利用関係の一致度に基づくソフトウェア部品分類手法
―利用関係の局所性を考慮した類似度計算手法についての考察―

2016SE066 大谷恭介　2017SE094 遠山貴駿
指導教員：横森励士

1　はじめに

我々の研究グループでは，ソフトウェア部品の利用関係がどれくらい一致するかに基づいて，ソフトウェア部品を分類する手法を提案している．提案手法の改善手法として，共通利用部品数による分類手法が提案され，局所性を考慮することが必要であるという知見が得られた．しかし，局所性を反映させる方法についてはまだ検討が行われていない．本研究では，局所性を考慮した手法として，文書検索における IDF(Inverse Document Frequency) を参考にした評価方法である ICUF(Inverse Conponent Use Frequency) を提案し，ICUF に基づいた分類手法を提案する．実際のソフトウェアに対して適用し，分類結果を比較することで，提案手法が分類結果の精度向上に役立つかを検証する．

2　背景技術

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のモジュールの一部を指す．各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，有向辺は部品間の関係である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　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]．[1] での評価実験では，得られた部品群内の部品の多くが類似度を持つことが確認されている．

藤田らは，類似度計算の精度を向上させるための手法として，利用部品の一致度ではなく，共通利用部品数に基づいて類似度を計算する手法を提案した [2]．評価実験からは，ほとんどの部品について結果は同じであったが，一部の部品の結合の仕方が変わり，それらについては類似性を持つ部品が集約されやすくなったことを示した．[2] では，『「その部品を利用する部品の数が少ない部品」への利用関係は局所性を持つ』と仮説を立てて，局所性を考慮した類似度計算手法も提案した．共通で利用されている部品それぞれについて，それらがソフトウェア全体で実際にいくつの部品から利用されるかを調べ，その数が一定以下の部品について局所性を持つと判断して，類似度の計算に重みをつけた．その結果，局所性を持つ部品への利用関係に重みをつけることで，分類結果が具体的にどの部品を利用しているかに沿った形で分類することができ，類似点の根拠が示しやすくなったことが示されている．

3　ICUF に基づいた類似度計算手法の提案

3.1　研究の動機

我々の研究グループでは，ソフトウェアの各部品が利用している部品の一致度に基づいてソフトウェアを分類する手法を提案し，得られた分類結果がソフトウェアを機能面から分類できていることを示した．この手法は，ソフトウェアの中を分類することで，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．

過去の研究では，共通部品数に基づいてソフトウェアを分類することで分類の精度が上がるという結果が得られたとともに，局所性を反映させることが有効であるという知見が得られた [2]．しかし，局所性を反映させる具体的な方法についてはまだ十分な検討がなされておらず，局所性を反映させる方法の検討が必要である．

3.2　研究の目的

局所性を考慮する手法として，IDF のような文書検索において一般的に用いられている手法がある．この手法は，単語が出現する文献の数に基づいて，文献に出現しにくい単語に重みづけを行う手法である．

本研究では，局所性を考慮する手法として，ICUF に基づいた類似度計算手法を提案する．ICUF は IDF を参考にして，利用されることが少ない部品への利用関係に重みづけを行う．各手法で得られた部品群について，どの箇所が同じでどの箇所が異なるかや，異なる箇所についてどのような点で違いがあるかなどを調査し，局所性を反映させる方法として，提案手法が妥当であるかを調査する．

3.3　ICUF(Inverse Component Use Frequency) について

IDF は，主に文書検索において用いられる手法であり，ある単語が検索対象の文書のうち，いくつの文書で出現しているかを表す．特定の少数の文書にしか登場しない単語であればあるほどその単語の IDF 値は高くなる [3]．IDF

では，一般的に計算方法を以下のように定義する．

$$IDF = log(\text{総文書数}/\text{ある単語が出現する文書の数}) + 1$$

ICUF は，ある部品が「分析対象内のソフトウェア部品のいくつから利用されているか」に基づいて重みづけを行う方法である．少数の部品にしか利用されない部品ほど，ICUF の値が高くなるように，以下のように部品 X への利用関係について ICUF(X) を定義する．

$$ICUF(X) = log(\text{分析対象内の総部品数}/\text{X を利用する部品数}) + 1$$

3.4 ICUF に基づいた場合の類似度の計算手順

ICUF に基づいて 2 つの部品間の利用関係の類似度を計算する手順を以下のように定義する．

1. 2 つの部品の共通利用部品の集合を求める．
2. 各共通利用部品について，ICUF の値を求め，その値をその共通利用部品への利用関係の重みとする．
3. 2. で求めたそれぞれの共通利用部品の重みの総和を 2 つの部品間の類似度とする．

