

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 5日現在

機関番号：13901

研究種目：若手研究(B)

研究期間：2009～2011

課題番号：21700042

研究課題名（和文）

ミスコードリポジトリを用いたソフトウェア開発支援

研究課題名（英文）

Software Development Environment using Miss Code Repository

研究代表者

渥美 紀寿 (Atsumi Noritoshi)

名古屋大学・情報科学研究科・特任助教

研究者番号：70397446

研究成果の概要（和文）：

本研究では、ソースコード解析結果を汎用的に利用でき、ソースコードに基づく開発支援ツールを効率良く実装可能にするためのソースコードの表現方法を提案した。また、ソースコードの違反を検出するために柔軟にカスタマイズ可能なコーディングチェッカを実現した。さらに既存のソースコードから再利用可能なコード断片を抽出する手法を提案した。

研究成果の概要（英文）：

We proposed several source code representations that can commonly use the source code analysis results on various tools and enable to build efficiently development support tools based on source code. We also built a flexibly customizable coding checker to detect violations of coding rules. Moreover, we proposed several methods to extract reusable code fragments from existing source codes.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	1,600,000	480,000	2,080,000
2010年度	800,000	240,000	1,040,000
2011年度	900,000	270,000	1,170,000
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：ソフトウェア

キーワード：開発環境, ソフトウェア保守支援, 問題解決経験, ソフトウェア再利用

1. 研究開始当初の背景

ソフトウェア開発において適切な設計、実装が何かを定義することは困難である。現段階では実装が適切であることを評価するための方法として、実行してその動作を確認するテスト、振舞いを抽象化してモデル化し、特定の性質を満たすことを検証、ソースコードの構造や標準ライブラリの使用法などを静的に検査するなどが行われている。しかし、これらの方法では、不具合を検出することは可能であるが、適切な実装にするためにどの

ように修正すれば良いかを示すことができない。

ソフトウェア開発において、実装工程ではテストと修正が繰り返し行われるが、その際のノウハウは開発者間で共有できていない。ソフトウェア開発において多様なプログラム部品が様々なドメインで実装され、それらを用いて多様なソフトウェアが構築されている。特定のドメインで利用されるプログラム部品はそれを用いて構築したソフトウェアの開発過程において、様々なコーディング

ノウハウが蓄積される。そのノウハウの一部はブログやプログラム部品提供者の課題管理システム等において公開され、情報が共有されているが、十分に活用できていない。

2. 研究の目的

本研究では、個々の開発者が実装中に行った修正ノウハウを形式化し、それをを用いることによって他の開発者を支援することを目的とする。

本研究ではプログラム部品利用時の失敗事例を元に、ミスコードとそれを修正したコードを蓄積する手法を提案する。また、蓄積したミスコードとその修正方法を検索するシステムを構築する。ミスコードを示すことにより、プログラム部品利用者に対して誤った利用法をしないう注意を促すことが可能となり、同様の誤りを防ぐことが可能となる。さらに、これを利用したテストケースを自動生成することにより、信頼性の高いソフトウェアを構築するための開発支援を実現する。

3. 研究の方法

本研究は以下のことを行うことによって、開発活動履歴に基づいた開発支援を行うことを目指す。

- 1) 開発活動履歴の蓄積
- 2) 蓄積されたデータの解析
- 3) ミスコードの判定とその修正の判定
- 4) ミスコードとその修正コードを用いた修正支援

1) の蓄積されたデータのうちソースコードを中心に関連する情報を解析するため、ソースコードの解析が重要となる。ソースコード中の必要となる情報は開発支援の種類によって異なるため、ソースコードの解析結果を汎用的に利用できるようにする必要がある。

また、ミスコードとして登録するためにソースコードの誤りを検出することが必要となる。この判定には JUnit などのテストツールによる判定の他、静的解析ツールによる判定、不具合情報を用いた判定などが考えられる。既存の手法やツールでは不十分な点もあるため、それらを改良する必要がある。

4. 研究成果

本研究課題の成果は 1) 汎用的に利用可能なソースコードの解析結果の表現方法の提案, 2) カスタマイズ可能な静的コーディングチェッカの実現, 3) 再利用可能なコーディングパターンの抽出である。

- 1) 汎用的に利用可能なソースコード解析結果の表現方法

本研究では、ソースコードをもとに開発者の支援を行うリファクタリングツールや波及解析ツール、テスト支援ツール、コーディング支援ツール、メトリクス計測ツールなどの下流工程における開発支援ツールを効率良く実現するためのソースコード解析結果の XML 表現を提案した。提案表現では、ソースコード中の全ての字句をそれぞれの字句の種類を表すタグで囲み、同じ識別子には同じ id を属性として付加することによって記号表の情報を表現した。型情報はソースコード中の宣言箇所以外には字句として表れないため、識別子の出現箇所において型情報を属性として付与した。

