

K-005

PAD を利用したパズル型プログラミング学習システム

Puzzle type programming learning system using PAD

猪俣 順平[†]
Junpei Inomata

寺田 実[†]
Minoru Terada

丸山 一貴[‡]
Kazutaka Maruyama

1 背景

従来のプログラミング教育手法では、プログラム言語の文法理解に重点が置かれているため、「プログラミングは難しい」と感じ、プログラミングへの興味を失ってしまい、学習者のモチベーションの低下を招く可能性がある。

また、プログラミングの習得には自分でプログラムを作ることが重要である。しかし従来の教育手法では、作成したプログラムが正しい動作・出力を得られているのか学習者自身で判定することが難しいという問題点がある。

これらの問題を解決するため、本研究では、プログラム要素を直接操作し、パズル形式で学習を行う手法に着目した。パズルの要素には、Problem Analysis Diagram(以下 PAD とする) という手法を利用した。

2 Problem Analysis Diagram(PAD)[1]

PAD とは、計算機プログラムを作成するときにアルゴリズムやプログラムの処理の構造を図で表す手法である(図 1)。

PAD は、構造化プログラミングというプログラムの個々の処理を小さな単位に分解し、階層的な構造にして表現する方式に基づいており、しばしばプログラミング教育の初歩として学習に用いられている。

3 目的と着目点

本研究では、PAD をパズル形式で組み立てることにより、アルゴリズムを作成できる学習システムを提案する。

また、教員が新たな問題を作成し、投稿するためのインタフェースを実装する。このインタフェースにより学習者が一方的に学習するのではなく、このシステムを利用し、教員が学習者にプログラミング教育を行う学習環境として機能するようにする。さらに、作成したアルゴリズムを実行できる機能により、学習者に分岐やループといったプログラム要素の動作を視覚的に理解支援するインタフェースを備えたシステムを実装する。

それにより、従来のプログラミング学習法に比べ、より手軽にプログラミングの学習を行うことのできる学習環境を作成し、プログラミング学習者の学習支援を目的とする。

3.1 PAD の利用

アルゴリズムを図で可視化することにより、アルゴリズムの全体構造を視覚的に捉えることができ把握しやすくなる。

アルゴリズム可視化の主な手法として、フローチャー

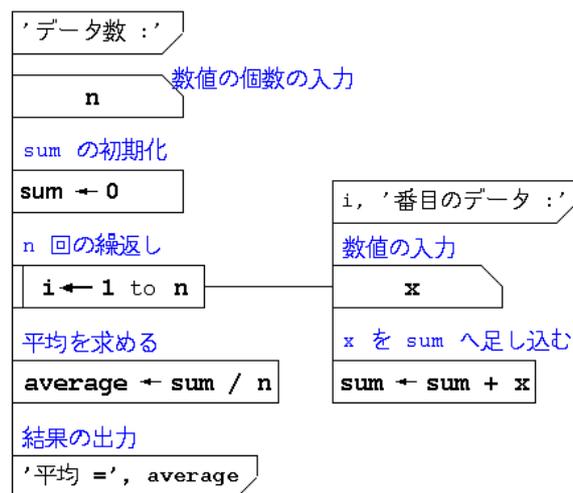


図 1 PAD 表示例：n 個の入力の平均値を求めるアルゴリズムの PAD を表している

ト、PAD、NS チャート (Nassi and Schneiderman chart) などがある。

PAD と NS チャートは構造化プログラミングとの対応関係が明確であるという特徴がある。可視化によく使用されるフローチャートは、制御の流れが 4 方向 (上下左右) 自由に描けてしまうため、制御構造の把握には適していない。また、NS チャートと比較して PAD はドラッグ&ドロップで組み立てやすい構造になっているため、本研究では可視化手法として PAD を選択した。

現在最も広く普及しているソフトウェアの図法として UML(Unified Modeling Language) がある。UML はシステムをモデル化する際には非常に有効な手段であるが、プログラミング教育で使用される小規模なプログラムのモデル化には適していない。それに対し、PAD は分岐やループといったプログラム構造をミクロな視点から可視化することが出来るため、プログラミング教育に有効である。

