# プログラムの動作の可視化によるプログラムの動作理解の支援

宮崎壮麻*1・TRAN THANH TUNG*2・高瀬治彦*2・北英彦*2
Email: kita_yh@yahoo。co。jp

*1: 三重大学工学部電気電子工学科
*2: 三重大学大学院工学研究科

◎Key Words　　動作理解支援，可視化システム，プログラム修正

## 1. はじめに

近年，情報社会の発展とともにプログラミング学習の重要性が高まっている。一般にプログラミング学習方法として，演習型の授業が行われている。プログラム作成の演習において，学生の中にはプログラムを実行してみて動作の間違いことに気が付いてもそれを取り除けないものがいる。間違いを取り除くためには，実行途中のどこで・どうして正しい動作をしなくなったのかを知る必要がある。熟練者は頭の中でプログラムを 1 行ずつ実行することで変数や配列の値がどのように変化するのかを考えるが，初学者はこれを行うことが難しい。

プログラムの配列や変数の値を知る方法として，一般的にデバッガが用いられる。しかし，デバッガは初学者向けに作られていないことが多い。そこで，プログラミング初学者のためにプログラム実行中の配列や値を可視化し，プログラムの動作確認の支援をする可視化システムが開発されており[1,2]，可視化システムはプログラミング学習に効果があると知られている[3,4]。既存のものはプログラムのある行の実行直後の変数や配列の値を図的に表示するものであるが，どの変数やどの配列の要素が変化したのが把握しにくい。ステップごとの変数や配列の値を可視化することで，学習者は各ステップでの変化が分かりやすくなる。しかし，プログラムの動作を理解するために，ステップごとの変化だけでなく，ブロックやメソッドを通した時，変数や配列の値がどのように変化するのかを把握することも重要である。既存の多くの可視化システムはステップごとの変化しか行われていないため，大局的なプログラムの変化の理解を支援していない。

本研究では，初学者向けの可視化システムを提案する。ステップごとの実行前後の比較の可視化，ブロックの実行前後の比較の可視化，という 2 種類の可視化を行い，学習者に動作理解の支援をする。ステップごとの実行前後の比較では，ステップごとの実行前と実行後を対比して表示し，変化した部分に色付けをする。変化した理由が明示する。ブロックの実行前後の比較では，ブロックの実行前後を比較し，変化した部分に色付けをする。これらの変化を強調し，可視化することでプログラムがどのように動作し，変化するのかを支援する。最後に構築した可視化システムを初学者に使用してもらい，アンケート調査により本システムの有効性を確かめた。

## 2. 先行研究

プログラムの動作理解の支援を行うシステムに関する先行研究として伊藤の可視化システム紹介する。

本研究室の院生である伊藤は，Java を学習する初学者を対象としてプログラムの動作中の配列や変数の値の移り変わりを可視化するシステムを開発した[5]。このシステムは既存システム Online Python Tutor[6]を拡張して，1 行の実行の前後の変数・配列の値の変化を 1 つの画面で閲覧できるようになった。また，変数・配列の値の変化している部分に色づけをして強調表示を行う。可視化部分に実行前後の状態を表示して変化を示すことで，プログラムのソースコード 1 行の実行前後の動作を確認できるようになっている。伊藤の可視化システムの使用画面を図 1・図 2 に示す。

伊藤のシステムはステップ実行時に実行前後を比較し，変化する変数・配列の値に色付けし，可視化を行っているが，なぜ次のステップで変数・配列の値が変化するのか理由を提供していない。また，ブロック実行時に実行前後の比較を行なっていない。

