

23. 事務処理用簡易言語†

魚 田 勝 臣††

1. はじめに

事務データ処理は企業における計算機応用の約 80% を占めると言われており、重要な応用分野である。特に最近分散処理指向とあいまって、大企業が各部門ごと計算機を導入する傾向にあり、また、オフィス・コンピュータなどの低価格化に呼応して中小、零細企業に到るまで計算機の導入が盛んである。これらのほとんどが事務データ処理（以下、本文では事務におけるデータ処理を事務処理と記す）に使われており、事務の機械化気運に支えられてますます需要が高まると予想されている。これにともなって技術者不足を来しており、ハードウェアの低価格化によるソフトウェアの比重の増大を軽減するためにも、親しみやすく生産性の高い言語システムが望まれている。

事務処理用言語は大別して

手続き向き言語……共通言語として COBOL があり、

そのほか PL/I や BASIC など使用されている、

非手続き向き言語……RPG に代表され、簡易言語とも呼ばれ、各計算機に独自の言語が研究・開発されている、

に分けられる。このうち COBOL などの共通言語については、この大特集の別の解説に譲るとして、ここでは機種間共通性を度外視して構築された事務処理用言語の現状と動向について記述する。

技術計算と比較した場合、事務処理の特質の主なものをおげると次の通りである。

- ① 同一内容の処理が大量の入力データにより反復的に実行される。
- ② 処理の内容は定形化されているが、例外処理もかなり多い。また外的制約条件（制度等）も多い。
- ③ 入出力の方式が多様である。
- ④ 情報と人、物、資金との同期化が必要である。

† Business Processing Languages by Katsuomi UOTA (Mitsubishi Electric Corp. Head Office, Small Business Computer Division).

†† 三菱電機(株) オフィスコンピュータ部

⑤ システムが流動的であり、常に修正を要する。事務処理用言語の構築にあたってこれらの特質から派生する要求事項を満足させなければならない。半面、処理の内容が定形化されているという性質は簡便な言語の構築に利用できる。

つぎに事務処理の機械化のための作業内容から言語に対する要求事項を考察する。機械化の手順は通常図-1のように考えられている。広義のプログラム作成の作業は点線の枠で囲んだ部分であるので、言語としては第一にこの部分の生産性を向上させる方策を考えねばならない。その上でそのほかの作業項目の生産性の向上を図ることに目を向ける必要がある。

本文の対象としている事務処理用言語は各種研究機関、計算機のメーカ、販売会社、ソフトウェア会社やユーザなどで独自に研究開発されてきたため、非常に多数存在し時代とともに変遷している。よって到底すべての言語を調査したうえで動向を紹介することはできない。本文ではこの種の言語の大略の動向を非手続き向き言語を中心に述べ、言語例として国内と外国の各一種を紹介する。国内で研究開発された言語も多数あるが、紙数の都合で、学会の論文誌などで公表されたもので比較的普及度の高いものを1つだけ選んだ。

なお、本文を執筆するにあたり、情報処理学会会誌編集委員会で収集された事務処理用簡易言語に関する

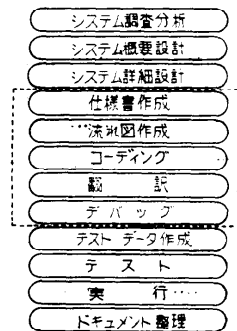


図-1 機械化手順

アンケート調査を参考にした。紙数の都合でここには多くを紹介できないが、別途とりまとめ報告される。

2. 最近の動向

2.1 特徴的な諸概念と最近の動向

(1) 手続き向きと非手続き向きの言語

汎用性の観点からすれば手続き向きが優れていることは言うまでもない。しかし、手続き向き言語は問題の解決にあたって、How (いかにして、すなわち処理の方法) を記述しなければならず、便利さの面で、What (何を、すなわち処理の内容) を記述すれば良い非手続き向き言語より劣る。

非手続き向き言語では簡便さを追求する方法として次の2つのアプローチが採られている。

① 既定論理の設定によって記述項目を簡略化する
前述のように事務処理は定形的なパターン(データ入力、チェック、ファイル更新など)に分類することができる。この特質を利用して、各パターンの処理論理を既定論理として定め、それらについては内容の記述を不要とし、各業務における変化の部分を指定によって補う方法である。

② 言語の表現方法の工夫によって簡便化する
処理内容を計算機に入力する際に、一旦記入用紙にコーディングする方式と直接計算機と対話しながら作業をすすめる方式とが採用されている。

