BlockEditor における VisualScopeChecker の学習効果の分析

Analysis of learning effect of VisualScopeChecker in BlockEditor

田中　良樹[*1]，　松澤　芳昭[*2]，　酒井　三四郎[*3]
Yoshiki TANAKA[*1]，　Yoshiaki MATSUZAWA[*2]，　Sanshiro SAKAI[*3]
*1*2*3 静岡大学情報学部
*1*2*3 Faculty of Informatics, Shizuoka University
Email: cs11056@s.inf.shizuoka.ac.jp

あらまし：ビジュアルプログラミング環境である BlockEditor における VisualScopeChecker の学習効果を分析した．文科系の大学生に対し分析を行ったところ，VisualScopeChecker は Java 言語のコンパイルエラー文よりも理解しやすくスコープの理解に役立つ可能性があるということが分かった．
キーワード：プログラミング学習，ビジュアルプログラミング

1.　はじめに

現在，初学者向けの学習環境として，ブロック型ビジュアルプログラミング言語とテキスト記述型言語の相互変換を行う学習システム BlockEditor[(1)]が利用されている．BlockEditor ではブロック型ビジュアルプログラミング言語とテキスト記述型言語（Java）の相互変換が可能である．

本研究では，BlockEditor のアニメーション付きスコープチェック機能（VisualScopeChecker）に着目し，学習者のプログラミング編集履歴を利用して学習者にインタビューを行うことで，この機能の学習効果を分析した．

2.　先行研究

ビジュアルプログラミング言語を用いた教育支援システムとして，Squeak Etoys[(2)]や Scratch[(3)]が挙げられる．ビジュアルプログラミング言語はテキスト記述型言語と異なり文法エラーを回避できることから，アルゴリズムの構築や理解に集中することができ，プログラミング初学者にとって使いやすい言語とされている．

BlockEditor の利用率の分析は松澤ら[(1)]によって行われている．BlockEditor が Java によるプログラミング構築能力の足場かけとして有用に機能しているが，BlockEditor が理解度の向上に寄与した直接的なデータは得られていないとされている．

3.　分析の目的

BlockEditor の機能の一つとして，リアルタイムなスコープのチェック機能がある．この機能はスコープの異なる変数を参照した場合，ブロックを反発させてブロックの連結を不可能にし，同時にスコープの異なっている参照のブロックをハイライト表示するという機能である．図 1 にスコープが誤っている例と，そのときの VSC の動作イメージを示す．この機能を VisualScopeChecker（以下 VSC)とする．

VSC はアニメーションによって視覚的にスコープの違いを理解できる．そのためプログラミング初学者にとって，テキスト型言語におけるコンパイルエラー文よりも，VSC の方がスコープによるプログラムの誤りの理解，プログラムの修正が容易であると考えられる．

本実験ではこのビジュアル言語特有の視覚的なプログラミング編集補助機能に着目し，この機能が学習者の理解度の向上に貢献しているかを明らかにする．具体的には，学習者の学習者のプログラミング編集履歴から VSC の動作した部分を発見し，分析を行うことで，VSC の学習効果を明らかにする．

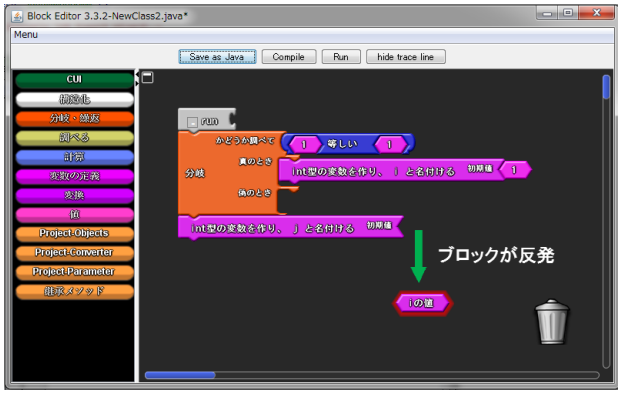

図 1 VSC の動作イメージ

4.　分析ツール

現在，ログデータの復元には PPV(Programming Process Visualizer）[(4)]が使用されている．PPV はテキスト言語でのログデータを解析するツールである．

今回はこの PPV に機能を追加して BIVi(Block Image Visualizer）を開発した．図 2 に BIVi の動作イメージを示す．BIVi は BlockEditor での編集履歴を復元して閲覧可能にするツールである．ブロックがキャンパスに追加されたとき，ブロック同士が連結したとき，ブロックが分離したときの各ブロックの様子を観察することができる．今回はそれに加え VSC が動作した際の様子も観察できるように実装を行った．

