

コードクローンの利用過程における 不具合混入原因を分析するための ツールの提案と評価

大石 聖也

コードクローン

- ソースコード中に存在する, 互いに「一致」または「類似」するコード片
- ソフトウェア品質を低下させる要因
 - (例) 一部を変更する際, コードクローン全体を変更しないといけない
 - コストがかかる
 - 変更忘れから, 新たな不具合が発生
- ソフトウェア品質を低下させる要因として, 様々な研究がされてきた[1]

[1] 肥後 芳樹, 楠本 真二, 井上 克郎. コードクローン検出とその関連技術. 電子情報通信学会論文誌D, Vol.J91-D, No.6, pp.1465-1481, 2008年6月

既存研究

- コードクローン分析に関する研究[2][3]
 - コードクローンによるバグ伝播がどれほど発生しているか調査
 - 複数プロジェクトにまたがった開発者ごとのソースコードの再利用動向を分析
- コードクローン追跡に関する研究[4]
 - 生存期間が長く, 複数回修正されるコードクローンがどの程度存在するかを調査

[2]

[3]

[4] 堀田 圭佑, 肥後 芳樹, 楠本 真二. CRDを用いたコードクローンの生存期間と修正回数に関する調査. 情報処理学会論文誌 Vol.55 No.2 947-958 (Feb. 2014)

既存研究の問題点

- コードクローンの利用過程における不具合混入原因として, 開発者を考慮した分析がない

CCT (Code Clone Tracer) ^[5]

- コードクローンの利用過程における人的影響を調査することを目的として設計されたコードクローン追跡ツール
 - 機能
 - クローンの特定・追跡を行う機能
 - 不具合管理システムと連携する機能
- 開発者・不具合・コードクローンの紐づけが行える

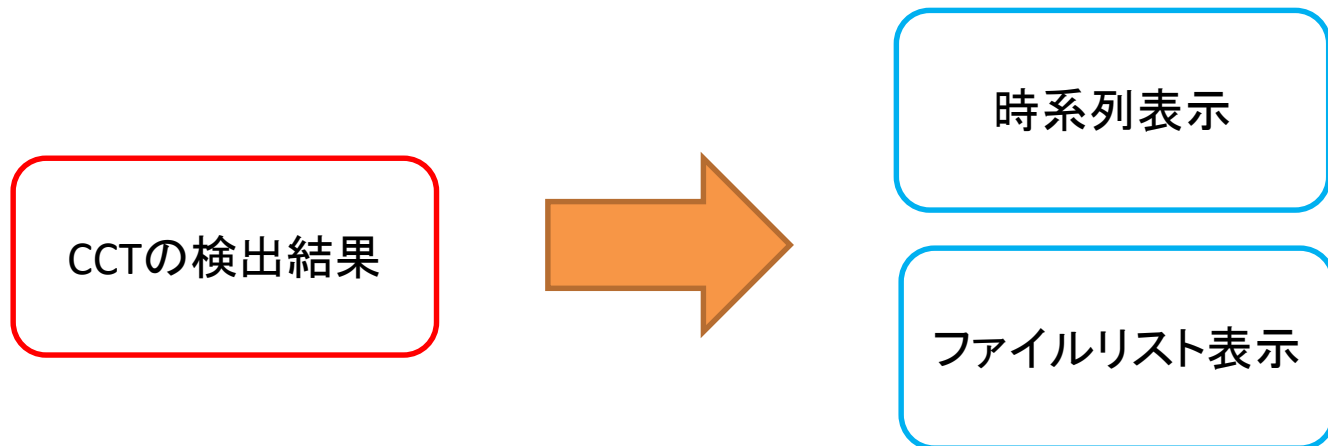
[5] 久木田 雄亮, 大平 雅雄. コードクローン変更過程における開発者のインタラクションとソフトウェア品質の関係. ソフトウェア・シンポジウム2017 in 宮崎. pp.120-129, 2017年6月

