

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年4月3日現在

機関番号：32643

研究種目：基盤研究(C)

研究期間：2009～2011

課題番号：21500907

研究課題名（和文） Java アプレットのプログラミング初学者のための
ビジュアルトレーシングシステム

研究課題名（英文） Visual Tracing System for Java Applet Novice Programmers

研究代表者 荒井正之 (ARAI MASAYUKI)
帝京大学・理工学部・教授
研究者番号：70212602

研究成果の概要（和文）：Java アプレットは、GUI のプログラムが容易に作成できるため、プログラムの学習者の興味を引きやすく、プログラミング初学者の学習に有効と考えている。しかし、Java アプレットはイベントドリブン型のため、学習者がイベントの発生によって実行されるメソッドのトレースができない、また、メソッド内のソースコードのトレースができない、などの問題がある。本研究では、これらの問題点を解決するために、Java アプレットプログラムのトレース能力の修得を目的とした学習支援システムを開発した。システムの特徴として、学習者がボタンやチェックボックスなどを押して、イベントを発生させた場合、どのメソッドがどのような順序で実行されるかがトレースできること、メソッド内の制御構造がフローチャートで表示され、ソースコードと対応づけながら、トレースできることなどがあげられる。

研究成果の概要（英文）：We developed a learning system that helps novice programmers of Java applets to obtain the capability of tracing source programs. Java applets are user-friendly because they have a graphical user interface (GUI). For this reason, Java applets are often used to learn programming. However, for novice programmers using Java applets, it is difficult to trace the executing methods, which are driven by events, such as clicking on a GUI button, and the source code itself. In order to solve this problem, we introduce a learning support system.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	1,600,000	480,000	2,080,000
2010年度	900,000	270,000	1,170,000
2011年度	900,000	270,000	1,170,000
総計	3,400,000	1,020,000	4,420,000

研究分野：総合領域

科研費の分科・細目：科学教育・教育工学

キーワード：教授学習支援システム

1. はじめに

大学の理工学部系学部情報系学科(以下情報系学科と呼ぶ)の学生にとって、プログラミングの学習は必須である。多くの情報系学科では、1年生からプログラミング教育がはじまり、その後3年生くらいまで、積み重ね的な教育を行っている。このため、情報系学科において、初学者が初期にプログラミングで

つまずくと、その後の学習に大きく影響を与えてしまう可能性がある。プログラミング初学者がつまずく理由を、我々は次のように考えている。

- ・ アルゴリズムを熟考して決定した後で、プログラムを記述するという習慣がつかない時期に、プログラミング初期教育が始まる場合が多い。論理的思考定着前の

- プログラミング学習には障害が多い。
- 既存のプログラミング言語の多くはヨーロッパ言語を使う人間が設計しており、印欧語の影響を強く受けている。従って、プログラミング学習においても、外国語の学習同様に、多くのプログラムに触れ、多くの優れたソースコードを読む、などが有効であると考えられる。しかし、このような学習環境が現状では整備されていない。
 - プログラミング学習に限ったことではないが、学習者個人に合った方法・手順で学習を進めるべきであるが、現状では、教科書などに沿った画一的学習が一般的である。

我々は、学習者が次のような学習過程を経て、プログラミングを修得していると考えている。

- 《ステップ 1》プログラムの全体構造の理解と、入出力文・変数・定数・演算子・式・変数宣言・変数の型などの理解。
- 《ステップ 2》接続、条件判断処理、反復処理などのアルゴリズムの基本構成要素の理解と、それらを使ったソースプログラムを 1 行 1 行追う(トレースする)能力の修得。
- 《ステップ 3》接続・判断・繰り返しなどのアルゴリズムの基本構成要素を用いて、簡単なプログラムを作成する能力の修得。
- 《ステップ 4》各種のアルゴリズムやデータ構造をプログラムとして記述する能力の修得。

2. システムの概要

2.1 システム構成

システム構成を図 1 に示す。システムは、教師用クライアント(Client for teachers), 学習者用クライアント(Client for learners), サーバ(Server)から構成される。サーバには学習教材生成サブシステム(Learning materials generating subsystem), 学生用クライアント(Client for learners)にはプログラム可視化サブシステム(Programming visualization subsystem)をインストールしなければならない。

