

開発者データを用いたコード クローン分析

久木田 雄亮

背景：コードクローン

- ソースコード中に存在する同一，類似するコード片のこと
- ソフトウェア保守を行う上で障害になりうると指摘されているもの
- 近年では，必ずしも悪影響を与えるとは限らないと考えられている
 - － 良いクローンと悪いクローンがある？

背景：コードクローン進化の分析

- 開発履歴情報を分析し，コードクローンがどのような進化をたどったかを特定
 - コードクローン追跡
 - 生成->要素の追加-> 要素の削除->消滅
- コードクローンの系譜を分析
 - どれくらいの間生存したか
 - 何回修正されたか

背景：コードクローン研究

- 既存の研究として，検出，管理など
 - 主にコードやバグデータに着目した研究が多数を占めている．
- 開発者とクローンの関係を分析した研究はあまりなされていない

目的

- 良いコードクローンと悪いコードクローンの生成される理由を開発者のデータを用いて明らかにする
 - 良いコードクローン
 - 修正回数：少，生存期間：長
 - 悪いコードクローン
 - 修正回数：多，生存期間：長 or 短

アプローチ

1. コードクローン追跡ツール（ECTEC改）を使い，コードクローンの系譜を特定
2. 誰がどのクローンを生成したかを特定
3. クローンを生成した開発者の特徴を分析
 - － 期間，コミット回数，議論の参加回数
4. どんな特徴を持つ開発者が良いクローン，悪いクローンを生成するか分析

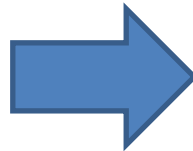
ECTEC

- Subversion リポジトリを解析し，コードクローンの系譜を特定するツール
- 入力
 - Javaで記述され，Subversionで管理されているソフトウェアのSubversion リポジトリ
- 出力
 - コードクローンの系譜に関する情報が保存されたデータベース
- 改善点
 - Git リポジトリに対応していない

ECTECのクローン検出方法

- ブロック単位のコードクローン検出手法
 - ソースコードからブロックを抽出し，ブロック同士を比較する
 - ブロック
 - クラス，メソッド，ブロック文（if文，for文）

```
public class ClassA{  
    public void methodA (int n) {  
        if(n == 10){  
            //some code  
        }  
    }  
    ...  
}
```



ブロック
抽出

```
ClassA.java, ClassA, 5  
methodA(int), 6  
if, n==10, 2
```

ECTEC改

- 現在あるECTECはSubversionのみに対応
- ECTEC改ではGitリポジトリに対応予定
 - JGitを使いGitリポジトリ操作
 - Gitリポジトリに対応することで...
 - コードの著者データを取得できる
 - Gitリポジトリを使ったプロジェクトを対象可
- 11月中に完成予定