

内部データの視覚化によるプログラミング支援ツール

黒河優介† 三浦悠太† 藤枝崇史†

清水智公‡ 服部隆志† 萩野達也†

†慶応義塾大学環境情報学部所属

‡慶應義塾大学政策・メディア研究科所属

概要 プログラムの動的な可視化はプログラムの振る舞いを理解することに対して効果的だが、表示する変数を適切に選択することが難しかった。本研究では初学者向けの学習教材と、アルゴリズムアニメーションを組み合わせることにより、初学者の学習を助けるシステムを提案する。具体的には穴埋め問題を用意し、穴埋め部分に記述された変数のみをアニメーションとして表示する

1. はじめに

プログラミング教育は、プログラミングを通してコンピューターの仕組みを理解してもらうことが目的である。しかし、多くの学習者がプログラミングに挫折してしまい、コンピューターの仕組みを理解するところまで達しない。本研究では、プログラミングに挫折する学習者を減らすために、学習ツールの提供を行う。なお対象とする学習者は、プログラミングの学習を始めたばかりで、繰り返し、条件分岐、配列などを学習するレベルに限定し、ツールは自習用の教材として活用してもらう。

2. 問題点

ここでは、プログラミング学習における問題点と、それに対する一般的な解決方法、及びその問題点について述べる。

2-1. プログラミング学習における問題点

プログラムには、入力を受け、処理をし、結果を出力するという3つのプロセスがある。プログラミングを学習する上で重要なのは、入力と出力の間に行われている処理の内容を理解することである。しかし、処理の部分を視覚的に見ることは出来ない。そのため、ソースコードの内容と実際に行っている処理が結びつかず、ソースコードに書いてあることが理解出来なくなってしまう。

2-2. 一般的な解決方法

この問題に対する解決策は、変数の値の変化の様

Programming support tool by visualizing internal data.

Yusuke Kurokawa †, Yuta Miura †, Takafumi Fujieda †, Noritada Shimizu ‡, Takashi Hattori †, Tatsuya Hagino †.

† Faculty of Environment and Information Studies, Keio University

‡ Graduate School of Media and Governance, Keio University

子を逐一表示する等の可視化が一般的である。

プログラムの動作を可視化する手法は、以下の二つが代表的である。

アルゴリズム・アニメーション：アルゴリズムをわかりやすく表示することが目的で、プログラムにあらかじめ表示用の情報を付加しておく。

ビジュアル・デバッグ：プログラムの動作状態をわかりやすく表示することが目的で、制御フローや変数の値を自動的に表示する。

2-3. 解決方法の問題点

アルゴリズム・アニメーションは、予め用意されたアルゴリズムに対し、学習者にわかりやすいように可視化を行うことが出来るが、学習者が書いた任意のプログラムを可視化することは出来ない。

ビジュアル・デバッグは、学習者が書いた任意のプログラムの可視化を行うことは出来るが、全ての情報が均一の細粒度で表示されてしまい、学習者はどの部分が重要であるか判断できずに混乱してしまう。

3. 解決案

これまで、プログラミングの学習における問題点と一般的な解決アプローチを述べた。ここでは本ツールの問題解決のアプローチを述べる。

3-1 可視化の方法

本ツールでの可視化は、ビジュアル・デバッグのように、プログラムの実行中に変数の中身を逐一表示し、その遷移をアニメーションさせながら表示している(図1)。

3-2. 情報の部分表示

学習者が書いたプログラムの中で、どの部分が重要であるかを自動的に判断するのは困難である。そこで、重要な部分だけを記入する穴埋め問題にする(図2)。

重要な部分を穴埋めとして、それ以外の固定部分を簡素化して表示を行う。簡素化は、アルゴリズム・アニメーションと同様に、プログラムに予め情報を付加する。文、又は変数に以下の指定が出来る。

隠蔽：まったく表示しない。

省略：表示するが、アニメーションを行わない。

高速：アニメーションの速度を上げる。

このようにすることで、プログラミングを学習する上で必要な情報の可視化を的確に行う。

4. 実装について

これまで、本ツールの特徴を述べてきたが、ここでは本ツールの実装について述べる。

4-1 実装

Java のアプレットで実装している。言語のパースーは独自のもので、再帰的降下を用いて実装した。コンパイラがバッファに、命令コードを出力し、仮想マシンはその命令コードを元に、アニメーションしながら実行していく。

なお、実装した仮想マシンにはレジスタという概念がなく、計算は全てスタック上で行う。そして、計算用スタックを使うことで計算のアニメーションを見せている。

4-2 言語仕様

本ツールでは独自言語を採用しており、学習者はその独自の言語でプログラムを書くことになる。ただし、int型 での変数宣言、for 文、if-else 文、代入文、四則演算などの基本的な部分はC やJava と同一のものとしているので、学習者は本ツールでの学習後に、スムーズにC 言語やJava 言語に移行できる。

5. 終わりに

本研究はプログラミングの学習支援ツールの提供を目的としており、本稿ではその実現方法について述べた。

今後は、プログラミングを学ぶ上で難しいとされている、ポインタの概念を表現できるように本ツールを拡張したい。また、ポインタとの深い関わりのあるリスト構造、木構造、ハッシュなどの視覚化が出来るように、構造体のような複合型を表現できるように拡張を行いたい。

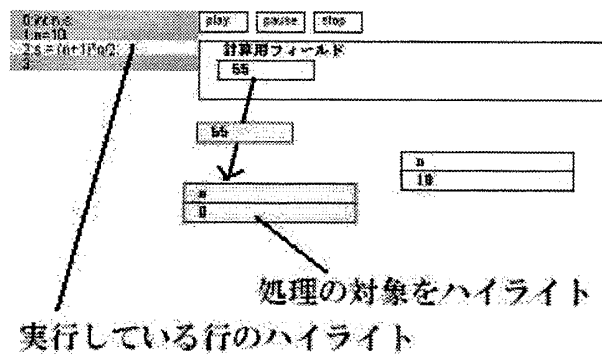


図1 情報の可視化

```
//1~nまで合計するプログラム
//sumに答えを
```

```
int i,n,sum;
sum=0;

for(i=0; [ ] : [ ] ){
    sum = [ ] ;
}
```

図2 穴埋め

参考文献

- [1] M. H. Brown:Zeus:A System for Algorithm Animation and Multi-View Editing
IEEE Workshop on Visual Languages, pp. 4-9, October 1991.
- [2] 石黒誉久, 井垣宏, 中村匡秀, 門田暁人, 松本健一:変数更新の回数と分散に基づくプログラムのメンタルシミュレーションコスト評価
TECHNICAL REPORT OF IEICE. SS2004-32
- [3] 喜多義弘, 川添貴議, 片山徹郎:初心者を対象としたJavaプログラム自動可視化ツールの実現に向けて.
TECHNICAL REPORT OF IEICE. SS2004-46
- [4] 藤崎友実, 掛下哲郎:構造化されたアルゴリズムを対象とした教育支援ツールPerseus.
TECHNICAL REPORT OF IEICE. SS2004-55