

コンピュータ・プログラムの管理

○ 名古屋大学・工 正 島田 静雄
同 教養部 田嶋 太郎
同 工 正 加藤 祖史
同 工 正 伊藤 義人

1. 問題の背景

コンピュータが、教育・研究に便利に利用されるようになって来た歴史は、名古屋大学においては、大型計算機センタの設立以来、ただだか10年餘に過ぎない。実験的研究と並んで、理論的研究の手段は従来、紙と鉛筆と云はれて来たが、最近ではコンピュータとプログラムと云いかえ得るまでになった。

さて、大学において、卒業研究を含めて、多くの学生がコンピュータと利用して来た。ところが、彼等の作成したプログラムの大部分は、卒業と同時に二度と利用されることがなく消失し、研究の財産として蓄積されることがない。したがって、同じような努力は毎年の卒業研究で繰り返される。しかし、中には良いプログラムが残されることもある。しかし、作成者だけにしか内容が理解できないことが多く、使い勝手が悪く、いつのまにか捨てられていく。

一般論として、プログラムの管理については、世界各国にも共通する悩みであることは、筆者らが個人的に話した多くの学者・研究者が認めている所であった。コンピュータに關した多くの国際会議において、討論が活潑になる諸題の大部分は、大なり小なり、管理の手法に分類して考へることが出来る。筆者らは、三次元の幾何モデルの創成と、その自動化に關する大きなプログラム・パッケージ"GEOMAP"を開発的に開発しているが、昨年より今年始めにかけて、フランス・スエーデンの研究者に"GEOMAP"を紹介する機会があった。この時の経験を含め、プログラムの管理について本稿で概説したい。

2. プログラムの開発者と利用者との相違

日本において、それも工学の分野においては、コンピュータ・プログラムの開発者と利用者との区別はない。しかし、欧米諸国においては、これは明確な職業上の区別を意味している。欧米人のコンピュータに対する感覚は、賢いタイプライタというものである。彼等にとって、タイプライタは日常のありふれた道具の一つであるので、コンピュータを使うということに對して潜在的な遠慮感を持つていない。つまり、誰もが、いつでもコンピュータの利用者になれる。それは、コンピュータの窓口がタイプライタであるからである。タイプライタに賢さを与えるのがプログラムである。当然のことながら、特殊な職業専用であることが理解できよう。

筆者の経験した所では、日本の大々建設会社程度の外国企業内で、自社内でプログラムを作成する人員を持つていない例が多い。設計技術者のすべてが、コンピュータの利用者である。プログラムはすべて購入するのである。プログラムの作成者は、利用者が使い易いように工夫し、また、コンピュータの機種に合わせたプログラムと提供する。プログラムの作成者が利用者に手渡す文書が手引書、つまりマニュアルである。このマニュアルを通じて、作成者と利用者との話し合いが持たれる。

3. プログラム言語について

コンピュータを動かす言語については、プログラムの作成者と利用者とは全く異なつた考へ方をする。利用者の立場に立つとき、日常の道具であるタイプライタを介して、彼等の言語と、彼等の

仕事に関連のある文章作法に便うことを希望する。著者発表されている種々の言語は、このような必然性の結果生まれ、比較的好評であるものが生み出されている。FORTRANを始め、COBOL, PL1, PASCAL, ALGOL, 等々は、いわばコンピュータ時代の第一世代の言語である。この時代では、コンピュータの利用者は、同時にプログラムの作成者であった。コンピュータのハードウェア製作者は、コンピュータを利用してもらう為に、上記のような言語を提模した。

プログラムの開発者と利用者との取組上の分化が進んで現在、ソフトウェア・ビジネスが独立するようになってきた。そして、プログラムの開発者側の立場に立つとき、コンピュータの機種に左右されない共通の言語が重要になってくるのは当然のことである。このような、よりプログラムの専門家を使う言語は、世界的に見て標準のFORTRANに統一されてきている。そして、FORTRANで種々の言語処理のプログラムを作って利用者の便に供している。筆者も、BASIC言語のコンパイラを標準のFORTRANで作成したものを使用している。

