

利用関係の一致に基づくソフトウェア部品のクラスタリング手法 — クラスタ形成の契機となる利用関係の調査 —

2018SE063 小田梨乃 2018SE081 曾我柚月

指導教員：横森励士

1 はじめに

近年は、長期間にわたる保守作業を受けているソフトウェアは非常に多く、そのようなソフトウェアは数多くの修正や機能追加などの措置が施されている。このようなソフトウェアの内部構造が複雑になっていく過程を色々な観点から分析することが、構造の複雑化の度合いを緩和し、それによってソフトウェアの理解度を保つ上で必要になると考えられる。我々の研究グループでは、ソフトウェア部品の利用先の類似性に基づいて、ソフトウェア部品を分類する手法 [1] を提案し、バージョン毎に適用した際にどのように分類結果が変化するかを [2] で調査したが、得られた特徴の一般性についてはまだ十分に調査できていない。本研究では、様々なソフトウェアに対して、バージョン毎に分類手法を適用し、バージョンの進行の途中で形成される部品群に着目する。新しく追加される部品群がどのくらい存在するかや、部品群形成のきっかけとなる利用関係がどのように追加されるかなどを調査する。ソフトウェアの保守において、開発者のどのような行動や決断が部品群形成に関わっているかを調査する。

2 背景技術

2.1 ソフトウェア部品と部品グラフ

一般にソフトウェア部品とは、その内容をカプセル化したうえで、環境においてそれらを交換可能な形で実現したシステムモジュールの一部である。本研究では、開発者が再利用を行う単位として Java のソースファイルをソフトウェア部品とみなす。ソフトウェアは複数のソフトウェア部品で構成されると考えることができる。継承、変数の宣言、インスタンスの作成、メソッドの呼び出し、フィールド参照など「ある部品が他の部品を利用する」、「他の部品からその部品が呼び出される」などといった関係を定義して部品グラフ上で表現することができる。

2.2 利用部品の類似度に基づくソフトウェア部品分類手法について

ある部品で目的の機能を実現する場合、他の部品で提供されている機能やライブラリを利用しながら目的となる機能を実現する。横森らは [1] で、利用先が一致している割合が高いほどその部品同士は扱う対象が似ていたり、役割が似ている部品になると考えた。[1] では、ソフトウェアの利用先がどれだけ一致しているかから樹形図を作成し、分類する手法を提案した。実験では、得られた部品群内の部品は関連性を強く認識できる部品の集合であった。

安藤らは [2] で、[1] の手法をソフトウェアの各バージョンに適用した際に、得られた部品群がどのように変化するかについての調査を行った。[2] の実験の結果は、ライブラリ部品の利用の一致度に基づく類似度で分類した場合、バージョンが進行しても部品群は変化しなかった。その一方でソフトウェア内で定義された部品の利用の一致度に基づく類似度で分類した場合、バージョンの進行とともに部品群が多く形成された。途中のバージョンで追加された部品がソフトウェア内を徐々に整理していると考えられる。提案手法により、ソフトウェア内の利用関係がどのように整理されていくかを追跡できるようになると考えられる。バージョン間の変化を分析する際に、ソフトウェア内の関係を整理するような部品が、どのように追加されるかということは、ソフトウェア内の構造の進化を利用関係の観点から説明するうえで、有用な情報だと考える。

3 クラスタ形成の契機となる利用関係の調査

3.1 研究の目的

[2] で分析した結果からは、ソフトウェア内で定義された部品の利用の一致度に基づいて分類した場合、新しくできた部品は新規に追加された部品からなる物だけでなく、既存の部品に利用関係が新たに加わり、新しく部品群となったものも多くみられた。ただし、分析したソフトウェアは1つで、この傾向が他のソフトウェアに対しても成り立つのかは分かっておらず、どのような部品がどのような目的で追加されたときに、既存の部品が部品群化するのかはよくわかっていない。

3.2 研究の課題

本研究では、様々なソフトウェアに対して [1] の手法を適用した場合に、新しく形成される部品群がどのような部品群であるかを調査し、バージョン間で部品群形成のきっかけとなる利用関係を持つ部品がどのように追加されるかを調査する。また、開発者のどのような行動が部品群形成に影響するかを調査する。知見から、バージョン間の分析において、利用関係の観点からという新たな分析の視点を追加できるようになり、より多面的にソフトウェアの成長を観察できるようになると考える。

