CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 INTRODUCTION

Today each and every field of human life such as business, education, travelling, healthcare, entertainment and governance to name a few are dependent on software systems and are affected by the software in one form or another. But unfortunately the software development is an error prone process and testing is used to find and remove these errors. A lot of money and other resources have been put by various agencies such as military, government, banking, insurance, universities, and healthcare into software. Many critical applications like pilot training simulators, weather forecasting, medical imaging, expert systems, tele-communications, image processing and so on are dependent of software. In these type of applications the software needs to be highly reliable and error free. Even a small bug in the software may cause havoc. It has been an established fact that many disasters were caused by faulty software and could have been avoided by thorough testing. To ensure the reliability of the software, it must be tested rigorously, so testing is a field of interest for the researchers working in the area of software engineering.

Object orientation is often heralded as a silver bullet to solve the problems in the field of software engineering. But as such, there are no silver bullets. Object oriented development needs a somewhat different approach than traditional procedural development. The need is generally felt to have some metrics which may guide the development team in the project development. Irrespective of the development methodology used, the software need to satisfy the goals for which it was designed. Testing is one such activity which is used to validate software. A lot of metrics have been proposed by the researchers to help the software professionals to predict the degree of testability of the product being developed. The modern day approach is to integrate the testability with all the phases of software development life cycle. The efforts, cost and resources required for testing may be reduced considerably if there is some mechanism to know which metrics should be used to achieve high degree of
Conclusion and Future Scope

testability in the product being developed. Some of traditional metrics or their modifications can be used in object oriented paradigm. However, for the quantitative measurement of object oriented features like inheritance, polymorphism, coupling, cohesion and message-passing researchers have proposed new sets of metrics.

The objectives of the present research work were:

1. To find the possibility of introducing the testability at the design level because the decision to change the design in order to improve testability after coding is very expensive and error-prone.

2. To explore the idea of evaluating testability in the development process as early as possible to reduce overall cost of testing object-oriented software.

3. To develop various testability measures and metrics to evaluate the testability of object oriented software.

4. To find new attributes and metrics that, integrated with the existing work will provide a comprehensive picture of all aspects of testability in the development process.

5. To rank the software metrics using expert opinion aggregation.

7.2 RESULTS AND DISCUSSION

The above objectives have been achieved in the present research work. New metrics namely average temporal complexity (ATC) and relation based testability metric (RTM) have been proposed.

ATC can be computed for any object oriented system, a Sugeno Fuzzy logic based system with three inputs namely depth of inheritance tree, number of methods in the class and number of overridden methods in the class. The proposed new metric has been validated theoretically by using well established Weyuker’s properties criteria and practically by applying it on three sample projects. The simulation results shows that the metric ATC gives a clear indication about the runtime behavior of the system under consideration. ATC will definitely help the future software professionals in the overall development process in general and in assessing the complexity of the system at runtime in particular. The metric is an indicator of external behavior of the software.
A new relation based testability metric (RTM) has been proposed to measure the complexity of the object oriented system. A new algorithmic approach has been presented to measure the complexity of the system considering its procedural and object oriented features simultaneously. The results obtained by applying this approach on the experimental projects have shown that the new approach has an edge over the existing approaches and is helpful for the developers. RTM provides an early indication of the complexity of the system and the developers may improve their design. As the metric can be used in early stages, it will help to evaluate the complexity of the system at an early stage. By using this approach better products may be designed in terms of understandability, maintainability and testability.

Many metrics have been proposed for measuring various aspects of the object oriented software. These metrics are based upon inheritance, coupling, cohesion, polymorphism, complexity etc. None of the researchers till date have thrown some light on relative importance of these metrics in a particular case. All the metrics seems to be equally important in absence of any particular method for their ranking. The proper ranking of these metrics may be very helpful for the software professionals working on a particular project. A distance based approach has been applied to rank the metrics for object oriented software in general and testability metrics in particular. For this study the expert opinions of about forty software professionals working in different IT firms/ educational institutions in National Capital Region India have been collected and analysed and a comprehensive ranking model using MATLAB has been developed for these metrics based upon the criteria and subcriteria.

It has been observed that no single metric is best suitable for each criterion, taken individually. The ranking of metrics changes with respect to criterion. The proposed deterministic quantitative model based distance based approximation method is capable for analysis, evaluation, selection and ranking of metrics for all criteria considered collectively in an integrated manner. The results are presented in terms of a merit value which has been used for ranking of the metrics. The ranking of the metrics based on individual criteria and overall ranking based on all criteria taken together is shown in the figure 7.1 below:
Conclusion and Future Scope

7.3 FUTURE SCOPE

1. For computation of ATC we have considered three input parameters and unity weight for all the inputs of the Fuzzy Inference System. The number of inputs may be increased to accommodate more object-oriented features of the software.

2. For computation of the relation based testability metric (RTM) one procedural metric and six object oriented metrics has been used. The idea can be furthered to include more procedural and more object oriented features to get more new metrics for assessing the complexity of the system under consideration.

3. The software metrics have been ranked using a well structured approach based on expert opinion aggregation and distance based approximation. Some other technique like matrix method etc. may also be applied for ranking of metrics.

4. In future, this method can be generalized to all categories of metrics used in software organizations to facilitate the proper ranking of the metrics.
5. The selection of metrics depends on number of criteria / sub-criteria and it is possible that one metric satisfies some of the selected criteria, a software professional is looking into, but not remaining others and the other metric may satisfy the other ones not the first ones. Therefore, multi objective decision making methods can be developed to optimize the selection process based on critical criteria.