

# COBOL 基本語調査\*

関根 智明\*\*

## 1. まえがき

情報処理学会の COBOL 研究会では、はやくから、COBOL を紹介し、その言語やコンパイラの研究にあたっている。その一つの作業として、COBOL が実際の仕事においてどのように使われているか、についても調査をしている。ここでその一部を中間報告する、これがきっかけになって、COBOL に関心をよせる人がさらにふえ、あるいは類似のデータを提供したり、別途に調査結果を発表したりする人があれば幸いである。

計算機のプログラムの要素をかぞえるときには、二つのやりかたがある。一つはソースプログラム上でのみかけの数あるいは記憶装置に格納されている数を見るものである。いま一つは命令の実行された回数をかぞえるものである。前者は言語の外部仕様をきめたり、プログラムの書記的作業、穿孔、コンパイラの構成などを評価するのに利用できるだろう。後者は計算機の実行速度を評価したり、仕事の種類による変動をみたりするのに使える。両者は反復や条件飛越があるので、一般には一致しない、しかし経験的には、両者は（かたよりはるかにせよ）非常によく一致し、また仕事の種類による変動も少ないことが知られている。宣言や指定に関する要素は前者による以外はかぞえようがない。

さて、このような調査には、初期には IBM 650 のものが発表されたし、最近では HIPAC 101 のものが伊東のシンポジウムで報告されている。このほかにも、多くの調査がなされていることとは思うが、公表されたものはあまり見当たらない。

ここで報告するのは前者のそれである。すなわち、COBOL のソースプログラムに各要素の書かれている回数をかぞえたものである。プログラムには一つ(A)除いてすべて、翻訳、実行、検査、手直しのすんだものであって、調査用にとくに作ったものではない。少なくとも一度、ものによっては日常作業に使われている本番のプログラムである。

## 2. 調査結果

全体をとおしてみた印象を述べる。

まず、編集 (Move) 20~50%、計算 (Add 他) 5~15%、飛越 (Go To) 10~20%、条件 (If) 10~20% の 4 種類の命令がきわめてよく使われている。これをまとめると、編集・計算が 40~60%、飛越・条件が 30~40% となる。

次に、COBOL の広範囲な仕様と比し、よく使われる仕様は、ごく小さい部分集合であるらしい。COBOL には非常に特殊な場合に適用される仕様と、どちらを選択しても同じことになる冗長な表現とがある。これらのうち、プログラミングで必要とされ、またプログラマに好まれる表現があることがわかる。そしてそれが案外常識的な、小さい範囲にまとまっている。

### 2.1 Read と Write

Read と Write の比率は 7:11 から 1:26 までにわたっている。入出力ファイルの数は Open/Close から推定するはかないが、入力命令の数はほぼ入力ファイルの数に等しいといつてよさそうであるのにたいし、出力命令の数は出力ファイルの数に比べてかなり多くなっているようにみえる。これは事務計算の特質にも、また COBOL 言語の特質にもよっている。すなわち、入力レコードの種類は原則として読みこんでみなければわからない。読みこんでからいろいろ判別し、それによって異った計算や編集を選択するような仕事とくに報告書作製の場合には多い。ファイル更新の場合にもこの傾向はみられる。

いずれにせよ、出力の場合には見出しレコード (heading) や合計レコード (footing) のように設計の異なったレコードが同一ファイルに混って出力される。

ここでかぞえたのは、ソースプログラムについてだが、実行的には仕事によりこの比率はいろいろに変化し、一概にいいきることとはできないだろう。たとえば、入力レコードの選別などをやっている場合には、実行される入力命令の回数のほうが出力命令の回数より多くなることがあり得る。

Read……At End の書きかたは、ファイルの終りを検出するものであるが、事務計算では必要な書きかたであると考えたので、とくにとりだしてはかぞえなか

\* A Word-Count of COBOL Programs by Tomoharu Sekine (Chairman, COBOL Committee of IPSJ; Keio Gijuku University)

