

## 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 —

石田 真樹      桑田 正行

電気通信大学

E-mail: {ishida, 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、本研究では、アルゴリズムの表記に PAD(Problem Analysis Diagram) を用いて 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ための支援ツールを作成し、このツールを利用した学習支援システムの構築を行っている。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 言語形式に変換したソースコードとの関係から、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。システムは Java アプレットとして提供され、学習者は利用するプラットフォームに依存することなく、いつでも利用することができる。

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

## WWW Based Learning Environment for C Programming

— An Algorithm Learning Support System with PAD Editor —

Masaki Ishida      Masayuki Kuwada

The University of Electro-Communications

E-mail: {ishida, 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. In this paper we describe about a learning support system that use a support tool to learn to make an algorithm in C programming. This system use PAD(Problem Analysis Diagram) for the representation of algorithms. Learners can make PAD in an interactive environment and study to make an algorithm in C Programming by referring the relation between an algorithm of this PAD and source codes in C language. This system is provided as Java applet, so learners can use it any time independent of their platform.

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

## 1 はじめに

近年、社会におけるさまざまなシステムにコンピュータが導入され、これによってシステムの電子化・自動化が進行している。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もまた、拡大し続けている。

コンピュータで使用される OS やアプリケーションなどのソフトウェアを記述するプログラミング言語は数多くある。しかし、ある一つのプログラミング言語を用いてプログラムを記述した場合でも、その記述形式がどのように他のプログラミング言語と異なるものであっても、与えられた問題の解決のために作成されるアルゴリズムの構造は基本的に同一である。そして、ここで構築されたアルゴリズムが効率のよいものであるかどうか、汎用性に優れているものであるかどうか、ということはシステム全体の性能や後々の拡張性に関わってくる。したがって、効率のよいアルゴリズムを構築する技術が重要となるが、アルゴリズムを構築する技術を習得するのは、プログラミング技術の習得を目指す学習者、特にそれが初心者である場合には、多くの場合難しいものとなる。

本研究では、プログラミングにおけるアルゴリズムの作成技術の習得に焦点を絞った学習支援システムの構築を目指している。アルゴリズムの表現に PAD(Problem Analysis Diagram) を用いることで視覚的にアルゴリズムの構造を把握し、理解できるシステムの構築を目指す。本稿ではこれを実現するためのツールとして作成した PAD エディタと、それを用いた学習支援システムの構築について報告する。

## 2 全体的な構想

コンピュータを学習支援に用いるということの利点を生かすために、対話的な形式による学習支援環境とネットワークの利用、学習履歴の管理という視点からの学習支援システムの構築を目指す。具体的には、以下に挙げる項目がその構想の柱となる。

- 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の構築

- 学習者が利用するコンピュータのプラットフォームに依存せず、特別に用意するソフトウェアも必要としないシステム
- インタフェースを対話性の高い、使いやすいものに構成し、学習者が複雑・面倒な操作を行うことなく学習を視覚的に進められる環境の提供
- 学習履歴の管理を行い、それを参照することで実現できるサービスを提供するシステムの検討
- 他の学習者との情報の交換を行うことのできる環境の提供

これらの構想を実現するために、Java でシステムの構築を行い、それを WWW (World Wide Web) で提供することで高い対話性を有しつつも WWW ブラウザ上から簡単に利用できるアルゴリズム学習支援ツールの作成を行った。

## 3 システム設計

前章における全体的な構想を踏まえ、次のような項目を考慮してシステムの設計を行う。

### (1) PAD の利用

アルゴリズムは、ある一連の仕事を行うために必要な手順の全体である。一般に、プログラミング言語のソースコードはテキストで記述されるため、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。日常使用している自然言語でアルゴリズムを記述できればよいのであるが、自然言語は曖昧であるために最終的にコンピュータで実行するプログラムを作成するためのアルゴリズムの記述には適さない。

アルゴリズムを図的に表現することにより、アルゴリズムの全体構造を視覚的にとらえることができ把握しやすくなる。アルゴリズムの図的な表現としては、流れ図(フローチャート、flow chart)、PAD(Problem Analysis Diagram)、N/S チャート(Nassi/Schneiderman chart)、HCP(Hierarchical and Compact description chart)、SPD(Structured Program

