

SOFTWARE CODE CLONE DETECTION MODEL USING HYBRID APPROACH

Amandeep Kaur

Student, M.Tech. Deptt. of Computer Sc.
Bhai Maha Singh College of Engineering

Mandeep Singh Sandhu

Assistant Professor, Deptt. of Computer Sc.
Bhai Maha Singh College of Engineering

ABSTRACT

The aspiration of this study is to understand and analyze the concept of software Cloning and its detection. Software cloning is an acuity in which source code is duplicated. Software cloning and its detection is one of emerging and most dominant area of research in the field of software engineering. There exist numbers of techniques to detect clone in software. The focus of this study will be given on acquiring and analyzing the concept of hybrid clone detection technique. We will try to devise an algorithm for detecting duplicacy in the software by using hybrid software clone detection technique. The algorithm will first compute the required software metrics that provide sufficient information regarding the software application and then depending on software metrics matches the clone will be detected. While detecting clone we will focus on line clone rather than token or word.

Keywords

Software Cloning, Hybrid Approach, Software Metrics, duplicacy etc.

1. INTRODUCTION

In software engineering, software cloning is the concept in which it is prevalent to reuse code fragments by simply copying and pasting the code with or without any modification. Due to copy and pasting software systems often contain sections of code that become almost mirror replica of one another, such a similar code is called code clones or cloning of code. It should be noted that code reuse is a standard practice in modern programming. Some of the major disadvantages of code cloning by using the concept of replication and copy-paste programming is that it leads to code bloat that significantly increases the technical depth of software products and making software maintenance expensive and time consuming [1]. The existing research shows that code cloning plays a momentous role in large software applications.

As said above software code clone [5] is defined as the fragment of code that is very similar to some another code that exist in the same of another file of the same application. In software engineering, code clone detection is one of the most favourable areas of research. Clone detection is live problem in industry and an active research area with plenty of work on detecting and removing clones from software. In technical terms code cloning is the progression of replicating code fragments. Cloning is mainly used to reduce the burden of a programmer of rewriting the same code again and again. In other words it introduces redundant code by merely copy and pasting the fragment of code at different location in the same file or in different files on the same application. Clone detection is not an

easier task rather it is very difficult to find the code clones in a software application. The research studies on open source and commercial code shows that around 66% of cloned code is modified, i.e. it's not an identical clone. There exist numbers of techniques [9] for detecting redundancy or clones in the software application. Some of the techniques are given as below:

- Text or String based clone detection technique
- Token based clone detection technique
- Abstract Syntax Tree clone detection technique
- Program dependency Graph clone detection technique
- Metrics based clone detection technique
- Hybrid clone detection technique

String based clone detection technique is one of the simplest clone detection technique. It divides the program into strings and then compare the string 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 program is decomposed into set of tokens and then compare each token with other to detect clones. Parse tree clone detection technique develops a parse tree of code to find clone. Metric based software clone detection technique is one the favourable choice of many software developers. In this case various software metrics are computed and then by analyzing the metrics one become able to detect software clone. Hybrid clone detection technique is complex but very effective technique for detecting clones in software application as compare to above said technique. In this case two or more existing clone detection technique are combined 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.

The objective of this research is to design and analyze a hybrid technique for detecting software clone in an application. Clones do not come into existence itself. Let us now understand why software professional introduce cloning the software application. The research has shown that the work load pressure [3] on software developer convince many software developers to use already implemented and tested software. It is noticed that copy-and-paste reuse is one of the most favorable methods of software cloning for software developers when entire library is of no use or when the elements to reuse are individually relatively small. The reason behind the occurrence of clone in the software is developer himself. The various reasons of introduction code clones in the software application are time limit, language limitation, reuse, accidental, developer performance, risk in new code etc.

2. OBJECTIVE OF THE STUDY

The various objective of this study are:

- To recognize “What is Software Cloning”
- To comprehend various software clone detection technique.
- To understand the need of software clone detection techniques
- To lay out various hybrid clone detection technique.
- To appreciate the working of hybrid software clone detection technique by devising an algorithm for detecting clones in the program using hybrid clone detection technique
- To develop graphical user interface for hybrid software code clone detection technique.