(2) 非手続き向き言語における各種の選択

事務処理がいかに定形的と言えども、それは技術計算との比較論であって、実際には無限の組合せが存在する。それらをすべて網羅しようとする限りなく手続き向き言語に近づき、非手続き向き言語の特長は失われる。したがって、この種の言語では、いろんな意味で汎用性を限定しある種の選択をして簡便さを高めようとする。限定の方法には各種あるが、最近はこの4種がよく採用されている。

① 計算機の規模の選択……大型、中型、オフィス・コンピュータなど機械規模の選択。

② 適用入出力装置の選択……磁気ディスクや磁気テープなどのローカル装置に限定するか CRT なども含めるかの選択。

③ 処理形態の選択……バッチ処理、オンライン処理、インライン処理などの選択。

④ 機能の選択……事務処理全体を対象とするか、たとえば対話管理の部分だけに集中するか、などの機能上の選択。

(3) 言語の表記法

言語の表記法として次の2種が採用されている。
自由形式……何らかの記述子(descriptor)をともなって書かれ、記入位置を定めないもの、空欄埋込形式……専用紙(表形式)の空欄を見出し欄に従って埋めて行く形式。

対話形で入力する言語では自由形式が採用され、記述子が計算機から出力されるので便利である。対話形でない自由形式の場合は記述子の用紙への記入が繁雑である。空欄埋込形式では記入位置は限定されるものの、用紙の見出し欄などに使用者へのコメントを付記できる長所があるので広く採用されている。

(4) プログラム入力

プログラム(非手続き向き言語で書かれたものはプログラムと言わないという意見もある)の計算機への入力方式には一括入力方式と対話形入力方式とがある。入力方式は言語のプロセッサのみに関連するようであるが、非手続き向き言語では表記法の選択とも関連して言語そのものにも影響をおよぼす。

ここで言う一括入力方式は、必ずしもカードやOCRシートなどによる入力方式を指しているのではなく、CRT を利用した場合でも複数の記入項(たとえば用紙に書いた場合の1行分の項目)を一括して入力する場合も含まれる。これに対して、対話形入力方式では基本的に1記入項ごとに計算機と対話して入力が進められる。両者を比較した場合、どちらかと言えば、一括入力方式が熟練者向き、対話形入力方式が初心者向きと言えよう。初心者もいずれ熟練者になるのを考慮し、最近では一つの言語で両方の入力方式を提供する傾向もある。

(5) ほかの言語やシステムとの関連性

① ほかの言語や言語理論などとの結びつき

COBOL、RPG とに関連したものが圧倒的に多い。そのほか FORTRAN や PL/I との関連も見られる。RPG はバッチ処理の既定論理と表記法が参考にされる。COBOL は PICTURE などの個々の概念が利用される場合がある。COBOL も含めて手続き向き言語が非手続き向き言語の一部として利用される場合には、非手続き向き言語で書けない部分を補うために使われるケースが多い。

そのほか、決定表(Decision Table)、構造化プログラミングやデータ・フローの概念を応用した言語も見られ、それぞれの特徴を活かしている。

② ほかのシステムとの結びつき

端末入出力指示書

ページ 03 ID RUNI
12 75 80

連番	テキスト指定	端末制御	行制御	カラム制御	項目名	入力項目	出力形式	一定数		入力項目のチェック	
								項目名	エラー表示	項目名	エラー表示
010	J	タイトル		2	15						
020	J			43	@JDATE						
030	J	ミダシ		4	11						
040	J		X	23	コード	IX	3				TE
050	J			32	トウイサキ	0					
060	J			6	2						
070	J										
080	J	メイウイ		7	3						
090	J		M	7	ショウヒン	IX	3				TL
100	J			14	ヒンメイ	0					
110	J		M	27	スワリヨウ	IX	4				ER@ERRoI
120	J			35	タンカ	0					
130	J			43	キンガク	0					
140	J	トータル		17	39						
150	J			42	コウアイ	0					
160	J										

図-2 PROGRESS の記入例

データ・ベースの普及にともない、それとの関連を持つものや、CRT のスクリーン・ドライバとの関連を持つものなど、ほかのシステムと相補う形の言語もある。

以上、種々の面から事務処理用言語の特有的な諸概念と最近の傾向を述べた。この種の言語は RPG を除いて共通性、互換性がまったくない。逆に、これらの専用言語によって機種ないソフトウェアの特徴を出そうという傾向もあり現在のところ規格化の動きはまったくない。

