

PAD によるソフトウェア開発システム†

岡田 謙 一†† 北川 節††

プログラムの制御構造をより明確に表現するために、従来のフローチャートにかわり幾つかの木構造チャートが発表されてきた。しかし、二次的な広がりをもつ木構造チャートを面積的に制約のある端末の画面上に表示するには、最適配置、視野移動などの画面処理を必要とし、小型計算機では応答時間に決定的な影響が与えられる。本論文は、木構造チャートの一つである PAD を取り上げ、階層化させた PAD を二つの画面に分割して表示する方法を提案し、それを実現したシステムについて述べている。本システムの特徴は、(1) PAD のもつ情報をフレーム接続部とテキスト部に分割し、それぞれを2台のマイコンに並列処理させることにより、視野の拡大と処理の高速化を図るとともに、(2) 各手続き間のコーリングシーケンスを自動的に作成してページ PAD という形で表現し、視点の位置を明確に把握することを可能にした。さらに、(3) 作成された PAD のプロッタへの清書出力機能、ならびに既存言語プログラムへの変換機能を備えた。本システムにより、小規模なグラフィック機能の端末をもつ小型計算機による実用的な木構造チャート開発ツールを実現する一つの方法を示した。

1. はじめに

プログラムの制御構造をより明確に表現するために、従来のフローチャートにかわり幾つかの木構造チャートが発表されてきた^{1),2)}。それと同時に、これらの木構造チャートの入力、編集が可能なシステム³⁾⁻⁵⁾、ソースプログラムから木構造チャートを自動作成するシステム⁶⁾⁻⁸⁾、入力された木構造チャートをソースプログラムに翻訳するシステム^{9),10)}などのいわゆる木構造チャート開発ツールの研究も進められてきている。

筆者らは、木構造チャートの一つである PAD (Problem Analysis Diagram)¹¹⁾の形式により、プログラムの入力、編集、実行を行う会話型システムを開発し、ミニコン上でこれを実現した。

PAD の記法では、フローチャートと同じように処理内容を各記号(以後フレームと呼ぶ)の中に記述していくので、視覚的に非常にわかりやすく制御構造を理解しやすいという長所があるが、反面スペース配分が難しくまた何らかのグラフィック処理が必要なため、大型で解像度が高く、また高速なグラフィック処理が可能な端末を必要とする。このことはフレーム数、ならびにフレームの処理内容を示すフレーム内のテキストが多くなったときにとくに問題となってくる。すなわち、詳細 PAD と呼ばれるコーディングレベルの PAD を対象とした場合、処理単位を一つ

の手続きに限定しても、手続き全体を1画面に表示することは、小規模なグラフィック端末ではよほど小さな手続きでない限りほとんど不可能である。

この問題に対処するためには、手続きの階層化、フレームの最適配置処理による PAD の縮小化、限定された視野による手続きの部分表示などの手法が考えられる。しかし、制御構造自体やテキストの変更、修正が頻繁に行われるエディタでは、最適配置処理や視野移動などの画面処理はオーバーヘッドとなり、小型の計算機では応答時間に決定的な影響が与えられる。

そこで筆者らは、小型計算機上でも実用的な PAD ツールを実現するため、小規模なグラフィック機能をもつ2台のマイコンにより構築されたマルチスクリーン方式の PAD 端末¹²⁾とそれを使用したシステムを開発した¹³⁾。本システムの特長は、(1) PAD のもつ情報をフレーム接続部とテキスト部に分割し、それぞれを2台のマイコンに並列処理させることにより視野の拡大と処理の高速化を図るとともに、(2) 各手続き間のコーリングシーケンスを自動的に作成してページ PAD という形で表現し、視点の位置を明確に把握することを可能にした。さらに、(3) 作成された PAD のプロッタへの清書出力機能、ならびに既存言語プログラムへの変換機能を備えた。

本論文は、2章で二次元的広がりをもつ PAD を限られた画面に表示するため本システムに適用した PAD の分割と階層化の基本概念を述べる。そして、3章で実現したシステムの概要を述べ、4章、5章で本システムのもつ各機能について詳説し、6章で評価を示す。

† A Software Development System Using PAD by KEN-ICHI OKADA and MISAO KITAGAWA (Department of Instrumentation Engineering, Faculty of Science and Technology, Keio University).

