

Supporting Clone Analysis with Tag Cloud Visualization

Manamu Sano

Eunjong Choi

Norihiro Yoshida

Yuki Yamanaka

Katsuro Inoue

Proceedings of the International Workshop on Innovative Software Development
Methodologies and Practices (InnoSWDev 2014), pp.94-99, Hong Kong, China, November
2014

コードクローンの研究

- 大規模なソースコードにおけるコードクローンの検出に関する技術[16,18]
- コードクローンの管理(クローンのリファクタリング)へと関心が移っている[6]
- 検出されたクローンを管理する際、開発者が手動ですべてをチェックするのは難しい

[6] R. K. C.K. Roy, M. F. Zibran. The vision of software clone management: Past, present and future. In Proc. of IEEE CSMR/WCRE, pages 18–33, 2014.

[16] L. Jiang, G. Mishnerghi, Z. Su, and S. Glondou. DECKARD: scalable and accurate tree-based detection of code clones. In Proc. of ICSE, pages 96–105, 2007.

[18] T. Kamiya, S. Kusumoto, and K. Inoue. CCFinder: a multilinguistic token-based code clone detection system for large scale source code. IEEE Trans. Softw. Eng., 28(7):654–670, 2002.

既存の研究

- 多数のコードクローンを含む部分の把握のために、いくつかの可視化技術の開発がされている[19][24][17]
 - 散布図[7][9][27]
 - ファイルまたはディレクトリにコードクローンが存在することを簡単に知ることができる

[7] J. R. Cordy. Live scatterplots. In Proc. of IWSC, pages 79–80, 2011.

[9] S. Ducasse, M. Rieger, and S. Demeyer. A language independent approach for detecting duplicated code. In Proc. of ICSM, pages 109–118, 1999.

[17] E. Juergens. Why and How to Control Cloning in Software Artifacts. PhD thesis, Technische Universität München, 2011.

[19] C. Kapser and M. Godfrey. Improved tool support for the investigation of duplication in software. In Proc. of ICSM, pages 305–314, 2005.

[24] M. Rieger, S. Ducasse, and M. Lanza. Insights into system-wide code duplication. In Proc. of WCRE, pages 100–109, 2004.

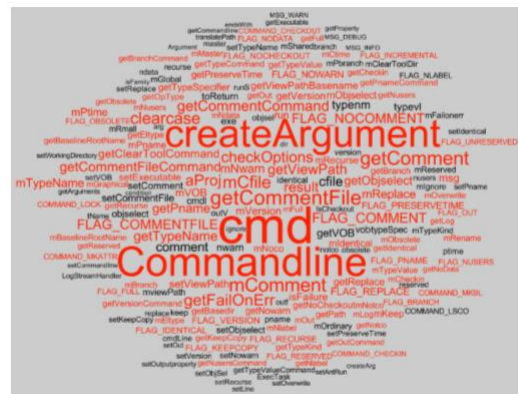
[27] Y. Ueda, T. Kamiya, S. Kusumoto, and K. Inoue. Gemini: Maintenance support environment based on code clone analysis. In Proc. of METRICS, pages 67–76, 2002.

散布図の問題点

- コードクローンの位置情報のみを提供
 - コードクローンがソースコードの一部に集中する理由を開発者が理解するのは困難
- コードクローンが存在する理由を理解するために、コードクローンを含むソースファイルをチェックする必要がある