CCTから必要なこと

- 検出結果が膨大なため, 複数ある項目から必要な要素を取得・整理が必要
- コードクローンの利用過程を把握するために, 図示化が必要

目的

- 目的
 - コードクローンの利用過程における不具合が混入する原因として, コードクローンを利用する開発者を考慮した分析を補助するツールを作成



ツール概要

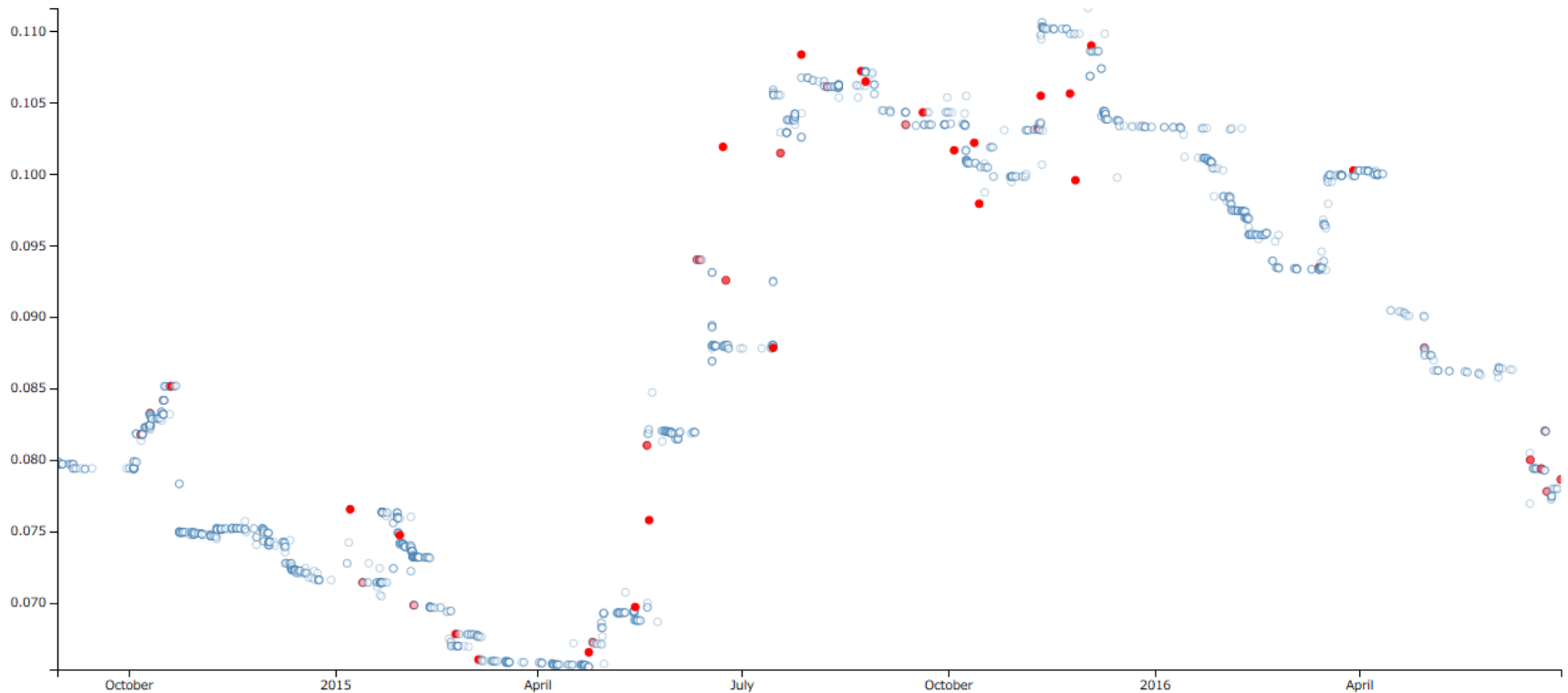
- 時系列表示
 - コードクローンの利用過程を俯瞰的に認識
- ファイルリスト表示
 - 問題があると考えられる箇所の変更の差分を確認