†† 慶応義塾大学理工学部計測工学科

2. PAD の分割と階層化

PAD はテキストの集合である従来のプログラムとは異なり、制御構造が木構造で視覚的に表現され、処理の種類がフレームの形状で識別されるという特徴をもつ。この PAD を画面上に表示しようとするとき単位処理あたりの表示量が増えるとともに、表示レイアウトを考慮する必要が出てくる。本システムでは PAD を階層化し、2 画面に表示することにより入力、編集処理の効率化を図っている。

2.1 PAD の分割表示

テキストを内部に含む本来の PAD では、可変量のテキストをフレーム内に収めるためのフレーム作りと、大きさの異なるフレームからなる木構造の効率的な配置に、多くの処理時間と情報を要する。この問題は、フレーム内のテキストが可変量であることに起因する。そこで、構造ならびにテキストの修正が頻繁に行われる編集作業においては、テキストを PAD から分離することを考える。テキストをフレームから取り除くことにより、選択フレームを除いたフレームの大きさを一定にすることができる。選択フレームについても水平方向の長さを一定にすることができ、また垂直方向の長さは枝数、従属するフレーム数によって変化するが、他のフレームの大きさが一定になったことにより長さの決定は容易になる。したがってテキストを PAD から分離することによってフレーム作り、最適配置に要する処理時間、情報が大幅に減少する。

テキストを取り除いたフレームには、フレームの識別とテキストとの対応付けのために、フレーム番号と呼ばれる番号を付加する。このフレーム番号は、制御の深さを表し水平方向にカウントされるレベル番号と、同一レベルでの制御の流れを表し垂直方向にカウントされる連続番号から成り、次の形式で表される。

$l.s$ (l : レベル番号, s : 連続番号)

フレーム番号は、取り除かれたテキストの代りにフレーム内に収められ、分離したテキスト (以後内部テキストと呼ぶ) はフレーム番号に従属する形で他のテキストと識別される。分離したフレーム群と内部テキストは、それぞれ独立した画面に同時に表示されるが、着色や字下げ処理により視覚的にも対応が容易になるようにする。

2.2 PAD の階層化

編集作業中には視野の移動が頻繁に行われるが、グラフィック画面を扱う場合、この視野移動が応答時間

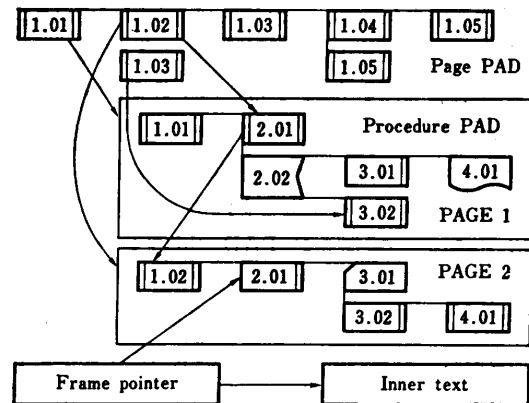


図1 階層表現した PAD—ページ PAD と手続き PAD
Fig. 1 Hierarchical structured PAD—page PAD and procedure PAD.

に大きな影響を与える。そこで、視野を一定の単位に固定して画面の書換えで視野を移動することとし、その単位を手続き、つまり一つの定義フレームで始まるフレーム群とする。編集作業を考えた場合、経験的に一つの手続きが同一視野内にあることが望ましいし、また手続き全体が1画面に存在することによって視野移動の回数は減少する。この1画面に表示される、テキストを分離したフレーム群を“手続き PAD”と呼び、この画面をとくにページと呼ぶ。したがって PAD モジュールは図1のようにページ化される。

従来のテキストエディタ、フルスクリーンエディタでは、一般に印刷されたプログラムリストと表示されたラインナンバで現在の視野を把握する。本システムの PAD エディタは、手続きを表示単位としているため視野の把握は容易になるが、さらに視点の位置を明確にするため全体のモジュール構造を表す新しい PAD を生成する。これは手続き PAD 間の相互参照関係を PAD で表現したものであり“ページ PAD”と呼び、このためにさらに1画面用意する (図1)。モジュール構造を表すには静的構造と動的構造が考えられるが、PAD エディタが単に PAD ファイルを作成するだけでなく、完成された PAD ファイルの実行、既存言語プログラムへの変換を行うシステムの一部として位置付けられていることから動的構造を採用している。したがって PAD ファイルの構造としては冗長な部分が含まれるが、論理構造が明確になりデバッギングに有効であると思われる。またページ化による視野の孤立化を防ぐとともに、無意味な視野移動を回避することができる。