\*\* 慶応義塾大学工学部管理工学科

第1表 資料の概要

プログラム	A	B	C	D	E
仕事	ファイル更新	編集	更新	編集 (領収書)	報告書
計算機 資料提供者 所属	IBM 7044 大駒誠一 慶応大学	IBM 7090 吉村鉄太郎 東芝 (当時)	IBM 7044 園田彰二 NHK	IBM 7044 園田彰二 NHK	IBM 7090 今井悟 IBM
翻訳時間 機械語の数 実行時間 (レコード数)	2472	40 秒			7 分 24000
データ名の最大長 修飾の最大深さ	16 字 2 重	13 字 3 重	1 重	1 重	11 字 2 重
手続きの行数 手続きの命令数 手続きのセクション数 " のパラグラフ数	67 70 1 13	140 185 0 21	720 655 9 91	855 1143 3 183	2500 3510 0 386
命令数/行数 命令数/パラグラフ数 Go/パラグラフ数 Go+Pcr./手続き数	1.0 5.4 0.9 1.2	1.3 8.8 1.2 1.4	0.9 7.2 1.2 1.3	1.3 6.2 1.4 1.7	1.4 9.1 1.3 1.8
入出力百分比 Move 四則 制御	20 30 11 39	12 45 6 37	11 39 15 35	6 27 18 49	1 53 11 35

った。

入力では Read 対 Read Into, 出力では Write 対 Write From の選択的な書きかたがある。これについて特別な説明はあまりなされていないが、文法からいっても、入出力システムの作りかたからいっても、入力では Read が、出力では Write From が標準の書きかたになるものと思われる。とくに単純な Write 命令 (From のない) が、もしソースプログラムに書いてあるとすれば、それは異例なプログラムテクニックをもちいているか、または誤りであろう。この点の調査はここには含めなかった。

## 2.2 Open と Close

Close にはファイルを閉じる場合と、リールを途中で閉じる場合 (Close Reel) とがあるが、後者の用例はなかった。一般に使いみちの少ない書きかたであろう。

ファイルを閉じるときに、テープリールをそのままにしておく (No Rewind), 巻き戻す, 巻きあげてその後参照できなくする (Lock) の三とおりの手当がある。ファイルの保護のためには最後の Lock の書きかたが安全であるが、試験運転のときや、モニタの統制の下にあるときには、かえって厄介の種になる、といわれている。Lock の書きかたが一度も使われていな

いのは、おそらく各計算室での作業の標準がそのように定められているからであって、各プログラマ個人の任意にばかりよるとは思えない。

リールの巻き戻しと放置の選択は、プログラムによって著しくかたよっている (11:0~0:6)。しかし、システムによっては、これらの指定の仕方が COBOL からはずされて、モニタに移されている場合があるので、上記のみかけの数字がそのまま作業の実態を反映しているともいえない。

Close No Rewind を書くと、リールは巻き戻されず、放置されるが、それは一つのリールのなかに複数個のファイルがある場合だと思われる。たとえば、モニタの標準入出力テープを使うようなときには、複数のジョブの入出力ファイルがそれぞれスタックされることがあるので、巻き戻しができなくなる。

Open Input Reversed は逆読みの指定であるが、金物の関係もあり全然使われていない。

Open 対 Close の比率はちょうど 1 対 1 である場合と、Close がやや多い場合とがある。後者は、プログラムの終りかたが幾とおりもあるためかもしれない。

## 2.3 Display と Stop literal

これは計算機が操作員にオンラインでメッセージを渡すもので、とくに後者の場合は、操作員の適当な処

置なり応答なりを待つものである。この使いかたはプログラムによるというよりは、各計算室の習慣なり標準なりによる。

つまり単にメッセージを渡しきりにし、おいてけぼりにするのならば、わざわざオンラインにする意味は少ない。また応答を期待して待つ場合には計算機の時間が無駄になる。したがって、ごく異常な状況について緊急な処置をとるための防護的な命令と考えるべきであろう。

なお、操作員が適当な動作をとるための判断資料とするのに十分な情報を表示しようとするならば、通常のタイプライタの速度ではおそすぎる(高価すぎる)ことも指摘しておこう(200字うつのに20秒はかかる)。