2.2 事務処理用言語の具体例

前述のように、この種の言語はきわめて多く存在する(ちなみに前記学会のアンケート調査でも 30 余種の言語が寄せられた)。それぞれが特徴を持っており、厳密に優劣を定め難い。ここでは国内および外国での各一言語を紹介する。

(1) PROGRESS.....プログレス

この言語は RPG の流れをくむもので比較的小規模な計算機の事務処理に標的を合わせて開発された非手続き向き言語である。バッチ処理、オンライン処理などのほか CRT (ワーク・ステーション) を使った対話型処理に便利である。図-2 にその記入例を示す。この図から判るように表記法としては空欄埋込方式を採用しており、プログラム入力はバッチ方式と対話方式を持つ。

記入用紙は図-2 も含めて 8 種類あり、記入を助けるコメントが充実しているのが特長である。

モデル業務を PROGRESS と COBOL とで作成した結果、コーディング行数が約 1/2 (バッチ処理の場合)~1/5 (対話型処理の場合) になり、プログラムの作成に要する時間が 1/2~1/3 に減少した。

(2) DGMS

DGMS は "Dialog Generation and Management System" の頭文字を取ったもの。事務処理を対話形管理とデータ・ベース管理とに分類し、本言語によって前者を、DBMS (Data Base Management System) という別の言語によって後者を実現する。すなわち、DGMS は CRT との対話処理だけを受け持つ機能限定型の言語である。図-3 に記入例を示したように、表記法としては空欄埋込方式を採用している。

記入用紙は図-3 に示したものを含め 3 種類で少ないのが特徴である。

DGMS は対話管理を担うもののコーディングの約 25% は FORTRAN で書かれる。

モデル業務による評価では、従来言語 (COBOL を指すと思われる) に比べて 1/2~1/10 のコーディング行数でプログラムでき、所要時間は 1/2 以下に短縮できたと報告されている。なお、本言語はまだ試行段階のようだが対話形管理とデータ・ベース管理との関連で興味深いので採り上げた。

3. 今後の見通し

事務処理の機械化を簡便に実現するために非手続き向き言語の方により大きな期待がかけられると思われる

ID	TYPE	REFERENCE/VALUE	X1	Y1	X2	Y2	lgt	edc	sal	int
1	TEXT	0.1								
2	TEXT	'ACCOUNT INFORMATION'	29	2			18	N	N	1
3	TEXT	'ENTER UPDATE INFORMATION, THEN PRESS ENTER KEY'	4	4			46	N	N	1
4	TEXT	'CUSTOMER ACCOUNT#:'	4	6			18	N	N	1
5	TEXT	'CUSTOMER#:'	4	8			10	N	N	1
6	TEXT	'CUSTOMER LAST NAME:'	4	10			19	N	N	1
7	TEXT	'CUSTOMER FIRST NAME:'	4	12			20	N	N	1
8	TEXT	'ADDRESS'	4	14			7	N	N	1
9	TEXT	'STREET:'	16	15			7	N	N	1
10	TEXT	'CITY:'	16	16			5	N	N	1
11	TEXT	'STATE:'	16	17			4	N	N	1
12	TEXT	'ZIP:'	16	18			4	N	N	1
13	TEXT	'CUSTOMER CREDIT LIMIT:'	4	19			22	N	N	1
14	EDMTC	ANUMR,AN-ENT	25	4			12	Y	N	1
15	EDMTC	CNUMB,CN-ENT	25	8			6	Y	N	1
16	EDTAT	LNAME,LN-ENT	25	10			12	Y	N	1
17	EDTAT	FNAME,FN-ENT	25	12			12	Y	N	1
18	EDTAT	STREET,STRE-ENT	25	15			20	Y	N	1
19	EDTAT	CITY,CI-ENT	25	16			20	Y	N	1
20	EDTAT	STATE,STATE-ENT	25	17			20	Y	N	1
21	EDMTC	ZIP,ZIP-ENT	25	18			3	Y	N	1
22	EDREL	CRLIN,CRL-ENT	25	19			6	Y	N	1
23	TEXT	2.15	5	21						
24	TEXT	1.14	5	22						
25	TEXT	1.16	5	23						
26	TEXT	0.8	49	23						
27	PFK	1								
28	PFK	2								
29	PFK	4								
30	PFK	10								
31	ERMSC	'UPDATE SUCCESSFUL'	2	24			17	N	N	2
32	ERMSC	'UPDATE NOT SUCCESSFUL'	2	24			21	N	N	2
33	ERMSC	1.27	2	24						
34	CURS		25	6						

