

図形によるプログラム入力システム

5J-9

島田啓一郎 藤田昌宏 田中英彦 元岡達
(東京大学工学部)

1. はじめに

現在、プログラミングは命令文を組み合わせた文字列で行なわれている。これは、コンピュータにとっては楽だが、必ずしも人間にとってわかりやすいものではない。そこで人間よりのプログラミング手段として自然言語や平面図形を用いる試みがなされている。平面図形を用いる例としてはフローチャートが有名である。人間はパターン認識能力にすぐれているため、このような平面図形を利用すると流れを理解しやすい。この平面図形によるプログラムの表現のひとつに、構造の見やすさを重視したPAD (Problem Analysis Diagram) がある。

本研究では従来人間の手にたよっていた図形→プログラム言語の変換をコンピュータにやらせるようにして、PADのみが人間との接点となるプログラム入力システムを製作した。

2. PADツールの概要

本研究で製作したPADツールは、PADをそのまま入力・編集・修正できるようにしたPADエディタとPADを実行できるようにしたPADコンパイラからなっている。その構成を図1に示す。本システムの使用により、

人間は画面上にPADによるプログラムを描くだけで、そのプログラムが実行できるようになる。

3. PADエディタ

PADはカラーグラフィックターミナル上から十字カーソルとファンクションキーを用いて直接入力する。プログラムは縦7行横3列をひとつの単位(ページと呼ぶ)として、この組み合わせで表現する。すなわち細かく手続きに分割して、ひとつの手続き(またはメインプログラム)がこの単位(1ページつまり1画面)におさまるようにする。各ページには手続き名に相当するページ名がついて、エディタにおいてはページ名を用いたそのページの呼び出し等もできるようになっている。

表1に作成したPADエディタの主なコマンドを示す。また図2にPADエディタで作成したプログラムの例を示す。(実際は箱ごとに色が違う美しいカラー表示となっていて見やすい)

4. PADコンパイラ

PADの図形情報、すなわちPADエディタにおける内部表現(中間言語)をPascalのソースプログラムに変換するトランスレータと既存のPascalコンパイラより構成される。PAD→Pascal

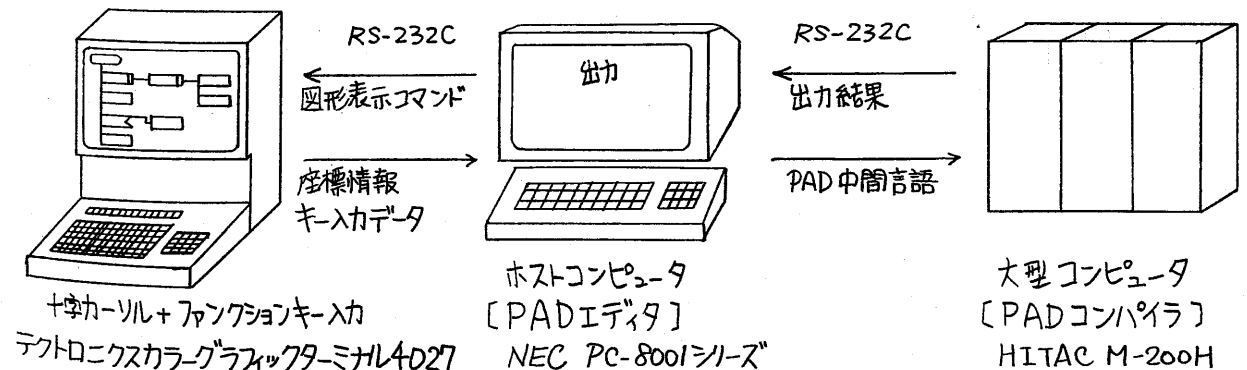


図1. 製作したPADツールの構成

トランスレータはPascalで記述した。トランスレート及びコンパイルは大型計算機上で行なわれる。なおこのPAD→Pascalトランスレータと全く同じものを本研究で作成したPADツールを用いて記述することができた。PADの内部表現(中間言語)は機械的にFORTRANにも変換できるのでPADはFORTRANを経由してコンパイルすることもできる。

5. PADの内部表現

本システムではそれぞれの箱について次の情報をデータとして記憶している。これが中間言語にあたる。

- ①箱の位置
- ②命令の種類
- ③内容
- ④すぐ下にある箱の番号(弟と呼ぶ)
- ⑤自分の右側につながっている箱の中で一番上にある箱の番号(長男と呼ぶ)
- ⑥特にif文の場合はelseに続く箱のうちで一番上にある箱の番号

6. 評価・検討

本システムではひとつの手続きもしくはメインプログラムは必ず1枚の画面におさめるようにした。そのため大きなプログラムにおいては全体をいくつかの手続きに分割しなくてはならない。こうすると構造が見やすくなる反面、やはり不便さもあるので、ハードウェアを改善してもう少し多くの箱が描けるようにしたい。

本システムは従来の文字列言語によるプログラミングに比べて構造が見やすく見た目も美しいプログラミングを可能にするとともに、そのハードコピーはそのまま理解しやすい最終ドキュメントとなりうる。これによりアプリケーションプログラムの管理への道が開ける。また本システムは初心者教育や構造化教育にも向いている。

なお人間と機械との接点として図形を用いたという点で本研究は他の分野(回路図設計、要求仕様書、データフローグラフ等)でも有効な応用が期待

される。

《参考文献》日立製作所中央研究所編「PADによるプログラムの開発」

表1. PADエディタのコマンドの概要
(□はファンクションキーを表わす)

PAGE	n	画面上に新たにnページ目を設定する。		
LIST	n	nページ目を表示する。		
LIST	<ページ名>	そのページ名のページを表示する。		
LIST		今表示されているページをもう一度表示する。		
LIST		今カーソルのある位置の箱が手続き呼び出しならその手続きのページを表示し、手続き宣言ならその手続きを呼んでいるページを表示する。		
CONNECT	□	CONNECTの押された時のカーソルの位置にある箱と□の押された時のカーソルの位置にある箱とを線で結ぶ。		
ERASE	□	同様にして線を消す。		
DELETE		カーソルの位置にある箱を消す。		
SAVE	<ファイル名>	PADのセーブ		
LOAD	<ファイル名>	PADのロード		
GO		PADの実行		
MOVE	□	箱の位置の移動		
IF	WHILE	UNTIL	FOR	READ
WRITE	PROGRAM	PROCEDURE	REMARK	VAR
----- それぞれカーソルのある位置に指定の箱を書く。				

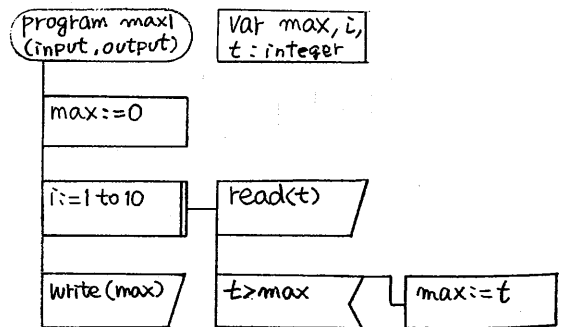


図2. PADツールで作成したプログラムの例