様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 24 年　6 月　12 日現在

機関番号：14603
研究種目：研究スタート支援
研究期間：2010～2011
課題番号：22800040
研究課題名（和文）自然言語処理を応用したコードクローン検出手法

研究課題名（英文）　A clone detection method using natural language processing

研究代表者
吉田　則裕（YOSHIDA NORIHIRO）
奈良先端科学技術大学院大学・情報科学研究科・助教
研究者番号：00582545

研究成果の概要（和文）：本研究では，変数名等の識別子名の類似性の基づくコードクローン検出ツールを実現した．コードクローンとは，ソースコード中の一部分(コード片)のうち，他のコード片と類似して いるものを指す．類似した識別子名の特定には，自然言語処理で用いられている類義語特定手法を用いた．本ツールの評価では，検出したコードクローンが同時に修正される頻度や，同一の欠陥を含む頻度などを調査することで，有用性を確認した．

研究成果の概要（英文）：In this study, we propose a code clone detection tool based on the similarity of identifiers such as variable names. Code clone is a code fragment that has similar code fragment to it in the source code. To identify similar identifier names, we use synonym identification method proposed in natural language processing. As the evaluation of the tool, we confirm the usefulness by the investigation of the frequencies that simultaneous modification is occurred in detected code clones and detected code clones has the same defect.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2010 年度	1,250,000	375,000	1,625,000
2011 年度	1,140,000	342,000	1,482,000
年度			
年度			
年度			
総　計	2,390,000	717,000	310,7000

研究分野：ソフトウェア工学
科研費の分科・細目：情報学・ソフトウェア
キーワード：コードクローン，ソフトウェア保守，自然言語処理，

１．研究開始当初の背景

コンピュータシステムの用途が多様化した現代社会において，ソフトウェアが担 う 社会的役割 は極めて大きいと言える．ソフトウェアには高い信頼性が求められる一方で，その開発にかけられる時間や投入できる人的．計算機資源は限られている．そこで，信頼性の高いソフトウェアを効率的に開発する方法の実現を目指した研究が盛んに行われている．このような研究分野は，ソフトウェア工学と呼ばれる．
ソフトウェア開発にかかるコストを増大させている要因の 1 つとして，ソースコード

中のコードクローンが挙げられる．コードクローンとは，ソースコードの一部分(コード片)のうち，他のコード片と類似したものを指し，コピーアンドペースト等により作成される．あるコードを修正するとその全てのコードクローンを見つけ出し，修正を行う必要が生じることがある（ 図 １ ）．特に，ソースコード中に欠陥が見つかった場合には，その欠陥を含むコード片のコードクローンを探し，検査する必要がある．しかし，ソースコード中のコードクローンを人手で探すためには大きな労力が必 要 と なる．特に，数千万行からなる大規模ソースコードが対象の場合，全てのコードクローンを人手で探すことはより困難となる．

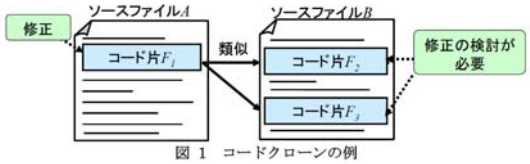

図 1　コードクローンの例

そこで，数多くの研究者がソースコード中のコードクローンを自動的に検出するツールの開発を行ってきた．これらコードクローン検出ツールは，ソースコードをツール毎に決められているプログラム表現（例えば，構文木）に変換し，そのプログラム表現上において，等価な部分をコードクローンとして検出する．

これらコードクローン検出ツールに共通していることは，コード片に含まれる識別子名（変数名など）が類似していても，ツール毎のプログラム表現において，差異が大きい場合はコードクローンとして検出しないことである．しかし，類似した識別子名を含むコード片は，類似した機能を持つことが多いため，同時にバグ修正や機能追加を行うことが多い．そのため，類似した識別子名を含む　コード片をコードクローンとして検出すべきであると考えた．

