

# 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 — 数百クラス規模のソフトウェアに適用した場合の有効性の評価 —

2018SE059 西川樹 2018SE099 渡部広夢 2018SE101 渡辺翼

指導教員：横森励士

## 1 はじめに

近年のソフトウェアは大規模化しており、ソフトウェアを構成する部品数も増大している。このような環境下で、部品間の類似性などを利用してソフトウェアの構成要素を効率よく把握することが求められる。我々の研究グループでは、各ソフトウェア部品が利用している部品の集合がどれくらい一致しているかを元に距離行列を作成し、クラスタリングを行うことで類似部品群を求める手法を提案した。評価実験の結果では、分類結果として得られた部品群のほとんどにおいて「部品群中の部品の共通点」と「共通で利用している部品の役割」が一致していた。提案手法は部品群内の部品の理解に共通で利用している部品を活用できることを確認した。しかし、数百クラス規模のソフトウェアに対しての評価実験が行われておらず、その観点からの評価が必要である。

本研究では、提案手法を数百クラス規模のソフトウェアに対して適用する理解の支援の手法として提案手法の実利用を考えた際にどのような分類結果となり、どのような問題が生じるかを調査する。具体的には、得られた部品群内の部品が共通性を有しているか、共通して利用している部品が部品の理解に貢献できるかを調査する。さらに、問題が生じて精度が低下する場合、どのような原因が考えられるかを調査し、実利用を考えた場合に提案手法について考えるべきことについて考察する。

## 2 背景

### 2.1 ソフトウェア部品と部品グラフ

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実現する環境において交換可能な形で配置できるようにしたシステムモジュールの一部を指す。ソースコードを記述している各ファイルを部品とみなし、全部品を構成要素とする部品グラフを構築する。部品グラフ上の頂点は各部品を表し、有向辺は部品間の関として利用関係を表現する。ある部品 A が他の部品 B を利用している場合、頂点 A から頂点 B への有向辺で表現する。

### 2.2 利用関係の一致度に基づくソフトウェア部品の分類手法

横森らは、ソフトウェア部品間の利用関係の一致度から各部品間の類似度を求め、類似度から距離行列を計算し、階層的クラスタリングによって樹形図を作成し、樹形図の葉として得られた部分から類似部品群を得るというソフトウェア部品の分類手法を提案した [1]。評価実験では、得ら

れた部品群内の部品の多くが扱う対象が同じである、役割が似ているなどの共通性を持つことが確認されている。

川瀬、橋本らは、分類結果として得られた部品群において、部品群中の部品の共通点が共通して利用している部品の役割と関連があるかの調査と、多くの部品から利用されている部品についてそれを利用している部品がどう分類されるべきかを調べた上で、その集合が樹形図にどう表れるかを調査した [2]。評価実験では、分類結果として得られた部品群のほとんどにおいて部品群中の部品の共通点と共通で利用している部品の役割が一致していた。

この手法は、ひとまとめにすべき部品などを一度に提示することができるなど、保守におけるソフトウェア理解の効率化を支援できると考えられる。コードブラウザ上でその部品と類似している部品を提示したり、同一の部品群に属している部品が共通して利用している部品を提示することで、似たような部品をまとめて理解するための環境ができると考えられる。

## 3 数百クラス規模のソフトウェアに対するソフトウェア分類手法の適用

### 3.1 研究の動機

過去の研究では [1][2]、数十クラス程度の小規模なソフトウェアに対して分類手法を適用した。クラスター分析の結果、樹形図の葉となる部分に共通性をもつ部品群が多く得られること、部品群内の「部品の共通点」と「共通して利用している部品の役割」が関連していることを確認した。実利用を考えた場合、より大規模なソフトウェアに対しても同じことがいえるかを検証する必要があると考えられる。より大規模なソフトウェアについて適用した場合でも有効性を示すとともに、生じる問題は何かを把握することを研究課題とする。

### 3.2 研究の内容

本研究では、利用関係の一致度に基づくソフトウェア部品の分類手法を、数百クラス程度の規模のソフトウェアに対して適用する。その場合でも、得られた部品群内の部品が共通性を有しているかや、共通して利用している部品が部品群内の部品の理解に貢献できるかについて調査する。有効性が維持できているか、どのような問題が生じるかを検証する。

