利用先や利用元の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

2014SE039 各務秀人　2014SE058 間瀬尚哉　2014SE087 澤井政斉
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．堀らはソフトウェアの利用関係から各部品の類似度を計算し，類似度を元にソフトウェア部品を分類することで似たような部品を抽出する手法を提案した [1] が，得られた部品群に本来含まれるべき部品が他に存在していたかについての評価はない．

本研究では，本来含まれるべき部品のうちどれだけを分類された部品群において含むことができたかという再現率の観点からも評価を行うことで，提案手法により得られた部品群の特徴を確認する．さらに，距離計算アルゴリズムに群平均法を用いてクラスタリングを行うにあたり，得られた樹形図のどの高さでグループ分けを行うかについて調査する．提案手法に基づいて類似部品の情報を提示することで，開発者が既存のコードをより効率的に理解することを支援できると考える．

2　関連研究

2.1　ソフトウェア部品グラフ

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，ソフトウェアを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さす [2]．本研究では，1 つの Java ソフトウェア全体を分析対象とし，それぞれのクラスのソースコードを記述しているファイルを部品とみなし，各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，辺は部品間の関係として利用関係を表現する．本研究では，ある部品 A が他の部品 B を利用している場合，A から B への利用関係が存在しているとみなし，部品グラフ上で頂点 A から頂点 B への有向辺で表現する．

2.2　ソフトウェア部品の利用関係の一致度をもとにした分類手法について

ある部品においてある機能を実現しようとする場合，他の部品で提供されている機能を利用しながら目的となる機能を実現する．堀らは [1] で，利用先に関して 2 つの部品が利用している部品が一致している割合が高いほどそれらの部品同士は目的や役割が似ている部品となるのではないかと考えた．また，利用元に関して共通の利用元を持っている割合が高いほどそれらの部品はある機能を実現するときにセットで使われるのではないかと考えた．[1] では，ソフトウェアの利用先（もしくは利用元）がどれだけ一致しているかから各部品の類似度を計算し，その類似度をもとにソフトウェアを分類することで，機能や役割が似ていると思われる部品を抽出する手法を提案した．実験では，利用先部品が多く一致している場合は，部品同士の役割が同じであったり，機能群としてまとまりを確認でき，関連性を強く認識できる部品の集合であることが確認できた．一方で利用元部品による類似度の分類では，利用先部品の場合と比べると関連性を確認しづらかったが，分類した部品群に共通の利用元部品を加えると 1 つの役割の一部を実現する部品群と認識できることが多く，意味付けが行えることを確認した．

2.2.1　分析手順

[1] で提案された分類手法における手順を以下に示す．

1. 分析対象のソフトウェアを分析して，クラス間の利用関係を入手する．
2. 1 で入手した利用関係から，各部品毎に利用先部品の集合，利用元部品の集合をそれぞれ作成する．
3. 利用先部品の類似度，もしくは利用元部品の類似度のどちらで分類するかを決めるとともに，各部品ごとに他の部品との類似度を求め，距離を計算する．
4. 各部品間の距離関係をもとに距離行列を作成する．
5. 距離行列を用いて階層的クラスター分析を行い樹形図を得る．得られた樹形図において，まとまりになっている部品から類似部品群を抽出する．
6. それぞれの類似部品群内の部品を調査し，どのような点で類似しているかを調査する．

2.2.2　部品間の距離

ある部品 A の利用先 (もしくは利用元) 部品の集合を L_{A}，ある部品 B の利用先 (もしくは利用元) 部品の集合を L_{B} としたとき，それらの集合間の類似度 $sim(L_{\mathrm{A}}, L_{\mathrm{B}})$ を Jaccard 係数を用いて示す．

$$sim(L_{\mathrm{A}}, L_{\mathrm{B}}) = \frac{|L_{\mathrm{A}} \cap L_{\mathrm{B}}|}{|L_{\mathrm{A}} \cup L_{\mathrm{B}}|} \quad (1)$$

この値は 0〜1 の値域を持ち，高いほど類似していることを示しているので，距離 $dist(L_{\mathrm{A}}, L_{\mathrm{B}})$ を以下のように定義し，距離行列の作成に用いる．

$$dist(L_{\mathrm{A}}, L_{\mathrm{B}}) = 1 - sim(L_{\mathrm{A}}, L_{\mathrm{B}})$$