Diagram) などがある<sup>[3]</sup>。

本研究では、他のものより見やすくわかりやすい、対象とするプログラミング言語の構文へ簡単な変換規則で機械的に変換できる、また保守性に優れているという特徴を持つ、問題分析図 PAD を使用してアルゴリズムを表現する。

よく使われる流れ図は、制御の流れが 4 方向(前後左右)になり、制御構造が把握しにくいので好ましくない。

## (2) 対話的な操作環境

学習者が初心者であることを想定して対話的な操作性に優れた GUI(Graphical User Interface) を有するシステムとする。これにより学習者が難しい操作を必要としないだけでなく、視覚的な情報の変化を追うことによる、より視覚的な学習の実現を計る。

## (3) Java によるシステムの構築

システムを Java アプレットとして作成することで、WWW での提供を可能とする。これにより、学習者はプラットフォームに依存することなく、さらに特別なソフトウェアも必要とせず、広く流通し、よく利用されている WWW ブラウザを通じてこのシステムを利用することができる。そして Java の高度な GUI 作成機能を用いてインタフェースを作成することで、システムと学習者との対話性を高いものとすることができる。

## 4 個別機能の仕様

全体的な設計方針に従ってシステムが実装すべき個々の機能を検討した結果、大きく分類して次の三種の機能の実現を試みることにした。

### (1) 対話的操作による PAD の作成

PAD に追加する箱の種類やその位置の指定などをマウスで行う。箱に格納するテキストは、テキストフィールドからキーボードで入力する。ここで入力されるテキストは、必ずしも C 言語の書式に従わなくてもよい。これは、自然言語を用いた疑似コードによるアルゴリズムの記述に対応するためである。C 言語の書式に従っていないテキストは、全て疑似コードとみ

なされる。

### (2) PAD とソースコード間の相互変換

作成した PAD を C 言語形式のソースコードに変換する。この変換作業によって、PAD の示すアルゴリズムが C 言語ではどのように記述されるかが明示される。

さらに、C 言語のソースコード(疑似コード含む)を PAD に変換する機能も実装する。身近で利用できるよいプログラムのソースコードを PAD に変換することにより、プログラムの構造や処理の流れを視覚的に把握でき、アルゴリズムの学習に役立つ。この場合、C 言語のソースコードを解析するためのパーサが必要となる。このパーサクラスのソースコードの作成は JavaCC を用いる。JavaCC とは、構文規則を記述したファイルからパーサの働きをする Java クラスを作成する Java アプリケーションである。構文規則を記述したファイルを JavaCC でコンパイルすることで、入力された文字列が定められた構文規則に従っているかどうかを判定する Java クラスのソースコードが作成される。

PAD と C 言語のソースコードの両方向の変換が可能になれば、双方の間にある関係がより明確に理解できるようになると期待される。

### (3) アルゴリズムの実行結果表示

箱に収められているテキストがすべて C 言語の書式に従っている場合は、これを実際に実行してその結果と結果までの処理の流れを表示することで、より理解を深められる。このとき利用者に提示すべき情報としては、

- 変数の内容の変化  
アルゴリズム内で宣言された変数が、その内容を実行過程でどのように変化させていくのかを表示する。
- 各箱の実行順序  
実際に PAD の示すアルゴリズムが実行された場合、どのような順番で各箱が実行されるのかを表示する。
- 実行時に発生したエラー  
実際に実行した結果、変数の初期化のし忘れや同名の変数の二重宣言等の構造的なエ

ラーが判明した場合には、その場所を表示する。  
といったものが挙げられる。

## 5 PAD エディタの実装

図1にPAD エディタの実装を示す。実装に用いたJDKのバージョンは1.1.8である。

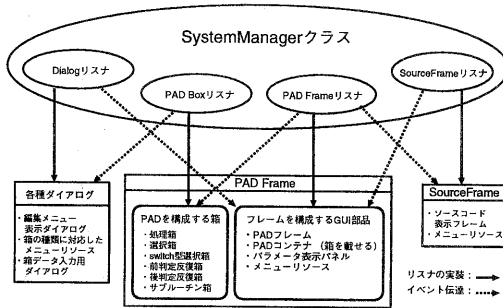


図 1: PAD エディタの実装

上図のように、PAD エディタを構成する各GUIコンポーネントは次のように大きく三種類に分類される。

- PAD フレームとそこで用いられるコンポーネント
- PAD 編集用のメニューが登録されるダイアログ
- ソースコードの表示と編集を行うフレーム

### 5.1 PAD フレームと PAD 編集用のダイアログ

PAD 編集フレームの外観を図2に示す。PAD を構成する各種の箱は、それぞれがJava.awt.Component クラスを継承して定義されたクラスのオブジェクトである。これは通常、白の背景色に黒のテキストで表示される。これらの箱はその前後の箱との関係を示すデータをフィールドとして有しており、配置された箱の集合自体が全体で一つのリスト構造のデータとなる。