２．研究の目的

本研究の目的は，識別子名の類似性に基づくコードクローン検出ツールを開発し，有効性の評価を行うことである．

本ツールは，単純に同一の識別子名を含むコード片をコードクローンとして検出するだけではなく，類似した用いられ方をしている識別子名を含むコード片もコードクローンとして検出する．その理由は．類似した用いられ方をする識別子であっても．開発者によって異なる名前が付けられることがあるため，類似した用いられ方をしている識別子名を等価であると見なした方が，類似した機能を持つコード片をコードクローンとして検出しやすいと考えられるからである．類似 した用いられ方をしている識別子名の特定は，自然言語処理の分野で提案されている類義語特定 手法を用いる．

有効性の評価では,以下のことを明らかにする．

(1) 本ツールがコードクローンとして検出したコード片が，既存のコードクローン検出ツールで検出されるか.

(2) 本ツールが検出したクローンセット(コードクローンの同 値 類)を提示したとき，保守作業を行う開発者を支援できるか．具体的には，それらクローンセットは，その後の保守作業において同時に修正されるか，もしくは同一の欠陥を含んでいるか.

(3) 既存のコードクローン検出ツールと比較して，本ツールのスケーラビリティ（検出時間やメモリ消費量）は高いか.

３．研究の方法

まず，識別子名の類似性に基づくコードクローン検出ツールの実装を行った．本ツールの実装方法の中 から，「ソースコードに含まれる類義語の特定」とそれを用いた「コード片間の等価性判定」について述べる．また,「本ツールの実装を行うための情報収集」について述べる.

各ソースファイルに含まれる語(基本的には識別子名.ただし,複数の語からなる識別子名は各語に分割)を抽出し,類義語(類似した用いられ方をしている語)の特定を行う．このためには，自然言語の類義語辞書をそのまま用いるという方法が考えられるが，これには２つの問題点がある.

・ 自然言語における類義語は，プログラミング言語における類義語と異なる可能性がある.

・ 特定分野のソフトウェアの開発では，分野固有の類義語が存在すると考えられ，そのような類義語は既存の類義語辞書には掲載されていないと考えられる.

そこで，類義語辞書を用いず,自然文書中から類義語を自動的に抽出する手法を実装した.このような手法は，語を共起関係に基づいてクラスタリングすることで類義語の特定を実現する．クラスタリング後，同一のクラスタに含まれた語が類義語として特定さ

れる.

コード片間の等価性を判定するため，構文木中の部分木に対して特徴ベクトルを付加した．特徴ベクトルは，部分木に含まれる語の分布を表し，類義語は同一の語として扱った．次に，特徴ベ クトル間の距離に基づいて部分木のクラスタリングを行った．最後に，同一クラスタに属した部分木に対応するコード片を等価なコード片として提示した．各部分木をベクトルで表現する利点は，部分木間の距離を求める問題をベクトル間の距離を求める問題に変換することで，計算量を低下させることができることである．

特徴ベクトルの数が膨大になると，クラスタリングにかかる計算時間が大きくなると考えられるため,LSH (Locality Sensitive Hashing)というハッシュ関数を用いたクラスタリングを行った．LSH は，「類似したベクトルは高い可能性で同じハッシュ値になり，異なるベクトルは高い可能性で異なるハッシュ値になる」という性質を持つハッシュ関数である．LSH はランダム性を持つため,適用するたびに値が異なるが，非常に高速であることから多くの分野でクラスタリングに用いられている．

４．研究成果

提案手法の評価として,「既存のコードクローン検出ツールと比較して,本ツール のスケーラビリティ(検出時間やメモリ消費量)は高いか」についても評価を行った．入手可能であり，かつ代表的なコードクローン検出ツールである DECKARD やCloneDR との比較を行った.各ツー ルともに,検出するコードローンの量をパラメータで変更できるようになっているため,パラメータを変更しながら比較を行った.プロジェクトの規模を変えながら検出時間を計測した結果を下図に示す.

