Reconstruction of Common Software Architecture from Legacy Software

Motivation

There is a great need in industry for methods to re-engineer legacy software into maintainable and reusable representations that can sustain rapid changes in computing, technological, business and social environments that happen today and are inevitable in the future. For example, cloud computing promises much benefits, but it also fundamentally changes the way software systems are built and operated. In the cloud, software applications are composed of services distributed over the internet nodes. Among benefits, cloud reduces development costs due to service reuse, and eliminates the high capital and maintenance costs of hardware. However, to monetize the business value of a huge base of legacy systems through cloud services, companies must re-engineer their legacy systems to position them for cloud computing.

Most companies today develop and support multiple variants of the same software system, each variant tailored to a specific customer or platform. As the number of system variants explodes, their development and maintenance becomes expensive. In such case, re-engineering of system variants into a Software product Line (SPL) that is managed from common set of adaptable, reusable core assets can greatly improve productivity.

Project Goal and Scope

The aim of this project is to develop methods for reconstructing a common architecture underlying legacy software that consists of system variants, all of which are similar but also different.

The following research issues are involved:

1) Analysis of similarities and differences among system variants with clone detection techniques.
2) Recovering architectures of system variants with reverse engineering, and clustering.
3) Analyzing similarities and differences of system architectures with novel combination of clone detection and graph differencing algorithms.
4) Unifying system architectures into a common architecture to be shared by system variants.
5) Validation of proposed methods in industrial projects.

The contribution of this project will be in applying existing methods in novel ways to analyze system variants and come up with common architecture.

In the Master Thesis, the student will explore the methodological framework for the re-engineering solution, and work on methods for steps 1) and 2) above. The project may continue into PhD to formulate the overall re-engineering method and validate it.

Contact:

Stan Jarzabek
Dept. of Computer Science, School of Computing
National University of Singapore
stan@comp.nus.edu.sg; www.comp.nus.edu.sg/~stan
tel 65-6516-2863; mobile 65-96255863
Computing 1, 13 Computing Drive
Singapore 117417

References:

[1] 2011. Jarzabek, S, Pettersson, U. and Zhang, H. “University-Industry Collaboration Journey towards Product Lines,” Int. Conf. on Software Reuse, ICSR’2011, S. Korea, June 2011
[2] 2011. Jarzabek, S and Trung, H. D. “Flexible Generators for Software Reuse and Evolution,” Int. Conf. on Software Eng., ICSE’2011, New Ideas and Emerging Results Track, Honolulu, USA, May 2011
[3] 2011. Basit, H., Ali, U. and Jarzabek, S. “Viewing Simple Clones from a Structural Clones’ Perspective,” Int. Workshop on Software Clones, *IWSC’2011, ICSE Workshop*, Honolulu, USA, May 2011
[4] 2011. Xue, Y., Xing, Z. and Jarzabek, S. “Analyzing Clones by Differentiation,” Technical Report, NUS, 2011.
[5] 2010. Xue, Y., Xing, Z. and Jarzabek, S. “Understanding Feature Evolution in a Family of Product Variants,” *17th Working Conf. on Reverse Engineering*, Boston, MA, October 2010, pp. 109-118
[6] 2010. Basit, H. and Jarzabek, S. “Towards Structural Clones: Analysis and semi-automated detection of design-level similarities in software,” VDM Verlag, 2010 (172 pages)
[7] 2009. Ye, P., Peng, X, Xue, Y. and Jarzabek, S. “A Case Study of Variation Mechanism in an Industrial Product Line,” *11th Int. Conf. on Software Reuse, ICSR09*, Falls Church, VA, USA, Sept. 27-30, 2009, pp. 126-136
[8] 2009. Basit, H. A., Jarzabek, S. “Data Mining Approach for Detecting Higher-level Clones in Software,” *IEEE Trans. on Soft. Eng.*, July/August 2009 (vol. 35 no. 4) pp. 497-514
[9] 2008. Zhang, Y., Basit, H., Jarzabek, S., Anh, D. and Low, M. “Query-based Filtering and Graphical View Generation for Clone Analysis,” *Proc. 24th IEEE Int. Conf. on Software Maintenance, ICSM’08*, Beijing, September 2008

[10] 2008. Rajapakse, D. and Jarzabek, S. “Towards generic representation of web applications: solutions and trade-offs,” Software, Practice & Experience, Volume 39 Issue 5, April 2009, pp. 501 – 530, Published Online: 27 Nov 2008

Project Plan:

Month	Task
1	Literature survey
2	Analysis of software similarities and differences in system variants
3	Reverse engineering low level design models from code
4	Analysis for software similarities and differences in models
5	Setting up conceptual framework for recovery of common architecture
6	wrapping up, report/paper writing