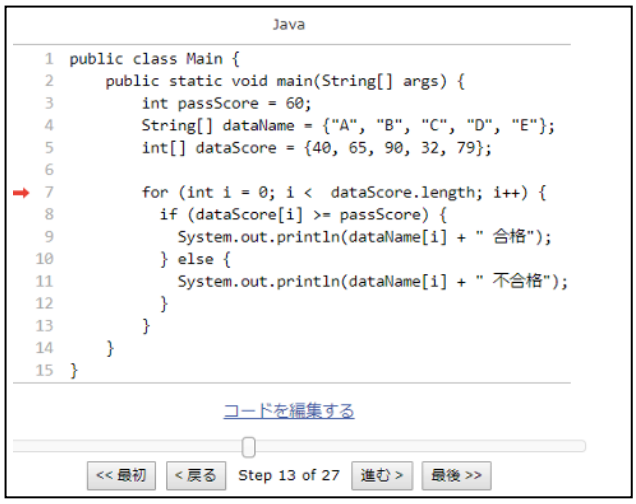


図 1　伊藤のシステムのソースコード部分

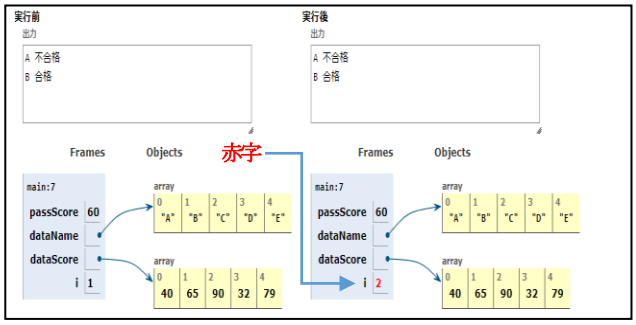


図 2　伊藤のシステムの可視化部分

## 3. 提案手法

学習者から受け取ったソースコードより，抽象構文木を生成する。生成した抽象構文木を分析し，受け取ったソースコードの要素とプログラム全体のトレースの間の関係を見つけることでプログラムの変化を把握し，変化の可視化を行う。

本システムは，Online Python Tutor と伊藤のシステムを拡張することで実装を行い，プログラムの動作理解を支援するために以下の 2 つの手法を提案する。

(1) ステップ実行時に変化する理由の可視化
(2) ブロックの実行前後の比較

### 3.1 ステップ実行時に変化する理由の可視化

プログラムのステップごとの実行前後の変化を可視化する。可視化を行っている命令に関係のある条件判定や繰り返し文の評価式の内容を表示し，その判定結果 TRUE, FALSE を表示する。同時に，計算式の内容を可視化部分に表示する。これらの表示により，学習者に変数・配列の値が変化した理由の把握を支援する。

実行時に変化する理由の可視化機能について説明する。図 3 の while 文は n > 0 の条件式により n の条件を確認する。n=234 を条件式に代入し，条件 n>0 をチェックした後に TRUE の判定結果が図 4 のように表示される。

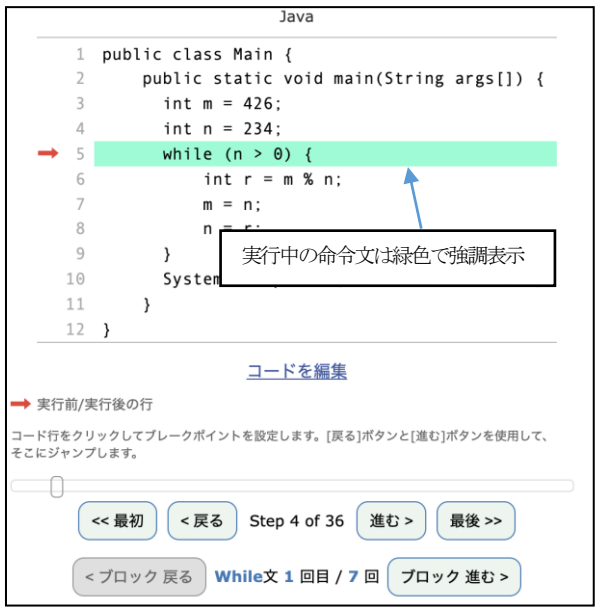


図 3　本システムのソースコード部分

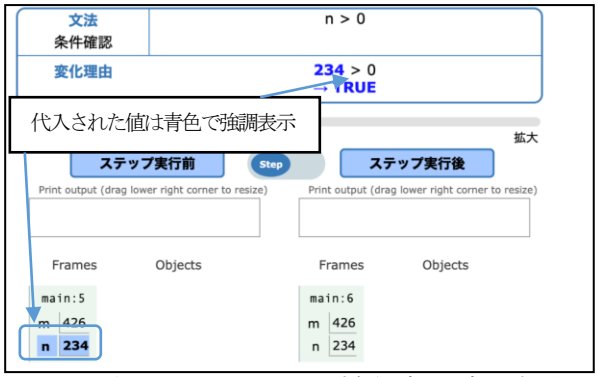


図 4　本システムのステップ実行時の可視化部分

### 3.2 ブロックの実行前後の比較

プログラムのブロック単位での実行前後の変化を可視化する。プログラムの繰り返し文やメソッドの動作は 1 ステップずつ実行を追いかけるだけでは理解しにくい時がある。ブロック単位で配列・変数の値の変化が可視化されることにより，繰り返しブロックの 1 ループごとの配列・変数の変化など，複数ステップを通した後，プログラムがどのように変化するのか分かりやすくなる。ブロックの実行前後を比較し，配列や変数の値が変更された箇所を強調表示することで，値がいつ変更されたのか分かりやすくなる。ブロックの実行前後の変化を表示することで，マクロな視点でプログラムの動作の把握を支援する。

