

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

- [1] I. Neamtiu and T. Dumitras, "Cloud Software Upgrades: Challenges and Opportunities," in *Proceedings of the International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA' 11)*, 2011.
- [2] R. Yokomori, M. Noro, K. Inoue, "Evaluation of Source Code Updates in Software Development Based on Component Rank," in *Proceedings of 13th Asia Pacific Software Engineering Conference*, pp. 327-334, 2006.

- [3] K. Inoue and R. Yokomori, T. Yamamoto, M. Matsushita, S. Kusumoto, "Ranking Significance of Software Components Based on Use Relations," *IEEE Transactions on Software Engineering*, Vol. 31, No. 3, pp. 213-225, 2005.
- [4] R. Yokomori, H. Siy, M. Noro, K. Inoue, "Assessing the Impact of Framework Changes Using Component Ranking," in *Proceedings of 25th IEEE International Conference on Software Maintenance*, pp. 189-198, 2009.
- [5] I. Jacobson, M. Griss, P. Jonsson, *Software Reuse*, Addison-Wesley Professional, New York, 1997.
- [6] C. Krueger, "Software Reuse," *ACM Computing Surveys*, Vol. 24, No. 2, pp. 131-183, 1992.
- [7] G. Blom, L. Holst, D. Sandell, *Problems and snapshots from the world of probability*, Springer, New York, 1994.
- [8] E. Gamma, R. Helm, R. Johnson, J.M. Vlissides, *Design Patterns: Elements of Reusable Object-oriented Software*, Addison-wesley Professional, New York, 1995.
- [9] L. Jiang, G. Misherghi, Z. Su, S. Gloudu, "DECKARD: Scalable and Accurate Tree-based Detection of Code Clones," in *Proceedings of the 29th International Conference on Software Engineering*, pp. 96-105, 2007.
- [10] Z. Li, S. Lu, S. Myagmar, Y. Zhou, "CP-miner: Finding Copy-paste and Related Bugs in Large-scale Software Code," *IEEE Transactions on Software Engineering*, Vol. 32, No. 3, pp. 176-192, 2006.
- [11] T. Kamiya, S. Kusumoto, K. Inoue, "CCFinder: A Multilinguistic Token-based Code Clone Detection System for Large Scale Source Code," *IEEE Transactions on Software Engineering*, Vol. 28, No. 7, pp. 654-670, 2002.
- [12] M. Ichii, M. Matsusita, K. Inoue, "An Exploration of Power-law in Use-relation of Java Software Systems," in *Proceedings of the 19th Australian Conference on Software Engineering*, pp. 422-431, 2008.
- [13] C.J. Kapser and M.W. Godfrey, " "Cloning Considered Harmful" Considered Harmful: Patterns of Cloning in Software," *Empirical Software Engineering*, Vol. 13, No. 6, pp. 645-692, 2008.
- [14] Y. Higo, T. Kamiya, S. Kusumoto, K. Inoue, "Method and Implementation for Investigating Code Clones in a Software System," *Information and Software Technology*, Vol. 49, No. 9-10, pp. 985-998, 2007.
- [15] M. Kim and D. Notkin, "Program Element Matching for Multi-version Program Analyses," in *Proceedings of the 2006 International Workshop on Mining Software Repositories (MSR' 06)*, pp. 58-64, 2006.
- [16] J. Davis, D.M. German, M.W. Godfrey, A. Hindle, "Software Bertillonage: Finding the Provenance of an Entity," in *Proceedings of the 8th Working Conference on Mining Software Repositories (MSR' 11)*, pp. 183-192, 2011.
- [17] D. German and A. Mockus, "Automating the Measurement of Open Source Projects," in *Proceedings of the 3rd Workshop on Open Source Software Engineering*, pp. 63-67, 2003.
- [18] J.D. Herbsleb, A. Mockus, T.A. Finholt, R.E. Grinter, "An Empirical Study of Global Software Development: Distance and Speed," in *Proceedings of the 23rd International Conference on Software Engineering*, pp. 81-90, 2001.
- [19] T. Zimmermann, P. Weissgerber, S. Diehl, A. Zeller, "Mining Version Histories to Guide Software Changes," in *Proceedings of the 26th International Conference on Software Engineering*, pp. 563-572, 2004.
- [20] P.M. Johnson, H. Kou, J.M. Agustin, Q. Zhang, A. Kagawa, T. Yamashita, "Practical Automated Process and Product Metric Collection and Analysis in a Classroom Setting: Lessons Learned from Hackstat-uH," in *Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE2004)*, pp. 136-144, 2004.
- [21] Y. Tamura and S. Yamada, "A Method of Reliability Assessment Based on Deterministic Chaos Theory for an Open Source Software," in *Proceedings of the Second International Conference on Secure System Integration and Reliability Improvement (SSIRI '08)*, pp. 60-66, 2008.
- [22] S. Black, S. Counsell, T. Hall, D. Biwes, "Fault Analysis in OSS Based on Program Slicing Metrics," in *Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2009)*, pp. 3-10, 2009.

- [23] J. Henkel and A. Diwan, "CatchUp!: Capturing and Replaying Refactorings to Support API Evolution," in *Proceedings of the International Conference on Software Engineering (ICSE '05)*, pp. 274-283, 2005.
- [24] D. Dig, S. Negara, V. Mohindra, R. Johnson, "ReBA: Refactoring-aware Binary Adaptation of Evolving Libraries," in *Proceedings of the International Conference on Software Engineering (ICSE '08)*, pp. 441-450, 2008.
- [25] K. Chow and D. Notkin, "Semi-automatic Update of Applications in Response to Library Changes," in *Proceedings of the International Conference on Software Maintenance (ICSM' 96)*, pp. 359-368, 1996.
- [26] J. H. Perkins, "Automatically generating refactorings to support API evolution," in *Proceedings of the Workshop on Program Analysis for Software Tools and Engineering (PASTE' 05)*, pp. 111-114, 2005.
- [27] Q. Zhangs, D. Qiu, Q. Tian, L. Sun, "Object-oriented Software Architecture Recovery Using A New Hybrid Clustering Algorithm," in *Proceedings of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 6, pp. 2546-2550, 2010.
- [28] E. Constantinou, G. Kakarontzas, I. Stamelos, "Towards Open Source Software System Architecture Recovery Using Design Metrics," in *Proceedings of the 15th Panhellenic Conference on Informatics*, pp. 166-170, 2011.
- [29] G. Antoniol, U. Villano, E. Merio, M. Di Penta, "Analyzing Cloning Evolution in the Linux Kernel," *Information and Software Technology*, Vol. 44, No. 13, pp. 755-765, 2002.
- [30] M. Mondal, C.K. Roy, M.S. Rahman, R.K. Saha, J. Krinke, K.A. Schneider, "Comparative Stability of Cloned and Non-cloned Code: An Empirical Study," in *Proceedings of the 27th ACM Symposium on Applied Computing*, 2012.
- [31] Y. Yamanaka, E. Choi, N. Yoshida, K. Inoue, T. Sano, "Industrial Application of Clone Change Management System," to be appeared in *Proceedings of the 5th International Workshop of Software Clones*, 2012.