Lab 10 Metrics

Lab report:

Software Design Lab 10: Analyzing Structural Quality with Software Metrics

Ronan Noonan

Student Number: G00384824

Date: 19/12/2023

Introduction

The objective of Lab 10 is to employ software design metrics to analyze the structural quality of the Apache Commons Text API. Software metrics are essential tools that provide quantitative measures to assess the quality of software design. These metrics offer insights into the complexity, maintainability, and overall robustness of software artifacts.

Methodology

The Apache Commons Text API was selected as the subject for analysis. The Metrics plugin for Eclipse was utilized to gather data on various metrics such as Cyclomatic Complexity, Coupling Factors, Cohesion, and Inheritance. The setup involved installing the Metrics plugin (https://metrics2.sourceforge.net/update/), configuring the ApacheText project within Eclipse, and ensuring the appropriate CLASSPATH settings for dependencies.

Analysis

Metrics data was collected as follows:

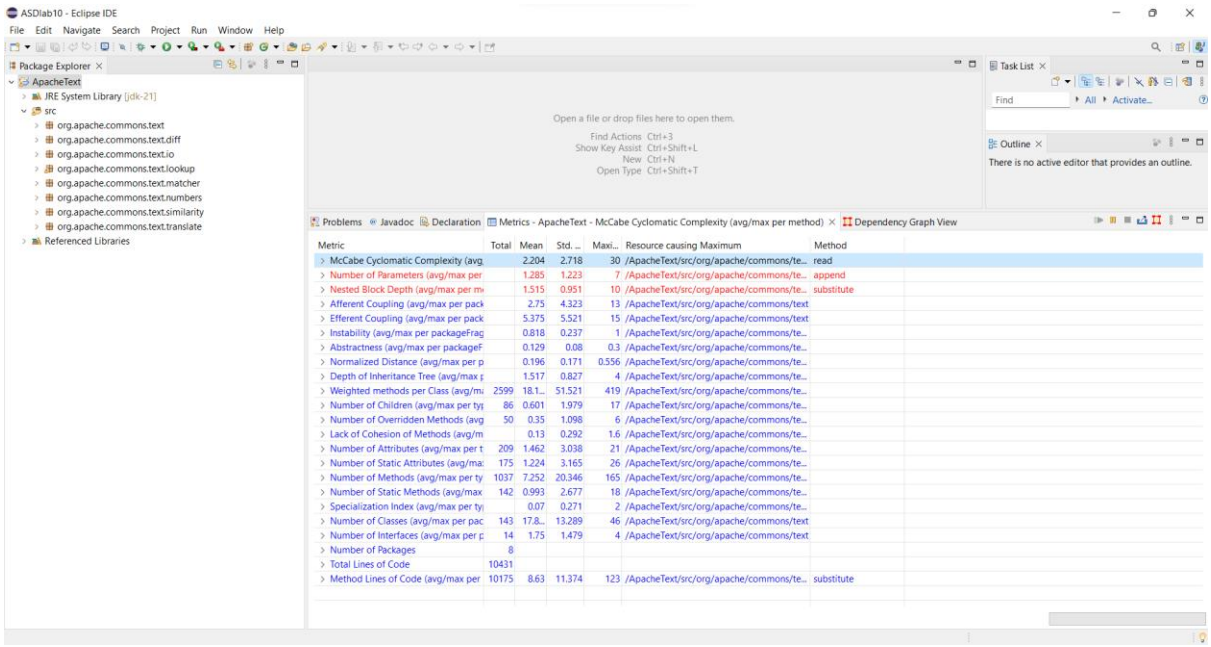
- McCabe Cyclomatic Complexity: The average was 2.718 with a maximum of 30 in the read method, suggesting that some methods may require refactoring to simplify complex decision paths.
- Number of Parameters (avg/max per method): With an average of 1.223 and a maximum of 7, the append method could be revised for parameter object to improve readability and maintainability.
- Afferent and Efferent Coupling: The average afferent coupling was 5.325, and efferent was 3.573, with the substitute method showing high coupling, indicating potential for better modularization.
- Instability: The low average instability of 0.237 suggests a stable codebase, but attention should be given to areas where the instability is maximum, which could lead to difficulties in future modifications.
- Depth of Inheritance Tree: An average of 1.517 with a maximum of 4 suggests a well-managed inheritance hierarchy, reducing complexity and increasing code reuse.