個々の箱は機能的には一種のボタンのようなものとなる。マウスカーソルに対する動作としては、マウスカーソルが箱の領域に侵入した場

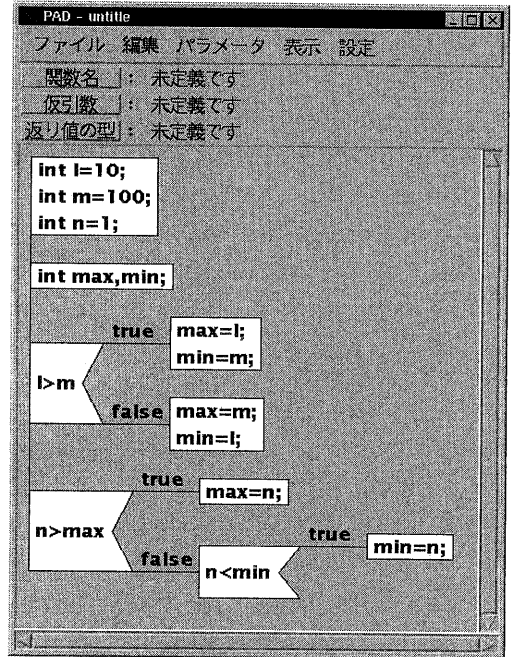


図 2: PAD 編集フレーム

合に箱は選択状態となって背景色が赤のハイライト表示となり、マウスクリックが検出されるとアクションイベントを発生させて、選択状態を維持したまま、この箱を載せているPADフレームにそのイベントを送信する。PADフレームはこのイベントを受信すると、編集メニューが登録されているダイアログを表示するためのイベントを発生させて編集メニューを表示させる。これが図3の状態となる。PAD編集用のダイアログは、各種の箱をクリックした場合に表示されるメニューの他、追加する箱のデータ入力やファイル入出力操作のためのダイアログ等も含む。

編集メニュー上にマウスカーソルが入ると、箱の場合と同じように背景色が反転し、メニューがハイライト表示となる。図3の状態例えば「箱を上追加」選択すると、メニューダイアログは消え、図4に示す新たなダイアログが表示される。このダイアログにテキストを記入し、箱の種類を選択すれば、新たな箱がPADに追加され、自動的に再配置が行われる。

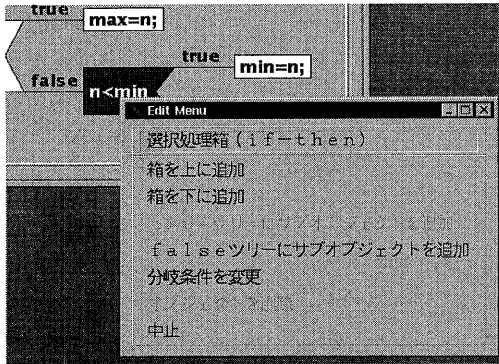


図 3: 編集メニュー

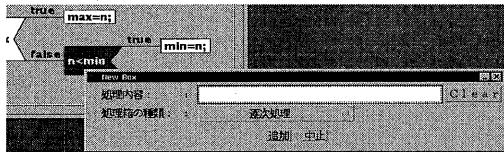


図 4: PAD の作成

制御構造を選択すると、必要な情報(条件式、処理など)の入力を要求され、それを入力すると、新たな箱がPADに追加され、自動的に再配置が行われる。

PADの作成のために実装されている編集機能に関しては、現状では削除や内容文字列の変更など、最低限の機能しか確保されていない。UndoやRedo、箱のコピーなど、編集機能として当然実装されているべき機能がまだ確保されていないので、使いやすさの点から考えれば、決して十分とは言えない。編集機能の強化は今後の重要な課題である。