(1)Teacher's interface

教員が学生や問題を管理するための、教員用クライアントに表示されるインターフェースである。

(2)Source program's template generating engine

教員がシステムにソースプログラムを入力するとき使用する。類似問題を作成することができる。

(3)Source program setting engine

ソースコードを入力し教材データベースに保存するために用いる。

(4)Commands for generating visualization data inserting engine

スケジュールデータを生成するための命令を自動的に挿入するために用いる。スケジュールデータについては、3章で詳述する。

(5)Database

作成された教材データや教材テンプレート、ソースコードを保存するために用いる。

(6)Visualization data generating engine

学習者用インターフェースの各画面の連携に必要なデータの生成に用いる。

(7)Executed methods display engine

Executed methods display engine で生成したデータを元に、メソッドのメソッド発生順序の可視化に用いる。

(8)Applet's pictures display engine

プログラムの実行により変化するアプレットを可視化するために用いる。

(9)Control structure display engine

プログラムの制御構造、実行行等を可視化するために用いる。

(10)Learner's interface

学生が実際にアプレットを動作させ、プログラムのトレースを行うインターフェースである。

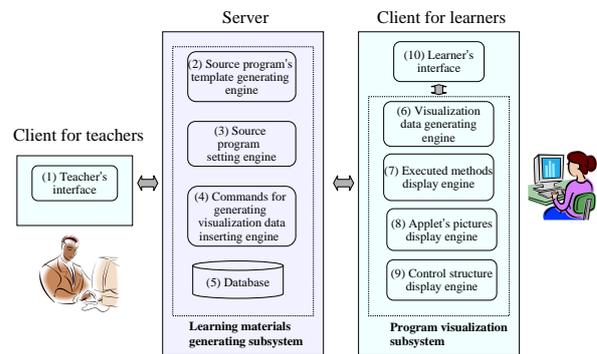


図 1 システム構成図

2.2 学習者インターフェース

本節では、図 1(10)に示した学習者インターフェース(Learner's interface)について、図 2 を例に説明する。

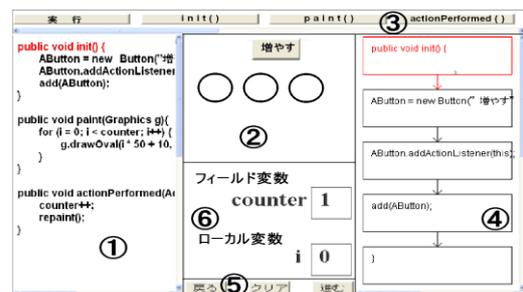


図 2 学習者用インターフェース

- ①ソースプログラム表示部：学習者がトレースを行うソースプログラムが表示される。
- ②アプレット表示部：①で示したソースプログラムの実行結果であるアプレットが表示される。
- ③実行メソッド表示部：②のアプレットで「ボタンを押す」等のイベントが発生すると、そのイベントに対応したメソッドがボタンとして順番に表示される。
- ④フローチャート表示部：③にあるメソッドを表したボタンを押すことで、そのメソッドの処理内容と手順がフローチャートで表示される。
- ⑤トレース用ボタン表示部：進む、戻る、クリアの 3 つのボタンが設けられており、進む・戻るボタンを押すと①④の該当部分が赤くなり、プログラムを 1 行単位でトレースすることができる。
- ⑥変数値表示部：選択中のメソッド内のローカル変数とフィールド変数を表示しており、⑤のボタンを押すことで、①④と共に変化していく。

3. プログラム可視化のためのデータ構造

図 1(6)に示した学習教材サブシステム(Program visualization subsystem)の可視化データ生成エンジン(Visualization data generating engine)では、スケジュールデータと実行メソッドデータを自動的に生成する。これらのデータを用いることにより、2.2 で述べた動作が可能となる。本章では、これらのデータの生成方法とデータ構造について述べる。

図 3 の(1)にスケジュールデータの生成方法を示す。図 2 の①に示したソースプログラムの 1 行 1 行に対して、システムはスケジュールデータを得るための命令を自動的に挿入する。スケジュールデータはクラスとして実装されており、図 3 の(2)の” [0] m=1,1,i=0,counter=1,img=null” は、” m=1” が実行されたメソッドの順序、次の” 1” がメソッド内のソースコードの行番号、i=0 がメソッドで使用されるローカル変数名とその値、” counter” がフィールド変数名とその値、” img” が実行中のアプレットの画像を表している。

