CS442_542 / Software Design Patterns

Mert Kılıçaslan – 26748
Mustafa Ata Onbaş – 22343

Software Design Patterns used while developing our framework:

Creational

Singleton Pattern:

The CloneDetectionMain class could be considered a Singleton, as it seems to be intended to be run as a standalone application, and there is no apparent need for more than one instance.

Factory Pattern:

The method getNormalizedText(Token t) in the CloneDetectionMain class can be seen as a simple factory method, as it creates a normalized token from an input token, abstracting away the specifics of how different types of tokens are normalized.

Behavioural

Command Pattern:

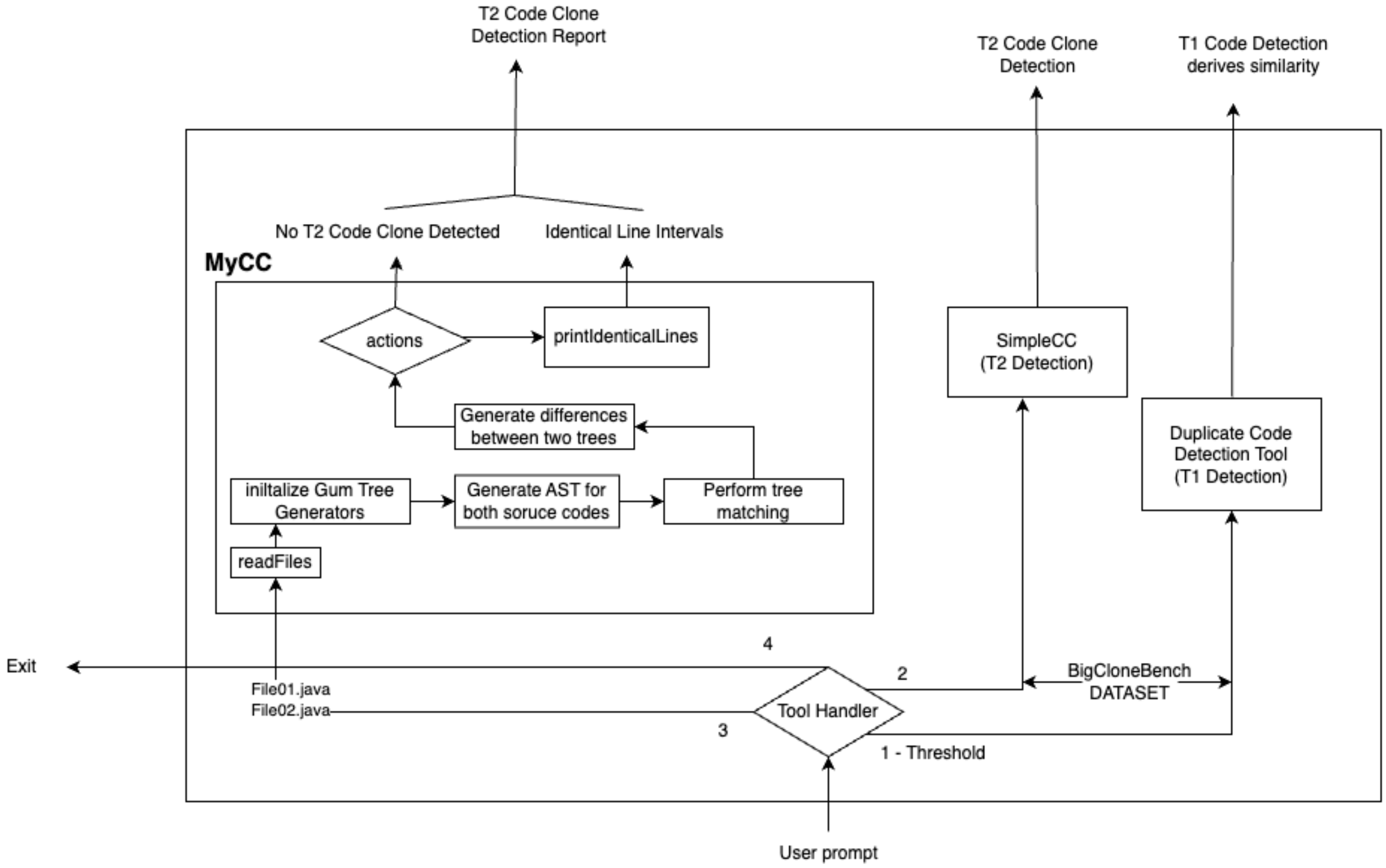
In this case, you could argue that the ProcessBuilder in the inner loop of SimpleCCExample represents the command pattern. It encapsulates the command to run the SimpleCC tool, including the classpath, main class, and file names, which are then executed at a later point using processBuilder.start().

Structural

Facade Pattern:

In a way, you can view the SimpleCCExample class as a Facade that abstracts the complexity of running SimpleCC tool for a list of files and collecting its output.

Code Clone Detection Analyser System Structure


T2 Code Clone Detection Report
T2 Code Clone Detection
T1 Code Detection derives similarity
No T2 Code Clone Detected
Identical Line Intervals
MyCC
actions
printIdenticalLines
Generate differences between two trees
iniltalize Gum Tree Generators
Generate AST for both soruce codes
Perform tree matching
readFiles
SimpleCC (T2 Detection)
Duplicate Code Detection Tool (T1 Detection)
4
Exit
2
BigCloneBench DATASET
File01.java
File02.java
3
Tool Handler
1 - Threshold
User prompt