

クラス間の関係理解のための メソッド呼び出し関係の可視化によるコードリーディング支援

Code Reading Support by Visualization of Method Call Relationship for Understanding the Relationship between Classes

鈴木 英梨花^{*1}, 酒井 三四郎^{*2}
Erika SUZUKI^{*1}, Sanshiro Sakai^{*2}

^{*1}^{*2} 静岡大学情報学部

^{*1}^{*2}Faculty of Informatics, Shizuoka University

Email: suzuki@sakailab.info

あらまし: コードリーディングが苦手な人にとって, 規模が大きなソースコードを読んで理解することは困難であり, 本研究ではコードリーディング支援のためのツールを提案, 開発した. 評価実験を行った結果, 提案ツールは明確に有効であるとはいえなかったが, 実装した機能はコードリーディング時に有効的に使用されていることがわかった.

キーワード: Java, コードリーディング, 可視化ツール

1. はじめに

プログラミング演習に取り組んでいる人は, その課題として他者が書いたソースコードを元にしてそこに新たな機能を追加することがある. その際に, まず対象のソースコードの概要を理解する必要がある. クラス数が 2,3 個程度のソースコードではクラスを順に見ていくことで概要を理解することができるが, クラス数が 10 個程度になるような規模が大きなソースコードの概要を掴むことは, 不慣れな人には困難である. 本研究では他者が書いたプログラムを理解するためのメソッド呼び出し関係に着目したコードリーディング支援ツールの開発を行う.

2. 関連ツール

これまでもプログラムを理解するための可視化ツールは存在している.

AmaterasUML⁽¹⁾はソースコードからクラス図を自動生成する Eclipse のプラグインで, クラスを複数選択するとそのクラス間の関係を可視化できる.

astah UML⁽²⁾はソースコード全体のクラス図を一度に生成するツールで, ノードの自動配置を行う.

Udoc⁽³⁾はクラス図を段階的に自動生成するツールである. 一つクラスを選択すると, そのクラスと関係のあるクラスが放射線上に展開される. さらに展開されたクラスを選択するとそのクラスと関係のあるクラスが展開され, 段階的に図が生成されていく.

Eclipse のプラグインである ispace⁽⁴⁾は Java のソースコードからコールグラフを自動生成するツールである. ノードはパッケージ, クラス, メソッドなど複数種類ある.

これらのツールには以下の 3 点の問題点がある.

- (1) クラス図は学習者にとってあまり馴染みがないため, クラス図による構造の理解は困難である

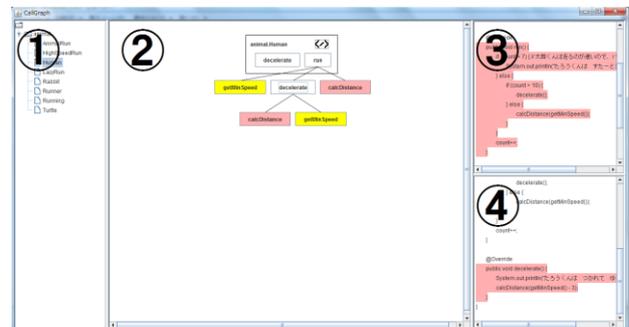


図 1 メソッド呼び出し可視化ツールの実行画面

- (2) プログラム全体を一度に図にするツールは線やノードが多すぎてしまい見づらくなる
- (3) 複数種類のノードがあると一目で理解するのが難しくなる

3. メソッド呼び出し関係可視化ツールの提案

本研究で提案するのは関連ツールで挙げた問題点を解消し, コードリーディング時により利用しやすいツールである. そのため, 学習者が知識を持たないクラス図ではなくメソッド呼び出し関係を可視化するツールを作成した. ツールの特徴は以下の通りである.

- (a) 選択したクラスを基点に図が展開
- (b) 段階的に図が展開
- (c) ノードに対応するソースコードの参照が可能
- (d) ソースコードを同時に参照できるので呼び出し元と呼び出し先の両方を参照可能
- (e) ノードがインターフェースのメソッドの場合, 実装クラスの一覧を参照可能

