

An Expository Perception on Code Clone

Shagun Arora

Assistant Professor

PG Department of Computer Science
BBK DAV College for Women, Amritsar
shagun.arora490@gmail.com

Abstract-- Software industry is the fastest growing diversion today. As the needs of software growing rapidly, maintenance of such type of software are getting more intricate, which is a foremost concern for the industry today. Numerous factors formulate the software maintenance cumbersome. Code clone is one of these factors. It is an exercise to replicate such code specks by copying which are common in different software. Although, it seems to be easy to use same code for different software but it complicates the task of maintenance of software and further changes required on clone part of the software. This study focus on the different types of code clone techniques and their related issue. Moreover, this study also different benefits and issues of code clone.

Keywords—Code Clone, Types of Code Clone, Merits and Demerits.

I. INTRODUCTION

In order to increase the productivity in the software development the developers copy the already existing code specks and then reuse them followed by some minor or major alterations. This type of reuse technique results in the replicas of the already existing code specks in the base code. These replicated or duplicated code fragments or specks are known as code clones and the procedure is known as software cloning. There are different definitions of software cloning. Duplication of code is one of the bad practices in software development that increases the maintenance cost. An error found in one part of the code requires rectification in all the replicated specks of the code. So, it becomes mandatory to find the associated specks throughout the code [2,3]. Moreover, cloning a code speck that contains any unknown fault may result in propagation of that fault to all copies of the faulty specks. This paper discusses the reasons to use code clone techniques, merits and demerits of code cloning.

II. BACKGROUND

If two specks of the source code are similar to each other then they are known as code clones. There are basically two categories of software cloning one on the basis of textual similarity and functional similarity. There are several reasons due to which software clones occur for instance code reuse by copy paste, program restructuring, programmers' limitation, language limitation, forking, cloning by accident etc. Code clones results in bug propagation and if a bug is detected in a code speck then all the code specks need to be checked against that bug. This study divides in different sections. Section II describes the types of software clones,

III. TYPES OF SOFTWARE CLONES

There are two types of similarities between the two code specks. Two code specks can be similar either on the basis of program texts or on the basis of functionalities.

- a. **Textual Similarity:** Textual Similarity of the code clone occurs when the cloned code is as it is copied from the original text. This may includes some extra text like comments. On the basis of textual similarity there are following three types of clones
- **Type1 (exact clones):** The two program specks are similar or identical except for the variations in white space (blanks, new line ,tabs) and the comments. Figure 1 shows the scenario of type 1 code clone.

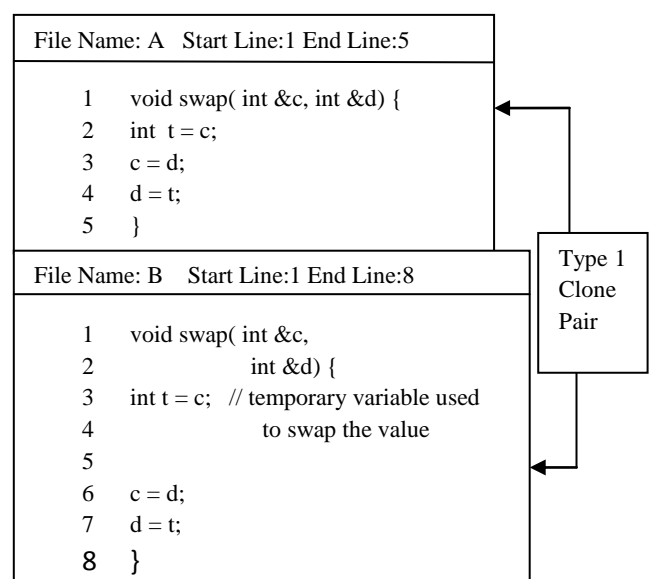
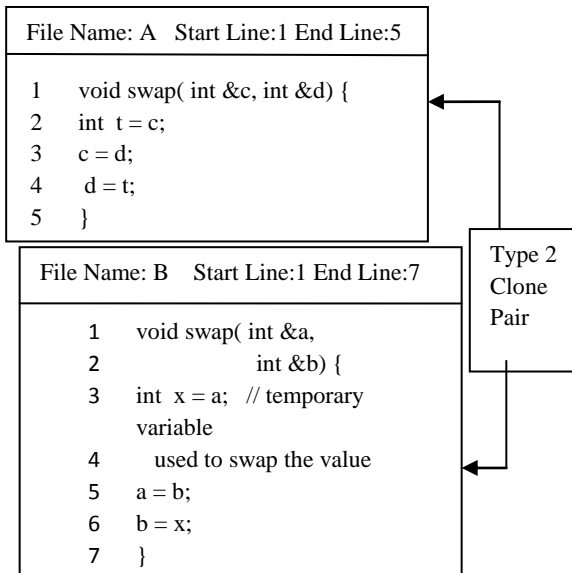


Figure:1 An example of type 1 clone Pair

Figure 1 states how type 1 code cloning can be possible. The file name A and B are textually similar except for the variations in comment, whitespace and layout.

