

PAD エディタを利用した C プログラミング学習支援システム構築に関する研究

永原 功策 桑田 正行

電気通信大学

E-mail: {nagahara, kuwada}@axion-gw.ee.uec.ac.jp

<あらまし> プログラミングの学習においては、問題解決のためのアルゴリズムの作成技術を習得することが重要である。そこで、アルゴリズムの表記に PAD(Problem Analysis Diagram) を用いて、アルゴリズムを対話的に作成するツール PAD エディタが Java で構築されてきた。本研究では、学習者が自らの意思で学習を進めるという形式での学習支援システムを目指し、PAD エディタを利用したシステムの構築を行っている。本論文では、今まで開発されてきたシステムの問題点を分析し、その問題点を解決するとともに、学習支援のために新たに実装した機能について述べる。

<キーワード> C プログラミング, アルゴリズム学習, PAD, Java, Servlet

C Programming Learning Support System with PAD Editor

Kousaku Nagahara Masayuki Kuwada

The University of Electro-Communications

E-mail: {nagahara, kuwada}@axion-gw.ee.uec.ac.jp

Abstract: Learning the ability of making an algorithm for problem solving is important in the learning of the programming. So we have made the PADEditor that is the tool for making algorithm interactively with PAD(Problem Analysis Diagram) . Our purpose is that the construction of the leaning support system with PADEditor for the learners progress study themselves with a subjective intention.

In this paper we describe about the problem of the system that has been developed until now and the function that implemented it newly for learning support, beside the problem is solved.

Keyword: C programming, algorithm learning, PAD, Java, Servlet

1 はじめに

インターネットに代表されるコンピュータネットワーク技術の普及と、コンピュータの低価格化により、コンピュータでインターネットを利用することがごく当たり前のこととなるほど、我々の生活と深い関わりを持つようになってきた。これに伴って、WWW(World Wide Web)を利用した、学習機会も身近になり、その頻度も増加している。また、社会における様々なシステムにコンピュータが導入され、それに伴い、これらのシステムを制御するソフトウェア技術の重要性が増大し、プログラミングの学習が幅広く行われるようになってきている。

このような背景から、WWW 上での利用を視野に入れ、アルゴリズムの表記に PAD(Problem Analysis Diagram) を用いて、アルゴリズムを対話的に作成する、プログラミング学習支援ツール PAD エディタが Java を用いて作成されてきた^[1]。このツールは、ウィンドウ上で PAD を用いて、アルゴリズムを記述し、その実行結果や変換したソースコードとの関連性から、プログラミングにおけるアルゴリズムの構造の理解を視覚的に進めることができるように作成されている。しかし、PAD 箱内に記述できる文字に制約がある、アルゴリズムの検証に十分な実行結果表示機能が備わっていない、学習データの保存や再利用が行えないというような問題がある。

また、これまでに、流れ図や構造化チャートなどの図を用いたプログラミング学習ツールやそのツールを利用した研究事例は、いくつか存在する^[2, 3]。しかし、本研究では、各々の学習者が自分自身のペースで学習を進め、問題解決は自分自身の力、あるいは同じ目的を持って学習を行っている他の学習者との協調によって行うことを目的としており、以下の二つの点が他のシステムとは異なる。

- 学習者が自分の都合(時間・場所等)に合わせて自学自習を進められる環境提供を提供するため WWW 上での利用を考慮している点

- 対象言語(C言語)を設定し、そのソースコードへ詳細化する過程を学習することを目的としている点

本研究では、前述したように、学習者が自らの意思で学習を進めるという形式での学習支援システムを目指し、PAD エディタを利用したシステムの構築を行っている。本論文では、今まで開発されてきたシステム^[1]の問題点を分析し、その問題点を解決するとともに、学習支援のために新たに追加した機能について述べる。

2 PAD エディタの概要

ここでは、これまでに作成されてきた、PAD エディタの主な特徴について述べる。

2.1 PAD 図と C 言語形式のソースコード間の相互変換機能

この機能は、学習者が作成した PAD 図を C 言語の形式のソースコードに変換あるいは、ソースコードから PAD 図に変換する機能で、PAD 図とソースコードの対比からアルゴリズムの構造理解を支援することができる。

2.2 実行順表示機能

この機能は、アルゴリズムの実行順序を PAD 図の色を変えて表示する機能である。この機能により、どのような順序でアルゴリズムで実行されるかを確認することができる。

