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 65-9625863
Computing 1, 13 Computing Drive
Singapore 117417

References:
Project Plan:

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