3. PROBLEM DEFINITION

We want to propose a hybrid clone detection model with user friendly interface that is capable of detecting clone in software by analyzing the various software metrics and then comparing each line of one file (application program) with every line of second file(application program).

In the proposed software clone detection model first of all various software metrics like line of code, Source line of code, software complexity and some of hasthead metrics will be computed. Let us understand the term metric. Software metric [8] is one of the important aspects of software engineering acts as an indicator for software attribute. It plays an important role in understanding the important concepts in the field of software engineering. The concept of software metrics came into existence 1970, the credit of which goes to Wolverton who performs a research on production ratio of the programmer by using the concept of LOC i.e. line of code. Further Somerville states that the metric is mainly classified as control metric and predictive metric. Predictive metric are able to determine both static as well as dynamic characteristics of the software. In this research we will focus on static predictive hasthead metrics. After computation of various software metrics the two source files will be compare line by line to check any redundancy, it should be noted that only line redundancy in the code will be checked and detected.

4. HYBRID CLONE DETECTION MODEL

We have developed the user friendly interface of hybrid clone detection by using one of most commonly used programming language i.e. Visual Basic 6.0. The code is based on the algorithm as given below:

```

Read PROGRAM1
Read PROGRAM2
Compute loc1, CC1, fn1, f1N1, f1n2, f1N2
Compute loc2, CC1, f2n2, f2N2, f2n2, f2N2
    
```

```

If (loc1==loc2 ||CC1==CC2||
f1n1==f2n1||f1N1==f2N1||f1n2==f2n2||f1N2==f2N2) Then
While not End (PROGRAM1)
Read line fl1
While not End (PROGRAM2)
Read line fl2
If (fl1==fl2) Then
Print “ Clone Detected”
Print line fl1
Exit
Else
Read nextline
Endif
While end
Read nextline
While end
Else
Print “Mismatch of All Computer Metrics”
End if
    
```

The visual user friendly interface based on above algorithm is as shown in the following diagrams:



The following screenshot shows how the above designed model will upload the two application program, compute their required metrics and then comparing and analyzing it for any clone if exist. The interface will show the clones in the form of software code lines.



Figure: Uploading application program files

After uploading the required application program file the designed model will compute the various halstead metrics as discussed above. After comparing the metrics of two program under study the designed solution will detect clone. In this case clone will be detected at line level i.e. we are focusing on determining duplicacy at line level only. The following picture will show how the design model show cloning in code.



Figure : Hybrid Clone Detection

5. ANALYSIS:

Let us consider two input files prog1 and prog2 in C programming for sorting an array using bubble sort and by using an insertion sort. By using the proposed model we will detect the line clone code in two source files by using the metric approach and text approach (lines).

Prog1

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a[10],i,j,n,temp;
    clrscr();
    printf("\n Enter the number of elements:");
```

```
scanf("%d",&n);
printf("\nEnter the elements:");
for(i=0;i<n;i++)
    scanf("%d",&a[i]);
for(i=0;i<n-1;i++)
{
    for(j=0;j<n-i-1;j++)
    {
        if(a[j]>a[j+1])
        {
            temp = a[j];
            a[j] = a[j+1];
            a[j+1] = temp;
        }
    }
}
printf("\nArray after sorting is :");
for(i=0;i<n;i++)
    printf("%d",a[i]);
getch();
}
```

Prog2:

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a[20],i, p, n, t;
    clrscr();
    printf("\n Enter the number of elements:");
    scanf("%d", &n);
    printf("\n Enter the array elements: ");
    for(i=1;i<=n;i++)
    {
        scanf("%d", &a[i]);
    }
    a[0]=.001;
    for(i=2;i<=n;i++)
    {
        t=a[i];
        p=i-1;
        while(t<a[p])
        {
            a[p+1]=a[p];
            p--;
        }
        a[p+1]=t;
    }
    printf("\ Sorted array is: ");
    for(i=1;i<=n;i++)
    {
        printf("\n%d",a[i]);
    }
    getch();
}
```

When the proposed model is used it find the following lines of code and declared it as clone code as the same lines of code are available in two different sorting program as given above.

Clone detected

```
#include<stdio.h>
#include<conio.h>
```