図 3 の(3)は、実行されたメソッドを表すボタンの情報であり、「実行メソッドデータ」と呼ぶ。スケジュールデータ内のメソッド順序と実行メソッドデータ内のボタン配列はリンクしており、例えば実行メソッドデータの配列番号[1]の「init0 スケジュールデータ内からメソッドの順序が 1 であるデータと呼び出す」は、スケジュールデータのメソッド順序が” m=1” のデータを参照する。図 2 の②に表示されるアプレットのボタンなどを押して、イベントが発生すると生成される。

(1)スケジュールデータの生成方法

```
public void init(){
    incrementButton = new Button("増やす");
    incrementButton.addActionListener(this);
    add(incrementButton);
}
```

学習者に表示する
ソースプログラム



```
public void init() {
    data.add(new SData(m, 1, 0, counter, img));
    incrementButton = new Button("増やす");
    data.add(new SData(m, 2, 0, counter, img));
    incrementButton.addActionListener(this);
    data.add(new SData(m, 3, 0, counter, img));
    add(incrementButton);
    data.add(new SData(m, 4, 0, counter, img));
    data.add(new SData(m, 5, 0, counter, img));
}
```

スケジュールデータを生成するための命令をシステムが挿入

(2)スケジュールデータの例

配列	メソッド順序	行	ローカル変数	フィールド変数	画像識別子
[0]	m=1,	1,	i=0,	counter=1,	img=null
[1]	m=1,	2,	i=0,	counter=1,	img=null
[2]	m=1,	3,	i=0,	counter=1,	img=null
[3]	m=1,	4,	i=0,	counter=1,	img=null
[4]	m=1,	5,	i=0,	counter=1,	img=null
[5]	m=2,	101,	i=0,	counter=1,	img=null

(3)実行メソッドデータ

ボタン配列	実行メソッド名	ボタンを視した時の処理
[1]	init()	スケジュールデータ内からメソッド順序が1であるデータ呼び出す
[2]	paint()	スケジュールデータ内からメソッド順序が2であるデータ呼び出す
[3]	actionPerformed()	スケジュールデータ内からメソッド順序が3であるデータ呼び出す

図 3 スケジュールデータと実行メソッドデータの例

4. イベント発生時の実行メソッドの可視化
本章では、図 2 の③に示したイベント発生時の実行メソッドを可視化する方法について述べる。

図 2 の③に示したメソッドの発生する順序の可視化を行うためには、実行メソッドデータを用いる。例えば、図 2 の③のような場合、実行メソッドデータに格納されたデータの順に init0, paint0, actionPerformed ボタンが表示される。表示された各ボタンを押すことによって、図 2 の①②④⑥の表示が変わる。例えば、init0 ボタンを押した場合は、①のソースプログラムは init0 メソッドの部分を赤く表示し、④には init0 メソッドのフローチャートを表示する。

5. 実行中のアプレットの可視化

本章では、図 2 の②に示したアプレット画像の可視化方法について述べる。

プログラムの実行によりアプレットも変化することがある。本節では、プログラムの実行により変化するアプレットを可視化することにより、トレースする方法について述べる。図 3(1)に示したように、システムはスケジュールデータを取得するための命令を自動的に 1 行ごとに挿入する。アプレットが変化する命令の直後には、図 4 に示すように

さらに画像を得るための命令を挿入する。図4は、drawOval と drawString によって描かれたアプレットの画像を取得して、img という変数に代入した後に、これらをスケジュールデータに格納する例を表している。このように、アプレットの画面が変化するたびに画像データが格納される。これらを用いることにより、アプレットもトレースできるようにした。

```

Image img = null;
paint(Graphics g){
  g = getGrubedGraphics();
  data.add(new SData(m, 101, 0, counter, img ));
  g.drawOval(.....);
  DispCapture();
  img = getImage();
  data.add(new SData(m, 102, 0, counter, img ));
  g.drawString(.....);
  DispCapture();
  img = getImage();
  data.add(new SData(m, 2, 103, counter, img ));
}

```

gの内容を内部処理するために画像変換用の命令を挿入

画像取得用の命令を挿入

画像取得用の命令を挿入

図4 アプレット画像の取得方法

6. メソッド内の変数の値と処理手順の可視化

本章では図2の①④⑥に示したプログラムの実行行、制御構造、変数の値の可視化について述べる。

6.1 変数の値の表示

スケジュールデータを用いて、図2の⑥に示した変数の値を表示する。例えば、図2の(2)の“m=1,i=0, couter=1,img=null”というデータを用いた場合、ローカル変数 i の値は0、フィールド変数を表す counter は1と表示される。