- *Type 2 (renamed/parameterized clones)*: The program specks that are identical syntactically/structurally but vary in identifiers, types, literals, layouts and comments. Figure 2 illustrates type 2 code clone.



Figur:2 An example of type 2 clone Pair

In the figure 2 the file name A and B are syntactically similar except for the variations in comment, variable names and layout.

- *Type 3 (Near miss clones/Gapped clones)*: Two copied code specks are identical but with alterations such as added or removed statements and the use of different identifiers, literals, types, whitespaces, layouts and comments [3].

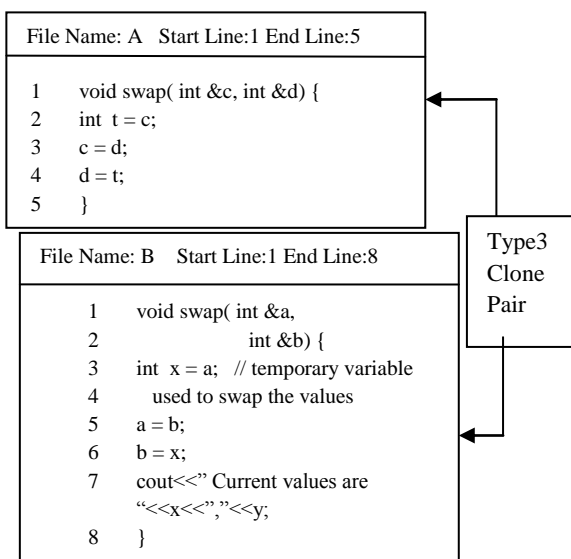


Figure 3: Type 3 Code Clone

In the above figure in addition to the changes in the comment, whitespaces, layout and variable name, a new line (line no. 8) has been added in file name B.

- b. *Functional Similarity*: If the two code specks are similar or identical in functionality i.e. they have identical pre and post conditions then these are known as semantic clones and also called as type 4 clones.
- *Type 4 (Semantic Clones)*: Two code specks are semantically identical but differ syntactically [1]. Figure 4 shows the illustration of Type 4 Code clone.

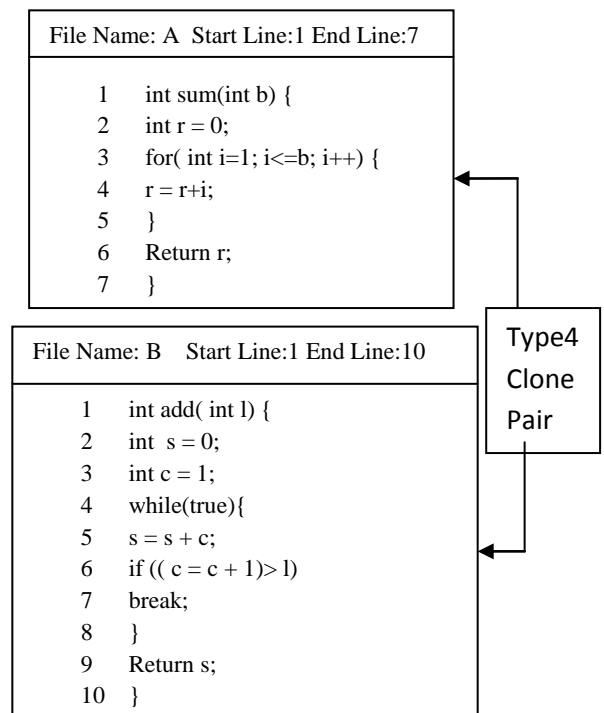


Fig:4 An example of type 4 clone Pair

In figure 4 both file name A and B represents function. The textual content is different in both the functions but both the functions are semantically similar i.e. they do the same thing i.e. addition.

