

コードクローン変更履歴可視化システム CCEvovis マニュアル

2019 年 12 月 6 日

1. システム概要

本システムは、複数バージョン間のコードクローンの変化を分析し、その情報を可視化し、開発者に提供することを目的としている。

分析対象としているのは、以下の2つのプロジェクト。

(1) ローカルにあるプロジェクト

分析をするためには、複数バージョンのプロジェクトを同一のフォルダに集めて、フォルダ名をバージョンの日付に改変する必要がある。

(2) Git のリモートリポジトリ

Public に公開されているリモートリポジトリの URL、ブランチ、分析期間、分析間隔を与えると自動で分析を開始。

2. 動作環境

以下のプラットフォームで動作することを確認。

Windows10

また、システムが動作する前提として、以下の前準備が必要

- Java の実行環境 (Java 1.8)
- Git の実行環境 (リモートリポジトリを対象とするなら必要)
- Python 2.6 (32bit 版) の実行環境

3. 構成

本システムは以下の2つのツールで構成されています。

• setting.jar

CCEvovis の設定ツール。設定ファイルを作成することが出来る。ただの設定ファイルを作成するツールなので、自身のエディタで設定ファイルを編集出来る。

• analyze.jar

CCEvovis 本体。ccm バッチファイルは analyze.jar を実行するスクリプトで、編集して設定ファイルの引数を変更して適宜実行出来る。

4. 設定方法

4.1. 主な実行の流れ

- (1) `setting.jar` を起動し, 分析を行うプロジェクトに関する各種設定項目を入力. または, 自身のエディタで設定ファイルの作成&編集.
- (2) `ccm` バッチファイルを編集して, `analyze.jar` に手順(1)で生成した設定ファイルを引数に設定.
- (3) `ccm` バッチファイルを実行
- (4) 出力先に指定したフォルダの¥users¥ユーザ名 (既定は `guest`) ¥以下に分析結果がある.

4.2. 設定ファイルの項目

4.2.1. 共通設定

- **PROJECT_NAME:**
プロジェクト名
- **ANALYSIS_NAME:**
分析の詳細な名前
- **TOOL:**
以下の 3 つのツールのどれかを入力してください
 - (1) **SourcererCC**
大規模なソフトウェアに対して, 意味的に処理が類似したコードクローンを高速に検出するツール. 検出時間は早め.
 - (2) **CCFinderX**
字句解析を用いることで, 構文的に一致した字句単位のコードクローンを検出するツール. 検出時間は遅め.
 - (3) **CCVolti**
情報検索技術を用いることで, 意味的に処理が類似したブロック単位 (関数単位より小さい粒度) のコードクローンを検出するツール. 検出時間は早め.
- **LANGUAGE:**
対象プロジェクトの言語
- **WEB:**
WEB インターフェースで結果を見たい場合は, `true` に設定.

- **HTML_DIR:E:**
分析結果が出力されるフォルダを指定
- **CSV:**
CSV 出力で結果を見たい場合は, **true** に設定.
- **WORK_DIR**
作業用のワークディレクトリ. 入力なしの場合の規定値は,
~¥CCEvovis¥file となる
- **USER_ID:**
ユーザ名を指定. 指定なしの場合は **guest** になる.

4.2.2. Git のリモートリポジトリを対象に設定する場合

- **GIT_DIRECT:**
Git のリモートリポジトリを対象に
検出を行う場合は **true** (LOCAL_TARGET は **false** に設定)
そうでない場合は **false** (LOCAL_TARGET は **true** に設定)
- **GIT_BRANCH:**
対象とするブランチ名を指定 ex) master
- **TARGET_DIR:**
リポジトリがダウンロードされるフォルダを指定
- **START_DATE:**
分析開始日の指定. 例えば 2018 年 5 月 21 日の場合は, 20180521 と入力.
- **END_DATE:**
分析終了日の指定
- **INTERVAL:**
分析間隔を入力. 一週間間隔で分析したい場合は, 7 を入力. 入力された分析開始日と終了日間で一週間ごとに分析を開始する.

4.2.3. ローカルのプロジェクトを対象に設定する場合

- **LOCAL_TARGET:**

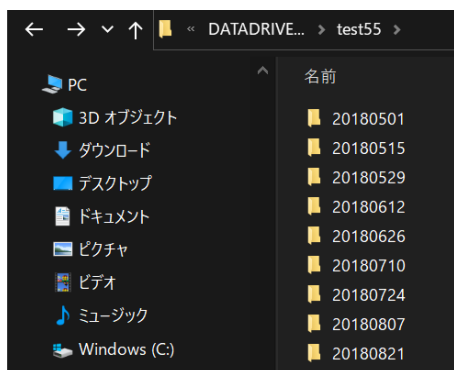
ローカルのディレクトリを対象に

検出を行う場合は **true** (GIT_DIRECT は **false** に設定)

そうでない場合は **false** (GIT_DIRECT は **true** に設定)

- **TARGET_DIR:E:**

分析対象のプロジェクトがあるフォルダを指定. ここにある全フォルダを分析対象とする. フォルダ名は日付にしないと分析できない. 例えば, 2019 年 9 月 10 日のバージョンの場合は, 「20190910」というフォルダ名にする必要がある. 下記画像参照.



5. 注意点

- **setting.jar** は動作環境によって画面描画に失敗する可能性がある. その場合は, ディスプレイの設定より画面のサイズを変更してみてください.
- 検出ツールによってバグがあるため, 分析出来ないプロジェクトがある. **CCFinderX** は, バグが多く, プロジェクトによっては分析できないことがある. このため, なるべく分析が比較的安定している, **SourcererCC** と **CCVlti** を使うことをお勧めする.
- 分析に関する **log** は **ccm.log** に出力される.

6. 連絡先

バグ, 質問, 分析できないプロジェクトなどありましたら, 以下にお気軽にご連絡ください.

メール : h-honda@ist.osaka-u.ac.jp

Skype : live:f7aa671626ff244f