図 5 に示すソースコードにより，繰り返しブロック実行前後の比較機能について説明する。図 5 の繰り返しブロックを操作する部分より，for ループは全部 6 回があり，2 回目のループを実行していることが分かる。繰り返しブロックの可視化を操作するブロック戻るボタンとブロック進むボタンは繰り返しブロックを実行時のみ有効になる。これらのボタンを操作することで繰り返しブロックの各ループの実行前後の比較が確認できる。

図 5 に示すソースコードのステップ 13 で実行する 13 行 for 繰り返しブロック実行前後の比較を図 6 に示す。for 繰り返しブロック 2 回目ループを実行する前の値と，2 回目ループを実行した後の値の両方を表示し，実行後で変化する値を強調表示する。図 6 の結果より，for 文が 2 回目ループをした後，変数 i と配列 resultArray の位置 1 の値が変化することが一目で分かる。

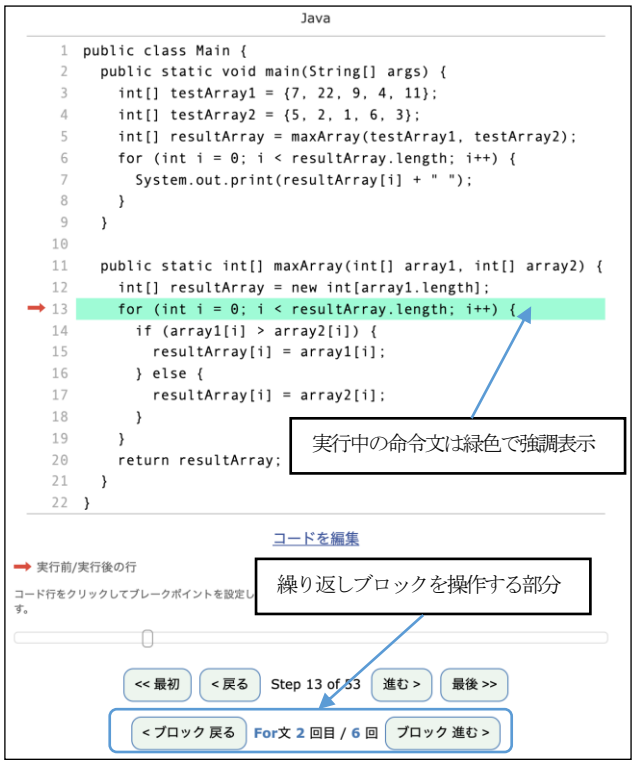


図 5　繰り返しブロック実行のソースコード部分

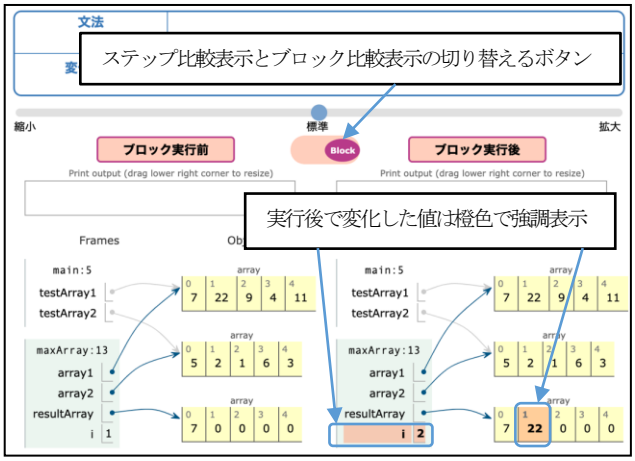


図 6　繰り返しブロック実行前後の比較の可視化部分

## 4.　実験とアンケート調査

本システムの提案により，学習者のプログラムの動作追跡に有効であるかを確認するため，以下の実験とアンケート調査を行なった。

### 4.1　実験

学習者は以下の 3 つの作業（各作業時間：20 分）を通して, 3 つのプログラムの誤りを修正してもらう形式で実験した。

(1)　可視化システムを使わずにプログラムを修正
(2)　Online Python Tutor を用いてプログラムを修正
(3)　本システムを用いてプログラムを修正