時系列表示

- x軸: 時間
- y軸: クローン密度
 - 各リビジョンの全コード行数にクローンがどれだけ含まれているか
 - $\text{クローン密度} = \frac{\text{全クローン行数}}{\text{全コード行数}}$
- 各ノード: コミット
 - 赤: コードクローンと不具合を結びつけたコミット

時系列表示

- x軸: 時間
- y軸: クローン密度
- 各ノード: コミット
 - 赤: コードクローンと不具合を紐づけしたコミット



ファイルリスト表示

- commitKeyId: CCTで作成したデータベース上のコミットID
- commitId: コミットID
- committer: コミットした開発者
- cloneStatus: コードクローンに対する変更
- filePath: ファイルパス

commitKeyId	commitId	committer	cloneStatus	filePath	
4123	0e0949d8458d3cbe48ecb2d62a0234697c475020	Ben Christensen	created	src/main/java/rx/Observable.java	src/main/java/rx/Single.java
4124	32179c631d18e120b4d8e70984cb032a4b551c63	Ben Christensen	changed	src/main/java/rx/Observable.java	src/main/java/rx/Single.java
4130	a40a4130edcdadc2340da5787ca30353018173f2	David Karnok	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java
4132	0aa8e92c19a3379f2fd82238766ee95eb68e8f44	David Karnok	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java
4133	26ea1e127799c104e7412c0ecb77a422f1b36f65	David Karnok	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java
4134	67f7988154ce61b5004784d654632045db7a9db1	David Karnok	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java
4135	0caeea85f2b311822d2b7f5fd7695055f4da9397	David Karnok	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java
4137	488fe1cb4e61132f40bc3ec715e81bb0b87a5b98	David Karnok	changed	src/main/java/rx/Observable.java	src/main/java/rx/Single.java
4139	d2a5d5db60c2d4687f0eb45362d80f03cdedb14d	David Karnok	changed	src/main/java/rx/Observable.java	src/main/java/rx/Single.java
4230	fc0c0d734fe0daab62d12b74459f79c7f1fb3187	Aaron Tull	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java
4231	97747fb6abddcbd554e2158b352b3e1fef25725	George Campbell	changed	src/main/java/rx/Single.java	src/main/java/rx/Observable.java

評価実験

- コードクローンの利用過程における不具合混入原因の分析を補助できたかを示す
 - RQ1: クローンの利用過程を把握できるか
 - RQ1.1: クローンの増減を確認できるか
 - RQ1.2: どのような開発者がクローンを使用しているか
 - RQ2: 不具合混入原因を分析できるか
 - RQ2.1: コードクローンの変更内容から不具合混入原因を考えられるか
 - RQ2.2: 開発者の影響から不具合混入原因を考えられるか

評価方法

- 本ツールとExcelで計11問の正答数, 回答時間を比較
- 回答者(学部生3人, 大学院生2人)
 - 本ツール: 3人, Excel: 2人
- 対象プロジェクト: RxJava

	対象リビジョン数	対象期間
Excel	2,628	2012/03/18～ 2014/04/30
本ツール	2,630	2014/04/30～ 2018/01/19

設定した質問 RQ1.1とRQ1.2

- RQ1.1: クローンの増減を確認できるか
 - Q1とQ2
 - 全体の中から, コードクローンに変化のある, 特定の期間を把握できるか
- RQ1.2: どのような開発者がクローンを使用しているか
 - Q3とQ4
 - 特定の開発者に注目し, その開発者がコミットしている期間を把握できるか

設定した質問 RQ2.1とRQ2.2

- RQ2.1:コードクローンの変更内容から不具合混入原因を考えられるか
 - Q5とQ6
 - 特定期間におけるコードクローンの変更内容から不具合混入原因を分析する補助ができるか
- RQ2.2:開発者の影響から不具合混入原因を考えられるか
 - Q7～Q11
 - 開発者の影響も考慮して, 不具合混入原因を分析する補助ができるか