4 比較実験

4.1 比較対象とする類似度計算手法について

本研究では，既存の手法を含めた 2 つの類似度計算手法によって得られた距離行列に基づいて階層的クラスター分析を行い，得られた部品群の比較を行う．以下では，分析対象のソフトウェアの部品の集合を C とする．ある部品 A(∈C) が利用している部品の集合を O_A，ある部品 B(∈C) が利用している部品の集合を O_B とする．いずれの場合も同部品間の距離 dist(A，A) は 0 とする．

1. 共通利用部品数に基づいた類似度計算手法
   C 中のすべての部品の組み合わせにおける類似度 sim の最大値を MAX(C) とする．2 つの部品間 A，B の類似度 sim と距離 dist を以下のように定義する．

$$sim(A, B) = |O_A \cap O_B|$$

$$dist(A, B) = 1 - \frac{sim(A, B)}{MAX(C) + 1}$$

2. ICUF に基づいて局所性を考慮した類似度計算手法
   ある部品 A,B が共通して利用している部品の集合 $O_A \cap O_B$ について考える．D($\in(O_A \cap O_B)$) について，ICUF(D) を求め，その総和を A，B 間の類似度 sim とする．C 中のすべての部品の組み合わせにおける類似度 sim の最大値を MAX(C) とする．2 つの部品間 A，B の距離 dist を以下のように定義する．

$$sim(A, B) = \sum_{D \in (O_A \cap O_B)} ICUF(D)$$

$$dist(A, B) = 1 - \frac{sim(A, B)}{MAX(C) + 1}$$

4.2 類似度計算手法間の比較方法について

4.1 の 2 つの方法で作成した距離行列を用いて，階層的クラスター分析を行い，樹形図を得る．樹形図に沿って類似した部品が最大限部品群に含まれるように類似部品群を抽出する．得られた類似部品群の相違点に着目し，どちらの分類手法がより適切かを部品間の関係を調査して評価する．本稿では，共通利用部品数に基づいた場合と ICUF に基づいて局所性を考慮した場合の比較を行い，どのような変化が起こりうるかについて考察する．

4.3 各類似計算手法における部品群の作成方法について

階層的クラスター分析によって得られた樹形図から類似している部品を樹形図に沿って部品群に加えていく．類似性を評価するために，以下のような基準を作成し，いずれかの基準を満たす部品を関連性を持つ部品とする [4].

基準 1 分類された部品の扱う対象が同じである．

基準 2 分類された部品の役割が同じである．

基準 3 部品の役割や対象やパッケージなどから 1 つの機能群としてまとめられると考えられる部品群である．共通の利用元を考慮することで，機能群と認識できる場合も含む．

4.4 分析対象アプリケーション

本研究では，JasperReports Library と PlotDizitizer という Java アプリケーションに対して適用実験を行った．JasperReports Library は帳票用のライブラリで，88 のソースファイルで構成されている．PlotDizitizer は印刷後の数値としての情報を失ってしまったグラフから値を読み込むソフトで，80 のソースファイルで構成されている．

4.5 分類結果

JasperReports Library を分析対象のソフトウェアとして 4.1 の 1，2 の手法で得られた樹形図を図 1，2 に表す．表 1 では、ICUF を用いて 4.3 の基準 1，2，3 に従い，抽出した類似部品群を示す．結合の仕方は異なるが，得られる部品群に大きな違いは見られなかった．表では，左から部品群の ID，部品数，主な部品，共通点，共通利用部品数に基づいた分類での部品群 ID について表している．変化がみられた部品群について下線付きで表現する．共通利用部品数に基づいた場合は，54 個の部品が 10 個の部品群で得られた．ICUF に基づいた場合は，55 個の部品が 11 個の部品群で得られた．

PlotDigitizer を分析対象のソフトウェアとして 4.1 の 1，2 の手法で得られた樹形図を図 3，4 に表す．表 2 では，JasperReports Library の場合と同じ形式で得られた部品群を紹介する．新しく得られた部品群は二重下線付きで表現する．共通利用部品数に基づいた場合は，22 個の部品が 7 個の部品群で得られた．ICUF に基づいた場合は，29 個の部品が 10 個の部品群で得られた．

