Asian Journal of Information Technology 15 (20): 4071-4076, 2016
ISSN: 1682-3915


# Automatic Test Case Reduction in Object Oriented System Using Clustering and Fuzzy Logic

[1]B. Geetha and [2]D. Jeya Mala
[1]Department of Computer Applications,
KLN College of Engineering, Pottapalayam, India
[2]Department of Computer Applications,
Thaiagarjar College of Engineering, Madurai, India

**Abstract:** Software testing plays a key role in software development life cycle. The organization of software testing helps to reduce the time and cost. Object dependencies create a major problem while doing unit testing in object oriented software system. This study presents a novel approach to reduces the number of test cases using clustering and fuzzy logic. It also solves the ordering of objects using reflexive technique in object oriented system. This approach consists of three phases. The first phase identifies the type of object using reflexive method. In this second phase we establish a priority in order to prevent the anomaly of testing order. In the third phase, we lessen the no of test cases in object oriented system.

**Key words:** Object oriented testing, independent and dependent classes, testing order, anamoly, priority

## INTRODUCTION

Software testing is an a significant activity of software development life cycle which ensures the quality of object oriented system. However, software testing in object oriented systems is little bit tricky as complexity transformed from functions and procedures as in traditional procedural systems to the interconnections among its polymorphism, inheritance, etc. A major dispute to the software developers remains how to reduce the cost while improving the quality of software testing in object oriented system.

**Major issues involved in object oriented testing:** The issues of testing object oriented systems are inheritance, encapsulation and polymorphism. The other issues of object oriented systems are decentralized code, encapsulation, testcase identification and raising exceptions.

**Decentralized code:** With traditional systems where all the functions are centralized, locating the source of a bugs do not require us to care outside the boundary of the program. The same usefulness that was in a solitary procedural project is broken out into numerous litter classes in an framework. it can be contended that little classes facilitate the procedure of finding the wellspring of a bug in light of the fact that little pieces of code are inspected. In any case, the reflection is that the code is no more brought together so finding the source of bud requires looking in numerous spots.

**Encapsulation:** Encapsulation is a technique which hides information where the interface of a business logic of a program unit are separated syntactically. This enables the programmer to hide design decisions within the logic to shrink the possible interdependencies with other components by means of interface. Therefore, if there is a change in the super class then it is mandatory to retest all its subclasses because they inherit its methods. Due to data hiding there is no visibility of the insight of objects. This data abstraction makes it is difficult to check what is inside an object during testing.

**Inheritance:** Legacy is one of the important qualities of article situated programming. It implies the characters for a class that are inherited by its subclasses, unless it is generally expressed. So, it gives the reusability. Along these lines it is unequivocally in light of legacy that we find issues emerging regarding testing. It too confounds between class testing since different classes are coupled through legacy.

**Literature review:** Many researches have proposed a wide range of solutions to reduce the number of test cases in object oriented system.

**Corresponding Author:** B. Geetha, Department of Computer Applications, KLN College of Engineering, Pottapalayam, India

**Soft computing approaches:** Bipin Pandey and Rituraj Jain applied soft computing in different types of software testing. By using the techniques for software testing, the result has been enhanced and the overall performance of testing was improved.

**Testing least dependent class:** Dadhich and Sehgal (2012) preferred the test order from least dependent class to most dependent class. The result showed that most dependent class to least dependent class saves the time and efforts.

**Random test case generation:** Ingle and Mahamune explained the different kinds of techniques for generating test cases to fulfill test case analysis standard. The path oriented test recognizes path for which test case has to be created, still the path might be located infeasible and the test data generator enhances the input that is pass through the path.

**UML sequence diagram:** Shanthi1 and Mohan Kumar suggested the test case generation by means of UML Sequence diagram using Genetic Algorithm where test cases are optimized. This approach is important to identify location of a fault in the implementation, thus reducing testing effort. Moreover, this method for test case generation suggests the developers to improve the design quality, reduce software development time and find faults in the implementation early stage of software development (Ganatra *et al*., 2011). Figure 1 shows the proposed system.