実験結果 全回答

	本ツール	Excel		本ツール	Excel	差分
Q1	2 (67%)	1 (50%)	Q1	02:42.3	04:21.4	-01:39.1
Q2	0 (50%)	1 (50%)	Q2	06:12.5	13:47.2	-07:34.7
Q3	3 (100%)	2 (100%)	Q3	03:54.8	05:05.9	-01:11.1
Q4	0 (0%)	2 (100%)	Q4	03:24.5	01:45.3	+01:39.2
Q5	3 (100%)	1 (50%)	Q5	00:08.0	07:20.1	-07:12.1
Q6	0 (0%)	1 (50%)	Q6	04:10.0	13:09.4	-08:59.4
Q7	3 (100%)	0 (0%)	Q7	00:51.7	19:25.1	-18:33.4
Q8	1 (33%)	0 (0%)	Q8	03:08.6	10:04.3	-06:55.7
Q9	2 (67%)	1 (50%)	Q9	03:09.7	12:37.4	-09:27.7
Q10	0 (0%)	0 (0%)	Q10	05:41.9	02:05.3	+03:36.6
Q11	0 (0%)	0 (0%)	Q11	01:26.9	02:29.9	-01:03.0
正答率	42.4%	40.9%	平均	03:10.1	08:22.9	-05:12.8

実験結果 全回答

	本ツール	Excel		本ツール	Excel	差分
Q1	2 (67%)	1 (50%)	Q1	02:42.3	04:21.4	-01:39.1
Q2	0 (50%)	1 (50%)	Q2	06:12.5	13:47.2	-07:34.7
Q3	3 (100%)	2 (100%)	Q3	03:54.8	05:05.9	-01:11.1
Q4	0 (0%)	2 (100%)	Q4	03:24.5	01:45.3	+01:39.2
Q5	3 (100%)	1 (50%)	Q5	00:08.0	07:20.1	-07:12.1
Q6	0 (0%)	1 (50%)	Q6	04:10.0	13:09.4	-08:59.4
Q7	3 (100%)	0 (0%)	Q7	00:51.7	19:25.1	-18:33.4
Q8	1 (33%)	0 (0%)	Q8	03:08.6	10:04.3	-06:55.7
Q9	2 (67%)	1 (50%)	Q9	03:09.7	12:37.4	-09:27.7
Q10	0 (0%)	0 (0%)	Q10	05:41.9	02:05.3	+03:36.6
Q11	0 (0%)	0 (0%)	Q11	01:26.9	02:29.9	-01:03.0
正答率	42.4%	40.9%	平均	03:10.1	08:22.9	-05:12.8

回答に関する考察

- Q1, Q3, Q5, Q7, Q9
 - 全体の中から特定の期間・開発者を発見できるか
- 利用過程を俯瞰的に認識し, 問題を含むだろう期間・開発者の発見に有効

	本ツール	Excel
Q1	2 (67%)	1 (50%)
Q3	3 (100%)	2 (100%)
Q5	3 (100%)	1 (50%)
Q7	3 (100%)	0 (0%)
Q9	2 (67%)	1 (50%)

	本ツール	Excel	差分
Q1	02:42.3	04:21.4	-01:39.1
Q3	03:54.8	05:05.9	-01:11.1
Q5	00:08.0	07:20.1	-07:12.1
Q7	00:51.7	19:25.1	-18:33.4
Q9	03:09.7	12:37.4	-09:27.7

ツールの使用性に関する考察

- 11問の平均から
 - 1.5%正答率が高い
 - 5分12秒早く回答できた
- Excelを使う場合: 使う人のスキルが必要
- 本ツールの場合: 使う人のスキルに関係なく, 補助ができた

	本ツール	Excel
正答率	42.4%	40.9%

	本ツール	Excel	差分
平均	03:10.1	08:22.9	-05:12.8