4章(3)で挙げた実行結果の表示機能についても未実装である。これは、C言語の書式に適合しているかどうかを判定するパーサが完成していないからである。このことについて別の視点からの検討も行っているが、この詳細は次章で述べる。

PADの上辺にある三つのパラメータ(関数名、仮引数、返り値の型)の表示領域は現時点ではまだ未実装の機能のためのものであり、こ

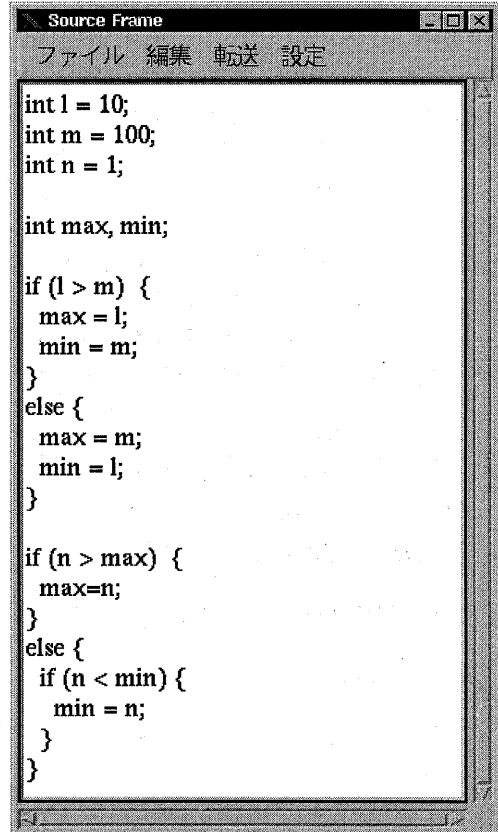


図 5: ソースコードの表示・編集フレーム

れについては次章で述べる。

## 5.2 ソースコードの表示・編集フレーム

ソースコードの表示と編集に用いるフレームは、PADが示すアルゴリズムをC言語(あるいは疑似コード)に変換した結果を表示するフレームである。

図5はソースコードの表示・編集フレームの外観である。ここに表示されているソースコードは、図2のPADが示すアルゴリズムをC言語の書式に従ってコード化したものである。このコード化は、システムが自動でPADの構造を解析して行う。反対に、このフレームのテキストエリアに記述されているCコード(または疑似コード)を編集後、PAD編集フレームに転送して、修正したPADを表示させることも可

能である。

### 5.3 各コンポーネントの連携動作の管理

前述の三種のコンポーネントはそれぞれが大きく一つのクラスパッケージとしてまとめられている。クラス間の階層構造を図6に示す。この図にあるように、すべてのGUIコンポーネントの上層に当たる位置にSystemManagerクラスが定義されている。SystemManagerクラスは図1に示すようにその内部にインナクラスとして各種のイベントに対するリスナクラスが定義されており、これらのオブジェクトを各コンポーネントに実装することでコンポーネント間を往来する全てのイベントの管理を行う。このため、SystemManagerクラスはシステム全体の連携動作の管理を行うことになる。

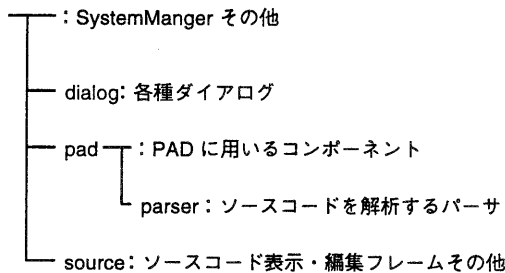


図6: クラス・パッケージの階層構造

SystemManagerクラスが実際に行う管理は、あるGUIコンポーネントから発生したイベントを別のGUIコンポーネントに伝達するという作業となる。したがって、全てのコンポーネントから発生するイベントを集中的に管理するといっても、その実際の作業は各コンポーネント間でのイベントの中継作業がほとんどを占め、箱の追加や並べ替えといった実質的な作業のためのメソッドの大部分はそれぞれのコンポーネントが持ち、SystemManagerクラスはその実行を指示するだけである。

このような構造をとるため、システムを構成するGUIコンポーネントの実装の大部分が機能的に独立しており、単独での動作が可能なものが多い。そのため、将来的な機能拡張や他のアプリケーションでの再利用を容易に行うこ