表 1 分析対象アプリケーション

	version	ファイル数
Art of Illusion	16	313
Filebot	2.0	238
bacnet	1.0	279

### 3.3 評価項目

数百クラス程度の中規模なソフトウェアを対象として、以下のリサーチクエストを設定し評価実験を行う。

1. どの程度の部品が部品群に含まれるか？  
ある程度の割合で利用部品がない部品や利用関係が一致しない部品が存在すると考えられるので、そのような部品が占める割合を確認する。
2. 部品群はいくつできるか？  
より大規模なソフトウェアに対して適用した場合、大きな部品群が得られるのか、細分化したものが得られるのかについて、部品群の数や部品群 1 つあたりの部品数を調査する。
3. 機能的な共通点を持たない集合はいくつできるか？  
ソフトウェアの規模が大きくなると、利用する機能の数は増える。主機能と関連しない周辺の機能の利用により、利用関係が一致することで形成される部品群も増えると考えられる。共通利用部品を持つが、機能的な共通点が見つからない集合がどれくらいできるのかを調べる。
4. それぞれ部品群中の部品の主な機能についての共通性が、共通で利用している部品の役割と一致するか？

## 4 評価実験

### 4.1 実験の内容と分析対象

本研究では、表 1 で表す Art of Illusion と FileBot, BACnet for JavaJava の 3 つのアプリケーションに対して適用実験を行った。Art of Illusion は 3D モデリングやレンダリング、アニメーションを付けられるソフトウェアであり、Ver1.6 は 313 個のソースファイルで構成される。FileBot は映像ファイルの名前を変更したり、字幕をダウンロードするソフトウェアであり、Ver2.0 は 238 個のソースファイルで構成される。

BACnet for Java は Java 用に作成された BACnetI/P や MS/TP プロトコルを実装するソフトウェアであり、Ver1.0 は 279 個のソースファイルで構成される。

### 4.2 分析の結果

それぞれのアプリケーションに対して分類手法を適用し、図 1, 図 2 のような樹形図を作成し、部品群の抽出を行った。以下ではそれぞれの分類結果を紹介する。

#### 4.2.1 適用結果 (Art of Illusion)

分類を行った結果、313 クラス中 32 クラスが互いに共通利用部品を持たず、約 1 割の部品が今回の分類では分類不能であった。また、74 個が葉の部分から部品群を作成するアプローチでは分類できなかった。残りの、207 部品からなる 55 個の部品の集合を確認できた。そのうち機能的な共通点を確認でき部品群とみなすことができる集合は 49 個 195 部品で残りの 6 個 12 部品は機能的な共通点を確認できなかった。部品群については 1 つあたりの部品数は 4.0 個であった。次に 49 個の部品群について、部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致しているかどうかを確認したところ、48 個の部品群において、それぞれ部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致していることを確認できた。残りの 1 個は部品群中の部品の主な機能についての共通性と共通で利用している部品の役割との間で関連性がなかった。例として、PanelDialog, MessageDialog, EntriesDialog, ComponentsDialog, ColorChooser, TransformPointsDialog, TransformDialog, PreferencesWindow, ImagesDialog で構成される集合を紹介する。PanelDialog, MessageDialog, EntriesDialog, ComponentsDialog, ColorChooser, TransformPointsDialog, TransformDialog, PreferencesWindow, ImagesDialog は共にダイアログの表示機能を持ち、共通点があると言える。だが、共通利用部品の Translate は言語の変換機能なので、部品の主たる機能における共通点と共通利用部品には関連性がないと言える。

#### 4.2.2 適用結果 (FileBot)

分類を行った結果、238 クラス中 68 クラスが互いに共通利用部品を持たず、約 3 割の部品が今回の分類では分類不能であった。また、38 個が葉の部分から部品群を作成するアプローチでは分類できなかった。残りの 132 部品からなる 44 個の部品の集合を確認できた。そのうち機能的な共通点を確認でき部品群とみなすことができる集合は 38 個 119 部品で残りの 6 個 13 部品は機能的な共通点を確認できなかった。部品群については 1 つあたりの部品数は 3.1 個であった。次に 38 個の部品群について、部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致しているかどうかを確認したところ、すべての部品群において、それぞれ部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致していることを確認できた。

#### 4.2.3 適用結果 (bacnet)

分類を行った結果、279 クラス中 14 クラスが互いに共通利用部品を持たず、5% の部品が今回の分類では分類不能であった。また、46 個が葉の部分から部品群を作成するアプローチでは分類できなかった。残りの 219 部品からな