3 利用先や利用元の部品の共通性に基づくソフトウェア部品の分類手法の評価

3.1 研究の動機

過去の研究 [1] では，利用先や利用元の一致度に基づきクラスター分析を行った結果，分類された部品群の部品同士が強い関連性を示したことは確認できたが，本来含まれるべき部品をどれだけ抽出できたかという再現率の観点からの評価がなされていない．本研究では，本来含まれるべき部品をどれだけ抽出できたかという再現率の観点から評価を行い，手法の有効性を網羅性を含めて検証する．さらに，閾値の設定箇所によって得られる結果が変化することから，最適な結果を得ることができる閾値をどのように求めることができるか調査する．これにより大規模化し複雑化したソフトウェアの構成要素を効率よく把握することを助ける提案手法が利用できるかを確認するとともに，より適切な分類結果を求めることができる方法を調査する．

3.2 クラスタリングに用いる最適な距離計算アルゴリズムの評価

階層的クラスター分析に用いる距離計算アルゴリズムには，最近距離法，最遠距離法，群平均法，メディアン法，重心法，ウォード法などがあるが，今回の研究の分析手順では各部品間の距離をすでに求めている．そのためクラスター間の距離を求める際に重心を用いる距離計算アルゴリズムは意味をもたず，距離を再計算するアルゴリズムを用いるのが良いと考えた．最近距離法，最遠距離法，群平均法などが当てはまると考えられるが，群平均法を用いて得られた樹形図のどの高さでグループを生成するかで最近距離法，最遠距離法と同じ結果を得ることができることから，本研究では類似部品を抽出するにあたって群平均法をクラスタリングに用いて実験を行う．実験では，部品群を抽出するのに適切な高さを閾値として設定し適切な閾値について調査する．

3.3 本研究における調査内容

以下のリサーチクエスチョンを設定し，評価実験を行う．

1. 分類された部品集合内の部品はどのような着眼点で機能や役割が似ているといえるか？
   どのような着眼点で似ているといえるかについての基準を複数用意し，分類された部品の集合がどの基準を満たすかを調べる．これにより，提案手法によって得られた部品集合が持つ特徴を調査する．
2. 分類できた部品集合に本来入るべき部品が他にどのくらい存在するか？
   1 で類似していると判断した部品集合を対象として，部品集合の外の部品で，その観点で類似している部品がどれだけ存在するかを調べる．これにより，本来含まれるべき部品がどれだけ分類できたかという観点から調査し，網羅性を確認する．
3. 群平均法により分類されたクラスターをどの閾値で部品群へ分けるべきか？
   それぞれの閾値で分類された部品群を評価し，どの閾値で分類するのが適切かという観点で比較する．

3.4 類似度の定義について

分類を行った結果，複数の部品からなる部品の集合が得られるが，それらがどのように類似しているかについて様々な観点から判断することができる．以下では，どのような観点で似ていると考えられるについて複数の判断基準を設定し類似性を判断するための基準とする．その条件を満たした場合，どのように似ているといえるか説明する．

基準 1 部品の派生元や実装しているインタフェースが同じである．派生元が同じであることや，共通の機能を持つという観点から類似していると考えられる．

基準 2 利用先の 50 %以上が同じである．同じ対象を扱っている場合，同じ対象に対する機能群を構成していると考えられる．

基準 3-A 部品群を 1 つのグループをしてみたときに，部品の役割や扱う対象から機能群を構成している．1 つの大きな役割に対して，その一部を実現する部品群であるという点で類似していると考えられる．

基準 3-B 基準 3-A の条件は満たさないが，部品群の共通の利用元を含めれば，機能群を構成すると認識できる．共通の利用元を含めることで，ある 1 つの機能を実現する部品群であるという点で類似していると考えられる．

基準 4 クラス内に同一のシグニチャのメソッドが複数存在する．分類された部品が 1 つの役割を果たす部品でまとまり，同じ目的の処理が行われている部品であるという点で類似していると考えられる．

基準 5 同じパッケージに所属している．パッケージ構造は開発者が役割や性質，目的などに応じて分類したものであり，同じ目的の部品群であると考えられる．

基準 6 ファイル名の一部が一致している．ファイル名には，そのファイルの機能を明示するという慣習があるので，ファイル名の一部が一致していることで類似した役割を持つと考えられる．

