1L-01

ソフトウェアメトリクスを用いたソースコード検索手法の提案

曽和　寛史† 尾花　将輝‡ 深海　悟‡

大阪工業大学大学院情報科学研究科†　　大阪工業大学情報科学部‡

1　はじめに

近年，社会的なソフトウェアの役割は大きくなっており，それに伴い様々なサービスがソフトウェアで提供されている．また，ソフトウェア開発は様々なサービスを提供するために大規模化している，しかし，その反面，短納期化が進んでおり，開発者はより効率的なソフトウェア開発が求められている．本研究では，近年のオープンソースソフトウェア（以下オープンソース）の品質の高さに着目し，ソフトウェアメトリクス（以下メトリクス）を用いたオープンソース再利用のためのソースコードの絞込み手法を提案する．

膨大なオープンソースから目的とするコードの類似コード候補を絞り込むことで，コードクローン技術等の検出技術を用いて詳細に類似コードを自動的に検索する[1]．これにより，膨大なソースコードの中から類似のソースコードを開発者に掲示することで，ソフトウェア開発の効率化が期待できる．

ソースコードの再利用を行うシステムとしてGitHub Code Search [2] や SPARS-J[3]がある．これらの検索システムはキーワードを用いて，オープンソース内のソースコードを検索が可能である．しかし，これらはユーザがクエリを入力する必要がある．また，コードクローン検出を用いたソースコードの再利用方法等も提案されている[4]．しかし，コードクローン検出はオープンソース等の膨大ソースを検索するには膨大な時間を要する．

2　提案手法の概要

本提案は，オープンソースとして公開されているソースに対し，変数の数等でメトリクスをメソッドごとに計測し，開発中のソースのメトリクス値との類似度を計測することでソースコードの絞込みを行う．類似度が閾値以上ならば，求めるソースコードとし，絞込み後は詳細なコード検索としてコードクローン検出を利用する．

本提案の概念図を図 1 に示す．本提案では，オープンソース等で公開されるソースを対象にメソッド単位で戻り値，引数，ローカル変数の数に関するメトリクスを計測する．これらをデータベースへと登録し，登録したソースと現在開発中のソースとで類似度を計測することで，登録したソースから類似するメソッドの絞込みを行う．絞込みを行うことで，計算量の多いコードクローン検出で更なる詳細な類似コードを検出し，自動的に類似メソッドを掲示する．

3　提案手法

3.1　本提案のアプローチ

本研究の目的はオープンソース等のソースからメソッド単位でコードクローン検出を行うためにソースコードの絞込みを行うことである．そのために，本提案ではメソッドからの戻り値や引数の数，ローカル変数の数に着目した．また，ローカル変数の数は int 型，double 型と言うようプリミティブ型ごとにメトリクスを採取している．

これらのメトリクスは，プログラムを開発する際に利用する重要な要素であると考える．例えば，あるメソッドを開発する際にはまず，引数，戻り値を決定して開発することが一般的である．また，そのメソッドをどのような機能を持つものとするかの鍵を握るのは if 文や for 文等の制御文と考えられるが，それらは全て変数によって制御されると考える．つまり，ローカル変数や引数等の型や数が同じメソッドは、機能も同じである可能性が高いと考える．

そこで，本研究で調査対象とするプロジェクト 60 個(約 23 万メソッド)に対し，戻り値，引数，ローカル変数が完全一致するメソッドがどのぐらい存在するかを調査した．具体的には、対象とした約 23 万メソッドに本学の Java 演習で学生が作成したソース（ソースファイル数 99 個）を 23 万個のメソッドに埋め込み，これらと各メソッドの引数の数，戻り値，ローカル変数の数が完全一致するものが何個あるかを確認した．ただし，本研究ではローカル変数が全く存在しないプログラムはカプセル化されたメソッドとの区別が困難なため，検出の対象外とする．

Code reuse using software metrics
† Graduate School of Osaka Institute of Technology
‡ Osaka Institute of Technology