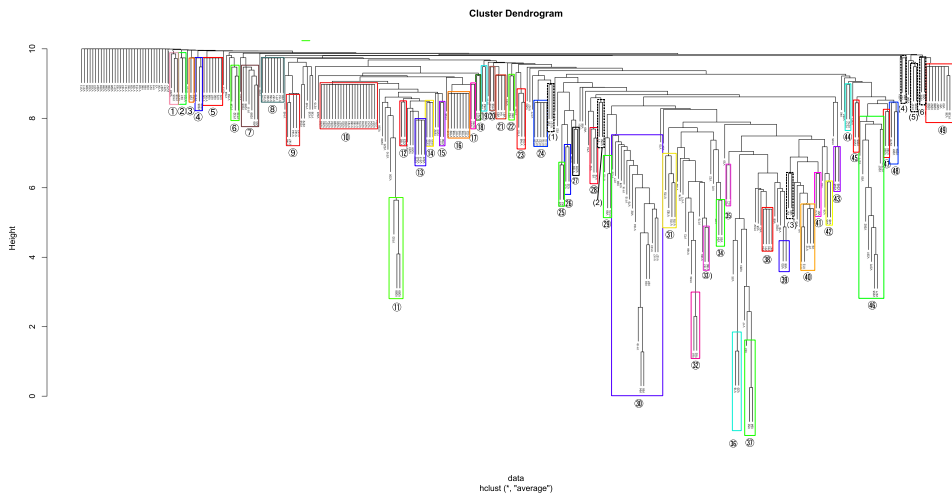


図 1 (Art of Illusion) 利用先の一致度で分類した結果の樹形図

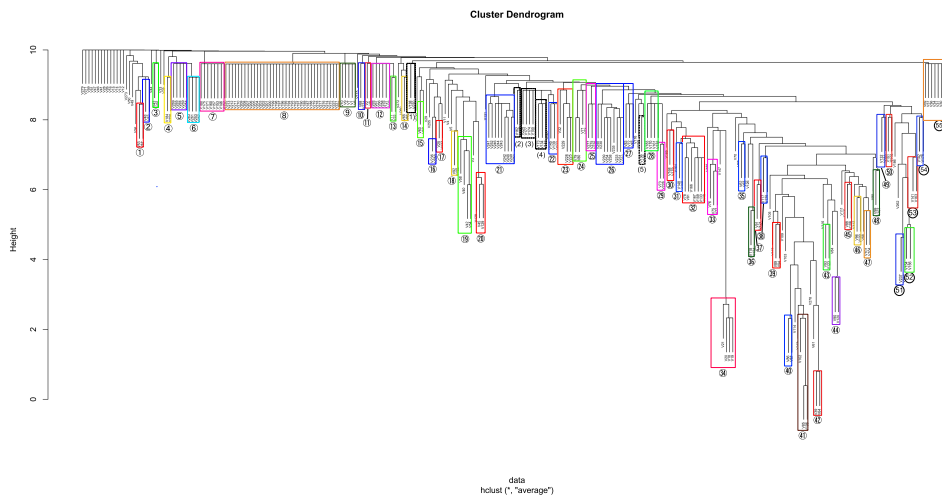


図 2 (bacnet) 利用先の一致度で分類した結果の樹形図

表 2 部品群の抽出結果

	ファイル数	部品群以外の部品数	部品群の数	部品群の部品数	関連なし集合の数	集合の部品数
Art of Illusion	313	32+74	49	195	6	12
Filebot	238	68+38	38	119	6	13
bacnet	279	14+46	55	205	5	14

表 3 「部品群内の部品の機能の共通点」と「共通利用部品」の関連性の判定結果

	部品群数	関連あり	関連なし
Art of Illusion	49	48	1
Filebot	38	38	0
bacnet	55	51	4

る 60 個の部品の集合を確認できた。そのうち機能的な共通点を確認でき部品群とみなすことができる集合は 55 個 205 部品で残りの 5 個 14 部品は機能的な共通点を確認できなかった。部品群については 1 つあたりの部品数は 3.7