とができる構造となっている。例えば、PAD-Frameクラスはコンストラクタの引数にファイル名を与えることで指定されたファイルの内容をC言語のソースファイルとして解析し、単独のフレームとして起動し、ソースファイルのアルゴリズムをPADにして表示する。同様に、各種のダイアログやソースコードの編集フレームもまた、単独での起動が可能である。

このようにGUIコンポーネントが機能的に分離しているために、SystemManagerクラスは各コンポーネントを独立して管理することができるほか、同種のフレームオブジェクトを同時に作成することも容易に行うことができる。これは、例えばPAD上のサブルーチンの箱をクリックすることで新たなPAD編集フレームを作成してそのサブルーチンのPADを表示するというような場合に便利な構造である(ただし、現時点ではこの機能はまだ実装されていない)。

## 6 評価と課題

本研究で作成したPADエディタは未実装の部分が多くあるものの、対話的操作によるPADの作成・PADとソースコードの相互変換という二つの機能についての実装が完了した。この二つの機能によって学習者は、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となる。このことにより本システムは、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の学習支援を行う。

一方で課題となっているのは、大きく分けて三点ある。

### (1) 編集機能の強化

5.1節で述べたように、現状のように編集機能が限られてしまっは、いかに対話的にPADの作成を進められたとしても使いやすいものとは言えない。したがって、PADの編集にあたる機能面の強化を進めなければならない。

### (2) 未実装となっている機能の実現

現時点では、4章で述べた(1)～(3)の機能

のうち、(3)についてまだ完全には実装されていない。この機能が実現すれば、本システムはアルゴリズムの構造理解に関して、かなり有力な支援システムとなり得ると考えている。現時点でこの機能が実現されていない原因は、先にも述べた通り、C言語の書式に適合しているかどうかを判定するパーサが完成していないことにある。これは、C言語の書式の、特に数式と変数の表現について、どこまで忠実に再現するのかを検討中であることも原因の一つとなっている。C言語での記述では、例えば下記のような

```
i = j = 0;
y = - ++x;
z = x > 0 ? 1 : 0;
w = x + y > z;
a = (float) b;
a = sizeof (float);
**p = 1;
```

といった記述は全て、文法上正しい式となる。しかし、このシステムの利用対象をプログラミングの初心者と考えるならば、上記の記述の中でもいくつかについては、使わない方がむしろわかりやすい。例えば、3行目の

```
z = x > 0 ? 1 : 0;
```

という式は、形式は代入式だがその処理内容は明らかに if-else 型の分岐処理である。このような記述について、どこまで再現するのか、その線引きをどう行うのかを、検討しなければならない。

### (3) データの保存と再利用

最後に、本システムは 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ようになってきている。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ることとなる。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べきである。現在、そのための手

段として、Java サブレットと連携することでデータをサーバ上に保存するという方法を検討しており、そのためのシステム作りを進めている。この方法では、システムが存在しているサーバ上に複数の利用者のデータが共存することになり、これを利用者間で共有し、互いに情報交換することで有効に活用できると思われる。さらに本システムでは、現時点ですでにサブルーチンを表す箱が実装されており、この箱と保存データとを連携させることによって、サーバ上の保存データをサブルーチンとして再利用することも可能となる。保存した PAD データをサブルーチンとして再利用するためには、このサブルーチンを呼び出す際に必要となる関数名と仮引数、戻り値の型の設定をデータ保存の際に行っておく。図 2 における上辺のパラメータ表示領域は、このとき保存されるパラメータの内容を表示するためのものである。

## 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、実装が済んでいる部分に関しては、PAD の編集とソースコードの変換機能によって、両者の関係を視覚的な情報としてかなり明確に提示できたと考えている。今後は、未実装の機能の実現を急ぐと共に、課題として挙げられた編集機能の強化、データの保存と再利用を行うシステムの実現を進め、完成度をさらに高いものにしたいと考えている。

## 参考文献

- [1] 谷川健, 富士隆, 三枝武男: COLBA を用いた分散学習システムの開発, 情報処理学会論文誌, Vol.39, No.7, pp.2380-2390 (1998).
- [2] C. Chou: Developing Hypertext-Based Learning Courseware for Computer Networks: The Macro and Micro Stages, IEEE TRANSACTION ON EDUCATION, Vol.42, No.1, pp.39-44 (1999).

- [3] 阿部圭一著：「ソフトウェア入門」(第2版) 共立出版(1989).