N-009

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† 酒井 三四郎†
Erika Suzuki Sanshiro Sakai

1. はじめに

プログラミングにある程度慣れた学習者がプログラミング演習に取り組むとき，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つかむことは，コードリーディングに不慣れな人には困難である．このような背景から，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. 関連研究

これまでにもプログラムの概要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．

astahUML[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状に展開される．さらに展開されたクラスの一つ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．

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 Eclipse のプラグインである．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，本研究の対象者にとって以下の 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，クラス図によるソースコードの概要の理解は困難である
(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研究で上げた問題点を解消し，

† 静岡大学大学院総合科学技術研究科
Shizuoka University Graduate School of Integrated Science

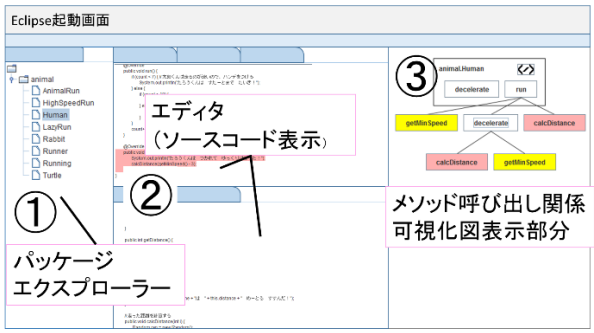

図 1：Eclipse プラグイン化したツールの概要図

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，ツールでは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の可視化を行う．さらにツールを Eclipse のプラグインとして動作させることで，宣言の参照やホバー表示など Eclipse のコードリーディング支援に役立つ機能も使用できる．また，Eclipse ではソースコードの編集，実行も行うことができるため，可視化された図を見ながらのソースコードの編集を行い，実行してその挙動を確認することが可能になる．図 1 はメソッド呼び出し可視化ツールを Eclipse のプラグインとして実現したときの画面構成を示したものである．①はパッケージエクスプローラーであり，ここからメソッド呼び出し関係を可視化させたいクラスを選ぶ．②はエディタであり，図のノードに対応するソースコードを複数参照できる．③はメソッド呼び出し関係を可視化した図を表示するビューである．

Eclipse のプラグインとしてツールを実装する前に，まずはメソッド呼び出し関係を可視化した図がソースコード理解に有効であるかということを示すために，スタンドアロンなツールを開発した．ツールの特徴は以下の通りである．

(a) 選択したクラスを起点に図を展開
(b) 段階的に図を展開
(c) ノードはメソッドであり，ノードに対応するソースコードを参照可能
(d) ソースコードを 2 つ表示できるため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 2 はメソッド呼び出し関係を可視化した図である．Human クラスが起点となり，図が展開していく．①の run メソッドをクリックすると run メソッドが呼び出している②のメソッド群が表示される．次に②のメソッド群の中の decelerate メソッドをクリックすると decelerate メソッドが呼び出している③のメソッド群が表示される．なお、各メソッドのクラス名はメソッドのノードにマウスオーバーす

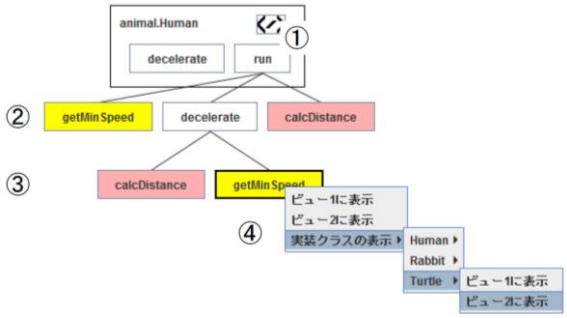

図 2　メソッド呼び出し関係図

ることで表示される．また，同じメソッドが複数表示される場合、同じ色で示す．④はソースコードを参照するためのメニュー部分であり、ソースコードを参照したいメソッドのノードを右クリックすると表示される．

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（呼び出し階層を開く，型階層を開く）との有効性を比較する．このとき，Eclipse はプラグインなどの Eclipse の標準の機能以外は使わないこととする．

4.1 仮説

本実験では以下の仮設の検証を行う．

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

被験者は静岡大学情報学部 CS プログラムの 3 年生 8 人で，被験者を表 1 のとおりに 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 を比較して理解の容易さやメリット，デメリットを回答してもらう．問題の正答率と解答時間，実験の観察，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．

4.3 実験結果と考察

実験結果は表 2 の通りであった．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 検定を行ったが有意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事前に測ら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．

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時のほう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「表示されているものが少なかった」「動作が簡単だった」というものがあった．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．

表 1：問題と使用ツールの組み合わせ

	1 回目	使用ツール	2 回目	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

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．実験結果から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支援に有効的であると感じていることがわかった．このことから，メソッド呼び出しの可視化はソースコードの概要をつかむことに有効であるといえる．今後は今回の評価実験で挙げられた問題点やツールの不便な点を改善し，Eclipse プラグイン化した可視化ツールの評価実験を行い，再度有効性を評価する．

参考文献

[1] Project Amateras : AmaterasUML Project Amateras, http://amateras.osdn.jp/cgi-bin/fswiki/wiki.cgi?page=AmaterasUML (2016/11/16 参照)
[2] Change Vision: 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] Christopher Deckers: The UDoc project, http://udoc.sourceforge.net/main/index.html （2016/11/16 参照）
[4] ispace Team: ispace : Ispace - Home browse, http://web.archive.org/web/20111203070026/http://ispace.stribor.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