**Proposed system**

**Backpropogation algorithm:** Chandra *et al.* (2015) concludes that the resilient back-propagation algorithm is fastest and performs best in both efficiency and accuracy measures. Conjugate Gradient algorithm gives best sensitivity, whereas Levenberg-Marquardt algorithm with Bayesian Regularization gives best specificity but it is the slowest when training time is considered. Classification performance of each algorithm improves with increased input factors. If the number of factors are high, increased number of neurons improves the performance of algorithm.The training time increases for all algorithms when either number of inputs are increased or number of neurons in hidden layer are increased (Solanki and Jethva, 2013).

**Proposed system:** In our proposed system, we bring the object under two major classification. Dependent object independent object. We classify the object using reflexive technique. We give highest priority to dependent object while doing software testing. Because, dependent object covers more number of data members. Then, second priority moves to independent objects. We test both objects using clustering and fuzzy logic. After classifying the object oriented testing, we proposed a new technique for reducing the number of test cases without compromising the quality of the software. The backbone of our proposed technique is reflection,clustering and fuzzy logic. In this study, we discuss the strength of reflection, clustering and fuzzy logic which brings an optimal solution to object oriented testing. We discuss the components of our approach one by one.

**Reflection:** Reflection is one of advanced features of object oriented languages. This technique has the ability to find the type of objects at run time. It supports various methods which helpful to find out whether an object is a dependent object or independent object. The proposed system uses the reflection technique to classify whether the object is an a independent object or dependent object. This can be used by testers and developers who have a strong grasp of the fundamentals of the object oriented language. Since, it is a sensitive technology, it requires a run time permission of the security manager.

**Clustering:** Clustering plays an important role in a broad range of applications. It is a data modeling technique which can partition a collection of data into a set of meaningful subgroups. Our proposed system uses the clustering techniques for grouping the data item of both dependent and independent objects. It groups the data items based on data type. For example the data items of an object may have different data types like integer, string, short, long, doube, etc. Based on the data types the clustering techniques classifies each data item of an object and stores it in the meaningful groups. That is the data items belong to integer are grouped together. Similar to that the data items belongs to string or integer group together. Likewise,each group is created for a particular datatype. These kind of groups help us to test items very easily. Clustering technique gives a way to select a test data item from each group and helps to reduce the number of testcases. Thus, it save time and cost system resources.

**Testing the object using fuzzy logic:** One of the cost effective software testing methodology is fuzzy logic. Many researchers have utilized the effectiveness of fuzzy

