

研究種目：基盤研究（A）  
 研究期間：2005～2008  
 課題番号：17200001  
 研究課題名（和文） コードクローンリポジトリを用いた  
 統合ソフトウェア保守支援環境の構築  
 研究課題名（英文） Integrated Software Maintenance System based on  
 Code-Clone Repositories  
 研究代表者  
 井上 克郎（INOUE KATSURO）  
 大阪大学・大学院情報科学研究科・教授  
 研究者番号：20168438

## 研究成果の概要：

本研究では、ソフトウェア中の一致した部分を示すコードクローンを用いることによって、ソフトウェア開発および保守の際に利用できる種々の分析手法や、その手法を用いた分析ツール等について研究を行った。また、研究成果であるツールを企業等の実際の開発現場に利用してもらうことにより、ツールの有用性が高いことを確認した。

## 交付額

(金額単位：円)

	直接経費	間接経費	合計
2005年度	14,800,000	4,440,000	19,240,000
2006年度	6,500,000	1,950,000	8,450,000
2007年度	6,500,000	1,950,000	8,450,000
2008年度	6,500,000	1,950,000	8,450,000
年度			
総計	34,300,000	10,290,000	44,590,000

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、ソフトウェアメトリクス、ソフトウェア保守、類似度

## 1. 研究開始当初の背景

近年、ソフトウェアライフサイクルにおける保守の重要性が高まってきている。ソフトウェアライフサイクルの全コストに占める保守コストの割合は90%以上になるという報告もあり、ソフトウェア保守作業の改善は開発現場において深刻な課題となっている。最近ではオープンシステムやインターネットの時代となり、保守対象となるソフトウェアの種類も増加してきている。それに伴い、新規にスクラッチから作成する開発ではなく、既存のソフトウェアに対する機能変更や

追加が主となる開発が増加してきている。従って、ソフトウェアの保守性の改善は、将来にわたって続く保守作業を考える上で重要な課題となっている。

保守作業を阻害する最も大きな原因の一つとして、コードクローンが指摘されている。コードクローンとは、プログラムテキスト中の同一、あるいは、類似したコードの断片を意味する。コードクローンは、「コピーとペースト」によるプログラミングや意図的に同一処理を繰り返して書くということにより、プログラム中に作り込まれる。例えば、機能変更をあるコード片に行う場合を考える。も

し、そのコード片のコードクローンが存在する場合には、それら全てのコードクローンを確認して、必要があれば全てのコードクローンに同様の変更を行う必要がある。大規模プログラムから効率良くコードクローンを検出し、それらを除去するための手法が必要とされている。

研究代表者らはこれまでにコードクローン検出技術に関する研究を行ってきた。さらに、試作したコードクローン検出ツール CCFinder を複数のソフトウェア開発組織に試用してもらい、実際の開発現場からフィードバックを受けている。結果として、実際のソフトウェア開発現場においては、検出したコードクローンの利用目的に応じた支援が求められていることがわかった。

## 2. 研究の目的

近年、既存ソフトウェアに対する機能変更や追加が主となる保守作業が増加してきており、ソフトウェアの保守性の改善は、将来にわたって続く保守作業を考える上で重要な課題となっている。

ソースコードに対して保守作業を行う際、大きな問題としてコードクローンの存在が挙げられる。コードクローンはソースコード中の同一あるいは類似したコード断片であるが、これらは、開発中にコピーペースト作業などによって発生する。

本研究では、コードクローンを効率良く検出し、それを視覚的にわかりやすく表現した上で、検出されたコードクローンに対する分析を容易に行うための環境を構築する。これらはソフトウェアの開発時および保守時に用いることを想定しており、作業の効率化やソフトウェアの見える化を推進することを目指す。

## 3. 研究の方法

本研究ではまず、既存のコードクローン検出ツール CCFinder の出力を、将来の分析のために扱いやすくするための基盤整備を行った。この成果を踏まえ、まずコードクローン分析ツール Aries を構築した。また、コードクローンから計算した種々のメトリクスを用いて分析を行うツール Gemini を構築した。

