

科学研究費助成事業 研究成果報告書

平成 29 年 6 月 8 日現在

機関番号：14301

研究種目：基盤研究(B) (一般)

研究期間：2012～2016

課題番号：24300006

研究課題名(和文)バリエントライフサイクルモデルに基づく多品種並行開発手法

研究課題名(英文)Multi-variant development management method based on variant life cycle model

研究代表者

阿草 清滋 (AGUSA, Kiyoshi)

京都大学・学術情報メディアセンター・研究員

研究者番号：90026360

交付決定額(研究期間全体)：(直接経費) 13,800,000円

研究成果の概要(和文)：本研究の目的は、機能の細部が異なる同種の製品を継続的に並行して開発する多品種並行開発において、それぞれの環境に適応するようカスタマイズしたソフトウェアバリエントを管理・運用するための新しい開発手法を確立することである。
効果的な多品種並行開発のための共通性・可変性の管理方法について整理し、単一ソースコードに複数のバリエントが記述されている場合の静的・動的な解析方法、バリエント開発に必要なトレーサビリティ管理のための構成要素間の依存関係抽出・活用手法について研究を行った。

研究成果の概要(英文)：Multi-variant software is a product family consists of variant products that have common features and variations of feature details with configurations. We aimed to develop a method for the management of multi-variant software development when variant products for various environments are developed and evolved in parallel.
We study a method to manage commonality and variability in a multi-variant software and developed static and dynamic analysis methods for a multivariant software implemented as a single code base. As a basis of the traceability management in a multi-variant software development, we also developed techniques to extract and leverage latent dependencies among software artifacts in variant products.

研究分野：ソフトウェア

キーワード：ソフトウェア工学 ソフトウェア構成管理 ソフトウェア保守 ソフトウェア開発支援

1. 研究開始当初の背景

今日、社会情報基盤から組込み機器まで、さまざまなシステムの機能の大部分はソフトウェアによって実現され、社会の発達・進歩がソフトウェアの発達・進歩によってもたらされる。近年のクラウド技術の発展により、機器だけでなくソフトウェアによって実現されるサービスまでもがグローバル化され、迅速な開発・進歩が求められる。グローバルマーケットを対象とするサービスや基盤、機器は、法令や文化といった地域の特性や、様々な顧客ニーズに柔軟に適応させる必要がある。結果として、機能の細部が異なる同種のソフトウェアであるバリエーションを継続的に開発する必要がある。このようなソフトウェア群を迅速に開発するために、多品種並行開発が行われる。

機能の細部が異なる同種の製品を効率良く開発するためには、より多くの部分を共通化し、共通部分との差分を正確に理解しながら開発を行う必要がある。これらの差分は部品の実装だけでなく部品の組み合わせ方法も考慮する必要があり、バリエーション数の増加に合わせ管理対象は膨大なものになる。

関連度の高いソフトウェア群を効率良く開発するため、再利用とドメイン分析を組み合わせたソフトウェアプロダクトライン (SPL) が提案されている。再利用による効率化は多くの実績があるため様々な業種において SPL が実践されているが、SPL は事前に計画した製品ファミリーから、新たに単一のソフトウェアを作ることを想定しており、並行して複数のバリエーションを開発することが想定されていない。特に、複数のバリエーションを並行開発する場合の構成管理手法および、コアセットと呼ばれる共通部分の継続的な更新に関しては支援手法が存在せず、SPL 型開発の課題となっている。

2. 研究の目的

本研究の目的は、機能の細部が異なる同種の製品を継続的に並行して開発する多品種並行開発において、それぞれの環境に適応するようカスタマイズしたソフトウェアバリエーション(バリエーション)を管理・運用するための新しい開発手法を確立することである。仕様策定から完成までのバリエーションのライフサイクルモデルを定義し、モデルに基づく新しい変更管理手法や構成管理手法を定義することによって開発の効率化を計る。