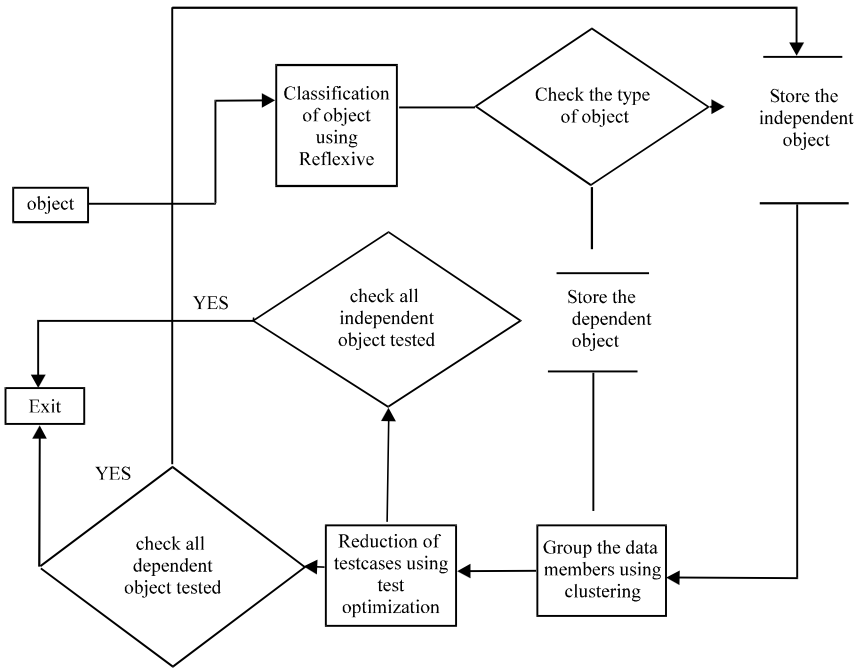


Fig. 1: The proposed system

logic to test the object based software testing. In our proposed approach we use rule based fuzzy logic (Ganatra *et al*., 2011). To test all kinds of data types of data members of an object, we choose only one data member from each cluster. We apply if then rule for testing the each selected data member. For numeric related data types, we generate test cases related to positive number, negative number, zero and boundary value of positive number and negative number. If the data type of data member belong to string, we generate three types of test cases valid string invalid string null value. For each valid value, we give 0.1 credit point. If total value of credit point is equal to the expected credit point, then, it means that all the data types of data members of objects is examined carefully with minimum number of test cases with suitable credit points (Table 1).

## RESULTS AND DISCUSSION

Our experimental setup consists of an application which is implemented in java bean. These applications are executed in the Netbeans IDE. We utilized the power of java for classification of objects, finding data types, storing a cluster of object members, etc. We evaluated our proposed system using student developed object oriented system namely, student result evaluation, library management system (Algorithm 1 and Fig. 2 and 3).

Table 1: The credit point allocated to each type of test cases

| Data type | 0-10 | -1to-1 | 0 | Min(a-z,length of 5 charactes) | Min(a-z) |
|---|---|---|---|---|---|
| Int | 0.1 | 0.1 | 0.1 | 0 | 0 |
| Short | 0.1 | 0.1 | 0.1 | 0 | 0 |
| Long | 0.1 | 0.1 | 0.1 | 0 | 0 |
| Double | 0.1 | 0.1 | 0.1 | 0 | 0 |
| String | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

**Alogrithm 1: Implementation code for classification of objects:**

```
import java.io.*;
class sample{
public void method()
{
System.out.println("hai");
}}
class simple
{
public void meth(){
System.out.println("hoi");
}}
class sumple extends simple
{
public void met(){
System.out.println("welcome");
}}
class MainClass{
public static void main(String ar[])throws IOException
{
System.out.println("Enter the choice");
DataInputStream dis=new DataInputStream(System.in);
int x=Integer.parseInt(dis.readLine());
String name=" ";
Class c;
if(x==1)
{sumple s=new sumple();
c=s.getClass();
c=c.getSuperclass();
```

