国産の COBOL コンパイラ*

情報処理学会　COBOL 研究会**

はじめに

情報処理学会 COBOL 研究会では国産電子計算機の COBOL コンパイラの開発状況について調査してきたがひととおりまとまったので報告する.

1960年に制定された COBOL-60 は，1961年に大幅な改訂を加えられたが，米国では直ちにコンパイラの作成が開始された．一方日本では和訳 COBOL[1]出版（1963年）後になっても COBOL コンパイラ作成の動きはなかった.

その理由としては，(1) COBOL の実用性がはっきりわからなかった，(2) ALGOL や FORTRAN に比して，かなり大がかりなコンパイラとなり当時稼動していた小型の国産計算機ではできないのではないかと思われた，(3) メーカは FORTRAN とかアセンブラとかモニタの作成に追われて，COBOL にまで手がまわらなかったなどが考えられる.

1963年の後半になって外国製計算機の COBOL がデータ処理用にどんどん使われだし，しかも，IBM 1401 のような超小型計算機にも COBOL コンパイラがそなえられているのを見てあわてて国内メーカも，COBOL を作成し始めたのであった.

COBOL は始めから共通性とか読みやすさに重点をおいた言語で，冗長度が多く，同じ仕事をさせるのに何通りもプログラムの書き方ができ，それだけコンパイラが複雑になって COBOL を作るのは，だいたい FORTRAN の数倍ないし10倍ぐらいの手間がかかるといわれている．共通性については現状では完全な互換性をもつものはなく，あってもわずかに同じシリーズの計算機で小から大への互換性があるのみである．これはコンパイラ作成者の責任もあるが，COBOL の言語自身が比較に金物固有の collating sequence を使うとか，計算機ごとに結果が異なる SYNCHRONIZED や USAGE などがあって完全な共通言語としては未完成なせいもある.

またすでに

ケツカ　ヲ　ゴウケイラン　ニ　ウツス.

といった，完全に日本語になっているコボルも完成しており公的に認められた日本語 COBOL の仕様の作成が望まれている.

2. コンパイラの概要（第1表）

調査項目1は COBOL コンパイラ作成に要した労力，プログラムステップ数，フエーズの数，コンパイル時間，などを調べたものである．これらのデータだけではコンパイラが良いとか悪いとかいうことはできないが今後コンパイラを作成する際には大いに参考となるだろう.

2.1 完成日（2）

これはともかくも顧客にコンパイラのテープを最初に納入できた時期で，いずれも納入後数多くの虫退治やクレームの処理に大きな労力を使わなければならなかったことはいうまでもない．これらの計算機が発表されたのは，かなり時間の差があるにもかかわらず，COBOL のコンパイラができあがったのは 1965 年の後半から 1966 年前半の1年足らずの間に集中している．これは日本のメーカが始めは COBOL の作成にそれ程積極的ではなかった(?)のが，外国製の COBOL コンパイラが実用されているのをまのあたりに見てあわてていっせいに作り始めたためだろう.

2.2 労　力（3）

これはだいたいの目安で，実際にコーディングやデバッグにたずさわった人達から，大まかな見当で答えてもらったものである（したがって，どちらかといえば定性的なデータである）．一応，一日8時間，1月25日働らいたものとして換算してあるが，プログラムの経験の度合，コンパイラの規模，コンパイラ作製に便利なアセンブラの有無などによって大幅に結果は異なるものである．あまり大規模な COBOL でなければ 3～8 人年位の労力でよいように思われる.

2.3 命令数（4）

中間言語として既製のアセンブラ言語を経由する場合にはアセンブラの命令数は入れてない．いずれの場合も COBOL コンパイラの命令数は1万以上10万未満ということがいえそうである．主記憶が2,000語く

* COBOL Compilers Made in Japan
** COBOL Committee of The Information Processing Society of Japan

