

機関番号：14401

研究種目：基盤研究 (C)

研究期間：2008～2010

課題番号：20500033

研究課題名 (和文) コンテキストに応じたソフトウェア保守管理支援に関する研究

研究課題名 (英文) Study on Software Maintenance Support Based on its Context

研究代表者

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

研究成果の概要 (和文)：本研究では、ソースコードの解析結果 (コードクローンや複雑度メトリクス等) に基づいて、ソフトウェアの保守性をより効率よく改善するための枠組みの提案とその支援システムの開発を行った。具体的には、まず、ソフトウェア保守に影響を与える要因の調査、それらの要因に基づき、ソフトウェア保守に関連したデータを収集・分析・対策の作成を支援する保守ナビゲートシステムの開発、実際の保守における提案手法・ツールの有用性評価である。

研究成果の概要 (英文)：This research proposes a support techniques for software maintainability based on source code analysis results (ex. code clone analysis, complexity metrics measurement, etc.) and develops the support system. The results consist of the followings: a survey of factors/attributes that affect software maintenance activities, a proposal and development a software maintenance support system and evaluations of the system.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2008 年度	1,500,000	450,000	1,950,000
2009 年度	1,000,000	300,000	1,300,000
2010 年度	1,000,000	300,000	1,300,000
総計	3,500,000	1,050,000	4,550,000

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：ソフトウェア工学，ソフトウェア保守，コードクローン，リファクタリング，プログラム依存グラフ

1. 研究開始当初の背景

近年、ソフトウェア開発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。これまで、一般に保守が長期間にわたるエンタープライズ系のソフトウェアに対して、レガシーシステムの問題が多く指摘されてきた。しかし、最近では携帯電話等に代表される組込ソフ

トウェアにおいても新規にスクラッチから作成する開発ではなく、旧機種のソフトウェアを流用した上で、機能変更や追加を施すという開発が増加してきている。また、流用される規模も増加の一途をたどり、現在では数百万ステップ以上という、エンタープライズ系レガシーシステムの規模とほぼ同じものになりつつある。従って、エンタープライズ系、組込系ソフトウェアも含めて大規模な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問題と

なっている。

ソフトウェアの保守性はソフトウェアの複雑さで評価されることが多かった。具体的には、ソフトウェア中のモジュール（関数、手続き、クラス等）の凝集性（機能のまとまり具合を評価する）、結合性（他のモジュールとの関連性を評価する）、あるいは、制御の複雑さ等が用いられてきた。近年では、ソフトウェア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。機能変更をあるコード片に行う場合、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よ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。申請者はこれまでにコードクローン分析に関する研究を行ってきた。その研究を通じて、コードクローン検出ツール CCFinder、分析ツール Gemini 等の開発に携わってきた。また、それらのツール群を複数のソフトウェア開発組織に試用してもらい、実際の開発現場での適用可能性について評価を行ってきた。

その結果、多くの開発組織よりツールの適用結果についてフィードバックを受けた。その多くは、確かにコードクローンは数多く発見できたが、それに対してどのような対策をとればよいかがよく分からないというものであった。保守性を阻害することは理解できているが、納期やコストが厳しい状況では、全てのコードクローンを集約することはできない。また、コードクローンを集約することにより、集約されたモジュールの結合度の悪化を招き、複雑さの観点では逆に悪くなるのではないかという指摘もある。従って、ソフトウェアの保守性を考えた場合、ソースコードの分析結果として同じような特性をもったとしても、様々な開発組織の環境条件（ソフトウェアのドメイン、要求される品質、コスト等）によって保守性の良し悪しが変わってしまうことが確認された。

2. 研究の目的

本研究では、ソースコードの解析結果（コードクローンや複雑度メトリクス等）に基づいて、ソフトウェアの保守性をより効率よく改善するための枠組みの提案とその支援システムの開発を行う。具体的には、以下の(1)～(3)の実現を目指した。

- (1) ソフトウェア保守のコンテキスト（上述した、ソフトウェアに関連する開発組織の環境条件等）において重要な要因の調査
- (2) (1)の枠組みで定義されたコンテキストに

基づき、ソフトウェア保守に関連したデータを収集・分析・対策の作成を支援する保守ナビゲートシステムの開発

- (3) 実際のソフトウェア保守における提案手法・ツールの有用性評価

3. 研究の方法

以下の方針で研究を行った。

(A) ソフトウェア保守に関連した規格等の調査：ソフトウェア保守作業を特徴付ける要因やソフトウェアの保守性の評価手法について調査を行った。ソフトソフトウェア開発プロセスやプロジェクト管理における管理項目等（例えば、ISO/IEC, JIS 等の標準として認定されている、共通フレーム（ISO/IEC 2007, JIS X 0160）やプロジェクトマネジメントの知識体系 PMBOK (Project Management Body of Knowledge)）等を対象とした。

