

Towards the Software Cloning: A Concept

Amandeep¹, Pooja Gupta² and Gurpreet Kaur³

¹Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India, aanya.aman29@gmail.com

²Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India, pujagupta008@gmail.com

³Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India, gurpreetse.cgc@gmail.com

*Correspondence: aanya.aman29@gmail.com

ABSTRACT- Similarity and duplicity of source code has a bad and good impact on software development and maintenance. The copying, modifying, renaming a block of codes are considered as a clones. Software cloning is a process of copying and pasting or finding some similarities between the software for major and minor modifications which are done by the developer to increase the productivity. Anecdotal evidence shows that to maintain the software code is much difficult than the software without code due to their maintenance cost which is increased by cloned codes. This paper discusses what is software cloning, its systematic view including clones analysis, clones detection and clones removal and their techniques and tools.

General Terms: Source Code Cloning, Model Clone, Techniques, Text Based Similarity, Function Based Similarity

Keywords: Software Cloning, Clone Analysis, Clone Detection, Clone Removal, Clone Avoidance.

ARTICLE INFORMATION

Author(s): Amandeep, Pooja Gupta and Gurpreet Kaur;

Received: 11/02/2022; **Accepted:** 23/05/2022; **Published:** 15/07/2021;

e-ISSN: XXXX-XXXX;

Paper Id: IJCSR-010104;

Citation: 10.37391/IJCSR.010104



Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

System engineering is concern with deployment, architectural design and integration whereas software engineering is concern with development, quality, testing and control of the system. Software engineering is an approach to the development, design operation and maintenance of software and its main focus in on to assure the quality of the product, detect and prevent system from bugs by testing or analysis. Software developer are intended to do copy and paste activity to reduce the efforts and schedule.

A code is a symbols, letters, or words given certain arbitrary meanings, used for transmitting messages requiring secrecy or brevity. Code is a system of symbols, rules used to represent instructions on the computer. It also specified as computer program. So, the cloning of such symbols, rules, letters and words has a major problem. Basically, software cloning is nothing it just a copying the content of one software and pasting it on the other software for saving time, efforts and complexity.

From the decade survey on code clones, it is introduced in 1990's. Code clone is one of the factor which making software maintenance more difficult. By analysis of software application it results that the addition of some extra similar functionality but not identical to some existing logic within a system. A code clone is a similar or duplicate source code or created either by replication or some modifications from the original are called clones. The code clone activity increases the maintenance cost

of the software *i.e.*, if changes perform on one code fragment the similar changes will perform on other code fragment and it also causes the code bloating. Moreover, it increases the length of the code.

As, software development life cycle contains two types of clones such as:-

1. Code clones
2. Model clones

Code Clone: Code clone are the code segments in source file which are identical or similar to each other. The duplicated segments are called clones and the method in which copying or modifying is done is called cloning. Whereas, cloning of the codes are present in the implementation phase which is considered as the coding phase. The implementation phase is just after the designing phase.

Model Clone: Like code clones, model clones are the duplicity of the model fragments, its attributes, behavior etc. It is presented in the designing phase of the software development life cycle before the implementation phase and after the feasibility phase.

Cloning has a negative impact in software industry. To make some changes in the software cloning some actions are being performed such as:

(a) **Consistent change:** The consistent change in a clone group of the clone fragments are updated uniformly.

(b) **Inconsistent change:** Inconsistent change the clone fragments are updated divergently [1]. And it increases the risk of fault in system.

(c) **No change:** The previous study shows that around 7% to 23% of the source code in a software system contains code clone [2]. Sometimes it is not effective to remove the clones. So, improve the reusability and maintainability of software

from clone code by using the concept of modularity and refactoring.