3.3 リサーチクエスト

以下の3つのリサーチクエストについて分析する。

1. 最新のバージョンで得られた部品群のうちどれだけが過去のバージョンで確認できるか？

ソフトウェア内で定義された部品への利用関係で分類

した際に、途中で追加される部品によってどれくらいの部品群が新しく作られているのかを調査する。

最初から部品群が形成されている部品群が存在する一方で、途中の機能追加の際に利用関係が追加され、部品群ができる場合も考えられる。ソフトウェアを管理していくうえで、追加されたソフトウェア部品がどの程度部品群の形成に関わるのか調査する。

2. 新規形成された部品群はどのような部品の集合か？

新規に追加された部品が新しく部品群を形成するのか、既存の部品がまとまり新たな部品群を形成するのかを調査する。

3. 部品群形成のきっかけとなるのはどのような利用関係が追加されたときか？

既存の部品がまとまってできた部品群について、部品群が生成された更新の前後の利用間に着目し、どのような利用関係の変化が起きたかを調査する。ソフトウェア理解の面から、どのような利用関係を追加したときに部品群が形成されるのかを調べる。

4 評価実験

4.1 実験の概要と手順

3.3で提示したリサーチクエストに基づいて、実際のソフトウェアを分析対象として適用実験を行う。5つ以上のバージョンを持つソフトウェアを選出し、分析区間を決定し、最初、途中、最後のバージョンのソースコードや成果物を入手する。

1. 最新のバージョンにおける部品群の入手

最後のバージョンを対象に分析し、樹形図を得る。葉となる部分から類似性を判断し、類似性を持つ部品が最大限含まれるように部品群を求める。

2. 最初のバージョンにおける部品群の入手

最初から存在する部品群を抽出する。最初のバージョンで得られた部品群は分析対象から外す。

3. 途中で作られた部品群の調査

最初のバージョンで存在しない部品群について途中のバージョンを分析し部品群ができたバージョンをそれぞれ求める。その前後のバージョンにおいて、部品群ができた後、できる前の共通利用部品を調査し、共通利用部品について部品群が出来る前後での違いを見つける。

4.2 評価実験 1

実験 1 では、Art of illusion を対象として分析する。Art of illusion は 3 次元コンピュータグラフィックのためのソフトである。Art of illusion における ver1.0 から ver1.6 の各バージョンの部品数を表 1 に示す。

Art of illusion の ver1.6 を最新のバージョンとし、樹形図を得て部品群を確定する。ver1.6 で得た樹形図を図 1 に

表 1 Art of illusion での各更新における部品数

	総部品数
ver1.0	281
ver1.1	291
ver1.2	309
ver1.3	324
ver1.4	334
ver1.5	355
ver1.6	360

示す.ver1.6 では 30 個の部品群が得られた。次に、ver1.0 を最初のバージョンとし、樹形図を得て部品群を得る。ver1.0 で得られた樹形図を図 2 に示す。図 2 では 11 個の部品群が保たれており、図 1、図 2 でそれらの部品群を黒色で示す。残りの 19 個の部品群は途中のバージョンで構成されたことが分かる。

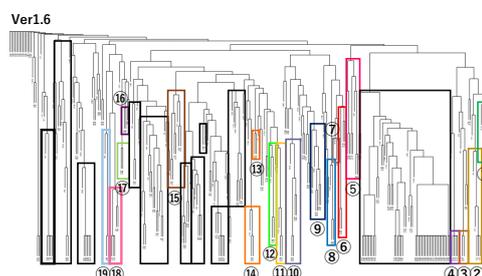


図 1 最新 (ver1.6) において得られた部品群。

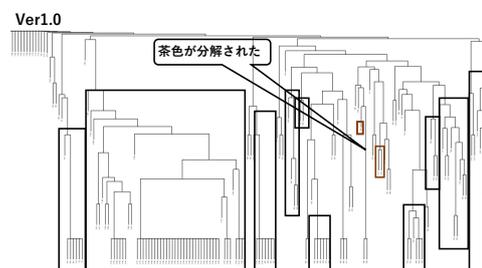


図 2 最初 (ver1.0) において得られた部品群

途中で形成される 19 個の部品群を調査したところ、12 個の部品群が途中のバージョンで新たに追加された部品によって部品群化した一方で、残り 7 個の部品群は、既存の部品が利用関係の追加により、新たに部品群化した事例であった。以下では、既存の部品がバージョンの更新により部品群化したパターンについて 4 つの部品群を紹介する。どの部品群もバージョンの更新により新たな利用関係が加わった。

- ver1.2 と ver1.3 の間で図 1 の (⑤) の部品群が形成された。この部品群は、オブジェクトの性質や、明るさの幅やカラーなどを表す部品群である。ソフトウェア内の共通利用部品を比較したとき、新たな利用関係として、RGBColor クラスを共通して利用するようになったことが分かった。このク

