

平成 22 年 3 月 31 日現在

研究種目：若手研究(B)  
 研究期間：2007～2009  
 課題番号：19700033  
 研究課題名(和文) ソフトウェア開発における変更の影響を測定するためのメトリクスに関する研究  
 研究課題名(英文) Research about metrics for measuring changes in software development

研究代表者  
 横森 励士 (YOKOMORI REISHI)  
 南山大学・情報理工学部・講師  
 研究者番号：40379152

## 研究成果の概要(和文)：

本研究では、大規模なソフトウェアの開発過程におけるソフトウェアの内部の構造の変化を理解するために、部品間の利用関係やその変化度を用いてその内部構造の変化を評価する手法を提案した。本研究を通じて、部品間の利用関係を評価して順位付けする手法(コンポーネントランク)が大規模な変更やソフトウェア内部で重要な役割を果たす部品の変更を補足するために有効であることや、ソフトウェア開発過程においてフレームワークとそれを利用する部品間の利用関係がどのように成長するかなどを明らかにすることができた。

## 研究成果の概要(英文)：

In this research, I suggested an evaluation method based on use relation between components and based on a degree of change of use relation during development, to understand the change of inside of software architecture for large software systems. Through the experiment, I confirmed that component rank, that is a ranking method based on use relation between components, is useful for detecting important updates, such as a large scale of future implementation, and changes or refactoring to core components, and so on. I also studied about the process of use relations' growth between framework components and application components during development.

## 交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2007年度	1,100,000	0	1,100,000
2008年度	600,000	180,000	780,000
2009年度	600,000	180,000	780,000
総計	2,300,000	360,000	2,660,000

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：ソフトウェア部品，利用関係解析，Java，オープンソースソフトウェアへの適用

## 1. 研究開始当初の背景

現在のソフトウェア開発においては、多くの人数で一つの大きなシステムを構築することが当たり前になっている。開発において

は、CVSなどのソフトウェアリポジトリを介して開発の成果物が共有および管理され、システムの開発を機能ごとに分けた上で様々な機能追加が分担してなされるというアプ

ローチが一般的になっている。

しかしこの機能追加を基本作業とした開発環境下では、それぞれの開発者が自分の担当する部分の管理は可能であったとしても、全体を統括する人間にとっては巨大なソフトウェアがどこまで完成しているかなどの進捗状況を確認することは非常に困難である。また、現状までに発生したシステム構築における重要な出来事を把握することもそれぞれの担当者の報告に頼らざるを得ない部分が発生しやすい。その場その場で場当たりのようなコントロールができて、全体としてプロジェクトを適切に管理することは非常に難しいタスクとなっている。

これらの問題点をもとに、CVSなどのソースコードや、メールの履歴、バグ管理情報の履歴などを可視化することで、定量的なメトリクスという形で開発者間の情報を共有しようとする試みが行われている。これらの取り組みは、基本的には有用であると考えられるが、現状ソースコードからの情報取得がLOCだけであるなど満足に情報を提供できているとは言いにくい。例えばソースコードのLOCなどの情報は簡単に手に入り、ある一定以上の情報的価値があることは広く知られている。ただし、それらの情報は単にソースコードの一面を表しているだけで、多面的な情報をもとに評価を行わないと、現状の認識を誤って行ってしまいう可能性が高くなる。実際には、ソースコードから得られる複数の情報をもとに、ソースコードに機能が追加されていく過程（ソフトウェアの成長過程）を総合的に判断するという作業が必要である。

## 2. 研究の目的

本研究は、ソフトウェア内部に存在する部品間の利用関係がソフトウェアの成長過程においてどのように変化するかということをも明らかにすることで、利用関係の変化に関するメトリクスをソフトウェアの成長過程を評価するための第2の視点の候補として確立することを目的としている。

これらのソフトウェア内部の構造の変化も、前述したLOCなどのソフトウェアから取得できるメトリクスと同様にソフトウェアの成長を観察する上での一つの視点として成り立つと私は考えている。今までソフトウェアの成長過程においてソフトウェアの成長をソフトウェア内部の関係の変化から捉えようとした研究はあまりない。様々な観点からソフトウェアの成長を総合的に判断するための環境を構築するために、ソフトウェア内部の関係の変化を調査することは欠かせないものになると考えている。