Software cloning is just copying and pasting of the contents but it also considered the functionality and syntax of the software programs. Basically, clones are broadly classified into 2 Categories:

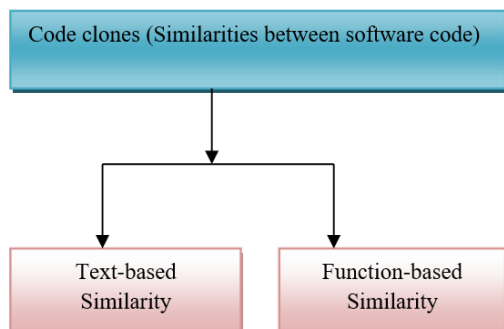


Figure 1: Types of software cloning

a.) Text Based Similarities: Cloning of the text based is referred to the cloning of the software content, description, whitespaces, layouts which causes a terrible in the maintenance phase.

b.) Function Based Similarity: Cloning of the function, syntax of the software programs. Including with the software functions, conditions, interactions are considered.

These text based similarity and function based similarity are comprises into some types which are:

Types of clones are as follows:

(i) Type I-Original Clones: It consists with some similar clones or exact clones which are copied from the original code with small changes at lines, comments and white space.

(ii) Type II-Modified Clones: In this clones the properties of the codes are modified such as structure, variables, names etc. Adding, these are also exhibit the property of type-1 clone.

(iii) Type III-Near-Miss clones: In this type of cloning clones are specified as the addition or deletion of the statements in the copied code from the original code. These exhibit the property of both i.e type-1 and type-2. But this type is little difficult to understand as compared to others and also difficult to detect the clones.

(iv) Type IV-Semantic Clones: In this cloning in which the two code fragments are developed by two different programmers their logic is intentionally and unintentionally same, their concepts, syntax are same for maintaining the functionality between the code fragments.

Similarly, type-1, type-2, type-3 are comes under text-based similarity and type-4 comes under function-based similarity.

Moreover, a different clone which is arises due to unintentionally or accidental copy and paste activity which behaves like similar to the original code clones are called as accidental clones or unintentional clones. It also contains same set of API's. The cloning has a bad smell in the software industry. So, it is necessary to find the ways for the occurrence of the software clones: Some of them are as such:

- i). Due to copy and paste activity which increases the maintenance cost of the system.
- ii). Due to minor changes in code segments called change-prone clones and it reduces the maintenance cost of the system.
- iii). Volatile clones these are created and removed during the observational period, these clones are arises for short or particular span of time, so these clones are also called as short-lived clones.

It doesn't only increase the maintenance cost but it also increases the design problem such as redundant component are present in the form of duplicate fragment.

2. SYSTEMATIC VIEW OF SOFTWARE CLONING PROGRESS

Software cloning is just a duplicity or similarity between the code fragments. To find the clones is a biggest task. So, the classification of the software cloning is supposed, which are followed in a systematic form for the removal of software clones and make the software reliable and consistent. And also increases its quality. The process of software cloning intended with some important steps which are as follows:

- I. Analysis of clone
- II. Detection of clone
- III. Removal of clone

This systematic view of software cloning is described below with the explanation of their every module.

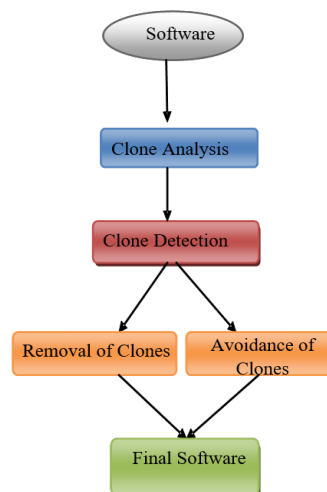


Figure 2: Systematic View of Software cloning

2.1 Analysis of Clones

The analysis of the clones is the initial step in the cloning process. In this clones are analyzed from the source codes with different ways:

- I. Static Clone analysis
- II. Dynamic Clone analysis.