分析の基礎となる2つのツールによる成果を踏まえ、コードクローンの中からリファクタリング可能な部分を抽出するツール CCShaper を用いて、リファクタリングのコストについて考察を行った。また、ソースコードの開発履歴から、開発の各時点における

クローン分析を行い、その結果を視覚的に表現するツールを作成した。また、種々のツールを企業等で運用してもらうことにより、ツールの有用性について評価を行った。

さらに、従来のコードクローン分析では扱えなかった、わずかな不一致部分を含むコードクローンを検出するためのアルゴリズムについて考察を行い、CCFinder の出力結果からそのようなコードクローンを発見するためのツールを作成した。また、検出されたコードクローンのすべてが保守作業を阻害する要因ではない、という企業等での運用実績を踏まえ、事前に問題を抱えているソースコードを入力として、そのソースコードと類似する他のソースコードを検出することができるツールを作成した。

最後に、Gemini 等で用いていたメトリクスデータをソフトウェアの保守性を計測するためのメトリクスとして利用することにより、従来用いられてきたソフトウェアの複雑度メトリクスによる保守性の計測よりもより精度の高い結果が得られることを確認した。また、より荒いプログラムの同一性についての問題について取り組み、部品の利用関係やインターフェイスの類似性を用いたソフトウェアの類似判定について研究を行い、その成果を既存の部品検索システム SPARS/J へ組み込み、その有用性を確認した。

## 4. 研究成果

4年間の研究期間の間、コードクローンの検出および分析に関する研究を行い、以下のような成果を得た。

### (1) 平成17年度

ソフトウェアのソースコードは近年大規模になっており、また、実行時のパフォーマンスを稼ぐためにループを意図的に展開したコードクローンが作り込まれる場合があるなど、コードクローンの発生は避けられない問題となっている。そこで、コードクローン検出ツール CCFinder の出力結果を利用して、リファクタリングを目的としたコードクローンの分析を行うツールである Aries を構築した。Aries は対象となるプログラム言語の構文情報を利用することにより、プログラムとして意味のまとまりとなり得る部分のみを検出したクローンから選び出すことにより、クローン分析をリファクタリングに応用することが容易となった。また、保守作業時における分析作業を支援するため、種々のメトリクスデータを用いて分析するツール Gemini を構築した。Gemini では、コードクローンを対象とした種々のメトリクスをあらかじめ計算し、それをグラフとし

て表現することによって、視覚的にクローンの性質を容易に理解することが可能となった。また、Gemini を用いた分析手法について提案を行い、実例を通じてその有効性について検討を行った。

#### (2) 平成 18 年度

コードクローン検出ツール CCFinder の出力から、リファクタリング可能な条件を満たすコードクローンを抽出するツール CCShaper を用いて、コードクローンとなっている箇所をリファクタリングする場合のコストと効果について調査を行った。その結果、リファクタリングによりコードを対象としたメトリクス値の変化は少なかったものの、その後の保守作業を行う場合の修正回数を削減する効果があることがわかった。

また、CCFinder を含め、既存の検出ツール等はある時点のソースコードを対象とした分析しか行えていなかった。そこで、ソースコードを時系列順で解析し、コードクローンがどのように変化していったか、また、その変更が加えられた際どのような開発者が関係していたかを分析するためのツールを作成した。作成したツールを比較的大規模のオープンソースソフトウェア開発に適用し、実にどのような開発が行われ、それによってクローンがどのように変化していったかを追跡できることを確認した。

最後に、これまでの研究で作成したコードクローン分析のための種々のツール類を、企業等で実際にソフトウェア開発に携わっている方々に利用してもらい、既存のツールがどの程度有効であるかを確認し、得られたフィードバックを基にツールをより実用的となるよう改良した。また、改良した分析ツールを日本の主要ベンダー 5 社が共同開発したソフトウェアに対して適用することにより、クローン分析作業を的確にかつ容易に行うことができることを確認した。