基準 7 コードクローンを有している．同じ機能，同じ役割を実現しているために，関連していると考えられる．

4 評価実験 1:適合率と再現率の評価

4.1 実験における調査項目

jlgui を対象として，提案手法を適用する．jlgui はイコライザ機能を提供する Java アプリケーションで，70 のソースファイル (部品) で構成されている．

調査においては，各部品群に対して適合率と再現率の 2 つの指標を用いる．部品群内の部品を 1 つ無作為に抽出

し，その部品を A とする．このとき，部品群内で A とその基準で関係を持つ部品数を，部品群中の部品数で割ったものを適合率，部品群内で A とその基準で関係を持つ部品数を，ソフトウェア中で A とその基準で関係を持つ部品数で割ったものを再現率と定義する．F 値を，正確性を示す適合率と網羅性を示す再現率の 2 つの評価指標の調和平均とし，総合的な性能を計測するために用いる．

4.2 評価実験 1 の結果

利用先，利用元の群平均法で得られた結果をそれぞれ図 1，図 2 に示す．利用先においては，基準線以下で結合した 8 個の部品群が抽出でき，それぞれの部品群に対し，各基準に対する適合率を表 1 に示した．表 1 の結果から，部品群 1〜7 はほとんどの基準で高い割合を示したが，部品群 8 は 1 つの基準も満たさなかった．部品群 8 は音楽プレイリストの表示の根幹となる部品と音楽ファイルを検索する部品の 2 部品で構成されており，2 つの部品は共通のユーザインタフェースの部品を使っているが，強い関係性はないことを確認した．利用元においても，図 2 のように基準線以下で結合した 14 個の部品群が抽出でき，それぞれの部品群に対し，各基準に対する適合率を表 2 に示した．基準 1 と基準 7 に関してはほとんどの部品群で低い割合を示したが、そのほかの基準ではほとんどの部品群が高い割合を示し，中でも基準 3-A，B についてはほとんどの部品群が満たすことができ，全体の分類結果としては機能群として分類されると考えられる結果であった．

表 3，4 は，部品群として分類されている部品のうち，基準を満たしている部品がどれだけ含まれているのかという再現率を示しており，表 3 では，表 1 で条件を満たした各基準について，その条件における再現率を示す．部品群 7，8 に関しては，他の部品群に比べ割合が低い結果や元々機能群として見られていない結果であり、適切な部品が集合していると考えられる結果は得られなかった．しかし，ほとんどの基準で高い割合を示しており，本来含まれるべき部品が多く含まれている部品群と考える．表 4 に，表 2 で条件を満たした各基準の再現率を示す．部品群 4，11 は高い割合で類似性を持つ部品が分類されている結果となっているが，その他の部品群では本来含まれるべき部品が部品群内に含まれているとは言えず，求める部品のうちの一部のみを得ていると考えられる．

5 評価実験 2：適切な閾値の評価

5.1 実験における調査項目

Jlgui の利用先，利用元で得られた樹形図に対してそれぞれ 3 通りのグループ分けの基準となる閾値を設定する．第 4 章で求めた適合率，再現率を用いてそれぞれの部品群における各基準ごとの F 値を求め，最も F 値の平均が高い箇所を最適なグループ生成における区切り位置と判断する．

5.2 評価実験 2 の結果

jlgui の利用先，利用元に関してどこに閾値を設定するのが最適であるか，F 値を用いて判定を行った結果を示す．表 5 は jlgui の利用先において閾値を 5.8 と 6.8 と 7.8 で設定し，それぞれの部品群において各基準ごとに F 値の平均を求めた結果である．基準 2，3 に関しては閾値 5.8 の F 値の平均が高い値を示しているが，設定した 3 箇所の閾値全体では，6.8 が最も大きな F 値を示しており，3 つのうちでは最適な閾値であると考えられる．

利用元に関しても同様に，2.8，4.3，5.8 で閾値を設定し F 値を求めた．その結果を示したのが表 6 である．基準 1 では閾値 5.8 が，基準 5 では閾値 2.8 が高い値を示しているが，他の基準に関しては閾値 4.3 が高い値を示し，各基準全体の平均では閾値 4.3 が最も高い値を示していることから 3 つのうち最適な閾値であると考える．