(B) ソフトウェア保守に影響を与える要因分析：ソフトウェア保守に影響を与える様々な要因間の関係を分析した。例えば、ソフトウェアの保守性の評価方法である複雑度メトリクスとコードクローンの関係等である。

(C) ソフトウェア保守のコンテキストを定義するための枠組みの検討：上記(A), (B)の調査結果、分析結果に基づいて、ソフトウェア保守を特徴付けるための枠組みを検討する。

(D) 保守支援ナビゲートシステムの開発：上記の成果に基づいて、保守支援ナビゲートシステムを開発する。開発にあたっては、ソフトウェア保守において重要となる以下の作業に対して支援手法とツールの開発を行う。

(1) 改良保守1：ソフトウェア保守対象の粒度をモジュール単位として、Fault-Prone モジュールの特定を目指す。具体的なアプローチとしては、ソフトウェア開発履歴におけるソフトウェアメトリクス値の変遷を用いて、工数を集中すべきモジュールの特定を行う。

(2) 改良保守2：ソフトウェア保守対象の粒度をメソッド単位として、Fault-Prone メソッドの特定を目指す。具体的なアプローチとしては、メソッド呼出やメソッドで使用される変数に着目し、呼出の種類や変数のスコープを特徴づけ、不具合が発生する可能性が高いメソッドを特定する。

(3) 理解支援：プログラム依存グラフを用いたソースコード可視化手法を実現する。可視化情報を用いて、リファクタリングすべき箇所の候補を利用者に提示したり、詳細部分を省略することで直観的にソースコードの構造を把握できるような支援を目指す。

(4) 再利用支援：既存のソフトウェアに実装された機能を効率的に把握・再利用するため、コードクローン検出技術を用いて、複数のソフトウェアから同様の機能を実現した関数

集合群を検出する手法を実装する。

4. 研究成果

3. で述べた研究方針について、以下の成果を得た。

(A) ソフトウェア保守に関連した規格等の調査：ソフトウェア保守作業を特徴付ける要因やソフトウェアの保守性の評価手法について、主に ISO14764 を中心に調査を行った。

(B) ソフトウェア保守に影響を与える要因分析：ソフトウェア保守に影響を与える様々な要因間の関係を分析した。特に、長期間にわたり保守が行われているソフトウェアに関しては、各種複雑さメトリクスの値、コードクローン情報による定量化が必要であることより、オープンソースソフトウェアを中心に、様々な複雑度メトリクスの計測やコードクローン分析を行い、保守における値の特性を調査した。

(C) ソフトウェア保守のコンテキストを定義するための枠組みの検討：(A)、(B) の調査結果、分析結果に基づいて、ソフトウェア保守を特徴付けるための枠組みとしては、定量的評価が可能な複雑度やコードクローンに関する属性を中心に検討する方向を決めた。

(D) 保守支援ナビゲートシステムの開発：上記の成果に基づいて、保守支援ナビゲートシステムを開発する。開発にあたっては、ソフトウェア保守において重要となる以下の作業に対して支援手法とツールの開発を行う。

(1) 改良保守 1：ソフトウェア保守対象の粒度をモジュール単位として、Fault-Prone モジュールの特定を行った。具体的なアプローチとしては、ソフトウェア開発履歴におけるソフトウェアメトリクス値の変遷を用いて、工数を集中すべきモジュールの特定した。提案手法をオープンソースソフトウェアに適用した実験の結果、対象ソフトウェアの規模に関わらず、有益な情報を得られることが確認された。

(2) 改良保守 2：Fault-Prone メソッドの特定を行った。具体的なアプローチとしては、メソッド呼出やメソッドで使用される変数に着目し、呼出の種類や変数のスコープを特徴づけ、不具合が発生する可能性が高いメソッドを特定した。提案手法をオープンソースソフトウェアに適用した実験の結果、変数使用行率が低い変数が 1 つでも含まれるメソッドに 1 行あたりのフォールト数が多いことが確認できた。また、最低変数使用行率・加重平均変数使用行差・最大変数使用行差の Spearman の順位相関係数は、行数とサイクロマチック数の順位相関係数より絶対値が大きく、よりフォールトとの相関があると確認した。

(3) 理解支援：プログラム依存グラフを用いたソースコード可視化手法を提案し、その手法に基づくシステムを実装した。提案手法では、文の間のデータの繋がりに着目し、データの繋がりの強い部分が一つの機能を表すと考え、メソッド抽出リファクタリングの候補を求め、自動的に提示する。本システムは保守ナビゲートシステムにおける、リファクタリング支援機能に相当し、メソッド抽出リファクタリングをすべき箇所の候補を利用者に提示する。

