Design and development of overhead detection and improvement system for AOP

Jashandeep kaur¹, Rasbir Singh²
¹Research scholar, ²Assistant Professor
Department of CSE
RIMT-IET Mandigobindgarh

Abstract—In software engineering the problem of cloning is very common. Most of the developers who are indulge in coding phase of the SDLC tries to copy the code which occurs again and again in code. Hence it is difficult to maintain the cloned data. It is not easy to recognize the original and copied data. Hence to solve the problem of cloning in system code, large number of techniques is developed. Clone detection models in UML inspect duplicate parts in the software, which can be rephrase and lower the maintenance cost. The technique used in this deals with the class diagrams of AOP system. This technique is an aid to detect the overlapped activities before completion of the SDLC process. In this two files are compared for the purpose of detecting the ratio of cloning among both of the files. The files compared in this are XML files. The efficiency of the technique is proved after simulation. Hence it is observed that the technique of this is much better than the traditional techniques for removing or detecting the cloning.


I. INTRODUCTION

Cloning is mostly exists at the time when the code of the software is written. In coding phase most of the times developers tries to copy and paste to the code which is reused in coding. This pasted copy of the code is known as the clone of the original data. The code which is copied is known as the code cloning. It is very tedious task to recognize the original code and clone of the original code. The part of the code which is copied is known as fragments. It is not easy to maintain the clone of the code as compare to the original code [9]. Hence there is a need to remove the clones from the code since cloning leaves adverse effects on maintenance phase of the software [27].

Let’s assume there is huge amount of coding of a system is available and the whole code is a clone of original code. Hence to maintain such system is quite expensive as compare to other systems. It is very difficult task to remove or finding clones from such systems. The copy, paste leads to the in-accuracy in maintenance of the system. There are many research have conducted to generate such technique which can find the cloning automatically when it exist [19].

II. TECHNIQUES OF CLONE DETECTION

For detecting or finding the cloned code is a process which includes two phases. These two phases are transformation and comparison phase. In first phase the source code is converted into internal form so that comparing becomes easy by implementing the algorithms. In second phase the code is compared in order to find the cloned code. Various techniques are developed for clone detection. Each of the technique has its own merits and demerits. Some of the clone detection techniques are as follows:

A. String Based

String Based is an approach used for clone detection in file. This technique follows the idea of string transformation for converting the file format to the internal format. String transformation makes a code variant to the programming codes. In comparison phase various string comparison algorithms are applied. In this technique following is also included:
- Simple line Matching
- Parameterized Line

B. Token Based

Token based is another approach for clone detection. It is much reliable and able to generate more accurate results. In this technique transformation of the code is done by applying algorithms. The transformation algorithm is created by using a stream known as token. This token is extracted from code itself. Transformations performed by using lexer (tokens) make the comparison phase easy.
- Parameterized matching.

C. Parse-tree Based

Parse tree based is an algorithm used for clone detection. It is a complex algorithm which creates the parse tree. Parse tree based algorithm is used in various applications because it allow to add distinct algorithm for comparison also. It includes the following:
- Metric Fingerprints.

III. PROBLEM FORMULATION

UML based development appears to progress. In object oriented development, the complete working of the software is provided by the UML. Analyze, designing and implementation of the software based systems are done using standard modeling language i.e. UML. Clone detection models in UML inspect duplicate parts in the software, which can be rephrase and lower the maintenance cost.

IV. PROPOSED WORK

This approach basically deals with the class diagram of the AOP system. Two or more activities communicate with each other on the bases of association, condition and constraints. In this approach suffix array is implemented using array list to detect model clones. This will helps the
developer to detect the overlapping of the activities at earlier stage in the software development process; suffix array gives faster result due to lesser memory consumption. The proposed work is done on the basis of AOP, by considering all the aspects of the software based system.

The main objectives of the proposed work are:

- This approach is used to enhance the existing system to detect the overlapping between two UML Diagrams of the system.
- Proposed approach used based upon the Aspect Oriented Programming and the existing approach is on the bases of Object Oriented Programming.
- This approach is detect the detail overlapping between the two similar Class Diagrams

V. METHODOLOGY.

In this propose work the AOP systems are proposed developer to detect the overlapping of the activities at earlier stage in the software development process. The methodology of the proposed work is defined below:

1. **Upload two XML Files**
2. **XML Parsing**
3. **Show all the activities and function of the specific activity**
4. **Apply technique to detect similarities between two activity diagrams**
5. **Show the different similarities between two in a tabular form**
6. **Show the collaborate result for the two different activity diagrams in tabular & graph**
7. **Exit**

Figure 1 Block diagram of proposed technique

- Initially the two ULM files of the AOP are converted into the XLM files and are uploaded for the further processing
- Next step after the uploading the files is to collect the data from both the files for gathering the data from the Xml files the xlm phrasing technique is used
- What is the purpose of activity diagrams which should be stored in respective associative array to uniquely define each and every activity and their purpose?

- Now we have to compare first associative array with second one to find the similarities between both the systems.
- After getting the result, the user gets to know about similarities and dissimilarities about the target and the developed system

VI. RESULTS AND DISCUSSION

In this section there is discussion about the results of proposed method of the class diagram of the AOP system. In this paper method is used for creating communication link between two or more activities within software. By using this method the relation between two or more clones can be determined easily. The efficiency of the technique is proved with the help of simulation. The graph given below depicts the comparisons between the proposed and the traditional algorithm. The proposed technique is considered to be efficient than the traditional approach.

Figure 2 shows a dialogue box. In this dialogue box, User selects two files which are to be compare. Both XML files will be compared for cloning. After selecting the files click on Extract data button. This will extract the data like attributes; functions, classes and aspects of both files.

![Figure 2 Browsing of files](image1)

In figure 3 a new window will open it contains the extracted data of both files. There are two columns in this window to present the data of both files separately. The text in green color mention the method functions of file, the text in black color represents the classes, text in purple defines the list of attributes of file and text in dark red color is used to represents the aspects that are contained in the file.

![Figure 3 Normalization of both input files](image2)
The window shown in figure 4, represents the results of XML parsing. Here parsing refers to the results that are obtained after comparing both of the xml files. In this the result is in tabular form contains 11 columns. These columns contain the value or ratio of cloned data w.r.t both files. As shown in figure first two columns contains the data of both files, second column list the number of total classes in both of the xml files. Third column and fourth column list the number and percentage of cloned classes respectively. Rest of the columns lists the number and percentage of clone attributes and operations.

Following figure 5 shows the comparison of traditional techniques and proposed technique. It shows the ratio of the cloned classes, cloned aspects and cloned operations with respect to both XML files.

VII. CONCLUSION AND FUTURE SCOPE

SDLC is Software development life cycle. It is a step wise process for developing software. It contains phases like requirement gathering, feasibility study, designing, coding, Implementation, testing and maintenance. Cloning is a problem which exists in coding phase. Cloning refers to the process of pasting the data or code which is occurring gain and again in the code. The cloned data leads to the problems or obstacles in maintenance phase. It is not easy to maintain the cloned data. Hence a technique is proposed in this paper which is quiet efficient in detecting the cloned data in early stage before the completion SDLC of AOP system. The efficiency of the proposed technique is proved in the results and discussions section by using graphs.

In future the proposed technique can be enhanced by adding some other parameters to detecting the cloning in system. This technique can be modified to make it more efficient and effective.

REFERENCES


