A UML Approximation of a Subset of the CK Metrics and Their Ability to Predict Faulty Classes

CAMARGO CRUZ Ana Erika
School of Information Science,
Japan Advanced Institute of Science and Technology

September 2011

Abstract

Design-complexity metrics, while measured from the code, have shown to be good predictors of fault-prone object-oriented programs and related to several other managerial factor such as productivity, re-work effort for reusing classes and design effort, and maintenance effort. Some of the most often used metrics are the Chidamber and Kemerer metrics (CK). Because earlier assessments of such managerial factors are desirable, prior to the code implementation, our research mainly concerns two topics. The first one concerns to the how can we approximate the code CK metrics using UML diagrams; and the second one concerns the use of such UML approximations to predict faulty object-oriented classes.

First, we define our UML metrics, approximations of the Weighted Methods per Class (WMC), Response For Class (RFC) and Coupling Between Objects (CBO) CK code metrics using UML communication diagrams. Second, we evaluate our UML metrics as approximations to their corresponding code metrics. Third, in order to improve the approximations of our UML metrics, we study the application of two different data normalization techniques, and select the best one to be used in our experiments. Finally, because code CK metrics have shown repetitively their ability to predict faulty code in several previous works, we evaluate our UML CK metrics as predictors of faulty code. In order to do so, we first construct three prediction models using logistic regression with the source code of a package of an open source software project (Mylyn from Eclipse), and we test them with several other of its packages. Then, we applied these models to three different small-size software projects, using, on the one hand, their UML metrics, and on the other hand, their corresponding code metrics for comparison.

The results of our empirical study lead us to conclude that the proposed UML RFC and UML CBO metrics can predict fault-proneness of code almost with the same accuracy as their respective code metrics do. The elimination of outliers and the normalization procedure used were of great utility, not only for enabling our UML metrics to predict fault-proneness of code using a code-based prediction model but also for improving the prediction results of our models across different software packages and projects. As for the WMC metrics, both the proposed UML and its respective code metric showed a poor fault-proneness prediction ability.

Our plans for future work mainly concern the exploration of other areas of research in which our UML metrics can be applied and as for the topic of fault prediction the following subjects for further study have been considered: data normalization and other pre-processing techniques, the study of other metrics to be included in our prediction models (such metrics should be easily obtainable before the implementation of the system and different to design-complexity metrics), and other methodologies to predict fault-proneness of code (different to logistic regression).

Key Words: **CK metrics, UML metrics, design-complexity metrics, fault prediction, logistic regression**