4 FORTRANの標準

当面の間、プログラム作成者の使用する言語は、FORTRAN-IVにはほぼ統一されてきている。日本のJIS-FORTRAN 7000水準のことも上である。したがって、JISに準拠してプログラムが書かれていけば、まず殆んど支障はないが、コンピュータの機種による幾つかのハード上の相違が微妙にからむので、プログラム作成上注意する点がある。これを列挙すると、次のような項目になる。

- (1) メモリ 1 word の Bit 数による影響 : 現在、16, 32, 36, 48, 60 等の相違がある。
整数と実数とによって word の使用長さの異なるものは、EQUIVALENCE 文の使い方に注意する。
- (2) Bit の数え方による相違 : 二つの連続した 2 words とつないで、一つの連続した記憶領域と看做するとき、その並べ方に二種ある。Byte-machine でも同じような注意が必要である。
倍精度の実数と単精度の実数とを EQUIVALENCE 文と関連づけたとき、高位と低位のビット配列がどうなるかは注意が必要である。
- (3) 文字コード系 : EBCDIC, ASCII, JIS それぞれ微妙な相違があるので、文字の引用を数値で行なわない。また、mH..... の使い方に注意する。
- (4) 1ワード当りの文字数 : 機種によって、1ワードに記録される文字数が異なる。
- (5) AND, OR, XOR など の機能 : 論理演算の機能については機種によって仕様が異なっている。引数の型が整数に限ることもよく見受けられる。
- (6) ENTRY 文, Abnormal RETURN の使い方は特に注意する。FUNCTION 文中には、ENTRY 文を使用しない。また、SUBROUTINE 中の ENTRY 文は、ARGUMENT を含ませないか、もしくは、同一の ARGUMENT の並びでなければならぬ。
- (7) プログラムの単位をなるべく小さくし、次のような単位にわけろ。
 - a) 入力の処理に関するプログラム : READ 文を持つ。
 - b) 出力の " " " : WRITE 文を持つ。
 - c) 実行・計算を行なうプログラム : 一切の入出力文を持たない。特に注意することは、再入可能となるように設計を扱うことである。
- (8) PLOT など閉ループ・ループは、サブルーチンとして内部的に使用可能。
- (9) 以上が基本的な原則であるが、プログラムを容易くするため種々の工夫が考案されている。これは、プログラム作成者の個性に委ねるところが大きい。一般論として、コメントを多く入れること、変数名に実体に近い命名をすること、などをつけることが出来る。

5. 意を決定のグラフ (Decision Graph)

構造物の設計をコンピュータ・プログラムに作るとき、流れ図の基本は図-1のように表現可能である。ある条件を決定する。計算の結果を許容値と比較する。条件に満たないとき、初期条件を設定し直す。一般に、与えられた条件に対し、唯一つの答が容易に引き出せる問題を P-問題 (positive) といい、答が出し難いもの、また、必ずしも答がないものを N-問題 (negative) 又は逆問題という。

構造解析は P-問題であり、設計は N-問題である。したがって、設計計算をプログラム化する場合、P-問題をプログラム化し、人間が結果を判断するように計画する。このように扱えば、コンピュータと端末機から会話機能を使って処理できるようにしたため、次第に重要な方法となった。

そこで、設計の進め方を分析し、図-2 のようなグラフとすることは容易な決定のグラフと呼ぶことは可能。ここに節点 (path) の始点または終点である。道は、一意的な処理を経て別の節点に進むことができる。二重の節点は、進むべき道が二つ以上ある場であり、どの道を選ぶかの決定は設計者が行なうとする。

さて、図-2 のグラフに元々プログラムが完成したと考えるとき、節点の場はコンピュータが設計者に指示を仰ぐ場であるから、物理的には端末のキーボードで、1ヶ所である。つまり或つかの節点は、すべて同一点であり、すべての道はループして元に戻ると思えることができる。これを模式的に図-3で示す。