#### (3) 平成 19 年度

既存のコードクローン検出手法の多くは、連続する一致部分を求めるアルゴリズムを採用している。そのため、連続する一致部分中にはギャップ（不一致部分）を含むことができず、従ってギャップドクローンを検出することができなかった。そこで、既存のコードクローン検出手法によって得られたコードクローンを入力とし、グラフマイニングアルゴリズムの 1 つである AGM アルゴリズムを用いることによって、ギャップを含むコードクローンを検出する手法について研究を行った。一般的に、ギャップを含むコードクローンを直接求める手法では計算量の爆発が起きやすいが、本手法を用いることによって高速にギャップを含むコードクローンを検

出することができるようになった。また、本手法を既存のコードクローン検出ツールである CCFinder の後処理システムとして実装し、複数のオープンソースソフトウェアに対して適用を行った。その結果、多数のギャップを含むコードクローンを検出することができた。

ソフトウェアの保守を行う際、コードクローンの存在が保守作業を困難にするとされているが、実際には検出され得るすべてのコードクローンがそのような要因となるわけではない。そこで、事前に何らかの問題を抱えていることがわかっているソースコードを入力として与え、その入力のコードクローンを検出するための手法について考察を行った。この時、単にソースコード上での文字の並びが一致するかどうかではなく、ソースコード中の識別子や構文要素といった、ソースコードの構造的な特徴を用いることにより、保守作業において問題となるコードクローンのみを検出できるようにした。

#### (4) 平成 20 年度

「コードクローンに基づく保守性の評価手法」については、コードクローンに関するメトリクスを用いて、フォールトを含むモジュールの予測を行い、保守性の評価に用いた。この研究では、従来用いられてきた複雑度メトリクスのほかに、コードクローンに関するメトリクス RNR (Ratio of Non-Repeated code) を加えた。そして、ロジスティック回帰分析により、フォールト含有モジュールを予測した。この手法に対して、実際のシステム開発データを適用した実験の結果、従来手法に比べて予測精度が向上することを確認した。

また、「コードクローン抽出による再利用ライブラリの構築手法」に関しては、再利用ライブラリに登録するソフトウェア部品間の同一性の問題に着目した。今までは、コードクローンによる判定やソフトウェアメトリクスに基づく判定を用いて同一ソフトウェア部品の判定を行ってきたが、小規模部品では、うまく判定できないなどの問題があった。そこで、部品の利用関係やインターフェイスの語句の類似性を分析し、同一コンポーネントの判定を行う手法を開発した。この手法は、以前の手法では誤判定した場合も正しく判定できるようになった。この手法を実現するサブシステムを実際に作成し、Java 部品検索システム SPARS/R に実装した。

#### (5) 総括

これらの一連の研究により、コードクローンを効率良く発見することはもちろん、その発見したクローンをその後の開発および保守に如何に役立てればよいか、といった、これまでに既存の研究が及ばなかった分野に

において多くの成果を得ることができた。また、単に成果を生むだけにとどまらず、企業等でその成果を利用してもらうことにより、より広い範囲でコードクローンを用いたソフトウェア分析を行えるようになった。

#### (6) 成果の国内外における位置づけおよびインパクト

本研究の成果は、国内外の論文誌、および国際会議や国内シンポジウム・研究会等で発表を行っている。本研究の4年間の研究期間を通じ、コードクローンに関する研究は広がりを見せていたが、われわれの研究グループは中でも主要な位置を占めており、その影響力は非常に大きいといえる。

その一例として、2009年3月に開催されたコードクローンに関する国際ワークショップ(The 3<sup>rd</sup> International Workshop on Software Clones, IWSC2009)において、本研究課題の研究代表者である井上はワークショップ主催者の一人として、当該ワークショップを成功に導いた。

