

Visualizing Reusable Code patterns based on Object-Oriented Reusability Metrics

Eun Young Byun, R. Young Chul Kim*

SELab., Dept. of Computer and Information Communication, Hongik University, 2639, Sejong-ro, Jochiwon-eup, Sejong, Korea 30016, eybyun@selab.hongik.ac.kr

**Dept. of Computer and Information Communication, Hongik University, 2639, Sejong-ro, Jochiwon-eup, Sejong, Korea 30016, bob@selab.hongik.ac.kr*

Reusability is necessary to improve software productivity and quality. At this time, there is an issue of “Which module should be reused? and how to identify the reusable module?”. This is very important to determine a whole software quality. Otherwise, to cause potential errors may depend on the quality of a module. To identify any reusable module, there is a need for reuse metrics to identify reusable modules. Existing metrics do not adequately reflect Object-Oriented mechanism, and lack of dynamic factors. To solve this problem, we suggest to define Object-Oriented reusability metrics, and automatically visualize to identify reusable modules. This allows developers to improve productivity through developing new system based on legacy system.

Keywords: Reusability, Object-Oriented, Reusable module, Metrics, Visualization

1. INTRODUCTION

Recently, in a maintenance, frequent update is undergoing in software market. This causes to decrease productivity and increase in time/cost. At this time, reuse is a technique to improve software productivity like process improvement [1,2]. Representative examples of such reuse are API, open source, library, and the like. Many programmers are reusing these elements to improve the efficiency of project development. In particular, reuse is important in the domain of developing the next version of the program based on the previous version of a program.

In reuse, there is an important issue of “Which module should be reused?”. The reasons are as follows:

- 1) if the quality of the reuse module is low, the quality of the program may be deteriorated.
- 2) if a reuse module has strong coupling with other modules, this module may cause potential errors.
- 3) if the reuse module is unused, the code is the dead code.

To solve this problem, we need to define reuse metrics to identify reusable modules.

In this paper, we derive problems of existing metrics and define Object-Oriented reusability metrics that are extended

and improved. We also develop the system to measure these metrics, which automatically identifies reusable modules. With this approach, we can visualize reusable modules in the architecture of software with invisible characteristics. This provides a general understanding of the program to developers, and guidelines for reuse.

The chapter 2 describes software visualization and problems of the previous reusability metrics in related work. The chapter 3 defines the improved Object-Oriented reusability metric. The chapter 4 explains the automatic reusable identification on the visualization system. Finally, we mention conclusions and future work.

2. RELATED WORK

2.1. Software Visualization

Previously, the following visualization studies were conducted [3-5]. By visualizing an architecture of the software with invisible characteristics, we improve the programmer’s understanding of the program and measure the complexity of the code by measuring various factors such as cohesion, coupling, bad smell, and performance. The complexity can lead to potential errors, resulting in lower productivity. Inevitable complexity cannot be reduced, but we must reduce unnecessary complexity. In this paper, we focus