表 1 適合率 (利用先部品)

利用先	ファイル数	基準 1	2	3-A	3-B	4	5	6	7
1	5	80%	100%	100%	-	80%	80%	80%	0%
2	4	100%	100%	100%	-	100%	100%	100%	75%
3	9	44%	78%	67%	-	88%	78%	67%	0%
4	4	100%	100%	100%	-	100%	100%	100%	100%
5	5	100%	100%	100%	-	100%	100%	100%	60%
6	3	100%	100%	100%	-	100%	67%	100%	100%
7	4	0%	100%	75%	-	0%	50%	50%	0%
8	2	0%	0%	0%	-	0%	0%	0%	0%

表 2 適合率 (利用元部品)

利用元	ファイル数	基準 1	2	3 − A	3 − B	4	5	6	7
1	2	0%	0%	0%	100%	0%	100%	0%	0%
2	2	0%	0%	0%	0%	0%	100%	0%	0%
3	2	0%	0%	0%	100%	0%	100%	0%	0%
4	5	100%	100%	100%	-	100%	100%	100%	67%
5	2	0%	100%	0%	0%	100%	100%	0%	0%
6	2	0%	100%	0%	0%	0%	100%	0%	-
7	5	40%	80%	80%	-	60%	80%	80%	-
8	4	0%	50%	0%	0%	50%	50%	100%	0%
9	3	0%	0%	0%	100%	67%	67%	0%	0%
10	2	0%	0%	100%	-	0%	0%	0%	0%
11	2	0%	0%	100%	-	0%	0%	0%	0%
12	4	50%	50%	0%	0%	0%	0%	0%	0%
13	2	100%	100%	100%	-	100%	0%	100%	100%
14	2	0%	0%	100%	-	0%	100%	100%	0%

表 3 再現率 (利用先部品)

利用先	ファイル数	基準 1	2	3-A	3-B	4	5	6	7
1	5	80%	100%	100%	-	80%	80%	80%	-
2	4	67%	50%	58%	-	58%	58%	44%	38%
3	9	-	60%	60%	-	80%	33%	60%	-
4	4	80%	80%	100%	-	67%	80%	100%	100%
5	5	62%	50%	55%	-	38%	55%	55%	100%
6	3	37%	50%	60%	-	100%	33%	60%	42%
7	4	-	23%	42%	-	-	28%	40%	-
8	2	-	-	-	-	-	-	-	-

6 考察

6.1 評価実験 1

jlgui の利用先の一致度を用いて分類した場合，適合率，再現率ともに高い割合を示すことができ，その結果は本来含まれるべき部品の多くを含むことができると考えられ

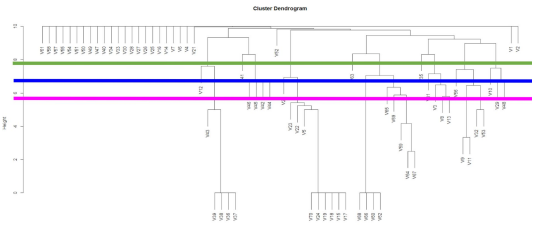

図 1　利用先部品の類似度で作成した樹形図 (群平均法)

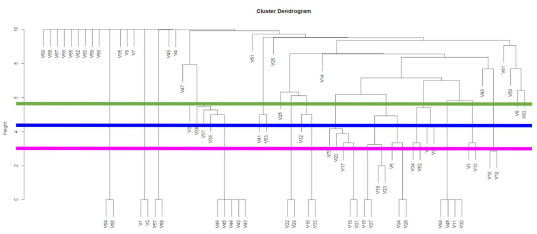

図 2　利用元部品の類似度で作成した樹形図 (群平均法)

表 4　再現率 (利用元部品)