図-3 DGMS の記入例

る。その研究・開発の方向として次の5点があげられる。

(1) 目的、範囲の拡大

非手続き向き言語の研究の進展にともなって事務処理自体の分析が進行している。その結果、簡便さを失わないで機能範囲を拡大できるようになると思われる。また、プログラム作成だけでなく、図-1に示したシステム設計やドキュメント整理にも便利で、システム設計書やプログラムの仕様書なども兼ねられるような言語方式も開発されるものと思われる。

(2) 対話形言語の進展

言語の学習ができるだけ少なくすむように、計算機との対話によるプログラム作成が現在より更に進歩すると思われる。現状では計算機からの質問に人間が答える方式が採られているが、一方では人間が自然言語に近い文章で計算機に指示する方式も研究されている。この面での進展が期待されるが、問題は人間と計算機の能率低下を防ぐ方策の確立である。したがって生産性の面で優れる現在の一括入力方式に形式言語による対話を加味した方式が根強く残るものと思われる。

(3) オフィス・オートメーション関連機器との結びつき

オフィスの機械化に関する議論が盛んであるが、筆

者は当分の間ファクシミリやワードプロセッサなどの単機機器の進歩と、それらと計算機との結合が主題になると考えている。したがって、事務処理用簡易言語としては、これらの機器とのインタフェースを言語機能とそのプロセッサに持たせることが当面の課題と考えられる。

(4) データ・ベース・システムとの結びつき

特にデータ・ベース用のエンドユーザ言語との結合ないし統合した言語が具体化されよう。事務処理システムの実現のために一つの言語でカバーする方策を探ることが重要である。それが不可能な場合には少なくとも言語上の思想を一貫させるようにしたい。

(5) 規格化は困難

第2章で述べたように現在は規格化の動きすらない。この原因の一つとして事務システムや計算機システムとの関連がきわめて強く、それらが進歩している以上、事務処理用言語も改革せざるを得ないことがあげられる。この状態は当分の間続くだろうから規格化は困難である。しかし研究・開発の方向を定め能率をあげることは重要であるから、衆知を集めて規格として受け入れられるような言語を研究開発することが望まれる。

参考文献

- 1) IBM: IBM System/34 RPG II Reference Manual SC-21-7667, 1977/7.
- 2) 魚田勝臣, 小碓暉雄, 小柴征雄, 北村桂二, 荒井訓男: MELCOM 80 シリーズ・モデル 31 小形電子計算機システム, 三菱電機技報, Vol. 49, No. 5, pp. 378-383 (1975).
- 3) Leavenworth, B.M. and Sammet, J.E.: An Overview of Nonprocedural Languages, ACM SIGPLAN NOTICES, Vol. 9, No. 4, pp. 1-12 (1974).
- 4) Prywes, N.S., Pnueli, A. and Shastry, S.: Use of Nonprocedural Specification Language and Associated Program Generator in Software Development, ACM Transaction on Programming Languages and Systems, Vol. 1, No. 2, pp. 197-217 (1979).
- 5) Hammer, M.M., Howe, W.G. and Wladawsky, I.: An Interactive Business Definition System, ACM SIGPLAN NOTICES, Vol. 9, No. 4, pp. 25-33 (1974).
- 6) 明光英樹, 石田秀信, 大西通雄: データ・フローを応用したプログラム開発ツール, ビジネス・コミュニケーション, Vol. 17, No. 12, pp. 56-

- 61 (1980).
- 7) Leavenworth, B. M.: Nonprocedural Data Processing, *The Computer Journal*, Vol. 20, No. 1, pp. 6-9 (1977).
- 8) Hammer, M., Howe, W. G., Kruskal, V. J. and Wladawsky, I.: A Very High Level Programming Language for Data Processing Applications, *CACM*, Vol. 20, No. 11, pp. 832-840 (1977).
- 9) 魚田勝臣, 溝口徹夫, 小碓暉雄, 富沢研三: インライン処理指向のプログラミング言語, *情報処理学会論文誌*, Vol. 21, No. 6, pp. 433-441 (1980).
- 10) Carlson, E. D. and Metz, W.: Integrating Dialog Management and Data Base Management, *Information Processing 80*, S. H. Lavington (ed.) IFIP 1980, pp. 463-468 (1980).
- 11) Haskell, R. and Harrison, P. C.: System Conventions for Nonprocedural Languages, *The Computer Journal*, Vol. 23, No. 2, pp. 132-141.

(昭和56年2月2日受付)