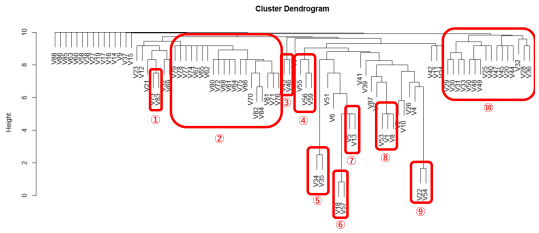

図 1　JasperReports Library に対して共通利用部品数に基づいた手法を用いて得られた樹形図と類似部品の集合

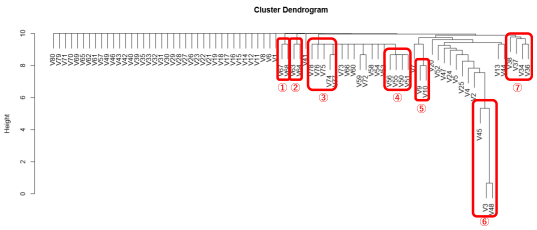

図 3　PlotDigitizer に対して共通利用部品数に基づいた手法を用いて得られた樹形図と類似部品の集合

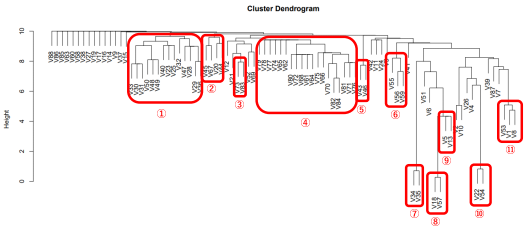

図 2　JasperReports Library に対して ICUF に基づいて局所性を考慮した手法を用いて得られた樹形図と類似部品の集合

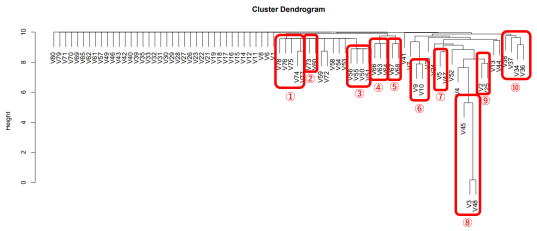

図 4　PlotDigitizer に対して ICUF に基づいて局所性を考慮した手法を用いて得られた樹形図と類似部品の集合

部品群番号	部品数	主な部品	共通点	1
1	14	JRImage, JRLine など	レポートに関連	10
2	4	JRTElement, JRTextElement など	レポートに関連	10
3	2	JRInitialValueExpressionFactory など	ファクトリー	1
4	19	JasperDesignFactory, JRBandFactory など	ファクトリー	2
5	2	JRStaticText, JRTextField	テキスト	3
6	3	JasperDesignViewer, JasperViewer など	レポート表示	4
7	2	JRPrinter, JRPrinterPDF	印刷	5
8	2	JRFiller, JRBandDesignViewer	まとめる	6
9	2	JRBand, JRElementGroup	まとめる	7
10	2	JRHorizontalFiller, JRVerticalFiller	レポート記入	9
11	3	JasperDesign, JRClassGenerator など	レポートに関連	8

表 1　JasperReports Library に対して ICUF に基づいて局所性を考慮した手法で得た部品群

部品群番号	部品数	主な部品	共通点	1
1	5	ScreenMenu, JScreenMenuBar など	スクリーンメニュー	3
2	2	SpecialFolder, FolderDialog	フォルダー	なし
3	4	AboutJMenuItem, QuitJMenuItem など	メニュー項目	4
4	3	MRJ23EventProxy, JD2AppleEventHandlerThunk など	イベント	2
5	2	MRJ4EventProxy, MRJAdapter	MRJ	1
6	2	XMLoader, XMLSaver	XML に関連	5
7	2	AutoOptionsDialog, ScaleInputDialog	ダイアログ	なし
8	3	AppWindow, TableWindow など	ウィンドウ	6
9	2	AppPreferences, MDIApplication	アプリケーションの設定	なし
10	4	CSVReader, FloatTable など	表	7

表 2　PlotDigitizer に対して ICUF に基づいて局所性を考慮した手法で得た部品群

5　変化した点についての考察

5.1　JasperReports Liblary での変化について

JasperReports Library に対して，ICUF に基づいて分類した際に得られた部品群について，共通利用部品数に基づいて分類した際に得られた部品群と比べて変化があった部分を調査した．変化した部品群は，3 つあった．表 3 では，変化のあった部品が利用した部品を列挙し，その部品を利用している部品数，ICUF 値について示す．

部品群 4 での変化は，JRFontFactory が部品群 2 の部品群に移動したことである．これは，共通利用部品数に基づいた場合に部品群 2 で JRBaseFactory への利用関係で結びついていたのが，ICUF に基づいた場合に部品群 2 で JRFont への利用関係で結びつくようになったことを示している．JRBaseFactory は，共通利用部品数に基づいた場合に部品群 2 で共通して利用している部品である．JRFont は，JRFontFactory が部品群 2 で JRTextElement と共通して利用している部品である．表 3 の ICUF 値から，JRFontFactory は，JRBaseFactory への利用関係 (1.58) より JRFont への利用関係 (2.10) の重みが約 1.3 倍強くなったことが分かる．

部品群 1，2 での変化は，それぞれ JRTImage と JRGraphicElement への利用関係が強い部品群と JRFont への利用関係が強い部品群に分離した．共通利用部品数に基づいた場合に部品群 10 で JRGrapicElement への利用関係で結びついていたのが，ICUF に基づいた場合では，部品群 1 で新しく JRImage が分類されることで JRTImage への利用関係でも結びつくようになった．JRTImage は，