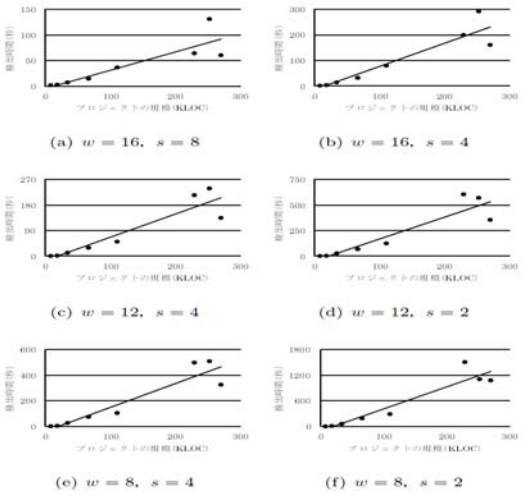

図２　本手法の検出速度

本ツールの検出時間は，ソースコードの規模に対して線形に増加している．通常，意味を考慮したコードクローン検出手法は，ソースコードの規模に対して二乗のオーダの検出時間を要するため，本ツールのスケーラビリティは高いと考えられる．

また，他のツールとの比較実験の結果を下表に示す.

	javadoc	ant	jdk1.5.0	swing
提案手法(w=16,s=8)	2秒	3秒	7秒	15秒
提案手法(w=16,s=4)	2秒	4秒	15秒	32秒
提案手法(w=12,s=6)	1秒	3秒	12秒	30秒
提案手法(w=12,s=3)	3秒	6秒	29秒	71秒
提案手法(w=8,s=4)	3秒	6秒	28秒	77秒
提案手法(w=8,s=2)	4秒	14秒	62秒	194秒
既存の手法	34秒	2283秒	39秒	142秒

表１　既存手法との比較実験の結果

今後，「本ツールがコードクローンとして検出したコード片が,既存のコード クローン検出ツールで検出されるか」を評価する.

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計３件）

① 政井智雄，吉田則裕，松下誠，井上克郎，″テンプレートメソッドの形成に基づく類似メソッドの集約支援″,ソフトウェア工学の基礎 XVII，125-130，2010，査読有
② 山本哲男，吉田則裕，肥後芳樹，″ソースコードコーパスを利用したシームレスなソースコード再利用手法″，情報処理学会論文誌，53 巻，2 号，644-652，2012，査読有
③ 神谷年洋，肥後芳樹，吉田則裕，″コードクローン検出技術の展開″，コンピュータソフトウェア，28 巻，3 号，29-42，2011，査読有

〔学会発表〕（計２０件）

① Norihiro Yoshida，″Detection of Chained Clone and Its Application″，9th CREST Open Workshop，2010.11.23，London，UK
② 山本哲男，吉田則裕，肥後芳樹，“ソースコードコーパスを利用したシームレスな再利用支援”，電子情報通信学会 ソフトウェアサイエンス研究集会 2010 年 12 月研究集会，2010.12.25，群馬
③ Yu Kashima, Yasuhiro Hayase, Norihiro Yoshida, Yuki Manabe, Katsuro Inoue, “A Preliminary Study on Impact of Software