さらに、ソフトウェア内部においては、部品間の関係として、クローンの存在など、

様々な関係を定義することができる。今後それらの関係に基づいてソフトウェアの成長を評価する際にも、本研究などで蓄積した評価についてのノウハウを生かすことができると考えている。

## 3. 研究の方法

本研究の研究期間においては、利用関係の変化に関するメトリクスを用いて、次に示す2種類の実験を行った。これらの実験を通じて、ソフトウェアが開発されていく過程でどのように利用関係が構築されていくかを明らかにする。

実験1：この実験では、開発履歴におけるそれぞれの変更作業を対象として、利用関係に基づく評価順位が変更作業の前後でどのくらい変動するかを評価し、順位変動が大きく起こる変動がどのような内容の変更作業であるかを調査した。

利用関係に基づく評価順位付け手法として、コンポーネントランクを利用している。それぞれの部品について、変更前と変更後の部品の順位を測定し、全体としての順位の変動度を求め、それをその変更における利用関係の変動度と定義した。

定義した変動度をもとに、Javaで開発された複数のオープンソースプロジェクトの開発履歴を取得した上で、取得した開発履歴に適用しそれぞれの変更の更新度を抽出した上で変動度の高い変更を抽出し、それらの更新作業中における更新内容を確認した。

実験2：具体的にどのような形で部品間の利用関係が成長し、コンポーネントランクがどのように変動するかを調べることを目的としている。フレームワークにおけるライブラリとそれを利用する部品間の利用関係に着目し、開発過程においてリリースされたそれぞれのバージョンにおいて、ライブラリと利用部品間の利用関係を抽出しその変化を確認した。

この実験では、前実験で利用したコンポーネントランクのほかに、部品間の利用関係を部品グラフとしてモデル化し、それぞれの部品における入力辺、出力辺を求めている。実際のオープンソースプロジェクトを対象に、それぞれのバージョンにおけるコンポーネントランクやバージョン間の変動度を計算するとともに、アプリケーション側からフレームワーク中の部品への利用関係だけを抽出し、そのような利用関係が開発の進行により機能が追加されていく中で、どのように成長するかを確認した。

#### 4. 研究成果

##### 実験1の研究成果：

実際に12のオープンソースプロジェクトを対象に実験を行った。ここでの実験では更新日毎に更新を分け、分析の粒度としている。

- ・開発が進行するにつれて、変動度の値がだんだん減少していった。開発初期の段階では、必要な機能の実装により利用関係の変化が起こりやすく、高い変動度の値を示すが、ソフトウェアの機能が充実するにつれて変動度の値は徐々に下がっていく。このことから、開発の進行により必要な機能やデータが固定化されるのと同様に、利用関係も固定化されていくことが推測できる。
- ・各更新において最も順位が変動した部品の変動幅も確認したが、こちらは開発の時期にあまり影響されない値であった。このことは、更新の間隔を日としたことで一回一回の更新の規模は開発の時期を通じてあまり変わらないことを示していると考えられる。
- ・前後の更新と比べて相対的に高い値を示している更新を抽出した。その変更内容をCVSwebなどの情報をもとに筆者が調査したところ、抽出された更新には単なるバグ修正はほとんど見られず、そのほとんどは以下に示す4種類の変更に属する更新であった。利用関係に大きな変更をもたらす更新だけでなく、中心的な機能を果たす部品への更新も変動度が高い更新として見つけ出すことが容易になると考えられる。

(1) 大規模な機能追加

(2) コア部品への変更を含む機能追加

(3) システムを横断する機能の追加

(4) 作り直しやリファクタリングなどのコード整理

- ・コンポーネントランクは利用関係上重要な部品を上位に順位付けでき、更新によって重要な部品が発生した場合は、更新後のその部品の順位もそれに合わせて上昇している。
- ・コードの変更量を用いることで、コードの変更量の大きい更新を抽出することができるが、リファクタリングやコード整理、コア部品などへの機能追加などを含んでいるソフトウェアに大きな影響をもたら

