

CCT ソースコード読解

1.CCT の処理の大まかな流れ

- ・コードクローン追跡機能
- (1)リビジョン関係特定処理
- (2)コード片特定処理
- (3)クローン特定処理
- (4)クローン追跡処理
- ・コードクローン分析機能
- (5)クローン関係者特定処理
- (6)不具合クローン特定処理

2.CCT を構成する JAR ファイル

STEP1: コミット情報の抽出

CommitInfoExtractor.jar

STEP2: ファイル情報の抽出

FileDetectorMain.jar

STEP3: 最も古いコミット情報から,コード片とコードクローンを抽出

CloneDetectorInFirstCommitMain.jar

STEP4: すべてのコミット情報から,コード片とコードクローンを抽出

CloneDetectorInNextCommitMain.jar

STEP5: クローン同士にリンクをはる

CloneSetLinkerMain.jar

STEP6: クローンリンクを類似度を用いてはる

DeleteNewCloneSetCheckerMain.jar

STEP7: クローン情報を取り出す

CloneTraceMain.jar

STEP8: SZZ の実行

ExecuteSZZMain.jar

STEP9: クローン変更情報と SZZ の比較

CloneComparatorSZZMain.jar

3.build.xml に記述されている main クラス

CommitInfoExtractor.jar-

commitInfoExtractor.CommitInfoExtractor

FileDetectorMain.jar-

filedetector.FileDetectorMain

CloneDetectorInFirstCommitMain.jar-

main.CloneDetectorInFirstCommitMain

CloneDetectorInNextCommitMain.jar-

main.CloneDetectorInNextCommitMain

CloneSetLinkerMain.jar-

main.CloneSetLinkerMain

DeleteNewCloneSetCheckerMain.jar-

checkcloneset.DeleteNewCloneCheckerMain

CloneTraceMain.jar-

cloneExtractorTest.CloneTraceMain

ExecuteSZZMain.jar-

szz.ExecuteSZZMain

CloneComparatorSZZMain.jar-

szz.clonecomparator.CloneComparatorSZZMain

4.それぞれの main クラスの概要

4.1.commitInfoExtractor.CommitInfoExtractor

解析対象の git リポジトリから/.git ディレクトリを探し、コミットした情報を取り出す