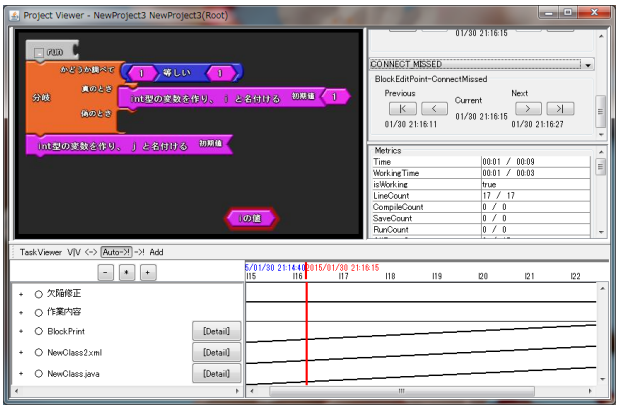

図 2 BlockImageVisualizer

5. 実験方法

5.1 仮説

今回の実験で検証する仮説は以下の 2 つである．

[仮説 1] VSC がスコープの誤りの理解に貢献している

[仮説 2] VSC は Java で記述した際のコンパイルエラーに比べてスコープの誤りを理解しやすい

5.2 実験対象

2014 年静岡大学情報社会学科 1 年生後期の講義（プログラミング）を受講中の学生 9 名を対象に実験を行った．被験者は授業内でブロック言語と Java 言語のどちらでもプログラミングを行った経験があり，スコープの概念については学習済みである．

5.3 実験手順

実験を始める前に実験担当者から被験者に対して BlockEditor の操作の確認と，完成させるプログラムの説明を行った．次に，被験者は BlockEditor を用いてプログラミングを行った．プログラミングに行き詰まった被験者には，実験担当者が段階的にヒントを出し，最終的に完成するまで誘導を続けた．プログラムが完成したら，実験担当者が BIVi を用いてプログラミングの履歴の分析を行った．VSC が動作した部分があればその点についてインタビューを行った．

インタビューは以下の 2 点を中心に行うものとする．

- VSC は理解に役立ったか
- VSC は Java で記述した際のコンパイルエラーに比べて理解しやすいか

6. 実験結果

実験の結果，9 名全てがプログラムを完成させるまでの間に VSC を動作させた．

9 名中 1 名は VSC によって変数のスコープが誤っているということに気付き，直ちにプログラムを完成させることができた．6 名は，参照できない変数ということに気付いたが，ヒントなしではプログラムを完成させることができなかった．残りの 2 名は，プログラムのどこかが間違っているということには気付けたが，変数が間違っているということには気付けなかった．

VSC によって問題を解決できなかった被験者 8 名のうち，6 名は Java によるコンパイルエラーより VSC の方が分かりやすいと評価した．評価の理由としては，スコープの違っているブロックが反発することによって明確に違いを認識できること，初学者にとって Java のコンパイルエラー文が難解であること，視覚的にスコープの違いを把握できることなどが挙げられた．

7. おわりに

VSC によってプログラムの完成に至った被験者は 1 名のみであったが，半分以上の被験者が VSC によって参照できない変数であったということに気付いていた．スコープの誤りに気付くためには参照できない変数であることに気付く必要があることから，VSC はスコープの誤りの理解に，ある程度貢献しているのではないかと考えられる．

VSC によってプログラムの完成には至らなかったものの，Java のコンパイルエラーに比べ，VSC の方が誤りを理解しやすいと評価した被験者が多数存在したこと，そして，評価の理由は正当であったことから，仮説 2 は支持されると考えられる．

本研究では，初学者にとってスコープはコンパイルエラー文よりも視覚的に構文の誤りを表示する機能の方が理解しやすいということが明らかになった．今後は，スコープだけでなく他の構文エラーにおいても同様に視覚的に構文の誤りを動的に表すような機能を実装し，学習者の理解に貢献できるのかを検証していきたいと考えている．

参考文献

(1) 松澤芳昭，保井元，杉浦学，酒井三四郎：ビジュアル-Java 相互変換によるシームレスな言語移行を指向したプログラミング学習環境の提案と評価，情報処理学会論文誌 2014-01-15(55), pp.57-71(2014)

(2) Ingalls, D., Kaehler, T., Maloney, J., Wallace, S. and Kay, A.: Back to the Future: The Story of Squeak, A Practical Smalltalk Written in Itself, Proc. ACM OOPSLA `97, pp.318-326 (1997)

(3) Scratch Team Lifelong Kingdergarten Group MIT Media Lab/:Scratch -imegine.program.share-http://scratch.mit.edu/(2015 2/6 取得)

(4) 松澤芳昭，岡田 健，酒井三四郎：＂Programming Pro-cess Visualizer：プログラミングプロセス学習を可能にするプロセス観察ツールの提案＂，情報処理学会情報処理シンポジウム SSS2012，pp. 267-264（2012)