CHAPTER 7

CONCLUSION AND FUTURE WORK

The review of literature shows that there is no model available for testing inheritance related bugs in object oriented programs. Hence an innovative model has been proposed in the name of ‘IRBT’ model in this thesis.

The model has been developed to analyze the source code of object oriented programs. The whole program analysis has been executed for exemplifying object oriented program parameters such as number of classes, lines of code, comment lines, methods, attributes and objects. The fragment analysis has been implemented to fetch the information about availability of inheritance and number of methods in the classes. The classes, methods and attributes have been analyzed for the object oriented programming. Classes with inheritance hierarchy and the relationship between them have been identified. Implementation phase inheritance related bugs have been detected such as Naked Access, Inadvertent Binding, Spaghetti Bug and Naughty Children.

The major steps in the IRBT model are data flow testing for source code analysis, whole program analysis and fragment analysis testing, analysis of inheritance property of object oriented programs and testing for bugs due to inheritance in Object Oriented Programs.

Data flow testing of source code gives information about the source code especially written by using object oriented language. Source code is
completely analyzed while most existing program analysis provides syntactic structure of the program.

The whole program analysis and fragment analysis testing are used in finding software metrics parameters and these parameters are used for upgrading the version. First, the whole program is analyzed by using interprocedural data flow analysis. Fragment analysis is done by dividing a program into fragments, for example, into classes. Every line of codes is analyzed by using compiler techniques and lines of code, attributes and methods are found. All gathered parameters are stored in knowledge base. In this work, Java based programs were taken for testing.

Data flow testing of inheritance property analysis identifies inheritance related information. Sunsoft Java packages have been analyzed by using data flow testing class hierarchy analysis. The inheritance related information such as packages, classes in each package, number of lines after removal of document line for analysis, methods in a class and classes with levels have been tested. The inheritance data flow testing tool has been developed for this work.

The inheritance related bugs arise either at design phase or at implementation phase. The IRBT model proposed in this thesis aims at identifying inheritance related bugs only with regard to implementation phase. The Inheritance related bugs in Object Oriented Programs such as Naked Access, Spaghetti inheritance bugs, Naught Children, Inadvertent Binding and Name Conflicts have been found using IRBT model. These are the inheritance related bugs with regard to the implementation phase. But, there are other bugs such as ‘Square Peg in a Round Hole’, ‘Fat Interface’, ‘Weird Hierarchies’ and ‘Worm Holes’ that may be raised when the proper design is not made. The implemented model is useful for finding inheritance bugs during object-oriented development and all software engineering tasks.
Hence, it can be clearly understood that the proposed IRBT model is an innovative model used in testing the Object Oriented Programs for identifying inheritance related bugs which is not found to be performed by any other existing testing methods.

In the future, data flow testing can be extended for identifying design based inheritance related bugs also. This can be useful for program development, further changes in programs and other software engineering tasks. The current system is designed to find out the inheritance related bugs from the Java products only. In future the system can be enhanced to detect multiple inheritance hierarchy. Tools can be developed for other object oriented programs like Smalltalk and Objective-C. The current bug detection scheme can be integrated to a compiler to detect the hidden errors during the compile time. The knowledge base preparation output can be used to find Object Oriented metric for improving the quality of product. The current system is developed as static analysis tool to test the source code. The same concept can be implemented under the dynamic testing mechanism to analyze the product using the byte code to analyze the third party programs.

This thesis, presents a framework using which data flow information can be collected for the analysis of a program. There are several key features like Polymorphism, Overloading and Exception Handling etc., that require further research related to this work. Further models may be developed for bug detection, languages like Smalltalk, C++ and Objective-C.