このページ PAD は PAD の入力、編集過程で自

動的に作成されるが、これは入口名参照がフレームの形状で認識されるという PAD の性質により、比較的容易に行うことができる。ページ PAD のフレームは手続き PAD と同様にフレーム番号を含むが、ページ PAD のフレーム番号は、各手続き PAD のトップレベル（レベル番号1）のフレーム番号である。したがってページ PAD 内のフレームは、同一フレーム番号をもつ各手続き PAD のトップレベルフレームと内部テキストを共有することになる。

このように PAD エディタでは、次の三つの階層の情報に PAD を分割し、2画面に選択表示する。

- 1) ページ PAD (手続き PAD 間の相互参照関係を表す PAD)。
- 2) 手続き PAD (テキストを分離したフレームからなる手続き単位の PAD)。
- 3) 内部テキスト (フレーム番号によって対応付けられるフレーム内のテキスト)。

これらの画面は、フレームポイントと呼ぶポイントによって連結される。このフレームポイントは値としてフレーム番号をもち、現在注目しているフレームを指すことによって実際の入力、編集作業を管理する。手続き PAD 内でフレームポイントを移動することにより内部テキストが識別され、トップレベルで移動することによりページ移動を行うことができる。

3. システムの概要

本システムは図2に示すように、編集機能を受持つ PAD エディタ、消書出力機能を受持つ PAD ライタ、既存言語変換機能を受持つ PAD トランスレータから構成されている。

まず PAD エディタにより、PAD をキーボードか

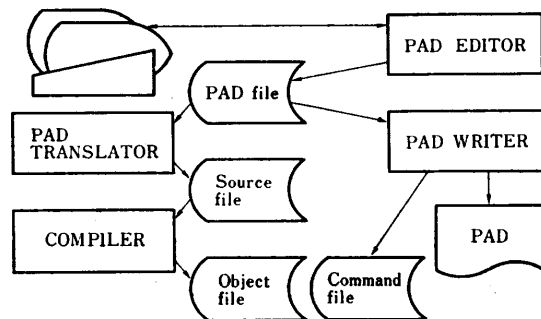


図2 システムの構成と処理の流れ

Fig. 2 Configuration and processing flow of the system.

ら入力、編集した〈file 名〉.pad という PAD ファイルを作成する。現在、PAD の内部テキストはシステム記述言語 SIMPL 準拠の形式で書かれているので、PAD トランスレータにより C 言語のソースファイル〈file 名〉.c に変換し、C コンパイラでコンパイルしてオブジェクトファイル〈file 名〉.o を作る。一方、PAD ライタは PAD ファイルを解析してプロッタコマンドファイル〈file 名〉.plt を登録し、これを実行することにより PAD を通常の形式、すなわち内部テキストをフレーム内に収めた形式でプロッタに出力する。

本システムで使用するマルチスクリーン方式の PAD 端末は、1 ドット 8 色で 640×400 ドットの小規模なグラフィック機能をもつ 2 台のマイコンで構築されており、最大 9600 ボーの RS-232 C でホストコンピュータと接続される。1 台のマイコンが、ホストコンピュータとの通信、ユーザコマンドの処理、PAD のグラフィック処理を行い、他の 1 台が PAD のテキスト処理を行う。

複数画面を使用する方法としてマルチスクリーン方式とマルチウィンド方式が考えられるが、本システムでは、(1)視野を可能な限り広くする、(2)グラフィック処理を少なくする、(3)表示単位あたりのデータ量を少なくする、(4)小規模なグラフィック機能のマイコンを利用して端末を構築する、等の理由によりマルチスクリーン方式を採用した。

4. 編集機能—PAD エディタ

PAD を扱うには、フレームの処理とテキストの処理を合わせて行わなければならない。PAD エディタでは、フレームと内部テキストが分離していることにより、フレームの大きさ、配置を考慮することなく連続して PAD を入力することができる。また編集作業時にも、ほとんどのフレームの大きさが一定であることから、フレームの移動、変更を容易に行うことができ、さらにしばしば発生するタイプミスなどによる内部テキストの変更、挿入時において PAD を書き直す処理から逃れることができる。