ラスは光の三原色「赤、緑、青」で色を表すために使用されるクラスである。

- ver1.4 と ver1.5 の間で図 1 の (⑥) の部品群が形成された。この部品群は、ダイアログボックスに関する機能の部品群である。ソフトウェア内の共通利用部品を比較したとき、Translate クラスを共通して利用するようになった。このクラスは様々な言語に翻訳できるようにするための機能である。
- ver1.5 と ver1.6 の間で図 1 の (⑬) の部品群が形成された。この部品群は、script のパッケージに属する部品群である。ソフトウェア内の共通利用部品を比較したとき、ObjectInfo クラスを共通して利用するようになったことが分かった。このクラスは Object3D 内のオブジェクトの役割を記述するクラスである。
- ver1.3 と ver1.4 の間で図 1 の (⑱) の部品群が形成された。この部品群は、HDR 画像を生成し、平均画像を計算する機能の部品群であると考えられる。ソフトウェア内の共通利用部品を比較したとき、ComplexImage クラスを共通して利用するようになった。このクラスは各ピクセルにオプションの追加の浮動小数値を使用して画像を格納する機能である。

4.3 評価実験 2

実験 2 では、Filebot を対象として分析する。Filebot は映画やアニメなどの名前を変更したり、字幕をダウンロードしたりするためのソフトである。Filebot における ver2.6 から ver1.9.8 の各バージョンの部品数を表 2 に示す。

表 2 Filebot での各更新における部品数

	総部品数
ver1.9.8	272
ver2.0	273
ver2.3	290
ver2.4	298
ver2.6	312

Filebot では、ver2.6 を最新のバージョンとし、樹形図を得て部品群を確定する。ver2.6 で得られた樹形図を図 6 に示す。ver2.6 のバージョンでは 36 個の部品群が得られた。次に、ver1.9.8 を最初のバージョンとし、樹形図を得て部品群を得る。ver1.9.8 で得られた樹形図を図 7 に示す。図 7 では 12 個の部品群が保たれており、図 6、図 7 でそれらの部品群を黒色で示す。残りの 24 個の部品群は途中のバージョンで構成されたことが分かる。

途中で形成される 24 個の部品群を調査したところ、5 個の部品群が途中のバージョンで新たに追加された部品によって部品群化した一方で、残りの 19 個の部品群は、既存の部品が利用関係の追加により、新たに部品群化した事例であった。既存の部品がバージョンの更新により部品群化

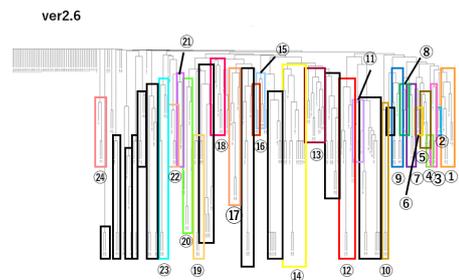


図 3 最新 (ver2.6) において得られた部品群

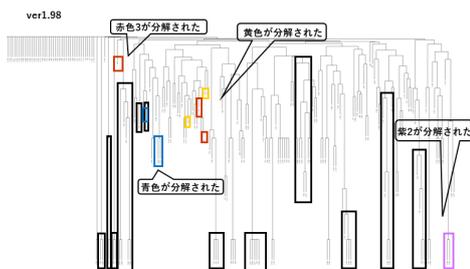


図 4 最初 (ver1.9.8) において得られた部品群

したパターンについて 4 つの部品群を紹介する。どの部品群もバージョンの更新により新たな利用関係が加わった。

- ver2.0 と ver2.3 の間で図 3 の (③) の部品群が形成された。この部品群は、オブジェクトの性質や、字幕に関する部品群である。ソフトウェア内の共通利用部品を比較したとき、新たな利用関係として、SubtitleProvider クラスを共通して利用するようになったことが分かった。このクラスは字幕を提供する機能である。
- ver2.0 と ver2.3 の間で図 3 の (④) の部品群が形成された。この部品群は、ファイルのコピー、移動などの一般的な操作を行う機能の部品群である。ソフトウェア内の共通利用部品を比較したとき、FileInfo クラスを共通して利用するようになった。このクラスはファイルのコピー、移動、名前変更など一般的な操作を行う機能である。
- ver2.0 と ver2.3 の間で図 3 の (⑱) の部品群が形成された。この部品群は、名前の変更やデータの分析などに関する部品群である。ソフトウェア内の共通利用部品を比較したとき、TransferablePolicy クラスを共通して利用するようになったことが分かった。このクラスは柔軟性を持ったライブラリを作ることが可能なクラスである。
- ver2.0 と ver2.3 の間で図 3 の (㉓) の部品群が形成された。この部品群は、動画に関する部品群であると考えられる。ソフトウェア内の共通利用部品を比較したとき、Movie クラスを共通して利用するようになった。このクラスは動画を制御したり情報を得たりする機能である。

5 考察

5.1 最新のバージョンで得られた部品群のうちどれだけが過去のバージョンで確認できるか?

