

ソフトウェア部品の階層的構成関係による ソースコードの類似性評価

渥 美 紀 寿[†] 山本 晋一郎^{††}

本稿ではソースコード中で利用しているソフトウェア部品を構成する、低レベルなソフトウェア部品によるソースコードの類似性の評価および、ソフトウェア部品の構成関係に基づく階層的な類似関係による類似ソースコード郡の分類の必要性について述べる。

1. はじめに

本章ではアンケート回答者である渥美について、ソースコード類似性との関わり、これまでのソフトウェア開発経験、Java との関わりについて述べる。

これまでにソフトウェア部品の利用例の分類に関する研究、利用しているソフトウェア部品によるソフトウェアの分類に関する研究を行ってきた。これらの研究ではソースコード中に現れる、汎用的に利用されているソフトウェア部品に着目し、ソフトウェア部品の利用例やソフトウェアの類似性の評価を行った。ソースコードそのものの類似性についての研究は行っていないが、ソースコードから得られる情報を利用したソフトウェアの類似性の評価を行ってきた。

ソフトウェアの開発経験は 10 年程度である。これまでに開発したソフトウェアの多くは C 言語であるが、Web アプリケーションや簡単な GUI アプリケーションなど Java を利用したソフトウェアもいくつか開発している。

2. アンケート回答

本章では、渥美が行ったアンケートの回答結果と、その回答理由について述べる。

2.2 ソース No.2 についての回答理由

観点 A, C において yes と回答したが、不用意にメソッドのアクセス修飾子を public にすると、フィールド i の値が変更可能になってしまうため、注意が必要である。

[†] 南山大学

Nanzan University

^{††} 愛知県立大学

Aichi Prefectural University

表 1 設問への回答

ソース No.	A	B	C	D	X
1	yes	yes	yes	yes	-
2	yes	yes	yes	yes	-
3	no	yes	yes	yes	-
4	no	yes	no	yes	-
5	yes	yes	yes	yes	-
6	yes	yes	yes	yes	-
7	yes	yes	yes	yes	-
8	yes	no	yes	yes	-
9	no	yes	yes	yes	-
10	yes	yes	yes	yes	-
11	no	no	no	no	-

2.3 ソース No.3 についての回答理由

インタフェースメソッド getElementById が不要で、コード片 3-2 にまとめるのであれば良いが、不要なメソッドを提供するようにすべきではない。コード片 3-1, コード片 3-2 の両方を必要としているのなら、継承を利用して重複コードを減らすべきである。

2.5 ソース No.5 についての回答理由

メソッド add, get だけを利用しており、その他の個々のクラス独自のメソッドを利用していないため、類似していると判断した。

2.8 ソース No.8 についての回答理由

異なるアルゴリズムの実装を必要としないのであれば、どちらか一つにすべきである。

2.11 ソース No.11 についての回答理由

コード片 11-1 においてフィールド i の持つ意味として、true と false しかないのであれば、まとめるべきであるが、そうでなければ、まとめることはできない。しかし、i の持つ意味をこの 2 つのコード片だけで判断することはできないため、類似しているかどうかを判断することはできない。

3. 議 論

我々は機能を実現したソースコードを検索する手法について研究している。開発者が必要な機能の実現方法を検索する場合、必要なソフトウェア部品およびそのソフトウェア部品の構成方法を求めている。これを実現するために、我々は機能を実現したソースコードをその実現方法によって分類する手法について研究している。そのため、アンケートの設問のようなソースコード中の字句レベルでの類似評価ではなく、機能レベルの類似評価が必要である。

Google Code Search や SPARS-J²⁾、Koders などのソースコード検索システムを用いて、キーワードにより、既存のオープンソースソフトウェアからソースコードを検索することが可能である。しかし、検索結果は膨大になり、膨大な検索結果の中から開発中のソフトウェアで利用可能なソースコードを選択することは非常に困難である。

ソフトウェアは関数ライブラリやクラスライブラリなどの汎用的に利用可能なソフトウェア部品を用いて構築される。ソフトウェア部品は低レベルな機能を実現したものであり、それらを組み合わせることによってソフトウェアが構築されている。そこで、ソフトウェア部品の組み合わせが機能を表現していると捉え、利用しているソフトウェア部品によってソフトウェア間の類似性を評価する手法を提案した¹⁾。C 言語を対象とした実験では、図 3 のように類似したソフトウェアを 2 次元上で近くに配置することができた。図 3 の右側の拡大部分では、ネットワーク情報サービス (NIS) 関連のコマンドや遠隔手続き呼び出し (RPC) 関連のコマンドなど、ネットワーク関連のコマンド群が集まっている。

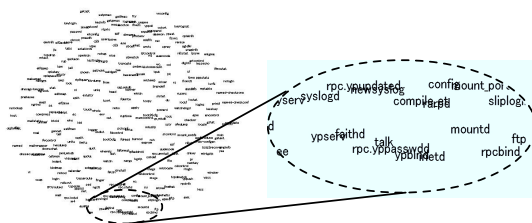


図 1 FreeBSD コマンドの 2 次元配置

ある機能を実現したソフトウェア部品を分類するためには、ソフトウェア全体ではなく、その一部で利用しているソフトウェア部品がどのように構成されているかを調べる必要がある。しかし、低レベルなソフトウェア部品を組み合わせることでより高レベルなソフトウェ

ア部品が構成され、ソフトウェア部品の構成は階層的になっている。ソフトウェアによってソフトウェア部品の構成関係は異なっており、単純にソフトウェア部品間での比較では実現している機能によって分類することができない。また、開発者は必要なソフトウェア部品を知りたいだけでなく、それらの構成方法も求めているため、同じ機能を実現したソフトウェア部品がそれぞれどのように個々の低レベルなソフトウェア部品を構成しているかを示すことが必要である。

例えば、データベースへの問い合わせをするソフトウェア部品として、JDBC だけを利用したソフトウェア部品と、Hibernate を利用したソフトウェア部品があるとする。JDBC だけを用いたソフトウェア部品では、DriverManager, Connection, PreparedStatement, ResultSet などのソフトウェア部品が利用されるが、Hibernate を用いたソフトウェア部品では、SessionFactory, Transaction, Query などのソフトウェア部品が利用される。Hibernate は JDBC を利用した高レベルなソフトウェア部品であるため、Hibernate のソースコードには JDBC のソフトウェア部品である、DriverManager, Connection, PreparedStatement などが利用されている。ソフトウェア部品間の構成関係を利用することにより、これらのソフトウェア部品が類似したソフトウェアとして分類することが可能となる。

そこで我々はソフトウェア部品の階層的な構成関係を考慮したソースコードの類似性の評価が必要であると考えている。具体的には、ソフトウェア部品の階層構造における低レベルのソフトウェア部品群により、機能の類似性を評価する。また、ソフトウェア部品の階層構造の類似性により、ソフトウェア部品の構成方法の類似性を評価する。これらの評価に基づきソフトウェア部品群を分類することにより、利用しているソフトウェア部品の違いおよびその構成方法の違いを明確にすることが可能となる。

参 考 文 献

- 1) 渥美紀寿, 末次亮, 山本晋一郎: 多次元尺度構成法によるソースコード検索におけるソフトウェアの可視化, ウィンターワークショップ 2009・イン・宮崎 論文集, Vol. 2009, No. 3, pp. 23-24 (2009). 情報処理学会シンポジウムシリーズ.
- 2) 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).