様式 C-19

科学研究費補助金研究成果報告書

平成　22 年　5 月　14 日現在

研究種目：若手研究(B)
研究期間：2008　～　2009
課題番号：20700024
研究課題名（和文）　超大規模ソフトウェアを対象としたコードクローン分析基盤環境

研究課題名（英文）　Code-Clone Analysis Environment for Huge Software Repositories

研究代表者
松下 誠 (Makoto Matsushita)
大阪大学・大学院情報科学研究科・准教授
研究者番号：60304028

研究成果の概要（和文）：本研究では、超大規模なソフトウェアを対象としたコードクローン分析を行うことを目的として、コードクローンを作成した開発者の情報を収集することによるクローン分析手法や、コードクローン分析手法を UML で記述された設計文書に対して適用することによる設計の再利用手法などを実現する、コードクローン分析のための基盤環境について研究を行い、ツールの実装を通じて手法の有効性を確認した。

研究成果の概要（英文）：In this research、 we propose several methodologies on detecting huge software repositories、 including a method that employs software developer analysis、 and a method for detecting reusable design documents。 Through experiments with prototype tools、 we confirmed on its effectiveness。

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2008 年度	1,600,000	480,000	2,080,000
2009 年度	1,700,000	510,000	2,210,000
年度			
年度			
年度			
総　計	3,300,000	990,000	4,290,000

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、リポジトリ分析

１．研究開始当初の背景

ソフトウェアに対する分析手法として、近年、ソースコード中の重複した部分を発見するコードクローン検出技術が注目されてきている。コードクローンの分析を行うことにより、単にソースコードを共有している部分が発見できるだけでなく、ソフトウェアの再利用の程度や、あるいはソースコードの盗用といったソフトウェアのさまざまな側面を見ることが出来る。しかしながら、コードクローン分析は、特に対象となるソースコードが超大規模になった場合、アルゴリズムを工夫しても単一計算機上で分析することは非常に困難であり、そのた

めこれまで単一の大規模システム程度にしか分析されてこなかった。

研究代表者はこれまで、コードクローンや大規模ソフトウェアに対する検索技術等、ソフトウェア工学の各領域において研究を行ってきた。たとえば、Java 言語を対象として、複数のソフトウェアの集合である十数万クラス規模の大規模ソースコードに対し、実用レベルで利用可能なソースコード検索システムに関する研究を行っており、Java クラス間の依存関係解析の結果を用いて利用頻度の高いクラスを優先した検索結果の表示を行うための手法を確立している。また、膨大なソフトウェアの中からコードクローンを発見するためのツールや、ツールが発見した結果を用いてコードクローンをその性格等によって分類するための環境、および分析するための手法を確立してきている。

一方、超分散型のプログラム実行環境として、グリッドコンピューティング等の環境が注目を集めている。この種の環境は、単一の処理を細かく分割し、大量の計算機等を用いて演算を行うことで、大量の計算を低いコストで実行することを目標としている。そこで、超分散型のプログラム実行環境をソフトウェア分析を行うための実行環境として用いることにより、さらに大規模なソースコードを対象としたコードクローン分析環境を構築することが可能ではないかと考えた。

２．研究の目的

本研究は、これまでの成果を踏まえ、より大規模なソースコード、具体的には、世界中で広く用いられている全てのソースコードを扱うことができる程の規模を対象としたソースコード分析基盤を構築することを目指している。また、構築したコードクローン分析基盤を運用することにより、世界中のソフトウェアがどのような性格を持っているのか、全体像を明らかにすることを目指す。また、コードクローン分析基盤を他のソースコード解析技術へ応用する可能性についても検討を行う。

そのために、超大規模なソフトウェアを対象として、コードクローンの分析を行うための分析基盤を構築し、オープンソースソフトウェア等のインターネット上に存在する多種多様なソフトウェアを対象として分析を行う。これにより、世の中に存在する全てのソフトウェアが、どの程度再利用を行い、あるいはソースコードの単純な流用を行っており、またそれらの結果と、一般的に言われているソフトウェアの品質がどのような関係となっているか、を明らかにすることを目標とする。

３．研究の方法

本研究ではまず、ソフトウェア分析を行うためのモデルについて検討を行う。大規模でかつ複数のソフトウェアを同時に取り扱うことを前提として、コードクローン分析を分散型で行うためのモデルの構築を行う。本モデルは、以下のような点に考慮して設計する。

- 入力となるソフトウェアの大きさには依存しないこと。

  本システムでは、数億行あるいは数 GB といった非常に大規模で、かつ複数のソフトウェアを分析対象とすることを目指す。

- 分析結果が容易に統合できること。

  本システムでは、分析する対象および内容を、細かな問題に分解し、分析を行った後その結果を結合する、といった一般的な方法を用いる。そのため、個別の分析時における入出力情報、および分析内容が分割可能なようにすることはもちろん、その結果を集めて１つの分析結果とすることが可能でなければならない。

- 可搬性が高いこと。

  分析システムが動作する環境や、分析内容は状況に応じて変化することが考えられる。そのため、本分析モデルでは、動作環境や分析内容に依存しない形でソースコードおよび分析内容を扱えるようにしなければならない。