そこで、個々の道は互いに独立で、一意的な処理であることを考えれば、これを別々のプログラム単位に作成可能である。そして、どのループと選ぶかの交通整理を行なわせるプログラムがあれば、図-2 に示した設計プログラムが作成可能になる。これを **path Analyser** と呼ぶ。

path Analyser は、一つの独立したプログラムパッケージとして作成されている。そして、或つかのサブルーチン化したプログラムで、図-2 のような処理の流れになるように組立て、実行させる。この際、前節で示したように、プログラムを標準化しておくことにより、**path Analyser** を効果的に利用可能になる。

逆に言えば、**path Analyser** が利用できるように、プログラム単位を単純化することが必要になり、これはプログラム全体の管理に役立つわけである。

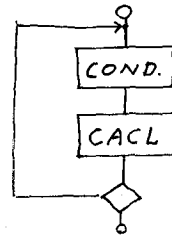


図-1

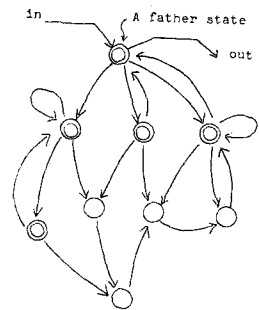


図-2

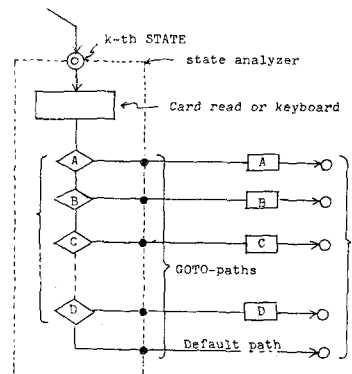


図-3

6 会話処理用言語としての BASIC

最近、パーソナルコンピュータの爆発的な普及に刺激され、BASICの利用者が急増した。BASICは、初心者也容易に理解でき、また、プログラムを書いた、直ちに実行できる便利さと電卓の機能とを合わせ持っている。多くの利用者がBASICに慣れた上では、FORTRANのプログラムをその次に習う、という傾向が見られる。

一方、科学技術計算では、FORTRANが主流であり、内題¹⁾とは各自勝手な書式でデータの入出力を行なっている。そこで、データの扱い、会話機能の仕様はBASIC言語と同じ仕様とするprocessorを開発した。下図はそのテストの一例である。特に注目する点は、DATA LISTの表現方法に工夫が凝らされている。このプログラムのひとつの特徴は、利用者が自分でCOMMAND文を定義できるようにしていることであって、これによって、利用者がFORTRAN SUBROUTINEと、あるいはBASICのCOMMANDのように使用することが出来る。

OK

.... LIST

```

10 READ ITYPE, NUMNP, NUMFL, NUMAT
20 PRINT ITYPE, NUMNP, NUMFL, NUMAT
30 DIM F(NUMAT), PR(NUMAT)
40 READ E
50 READ PR
60 PRINT E
70 PRINT PR
80 READ N
90 PRINT N
100 DIM X(N), Y(N), IUXTYP(N), UX(N), IUXTYP(N), UY(N)
110 READ X
120 PRINT X
130 READ Y
140 PRINT Y
150 READ IUXTYP
160 PRINT IUXTYP
170 READ UX
180 PRINT UX
190 READ IUXTYP
200 PRINT IUXTYP
210 READ UY
220 PRINT UY
230 DATA 3, 4, 5, 6
240 DATA 1 TO 11 STEP 2
250 DATA 1.5 TO 4.0 STEP 0.5
260 DATA 3
270 DATA 3*0.45
280 DATA 10, 20, 30
290 DATA 1, 2, 3
300 DATA 4, 5, 6
310 DATA 7, 8, 9
320 DATA 10, 11, 12
330 DATA 100, 200, 300

```

OK

.... RUN

3	4	5	6	
1.00000	3.00000	5.00000	7.00000	9.00000
1.50000	2.00000	2.50000	3.00000	3.50000
3				4.00000
0.450000	0.450000	0.450000		
10.0000	20.0000	30.0000		
1 2 3				
4.00000	5.00000	6.00000		
7 8 9				
10.0000	11.0000	12.0000		

OK

本研究は、文部省科学研究費補助金、試験研究「陰影・着色と音読した自動作画のソフトウエア開発」に関連して行なった研究成果の一部である。記して感謝の言葉といたす。