4.1 ページ画面と手続き画面

前述のように、本システムの PAD 端末は 2 台のディスプレイを備え、二つの画面に独立した情報を表示することができる。PAD エディタでは、階層化された PAD を効率よく取り扱うために、ページ PAD と手続き名リスト、手続き PAD と内部テキストの

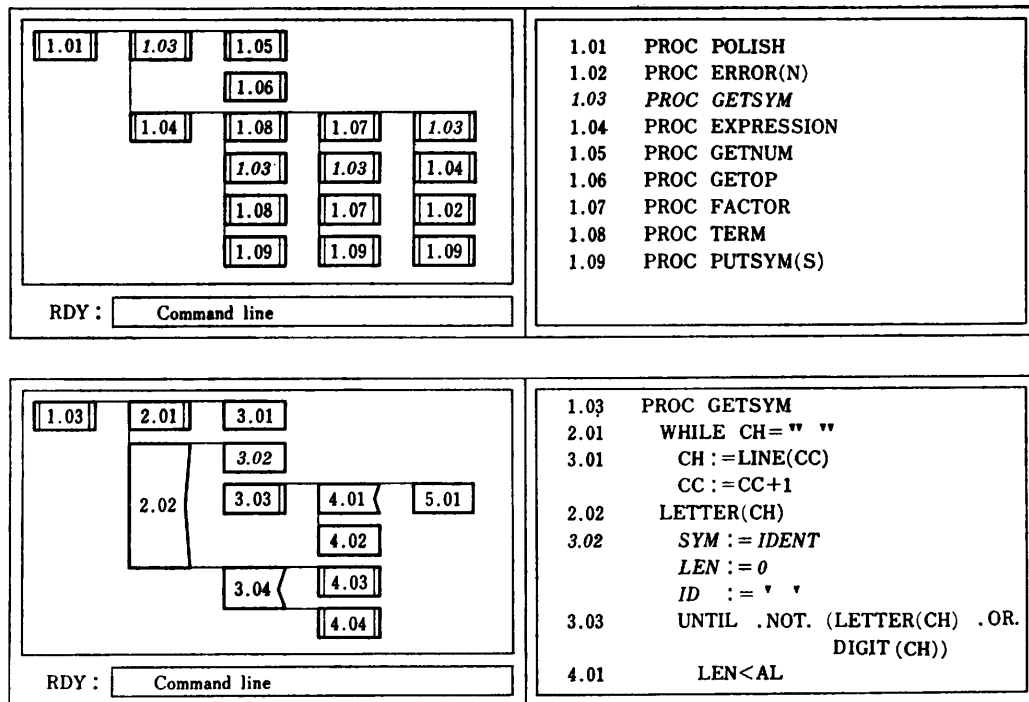


図 3 ページ画面(上)と手続き画面(下)の表示例

Fig. 3 Examples of page screens (upper) and procedure screens (lower).

組合せによる画面が用意されており、前者をページ画面(図3上)、後者を手続き画面(図3下)と呼ぶ。各マイコンは2画面分のVRAMを備えており、ページ画面と手続き画面の切り換えは瞬時に行われる。

ページ画面では、選択された手続きを示すフレームとその内部テキスト、すなわち手続き名が青く着色され、被編集手続きのモジュール全体の中で占める位置の把握を容易にする。手続き画面では、フレームポインタの指すフレームとその内部テキストが青く着色され、また制御の流れの分岐を示す判定、選択フレームとその内部テキストが緑色に着色され注意を促す。両画面とも横方向に16、縦方向に26の基本フレームを表示することができ、この16×26という値が一つの手続きPADで扱える最大の大きさとなる。また内部テキストは、フレームをツリーワークした順、すなわち通常のコーディング順に、フレーム番号に続きレベルによる字下げ処理をした形で、フレームポインタの指すフレームの内部テキストを画面の中心にして一つの手続きの中の25行が表示される。

