Proposal for MSE Capstone Project

Project Title:

Student Name:

Project Sponsor:

Faculty Advisor:

Date of submission:

A Metrics Tool for Evaluating Object-Oriented Programs

Objective
The aim of this project is to develop a metrics tool to evaluate object-oriented programs written in one of the four languages – Java, C++, Smalltalk and Eiffel.

Background
Quality assurance plays a major role in software development. One way to evaluate the quality of a program is to use a quantified approach using metrics. Using this approach, the quality of a program can be evaluated by extracting primitive metric values from the program such as the number of statements, number of control paths, number of shared variables and so on, and then applying these values to some predefined metric formulas. The result of evaluating the formulas will be compared against some benchmark values for the chosen application. As an example, consider a metrics formula that determines the coupling between two objects in an object-oriented program. This formula may be defined as

```
Coupling factor = f (number of outside methods called by this class) *
                  g (number of methods within this class called by methods
                      from other classes) *
                  h (number of public attributes of this class)
```

The functions “f”, “g” and “h” must have been defined before. The theoretical evaluation of this metrics formula may assert that the result of evaluating this formula for every class in a program must be less than or equal to 0.5.

Experience reports indicate that metrics tools are very useful in evaluating and improving object-oriented programs and also help in establishing organizational rules on developing similar programs. Since this approach requires vast computing needs, the support of a metrics tool is evident. Due to the constantly evolving nature of object-oriented program qualities and advancements in object-oriented program metrics, the metrics tools are expected to be easily modifiable and can be adapted to changes in both formulas and language constraints.

Current Project
The current project focuses on developing a metrics tool to evaluate object-oriented programs written in one of the four languages – Java, C++, Smalltalk and Eiffel. The tool must have the following characteristics:

- The user must be able to select one of the four languages for evaluation.
- The user must be able to select a program for evaluation through a file browser.
- The user must be able to load a set of metrics formulas from a file.
- The user must be able to select a subset of formulas for evaluation at any one time.

- The user must be able to evaluate one or more formulas at any one time.
- The tool must evaluate and preserve the results in a temporary file so that the user can decide to view the results or save the results. In case the user decides to save the results, the tool must be able to save the results in different formats (see the next item) selected by the user.
- The user must be able to view the results in different formats such as textual, bar charts, pie charts and curves.
- The user must be able to view each step of evaluation or only the final result.
- The tool must provide options to enter metrics formulas dynamically. It must also provide options to save these formulas.

Challenges

The following are some of the challenges in this project:

- Object-oriented concepts are implemented differently in different languages. This poses a great challenge in accommodating one set of metrics for object-oriented concepts when the tool can be used for programs written in four different languages.
- Loading and saving user-defined metrics equations poses some challenge especially when it involves a lot of mathematical equations. The format of display and the representation of these equations must be worked out.
- Testing this product poses another big challenge as with any other metrics tool because of the variations in metrics formulas.

Project Schedule

The following schedule is proposed by the student, and is agreed by the project sponsor:

Phase	From	To	Credits
Study the problem; literature survey	Jan 01, 2002	Feb 28, 2002	1
Developing Requirements Document And problem analysis	Mar 01, 2002	Apr 30, 2002	3
Developing Design Document Includes User Interface Design	Jun 01, 2002	Aug 31, 2002	3
Implementation and Test	Sep 01, 2002	Nov 30, 2002	3
Demonstration and Project Report	Dec 01, 2002	Dec 31, 2002	2
		Total	12

The schedule does not include the time for the oral examination.
The student will be on vacation between May 01, 2002 and May 31, 2002.

Resources

The student will use the computing facilities in the university and also his/her personal computer to complete the project. The project sponsor will provide data for testing.