2.3 書式判定機能

PAD 図各々の処理箱内のテキストが C 言語形式の書式に適合しているか判別し、適合していない箱の色を変え、表示する機能である。これにより、どの箱の処理が擬似コードであるかが一目瞭然となる。また、変換前に書式にエラーがないかを確認することが可能である。

3 これまでのシステムの問題点

2章で述べてきたPADエディタの機能を用いることで、任意のアルゴリズムをPADという形式で視覚的に作成し、そこで作成したPADとソースコードとのアルゴリズム記述時における対応関係を知ることが可能である。このことにより、これまでのシステムにおいても、対話的にアルゴリズムの学習を進めることのできる環境の提供という形で学習支援を行うことができる。

しかし、課題となっている部分もある。それらの点を以下に問題点として示す。

3.1 PAD図作成時の問題

PAD図作成時、関数定義用の処理箱がないため、関数定義が行えないという問題点がある。そのため、C言語ライクの擬似コードだけを使用する場合は良いが、擬似コードからさらに実際の開発環境でも動作させることができる、マクロ等を含むC言語の記述形式に変換することを想定した場合、不十分となる。

3.2 文字の制約に関する問題

これまでのシステムでは、扱える文字に制約があり、PAD図とソースコード間の相互変換パーサで定義されていない記号や自然言語等をうまく変換できなかったり、PADの箱内に記述ができないという問題点がある。

3.3 アルゴリズム実行結果の表示に関する問題

2.3節で述べたように、エラー表示に関しては、個々の処理箱内に記述した文字列が、擬似コードかどうかを色を変えることで、どの箱の処理が擬似コード、あるいはエラーなのか、すぐに見つけることができる。しかし、処理箱内に記述がエラーの場合その原因が表示されないため、エラー情報の提示としては不十分である。また、アルゴリズム検証機能に関して、ア

ルゴリズム実行結果をPAD図上で視覚的に確認すること可能であるが、実際にC言語形式のソースコードに変換した時、数値がどのように変化していくかを知ることができないため、実行結果の表示機能としては不十分である。

3.4 データの保存と参照

Javaアプリケーションとしても実行が可能であるが、元々JavaアプレットとしてWWW上から提供することを目的として設計されているため、作成したPADをローカルファイルに保存することができない。したがって、学習者が作成したPADはその場限りで破棄されてしまうという問題点がある。学習者にとって、以前に自分が作成したアルゴリズムを現在のもものと比較するのは有益なことと思われるので、何らかの保存手段を用意しておく必要がある。

4 追加する機能の仕様

3章で述べた問題点を踏まえて、システムが実装あるいは、強化すべき個々の機能を検討した結果、大きく分類して次の4種類の機能の強化、実現をすることにした。

4.1 編集機能の強化

これまでの、一つのファイルに一つの関数ということ想定し、システムが設計されてきた。そのため、関数呼び出し側の処理箱は参照箱として存在したが、関数を一つのまとまりとして定義することができなかった。そこで、編集機能強化として、関数定義を表現する、処理箱を作成し、一つのファイル内で複数の関数を定義できるようにする。

4.2 変換機能の強化

これまでのシステムでは、数値や文字列を利用したアルゴリズムの作成を想定し、構築されてきた。しかし実際、アルゴリズムを作成していく時、いきなり数値や文字を使った擬似コー

ドで記述することは、PAD 図作成時においても初心者には難しい。また、アルゴリズムを作成する時は、大抵の場合、まず問題を大局的に捉え、徐々に詳細化していく、段階的詳細化の手法が有効であると考えられる。そこで PAD 図上で日本語を利用した記述を可能とし、その PAD 図と擬似コード間の相互変換も行えるようにする必要がある。

4.3 アルゴリズム実行検証機能

PAD 図を作成し、最終的に C 言語形式のソースコードに変換した後、そのコードにどのようなエラーが発生しているのかの詳細を通知し、エラーがなければ、実際に実行し、変数値の変化を表示する。また、実行時に、ソースコードエディタと PAD 編集・表示ウィンドウ上で実行している箇所を提示し、対応関係を理解できるようにする。

4.4 データの保存と参照機能

3.3 節で述べた問題点を踏まえ、サーバ側で学習者が作成した PAD 図をソースコードの形式 (テキスト形式で保存) させる。そのための手段として、Java サブレットと連携し、ファイルの保存、参照、階層の移動など基本的な機能を実現する。

5 システムの実装

図 1 にシステム全体の構成を示す。ここでは、4 章で述べた機能を実現するために作成を行った、新たに追加した処理箱、解析部分、実行結果表示部分、保存参照部分について述べる。