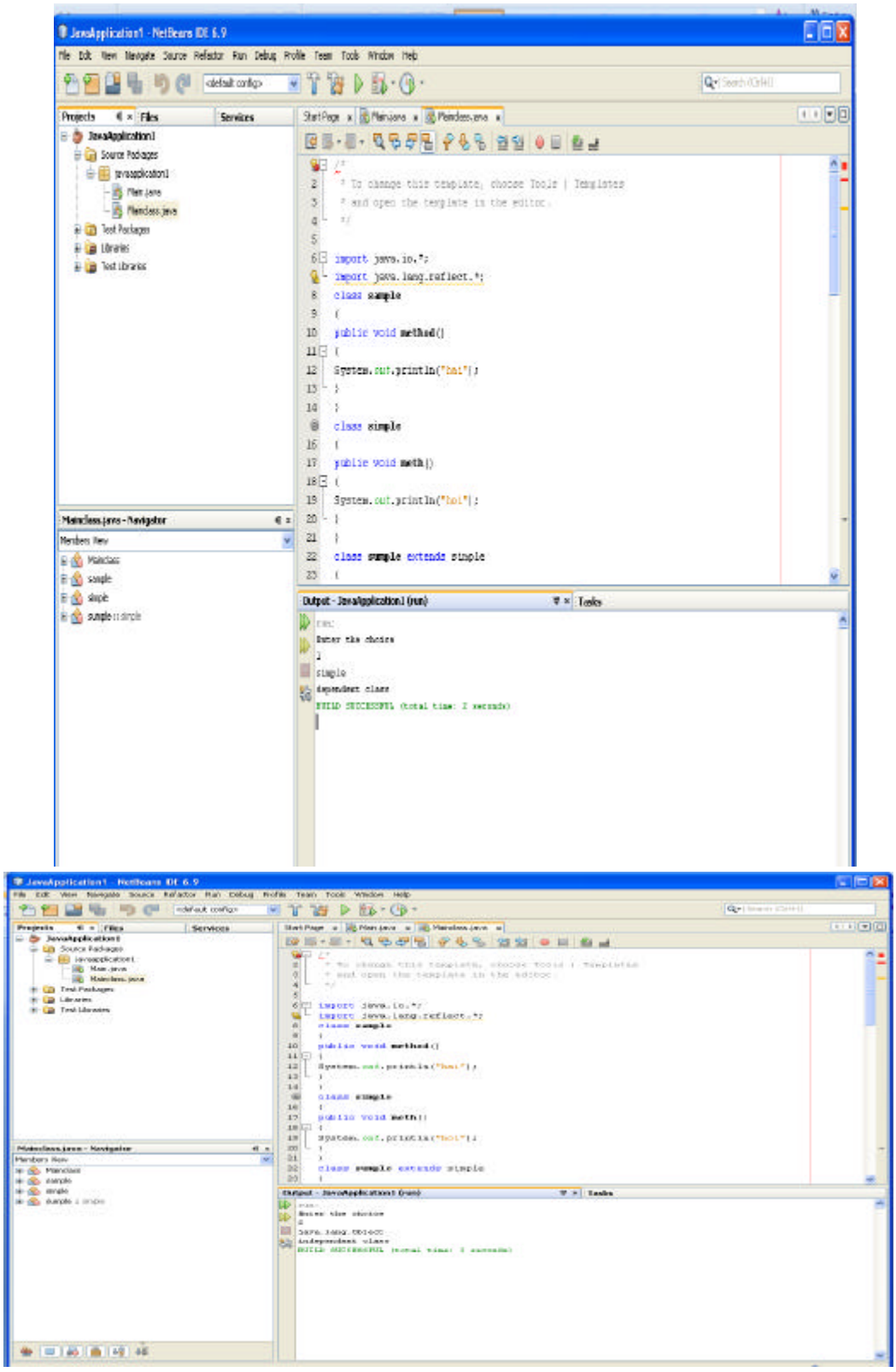


Fig. 2: Experimental setup of application

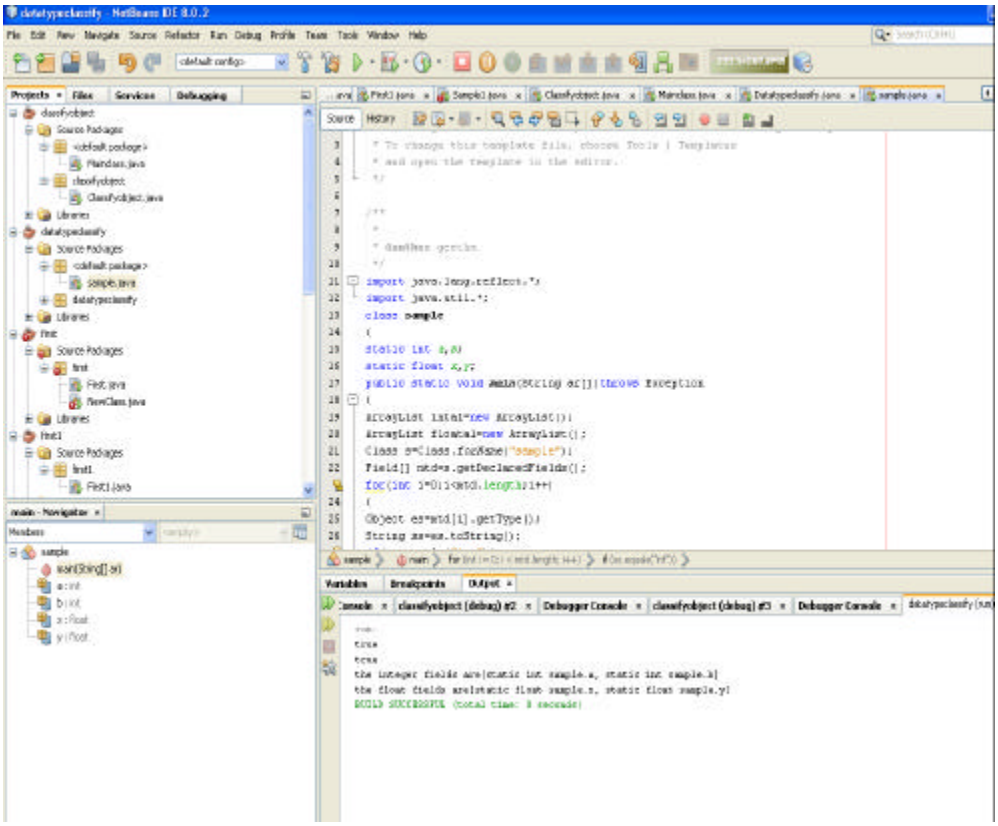

Fig. 3: Experimental setup of an application in Java bean

```
name=c.getName();
}else{
sample s=new sample();
c=s.getClass();
c=c.getSuperclass();
name=c.getName(); }
if(name.equals("Object")){
System.out.println("independent class");}
else
System.out.println("dependent class");}}
```

**Algorithm 2: Code for reducing the test case**

```
Import java.lang.reflect.*;
Import java.util.*;
Class sample
{
static int a,b;
Static float x,y;
Public static void main(String ar[]) throws Exception
{
ArrayList intal=new ArrayList();
ArrayList floatll=new ArrayList();
Class s=class.forName("sample");
Field[] mtd=s.getDeclaredFields();
for(int i=0;i<mtd.length;i++)
{
Object es=mtd[i].getType();
String ss=es.toString();
If(ss.equals("int"))
{
System.out.println(intall.add(mtd[i]));
}
else if(ss.equals("float"))
{
Flaotal.add(mtd[i]);
}
}
System.out.println("the integer fields are"+intal);
System.out.println("the flaot fields are "+floatal);} }
```

## CONCLUSION

In this study, we presented an approach that is classify the object using reflexive technique and store it. Then these objects are grouped using clustering technique. Clustering technique gives a way to select a test data item from each group and helps to reduce the number of testcases. Thus, it save time and cost system resources. The scope of our proposed system is fully automated tool to be developed. The tool will facilitates all levels of object oriented software testing and save time and reduce the cost of the software.

## REFERENCES

Chandra, A., M. Suaib and D. Beg, 2015. Web spam classification using supervised artificial neural network algorithms. Adv. Comput. Intell. Intl. J., Vol.2,

Dadhich, R. and S. Sehgal, 2012. Analyzing the efficient test order for integration testing. Adv. Comput., 3: 35-42.

Ganatra, A., Y.P. Kosta, G. Panchal and C. Gajjar, 2011. Initial classification through back propagation in a neural network following optimization through GA to evaluate the fitness of an algorithm. Intl. J. Comput. Sci. Inf. Technol., 3: 98-116.

Solanki, S. and H.B. Jethva, 2013. Modified back propagation algorithm of feed forward networks. Intl. J. Innovative Technol. Exploring Eng., 2: 131-134.