4 関連研究

4.1 石田らの研究 [2]

石田らは、C プログラミングでのアルゴリズム作成の学習を行う際に、PAD を利用することで学習者の理解支援を行うツールを実装した。このシステムでは、学習者はアルゴリズムを PAD 形式で対話的に作成し、作成した PAD を C 言語形式のソース

[†] 電気通信大学, The University of Electro-Communications

[‡] 東京大学 情報基盤センター

コードに変換を行う。学習者は、作成した PAD が C 言語でどのように表現されるのか観察しながら、C プログラミングでのアルゴリズム作成の学習を進める。

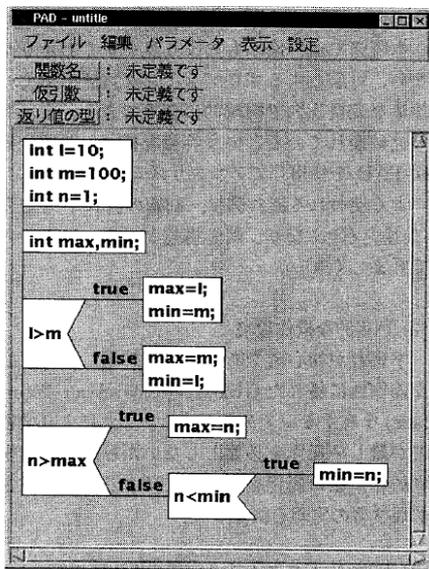


図2 石田らの研究ツールの動作画面

4.1.1 本研究との関連性

本研究と同様に PAD を利用しアルゴリズムの学習支援を行っている。石田らの研究では、学習者が新しいアルゴリズムを作成するため解答というものが存在しない。そのため、作成したアルゴリズムが正しい動作を行うのか学習者自身で確かめなければならない。一方、本研究では解答の用意された問題に対し学習を行うため、作成したアルゴリズムが正しい動作を行うのかシステムが判断することが出来る。

4.2 Parson's Programming Puzzles[3]

Parson's Programming Puzzles はパズル形式でプログラミングの基礎となる構文の学習を支援するシステムである。このシステムでは、事前に用意されたサンプルコードの順番をランダムに配置し実際には使わないダミーのコードを加えて表示する。学習者は必要なコードを取捨選択し、ドラッグ&ドロップによりコードを並び替えサンプルコードを完成させることにより学習を行う。

4.2.1 本研究との関連性

本研究では、Parson's Programming Puzzles と同様にプログラミングの学習をパズル形式で行っている。Parson's Programming Puzzles ではパズルの要素がテキストで記述されているため、そこに表現されているアルゴリズムの全体像を視覚的に把握するのは難しい。一方、本研究ではパズルの要素に PAD を利用していることにより、分岐やループといった構文を階層的に表示できる。そのため全体的な構造を把握しやすくなっている。

5 提案システム

3章で述べた学習を行うための問題解答インタフェースとプログラミング教育の教員が新たな問題を作成し登録を行う問題

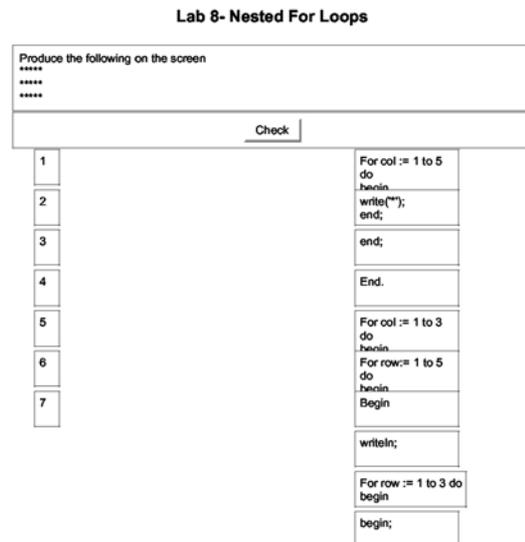


図3 Parson's Programming Puzzles の動作画面:画面右に表示されているブロックを取捨選択しドラッグ&ドロップで画面左の数字欄にコードを並べ変えていく