具体的には、バリエーションライフサイクルにおける以下の点を明らかにする。

- ・バリエーション構成管理手法: バリエーション間の関係を考慮し、個々のバリエーションの変更が影響を与える範囲を明確にするための管理手法を提案する。また、特定のバリエーションの変更履歴を追跡できるように、ソースコード断片とそれに対応するバリエーションの関係を版管理するための手法を提案する。また、バリエーションソフトウェアは複雑な設定項目により構成されるため、その構成の理解支援に関する研究も実施する。

- ・バリエーション設計実装手法: バリエーションには、ソースコードを切り替える静的バリエーションだけでなく、実行時に挙動を変化させる動的バリエーションも存

在する。これらの実装方法を体系化し、双方の実装や実装方法の切り替えの自動化などを支援する手法を開発する。また、管理対象のバリエーション群の不具合や機能追加などの際に、バリエーションの仕様と実装のトレーサビリティ情報を用いて効果的に修正を行う方法を確立する。さらに、複数のバリエーションの開発履歴からトレーサビリティ情報を復元する手法を開発し、高コストとなるトレーサビリティ情報の管理を支援する方法を開発する。

3. 研究の方法

まず、バリエーションライフサイクルモデルの検討を行った。検討に際して効果的な多品種並行開発のための共通性・可変性の管理方法について整理し、共通性を認識するためのトレーサビリティ管理、単一ソースコードに複数のバリエーションを記述する際の解析方法について研究する。また、並行に開発が進む過程において、管理されるべきトレーサビリティリンクの種類およびその活用方法について調査し、これらのトレーサビリティリンクを自動的に認識するための基礎研究を実施する。

また、マルチバリエーションソフトウェアの実装方法の1つである、コンパイルスイッチを活用した実装方法に着目し、プリプロセス命令の関係を解析する研究を行う。マルチバリエーションソフトウェアを解析する基盤として、ソースコードを抽象構文木レベルではなく、ホワイトスペースやコメントといった全ての情報を含んだソースコード全体を管理するための基盤技術を開発する。バリエーションソフトウェアの解析については、構築管理ツールの設定ファイルに複雑に記述されているバリエーション構成に必要な依存関係を、構築管理ツールの実行履歴から抽出する技術に関する研究を行う。

4. 研究成果

(1) バリエーションの構成方法と修正頻度の関係の調査

コンパイルスイッチを用いたマルチバリエーションソフトウェアの実現方法において、各バリエーションはコンパイルスイッチの組み合わせで表現されるが、その構成方法が保守性に影響を及ぼす可能性がある。マルチバリエーションソフトウェアにおけるバリエーション構成方法に着目した保守性評価に有用な要素を発見することを目的とし、バリエーションの構成方法と修正頻度の関係に関する調査を行った。

本調査では、版間で変更されたコードが依存している前処理条件を検出し、その前処理条件を含む前処理条件の制御構造をソースコード解析によって求め、前処理条件によって制御されるコードの行数、制御構造中のコードの行数、制御構造中のコンパイルスイッチの数、ネストの深さと改変回数との間の関係について調査した。OSS を対象に調査したところ、制御構造の違いによる改変頻度の差は見られなかったが、制御されるコードの改変頻度が高い前処理条件が存在すること、依存するコンパイルスイッチが多い

ほど改変頻度が高いことがわかった。

(2) ソフトウェアの構築過程のモデル化

多品種並行開発においては、多様で複雑なコンフィギュレーションが、ソフトウェアの設計、開発、テスト、保守に影響を与えるため、ソフトウェアの構築過程を表現して管理するモデルの必要性が増加する。本研究では、大規模かつ複雑化したソフトウェアの構築過程における複雑な関係表現することを目的として、ソフトウェアの構築過程を表現するモデルを提案した。

ソフトウェアの構築フェーズでは、多くのオプションを与えられたビルドコマンドが、複雑な依存制約の下で実行されるため、昔からmakeを初めとする各種のビルドツールが提案されている。ただし、それぞれのビルドツールに対して専用の記述が必要であり、残念ながら決定版と呼べるものはない。そこで、本研究では、ソフトウェア構築のログを解析して、コマンドとオプションの関係、ソースファイルの生成ファイルの関係、などを取得する手法を提案し、ツールとして実装した。このツールを用いることで、大規模なソフトウェア構築ログから必要な情報を検索することや、利用者が見やすい形式で可視化を行うことが可能となった。また、ビルドに使用されるコマンド、実行手順やファイルの生成関係を読み解くことにかかる時間を短縮することができた。