4.2 PAD エディタの内部モード

PAD エディタは入力モード、編集モード、内部テキストモードから構成されており、編集モードから入力モード、内部テキストモードへの遷移はコマンドに

より指定され、その逆は空行の入力により起こる。次に各モードの機能について述べる。

1) 入力モード

8種類のフレーム(定義、反復、判定、選択、処理、入力、出力、宣言)に割当てられたファンクションキーに続き、その内部テキストをキーボードより入力することにより、新規なPADの作成あるいは既存のPADへのフレームの挿入を行う。一つのフレームに入れられる内部テキストの最大は50字×9行である。二つの画面には、作成中の手続きPADとその内部テキストが表示される。

2) 編集モード

ページ画面と手続き画面を切り換えながら、フレームポインタとフレーム番号を中心に表1に示すコマンドにより、PADをフレームレベルで編集する。このモードはPADエディタの中心となるモードであると同時に、PADライター、PADトランスレータの起動も行う。

3) 内部テキストモード

フレームポインタが指すフレームの内部テキストの編集を行うモードである。手続き画面が選択されるが、内部テキストを表示している画面の中心部に編集作業領域が設けられ、フレームポインタが指すフレー

表 1 PAD エディタのコマンドとその機能
Table 1 PAD editor commands and their functions.

コマンド	機能
編集モード	
Up, Down, Top, Bottom, DCL, {frame no.}	フレームポインタの移動
INsert, DElete, Change, COpy, Move, REWrite, ADD	フレームの挿入, 削除, 入換, 複写, 移動, 書換 選択肢の追加
Find	文字列の検索
X	ページ画面と手続き画面の切り換え
Save, OUTput, TRANsLate	PAD ファイルの保存, 出力, ソースプログラムへの変換
INsert, ENter, END	入力モードへの遷移 内部テキストモードへの遷移 PAD エディタの実行終了
内部テキストモード	
INsert, DElete, COpy, Move, REWrite, Change	行単位での挿入, 削除, 複写, 移動, 書換 文字列の入換

ムの内部テキストが行番号付きで格納される。この行番号と表 1 に示すコマンドを使用し、ラインエディタ形式で内部テキストの編集を行う。また編集作業領域の上下には前後のフレームの内部テキストが表示される。

各モードとも端末からホストコンピュータには、図 3 で示したコマンド行と必要な場合にはファンクションキー番号が転送される。ホストコンピュータから端末には、コマンド転送とテキスト転送の二つのデータ転送が行われる。コマンド転送は 10 バイトを単位としており、(1)コーリングシーケンスを示すページコマンド、(2)フレームの種類、連結関係、内部テキストの行数を示すフレームコマンド、(3)選択フレームの分岐先を示すケースコマンド、(4)モード、画面の切り換えをする端末制御コマンド、(5)文字列のサイズと種類を示すテキストコマンドがある。文字列転送は 80 バイトを単位としており、手続き名、内部テキスト、システムメッセージ等が前述のテキストコマンドに続いて送られる。

4.3 PAD ファイルの構造

PAD ファイルは次に示す四つのテーブルから構成される。ページテーブルは一つ作成され、各ページのフレーム数をレベルの深さごとに記録する。フレームテーブルはページ数だけ作成され、フレームの種類、

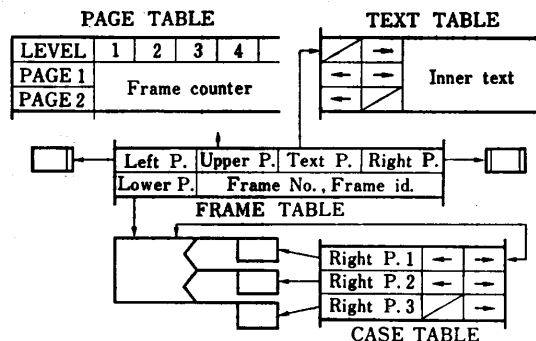


図 4 システムテーブル間の関係
Fig. 4 Relationships among system tables.