作成インタフェースの実装を行った。

5.1 問題解答インタフェース

システムを利用する際に、

- アルゴリズム問題文、
- 問題に対する解答の PAD、

をプログラミング教育の教員が事前に用意しておく。

システムは、初めに解答の PAD を部品 (PAD の一命令) に分解する。次に学習者に対し、アルゴリズムの問題文と分解された部品の集合を提示する。

学習者は与えられた問題に対して、その解答の PAD となるように分解された部品を組み立てていく。これにより、学習者は部品をマウス操作でパズルのように組み立てるだけで、プログラミングの学習を行うことが出来る。

5.1.1 ダミーブロック

このシステムでは、問題に必要なブロックだけでなく実際には使用しないダミーのブロックを追加し表示できる機能がある。この機能により、学習者は提示されたブロックを単純に並び替えるだけでは正解を出せなくなり、必要なブロックを取捨選択する必要がある。それにより、学習者はより深く問題の分析を行う必要があるため、学習効果の向上が期待できる。

5.1.2 PAD 実行機能

本システムでは、PAD を組み立ててアルゴリズムの学習を行えるだけでなく、構築途中の PAD を実行し、PAD がどのような振る舞いをするのか学習者が確かめることが出来る機能を持つ。

この機能を実行すると、現在構築中の PAD を実行し、PAD の最初から最後のステップまでの、変数の変化を 1 ステップずつ見ていくことが出来る。

5.1.3 画面構成

図4に提案システムの動作画面を示す。

システムの画面は、以下の4つで構成されている。

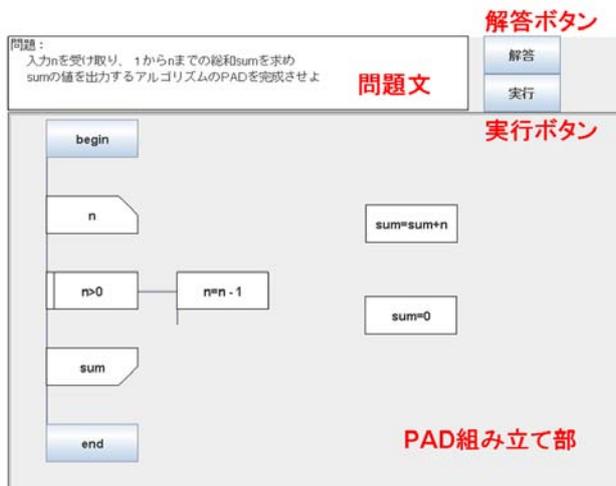


図4 提案システムの動作画面

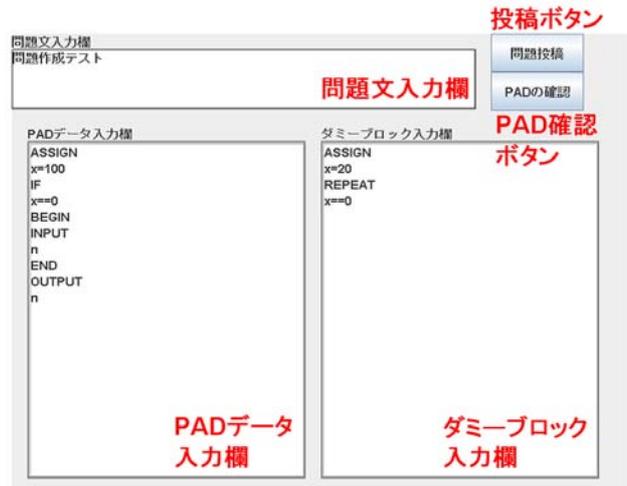


図5 問題作成インターフェースの画面構成

問題文

PAD 組み立て部

学習者が PAD の組み立てを行うフィールドを表示している。PAD のブロックをドラッグ&ドロップすることにより、PAD の任意の位置に挿入したり、組み立てた PAD からブロックを除去できる。

