CHAPTER 1

Introduction to This Work

Abstract

Since all software developers' main concern is to reduce cost and effort, they spent a lot of time in design phase to guarantee the design quality by predicting and correct potential design problems earlier before implementation time.

In this project, I'll introduce DEMOS, Design Measurement Tool for Object-Oriented Software. *This tool goal is to measure object oriented design aspects based on structural properties of UML design elements*.

The tool will support broad design measurement based on static structure and dynamic behaviors. This will provide developers with early quality feedback. Also, this tool will allow developers to make a comparative study of object-oriented features.

In this tool, I plan to investigate which are relevant metrics for dealing not only with the *static structures* of models, but also with its *behaviors*. For such metrics, possibilities of metrics formalization (and even creation) based on other UML diagrams can be easily done.

Then, I'll use this tool for comparative study. Definitely, coupling, cohesion, inheritance, encapsulation, and other object-oriented features affect effort and time spent on implementing a given design.

However, what is the nature of this effect? For example, does using inheritance reduce implementation time or effort or increase it? Using inheritance allows reuse, however, it increases the cognitive complexity needed to understand a class deep in the inheritance tree.

Such a study would result in a sort of thresholds or optimal ranges for metrics such as depth of inheritance tree and number of sub-classes.

Introduction

Nowadays, object oriented programming became the most popular developing method. Many programming languages support the object oriented paradigms, such as: c++ and java.

However, there is hidden factor play the main role in determine the budget, cost and development plan, which is the “Design”. In software engineering, design phase is known as the most consuming factor of the project effort, because bad design will lead to implementation problems and rise the need to re assess the design again and more again.

Now, the primary question here is how to evaluate the design before going to the next step? So software engineers come up with a “Design Metrics”, which can examine and evaluate a given design in order to detect bad programming behavior.

In this project, we design and implement a tool that automates the metrics measurement. It gives the user the ability to assess the design static and dynamic behaviors. This tool gives the user the ability to add design project UML diagrams and design metrics, execute them and return the result in different views.

There is post extraction method, which provide user with different design detection and assessment criteria.

The main distinguish factor in this tool is that it includes the dynamic UML diagrams such as state and activity and it gives the user to evaluate the given design.

DEMOS is so flexible and scalable due to its implementation which is based on Xquery. This allows many attractive features and promises us with good future improvements.