第1表 コンパイラの概要

① 計算機	② 完成日	③ 労力	④ 命令数	⑤ ④+③	⑥ フェーズ数	⑦ コンパイル時間 例題1	例題2	⑧ コンパイルに必要な最小構成	⑨ COBOL コンパイラのオブジェクトプログラム	⑩ 備考
(イ) FACOM 230-10	65年11月	人月 30	10,000	333	11 *l*			コア 4Kb ドラム 65Kb PTP, PTR, TW	機械語	カナの名前が使える超小型 COBOL.
(ロ) FACOM 230-30	65年11月	15	6,000	400	5 *p*	分 6.9	分 18.0	コア 32Kb, MT×4 CR, LP, TW	アセンブラ	カナの名前が使える
(ハ) FACOM 230-50	66年3月	60	30,000	500	7 8	2.3	3.6	コア 16KW MT×4 CR, LP, TW	アセンブラ	
(ニ) FACOM 241	65年6月	30.	24,000	800	5 6	4.0		コア 9Kb MT×4 CR, CP, LP	アセンブラ	国産初の COBOL コンパイラ
(ホ) TOSBAC-4300	66年6月	80	10,000	125	2 5			コア 40KC MT×4 CR, LP, TW	アセンブラ	
(ヘ) MELCOM-1530	65年9月	70	12,000	171	3 10	2.0	5.0	コア 8KW MT×4 CR, LP, TW	アセンブラ	
(ト) MELCOM-3100	66年9月	76	14,000	185	3 11			コア 24KC MT×4 CR, LP, TW	アセンブラ	仕様の上では一番大きなコンパイラ
(チ) NEAC-2200	65年11月	188	72,500	385	12 40	1.2	1.9	コア 16KC MT×4 CRP, LP	機械語	カナの名前が使える.
(リ) NEAC-2200	66年3月	200	77,000	385	15 42			コア 16KC MT×4 CRP, LP	機械語	全部日本語で書く.

Kb: 1000 バイト, KW: 1000 語, KC: 1000 字, MT: 磁気テープ, CR: カード読取装置, CP: カード穿孔装置, LP: 印刷装置, TW: タイプライタ, CRP: カード読取穿孔装置

らいで磁気テープやディスクなどの補助記憶装置のない小型の計算機で小規模の ALGOL が実用になっていたが, COBOL は言語の性質上どんなに小規模のものでも数回パスし, 1万以上の命令が必要になるようである. (5) は1人1ヵ月で何ステップくらいコーディングできるかということであるが, いずれも3桁である. コーディングだけなら1日に数百ステップを書くことも可能であるが, 仕様をきめて, 流れ図を書き, コーディングし, デバッグするとなると, 分業でするとしても1ヵ月平均数百ステップというのはけっして少ない数字ではないだろう.

2.4 フェーズ (6)

ロジカルなフェーズ (*l*) とフィジカルなフェーズ (*p*) とにわけた. 前者はソースプログラムを何回走査するかということで, 後者はコンパイラのプログラムが何回入れかわるかということである. コンパイラの作り方によって大きく異なっている.

2.5 コンパイル速度 (7)

例題を作ってコンパイル時間を測定したもの. 例題1は水道料金計算で全部でカード70枚, 例題2はファイルの更新プログラムで約170枚である. プログラムの大きさによってコンパイルの時間が大きく異なるものとそれほど変化しないものとある.

2.6 最小構成 (8)

(イ) を除いて全部のコンパイラが最低磁気テープ4台を要求している. 磁気テープ4台が COBOL コンパイルに最適とはいえないが, 営業政策上できるだけ小規模の構成でもコンパイルできるようにとすると必然的に磁気テープ4台となるようである. あるいは, COBOL のコンパイルには他に補助記憶装置がない場合には磁気テープ4台以上ないとできないといえるかも知れない. あるいは3台以下でできるとしても大幅に効率が悪くなるということは確実である.

2.7 オブジェクトプログラム (9)

COBOL コンパイラはソースプログラムをいったんアセンブラ言語におとすものの方が多い. これはアセンブラがいわゆる IOCS や, 使いやすいマクロ命令がある場合にはコンパイラは簡単になるが, コンパイルの速度は遅くなる. アセンブラがただ機械語の命令や番地を記号で書くだけといった単純な場合には, アセンブラを経由させても直接機械語の命令を作り出してもそれほど手間は変わらないのでアセンブラ経由でない方が良い.

アセンブラを経由する場合には, コンパイラが作ったほとんど誤りのないアセンブラ言語のプログラムを人間が直接作ったアセンブラ言語のプログラムと同じようにチェックしながら読み込むもので非常に無駄をしている. アセンブラを経由させる場合でも, アセンブラとコンパイラ間の緊密な連絡をとり, COBOL の場合には通常のアセンブルの最初のフェーズぐらいは