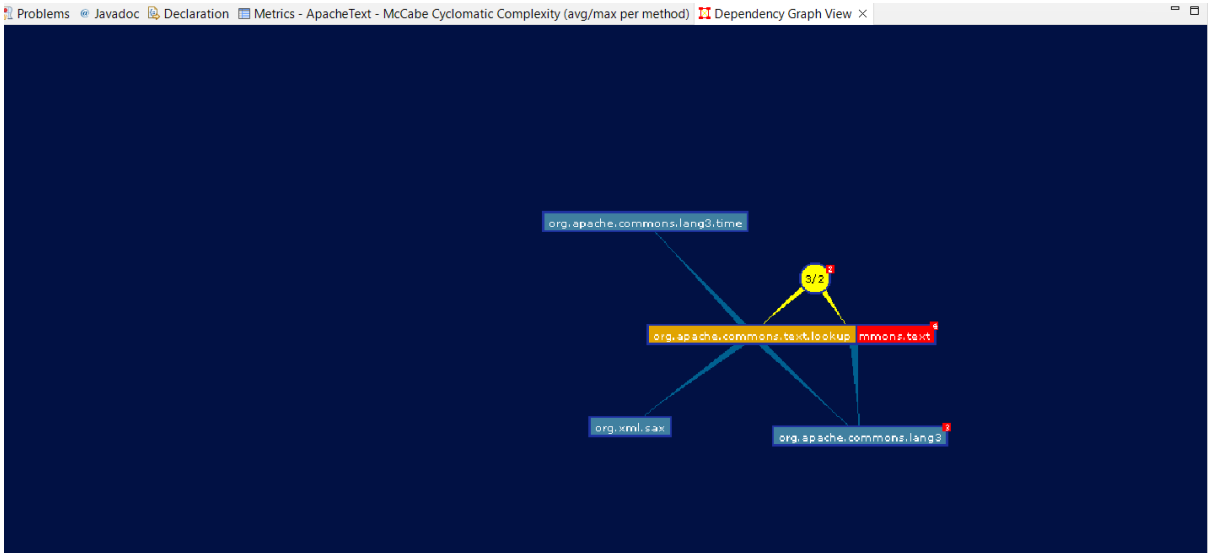
The dependency graph analysis identified org.apache.commons.text.lookup as a central node, indicating it has significant dependencies and may be a critical point of failure or complexity.

Conclusion

The analysis indicates that the Apache Commons Text API exhibits a balanced use of object-oriented principles, with areas of high complexity that could benefit from refactoring. The stability of the packages is satisfactory, with some exceptions which could be optimized to reduce future maintenance overhead.

Appendices


ASDlab10 - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
ApacheText
JRE System Library [jdk-21]
src
org.apache.commons.text
org.apache.commons.text.diff
org.apache.commons.text.io
org.apache.commons.text.lookup
org.apache.commons.text.matcher
org.apache.commons.text.numbers
org.apache.commons.text.similarity
org.apache.commons.text.translate
Referenced Libraries
Open a file or drop files here to open them.
Find Actions Ctrl+3
Show Key Assist Ctrl+Shift+L
New Ctrl+N
Open Type Ctrl+Shift+T
Task List
Find
All
Activate...
Outline
There is no active editor that provides an outline.
Problems
Javadoc
Declaration
Metrics - ApacheText - McCabe Cyclomatic Complexity (avg/max per method)
Dependency Graph View
Metric
Total
Mean
Std. ...
Maxi...
Resource causing Maximum
Method
McCabe Cyclomatic Complexity (avg
Number of Parameters (avg/max per
Nested Block Depth (avg/max per m
Afferent Coupling (avg/max per pack
Efferent Coupling (avg/max per pack
Instability (avg/max per packageFrag
Abstractness (avg/max per packageF
Normalized Distance (avg/max per p
Depth of Inheritance Tree (avg/max p
Weighted methods per Class (avg/ma
Number of Children (avg/max per typ
Number of Overridden Methods (avg
Lack of Cohesion of Methods (avg/m
Number of Attributes (avg/max per t
Number of Static Attributes (avg/ma
Number of Methods (avg/max per ty
Number of Static Methods (avg/max
Specialization Index (avg/max per ty
Number of Classes (avg/max per pac
Number of Interfaces (avg/max per p
Number of Packages
Total Lines of Code
Method Lines of Code (avg/max per


Problems
Javadoc
Declaration
Metrics - ApacheText - McCabe Cyclomatic Complexity (avg/max per method)
Dependency Graph View
org.apache.commons.lang3.time
3/2
org.apache.commons.text.lookup
org.xml.sax
org.apache.commons.lang3