# Analysis of Van Schedule

## Overview

We used two tools to analyze the design of SchemaAnalyst, JDepend which allowed us to compare the number of afferent versus efferent couplings and how this affected the overall stability of a program. The second tool that we used to analyze the design of our systems was JavaNCSS. JavaNCSS calculates a number of metrics, for different levels of a system (e.g., function level, class level, package level), but we chose to focus on three metrics. Those metrics were Average NCSS, Total NCSS, and Average CCN. We chose to analyze these metrics for the class and function levels. More detail in regards to our findings will be provided in the following section.

## Analysis of Findings

As aforementioned, the first tool that we used to measure the quality of the design for our systems was JDepend. We used JDepend to find the number of afferent and efferent couplings. From these metrics, we were able to conclude that the more afferent couplings and fewer efferent couplings. Although the system shows a great deal of packages that have responsibility to the system as a whole there are other aspects that JDepend test for that shows the instability of a package in the sysem. For example, looking at the data found, we determined that the "org.schemaanalyst.mutation" package can be determined as unstable due to the ratio of the number of efferent coupling (Ce) to the total number of coupling, (Ca+Ce). All together, the system is stable.

### JavaNCSS- Overview

The second tool used to measure the quality of the design for SchemaAnalyst was JavaNCSS. JavaNCSS calculated three metrics for both classlevel and function level used when analyzing our system. These metrics were the average NCSS, average CCN, and total NCSS. At a function level, the open-source systems tended to have less non commented source statements than the our systems.

### Analysis of Findings

At an object level, Van Schedule possessed on average of non commented source statements per package 705.50 and number of functions per package to be about 90.50. These numbers in the Van Schedule were good, allowing us to conclude that these open-source tools are designed well according to JavaNCSS. ALso the Average CCN, cyclomatic complexity number, was 1.7