JRImage が部品群 1 で JRPImage と共通して利用している部品である．JRGraphicElement は，JRImage が部品群 1 で JRLine などの 6 個の部品と共通して利用している部品である．表 3 の ICUF 値から，JRImage は，JRGraphicElement への利用関係 (2.04) より JRTImage への利用関係 (2.47) の重みが約 1.2 倍強くなったことが分かる．

共通利用部品名	被利用部品数	その部品への利用関係の ICUF 値
JRTImage	3	2.47
JRFont	7	2.10
JRGraphicElement	8	2.04
JRBaseFactory	23	1.58

表 3 JasperReports Library で結びつきの変化に関連した利用関係

5.2 PlotDigitizer での変化について

PlotDigitizer に対して，ICUF に基づいて分類した際に得られた部品群について，共通利用部品数に基づいて分類した際に得られた部品群と比べて変化があった部分を調査した．変化した部品群は，1 つで新しく 3 つの部品群ができた．表 4 では，JasperReports Library と同じように変化のあった部品が共通して利用している部品について紹介している．

部品群 4 での変化は，MRJ23EventProxy が新しく分類されたことである．これは，共通利用部品数に基づいた場合で分類されていなかった部品が，ICUF に基づいた場合に部品群 4 で AppleEventHandler への利用関係で結びつくようになったことを示している．AppleEventHandler は，MRJ23EventProxy が部品群 4 で JD2AppleEventHandlerThunk と JD3AppleEventHandlerThunk と共通して利用している部品である．表 4 の ICUF 値から，MRJ23EventProxy は，AppleEventHandler との関係 (2.43) があることで新しく分類された．

新しくできた例として，部品群 9 では，AppPreferences と MDIApplication が新しく分類された．これは，共通利用部品数に基づいた場合で分類されていなかった部品が，ICUF に基づいた場合に部品群 9 で Prefereces と AppUtilities への利用関係で結びつくようになったことを示している．Preferences と AppUtilities は，AppPeferences と MDIApplication が部品群 9 で共通して利用している部品である．表 4 の ICUF 値から，AppPreferences と MDIApplication は，AppUtilities との関係 (2.00) より Preferences との関係 (2.12) が強いことが分かる．

5.3 ICUF に基づいて局所性を考慮した手法の考察

共通利用部品数に基づいた場合に比べ，ICUF に基づいた場合では，利用関係が固まっているところの分類は大きく変わらないが，多くの共通利用部品がある部品群でどの部品との関係を重視するかで分類の仕方が変わることが分かった．共通利用部品数に基づいた手法で得られた部品群は，ソフトウェアの大きな機能を表現するようなクラスをベースとして部品の分類が行われていた．ICUF に基づいて局所性を考慮した手法では，利用目的がはっきりしている部品をベースとして部品の分類が行われていた．ICUF に基づくことで，今まで分類されていなかった部品が説明がつきやすい利用関係に基づいて分類されることが多くなった．共通利用部品数に基づいた手法より細かい機能の面から部品間の共通点が見出しやすくなり，利用関係を説明しやすくなる手法であると考える．どちらの手法も必ずしも間違っている分類を行っているわけではないので，どちらが妥当であるかは判断できないが，併用して利用することで分類の精度の向上がみられると考える．

共通利用部品	被利用部品数	その部品への利用関係の ICUF 値
AppleEventHandler	3	2.43
EscapeJDialog	3	2.43
Preferences	6	2.12
AppUtilities	8	2.00
MDIApplication	10	1.90
MRJAdapter	18	1.65

表 4 PlotDigitizer で結びつきの変化に関連した利用関係

6 まとめ

本研究では，ICUF に基づいて局所性を考慮した類似度計算手法を提案し，共通利用部品数によって分類を行った結果と比較した，結果として，多くの部品を利用している部品の結びつき方に変化が起こった．広く利用されている部品の観点や，細かい機能の観点など，分類手法によって異なる観点から部品を分類していることがわかった．今後の課題として，ほかの事例に対する適用を行い，一般性に関する知見を見出すことなどが挙げられる．

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] 藤田翔太, 清水太智, 戸本了太:” 利用部品の共通性に基づくソフトウェア分類手法-類似度計算手法に関する考察-”, 南山大学理工学部ソフトウェア工学科 2019 年度卒業論文, 2020.

[3] 木村美紀:”TF-IDF を用いた文書分類の試み-A Case Study for Text Classification by Employing a Weighted Variable TF-IDF-”, 文学研究論集, 第 48 号, 2018. 2.

[4] 堀貴行, 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ソフトウェア工学科 2016 年度卒業論文, 2017.