6.1 Review of Objectives

6.2 Findings and Significance

6.3 Limitations of Current Work and Future Directions

6.1 REVIEW OF OBJECTIVES

In the beginning of present research work all the objectives were well thought and decided. The primary objective of this research work was to study and design object oriented software metrics. Extensive literature survey was carried out to understand existing research work in the area of object oriented measurement. After the review of literature, three major areas in object oriented metrics were targeted for further research. These targeted areas include Complexity, Fault-Proneness and Reusability metrics of classes. Six specific objectives were achieved as a part of this research work.

First objective was achieved by performing thorough study of existing object oriented metrics by focusing on taxonomy, definition, purpose, approach and validations. Second objective was achieved by first proposing new complexity model for classes which includes two new metrics - Total Method Call Complexity (TMCC) and Total Data Call Complexity (TDCC), and then comparing proposed model with four existing metrics (WMC, RFC, DIT and CBO). Third objective was fulfilled by identifying factors that can affect the fault-proneness of classes and suggesting existing metrics for measuring fault-proneness factors. Fourth objective was accomplished by suggesting machine learning based techniques for categorization of faulty and non-faulty classes. Suggested techniques were also applied on CK metrics for selecting best suitable metric(s) for categorization of classes. Fifth objective was achieved by first giving taxonomy for inheritance hierarchy based metrics for categorization of reuse and reusability metrics and then proposing five new metrics for evaluating reuse and reusability from inheritance.
hierarchy. New metrics includes Breadth of Inheritance Tree (BIT), Method Reuse Per Inheritance Relation (MRPIR), Attribute Reuse Per Inheritance Relation (ARPIR), Generality of Class (GC) and Reuse Probability (RP). Sixth objective was accomplished by correlating depth and breadth of inheritance hierarchy with reuse and complexity using five design oriented metrics - DIT, NOC, MIF, AIF and RFC. Four hypotheses (H1-H4) were derived before collecting the data out of which H1, H2 and H4 was accepted and H3 was rejected.

6.2 FINDINGS AND SIGNIFICANCE

This research work conducts study and survey of existing object oriented metrics which provides better understanding of existing object oriented metrics and its validations. It helps in setting objectives for conducting the research work. Further whole research work is divided into three parts - measuring complexity, fault-proneness and inheritance hierarchy of classes. Specific findings and significance of each part are given as follows:

First part (Chapter-3) proposes a new complexity model which computes Class Complexity (CC) in terms of Method Complexity (MC) and MC is further computed as a sum of Control Flow Complexity (CFC), Total Method Call Complexity (TMCC) and Total Data Call Complexity (TDCC). This model is useful for computing complexity by integrating complexity due to control flow, inheritance and polymorphism. Comparison of proposed model with four CK metrics (WMC, CBO, RFC and DIT) shows that CC is highly positively correlated with WMC and RFC as compared to CBO. It is believed that CC is also highly positively correlated with DIT due to involvement of many classes in method/data calls. Proposed model also shows that number of method/data calls, type of method/data calls, number of classes involved in method/data calls and control flow of methods highly influences the complexity of class. Proposed model is also useful for evaluating complexity of classes which shows the scope of improvement in the classes. This model helps in developing easily understandable classes which reduces the maintenance cost of software.

Second part (Chapter-4) describes fault-proneness of the class as a function of several factors such as method complexity, coupling, cohesion, inheritance, polymorphism and...
information hiding. Eight existing metrics are suggested that can be used to measure fault-proneness factors. This part also suggests measures used in two machine learning algorithms namely ID3 and CART for selecting best suitable metric(s) from given set of metrics for classification of faulty and non-faulty classes. The given techniques are applied on CK metrics (WMC, CBO, RFC and LCOM) using two case studies of 24 classes. Result shows that WMC is best and CBO is good metric whereas RFC and LCOM are unreliable metrics for classification of faulty and non-faulty classes. Study presented in this part is very useful for detecting fault-prone classes. On the basis of fault-proneness of classes, fault management activities can be planned for optimum utilization of efforts.

Third part (Chapter-5) suggests taxonomy for better understanding of inheritance hierarchy based metrics and proposes five new metrics for assessing reuse and reusability. Newly proposed metrics are Breadth of Inheritance Tree (BIT), Method Reuse Per Inheritance Relation (MRPIR), Attribute Reuse Per Inheritance Relation (ARPIR), Generality of Class (GC) and Reuse Probability (RP). Analysis of given metrics on a Case Study validates that these metrics are capable for assessing reuse and reusability. These metrics are helpful for increasing reuse and reusability in classes and comparing two alternative inheritance hierarchies of same problem. This part also finds out the relationship between dimensions of inheritance hierarchy (depth and breadth) with reuse and complexity. Four hypotheses (H1-H4) are tested in this regard. On the basis of results obtained hypothesis H3 is rejected and all other hypotheses are accepted. This study is helpful for increasing reuse and reducing complexity from inheritance hierarchy at design time.

6.3 LIMITATIONS OF CURRENT WORK AND FUTURE DIRECTIONS

The limitation of current work is that case studies used are small and have limited applicability. The current work shows that proposed metrics works well for simple applications. Results obtained are very interesting and encouraging; however metrics validity should be tested out on non-trivial industrial projects also. There is always need
of enhancement in the area of object oriented measurement. Number of specific potential enhancements to current work can be as follows:

Proposed complexity model is compared with only four metrics WMC, RFC, DIT and CBO. Given model can be compared with other complexity models/metrics available in literature. Measurement of fault-proneness considers only design-oriented factors however other non design-oriented factors can be considered for further research work. Fault-proneness of a class is defined as a function of member complexity, coupling, cohesion, inheritance, polymorphism and information hiding. But weight of these factors are not computed which could be the agenda for further research. Selection of best metric from given set of metrics for classification of faulty and non-faulty classes is limited to evaluation of individual metric; however combination of two or more metrics may be analyzed using similar study. Same study can be replicated with more algorithms of classification with emerging approaches such as neural network and fuzzy logic. Comparative analysis can be conducted between the classification techniques given in this research work and other techniques available in literature. Study of inheritance hierarchy based metrics is limited to only one type of reuse i.e. inheritance. Similar study can be conducted to include other methods of reuse also.