実装に用いた Java 環境は、JDK-1.4.2 と Eclipse-2.1.2 である。

5.1 関数定義を表現する処理箱

現システムで利用できる PAD 図の表現を図 2 に示す。関数を定義するための PAD 箱として、一般的な規格で用いられる PAD 箱と区別する

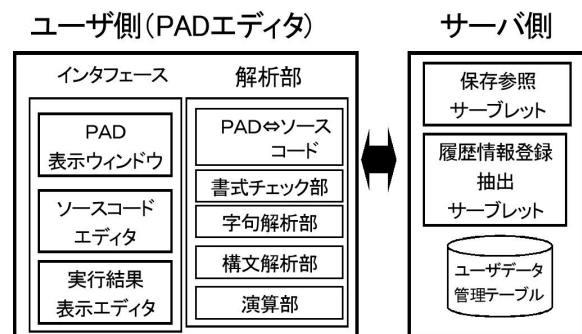


図 1: システム全体の構成

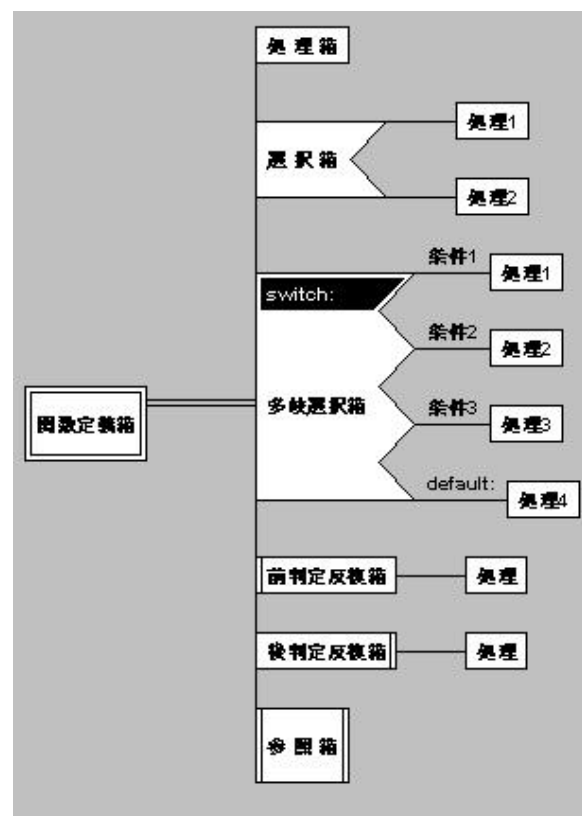


図 2: PAD 図の表現

ために図 2 左のようなシンボルとした。また、現システムで利用できる PAD 図の表現を図 2 に示す。また、PAD 図とソースコード間の相互変換パーサと、書式チェックパーサに日本語文の処理を追加し、自然言語を用いた PAD 図の作成や自然言語を用いた擬似コードと PAD 図を相互に変換することが可能となった。

上記の結果、一つのファイル内に記述された一通りの C 言語のソースから PAD 図への変換、または、抽象的な記述から段階的詳細化を繰り返し、最終的に実際の開発環境でもコンパイル可能なソースへの変換が可能となった。

5.2 アルゴリズム実行検証機能

ここでは、解析部分、実行結果表示部分について述べる。

5.2.1 解析部分について

アルゴリズム実行検証機能実現のために、箱を辿りながら実行させるのではなく、C 言語形式のソースコードに変換したものを解析し、実行しながら、PAD の処理内容と実行している処理内容をマッチングさせ、箱の色や箱内の変数情報を表示させるために、新たに C 言語形式のソースコード全体を解析するパーサを作成した。

(1) 字句解析部分

字句解析には、`java.io.StreamTokenizer` クラスを拡張し、C 言語の記号などを識別できるようにしたクラスを用いる。

(2) 構文解析部分

字句解析から得た字句情報をもとに構文木を生成し、構文エラーチェックを行う、構文定義には、C 言語の書式を限定的に適用した。作成された構文木を辿りながら、インタプリタ形式で実行される。

このように実現したため、実行処理中はメモリ使用量が多く、動作が遅くなるという欠点があるが、図 3 に示すように実行させたい関数を指定することで、関数単位での実行が可能である。

