ABSTRACT

In this thesis, an innovative model “INHERITANCE RELATED BUGS TESTING (IRBT)” is presented for testing of inheritance related bugs in object oriented programs.

Software Engineering is a strategy for producing quality software. The software development life cycle is the sequence of different activities that take place during development. Any software development consists of five phases namely Requirements Analysis, Design, Coding, Testing and Maintenance. Testing is execution of the software with data to ensure that the software works correctly.

Traditional Procedure Oriented Program testing and Object Oriented Program testing are different from each other. Object Oriented Program is different because of its special features such as Polymorphism, Inheritance, Data Abstraction and Encapsulation. The inheritance related bugs will not be found by traditional object oriented language compilers or editors. This innovative model will be useful for software development team which is especially working under the Java, Objective-C and Smalltalk platforms. In object oriented life cycle inheritance related bugs may occurs in two phases. They are design phase inheritance related bugs and implementation phase inheritance related bugs. This work identifies implementation of inheritance related bugs namely Naked Access, Inadvertent Binding, Spaghetti Bug and Naughty Children.
This research work is mainly based on Source Code Analysis, Whole Program Analysis and Fragment Analysis testing, Analysis of inheritance property of object oriented programs and testing inheritance related bugs in object oriented programs.

Source code analysis gives information about the source code written using object oriented language. Most existing program analysis provides syntactic structure of the program. With the data flow analysis, semantic structure analysis is possible. In this thesis, object oriented program source code is analyzed. This is implemented by Source Code Analyzer. It is a static testing tool, useful for analyzing Object Oriented Programs. Input is an object oriented program and the outputs are number of lines, number of semicolons, number of function braces and data type used. The existence of polymorphism is also identified in source code analysis.

Whole program analysis and fragment analysis testing will be useful to find out the object oriented properties of a program such as lines of code, number of classes, methods, objects in a class, comment lines and software metrics parameters such as depth of inheritance, response method for call and function points. Whole program is analyzed for classes, methods, objects and it is used for further analysis. Fragment is the one which divides a program into small pieces. Fragment analysis information are class name, methods, lines of code and objects in a particular fragment. In this work whole Java program is the input and every line of codes is analyzed by using compiler techniques. Outputs are classes, methods in a class, attributes in a method and object. In fragment analysis, a program is divided into pieces and analyzed for
whether inheritance property used or not is found in a class. Knowledge base stores information about program analysis and in future it can be used for software metrics.

Data flow testing is an appropriate testing method. This testing analyzes the structure of the software and gives the flow of property. This work is designed to analysis the set of classes and packages and output hierarchies of the classes, methods and attributes detected by using data flow testing. Class hierarchy analysis is used to find relationship between the classes. This testing is done by three major analysis such as Knowledge base preparation, Code analysis and Level analysis. Input of the tool is Object Oriented Program. Output of the tool is attribute description in classes, methods description and classes with levels list. It shows the hierarchy of the classes. Sunsoft Java Software is used for data flow testing.

Some of the property bugs are located in the program. This work is designed to detect the bugs with reference to the multilevel inheritance property. Implementation phase inheritance related bugs such as Naked Access, Inadvertent Binding, Spaghetti Bug and Naughty Children are detected by this module.

An innovative model IRBT has been developed for the first time to test inheritance related bugs in object oriented programming. It is not only useful for finding inheritance related bugs but also for testing and analysis of source code and inheritance property. The result of this can be efficiently used for finding software metrics.