また、われわれの研究グループは、2008年9月より、プログラム解析の研究では世界第一人者であるイギリス国 King's College London の Mark Harman 教授等の研究グループとコードクローン情報を用いた大規模ソフトウェアの解析に関する共同研究も始めており、本研究課題で得られたコードクローン検出・分析システムの研究成果を活用し、コードクローンに関する研究の更なる発展を目指している。

国内においては、本研究課題の成果として電子情報通信学会論文誌に発表した合計5編のコードクローンに関する論文が、「コードクローンに関する一連の研究」に係る連作論文として、平成18年度電子情報通信学会情報・システムソサイエティ論文賞(ISS論文賞)に選ばれ、研究代表者の井上、分担者の楠本、松下、肥後らが平成2007年11月29日に受賞した。また、情報処理学会論文誌に本研究課題の一成果として発表した論文「産学連携に基づいたコードクローン可視化手法の改良と実装」が、平成19年度情報処理学会論文賞に選ばれ、研究代表者及び分担者の井上、楠本、肥後ならびに研究支援者として研究員に雇用した吉田則裕が2008年5月20日に同賞を受賞している。

さらに、研究代表者の井上は、本研究課題において実施した「ソフトウェアの保守性に影響を与えるコードクローンの分析システム」やソフトウェア部品の重要度に基づく部品検索システム等を開発し、学術面のみならず、実際のソフトウェア開発現場での実用化という点から「プログラム解析技術を用いたソフトウェア開発支援に関する研究」に対する業績が評価され、平成19年度情報処理学

会フェローの称号が授与されている。また、電子情報通信学会からも、コードクローン検出技術や分析システムの開発をはじめとする「ソースコード解析技術の実用化に関する貢献」に対して、平成20年電子情報通信学会フェローの称号を2008年12月に受賞した。

このように、本研究課題は、情報・通信分野に関する国内の2大学会からも優秀な研究成果として認められており、本研究の成果は産業界・学会に大きく寄与しているといえる。

#### (7) 今後の展望

上記のワークショップ開催にも見られる通り、コードクローンに関する研究課題は世界的に見てさらに注目を集める分野の一つである。本課題はコードクローン研究に関して世界をリードするための大きな基礎と考えることができ、今後もさらに大きな飛躍が期待できる。

#### 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 8 件)

- ① 吉田則裕, 服部剛之, 早瀬康裕, 井上克郎 : 類義語の特定に基づく類似コード検索法, 情報処理学会論文誌, Vol. 50, No. 5, pp. 1506-1519, 2009年5月, 査読有り
- ② Yoshiki Higo, Shinji Kusumoto, Katsuro Inoue: "A metric-based approach to identifying refactoring opportunities for merging code clones in a Java software system", Journal of Software Maintenance and Evolution: Research and Practice, Vol. 20, Issue 6, pp. 435-461, November-December, 2008, 査読有り
- ③ 馬場慎太郎, 吉田則裕, 楠本真二, 井上克郎: Fault-Proneモジュール予測へのコードクローン情報の適用, 電子情報通信学会論文誌 D, Vol. J91-D, No. 10, pp. 2559-2561, 2008年10月, 査読有り
- ④ 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, 査読有り
- ⑤ 肥後芳樹, リビエリ シモネ, 松下 誠, 井上克郎: "大規模ソースコードを対象としたコードクローンの検出と可視化", 情報処理学会論文誌, Vol. 48, No. 11,

pp.3510-3519, 2007年11月, 査読有り

⑥ 吉田則裕, 肥後芳樹, 神谷年洋, 楠本真二, 井上克郎: "コードクローン間の依存関係に基づくリファクタリング支援", 情報処理学会論文誌, Vol.48, No.3, pp.1431-1442, 2007年3月, 査読有り

⑦ 肥後芳樹, 吉田則裕, 楠本真二, 井上克郎: "産学連携に基づいたコードクローン可視化手法の改良と実装", 情報処理学会論文誌, Vol.48, No.2, pp.811-822, 2007年2月, 査読有り