3 つのプログラムは簡単な順番になっているが, 学習者は作業中に次のプログラムに移ることを認める。実験後に学習者が回答してもらった報告結果より，デバッグできた人数を解析し, 従来のシステムと比較し, 本システムの有効性を確認する。

### 4.2　アンケート調査

本システムは学習者のプログラムの動作追跡に有効であるかを確認するため使用後にアンケート調査を行った。

- 質問Ⅰ，Ⅱ：本システムが提案した可視化機能が動作追跡するのに役に立ったかについての質問
- 質問Ⅲ，Ⅳ：学習者のコメントや感想

## 5.　結果と考察

4 章で行なった実験の報告結果とアンケート結果を以下に示す。

### 5.1　実験結果

各デバッグ作業の報告結果を図 9 に示す。プログラム 2 の報告では，作業 2（Online Python Tutor）で 48 人が「時間あってもできない」と答えたが, 本システムを使用した後，35 人に減った。この結果より，従来の可視化システム（Online Python Tutor）を使っても修正できない学習者が，本システムを使用すると修正できたことが分かった。また, 本システムは動作追跡の支援ツールであり, 本システムを使うと動作の間違いが必ずしも修正できるわけではないが，プログラムの動作追跡の支援に有効であることを確認した。

### 5.2　アンケート調査結果

**質問Ⅰ，Ⅱの内容と結果：**本システムが提案した機能の有効を確認するため学習者が本システムを使用した後, 質問Ⅰと質問Ⅱに回答してもらった。質問Ⅰの内容を図 7 に，その結果を表 1 に示す。質問Ⅱの内容を図 8 に，その結果を表 2 に示す。

**質問Ⅰ**：ステップごとを実行する際に可視化部分の変化理由の支援は役立ちましたか？

1.　役に立った
2.　どちらかといえば，役に立った
3.　どちらかといえば，役に立たなかった
4.　役に立たなかった

図 7　質問Ⅰの内容

表 1　質問Ⅰに対する回答

| | 変化理由の可視化は動作追跡に役に立ったと感じたか | | | | |
|---|---|---|---|---|---|
| 選択肢 | 1 | 2 | 3 | 4 | 合計 |
| 票数（人） | 39 | 29 | 12 | 6 | 86 |
| 百分率（%） | 45 | 34 | 14 | 7 | 100 |

**質問Ⅱ**：繰り返し文を実行する際に 1 つのループの実行前後の比較は役立ちましたか？

1.　役に立った
2.　どちらかといえば，役に立った
3.　どちらかといえば，役に立たなかった
4.　役に立たなかった

図 8　質問Ⅱの内容

| | ブロック実行前後の可視化は動作追跡に役に立ったと感じたか | | | | |
|---|---|---|---|---|---|
| 選択肢 | 1 | 2 | 3 | 4 | 合計 |
| 票数（人） | 39 | 23 | 18 | 6 | 86 |
| 百分率（%） | 45 | 27 | 21 | 7 | 100 |

表 2　質問Ⅱに対する回答

質問Ⅰ，Ⅱのアンケート結果より，「1（役に立った），2（どちらかといえば，役に立った）」の回答が 79%と 72%に達した。これより，本システムが提案した可視化機能は，プログラムの動作追跡を支援することに有効である。