2.1.1. Static Clone Analysis

The static analysis of the clones from software contains various static tools and techniques. There is no need to execute the program/code but it follows the formal review by the expert. The tools which are used for analysis of clones are language independent [3, 4]. These tools are used for large software systems where code is large and very difficult to review it manually. Static analysis of the software cloning classified into two constraints:

i. Categorization: In this process the clones are categorized according to their properties like type-1 clones are gathered into one cluster. Type-2 clones are into another cluster.

ii. Visualization: In Visualization process, various tools are used some are Duploc, CCFinder, Johnson, Clone Detective, Clone Dr, Clone Digger. And the clones are visually present in the form of scatter plot [5]

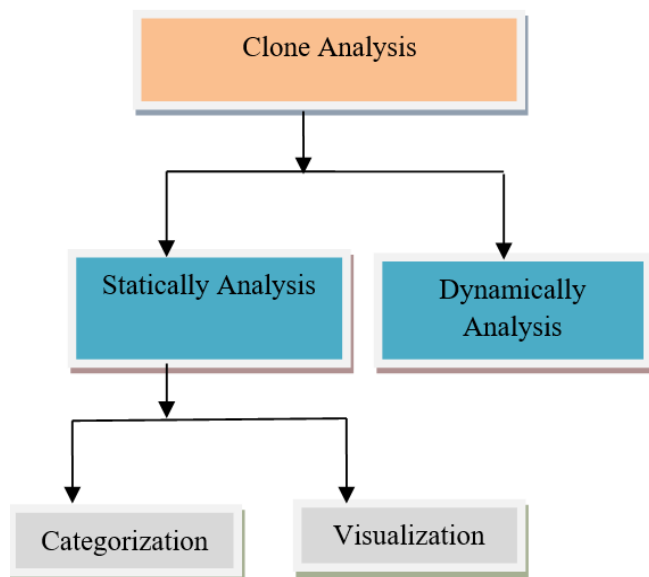


Figure 3: Software Clone Analysis

2.1.2 Dynamically Clone Analysis

After the execution of the one program, then analyses of software code is being performed. It reduces the testing time. Various tools which are used:

1. Token-based Tool
2. Parse-based Tool
3. Data-based Tool
4. Match detection tool

Dynamically clone analysis is basically relies on the tools and automatically executing the source code.

2.2 Detection of clones

Once clones are analyzed then clone detection process is performed. The main aim of clone detector is to find those pieces of code from source unit which have high similarity. There are a number of tools which are considered for clone detection process.

Furthermore, String based technique is one of the simplest clone detection technique. It splits the program into strings and then compare these with each other to find any clone if exist. Token based clone detection technique is better than string clone detection technique. In this case the application programs are decomposed into set of tokens and then compare it with each other to detect hidden clones. Parse tree clone detection technique develops the parsing tree structure for detecting clones. Metric based software clone detection technique is most likability for many software developers. In this technique a various software metrics are computed and then by analyzing the relevant metrics for detecting software clone.

Adding, Hybrid clone detection technique is complex and very effective technique for detecting clones in software application as compare to above described technique. In this case two or more existing clone detection techniques are merged to find clone of code. In technical terms the clone detection technique that uses a combination of syntactic and semantic characteristics is called hybrid clone detection technique. Following table depicts the technique with their relevant tools for the detection of the clones. The scenarios for the detection of the clones are as follows

Table 1: Tools and Technique for the detection of clones

Techniques	Tools	Type
Text-based Technique	1. Johnson 2. NICAD	Near-miss clones are identified
Lexical-based Technique	1. Dup 2. CCFinder 3. CP-Miner 4. Gemini	Modifies clones and Near-miss clones are identified.
Syntactic-based Technique	1. CloneDr 2. Ccdiml 3. Deckard	Only Near-miss clones are identified
Semantic-based Technique	1. Deckard	Exact clones are identified.