XML ではファイルを跨った関係性を表現することが困難なため、クロスリファレンス情報をソースコードの XML 表現とは別に表現する方法を提案した。この XML 表現には、各構文要素に対し、定義情報、引数情報、使用変数情報、呼び出し関数情報、被呼び出し関数情報、出現情報などファイル間を跨る情報を表現した。

提案表現の有用性を確認するため、コーディングチェッカとクロスリファレンサが容易に実現可能であることを示した。

2) カスタマイズ可能なコーディングチェッカ

本研究では、ミスコードを判定するためのツールとしてコーディングチェッカを開発した。コーディングルールは開発対象のドメインや開発プロジェクトによって個別に設定されるコーディング規約に柔軟に対応できるように、カスタマイズ性が高く、ルールの追加が容易となるように設計した。実際に組込みシステムで一般的に利用されているコーディング規約の多くの規約をルールとして実現することができ、その有用性を確認した。

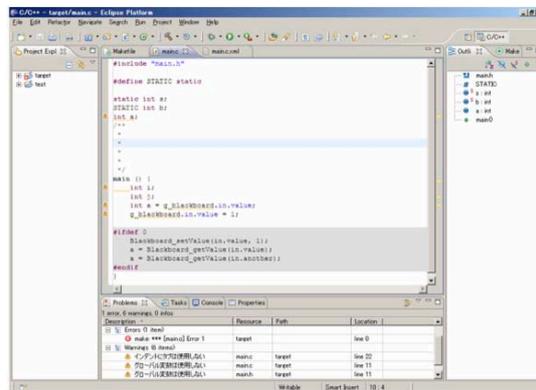


図 1 CX-Checker の動作イメージ

また、コーディング検査結果がソースコードの修正過程において変化する過程を調査

するために、コーディング規約違反に着目したソフトウェアメトリクスを提案した。提案手法では、コーディングチェッカの出力結果から、ルールや開発者ごとの違反の出現頻度、時間分布、空間分布に基づくメトリクスを提案し、オープンソースソフトウェアの開発履歴を用いてその効果を測定する実験を行った。

3) 再利用可能なコーディングパターンの抽出

本研究では、Java 言語におけるソフトウェア部品の典型的な利用方法の抽出手法と、JavaScript における環境依存を吸収するための代替コードを検出する手法を提案した。

Java 言語に対してメソッドの責務単位の処理の流れに着目し、その類似度に基づいてソフトウェア部品の利用方法をコードテンプレートとして抽出する手法を提案した。本手法ではクラス内でのコードの分離、継承を用いたコードの分離、委譲によるコードの分離によるソフトウェア部品の利用方法の違いを吸収するために、これらによる分離されたコードをインライン展開する。展開されたコードからソースコードの特徴としてメソッド呼び出し系列を抽出し、それをマイニング技術によってパターンを抽出することによって、典型的なメソッド呼び出し系列をコードテンプレートとして提示する。予備実験により、有用なコードテンプレートは少ないが、抽出可能であることを示した。

Web アプリケーションのクライアントサイドで利用される JavaScript は実行環境が多様であり、様々な実行環境を考慮した実装が必要となる。本研究では、Web 上の JavaScript コードを収集し、コード中のオブジェクト判定コードとその制御文内の関数呼び出しを抽出する。これをシーケンシャルパターンマイニングによって代替コードを検出する手法を提案した。実際に Web 上に存在する JavaScript コードを収集し、代替コードが検出可能であることが確認できた。

まとめと今後の課題

本研究では、ミスコードリポジトリを用いた開発支援を行うための基盤となる技術を中心に研究を行った。ミスコードリポジトリを構築し、各種データを蓄積しているが、それらを用いた開発支援は現段階ではまだできていない。

蓄積されたデータを解析する技術は整いつつあり、今後それらを用いた開発支援手法を提案し、ツールを実装する予定である。

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 6 件)