フレーム番号、接続関係、内部テキストへのポインタを保持する。ケーステーブルは選択フレーム数だけ作成され、選択フレームの分岐先を保持する。テキストテーブルはページ数だけ作成され、内部テキストを双方向リストにより管理する。図 4 に各テーブル間の関係の概略を示す。

フレーム間の接続情報としてフレームテーブルとケーステーブルが PAD 端末に送られ、端末自身が実際の配置を決定する。また、内部テキストの字下げ処理、ならびにフレームポインタが指している内部テキストのセンタリング処理も同様に端末自身が行う。

5. 清書出力機能と言語変換機能

編集作業中の PAD はフレームと内部テキストが分離されているが、最終結果として書類化するためには通常の PAD の形式、すなわちフレームの中にテキストを収めた形式に戻さなければならない。この章では、フレームに内部テキストを収め最適配置処理をして、X-Y プロッタに PAD の清書出力を行う PAD ライタと、作成された PAD ファイルを C 言語のソースプログラムに変換する PAD トランスレータについて述べる。

5.1 清書出力機能—PAD ライタ

PAD ライタは、PAD ファイルをページ単位で解析し、プロッタコマンドファイルを作成してディスクに登録する。そして、プロッタコマンドファイルを実行することにより、PAD を A3 横長方向で X-Y プロッタに出力する。出力形式としては、横方向すなわち制御の深さを示す方向には 1 枚の紙に収め、縦方向にのみページ替えることを許すものとする。PAD エディタでは PAD の入力、編集において各フレームの大きさ、配置をまったく考慮していないので、

PAD ライタはこの二つを決定しなければならない。

1) フレームの大きさの決定

フレームの大きさは内部テキストの量で決定されるが、横方向に制限があるので各フレーム独立に決定することはできない。そこで、まず(1)各フレームごとに内部テキストの最大長を求め、フレームの横方向の長さを仮決定する。(2)横方向にパスをたどり、制限長を越えるときはそのパス中に最長のフレームを求め単位長だけ短くする。この作業を横方向に制限長以下になるまで繰り返し、フレームの長さを修正する。(3)各フレームごとに内部テキストと決定されたフレーム長を比較し、内部テキストが長い場合には内部テキストを空白、カンマ、二項演算子などの位置で折りたたむ。

2) フレームの配置の決定

フレームの配置にあたっては、フレームをつなぐ線は1本の直線にすること、ページ替えによるフレームの分断は避けることを原則とする。まず PAD エディタと同じ形式、すなわち子フレームの先頭は親フレームのすぐ横に置く形に仮配置する。次に前述の原則を守れる範囲でフレームを上詰めて、配置を修正する。

決定されたフレームの大きさと配置からプロットコマンドファイルが作成され、ディスクに登録される。これを実行することによりプロットへは、日付、PAD ファイル名、ページ PAD、手続き名リストと各手続きの記載ページ、グローバル変数、手続き名、手続きに対するコメント、ローカル変数、手続き PAD の順に出力される。

5.2 言語変換機能—PAD トランスレータ

PAD トランスレータは、作成された PAD ファイルを既存の言語コンパイラを利用して実行するために、PAD ファイルを解析し通常のソースプログラムに変換するものである。

本システムでは、PAD の内部テキストはシステム記述言語 SIMPL (System Implementation Language) 準拠の形式である。これを SIMPL あるいは PASCAL のソースプログラムに変換するのは非常に容易であるが、まったく異なった形式の言語に変換することを試み、UNIX への移植も考慮して C 言語に変換する1パスのトランスレータを開発した。ホストコンピュータの制限から、一つの PAD ファイルは20個の手続きを最大としている。そこで PAD トランスレータは、複数の PAD ファイルを別々に変換

し、コンパイル後リンクできるように考慮されている。

6. システムの現状と評価

本システムは16ビットのミニコン (MS-30) 上で実現されており、PAD エディタは FORTRAN 77 で3,300行、PAD ライタと PAD トランスレータは C でそれぞれ1,900行、3,200行で記述されている。PAD 端末の制御プログラムは BASIC で記述されており600行と300行となっている。現在、端末制御プログラムは、画面の配置を実験的に変えていることからインタプリタ形式で実行されているが、決定された時点でコンパイルし ROM 化する予定である。

会話型システムでは、応答時間の速さが使いやすさに大きく影響する。PAD エディタでは、ほとんどのコマンドに対する応答時間は1秒前後であり実用上十分といえるが、一つの手続き PAD をすべて書き直すような処理には、端末のプログラムがインタプリタで実行されていることもあり8秒ほどかかる。しかしフレームと内部テキストが分離していること、ならびに一つの手続きが1画面に表示されていることにより、画面書き直しの必要性が大幅に減少し、はからずも本論文で提案した方法の有効性が示された。