両者をあわせた使用度数は3回から39回まで、百分率は0.1%から5.9%までに変化している。

#### 2.4 Move

Move は COBOL のもっとも基本的で、かつ強力な命令である。これによってデータの移動、表現の変換、位取り、編集ができる。どのプログラムにおいてもいちばん多く、しかも飛びぬけてよく使われている。矢島敬二氏は「computer は jumper である」と述べたことがあるが、そのいいかたを借りるなら、「COBOL は Mover である」となるうか。

FORTRAN や ALGOL におけるデータの移動が、一度に一要素にかぎられるのに反し、COBOL では多くの要素を含んだ構造を一挙に移せることは重要な差違である。この特別な場合として Move Corresponding がある。この書きかたを利用するかどうかにもプログラムの好みがあるらしい。これを利用したプログラムでは当然のことながら、データの名前の修飾が必要になるつまり集団項目中の基本項目には名前の重複があるからそれを単独にとりだすには修飾をする。

Move 命令による処理は、四則演算や比較など多くの仕様に内包されている。そのため Move がどのように翻訳され、目的プログラムになるかは、その COBOL コンパイラの優劣をはっきりさだめる。Move からどのような目的プログラムを発生させるべきか、またそのための翻訳の手続きはどうあるべきかということは、COBOL 研究会でもしばしば討論され、実例もあげられている。

#### 2.5 Add など

算術演算のためには、四則演算のいっいちに対応した命令と、数式を扱う命令とがある。実際に書かれるのは Add が比較的多く、Subtract が少し使われる。

乗算、除算になると、いっいちの命令を書くよりは、Compute のほうにうつるらしい。

Add の機能は Compute のなかに含まれているので、どちらの書きかたをとるかにはプログラマの個人差がある。比率にして80:16から3:380までである。

どのプログラムをとっても、算術演算の命令はこみにしても、Move 命令の数におよばない。事務計算、データ処理では、計算が少なく、編集や条件の判断が多いと一般にいわれているが、これは肯定できそうである。

#### 2.6 Perform

Perform は、サブルーチン呼出し、スイッチングテクニック、反復などによく似た動作をとる。初心者には取付きにくいともいわれるが、案外によく使われている。Perform Varying は添字を変化させた反復動作であるが、プログラムによって使われたり、使われなかったりである。

#### 2.7 Go To と If

Go To には、単純な Go To、Go To Depending、Alter される Go To の3種類があるが、後二者はほとんど使われていない。

Go To と If とをあわせると、百分率は30%から

第2表 手続き命令の百分比

百分比	A	B	C	D	E
Open Input Output	1.4	—	0.3	—	—
Open Input	—	1.1	0.5	0.3	0.0
Open Output	—	0.5	0.6	0.3	0.1
Close	1.4	1.1	0.6	1.0	—
Close Lock	—	—	—	—	—
Close No Rewind	—	0.5	1.2	—	0.2
Close Reel	—	—	—	—	—
Read	4.3	1.1	1.1	0.7	0.0
Write	7.1	4.9	1.7	1.7	0.7
Display	5.7	1.1	4.7	2.0	0.1
Accept	—	1.6	0.2	—	—
Move Corres.	8.6	0.5	—	—	0.1
Move	21.4	44.3	39.5	27.1	52.7
Add	4.3	4.9	12.2	12.9	0.1
Subtract	2.9	—	0.3	1.1	—
Compute	4.3	1.1	2.4	4.3	10.8
Perform	7.1	1.1	4.0	5.4	5.6
Perform Varying	—	1.1	—	—	0.1
Exit	—	—	0.9	—	0.3
Go To	17.1	13.5	16.3	22.7	14.5
Go To Depending	—	—	—	—	—
Alter	—	—	—	—	—
If	10.0	20.5	11.6	19.9	14.5
Stop literal	—	0.5	1.2	0.3	—
Stop Run	1.4	0.5	0.6	0.3	0.0