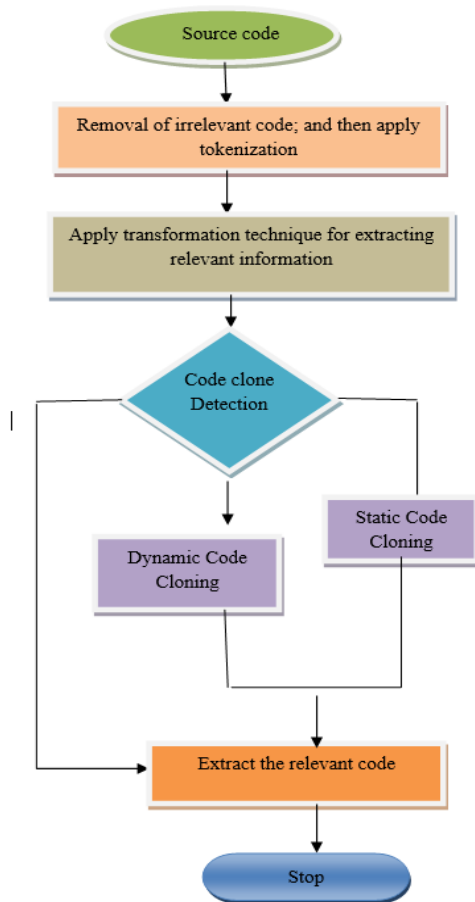


Figure 4: A General Clone Detection Process

In this clone detection process a source code is taken from where irrelevant information will be removed from the code, then apply tokenization approach. Tokenization is a technique from which relevant information is easily extracted in form of tokens. After this irrelevant information will be removed and relevant information is gathered by applying clone detection technique. In case, detection technique is not supposed to be applied for the detection of clones from the source code then fetch out the information and apply it on the tool for detection. Whereas, detection techniques are followed then it relies on the situation where the static code detection approach will be used or dynamic code detection approach. When static code detection techniques are considered then only a formal review of the code is reviewed by the experts. In dynamic code detection approach tools and techniques are used. After extracting the information the clones are represented in a scatter plot.

Independent component: In clone detection, clones between the two dependent codes are considered as a clone pair. If the code fragment shows as a clone pair has high cohesion and is in less coupling with other components of the software then it is only considered as an independent component. Then clones must be removed otherwise the clone should be avoided.

2.3 Removal of Clone and Clone Avoidance

Now, the clones are detected from the software. So, it requires removing the clones from it. The removal of the clones from the

software is must for the consistency and reliability. The removal of software clones are easily removed after the detection. It can be removed by manually through modification, renamed or by some developer dependent concepts.

3. CONCLUSION

Software cloning has a live problem in the industry and has a vast research area with plenty of work on detection and removal of clones from the software. There is healthy work is required in software cloning process mostly focus on software clone removal and software clone detection. Moreover, the defined techniques and tools are language dependent which causes trouble during the dynamic clone detection process. This paper represented the roadmap of all the techniques and tools to help the researcher for selecting the right tool and technique for their preferences.

REFERENCES

- [1] R. Komondoor and S. Horwitz, "Using Slicing to Identify Duplication in Source Code", in Proceedings of the 8th Int. Symposium on Static Analysis, 2001.
- [2] C.J. Kapsner, M.W. Godfrey, Supporting the analysis of clones in software systems: a case study", in Journal of Software Maintenance and Evolution: Research and Practice 18 61–82, 2006.
- [3] Chanchal K. Roy, James R. Cordy, Raner Koschke, "Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach", School of Computing, Queen's University, Canada, 2009.
- [4] Chanchal K. Roy, James R. Cordy, Raner Koschke, "Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach", School of Computing, Queen's University, Canada, 2009.
- [5] S. Schulze, S. Apel, C. Kästner, "Code Clones in Feature-Oriented Software Product Lines", Proceedings of the ninth international conference on Generative Programming and Component Engineering GPCE'10, Eindhoven, The Netherlands, Page(s): 103-112, 2010.
- [6] Jeong, Y. J., Cahe, H. S., Chand, C. K., "Semantics based Cohesion and Coupling Metrics for Evaluating Understandability of State Diagram", 35th IEEE Annual Computer Software and Applications Conference, 2011.
- [7] Mohammed Abdul Bari and Dr. S Ahmed, "Code Cloning: The Analysis, Detection and removal", International Journal of Computer Applications, Vol 20, 2011.
- [8] D. Rattan, R. K. Bhatia, M. Singh, "Software clone detection: A systematic review", in Information & Software Technology, Volume- 55, pp. 1165-1199, 2013.
- [9] Ritu Garg and Rajesh Bhatia, "Clone Removal v/s Clone Avoidance", International Journal of Scientific Engineering and Technology, ISSN: 2277-1581, Volume No.3 Issue No.6, pp : 765-766, 2014.



© 2022 by the Amandeep, Pooja Gupta and Gurpreet Kaur. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).