Similar Code Detection and Elimination for Erlang Programs", PADL 2010

[4] Bakr Al-Batran," Semantic Clone Detection for Model-Based Development of Embedded Systems", MODELS 2011

[5] Yang Yuan," Boreas: An Accurate and Scalable Token-Based Approach to Code Clone Detection", ASE ’12, ACM 978-1-4503-1204-2/12/09

[6] Radhika D. Venkatasubramanyam," Prioritizing Code Clone Detection Results for Clone Management", IWSC 2013, 978-1-4673-6445-4/13@ 2013 IEEE

[7] Hiroaki Murakami," Folding Repeated Instructions for Improving Token-based Code Clone Detection", 2012 IEEE 12th International Working Conference on Source Code Analysis and Manipulation 978-0-7695-4783-1/12 © 2012 IEEE

[8] Ian J. Davis," From Whence It Came: Detecting Source Code Clones by Analyzing Assembler".

[9] Miryung Kim," An Empirical Study of Code Clone Genealogies", ESECFSE’ 05, September 5–9, 2005, Lisbon, Portugal. ACM 1-59593-014-0/05/0009

[10] D. Gayathri Devi, "Comparison and evaluation on metrics based approach for detecting code clone", Indian Journal of Computer Science and Engineering (IJCSE)2011, ISSN : 0976-5166

[11] Chanchal K. Roy," Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach", 2009

[12] Deepak Sethi," Detection of code clones using Datasets", International Journal of Advanced Research in Computer Science and Software Engineering 2012 ISSN: 2277 128X

[13] Kanika Raheja," An Emerging Approach towards Code Clone Detection: Metric Based Approach on Byte Code", International Journal of Advanced Research in Computer Science and Software Engineering 2013, ISSN: 2277 128X