**質問Ⅲ，Ⅳの内容と結果：**本システムの良い点と改善点を調べるため，使用後に学習者に質問Ⅲ，質問Ⅳに回答してもらった。質問Ⅲの内容を図 9 に，その結果を表 3 に示す。質問Ⅳの内容を図 11 に，その結果を表 4 に示す。

**質問Ⅲ**（任意）：質問Ⅰ，Ⅱで「1（役にたった），2（どちらかといえば，役に立った）」を選択した方に質問です。
どのような点で役に立ったと感じましたか。

図 9　質問Ⅲの内容

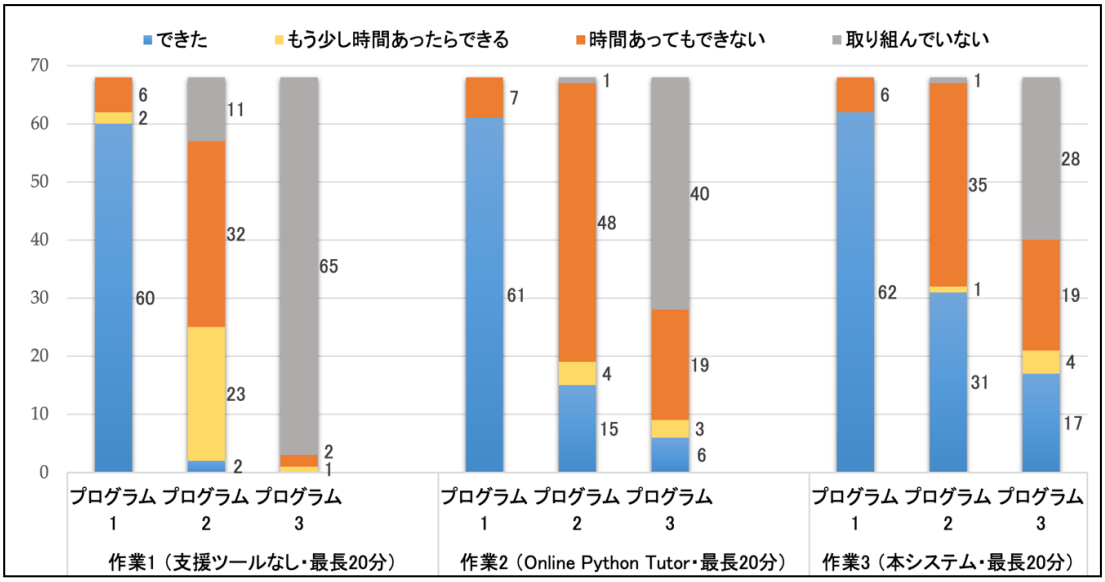


図 10　各デバッグ作業の報告結果

表 3　質問Ⅲに対する回答

| コメント | 票数 |
|---|---|
| (1)　変数の変化が分かりやすかった | 32 |
| (2)　条件が成立するか視覚的に確認できた | 15 |
| (3)　配列の入れ替えが分かりやすかった | 13 |

**質問Ⅳ**（任意）：質問Ⅰ，Ⅱで「3（どちらかといえば，役に立たなかった），4（役に立たなかった）」を選択した方に質問です。
どのような点で役に立たないと感じましたか。

図 11　質問Ⅳの内容

表 4　質問Ⅳに対する回答

| コメント | 票数 |
|---|---|
| (1)　使い方が分からない | 5 |
| (2)　見てもプログラムが理解できなかった | 5 |

表 3 の結果より，本システムに対して多くの肯定的なコメントを頂いた。表 4 の結果より，「使い方が分からない」などのコメントがあったが，実験当日の説明不足が原因だと考えられる。

## 6.　まとめ

本研究では，プログラミング初学者が自身で作成したプログラムの動作追跡の支援することを目的として，Online Python Tutor と伊藤の可視化システムを拡張し，以下の機能を実装した。

- ステップごとの実行時に変化する理由の可視化
- ブロック実行前後の比較の可視化

実験結果より，本システムがプログラムの動作追跡の支援に有効であることを確認した。また，本システムが提案した 2 つの機能についてどちらも 70%を越える割合で高い評価が得られた。

## 参考文献

(1)　中原進，紫合治：“オブジェクト指向プログラムの動作の可視化”，情報処理学会研究報告，ソフトウェア工学研究会報告，vol.2010，no.9，p.18（2010）

(2)　中村亮太，西村知博，松浦敏雄：“プログラミング入門教育用学習環境 PEN”，情報処理学会研究報告，p.6571（2017）

(3)　Sassi Bentrad and Djamel Meslati：“Visual Programming and Program Visualization − Toward an Ideal Visual Software Engineering System −”, ACEEE International，vol.1，no.3，pp.56-62 (2011)

(4)　Christine Alvarado，Briana Morrison，Barbara Ericson，Mark Guzdial，Brad Miller and David L. Ranum: “Performance and use evaluation of an electronic book for introductory Python Programming”, Technical Report GT-IC-12-02，Georgia Institute of Technology (2012)

(5)　伊藤 福晃，北 英彦：“プログラミング演習における動作確認を支援するためのプログラム動作の可視化”，2018PC カンファレンス論文集，pp.29-32（2018）

(6)　Philip J. Guo: “Python Tutor: Embeddable Web-Based Program Visualization for CS Education”, Proceeding of the 44th ACM Technical Symposium on Computer Science Education”, pp.579 - 584 (2013)