回答ボタン

作成した PAD と解答の PAD との比較を行うためのボタンを表示している。不正解であった場合、最初に間違っている PAD のブロックを赤枠で囲み示す機能を持つ。

PAD 実行ボタン

現在構築中の PAD に対して実行を行うボタン。このボタンを押すことにより PAD 実行機能画面に移行する。

5.2 問題作成インターフェース

本システムでは、教員が新しい問題を作り、登録する機能を持つ。アルゴリズム問題文と問題の解答となる PAD を作成し、その2つをセットとして新しく登録する。

図5にこの機能の画面構成を示す。

問題作成インターフェースの画面は、問題文入力欄、PAD データ入力欄、ダミーブロック入力欄、投稿ボタン、PAD 確認ボタンの5つで構成されている。PAD 確認ボタンは PAD データ入力欄で作成した PAD を GUI 上で確認するときに使用する。

5.2.1 PADstring 言語

問題作成インターフェースでは、PAD データの入力を本研究で作成した PADstring 言語と呼ばれる記法で行う。PADstring 言語は PAD を文字列で表現したもので、各 PAD 要素の順序や階層を記述する。Pascal や C 言語に似た記法になっており、それらの言語を利用したことのある人間であれば簡単に記述することが出来る。

PAD データの入力に PADstring 言語を用いた理由は、問題作成を行う教員はプログラミング技術が高いため、GUI 上で PAD を一つずつ作成するより文字列で入力したほうが速いと考えたためである。また教員による問題の保存・管理も文字列であるため容易にできる。

6 実装

6.1 システム構成

本システムは、クライアント側を Java Applet、サーバ側を Perl で作成された CGI で構成した。クライアント側では、ユーザインタフェース表示とサーバとの通信を行う。またサーバ側では、問題作成インターフェースで作成された問題の登録を行う。図6にシステム構成図を示す。

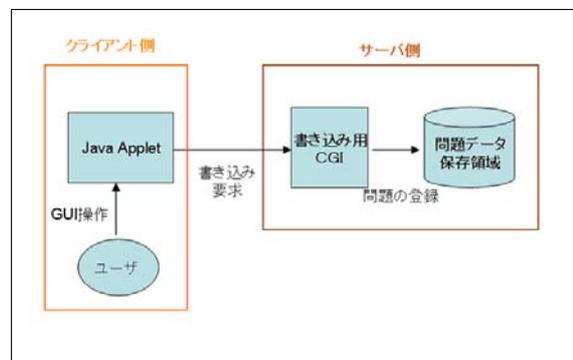


図6 システムの構成図

6.2 PAD 実行機能

PAD は構文木の一つであるため、PAD 実行時のブロックの実行順は容易に調べることが出来る。

そして、実行順の1ステップずつそれぞれのブロックが持つ代入式や、条件式が記述された文字列を、実行できる形式の計算式に変換するため解析を行い、式の値を評価して1ステップごとの変数の値を保存しておく。それにより、PAD 実行による1ステップずつの変数の変化を調べることが出来る。計算式の解析・評価にはオープンソースの四則演算実行クラス*1を使用した。

*1 四則演算実行クラス (JDK1.4 版):

<http://www.ne.jp/asahi/hishidama/home/tech/soft/java/eval.html>

7 評価

7.1 評価方法

本学情報・通信工学科の講義「情報工学工房」の受講者1年生5名の被験者を対象とした。被験者は全員PADを知らなかったため、実験前にPADの簡単な説明を行った。次に事前に筆者が用意したタスク1からタスク3の3つのアルゴリズム問題を提案システムを使って回答させ、学習内容の理解度と使用感の評価と自由記述のアンケートを行った。

7.2 評価結果

評価の設問を表1, 結果を表2に示す。

表1 評価項目

項目	内容
Q1	従来のプログラミング学習法(実際にJavaやC言語を記述・コンパイル・実行して学習)に比べ、気軽に学習を行うことが出来たか
Q2	本システムを利用することでPADの使い方が理解出来たか