画面上の制約から一つの手続きは基本フレームで16×26に制限されているが、手続きをあまり大きくしないことが読みやすいプログラムを作る原則であり、また実際に使用した感触からもこの値は妥当であると思われる。本システムでは小規模なディスプレイを利用して、比較的大きな PAD を扱うことを可能にしている反面、フレームと内部テキストを分離して2台のディスプレイに表示しているのを見づらいことも事実である。より高解像度で大面積のグラフィック端末を利用する場合には、フレームと内部テキストをマルチウィンド方式により一つの画面に表示するほうが、集中処理によるデータ転送量の減少、ならびに物理的な見やすさという点で有利である。しかし処理を集中することにより端末のグラフィック処理量が増加するので、小型計算機を対象とした場合、現時点では経済性と同時に画面処理時間、所要メモリの増加に問題が残ると思われる。

7. おわりに

各種木構造チャートを使用したプログラムの作成、管理の有効性は誰も認めるところであるが、木構造

チャート開発ツールを利用できる環境はまだまだ限られている。超大型のディスプレイ、高速な画面処理が可能な大型計算機システムによる PAD 処理が理想ではあるが、本システムは大量生産による安価なマイコンを利用して、ミニコン上で実用的な PAD ツールを実現する一つの方法を示したものである。

現在、本システムをさらに発展させ、独立した画面に複数のデバッギング情報を表示し、PAD 上でデバッギングを進めていく PAD テストインタプリタを作成中であり、これについては別の機会に述べたい。

謝辞 本システムの開発を手伝っていただいた鈴木理之氏(慶応義塾大学)、青山均氏(現(株)日本電気)、平林利彦氏(現(株)日本電気)に謝意を表します。

参 考 文 献

- 1) 佐藤匡正：プログラミング用ドキュメンテーション，情報処理，Vol.22，No.5，pp.383-389 (1981).
- 2) 林 秀幸：プログラム制御構造の新しい表現技法；木構造チャートが普及期に，日経コンピュータ，1月9日号，pp.71-83 (1984).
- 3) Pong, M.C. and Ng, N.: PIGS—A System for Programming with Interactive Graphical Support, *Softw. Pract. Exper.*, Vol.13, No. 9, pp.847-855 (1983).
- 4) Workman, D.A.: GRASP; A Software Development System Using D-Charts, *Softw. Pract. Exper.*, Vol.13, No.1, pp.17-32 (1983).
- 5) 米田實男, 田野実裕之：図を用いたプログラム設計における処理記述入力法，情報処理学会講演論文集，pp.603-604 (1984).
- 6) 前沢裕行, 齊藤和正, 森 清三：処理論理図自動作成システムの開発，情報処理学会論文誌，pp.411-412 (1981).
- 7) 高野 彰, 大川 勉, 堀川博史, 上原憲二, 春原 猛：図形定義を用いたプログラム図生成，情報処理学会論文誌，pp.545-546 (1983).
- 8) 郷 信義, 岡田直也, 岸本美紀, 長田芳一, 高橋美智子, 夜久竹夫：木フローチャート記述言語 HICHART の処理系実現について，情報処理学会講演論文集，pp.549-550 (1983).
- 9) Ng, N.: A Graphical Editor for Programming Using Structured Charts, *IEEE COMPCON 79, Spring*, pp.238-243 (1979).
- 10) 前沢裕行, 齊藤和正, 小林正和, 二村良彦：対話型の構造化プログラム設計・製造支援システム (PDL/PAD) の開発，情報処理学会講演論文集，pp.299-300 (1982).
- 11) 二村良彦, 川合敏雄, 堀越 彌, 堤 正義：PAD によるプログラムの設計および作成，情報処理，Vol.21，No.4，pp.259-267 (1980).
- 12) 鈴木理之, 岡田謙一, 横山光男, 北川 節：マイコンによるマルチ・スクリーン PAD 端末，情報処理学会講演論文集，pp.509-510 (1983).
- 13) 岡田謙一, 横山光男, 北川 節：機能分散方式による図式言語入力システムの試作，情報処理学会講演論文集，pp.645-646 (1984).

(昭和59年12月21日受付)

(昭和60年2月21日採録)