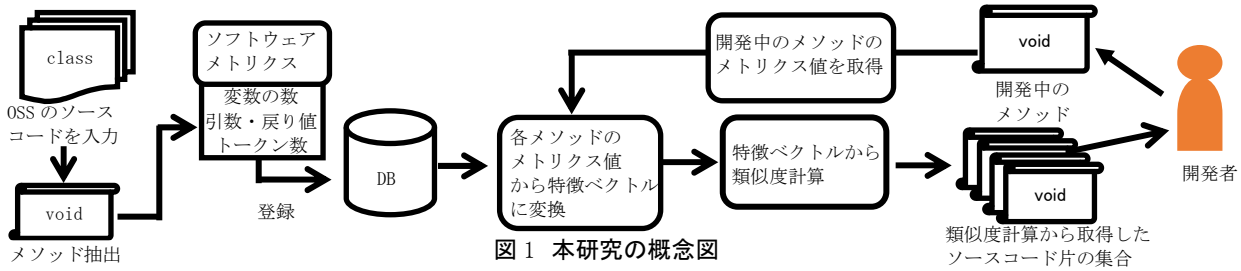

図 1　本研究の概念図

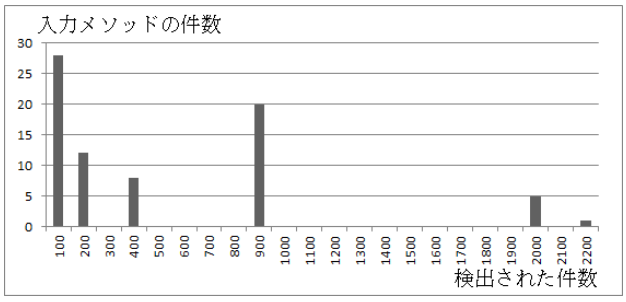

図２　メトリクスが一致したメソッドの度数分布

結果は対象のソースからローカル変数が無いものを除いた場合，メソッド数 74 個であり，74 個の各メソッドのメトリクス値が完全一致した個数を 100 個区切りの度数分布で示したものが図 2 である．最も少ない個数は 1 個，つまり，演習で作成したメソッドのみとなり，最も多いメソッドは 2148 個，1 メソッドあたりの平均は 453 個であった．これらの結果より，ローカル変数や戻り値，引数だけでも 1%以内まで絞り込めた事がわかった．

3.2　メソッドの類似度計算の提案

引数，戻り値，ローカル変数のメトリクスの完全一致からメソッドの絞込みの可能性があることがわかったが，完全一致の検索では実際の開発では非現実的である．そこで，本提案ではローカル変数の型ごとの個数からなるメトリクスベクトル間の類似度をコサイン関数により計測する．本研究で用いるコサイン類似度の式は以下である．

$$cosine\_similarity = \frac{\sum_{i=1}^{|V|} A_i B_i}{\sqrt{\sum_{i=1}^{|V|} A_i^2}\sqrt{\sum_{i=1}^{|V|} B_i^2}} \cdots (1)$$

A：検索対象のメソッドのメトリクスベクトル
B：開発中のメソッドのメトリクスベクトル
$|V|$：メトリクスベクトルの要素数

コサイン類似度はベクトル空間モデルであり，2 つのベクトルのなす角の大きさを 0 から 1 の値を示す事で，値が 1 に近づけば 2 つのベクトルはより類似していると捉える．これにより，メトリクスが完全一致だけではなく，部分一致でも類似メソッドの抽出することができる．

4　調査

本提案をオープンソースの 60 プロジェクト(Java ファイルで 26,493 個)に検索対象の Java ファイル(Java ファイルで 99 個)を用いて調査を行った．オープンソースには Apache Ant や Jedit 等が含まれており，メソッド数は 232,715 個である．検索対象の Java ファイルは本学の授業科目である Java 演習で学生が作成したソースコードであり，メソッド数は 74 個である．

調査結果，74 個のメソッドのうち，最も絞込み結果の個数が少ないメソッドは 23 個であり，最も絞込み結果が多いメソッドで 4876 個となった．また，平均は 3445 個となった．つまり，232,715 個から平均で 3445 個の 1.5%まで絞り込むことができた．また，検索対象のソースから数個，無作為に選別しローカル変数を数個削除し，絞込みをした結果，目的とするメソッドを検出できた事が確認できた．

5　おわりに

本研究では，Java 言語を対象にオープンソースと開発者の開発中のソースコードから類似メソッドを絞り込むための手法の提案と調査を行った．調査の結果，232,715 個のメソッドから目的のメソッドを 1.5%まで絞り込むことができた．今後は，コサイン類似度にオブジェクト型を含め，更なるメソッド絞込みの精度を高める．

参考文献

[1]T. Kamiya, S. Kusumoto, K. Inoue : CCFinder: a multilinguistic token-based code clone detection system for large scale source code , IEEE Transactions on SE-28-7, PP. 654-670 (2002) .
[2] GitHub , https://github.com/search
[3] 横森励士, 梅森文彰, 西秀雄, 山本哲男, 松下誠, 楠本真二, 井上克郎: Java ソフトウェア部品検索システム SPARS-J, 電子情報通信学会論文誌 D-I, Vol. J87-D-I, No. 12, pp. 1060-1068 (2004).
[4] 石原友也, 堀田圭祐, 肥後芳樹: 再利用実績に基づいたコード片検索手法の提案, 電子情報通信学会技術研究報告, ソフトウェアサイエンス 113(269), pp. 61-66 (2013).