⑧ 川口真司, 松下 誠, 井上克郎: "版管理システムを用いたクローン履歴分析手法の提案", 電子情報通信学会論文誌D, Vol.J89-D, No.10, pp.2279-2287, 2006年10月, 査読有り

[学会発表] (計 20件)

① 悦田翔悟, 伊達浩典, 石尾 隆, 井上克郎: "分散処理を用いたコーディングパターン検出ツールの実装", 情報処理学会第71回全国大会, 2009年3月10日, 立命館大学びわこ・くさつキャンパス, 査読有り

② 石尾 隆, 伊達浩典, 市井 誠, 井上克郎: "大規模パターンマイニングを用いた高品質ソースコードの検索", 情報処理学会ソフトウェア工学研究会ウィンターワークショップ2009・イン・宮崎, 2009年1月23日, 査読有り

③ 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", 情報処理学会ソフトウェアエンジニアリングシンポジウム2008 (SES2008), 2008年9月1日, 東洋大学白山キャンパス, 査読有り

④ Yasuhiro Hayase, Yii Yong Lee, Katsuro Inoue: "A Criterion for Filtering Code Clone Related Bugs", International Workshop on Defects in Large Software Systems (DEFECTS 2008), pp.37-38, July 20, 2008, Seattle Hilton Hotel, Seattle, Washington, USA, 査読有り

⑤ Norihiro Yoshida, Takashi Ishio, Makoto Matsushita, Katsuro Inoue: "Retrieving Similar Code Fragments based on Identifier Similarity for Defect Detection", International Workshop on Defects in Large Software Systems (DEFECTS 2008), July 20, 2008, Seattle

Hilton Hotel, Seattle, Washington, USA, 査読有り

⑥ 宮崎宏海, 肥後芳樹, 井上克郎: "アイテムセットマイニングを利用したコードクローン分析作業の効率向上", 電子情報通信学会ソフトウェアサイエンス研究会2008年7月研究会, 2008年7月31日, 公立はこだて未来大学, 査読無し

⑦ Yii Yong Lee, Yasuhiro Hayase, Makoto Matsushita, Katsuro Inoue: "Token Comparison Approach to Detect Code Clone-related Bugs", 電子情報通信学会ソフトウェアサイエンス研究会2008年3月研究会, 2008年3月3日, 長崎大学文教キャンパス総合研究教育棟1階108番講義室, 査読無し

⑧ 吉田則裕, 石尾 隆, 松下 誠, 井上克郎: "欠陥検出を目的とした類似コード検索法", 情報処理学会ソフトウェア工学研究会ウィンターワークショップ2008・イン・道後, 2008年1月24日, 愛媛大学, 査読有り

⑨ 肥後芳樹, 植田泰士, 楠本真二, 井上克郎: "AGMアルゴリズムを用いたギャップを含むコードクローン情報の生成", 電子情報通信学会ソフトウェアサイエンス研究会2007年12月研究会, 2007年12月17日, 島根大学松江キャンパス, 総合理工学部3号館2F多目的ホール, 査読無し

⑩ Yoshiki Higo, Yasushi Ueda, Shinji Kusumoto, Katsuro Inoue: "Simultaneous Modification Support based on Code Clone Analysis", The 14th Asia-Pacific Software Engineering Conference (APSEC2007), December 5, 2007, The Midland Square Building, Nagoya, Japan, 査読有り

⑪ 馬場慎太郎, 吉田則裕, 楠本真二, 井上克郎: "ソフトウェア保守性を評価するメトリクス間の関連分析", ソフトウェア信頼性研究会第4回ワークショップ, 2007年6月8日, 愛媛大学, 査読有り