- (1) 大須賀 俊憲, 小林 隆志, 渥美 紀寿, 間瀬 順一, 山本 晋一郎, 鈴木 延保, 阿草 清滋, CX-Checker: 柔軟にカスタマイズ可能な C 言語プログラムのコーディングチェッカ, 情報処理学会論文誌, Vol.53, No.2, pp.590-600, 2012 (査読有).
- (2) 桑原 寛明, 渥美 紀寿, 山本 晋一郎, プログラム解析技術のサービス化の試み, ソフトウェア工学の基礎 XVIII, 日本ソフトウェア科学会 ソフトウェア工学の基礎研究会, pp.243-248, 2011 (査読有).
- (3) 渥美 紀寿, 谷 聡貴, 大塚 直也, 小林 隆志, 山本 晋一郎, 阿草 清滋, 組込みソフトウェア製品ファミリーにおけるコンパイルスイッチによる構成管理手法, ソフトウェア工学の基礎 XVII, 日本ソフトウェア科学会 ソフトウェア工学の基礎研究会, pp.173-178, 2010 (査読有).
- (4) 上原 伸介, 小林 隆志, 渥美 紀寿, 山本 晋一郎, 阿草 清滋, トレースに基づくデバッグにおける欠陥箇所発見支援手法, ソフトウェア工学の基礎 XVII, 日本ソフトウェア科学会 ソフトウェア工学の基礎研究会, pp.55-64, 2010 (査読有).
- (5) 渥美 紀寿, 山本 晋一郎, 小林 隆志, 阿草 清滋, CASE ツール・プラットフォームのための C ソースプログラムの XML 表現とその応用, ソフトウェアエンジニアリング最前線 2010, 情報処理学会ソフトウェア工学研究会, pp.135-142, 2010 (査読有).
- (6) 大須賀 俊憲, 小林 隆志, 間瀬 順一, 渥美 紀寿, 山本 晋一郎, 鈴木 延保, 阿草 清滋, CX-Checker: C 言語プログラムのためのカスタマイズ可能なコーディングチェッカ, ソフトウェアエンジニアリング最前線 2009, 情報処理学会ソフトウェア工学研究会, pp.119-126, 2009 (査読有).

[学会発表] (計 1 1 件)

- (1) 大場 光明, 渥美 紀寿, 小林 隆志, 阿草 清滋, メソッド境界を越えた呼び出しパターン抽出のためのコールグラフ探索戦略, 情報処理学会 ソフトウェア工学研究会, 2012. 03. 16, 東京

- (2) 戸田 達也, 小林 隆志, 渥美 紀寿, 阿草 清滋, 実行トレース解析のためのデザインパターンに基づくオブジェクトグルーピング, 電子情報通信学会ソフトウェアサイエンス研究会, 2012. 03. 14, 沖縄.
- (3) Noritoshi Atsumi, Mitsuaki Oba, Takashi Kobayashi, Shinichiro Yamamoto, Kiyoshi Agusa, An XML Representation of Preprocessed Code for Static Analysis, The IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE2011), 2011. 11. 29, Hiroshima, Japan.
- (4) 渥美 紀寿, 小林 隆志, 山本 晋一郎, 阿草 清滋, バリエーション並行開発のための版管理機構, 日本ソフトウェア科学会第 28 回大会, 2011. 09. 28, 沖縄.
- (5) 藤谷 隆宏, 小林 隆志, 渥美 紀寿, 阿草 清滋, シーケンシャルパターンマイニングによる JavaScript の代替コード発見, 日本ソフトウェア科学会第 28 回大会, 2011. 09. 27, 沖縄.
- (6) Noritoshi Atsumi, Takashi Kobayashi, Shinichiro Yamamoto, Kiyoshi Agusa, An XML C Source Code Interchange Format for CASE Tools, The 35th Annual IEEE Computer Software and Applications Conference (COMSAPC2011), 2011. 07. 20, Munich, Germany.
- (7) 渥美 紀寿, プログラム解析のサービス化に向けて, ウィンターワークショップ 2011・イン・修善寺, 2011. 01. 21, 静岡
- (8) 高井 康勢, 渥美 紀寿, 小林 隆志, コーディング規約違反に着目したソフトウェアメトリクス, ウィンターワークショップ 2011・イン・修善寺, 2011. 01. 21, 静岡.
- (9) 小林 隆志, 大須賀 俊憲, 上原 伸介, 蛭牟田 英治, 林 英志, 間瀬 順一, 山本 晋一郎, 渥美 紀寿, 川口 直弘, 鈴木 延保, 阿草 清滋: CX-Checker: 柔軟なカスタマイズが可能な C 言語コーディングルールチェッカー, 情報処理学会 全国大会, 2010. 3. 4, 東京.
- (10) Eiji Hirumuta, Takashi Kobayashi, Noritoshi Atsumi, and Kiyoshi Agusa, Finding Alternate Javascript Codes with a Code Repository, The IASTED International Conference on Software Engineering, 2010. 02. 17, Innsbruck, Austria.
- (11) 渥美 紀寿, 山本 晋一郎, ソフトウェア部品の階層的構成関係によるソースコードの類似性評価, ソースコードの類似性ワークショップ, 2009. 09. 07, 東京.

6. 研究組織

(1) 研究代表者

渥美 紀寿 (ATSUMI Noritoshi)
名古屋大学・情報科学研究科・特任助教
研究者番号: 70397446

(2) 研究分担者

なし

(3) 連携研究者

なし