Software Fault Prediction with Object-Oriented Metrics Based Artificial Immune Recognition System

Cagatay Catal[1] and Banu Diri[2]

[1] TUBITAK-Marmara Research Center, Information Tech. Ins., 41470 Turkey
cagatay.catal@bte.mam.gov.tr
[2] Yildiz Technical University, Computer Engineering Dept., 34349 Turkey
banu@ce.yildiz.edu.tr

Abstract. Software testing is a time-consuming and expensive process. Software fault prediction models are used to identify fault-prone classes automatically before system testing. These models can reduce the testing duration, project risks, resource and infrastructure costs. In this study, we propose a novel fault prediction model to improve the testing process. Chidamber-Kemerer Object-Oriented metrics and method-level metrics such as Halstead and McCabe are used as independent metrics in our Artificial Immune Recognition System based model. According to this study, class-level metrics based model which applies AIRS algorithm can be used successfully for fault prediction and its performance is higher than J48 based approach. A fault prediction tool which uses this model can be easily integrated into the testing process.

1 Introduction

Software systems are becoming more and more complex and people's quality expectations are increasing. Therefore, it is necessary to manage these expectations as an engineering discipline called Software Quality Engineering [1]. Software Quality Engineering consists of many Quality Assurance activities and an important subset of them is testing. Other subsets are fault prevention, inspection, fault tolerance, formal verification and fault prediction.

These subsets can detect many software problems and even improve software testing process. Improvements in the testing process will reduce development lifecycle, project risks, resource and infrastructure costs. Because testing process is time-consuming and expensive, we may anticipate this problem with fault prediction models. These models provide a test strategy by focussing on fault-prone modules and testing duration decreases with this approach. In this study, we propose a novel fault prediction model to improve the testing process. Our goal is to predict the classes that will contain faults at the next release of an Object-Oriented System.

Current software metrics and defect data are used to construct the prediction model for the next release of software. Most of the datasets which locate in

J. Münch and P. Abrahamsson (Eds.): PROFES 2007, LNCS 4589, pp. 300–314, 2007.

PROMISE repository [2] have been collected at NASA as a part of Software Metrics Data Program for development projects. In this study, we have used datasets from this repository to construct our fault prediction model.

Metrics are independent variables and the fault-proneness of module is the dependent variable. Process or product metrics can be used for independent variables but mostly product metrics are used. Method-level and class-level metrics are two different metrics groups inside product metrics. Actually, Object-Oriented programming and procedural programming can benefit from method-level metrics because these programming paradigms have methods. In this study, class-level and method-level metrics have been used.

We use Object-Oriented metrics from Chidamber-Kemerer (CK) metrics suite [3] and we desire to enhance the performance of our prediction model with these metrics. Genetic Programming, Decision Trees, Neural Networks, Case-based Reasoning, Fuzzy Logic, Logistic Regression and Discriminant Analysis have been applied effectively for software fault prediction. As method-level metrics, Object-Oriented metrics are widely used for fault prediction and recent studies focused on these metrics [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], and [14].

In this paper, we applied class-level metrics and method-level metrics for AIRS based fault prediction model. We also investigated each of CK metrics within our proposed model. Our aim is not validating individual metrics, but applying AIRS to build a better fault prediction model. Furthermore, we used different statistical techniques such as correlation-based feature selection technique to select relevant features and investigated the power of the model. In addition, this study indicates that OO metrics are more useful than traditional metrics for our proposed model. Performance criteria shows that our model which uses CK metrics and lines of code provides better prediction capability than other models given in literature. According to experimental results, the performance of our prediction model is remarkable. This study is a part of our on-going research on software quality modeling and we aim to reach our vision by using Artificial Immune Systems paradigm.

This paper is organized as follows: the following section presents datasets and OO metrics. Section 3 describes evaluation criteria. Section 4 introduces natural and artificial immune systems. Section 5 provides experimental results. Section 6 presents conclusions and future works.

2 Metrics and Dataset

Chidamber-Kemerer OO metrics have been proposed in 1994 [3]. They have been used by many tool vendors and researchers . This study uses six Object-Oriented metrics of CK metrics. Also, we tried to get benefit from four more class level metrics too. Koru et al. [15] identified these four metrics for KC1 project, *Percent_Pub_Data*, *Access_To_Pub_Data*, *Dep_On_Child*, and *Fan_In*. We have also attempted to take advantage from traditional method-level metrics which are based on Halstead and McCabe metrics. McCabe is interested in the complex pathways and Halstead focuses on the readiness of the source code.