Evolution of Component Relationships between Framework and Application

Reishi Yokomori1  Harvey Siy2  Norihiro Yoshida3  Masami Noro1  Katsuro Inoue4

1 Department of Software Engineering, Nanzan University
27 Seirei-cho, Seto, Aichi 489-0863, Japan
{yokomori, yoshie}@se.nanzan-u.ac.jp

2 Department of Computer Science, University of Nebraska at Omaha
6001 Dodge Street, Omaha, NE 68182, USA
hsiy@mail.unomaha.edu

3 Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
yoshida@is.naist.jp

4 Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
inoue@ist.osaka-u.ac.jp

Received 10 January 2012; Revised 21 May 2012; Accepted 25 May 2012

Abstract. Most of today’s software applications are built on top of libraries or frameworks. The increasing number of cloud-based services gives rise to 3rd party frameworks that offer such services from a cloud platform. Just as applications evolve, frameworks also evolve. Such evolution is even more pronounced in frameworks that underlie cloud-based services. Upgrading is straightforward when the framework changes preserve the API and behavior of the offered services. However, major changes are introduced with the new framework release, which have a significant impact on the application. A framework user has to consider how to adjust to the new version. In this paper, we study the evolution of an application and its underlying framework through a multi-version analysis. For the analysis, we investigate two kinds of component relationships: one is component rank, the other is clone relation. Component rank measurement is a way of quantifying the importance of a component by its usage. As framework components are used by applications, the rankings of the components are changed. We confirm that upgrading to the new framework version has an impact to a component rank of the entire system. On the other hand, existence of code clone shows how application developers use existing framework code as a reference, and removal of clones shows which reuse activities were recognized as problematic. Analysis of results from these relationships provides useful insights into developers’ activities.

Keywords: software evolution, component relationships, use-relation, clone-relation

Acknowledgement

This research is supported by Nanzan University Pache Research Subsidy I-A-2 for the 2011 academic year.

References