40%となり、かなり多い。

第3表 手続き命令の使用度数

度 数	A	B	C	D	E
Open Input Output	1	—	2	—	—
Open Input	—	2	3	3	1
Open Output	—	1	4	3	5
Close	1	2	4	11	—
Close Lock	—	—	—	—	—
Close No Rewind	1	1	8	—	6
Close Reel	—	—	—	—	—
Read	3	2	7	8	1
Write	5	9	11	20	26
Display	4	2	31	23	4
Accept	—	3	1	—	—
Move Corres.	6	1	—	—	2
Move	15	82	259	310	1850
Add	3	9	80	147	3
Subtract	2	—	2	13	—
Compute	3	2	16	49	380
Perform	5	2	26	62	197
Perform Varying	—	2	—	—	3
Exit	—	—	6	—	10
Go To	12	25	107	260	510
Go To Depending	—	—	—	—	—
Alter	—	—	—	—	—
If	7	38	76	227	511
Stop literal	—	1	8	4	—
Stop Run	1	1	4	3	1

## 2.8 その他の手続き

手続きの行数と命令の数とを比較すると、ソースプログラムの1行あたり1.3個ほどの命令が書かれていることになる。しかし一つの例外では行数のほうがやや多くなっている。これはドキュメンテーションを考えて見やすい体裁にしたため、このプログラムでセクションの数がもっとも多いのもそのためであろうか。

COBOLの慣用では、プログラムの行数を命令の数だとみなして評価して大きな間違いはないだろう。計算室によっては、1命令を1行におさめる書きかたを標準にしているところがある。

パラグラフ当たりの命令の数は5.4から9.1にまたがっているがこれもかなり一定した値をとるとみてよいだろう。パラグラフの名前はGo ToやPerformで参照されるほかに、読みやすさのためにつけることがある。こうして5~10命令で1パラグラフが作られるといえる。

パラグラフの数にたいするGo To命令の数は、1.2ほどの値をとっている。一つのパラグラフに1.2個のGo To命令が書かれるといってもよいし、一つのパラグラフ名は、1.2個所から参照されるといってもよ

い。これにセクションとPerformとをくわえると、この値はもうすこし大きくなる。

データの名前の字数は最大30字まで許されるが、ここであらわれたのは最大16字であった。なお固有語のうちで最大のものは、Beginning-File-Labelなどの20字である。基本的なものではWorking-Storageなどの15字である。

翻訳速度は資料が不足であるが、1分当たりソース・プログラムのカード500枚から1,000枚という従来報告されている値が、大体500枚/分ぐらいになっているようである。

目的プログラムの語数は、Data Divisionを含むものであり、入出力システムやモニタは含まない。これも1命令当たり5~10語になるらしくみえる。この値はFORTRANなどの値ともほぼ重なっている。

## 2.9 レコードの記述

Data Divisionのいろいろなレコード記述のうちではPictureが飛びぬけて多い。これに含まれるい

第4表 レコードの記述の使用度数

度 数	A	B	度 数	A	B
77	8	8	S. Left	—	1
88 Value	2	5	Occurs	—	2
Picture	59	81	Size	9	—
Value	2	33	Class	1	—
Computational	9	10	レベル構造の深さ	3	4
S. Right	11	7			

ろいろな冗長な書きかたは、おそらくほとんど使われなくなるだろう(ASAのCOBOLでもそのようにきめられている)。

Valueによる定数の設定は、仕事の種類により、ことに報告書作製のプログラムでは多くなるだろう。

UsageはComputationalが、SynchronizedはRightがそれぞれ好んで使われる。この度数はレベル番号77の度数とほぼつりあっているが、ちょうど一致するわけではない。

## 3. あとがき

これはCOBOL研究会での調査をもとにした報告である。この作業には研究会の全員が参加した。標本プログラムの提供は、今井悟(日本アイビーエム)、園田彰二(NHK)、吉村鉄太郎(当時東芝)、大駒誠一(慶応大学)の諸氏からうけた。これらの諸氏ならびに後援してくださった方々や会社に厚く感謝したい。資料の整理と報告の執筆には、大駒誠一、西村彦彦の両氏が助力してくれた。(昭和40年7月20日受付)