**CS/SE 6301 Software Analysis and Comprehension**

**Spring 2016**

Project Proposal 2.0

**Common coupling detection in Java: srcML and Eclipse JDT/AST approach**

**Ronaldo Gonçalves Junior**

**Sungsoo Ahn**

**Bennilyn Quek**

April 5, 2016

Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 0.2 | Sung-Soo | Preliminary Proposal | 3/29/2016 |
| 1.2 | Ronaldo | Draft Proposal | 4/5/2016 |
| 2.0 | Ronaldo, Bennilyn, Sung-Soo | Final Proposal | 4/5/2016 |
|  |  |  |  |

### Introduction

This document is the proposal for the project – Common coupling detection in Java. It presents a plan to achieve the project goal, which is detect common coupling in Java source codes, calculate coupling degree, and provide common coupling information to developers and maintainers for high quality system.

### Common coupling

Common coupling (also known as Global coupling) occurs when two modules are associated to the same global data (e.g., usage of a global variable) [1]. In other words, any changes to this data could potentially imply changes to all the modules that uses it. One additional aspect worth assessing is the set of constraints that might limit how java files (for instance) could be coupled. According to [4], there are two distinct categories of common coupling: strong and weak. Strong common coupling is when it is possible to access a global variable from any files of the system. The last one, weak common coupling, is when the global variable can only be accessed within files of a same package.

### Java context

Some languages like Java do not have global variables [2]. In Java, all variables that are not declared as local variables are declared as fields of a class. That is, all variables are declared within the scope of either a class or a method. In Java, static fields (also known as class variables) exist independently of any instances of the class and one copy is shared among all instances; hence public static fields are used for many of the same purposes as global variables in other languages because of their similar "sharing" behavior [2].

### Other types of coupling

Coupling is not limited to shared global data (i.e., common coupling). There are five other types of coupling [3]: data coupling, stamp coupling, control coupling, external coupling, and content coupling. These types of coupling refer to other categories of source code connection, which include parameters, data structure, logic control, and so on and so forth. However, none of these coupling types is considered in this project.

### Project deliverables

|  |  |
| --- | --- |
| Due Date | Description |
| April 12 | * Interim project presentation (slide presentation, 5-15 mins) * Revised schedule. |
| April 14 | * Interim report document (paper and electronic[[1]](#footnote-1) form) |
| April 26 | * Final project presentation (slide presentation, 15-20 mins) |
| May 3 | * Final project report (paper and electronic1 form) |

### Methods, tools, and techniques

The following will be used for the project:

* srcML
* Eclipse JDT/AST
* Language: Java
* System: Apache OpenNLP

### Milestones

|  |  |  |
| --- | --- | --- |
| Milestone | Due Date | Description |
| Survey | 4/7/2016 | Survey and summarization of common coupling and detection journals |
| Prototype | 4/12/2016 | Implementation of a single instance of common coupling detection |
| Implementation | 4/21/2016 | Implementation and experiments of detecting common coupling |
| Validation | 4/26/2016 | Manual validation, discussion and presentation |

### References

1. Wikipedia. **Coupling (computer programming).** Accessed on: March 29. Available at: <<https://en.wikipedia.org/wiki/Coupling_(computer_programming)>>.
2. Wikipedia. **Global variable.** Accessed on: March 29. Available at: <<https://en.wikipedia.org/wiki/Global_variable>>.
3. Boukari Souley and Baba Bata. **A Class Coupling Analyzer for Java Programs.** West African Journal of Industrial and Academic Research Vol.7 No. 1 June 2013.
4. Michel Chaudron. **Design Heuristics and Architectural Styles (LL Chapter 9).** Accessed on: April 5. Available at: <https://rickvanderzwet.nl/trac/personal/export/3/liacs/se/slides/09\_Design\_Heuristics\_and\_Styles\_part2.pdf>.

1. Team’s website - https://github.com/rpgoncalves/sw-comprehension [↑](#footnote-ref-1)