

博士論文審査報告書

論文題目

A Study on Source Code Reengineering Frameworks Supporting Multiple Programming Languages

複数のプログラミング言語に対応するソース
コードリエンジニアリングフレームワークに
関する研究

申請者

Kazunori	SAKAMOTO
坂本	一憲

情報理工学専攻 高信頼ソフトウェア工学研究

2013年 2月

プログラムソースコードリエンジニアリングは、ソースコードを解析し再構成することでソフトウェアの品質向上を図るためのソフトウェア工学において重要な手法の一つである。プログラミング言語は少なくとも 2500 言語が確認されており、リエンジニアリングが対象とする言語は多様化している。

プログラミング言語にはそれぞれ得意とする問題領域が存在する。ソフトウェアが扱う問題領域は1つの言語が取り扱う問題領域の範囲よりも広いことがある。そのようなケースでは、複数の言語を利用してソフトウェアを開発することが多い。例えば Web アプリケーションは、クライアント・サーバモデルによって構築されるため、クライアントサイドとサーバサイドがそれぞれ異なる言語で開発されることが多い。しかし、多くのリエンジニアリングツールは少数の言語のみに特化しており、複数の言語を利用した開発に導入することが難しい。具体的には一部の言語について対応ツールが存在せず、また、そのツール開発に莫大なコストが必要となることがある。さらに、複数のツールを組み合わせるリエンジニアリングを適用する際に、ツールごとの仕様や動作の差異の影響を受け、一貫性のある適用ができない場合がある。

本論文では、上述の問題を解決するために、複数のプログラミング言語に対応するソースコードリエンジニアリングフレームワークの提案を行なっている。テストカバレッジ測定にのみ特化したフレームワークの提案について説明した後、それを発展させる形で、リエンジニアリング全般に対応するフレームワークの提案内容をフレームワークの適用事例とともに記述した。

各章の構成は以下のとおりである。

1 章では、本論文で扱う背景として代表的なプログラミング言語を示した。また、既存のテストカバレッジ測定ツールの比較を通して現状のソースコードリエンジニアリングツールの問題点を示し、本論文の目的を明らかにした。

2 章では、複数のプログラミング言語を利用したソフトウェア開発の必要性について述べた。また、申請者によるソフトウェア開発の経験をもとに、複数のプログラミング言語を利用した開発の事例を紹介した。

3 章では、複数のプログラミング言語に対応するテストカバレッジ測定フレームワーク **Open Code Coverage Framework (OCCF)** を提案した。既存のカバレッジ測定ツールが少数の言語のみに対応しているため、例えば、**Java** と **JavaScript** で開発された Web アプリケーションに対してカバレッジを測定する際に、ステートメントカバレッジは測定できるが、ブランチカバレッジを測定できないという問題がある。さらに、ステートメントの定義がツールによって異なるため、一貫性のない測定値が得られるケースがある。また、新しい言語が次々と開発されていく中で、新しい言語に対応するツール開発のコストが高いという問題、既存ツールでは新しいカバレッジの測定基準をツールに導入することができないという問題、一部のツールではデッドコードを測定対象にできないという問題を解決するために、**OCCF** を提案した。本論文では、種々のプログラミング言語の抽象構文木が類似している

点に着目して、抽象構文木上でカバレッジ測定用の特殊なコードを埋め込む手法を確立した。類似する抽象構文木に対する操作を共通処理として抽出することで、OCCF が再利用可能なコードとして共通処理を提供する。その結果、OCCF は 3 種類のプログラミング言語に対して、一貫した 4 種類のカバレッジ測定基準による測定を実現した。また、新しい言語および新しいカバレッジ測定基準の追加要求に対して、実際に被験者実験を通して既存ツールよりも追加が容易であることを示した。さらに、抽象構文木を通して測定用コードを埋め込むことで、デッドコードも測定対象として扱うことができた。以上から、OCCF が上述した問題点を解決できることを示した。

4 章では OCCF の有用性を確認するため、OCCF を利用してテストケース群を最小化する手法を提案した。しばしば、テストコードは製品コードとは異なりリファクタリングされないため、冗長なテストケースが蓄積される問題がある。冗長なテストケースはテスト実施時間をいたずらに増大させ、また、テストコードの保守性を低下させる問題を引き起こす。そこで本論文では、OCCF を用いてテストカバレッジを測定して、テストケースごとに得られた測定値を解析することで、重複したテストケースを検出する手法を提案した。テストカバレッジに測定基準がいくつかあるように、重複判定の基準をカバレッジに対応させる形で設けて、それぞれ OCCF を改良することで重複判定に必要な情報の収集を実現した。実際に、オープンソースソフトウェアである Jenkins の 4 種類のプラグインに対して、提案手法を適用することで、重複テストケースを検出できることを確認した。