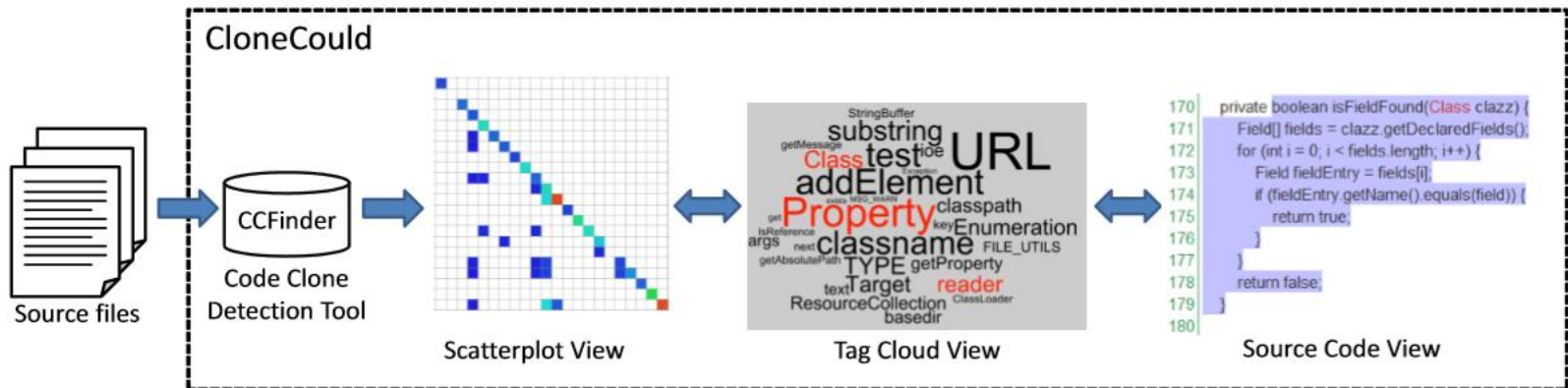
今回の研究

- タグクラウドの可視化に基づく、コードクローンの理解をサポートするコードクローン解析ツールCloneCloud
- タグクラウド
 - 与えられたデータの効率的な理解と情報検索のためのキーワード
 - 検出されたコードクローンの識別子