IV. REASONS FOR CODE CLONING

There are some reasons for software cloning and we list them below:

1. *Development Strategy* – Clones can be present in the software system due to different reuse and programming approaches for eg- reuse by copy/ paste in which the existing code is used by simply copying and pasting without any alteration. The design, functionalities and logic can also be reused if there is an already existing solution. The programming approach can be for example merging the two identical software systems which have same same functionality in order to produce a new one and clones can be present in the merged system.
2. *Lack of Restructuring*: The software programmers sometimes delay in restructuring the code due to time constraint which results in the increased maintenance costs.
3. *Programmers Limitations*: There are various limitations for the programmer for eg there is a time constraint for the

programmer and in order to deliver the software within the given time frame he uses the already existing solutions which result in software cloning [4]. Another reason could be lack of knowledge to the developer in the problem area so he then uses the existing solutions which again results in software cloning.

4. **Language Limitations:** Sometimes the programmers copy and paste the code due to the flaws in the programming languages. For eg some programming languages do not have abstraction mechanisms such as inheritance, generic types (templates in C++), parameter passing which is missing in assembly language so the developers implement these again and again and this repetition results in cloning.
5. **Forking:** Forking is to reuse the identical solutions with the hope that they will be divided notably with the development of the system.
6. **Fear of using the New Code:** Programmers are afraid of using fresh ideas in the existing software because they are afraid of the fact that the introduction of new code may result in a long software development life cycle.
7. **Cloning by Accident:** Clones may be introduced in the software by the programmers accidentally for eg, It may happen that two developers implement the same logic coincidentally which results in cloning. Another reason could be using the same set of APIs or libraries to execute similar protocols also leads to accidental cloning.

V. MERITS OF SOFTWARE CLONES

The various benefits of software cloning discussed in this section:-

1. **Helps in understanding the program:** If we understand the functionality of a code speck then it is possible to easily understand other files that contain a copy of that particular speck.
2. **Detects Harmful Software:** Clone detection techniques help in finding harmful software. We can compare one harmful software family with another in order to find the parts of one software that match with another.
3. **Finds the Usage Patterns:** If all the cloned specks of the same source speck are detected then all the functional usage patterns of that software speck can be found.
4. **Detects Plagiarism:** If we find the identical code then it can be helpful in detecting the plagiarism.
5. **Maintenance Benefits:** There are several maintenance benefits of clones for instance in order to keep the software architecture clean and understandable clones are introduced in the system. Keeping the cloned specks in the system may also speed up the maintenance.

VI. DEMERITS OF CLONES

Besides some merits of software clone there are some demerits of software clone. This section enlists some of the demerits of software clone.

1. **Higher Maintenance Costs:** If the code clones are present in the software then it will increase the post implementation effort as well as cost [5].
2. **Error Propagation:** If there is an error in the code speck and if we paste that code speck in different places then that error will be propagated in all the code specks. So, the probability of error propagation increases.
3. **Bad Design:** Cloning can also result in the bad design, lack of good inheritance structure or abstraction. So, it becomes difficult to reuse the part of the implementation in the upcoming projects.
4. **Impact on the system improvement/ modification:** If there is a duplicated code in the system then one needs additional time and effort to understand the already existing clones so it becomes tedious to add the new functionalities in the system or change the existing ones.
5. **Strain on resources:** Code cloning increases the size of the software system which puts a lot of strain on the resources of the system so the overall performance of the system degrades.

VII. CONCLUSION AND FUTURE PERSPECTIVE

In nutshell, the presence of code clone is being recognized as an emerging cause of concern in software industry. Due to the presence of code clone maintenance of software has become very difficult. Therefore, identification of code clones becomes extremely necessary in order to avoid the problems caused by them. This study focuses on reasons of code cloning, types of code cloning benefits and drawbacks of code cloning. The future aspect of this provides the simulation results to create differentiation between different types of software clone.

REFERENCES

- [1] Sheneamer, A., & Kalita, J. (2016). A Survey of Software Clone Detection Techniques. *International Journal of Computer Applications*, 137(10), 1–21.
- [2] Rattan, D., Bhatia, R., & Singh, M. (2013). Software clone detection: A systematic review. *Information and Software Technology* (Vol. 55). Elsevier B.V.
- [3] Khatra, N., & Rani, S. (2014). Code Cloning: Types and Detection Techniques, 5(5), 198–201.
- [4] Roy, C. K., & Cordy, J. R. (2007). A Survey on Software Clone Detection Research. *Queen's School of Computing TR*, 115, 115.
- [5] Morshed, M., Rahman, M., & Ahmed, S. (2012). A Literature Review of Code Clone Analysis to Improve Software Maintenance Process.