(3) ソースコードに基づくフィーチャ間の依存性の抽出

マルチバリエーションソフトウェアの実装において、バリエーション間の共通性・可変性を認識し、適切なバリエーションを実現するためには、バリエーション間の依存関係を把握する必要がある。バリエーション間の依存関係をソースコードから抽出するために、マルチバリエーションソフトウェアの実装方法の1つである、コンパイルスイッチを用いた実装方法に着目し、プリプロセス命令間の関係を解析する研究を行った。プリプロセス命令の制御構造に着目することにより、依存性のあるコンパイルスイッチ間の関係をグラフ構造として抽出し、バリエーション間の差異である可変箇所に対して構成可能な選択を支援するための基礎解析手法を提案した。

コンパイルスイッチはバリエーションを表現する目的の要素以外に実行環境やテスト、ログなど様々な目的で利用されるため、自動抽出したコンパイルスイッチ間の関係はバリエーション間の関係よりも細かい粒度の結果となる。本研究ではコンパイルスイッチ間の依存関係として出現頻度に基づく依存度が高い要素のみを抽出することで、重要なコンパイルスイッチのみを抽出する手法を提案した。

OSSを対象とした実験により、適切に依存度を設定することによって、機能の実現に関係するフィーチャ間の関係を抽出可能なことを確認した。

(4) 動的解析結果の静的解析を用いた一元化

ソフトウェアの大規模化・複雑化により、ソフトウェアを理解することは困難な作業になりつつあ

る。特に、多品種並行開発においては、複雑なコンフィギュレーションによる多様な相違がソフトウェアの複雑度を増加させるため、現実的な時間とコストでソフトウェアを理解して的確に保守することはより難しい。そのため、大規模で複雑なソフトウェアの理解に必要な箇所を提示して支援するツールが求められている。本研究では、ソフトウェアの概要把握を行う理解の初期段階にある開発者に対して、その手掛かりとして構造体型を特徴付ける手法を提案した。提案手法は以下の2つの特徴を持つ。

①動的解析による複数の実行履歴を統一的に扱うために、静的解析を用いて時点軸という概念を導入する。それにより、動的解析の結果を大局的に把握することができる。

②構造体型を利用頻度に基づいて特徴付けることにより、開発者がソフトウェアの機能を分類し、概要把握の支援を行うことができる。

前者の概要把握では機能を大まかに分類しソフトウェアの全体像を捉えることを目的とするため、機能と関連が高い構造体型を本手法によって特徴付けることで、概要把握に必要な機能分類の支援を行う。後者では、関数の呼び出し文脈に基づき構造体型のアクセス回数を計測し、構造体型を実行フェーズに特徴付けた。適用実験において、gzip(約6,000行、6構造体)については人手による分類と提案手法による分類が一致し、「入力→処理→出力」という実行フェーズに各構造体を特徴付けることができた。

(5) マルチバリエーションソフトウェアの静的バリエーション構成の解析に向けた静的解析ツールの開発

マルチバリエーションソフトウェアを解析する基盤として、ソースコードを抽象構文木レベルではなく、ホワイトスペースやコメントといった全ての情報を含んだソースコード全体を管理する基盤として、ソースコードに静的解析した結果をXMLタグとして挿入する基盤技術に関して論文としてまとめた。本技術を利用することにより、コンパイルスイッチの高精度な解析が可能となり、大規模なソフトウェア群を対象としてコンパイルスイッチの組み合わせと不具合の関係を調査することなどが可能となった。

(6) 開発者の行動履歴分析に基づく成果物間の暗黙の依存関係の発見と活用方法の開発

トレーサビリティリンクを自動的に認識するための基礎研究として、開発活動を記録し、暗黙の成果物間の依存関係を発見する手法の研究を行った。成果物群に対する参照・変更の履歴をもとに、変更に至った過程で参照されたファイルの順序付き集合の類似性によって、構成要素間の依存関係を分類する方法を提案した。この分類を活用することによって次に必要となる変更を推薦する手法について研究し、複数の依存関係が分類により効果的に区別され精度よく推薦できることを明らかにした。また、過去の推薦履歴も活用することによって、推薦精度がより向上することも明らかにした。さらに、長期間の実開発における活動履歴情報を分析することによ