次に、前述のモデルを用いることを前提とし、コードクローン分析を分散型ソフトウェア分析モデル上で行うためのアルゴリズムを決定する。既存のアルゴリズムは、単一のシステムが全てローカルに処理を行うことを前提としていたため、これを分散環境でも用いることが出来るように拡張する。

ソースコードの大きさが小さいときには、既存のコードクローン検出アルゴリズムを用いても十分高速であるため、まず入力となる大規模なソースコードを小さなソースコードの集合として捉え、それぞれを既存のアルゴリズムで分析し、その結果を統合して全体の結果とすることを考える。

以上の結果を踏まえ、特に分散型ソフトウェア分析モデルがどの程度の性能を持つかを確認するために、システムの試作を行う。分析システムは超大規模なソースコードを入力とし、コードクローンの分布状況を視覚的に表現するものとする。

４．研究成果

2 年間の研究期間の間、コードクローン分析基盤環境の構築を目指し研究を行った結果、以下のような成果を得た。

(１) 2008 年度の成果

ソフトウェアの保守性を悪化させる問題の１つとして、コードクローンの存在があげられる。しかし近年の研究では、コードクローンはすぐに消滅するものが多く、また意図してコードクローンとなるように作られているものもあり、ある時点で発見されたすべてのコードクローンが不適切とは限らないことがわかってきた。そのため、ある時点で存在するコードクローンを適切に管理することによって、問題のあるコードクローンを確実に発見することが重要であるとされている。しかしながら、問題を抱えているコードクローンを与えられたソースコードのみから発見することは困難である。

そこで、コードクローン分析と同時に、そのコードクローンを作成した開発者の情報を収集することによって、各開発者がどのようにコードクローンを作成、あるいは解消しているか、分析するための手法を提案した。また、オープンソースソフトウェアの１つえである PostgreSQL のソフトウェア開発を対象として、提案手法を用いた分析を行った。その結果、開発者の中には、有意にコードクローン数を増やす、あるいは減らす者がいることを、Kruskal-Wallis 検定を用いて定量的に示すことができた。本研究により、どの開発者が作成したかどうかを管理することにより、コードクローンの発生を予想することができると期待される。

(２) 2009 年度の成果

ソフトウェアの再利用を適切に行うことによって、ソフトウェア開発の生産性と成果物の信頼性の双方が向上するといわれている。再利用を行う対象は、大きく設計情報とソフトウェア部品に分けることが出来る。再利用コンポーネントやクラスライブラリなどにより、ソフトウェア部品の再利用は積極的に行われているが、設計情報を再利用する試みはあまり行われていない。設計情報の再利用を積極的に行うことを目的として、プロダクトライン開発と呼ばれる手法があるが、これは現在行われている開発の設計情報を、派生して行う開発に適用するものであり、過去すでに行われた設計情報を再利用することは想定されていない。一方、ソフトウェア部品の再利用を促進することを目指し、再利用可能な部品を開発中に適宜推薦することができる、部品の自動推薦手法に関する研究が行われてきている。

そこで設計情報の中でもソフトウェア設計の際によく用いられる UML 図に着目し、UML 図を再利用する際に用いることができる自動推薦ツールを作成した。本ツールは、既存 UML 図の作成ツールに UML 図の自動推薦機能を組み込むことによって実現しており、現在作成中の UML 図中に記述された文字情報を元に、それと類似した情報を持つ UML 図を既存の UML 図群を対象として検索を行い、検索結果を適宜別ウィンドウへ表示する。また、検索結果を選択することによって、現在編集中の UML 図へ、選択した UML 図あるいはその一部を取り込むことができる。本ツールを用いた実験を行った結果、全体の１割程度の図を記述した状態で検索を行うと、再利用可能な類似 UML 図を高い確率で推薦できることがわかった。これにより、UML 図を作成する際に既存の UML 図の再利用を行いやすくなると期待される。

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）

〔学会発表〕（計 4 件）

[1] 松下 誠、 早瀬 康裕、 松下 誠、 井上克郎: **"状況に応じた設計情報の再利用を支援する UML 図の自動推薦ツール"**、信学技報、vol。 109、 No.456, SS2009-54, pp.37-42 (2010 年 3 月 8 日, 鹿児島大学, 鹿児島, 日本).

[2] 松下 誠: **"ソフトウェア工学の新潮流（１）リポジトリマイニング"**, ソフトウェアエンジアリング最前線 2009, pp.21-24 (2009 年 9 月 9 日, 東京女子大学, 東京, 日

本).

[3] Yue Jia, David Binkley, Mark Harman, Jens Krinke, Makoto Matsushita: **"KClone: A Proposed Approach to Fast Precise Code Clone Detection"**, Proc. of Third International Conference on Software Clones (2009年3月24日, Fraunhofer IESE, Kaiserslautern, Germany).

[4] 東 誠, 肥後 芳樹, 早瀬 康裕, 松下 誠, 井上 克郎: **"コードクローンの複雑度メトリクスを用いた開発者の特徴分析"**, ソフトウェアエンジニアリング最前線 2008, pp.103-106 (2008年9月2日, 東洋大学, 東京都, 日本).

６．研究組織

(1)研究代表者
松下 誠 (Makoto Matsushita)
大阪大学・大学院情報科学研究科・准教授
研究者番号：60304028