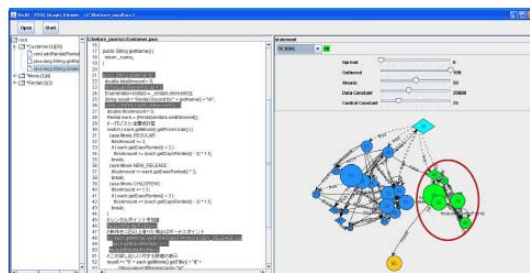


図 1 可視化システム画面例

提案手法を用いてリファクタリングの支援を行うツールを開発し、学生に対して実験を行った。その結果、グラフ操作など操作性の点で問題がありリファクタリング決定時間は長かったが、短い解析時間で結果を出力することができ、14 人中 12 人からグラフによるリファクタリング候補提示は直感的でわかりやすいという意見が得られた。

(4) 再利用支援：既存のソフトウェアに実装された機能を効率的に把握・再利用するため、プログラム依存グラフを用いたコードクローン検出技術を用いて、複数のソフトウェアから同様の機能を実現した箇所を検出する手法を提案し、その手法に基づくシステムを実装した。本システムは保守ナビゲートシステムにおける、修正保守支援機能に相当する。

5. 主な発表論文等

[雑誌論文] (計 5 件)

- ① 肥後, 宮崎, 楠本, 井上, "コードクローン検出に必要な計算コストの削減を目的としたプログラム依存グラフ頂点集約手法の提案," 電子情報通信学会論文誌 D, vol. J93-D, no. 9, pages 1727-1735, 2010 年 9 月. (査読有)
- ② 三宅, 肥後, 楠本, 井上, "多言語対応メトリクス計測プラグイン開発基盤 MASU の開発," 電子情報通信学会論文誌 D, vol. J92-D, no. 9, pages 1518-1531, 2009 年 9 月. (査読有)
- ③ Y. Higo, S. Kusumoto, and K. 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, no. 6, pages 435-461, November 2008. (査読有)

- ④ 馬場, 吉田, 楠本, 井上, "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, 2008年10月. (査読有)
- ⑤ 森崎, 吉田, 肥後, 楠本, 井上, 佐々木, 村上, 松井, "コードクローン検索による類似不具合検出の実証的評価," 電子情報通信学会論文誌D, vol. J91-D, no. 10, 2008年10月. (査読有)

[学会発表] (計8件)

- ① 兼光, 肥後, 楠本: "プログラム依存グラフを用いたリファクタリング候補の特定と可視化", 電子情報通信学会ソフトウェアサイエンス研究会, 2010.12.15 (群馬県渋川市)
- ② S. Yamada, M. Ugumori, and S. Kusumoto, "A Software Tag Generation System to Realize Software Traceability," In Proc. of 17th Asia Pacific Software Engineering Conference (APSEC2010), December 3, 2010. (査読有) (Sydney, Australia)
- ③ K. Hotta, Y. Sano, Y. higo, S. Kusumoto: "Is Duplicate Code More Frequently Modified than Non-duplicate Code in Software Evolution?: An Empirical Study on Open Source Software", In Proc. of the 4th International Joint ERCIM/IWPSE Symposium on Software Evolution, September 21, 2010. (査読有) (Antwerp, Belgium)
- ④ Y. Higo, K. Tanaka, and S. Kusumoto, "Toward Identifying Inter-Project Clone Sets for Building Useful Libraries," In Proc. of the 4th International Workshop on Software Clones, pages pp.87-88, May 8, 2010. (査読有) (Cape Town, South Africa)
- ⑤ Y. Higo, K. Sawa, and S. Kusumoto, "Problematic Code Clones Identification Using Multiple Detection Results," In Proc. of the 16th Asia-Pacific Software Engineering Conference, December3, 2009. (査読有) (Penang, Malaysia)
- ⑥ Y. Higo and S. Kusumoto, "Enhancing Quality of Code Clone Detection with Program Dependency Graph," Proc. of the 16th Working Conference on Reverse Engineering, pages 315-316, Oct. 15 2009. (査読有) (Lille, France)
- ⑦ S. Kusumoto, T. Edagawa, and Y. Higo,

"On an Automatic Function Point Measurement from Source Codes," In 2nd Workshop on Accountability and Traceability in Global Software Engineering (ATGSE2008), pages 27-28, December 2, 2008. (査読有) (Beijing, China)

- ⑧ Y. Higo, K. Murao, S. Kusumoto, and K. Inoue, "Predicting Fault-Prone Modules Based on Metrics Transitions," In Proc. of the International Workshop on Defects in Large Software Systems (DEFECTS2008), pages 6-10, July 20, 2008. (査読有) (Seattle, USA)

6. 研究組織

(1) 研究代表者

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

(2) 研究分担者

岡野浩三 (OKANO KOZO)
大阪大学・大学院情報科学研究科・准教授
研究者番号: 70252632

(3) 連携研究者

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