す更新は、必ずしもコードの変更量は大きくない。

- ・開発の初期は、機能追加により上位の順位が変わりやすいが、中期以降において上位の部品の順位の変動が激しい時点では全てコードの作り直しや整理が行われていた。通常このような場面ではその後の拡張性を考慮して作り直すことが多く、更新後に順位の変動が止まっていることから、これらの大規模な作り直しや整理が成功に終わったことが窺える。

これらの実験から得られた知見を通じて、開発履歴の分析において、LOCなどの一般的なメトリクスだけでなく、利用関係の面からも分析を行うことが有益であることを示した。

##### 実験2の研究成果：

この実験では、実際にリリースされたバージョンごとに利用関係を分析し、それらのバージョン間での利用関係の違いを調査した。

- ・実験1の場合と同様に、大規模な変更があったバージョンにおいて、コンポーネントランクの順位が全体的に大きく変動していた。また、サブシステム毎に変動度を確認したところ、変更があったサブシステムのみにおいて大きく順位が変動することがわかった。
- ・部品グラフのそれぞれの頂点におけるフレームワーク部品への出力辺は、バージョンが進むごとに増加するが、一つの頂点からの出力辺の最大値は開発期間を通じてあまり変化はなかった。おそらく、ある程度の大きさになった時点で開発者らによってより細かい部品に分解されるためであると思われる。
- ・部品グラフのフレームワーク部品における入力辺の数は、バージョンが進むごとに増加し、また一つの頂点への入力辺の最大値は開発期間を通じて増加している。これは出力辺の場合と違い、そのような部品は繰り返し利用されることが推奨されるためであろう。
- ・コンポーネントランクは、実装された機能に大きく影響を受けて変動する。中心的な役割を果たすデータ構造や機能などを抽出するのに有効である。また、同じような構造を持つ似た機能の部品同士は、それぞれ同じような順位変動になりやすい。

これらの実験から得られた知見を通じて、

利用関係を分析することで、ソフトウェア内部の構造において大きな変更があった更新作業を見つけやすくなると考えられる。入力辺、出力辺などの表面上の数値の変化を分析するだけでも、有益な情報が得られると思われるが、コンポーネントランクも用いることで間接的な利用関係も考慮して評価できる。

まとめ：

以上の実験を行い、論文誌に実験1の内容で一稿、国際会議に実験2の内容で一稿、それぞれ論文を公刊した。また、国内のシンポジウムにおいて、コンポーネントランクを題材として企画セッション内で発表を行った。

本研究で得られた成果を通じて、ソフトウェアの成長を理解するためには、複数の種類の情報を用いて、多面的に分析することが極めて効率的であることを確認した。将来的には、多種多様なメトリクスを分析可能な環境を構築した上で、ソフトウェアの開発の管理が行われるようになると思われ、その際に利用関係等の情報も分析対象となるであろう。また、ソフトウェア部品間に存在する様々な関係に基づいて分析を行う際に、本研究などで蓄積した評価についてのノウハウを生かすことができると考えている。

## 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1件)

- ① 横森 励士, 野呂 昌満, 井上 克郎, 開発履歴中のソースコードを対象とした更新の重要度を評価する手法の提案, 電子情報通信学会論文誌 D-I, 査読あり, VolJ91-D-I, No.4, pp 945-955, 2008.

[学会発表] (計2件)

- ① Reishi Yokomori, Harvey Siy, Masami Noro, Katsuro Inoue, Assessing the Impact of Framework Changes Using Component Ranking, in proceedings of 25th IEEE International Conference on Software Maintenance (ICSM2009), 査読あり, pp189 -- 198, Edmonton, Canada, 2009. 2009年9月23日ホテルマクドナルド(カナダ・エドモントン)にて発表.
- ② 横森 励士, ソフトウェア開発を支援するソフトウェア部品検索システム, ソフトウェアエンジニアリング最前線 2008, ソフトウェアエンジニアリングシンポジウム 2008, pp.25-32, 2008. 2008年9

月3日東洋大学白山キャンパスにて発表.

## 6. 研究組織

### (1) 研究代表者

横森 励士 (YOKOMORI REISHI)  
南山大学・情報理工学部・講師  
研究者番号：40379152