評価実験 1 では、最新のバージョンで 30 個の部品群が形成されていた。30 個の部品群のうち、11 個の部品群が最初のバージョンでも存在しており、19 個の部品群が更新の途中で新たに形成された部品群であった。評価実験 2 では、最新のバージョンで 36 個の部品群が形成されていた。36 個の部品群のうち、12 個の部品群が最初のバージョンでも存在しており、残りの 24 個の部品群が更新の途中で新たに形成された部品群であった。評価実験 1,2 からは、分析期間の最新のバージョンで抽出できた部品群の 3 から 4 割程度のみが初期のバージョンで確認できており、残りの部品群は、途中のバージョンで新たに形成されたものであった。ソフトウェア内での部品の利用関係を用いた分類はある程度機能追加が行われて、利用関係が成熟し変化が落ち着いた後で、行われるべきだと考えられる。

5.2 新規形成された部品群はどのような部品の集合か?

バージョンの進化により新たに形成される部品群は、「新たに追加される部品」で構成される部品群と、「新たに追加される部品を利用する部品」で構成される部品群に大きく分けられることが分かった。評価実験 1 では、新たに追加される部品で構成される部品群が 12 個で、新たに追加される部品を利用する部品で構成される部品群は 7 個であった。評価実験 2 では、新たに追加される部品で構成される部品群が 5 個で、新たに追加される部品を利用する部品で構成される部品群は 19 個であった。追加された部品群におけるこれらの構成比はそれぞれで異なり、プロジェクトや追加の状況によって変化すると考えられる。

5.3 部品群形成のきっかけとなるのはどのような利用関係が追加されたときか?

共通利用部品の追加によって新たに形成された部品群を調査したところ、今までの部品の機能をサポート、強化する機能や、既存の部品を整理してより使いやすくした機能に関する利用関係を新たに共通して利用するようになった部品であった。新しく追加する機能を利用する部品をどのように決めるかによって、利用部品の一致する範囲が決まるので、開発者がそれらを決定することが、部品群形成に影響すると考えられる。

5.4 考察のまとめ

部品群が形成される更新の前後で得られる共通利用部品の变化や相違点を確認した結果、部品群形成によって追加される機能は、既に存在する部品を補強するようなものであった。ソフトウェアの部品が新しく追加されることで、ソフトウェア内で部品の利用関係に関する情報が定義されていき、部品を利用する環境が徐々に整備される。部品を利用することで、部品が提供する機能により関連するよう

になり、分類手法による部品群が類似性を示しやすくなると考えられる。

6 まとめと今後の課題

本研究では、バージョンごとに利用先などの情報の一致度を用いて階層的クラスター分析を行い樹形図を得ることで、バージョン間の途中で部品群形成のきっかけとなる利用関係をもつ部品がどのように追加されるかを調査した。評価実験では、関連性を認識できる部品の集合は、主に新たな利用関係の追加によってより詳細な機能が加わることで得られていくことがわかった。また、ソフトウェア内の利用関係で分類する手法は、ソフトウェア内を整備するために遅れて追加される部品があるため、ある程度時間が経った後でこの手法を適用するのが 1 番良いことがわかった。今回調べたプロジェクトでは、既存の部品を補強や強化する新たな部品の利用関係の追加によって部品群化する事例が多く存在したが、これらの事例以外に部品群形成のきっかけとなる事象が存在するのかを他の事例を適応しながら考えていく必要がある。そのためにより複数の他のプロジェクトに対して実験を行い、違う知見が得られるのか、また、これらの実験で得られる結果を具体的にどのように役立てていくことができるのかを今後考えていく必要があるといえる。

参考文献

- [1] Reishi Yokomori, Norihiro Yoshida, Masami Noro, Katsuro Inoue : "Use-Relationship Based Classification for Software Components" , Proceedings of the 6th International Workshop on Quantitative Approaches to Software Quality (QuASoQ 2018) , pp.59-66 , 2018.
- [2] 安藤勇人, 白山遼祐, 竹市豊 : "利用関係の一致度に基づくソフトウェア部品分類手法 -ソフトウェア更新によって分類結果がどのように変化するかについての考察-" , 南山大学 理工学部 2020 年 卒業論文, 2021.