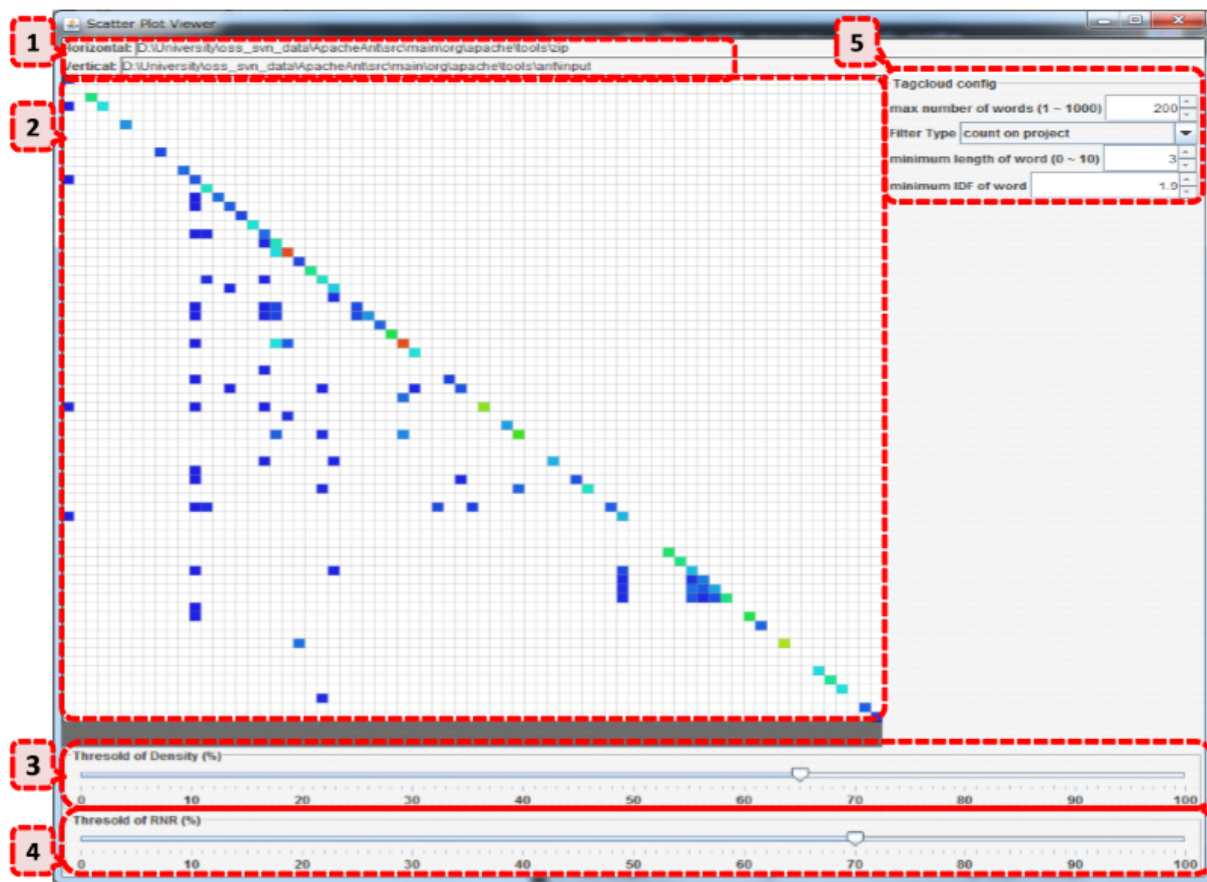


CloneCloudの主要なView

- 散布図
- タグクラウド
- ソースコード

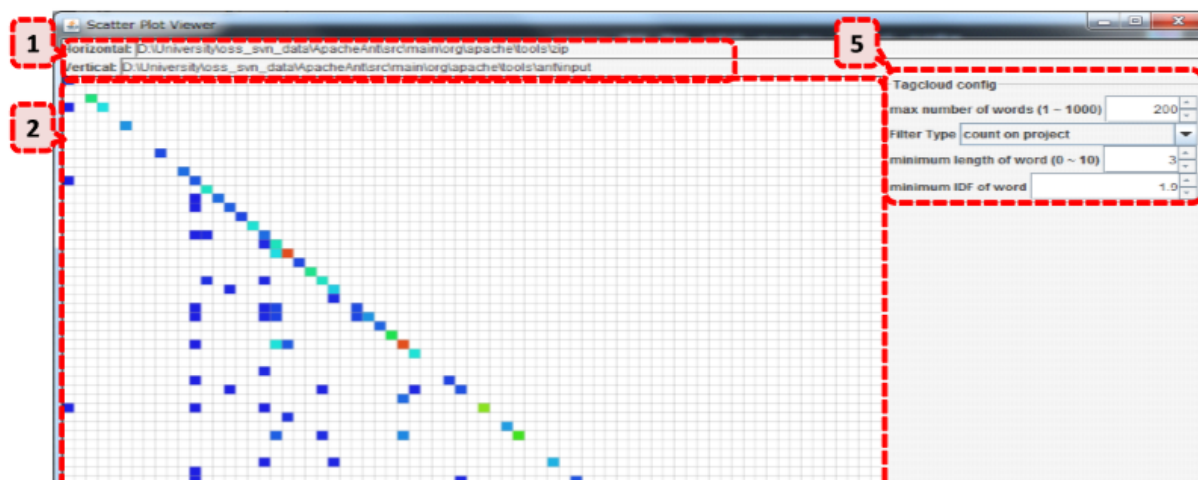


散布図



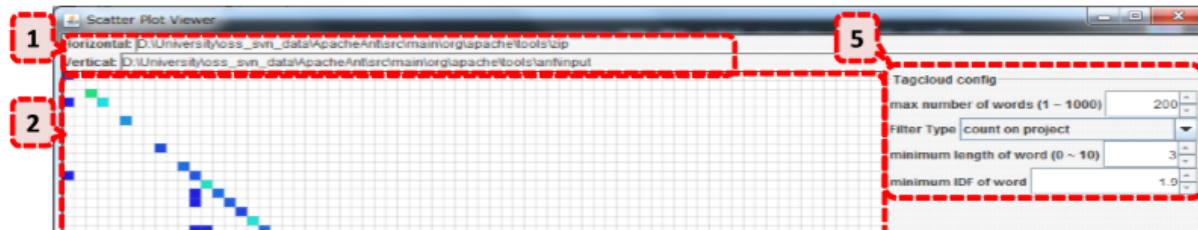
(a) Scatterplot View

散布図



- ラベル1: 縦軸、横軸ディレクトリの絶対パス
- ラベル2: 青色ほどクローン密度が低く、赤色ほど高い

散布図

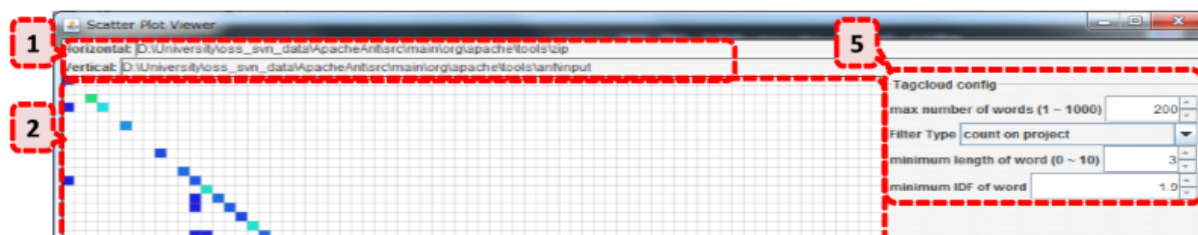


- ラベル3: クローン密度を設定
- ラベル4: RNR[13]を設定
 - 調査の必要がないコードクローンをフィルタリングする



(a) Scatterplot View

散布図



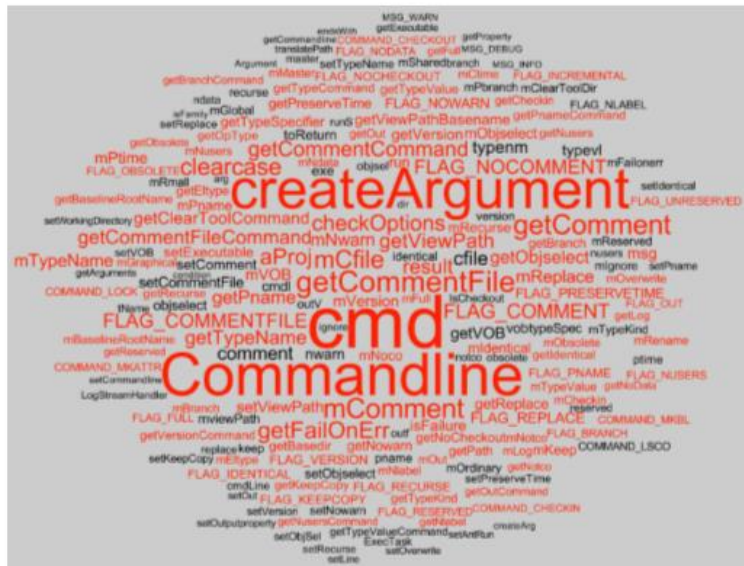
- ラベル5: タグクラウドのカスタマイズオプション
 - 表示される識別子の最大数
 - 識別子のフォントのサイズを決定する基準
 - 識別子の長さの最小値
 - 識別子の最小IDF値



(a) Scatterplot View

タグクラウド

- 散布図で選択されたディレクトリ内の識別子を表示
 - 黒のフォント: ソースコードのみが含む
 - 赤のフォント: コードクローンが含む
 - フォントのサイズ: 大きいほど頻繁に出現する



(b) Tag Cloud View

ソースコード

- 選択した識別子を含むコードクローンのソースコードを表示
 - 赤のフォント: 選択した識別子
 - 緑のフォント: 他の識別子

The screenshot displays an IDE interface with two panes. The left pane shows a list of clones containing the identifier 'createArgument'. The right pane shows the source code for the method 'getCommentFileCommand' with red and green font highlighting.

