A Longitudinal Study of Fan-In and Fan-Out Coupling in Open-Source Systems

Asma Mubarak, Brunel University, UK

Steve Counsell, Brunel University, UK

Robert M. Hierons, Brunel University, UK

ABSTRACT

Excessive coupling between object-oriented classes is widely acknowledged as a maintenance problem that can result in a higher propensity for faults in systems and a 'stored up' future problem. This paper explores the relationship between 'fan-in' and 'fan-out' coupling metrics over multiple versions of open-source software. More specifically, the relationship between the two metrics is explored to determine patterns of growth in each over the course of time. The JHawk tool was used to extract the two metrics from five open-source systems. Results show a wide range of traits in the classes to explain both high and low levels of fan-in and fan-out. Evidence was also found of certain 'key' classes (with both high fan-in and fan-out) and 'client' and 'server'-type classes with high fan-out and fan-in, respectively. This paper provides an explanation of the composition and existence of such classes as well as for disproportionate increases in each of the two metrics over time. Finally, it was found that high fan-in class values tended to be associated with small classes; classes with high fan-out on the other hand tended to be relatively large classes.

Keywords: *Coupling, Empirical, Fan-In, Fan-Out, Open-Source*

INTRODUCTION

Excessive class coupling has often been related to the propensity for faults in software (Briand et al., 1997). It is widely believed in the object-oriented (OO) community that excessive coupling between classes creates a level of complexity that can complicate subsequent maintenance and represents a 'stored up' maintenance problem. In practice, a class that is highly coupled to many other classes is an ideal candidate for re-engineering or removal from the system to mitigate both current and potential future problems. A problem that immediately arises however for the developer when considering re-engineering of classes with high coupling is: 'Do those classes have prohibitively large dependencies?' If so, then are those coupling dependencies 'incoming' or 'outgoing' dependencies? In theory, it is more

DOI: 10.4018/jismd.2011100101

difficult to modify a target class with high incoming and low outgoing coupling, since the former requires detailed and careful scrutiny of each of the many 'incoming' dependent classes and the possible side-effects of change.

In this paper, we investigate versions of five Open Source Systems (OSS) focusing on two well-known coupling metrics - 'fan-in' (i.e., incoming coupling) and 'fan-out' (i.e., outgoing coupling). We used an automated tool to extract each of the coupling metrics from those five systems. The research questions we explore are first, is it the case that classes with large incoming coupling naturally have low outgoing coupling and second, does this relationship worsen over time? In other words, does the potential maintenance problem become worse in terms of fan-in and fan-out values? Results showed a wide range of characteristics in the classes to account for high and low levels of fan-in and fan-out. Evidence was found of 'key' classes, comprising high fan-in and fan-out as well as 'client' and 'server'-type classes comprising high fan-out and fan-in, respectively. We explore the composition of the classes and reasons for the existence of such classes as well as for changes over time in the two metrics. We also found that high fan-in class values were associated with small classes; on the other hand, classes with high fan-out were comparatively large.

The main message that emerges from the research is that every system is likely to contain classes with relatively large amounts of coupling (whether fan-in, fan-out or both). The task of the developer is to ensure that such classes are monitored and a proactive stance taken to the possible re-engineering or refactoring of those classes. A strong link has been found between excessive coupling and faults (Briand et al., 1997) and it is widely acknowledged that too much coupling is harmful. The study presented is a first step to understanding the traits of this feature of systems at a broad level.

MOTIVATION AND RELATED WORK

The research in this paper is motivated by a number of factors. Firstly, previous research (Mubarak et al., 2008) has shown that there is a trade-off between coupling types – in particular, that between coupling through imported packages and the introduction of 'internal-to-the-package' coupling. In this paper, we explore the potential characteristics and trade-offs between fan-in and fan-out metrics over time. Second, we would always expect potentially problematic classes to be re-engineered by developers through techniques such as refactoring (Fowler, 1999); however, the practical realities of limited time and resources at their disposal means that only when classes exhibit particularly bad 'smells' (e.g. excessive coupling) (Fowler, 1999; Mantyla et al., 2006) are they dealt with. In this paper, we explore, over time, whether smells given by too much coupling do become 'smellier' and, if so, in what proportions. Third, it is likely that there are special types of class with either large fan-in or fan-out values (or both). Understanding the nature of these classes could help to understand the reason for these characteristics and even how and why these classes evolve as they do. Finally, the research is motivated by previous research (Mubarak et al., 2008) which showed that the fan-in and fan-out metrics tended to be relatively small for classes removed from a system. In other words, classes with either high fan-in and/or fan-out may be difficult to move or remove from a system. This question has inspired further examination of trends in the two metrics presented. The original fan-in and fan-out metrics were developed in early software metrics texts dealing with coupling and cohesion and structured programming (Stevens et al., 1974). Yet, very few studies have used fan-in and fan-out as a basis for empirical studies of coupling. A key objective of the