て、統合開発環境に対する操作の履歴を用いることによって、版管理システムの情報のみを利用する場合と比べ、同程度の推薦精度を保ちながら多くの変更に対応できる依存関係が抽出できることも明らかにした。

(7) 改版履歴分析に基づく成果物間の暗黙の依存関係の発見と活用方法の開発

開発者の行動履歴は記録されていない場合が多く、前述の手法はこれまでの資産に対して適用することができないため、実際の開発現場で既に記録している改版履歴を用いる手法を研究した。版管理システムに記録されている過去の変更履歴のみを分析することにより、ソフトウェア成果物間の依存関係を抽出しトレーサビリティ情報を復元する手法を提案した。本研究では、複数の大規模 OSS の開発履歴を詳細に分析することにより、時間的近接性の考慮と大規模な変更に対する取扱い方法の影響を調査し、依存関係の抽出精度を向上させる手法を明らかにした。これらの内容をまとめた論文は情報処理学会論文誌特選論文を受賞した。

また、マルチバリエーションソフトウェアの特徴を考慮した暗黙の依存関係の発見手法についても研究を行った。各版における前処理命令の出現箇所を分析することにより、前処理命令の影響を受ける箇所に対する変更を考慮した共変更ルールの抽出手法を提案した。前処理命令の分岐条件と変更の関係を解析することで、改版履歴中の変更属性を付与しルールを生成することが可能となる。この手法について、3 つの代表的な中規模 OSS を用いて評価実験をとおして、提案手法によって得られた共変更ルールは、既存手法と比較して推薦精度を表す MRR の最大値が向上することを示した。

マルチバリエーションソフトウェアの開発においては、バリエーションごとに利用される機能群が異なる構成をとることが一般的であり、機能間の関係性を把握しながら変更を行う必要がある。提案手法によって得られる共変更ルールはリフト値が高いことから、見落とされやすい機能間の依存関係を表現しており変更漏れに対する推薦について効果が高いことを明らかにした。

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 渥美, 小林, 山本, 阿草, ソフトウェア開発支援基盤のためのソースプログラムの XML 表現, 電子情報通信学会論文誌, 査読有, Vol.J97-D, 2013, 2681-2691

② 渥美, 桑原, MAFP: ソースコードに対する静的検査における警告の管理ツール, コンピュータソフトウェア, 査読有, Vol.33, 2016, 50-66
<http://doi.org/10.11309/jssst.33.4.50>

③ 森, A.Hagward, 小林, 改版履歴の分析に基づく変更支援手法における時間的近接性の考慮と同一作業コミットの統合による影響, 情報

処理学会論文誌, 査読有, Vol.58, 2017, 807-817

[学会発表](計 40 件)

① 渥美, 小林, 高田, ソフトウェア開発におけるトレーサビリティ確保のための開発環境の検討, 情報処理学会 ソフトウェア工学研究会, 東京都新宿区, 2012

② K. Noda, T. Kobayashi, K. Agusa, Execution Trace Abstraction based on Meta Patterns Usage, 19th Working Conference on Reverse Engineering, Kingston, Canada, 2012

③ T. Fujitani, T. Kobayashi, N. Atsumi, K. Agusa, Finding Cross-browser JavaScript Codes using Sequential Pattern Mining, Rodos, Greek, 2012

④ T. Kobayashi, N. Kato, K. Agusa, Interaction Histories Mining for Software Change Guide, 3rd International Workshop on Recommendation Systems for Software Engineering, Zurich, Switzerland, 2012

⑤ 丸岡, 小林, 阿草, 成果物アクセスの時間的局所性を考慮した変更コンテキストモデル, 電子情報通信学会 ソフトウェアサイエンス研究会, 福岡県福岡市, 2013