File	Start	End
CCCheckin.j...	282.13	291.10
CCCheckin.j...	300.13	309.10
CCCheckout...	407.13	416.10
CCCheckout...	425.13	434.10
CCCheckout...	444.13	453.10
CCCheckout...	462.13	471.10
CCMkattr.java	319.13	328.10
CCMkattr.java	337.13	346.10
CCMkattr.java	355.13	364.10
CCMkbl.java	304.13	313.10
CCMkbl.java	322.13	331.10
CCMkdir.java	190.13	199.10
CCMkdir.java	208.13	217.10
CCMkdir.java	229.13	238.10

File	Start	End
CCCheckin.j...	282.13	291.10
CCCheckin.j...	300.13	309.10
CCCheckout...	407.13	416.10
CCCheckout...	425.13	434.10
CCCheckout...	444.13	453.10
CCMkattr.java	319.13	328.10
CCMkattr.java	337.13	346.10
CCMkattr.java	355.13	364.10
CCMkbl.java	304.13	313.10
CCMkbl.java	322.13	331.10

```
462 private void getCommentFileCommand(Commandline cmd) {  
463     if (getCommentFile() != null) {  
464         /* Had to make two separate commands here because if a space is  
465            inserted between the flag and the value, it is treated as a  
466            Windows filename with a space and it is enclosed in double  
467            quotes ("). This breaks clearcase.  
468         */  
469         cmd.createArgument().setValue(FLAG_COMMENTFILE);  
470         cmd.createArgument().setValue(getCommentFile());  
471     }  
472 }
```

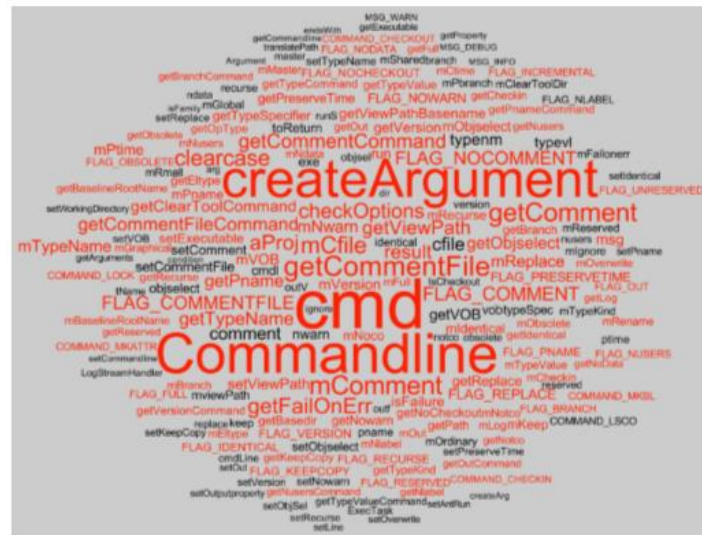


```
300 private void getCommentFileCommand(Commandline cmd) {  
301     if (getCommentFile() != null) {  
302         /* Had to make two separate commands here because if a space is  
303            inserted between the flag and the value, it is treated as a  
304            Windows filename with a space and it is enclosed in double  
305            quotes ("). This breaks clearcase.  
306         */  
307         cmd.createArgument().setValue(FLAG_COMMENTFILE);  
308         cmd.createArgument().setValue(getCommentFile());  
309     }  
310 }
```

(c) Source Code View

まとめ

- コードクローン解析ツールCloneCloud
 - 検出したコードクローンの識別子からタグクラウドを生成
 - フォントのサイズや色から、こういったコードクローンなのかを直観的に理解・分析するサポートを行う



所感

- フォントのサイズや色を使うことで、必要な識別子ははっきりわかるようになっている
- オプションを付けているのが面白いと感じた
 - 自分も付けてみたらいい