5 章では、複数のプログラミング言語に対応するソースコードリエンジニアリングフレームワーク UNICOEN を提案した。複数のプログラミング言語に対応するフレームワークは既存研究でいくつか提案されているが、対応するプログラミング言語に対する共通な言語モデルの提供、新しいプログラミング言語への対応の追加可能性、解析と再構成の両方への対応、静的型付け言語と動的型付け言語の両方への対応という観点を全て満足するフレームワークは存在していない。また、プログラミング言語を静的型付け言語と動的型付け言語に分類することができるが、一般に動的型付け言語はソースコードを解析するだけでは変数の型を判別することが難しく、静的型付け言語よりもリエンジニアリングが難しく、ツールの開発が進んでいない現状がある。

本論文では、ソースコードを解析する段階として、字句解析、構文解析、意味解析の三段階がある中で、静的型付け言語と動的型付け言語の違いが現れない字句解析と構文解析に着目して、二種類の解析で得られる情報を元に静的型付け言語と動的型付け言語の両方を含む 7 種類の言語に対して、共通の言語モデルの定義を行った。UNICOEN は共通言語モデルを中心に据え、リエンジニアリングツールの開発者のための API と対応言語を追加するための API の二種類の API を提供する。ツール開発者向け API では、言語モデル上の共通の抽象構文木に対して、ノードの探索、追加、置換、削除などの

操作系を提供する。また、対応言語追加者向け API では、ソースコードを解析して抽象構文木を生成する処理と、言語ごとの抽象構文木を UNICOEN 上の共通した言語モデル上のオブジェクトにマッピングする処理について、言語非依存に再利用可能な処理を提供する。

本論文では、UNICOEN の有用性を確認するために、7 種類のプログラミング言語から共通の言語モデルを抽出して、さらに、UNICOEN 上で 2 種類のメトリクス測定ツールの実装を行った。既存の言語処理系および既存の測定ツールと比較して、UNICOEN においてはより少ないコード行数で対応言語を追加できること、また、新しいツールを実装できることを示した。

6 章では、UNICOEN の有用性を確認するため、UNICOEN を利用して複数言語に対応するアスペクト指向プログラミング (Aspect-Oriented Programming; AOP) 言語処理系 UniAspect を提案した。UniAspect は言語に共通するモジュール結合可能箇所に対応しており、言語非依存な形でモジュールを結合する箇所を指定できる。結合するモジュールの内容は言語ごとに記述する必要はあるが、Java と JavaScript 言語で構成されたオープンソースソフトウェアに対して、1 つのアスペクトによりモジュール化することに成功し、既存 AOP 処理系の組み合わせよりも、保守性の高い形でモジュール化できることを示した。

7 章では、結論として本論文をまとめるとともに、今後の課題を示した。

以上を要するに、本論文では、複数のプログラミング言語を利用したソフトウェア開発に対するソースコードリエンジニアリング適用における問題点を指摘して、それを解決する方法として、複数言語に対応したテストカバレッジ測定フレームワーク、および、ソースコードリエンジニアリングフレームワークの提案および構築を行った。評価実験を通して提案フレームワークの有用性を示した上、実際にフレームワークを用いた測定や解析、再構成のためのツール開発および新手法の提案を行うことで、提案フレームワークによって新しいリエンジニアリングツールの開発を支援できることを示した。このことは、ソフトウェア工学において得られた知見の実用化を支援する優れた貢献となっており、ソースコードリエンジニアリングを中心としたソフトウェア工学の発展に寄与するところが大きい。よって、本論文は、博士(工学)早稲田大学の学位論文として価値あるものと認める。

2013 年 2 月

審査員

主査	早稲田大学准教授	博士 (情報科学) (早稲田大学)	鷺崎	弘宜
	早稲田大学教授	工学博士 (早稲田大学)	深澤	良彰
	早稲田大学教授	工学博士 (東京大学)	上田	和紀
	東京工業大学准教授	博士 (情報科学) (早稲田大学)	首藤	一幸