⑫ Simone Livieri, Yoshiki Higo, Makoto Matsushita, Katsuro Inoue: "Very-Large Scale Code Clone Analysis and Visualization of Open Source Programs Using Distributed CCFinder: D-CCFinder", The 29th International Conference on Software Engineering (ICSE 2007), May 23, 2007, Hilton Minneapolis Hotel, Minneapolis, USA, 査読有り

⑬ Simone Livieri, Yoshiki Higo, Makoto Matsushita, Katsuro Inoue: "Analysis of the Linux Kernel Evolution Using Code Clone Coverage", The 4th Workshop on Mining Software Repositories (MSR 2007), May 19, 2007, Hilton Minneapolis Hotel, Minneapolis, USA, 査読有り

⑭ リビエリ シモネ, 肥後芳樹, 松下 誠, 井上克郎: "D-CCFinder: 超大規模ソースコード集合を対象とした分散処理型コードクローン検出・可視化システム", 電子情報通信学会ソフトウェアサイエンス研究会 2006年12月研究会, 九州大学伊都キャンパス ウェスト2号館3階第4講義室, 2006年12月15日, 査読無し

⑮ 肥後芳樹, 楠本真二, 井上克郎: "コードクローン分析ツールGeminiを用いたコードクローン分析手法", 第5回クリティカルソフトウェアワークショップ (WOCS), 2005年11月2日, 東京国際交流館, 査読有り

⑯ 肥後芳樹, 神谷年洋, 楠本真二, 井上克郎: "リファクタリングを目的としたコードクローン分析ツール Aries", 情報処理学会ソフトウェア工学研究会組込みソフトウェアシンポジウム 2005, 2005年10月18日, 日本科学未来館, 査読有り

⑰ Norihiro Yoshida, Yoshiki Higo, Toshihiro Kamiya, Shinji Kusumoto, Katsuro Inoue: "On Refactoring Support Based on Code Clone Dependency Relation", The 11th IEEE International Software Metrics Symposium, September 19, 2005, Villa Olmo, Como, Italy, 査読有り

⑱ 肥後芳樹, 楠本真二, 井上克郎: "コードクローン分析ツールGeminiを用いたコードクローン分析手法", 電子情報通信学会ソフトウェアサイエンス研究会 2005年8月研究会, 2005年8月4日, 小樽商科大学 407 教室, 査読無し

⑲ 川口真司, 松下 誠, 井上克郎: "版管理システムを用いたコードクローン履歴分析", 電子情報通信学会ソフトウェアサイエンス研究会 2005年8月研究会, 2005年8月4日, 小樽商科大学 407 教室, 査読無し

⑳ Yoshiki Higo, Toshihiro Kamiya, Shinji Kusumoto, Katsuro Inoue: "Aries: Refactoring Support Tool for Code Clone", The 3rd Workshop of Software Quality

(3-WoSQ), May 17, 2005, Adam's Mark Hotel, St. Louis, Missouri, USA, 査読有り

[その他]

本研究課題で開発したコードクローン検出・分析システムなど (Gemini, Aries, Libra, CC-Finder など) を以下のURLに公開しており、企業のソフトウェア開発現場等において実際に利用されている。

<http://sel.ics.es.osaka-u.ac.jp/icca/index.html>

## 6. 研究組織

### (1) 研究代表者

井上 克郎 (INOUE KATSURO)

大阪大学・大学院情報科学研究科・教授  
研究者番号: 20168438

### (2) 研究分担者

楠本 真二 (KUSUMOTO SHINJI)

大阪大学・大学院情報科学研究科・教授  
研究者番号: 30234438

松下 誠 (MATSUSHITA MAKOTO)

大阪大学・大学院情報科学研究科・准教授  
研究者番号: 60304028

石尾 隆 (ISHIO TAKASHI)

大阪大学・大学院情報科学研究科・助教  
研究者番号: 60304028

肥後 芳樹 (HIGO YOSHIKI)

大阪大学・大学院情報科学研究科・助教  
研究者番号: 70452414

### (3) 連携研究者

該当無し