表2 評価結果

項目		1	2	3	4	5	
Q1	出来ない	0	1	0	2	2	出来た
Q2	理解出来ない	0	0	0	0	5	理解出来た

自由記述では以下のような意見が出された。

1. パズル感覚でアルゴリズムの学習が出来て楽しい。
2. 実行機能によりプログラムがどのように動いているの分かりやすかった。
3. 上下に移動するためのスクロール操作が思っていたの逆方向。
4. ブロックをPADに追加しても見た目に変化がないので、PADに追加できたの分かりにくい。

7.3 評価に対する考察

以下では、5名に対して行った5段階評価と自由記述アンケートの結果について考察する。表2の設問1より、従来のプログラミング学習法と比較して気軽に学習が出来ることがわかった。自由記述1からも従来の学習法に比べ学習者のモチベーションを向上させることが出来ていた。

また設問2より、全員PADの使用法を知らなかったにもかかわらずPAD使用法の理解の評価が非常に高かった。理由としては自由記述2のように、PADを実行し視覚的にプログラムがどのように動作するのか把握しやすいためであると考えられる。このことから、本システムはPAD使用法の導入にも有効であることがわかった。PADを学習することにより、構造化プログラミング手法の基礎を学ぶことが出来ることから本システムはプログラミング学習支援システムとして有効であることがわかった。

一方、自由記述3,4は本システムのユーザインタフェースが必ずしも使い易いものではないことを示している。より直感的で

ユーザビリティの高いインタフェースに改良する必要がある。

8 結論

本研究では、与えられた問題文からPADのブロックを組み合わせて問題を解くアルゴリズムのPADを作成するというプログラミング学習支援システムを提案した。また、学習者が回答を行うインタフェースと、プログラミング教育の指導者が新たな問題を作るインタフェースを実装した。

その結果、パズル感覚で気軽にプログラミングの学習を行うことが出来るため、プログラミング学習者のモチベーションを向上させることが出来た。

9 今後の課題

9.1 配列・関数の実装

現在のシステムでは、配列や関数といったプログラム要素の使用に対応していないため、今後実装を行っていく。配列や関数を利用できるよう実装することで、再帰処理やソートアルゴリズムなど、より複雑なアルゴリズムを作成でき、プログラミング学習者がより多くの問題に挑戦することが出来るようになる。

9.2 PADブロックの順不同問題への対応

現在のシステムでは、学習者の組み立てたPADが解答のPADと等しいか、ブロックの順序を確かめることで確認している。しかし、解答のPADと同じ順序でなく、ブロックの順序を入れ替えても問題文を解くアルゴリズムが作成できてしまうことがあり、この場合正しいアルゴリズムにもかかわらずシステムでは不正解と判断されてしまう。

この問題の対応策として次の方法を現在考えている。

現在はブロックの順序を確かめて比較を行っているが、PADを実行することにより得られる出力から正しいアルゴリズムであるか判断する。この方法では、テストデータを複数あらかじめ用意する必要が発生する。また、出力だけテストデータ通りにする不正なプログラムを作成される可能性がある。しかし、解答に使用するブロック数などを指定することにより制限を加えれば不正なプログラムを回避することが出来る。

参考文献

- [1] Y.Futamara, T.Kawai, H.Horikoshi, M.Tsutsumi. "Development of computer programs by problem analysis Diagram(PAD)". ICSE '81 Proceedings of the 5th international conference on Software engineering, pp.325-332, 1981.
- [2] 石田真樹, 桑田正行. "Cプログラミングの学習支援に関する研究: PADエディタを用いたアルゴリズム学習支援システムの構築". 情報処理学会研究会報告, 2000-CE-58, pp.41-48, 2000.
- [3] Dale Parson, Patricia Haden. "Parson's programming puzzles: a fun and effective learning tool for first programming courses". Proceeding ACE '06 Proceedings of the 8th Australian conference on Computing education, Vol.52, 2006.