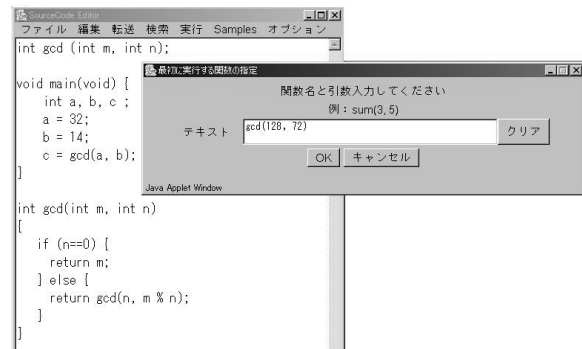


図 3: 関数単位で実行

5.2.2 実行結果表示部分

実行結果の表示部分として、以下の三つの GUI(Graphical User Interface) を作成した。

- エラー情報表示ウィンドウ
- 変数情報表示ウィンドウ
- `printf` 関数による出力ウィンドウ

(1) エラー情報表示ウィンドウ

エラー情報表示ウィンドウ (図 4) は、実行した際、エラーがあった場合、その行番号と詳細情報を表示する。例えば、下記のような重複宣言エラーのあるソースコードを解析すると、図 4 のように表示される。

```
void main(void)
{
    int a, b, m, n;
    int a;
    /* 以降省略 */
}
```

(2) 変数情報表示ウィンドウ

図 5 に示したように変数情報表示ウィンドウは、実行後の全ての変数情報が表示される。

(3) `printf` 関数による出力ウィンドウ

`printf` 関数による出力の様子を図 6 に示す。このウィンドウは、ソースコード内に `printf` 関数の参照文を記述すると、その構文木を辿った時、そこに記述された、文字列や変数が出力



図 4: エラー情報表示ウィンドウ

される。この関数の実装によって、学習者が実行中に知りたい、変数情報を以下の例のように、printf 関数の参照文を記述することでその情報を得ることができる。

例えば、以下のようなソースを実行すると、図 5、図 6 のように実行結果が表示される。

```
int a[] = {2, 3, 7, 11, 31, 50, 55,
          70, 77, 80};

void main (void)
{
    int N, key, low, high, mid;
    N = 10;
    key = 55;
    printf("配列の要素数 N=%d\n 探索するデータ key=%d", N, key);
    low = 0;
    high = N-1;
    while (low <= high) {
        mid = (low+high)/2;
        if(a[mid] <= key) {
            low=mid+1;
        }
        if(a[mid] >= key){
            high=mid-1;
        }
    }
}
```



図 5: 変数情報表示ウィンドウ

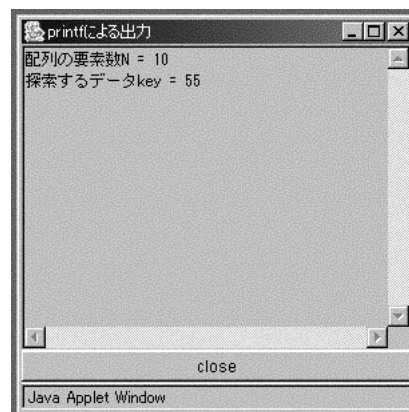


図 6: printf 関数による出力ウィンドウ

5.3 学習データの保存参照機能

学習データの保存参照は、HTTP(Hyper Text Transfer Protocol)を利用して、PAD エディタ(アプレット)とサーバ(サーブレット)の間で通信し、図7で示すように、ファイルの内容、ファイル名、ユーザ名、カレントディレクトリのファイルリスト、サーバ側での処理内容をカプセル化した、オブジェクトデータを送受信し、ログインしたユーザ名のディレクトリ内で保存・参照等の処理を行う。

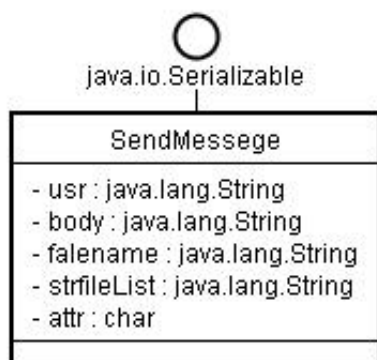


図 7: アプレット - サーブレット間で行き来するオブジェクト

このように図7で定義したクラスのオブジェクトをアプレット-サーブレット間でやり取りすることで、学習データの保存・参照、ディレクトリ階層の移動が可能となった。

6 評価と課題

本研究で強化・実装した機能によって、以下に示すようなことが可能となった。

(1) 編集機能・変換機能の強化

5.1 節で述べた機能強化によって、PAD 図による自然言語を用いた、抽象的なアルゴリズムの記述から、PAD 図とソースコードとの相互変換を繰り返しながら、擬似コードそしてC言語形式のソースコードへと詳細化する過程の学習を支援することが可能となった。

