Advanced Software Design – Metrics Lab

Lab Report

KYRYLO KOZLOVSKYI – G00425385
12-11-2025

Contents

1. Most Referenced Packages and Types

Using the Eclipse Metrics plugin, I analysed the Apache Commons Text API to identify packages and types with the highest coupling. The dependency graph revealed several knots, packages that act as central hubs with high fan-in or fan-out values.

1.1 Highest Fan-Out

These packages depend on many other packages, indicated by their high method counts and outgoing dependencies:

Rank 1: org.apache.commons.text - 761 methods (Extremely high)

Rank 2: org.apache.commons.text.lookup - 120 methods (High)

Rank 3: org.apache.commons.text.numbers - 81 methods (Moderate)

Rank 4: org.apache.commons.text.similarity - 77 methods (Moderate)

1.2 Highest Fan-In

Based on the dependency graph, these packages are heavily depended upon by others:

- org.apache.commons.text - Central hub, all other packages depend on it

- org.apache.commons.lang3 (external) - Heavily used throughout the project

- org.apache.commons.text.matcher - Used by text and lookup packages

- org.xml.sax (external) - Used for XML processing functionality

1.3 Circular Dependencies Detected

The metrics analysis identified a circular dependency involving three packages, creating a knot in the dependency graph:

- org.apache.commons.text

- org.apache.commons.text.lookup

- org.apache.commons.text.matcher

This circular dependency means these packages cannot be independently deployed or tested, and changes in one package may cascade through all three.

2. Package Stability

Using positional stability metrics (I = Ce / (Ca + Ce)), where Ce is efferent coupling and Ca is afferent coupling, I analyzed the stability of key packages. Stability ranges from 0 (maximally stable) to 1 (maximally unstable).

2.1 org.apache.commons.text - UNSTABLE

Estimated Stability: I ≈ 0.7-0.8

Supporting Metrics:

- Method Count: 761 methods (extremely high)

- Cyclomatic Complexity: Average CC of 2.204, but contains methods with CC up to 30

- Efferent Coupling: Very high - depends on lookup, matcher, similarity, translate, diff, numbers of packages plus external dependencies

- Afferent Coupling: High - all internal packages depend on it

- Circular Dependency: Part of 3-node cycle

Conclusion: Despite having many dependents (which should increase stability), the core text package is unstable due to its involvement in circular dependencies and extremely high method count. The high efferent coupling means it has many reasons to change, making it volatile.

2.2 org.apache.commons.text.lookup - UNSTABLE

Estimated Stability: I ≈ 0.8

Supporting Metrics:

- Method Count: 120 methods

- Efferent Coupling: High - depends heavily on core text package and external resources (DNS, files, URLs, XML)

- Afferent Coupling: Low - mainly used by StringSubstitutor classes

- Circular Dependency: Part of 3-node cycle

Conclusion: This package is unstable because it has many dependencies but few dependents. Being part of the circular dependency cycle further reduces its stability.

2.3 org.apache.commons.text.similarity - RELATIVELY STABLE

Estimated Stability: I ≈ 0.3

Supporting Metrics:

- Method Count: 77 methods

- Highest CC Method: findDetailedResults with CC=27 (algorithmic complexity)

- Efferent Coupling: Low - minimal dependencies, mainly on core utilities

- Afferent Coupling: Moderate - used by various text processing components

- Circular Dependencies: None

Conclusion: This package is relatively stable because it has few dependencies but is used by multiple other components. The high CC values are due to complex string algorithms (Levenshtein, Jaro-Winkler) rather than poor structure.

2.4 org.apache.commons.text.io - STABLE

Estimated Stability: I ≈ 0.4

Supporting Metrics:

- Method Count: 10 methods (smallest package)

- Highest CC Method: read method with CC=30 (highest in entire project)

- Efferent Coupling: Very low - minimal dependencies

- Afferent Coupling: Low - specialized I/O functionality

- Circular Dependencies: None

Conclusion: Despite containing the highest CC method in the project, this package is stable due to its small size, focused purpose, and minimal coupling. The high CC is localized to one complex I/O operation.

3. Summary

The Apache Commons Text API shows mixed stability. The core text package and lookup package are unstable due to high coupling and circular dependencies (I ≈ 0.7-0.8). In contrast, specialized packages like similarity and io are more stable (I ≈ 0.3-0.4) due to focused responsibilities and cleaner dependency structures. The circular dependency between text, lookup, and matcher packages is the most significant architectural issue that reduces overall stability.

4. Dependency Graph