Licenses on Copy-and-Paste Reuse”, International Workshop on Empirical Software Engineering in Practice 2010, 2010.12.7, 奈良
④ 齋藤晃, 吉田則裕, 松下誠, 井上克郎, “コードの生存期間を考慮したコードクローンと欠陥修正の関係調査”, 電子情報通信学会 ソフトウェアサイエンス研究集会 2010 年 10 月研究集会, 2010.10.14, 岩手
⑤ 鹿島悠, 早瀬康裕, 吉田則裕, 真鍋雄貴, 井上克郎, “ソフトウェアライセンスがコピーアンドペーストによる再利用に与える影響の調査” 電子情報通信学会 ソフトウェアサイエンス研究集会 2010 年 10 月研究集会, 2010.10.14, 岩手
⑥ Norihiro Yoshida, Masataka Kinoshita, Hajimu Iida, “A Cohesion Metric Approach to Dividing Source Code into Functional Segments to Improve Maintainability”, 16th European Conference on Software Maintenance and Reengineering, 2011.3.29, Szeged, Hungary.
⑦ Shunsuke Yoshioka, Norihiro Yoshida, Kyohei Fushida, Hajimu Iida, “Scalable Detection of Semantic Clones Based on Two-stage Clustering”, IEEE 22nd International Symposium on Software Reliability Engineering, 2011.11.29, 広島
⑧ Masakazu Ioka, Norihiro Yoshida, Tomoo Masai, Yoshiki Higo, Katsuro Inoue, “A Tool Support to Merge Similar Methods with a Cohesion Metric COB”, 3rd International Workshop on Empirical Software Engineering in Practice, 2011.11.1, 奈良
⑨ Yu Kashima, Yasuhiro Hayase, Norihiro Yoshida, Yuki Manabe, Katsuro Inoue, “An Investigation into the Impact of Software Licenses on Copy-and-Paste Reuse among OSS Projects”, 18th Working Conference on Reverse Engineering, 2011.10.16, Limerick, Ireland
⑩ Masayuki Tokunaga, Norihiro Yoshida, Kazuki Yoshioka, Makoto Matsushita, Katsuro Inoue, “Towards Collection of Refactoring Patterns Based on Code Clone Classification”, 2nd Asian Conference on Pattern Languages of Programs, 2011.10.7, 東京
⑪ Eunjong Choi, Norihiro Yoshida, Takashi Ishio, Katsuro Inoue, Tateki Sano, “Extracting Code Clones for Refactoring Using Combinations of Clone Metrics”, 5th International Workshop on Software Clones, 2011.5.23, Waikiki, HI, USA
⑫ 吉岡俊輔, 吉田則裕, 伏田享平, 飯田元, “近傍ハッシュ法を用いた 2 段階のクラスタリングに基づく Near-miss クローンの検出”, 電子情報通信学会 ソフトウェアサイエンス研究集会 2012 年 3 月研究集会, 2012.3.13, 沖縄
⑬ 伏田享平, 玉田春昭, 井垣宏, 藤原賢二, 吉田則裕, “プログラミング演習における初学者を対象としたコーディング傾向の分析”, 電子情報通信学会 ソフトウェアサイエンス研究集会 2012 年 3 月研究集会, 2012.3.13, 沖縄
⑭ 吉田則裕, 木下正喬, 飯田元, “プログラム理解のための凝集度に基づく機能候補抽出”, 日本ソフトウェア科学会第 28 回大会 2011.9.29, 沖縄
⑮ 藤原賢二, 伏田享平, 吉田則裕, 飯田元, “オープンソースソフトウェアを対象としたリファクタリングが欠陥混入に与える影響の調査”, 日本ソフトウェア科学会第 28 回大会, 2011.9.29, 沖縄
⑯ 山本哲男, 吉田則裕, 肥後芳樹, “ソースコードコーパスを利用したシームレスな再利用手法”, ソフトウェアエンジニアシンポジウム 2011, 2011.9.13, 東京
⑰ 井岡正和, 吉田則裕, 政井智雄, 井上克郎, “凝集度メトリクス COB を用いた Template Method パターン適用候補の順位付け手法”, 電子情報通信学会 ソフトウェアサイエンス研究集会 2012 年 7 月研究集会, 2011.7.30, 北海道
⑱ 吉岡一樹, 吉田則裕, 徳永将之, 松下誠, 井上克郎, “コードクローンの特徴に基づくメソッド引き上げリファクタリングパターンの提案”, 第 173 回ソフトウェア工学研究発表会, 2011.7.21, 岡山
⑲ Eunjong Choi, Norihiro Yoshida, Takashi Ishio, Katsuro Inoue, Tateki Sano, “Finding Code Clones for Refactoring with Clone Metrics: A Case Study of Open Source Software”, 電子情報通信学会 ソフトウェアサイエンス研究集会 2012 年 6 月研究集会, 2011.7.1, Seoul, South Korea
⑳ Kenji Fujiwara, Kyohei Fushida, Norihiro Yoshida, Hajimu Iida, “An Approach to Investigating How a Lack of Software Refactoring Effects Defect Density”, 電子情報通信学会 ソフトウェアサイエンス研究集会 2012 年 6 月研究集会, 2011.7.1, Seoul, South Korea

６．研究組織
(1)研究代表者
吉田 則裕 (Norihiro Yoshida)
奈良先端科学技術大学院大学・情報科学研究科・助教
研究者番号：00582545