(2) アルゴリズム実行検証機能

5.2 節で述べた機能の実装により、学習者が

自ら作成したアルゴリズムを検証し、自分自身の力で問題を解決するための一支援が可能となった。

(3) 学習データの保存参照機能

5.3 節で述べた機能の実装により、学習者が以前に作成したアルゴリズムを見直し再考したりといった反復学習や、サーバ側に保存することで、学習情報を利用者間で共有できるため、協調学習の一支援が可能となった。

しかし、一方で、以下の三点が課題としてある。

(1) 相互変換時における課題

現在、ソースコードから PAD 図へと変換する際、コメント文「/*~*/」で囲まれた部分は、全て取り除かれてしまう。段階的詳細化を行う過程で不要な部分は取り除いても良いが、学習者にとって有益なコメント情報も数多く存在すると考えられる。現在、そのための手段として仮想箱を作成し、C 言語で用いる「/*~*/」で囲まれた部分を PAD エディタ上にも反映して行くためのシステム作りを進めている。例えば、下記のようなコードを入力すると、図8のように PAD 編集・表示ウィンドウ上に表示される。

```
void main(void) {
    int m, n ;
    /*初期値の代入*/
    m = 128;
    n = 72;
    /*最大公約数を求める処理*/
    while(m != n) {
        if(m > n) {
            m = m - n;
        }
        else {
            n = n - m;
        }
    }
}
```

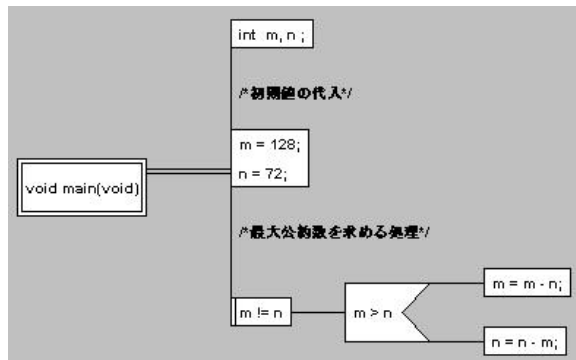


図 8: コメント情報の反映イメージ

(2) アルゴリズム検証機能における課題

5.2 節で述べた機能について一通り実装したが、まだ対応していない構造や演算も多く存在する。エラー解析に関しては、十分に拾いきれないエラーや一つのエラーが原因で生じるエラーを複数拾ってしまうという問題もある。したがって、これらの機能の強化と、まだ実現できていない PAD 表示ウィンドウ上とソースコードウィンドウ上での実行時の対応を進めなければならない。また、実行結果などの表示形式の工夫も今後の課題である。

(3) サーバ側のシステムに関する課題

5.3 節述べた機能実現によって、学習データを学習者同士で共有することも可能となった、しかし、ファイルに対するアクセス権限の制御など、実際にシステムとして提供する際に必要である機能が不十分である。

今後は、5 章で示した図 1 にある、学習履歴の管理等を含めてサーバ側で学習を管理する機能を実現する必要がある。

7 おわりに

本研究で作成したシステムは、予定していた機能の実装が完全には済んでいないために、残念ながらまだ、C プログラミングの学習支援ツールとしては十分に機能するレベルには達していない。しかし、今回実装した機能によって、アルゴリズムの作成、検証と一連の学習を支援することが可能となった。今後は、未実装

の機能の実現を急ぐとともに、課題として挙げた機能を実装し、学習支援システムとしての完成度をさらに高いものとしていたいと考えている。

また、1 章で述べたように、学習者が自らの判断で学習を進めることができるようにするため、本システムの機能強化だけでなく、PAD エディタの利用マニュアルや PAD エディタを利用した学習教材等の作成を行い、実証評価を行いたいと考えている。

参考文献

- [1] 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, 2000.12.
- [2] 神村 伸一: 構造化チャートを利用したアセンブリ言語教育の提案, 情報処理学会研究会報告, 1999-CE-51, pp.1-7, 1999.7.
- [3] 斐品正照, 松瀬賢太, 河村一樹: アルゴリズム学習を支援する JPADet の評価と教材開発, 情報処理学会研究会報告, 2002-CE-68, pp.31-38, 2003.2.
- [4] 阿部圭一著:「ソフトウェア入門」(第 2 版) 共立出版 (1989).