6.2 フローチャートの表示

図2の④に示したフローチャートの生成方法について説明する。システムは、図2の①に示したソースプログラムを構文解析し、フローチャート図記号および各命令の実行順番に関する情報を得る。これらの情報をもとにフローチャートを描く。例えば、if文の場合は条件判断を示す記号、for文やwhile文の場合は、繰り返しを示す記号を表示する。

6.3 実行中のソースコードの表示

スケジュールデータを用いて、図2の①⑥に示した実行中の命令と命令に対応するフローチャート記号を赤く表示する。例えば、図3(2)に示した“m=1,i=0,counter=1, img=null”というスケジュールデータの場合、2番目のデータ“1”が実行行を表しており、ソースコードの1行目と、それに対応したフローチャート記号を赤く表示する。

7. おわりに

本研究では、Java アプレットプログラミングの初学者を支援するためのシステムを提案した。本システムは、実行されるメソッドの可視化機能、プログラムの実行により表

示されるアプレットの可視化機能、メソッド内の制御構造の可視化機能などを持つ。

今後の課題としては、実授業におけるシステムの評価などが挙げられる。

8. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計6件)

(1) 荒井正之:プログラミング初学者のトレース能力の修得を目的とした学習支援システム, 日本 E-Learning 学会論文誌, Vol.10, pp.54-62, (2010).

(2) Arai,M.; Visual Tracing System for Novice Programmers of Java Applets using Aspect-oriented Programming, Proc. of the International Conference on Computational Intelligence and Software Engineering(CiSE2009), Wuhan China (2009).

(3) Ishizaki,Y., Shimokawa,R., Takahashi,K., Takahashi,Y., Nakamura,S., Fukuwara,M., Miyoshi,K., Yamamoto,H., Mizutani,K. and Arai,M.: Design of a Learning Support System to Obtain the Capability of Tracing for Novice Programmers of Java Applets, Proc. of E-Learn2008 Las Vegas, pp.3745-3751, (2008).

(4) Arai,M.: A Method of Evaluating Programs for Novice Programmer' Learning Support System, Computers and Advanced Technology in Education, pp.400-403(2007).

(5) Arai,M. and Yamazaki,T.:Design of a Learning Support System to Aid Novice Programmers in Obtaining the Capability of Tracing, The 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006), The Netherlands, pp.396 - 397, (2006).

(6) Arai,M. and Yamazaki,T.: Design of a Tool that Sets Problems of Tracing a Source Program for Novice Programmers, Proc. of The 3rd International Conference on Multimedia and ICT's in Education, Vol.2, pp.702-705(2005).

〔学会発表〕(計4件)

(1) 南城恵里佳, 荒井正之: プログラミングスタイルの基礎を身につけさせるツールの開発, 情報処理学会第73回全国大会, 2K-6, (2011)

(2) 水谷晃三, 高井久美子・渡辺博芳・荒井正之: チャットツールを活用したブレインストーミング法演習に関する一考察, 教育シス

テム情報学会第35回全国大会, 26-F2, (2010)
(3) 平塚政幸, 水谷晃三, 荒井正之: 携帯電話を利用した即時性のあるレスポンスアナライザ: 2010 春 JSiSE 学生研究発表会, pp39-40(2010.3)
(4) 大谷育弘, 小塩貴裕, 島崎智也, 中泉純, 赤羽根隆広, 荒井正之: Java アプレットのプログラミング初学者のためのトレース能力の修得を目的とした学習支援システムの開発, 情報処理学会第71回全国大会, 5L-1, (2009)

[図書] (計0件)

[産業財産権]

○出願状況 (計0件)

名称:
発明者:
権利者:
種類:
番号:
出願年月日:
国内外の別:

○取得状況 (計1件)

名称: プログラミング学習支援システムおよびプログラミング学習支援方法
発明者: 荒井正之
権利者: 学校法人帝京大学
種類: 特許
番号: 特許第4808041号
取得年月日: 平成23年8月26日
国内外の別: 国内

[その他]

ホームページ等

<https://www.ics.teikyo-u.ac.jp/~arai>

6. 研究組織

(1) 研究代表者

荒井正之 (ARAI MASAYUKI)

帝京大学・理工学部・教授

研究者番号: 70212602

(2) 研究分担者

(3) 連携研究者

以上