⑥ 渥美, 小林, 阿草, プリプロセス命令の制御構造を利用したフィーチャ間の依存性解析, 電子情報通信学会 ソフトウェアサイエンス研究会, 福岡県福岡市, 2013

⑦ 永井, 大久保, 粕谷, 山本, 実行履歴に基づくソフトウェア構築モデルの提案, ソフトウェアエンジニアリングシンポジウム 2013, 東京都港区, 2013

⑧ 竹治, 大久保, 粕谷, 山本, 齋藤, 大規模ソフトウェアの概要把握支援のための動的解析結果の静的解析を用いた一元化, ソフトウェアエンジニアリングシンポジウム 2013, 東京都港区, 2013

⑨ T.Toda, T.Kobayashi, N.Atsumi, K.Agusa, Grouping Objects for Execution Trace Analysis based on Design Patterns, The 5th International Workshop on Empirical Software Engineering in Practice, Bangkok, Thailand, 2013

⑩ 永井, 大久保, 粕谷, 山本, ソフトウェア構築ログ解析器を用いたソースプログラム解析, ソフトウェアエンジニアリングシンポジウム 2014, 東京都港区, 2014

⑪ 渥美, 桑原, 変更追跡機能を用いた静的検査ツールの効果的な利用法, 日本ソフトウェア科学会 ソフトウェア工学の基礎ワークショップ 2014, 鹿児島県鹿児島市, 2014

⑫ 渥美, 桑原, 静的検査ツールにおける false positive に関する調査, 第 22 回 ソフトウェア工学の基礎ワークショップ, 山形県天童市, 2015

⑬ T.Mori, A.Hagward, T.Kobayashi, Effects of Recency and Commits Aggregation on Change Guide Method Based on Change History Analysis, 10th International Conference on Software Engineering Advances, Barcelona, Spain, 2015

⑭ 蜂巢, 吉田, 阿草, 命令型プログラミング言

語における初学者向け動作理解支援ツールの提案, 第 22 回 ソフトウェア工学の基礎ワークショップ, 山形県天童市, 2015

⑮ 今西, 渥美, 森崎, 山本, 阿草, 前処理命令の制御構造とその構造内のコード変更に関する調査, 第 191 回ソフトウェア工学研究発表会, 大阪府吹田市, 2016

⑯ 中野, 林, 小林, 動的機能検索における関連度と探索戦略, 電子情報通信学会 ソフトウェアサイエンス研究会, 北海道札幌市, 2016

⑰ 今西, 渥美, 森崎, 山本, 阿草, バリエーションコードの変更履歴に基づく前処理条件の構造に関する特徴調査, 第 23 回ソフトウェア工学の基礎ワークショップ, 香川県琴平町, 2016

⑱ 小林, 蜂巢, 吉田, 阿草, データ構造と関数定義に着目したプログラミング学習者用テストケース評価記述方法の提案, 第 23 回ソフトウェア工学の基礎ワークショップ, 香川県琴平町, 2016

⑲ 森, 小林, 林, 渥美, 前処理命令による可変点を考慮した共変更ルール抽出, 電子情報通信学会 ソフトウェアサイエンス研究会, 沖縄県那覇市, 2017

⑳ A.Yamamori, A.Hagward, T.Kobayashi, Can Developers' Interaction Data Improve Change Recommendation? The 41st IEEE Computer Society International Conference on Computers, Software & Applications, Torino, Italy 2017
(他 20 件)

6. 研究組織

(1) 研究代表者

阿草 清滋 (AGUSA, Kiyoshi)

京都大学・学術情報メディアセンター・研究員

研究者番号: 90026360

(2) 研究分担者

山本 晋一郎 (YAMAMOTO, Shinichiro)

愛知県立大学・情報科学部・教授

研究者番号: 40240098

小林 隆志 (KOBAYASHI, Takashi)

東京工業大学・情報理工学院・准教授

研究者番号: 50345386

手嶋 茂晴 (TESHIMA, Shigeharu)

名古屋大学・未来社会創造機構・特任教授

研究者番号: 50394599

渥美 紀寿 (ATSUMI, Noritoshi)

京都大学・情報環境機構・助教

研究者番号: 70393446