図 1 は作成したツールの実行画面である. 図 1 中

の①はソースコードのクラス一覧，②はメソッド呼び出しを可視化した図を表示するパネル，③と④はソースコード表示用のパネルである。

4. 評価実験

評価実験を行い，作成したツールの有効性を示す。本実験では提案ツールと統合開発環境 Eclipse の同等の機能との有効性を比較する。

4.1 仮説

本実験では以下の仮説の検証を行う。

仮説 図を利用することでプログラムの概要の理解が容易になる

4.2 実験方法

被験者はCSプログラムの3年生8人で，被験者を表1の通りに4群に分け，提案ツールと統合開発環境 Eclipse を使用して2つの問題A,Bに取り組んでもらう。問題A,Bはそれぞれ別のソースコードを用意する。それぞれツールを使用してソースコードを読んでもらい，用意した問いに答えてもらおうと同時に，その様子を観察する。問題終了後にアンケートを実施し，提案ツールとEclipseと比較して理解の容易さやメリット，デメリットを解答してもらおう。実験結果と観察，アンケートから仮説の検証を行い，ツールの有効性を評価する。提案ツールとEclipseを使用したときのそれぞれの正答率と解答時間を比較し，提案ツールの正答率が高く，解答時間が短ければ提案ツールの方がプログラムを理解しやすいといえる。

4.3 実験結果と考察

実験結果は表2の通りになった。平均正答率，平均解答時間ともに明確な差はなく，どちらに対してもt検定を行ったが優位な差は認められなかった。その理由として被験者が少なかったことや問題として扱ったソースコードの規模が小さかったことが挙げられる。今回使用したソースコードはクラス数4,5個，インターフェース数が3個であり，その程度であればツールに頼らなくても解くことができてしまった可能性がある。また，被験者全員がEclipseを利用したことがあるのに対し，提案ツールは今回初めて使用したため，その使用方法を覚えることや，各機能を使いこなすことが難しく，ツールによる明確な効果が得られなかったのではないかと推測できる。他に，被験者のプログラミング能力や知識を測ることを行わなかったことで被験者間の能力差によって正答率や解答時間にばらつきが出てしまったことも考えられる。

表1 問題と使用ツールの組み合わせ

	1 問目	使用ツール	2 問目	使用ツール
1 群	A	提案ツール	B	Eclipse
2 群	A	Eclipse	B	提案ツール
3 群	B	提案ツール	A	Eclipse
4 群	B	Eclipse	A	提案ツール

実験後に被験者に行ったアンケートの「どちらのツールを利用したときの方がプログラムの理解が深まったと感じたか」という質問に対し，提案ツールと答えたのが7人，Eclipseと答えたのが1人だった。

提案ツールと答えた理由として「表示されていることが少なかった」「動作が簡単だった」というものがあつた。提案ツールはメソッド呼び出しを追うための機能に絞って実装しており，必要な動作や表示が少ない。一方Eclipseは様々な機能が実装されている分，情報量が多いと感じる被験者もいると分かった。

実験中，同じインターフェースのメソッドを実装している2つのメソッドを比較する際，実装クラスの一覧から必要なソースコードを選び参照している被験者や，一方のビューに呼び出し元メソッド，もう一方のビューに呼び出し先メソッドを表示しソースコードを読む被験者がいた。被験者が行ったこれらの動作は提案ツールに実装したメソッド呼び出し関係を理解するための機能を用いたものであり，コードリーディング中に有効的に使用されることが分かった。

本実験ではプログラミングの学習をある程度終えた人を対象としていたが，結果として提案ツールは明確に有効性があるとはいえなかった。そこで，提案ツールの対象者をプログラミング初学者とし，諸学者がプログラムの構造を理解するために使用することも検討したい。

5. おわりに

本研究ではコードリーディング支援ツールとしてメソッド呼び出しを可視化するツールの提案を行った。提案ツールは明確な有効性があるとはいえないが，被験者の多くが提案ツールはソースコード理解に有効的であると感じていた。今後はツールを使用する対象者の再考，ツールの機能の向上や実験内容の見直しを行い，再度提案ツールの有効性を測る。

参考文献

- (1) AmaterasUMLProjectAmateras,<http://amateras.osdn.jp/cgi-bin/fswiki/wiki.cgi?page=AmaterasUML> (2016/11/16 参照)
- (2) astah システム設計，ソフトウェア開発支援ツール | Astah,<http://astah.change-vision.com/ja/> (2016/11/16 参照)
- (3) The UDoc project,<http://udoc.sourceforge.net/main/index.html> (2016/11/16 参照)
- (4) ispace: Ispace-Home browse, <http://web.archive.org/web/20111203070026/http://ispace.scribor.de/index.php?n=Ispace.Home> (2016/11/16 参照)

表2 実験結果

使用ツール	平均正答率	平均解答時間
提案ツール	58.3%	21 分 34 秒
Eclipse	55.8%	20 分 49 秒