利用元	ファイル数	基準 1	2	3 − A	3 − B	4	5	6	7
1	2	-	-	-	50%	-	40%	-	-
2	2	-	-	-	-	-	40%	-	-
3	2	-	-	-	33%	-	40%	-	-
4	5	100%	71%	71%	-	83%	71%	71%	44%
5	2	-	13%	-	-	36%	10%	-	-
6	2	-	12%	-	-	-	10%	-	-
7	5	-	29%	80%	-	60%	19%	40%	-
8	4	-	26%	-	-	36%	19%	67%	-
9	3	-	-	-	27%	22%	10%	-	-
10	2	-	-	28%	-	-	-	-	-
11	2	-	-	28%	-	-	-	-	-
12	4	66%	13%	-	-	-	-	-	-
13	2	28%	66%	40%	-	50%	-	50%	66%
14	2	-	-	33%	-	-	40%	33%	-

表 5　F 値 (利用先部品)

利用先	ファイル数	基準 1	2	3	4	5	6	7	全体平均
基準 5.8	25	0.372	**0.670**	**0.730**	0.470	0.526	0.556	0.167	0.499
基準 6.8	36	**0.475**	0.617	0.671	**0.591**	**0.551**	**0.620**	**0.344**	**0.553**
基準 7.8	41	0.415	0.666	0.625	0.553	0.459	0.521	0.149	0.484

表 6　F 値 (利用元部品)

利用元	ファイル数	基準 1	2	3	4	5	6	7	全体平均
基準 2.8	29	0.056	0.163	0.318	0.262	**0.347**	0.144	0.044	0.191
基準 4.3	39	0.079	**0.201**	**0.408**	**0.287**	0.303	**0.238**	**0.130**	**0.235**
基準 5.8	46	**0.083**	0.126	0.379	0.153	0.297	0.144	0.046	0.202

る．利用関係の一致度で分類することで，分類された部品群が jlgui においてどのような役割を持つのかを効率的に把握することができると考えられる．

利用元の一致度を用いて分類した場合，いくつかの部品群内の部品は強い関連性を持つことがわかったが，再現率自体は高いとは言えない結果であった．より多くの強い関連を示す部品を含むための精度向上のための手法が必要であると考えられる．

6.2　評価実験 2

階層的クラスター分析では，閾値の設定する高さによって抽出される部品群が異なり、それに伴い得られる結果も変わってくる．グループ分けを行う際に，部品群の一致度の基準を高くしすぎると調査対象となる部品数が少なくなってしまい，低くしすぎるとグループ分けされた部品群の関連性が低くなってしまうことから，最適な閾値を設定する高さがあると考えられる．

今回適切な閾値を定める方法として F 値を用いて実験を行ったが，今回調査した範囲内，jlgui の利用先は閾値 5.8 から 7.8 の間に，利用元は閾値 2.8 から 5.8 の間に，最適な閾値の設定箇所が存在すると考えられた．他のソフトウェアも同様に，何か所か任意に閾値を設定し F 値を求めることで，そのソフトウェアにおける最適な閾値を設定する値が求めることができると考えられる．今回の実験結果から，適切な閾値の設定箇所は利用先と利用元に関して同じではない可能性があり，1 つのソフトウェアに関して利用先と利用元でそれぞれ閾値を求める必要があると考えられる．また抽出した部品群内の部品を分類する基準は，それぞれ得られた F 値の結果に差があり，基準ごとに部品をなるべく多くとったほうが良い場合や，強く関連した部品だけにした方が良い場合など，最適な閾値が異なることも考えられる．

7　まとめ

本研究では，ソフトウェアの利用先，利用元の一致度に基づいて分類を行う手法において，適合率，再現率の観点から評価を行った．評価実験の結果からは，利用先の一致度で分類した結果は，それぞれの類似についての基準で分類した時に含まれるべき部品を多く含んでいることが分かった．さらに，グループ分けの際の閾値についても考察を行い，適切な閾値が存在するであろうことを確認した．今後の課題として，他のソフトウェアに対しても同様の評価実験を行い，手法の精度を向上させるための方法について考察したうえで，これらの傾向の一般性を評価することで，開発者がコードを理解するために支援を行う方法を確立したい．

参考文献

[1] 堀貫行，後藤彗 :“利用先や利用元の部品の共通性に基づくソフトウェア部品分類手法の提案”，南山大学 情報理工学部 2016 年度卒業論文，2017.

[2] C. Kruegger :“Software Reuse”，ACM Conputing Surveys，vol. 24，no. 2，pp. 131-183，1992.

[3] R : “The R Project for Statistical Computing”，https://www.r-project.org/.