個であった。次に 55 個の部品群について、部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致しているかどうかを確認したところ、51 個の部品群において、それぞれ部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致していることを確認できた。残りの 4 個は部品群中の部品の主な機能についての共通性と共通で利用している部品の役割との間で関連性がなかった。例として、ConfirmedRequestService、UnconfirmedRequestServiceで構成される部品群を紹介する。ConfirmedRequestService と UnconfirmedRequestService は共に要求の管理機能を持ち、共通点があると言える。だが、共通利用部品の Lo-

calDevice は無線システムに関する機能なので、部品の主たる機能における共通点と共通利用部品には関連性がないと言える。

## 5 考察

### 5.1 リサーチクエスチョンごとの考察

リサーチクエスチョンに沿って、分析対象のソフトウェアに対して適用した場合の結果を考察する。が

1. どの程度の部品が部品群に含まれるか  
いずれの場合も全体の 5~7 割程度の部品が部品群に含まれて、他の部品と類似性を示していた。[2] の分析結果でも部品群に含まれる部品は全体の 6 割程度でソフトウェアの規模にかかわらず分類できる部品の割合は一定であると考えられる。
2. 部品群はいくつできるか  
いずれの場合も部品群 1 つあたりの部品数は 3~4 個程度であった。[2] の分析結果でも、部品群 1 つあたりの部品数は 4~5 個程度であった。ソフトウェアの規模と得られる部品群の 1 つあたりの部品数は関連が薄く、その部分品群の数が増えると考えられる。
3. 機能的な共通点を持たない集合はいくつできるか  
いずれの場合も、10% から 15% 程度の割合の部品の集合が主たる機能ではなく、周辺の機能の利用で集合化する事例が得られた。[2] のときと比較すると、このような事例の割合は一割程度で出現しており、規模が大きくなるにつれて、部品群の数が増え、この傾向を満たす集合が出現しやすくなると考えられる。
4. それぞれ部品群中の部品の主な機能についての共通性が、共通で利用している部品の役割と一致するか  
いずれの場合も 93% 以上の部品群が部品群中の部品の主な機能についての共通性と共通で利用している部品の役割が一致しており、全ての部品群が関連性を示したケースもあった。[2] でも同じ傾向を示しており、この性質はほとんどの部品群で成り立つ性質であると考えられる。

### 5.2 考察のまとめ

これらの結果から分析対象の規模が大きくなることで、分類されて部品群に属する部品の割合や、部品群の 1 つあたりの部品数は大体保たれ、その結果として、部品群の数が増えると考えられる。部品群の集合が多くなった際に、主たる機能ではなく関連の薄い機能の利用でまとまりとなる部品の集合が一定割合で存在することで、このような部品が一定数、出現するようになると考えられ、このような主でない機能でまとまる部品をどう扱うかが課題となると考えられる。主たる機能の面で共通性を示している場合、共通利用部品はその類似している機能に関連があり、共通利用部品を部品理解に使うことは有効であると考えられる。より大きな規模のソフトウェアに対して適用した場

合に、生じるであろう問題点として、先に挙げた部品の集合の数が増えることで主たる機能ではなく関連の薄い機能の利用でまとまりとなった部品の集合が一定割合の割合で出現するといった点が考えられる。このような集合に対しては、類似度の計算方法を変えるなどして、何通りかの方法での分類結果も別に用意し、それらの結果を組み合わせることで、より理解支援に役立つ結果が得られるのではないかと期待している。今後、それらの部品に対して、その他の分類結果が得られるかなどを追跡したいと考えている。また、開発期間や開発チームの編成など、クラス数以外にも影響を与える要因があるのではないかと考えられる。

## 6 まとめ

本研究では、利用関係の一致度に基づくソフトウェア部品の分類手法を数百クラス規模のソフトウェアに対して適用した場合でも、得られた部品群内の部品が共通性を有しているかや、共通して利用している部品が部品群内の部品の理解に貢献できるかについて調査を行った。評価実験の結果から、数百クラス規模のソフトウェアに適用した場合、主でない機能で結びつく集合が部品の集合が増加したことで一定数出現するようになったこと、主たる機能で結びついている部品群は部品群内の部品の共通利用部品と関連していることが分かった。今後課題としては、現状では有効性を 2 つ 3 つのプロジェクトでのみ確認できた状態なので、他のプロジェクトにも同様の評価実験を行い傾向を調べる必要がある。そして、主でない機能の利用が共通していることで出来上がる部品群に対して、どのように対応することが可能かを考察していくことが必要である。また、クラス数以外にも影響を与える要因があると考えられるので、開発環境や開発チームの編成なども考慮する必要がある。

## 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] 橋本敬太, 川瀬史也: "利用部品の共通性に基づくソフトウェア部品分類手法の評価", 南山大学 理工学部 2018 年度卒業論文, 2019.