第2表 開 発 要 素

要素＼計算機	(イ) FACOM 230-10	(ロ) FACOM 230-30	(ハ) FACOM 230-50	(ニ) FACOM 241	(ホ) TOSBAC 4300	(ヘ) MELCOM 1530	(ト) MELCOM 3100	(チ) NEAC 2200	(リ) NEAC 2200
IDENTIFICATION DIVISION									
DATE-COMPILED	×	×	○	×	○	○	○	○	○
ENVIRONMENT DIVISION									
MEMORY SIZE (Source)	○	×	×	○	×	○	○	×	×
RENAMING	×	×	○	×	○	○	○	○	×
MULTIPLE REEL	×	○	○	×	○	○	○	○	○
RESERVE ALTERNATE AREA	×	×	○	○	○	×	○	○	○
RERUN	×	×	○	×	○	×	○	○	×
DATA DIVISION									
FILE CONTAINS	×	×	×	×	×	×	×	○	○
SEQUENCED ON	×	×	×	×	×	×	×	○	○
CLASS	×	×	○	×	×	○	○	○	×
88（レベル番号）	×	×	○	○	○	○	○	○	×
Editing clause	×	×	×	×	×	○	○	○	×
SYNCHRONIZED	×	×	○	○	×	×	×	○	×
USAGE	×	○	○	○	×	○	○	○	×
RENAMES	×	×	×	×	×	×	×	×	×
REDEFINES	×	○	○	×	×	○	○	○	○
COPY	×	×	×	×	×	○	○	×	×
PROCEDURE DIVISION									
ADD SUBTRACT MULTIPLY DIVIDE	×	○	○	○	○	○	○	○	○
COMPUTE	○	×	○	○	×	○	○	×	×
EXAMINE	×	×	○	×	×	○	○	○	○
MOVE CORRESPONDING	×	×	○	○	×	○	○	○	×
USE	×	×	○	×	×	○	○	○	○
INCLUDE	×	×	×	×	×	×	×	○	×
その他									
演算項目の最大桁数	24	18	18	14	18	15	15	64	64
添字の次元	1	2	3	1	3	3	3	2	2
修 飾	×	×	○	○	○	○	○	○	×

省略するようにすべきである.

3. 開発要素（第2表）

調査項目2では COBOL-61 で書ける書き方のうち，各計算機のコンパイラで受け入れてくるものについて調査した．調査はたとえば，GO TO ひとつをとっても行き先を省略したものが書けるかとか GO TO DEPENDING の書き方ができるかなど，きわめて詳細にわたっており，EXAMINE だけでも 11 項目になる．ここではその全部はのせられないので，一部を抜粋したものが第2表である.

COBOL-61 には必須 COBOL と選択 COBOL の区別がある．その後 COMPACT COBOL[1], COBOL-61 Extended[2] などが発表され現在の最新版は COBOL-65[3] であるが，ここに示されている COBOL はいずれも，そのどれでもない．多くは COMPACT COBOL とか必須 COBOL を一応の基準としているが，計算機の特性とか作成の面倒さなどから適宜変更している.

3.1 IDENTIFICATION DIVISION

これはオブジェクトプログラムには全く影響を与えず，コンパイラはほとんど，読みとばすだけで済むので，いずれも制限なしになっている．ただ DATE-COMPILED だけが必須 COBOL に入っていないので省略しているものが 2，3 ある.

3.2 ENVIRONMENT DIVISION

Source Computer の記憶容量が指定できるものもあるが，記憶容量の多少によってコンパイルの方法を変えているものはない．また Source Computer と異なる Object Computer を指定できる COBOL もない.

RENAMING は有効な書き方であるが，修飾がで

きないと意味ないので RENAMING ができるものは修飾が可能になっている．MULTIPLE REEL と RESERVE ALTERNATE AREA はその計算機システムが持っているいわゆる IOCS の機能によっているようである．COBOL-61 の文法書では少なくともひととおりの RERUN（再運転）の方法を与えなくてはならないことになっているが RERUN が全くできないものもいくつかある．

3.3 DATA DIVISION

File の指定の項で FILE CONTAINS とか SEQUENCED ON のようにオブジェクトプログラムには無関係の説明文は（チ）と（リ）を除いて省略してしまっている．

Data の指定の項で PICTURE があれば必ずしも必要のない Editing Clause, CLASS などはほとんどが省略している．SYNCHRONIZED とか USAGE のように計算機によって必要ないものは，削除してある．RENAMES（再命名）はどこのコンパイラも採用していない．またプログラムを書くとき非常に有効な COPY は（ヘ）と（ト）だけしか使えない．

REDEFINES（再定義）のないものがあるがこれがないとプログラムを書くのに支障をきたすことがあると思われる．CONSTANT SECTION はないのさえあるし，あっても WORKING-STORAGE SECTION と全く同じあつかいをするものが多い．

3.4 PROCEDURE DIVISION

演算用動詞で ADD, SUBTRACT, MULTIPLY, DIVIDE の四つが必須 COBOL で COMPUTE が選択 COBOL なので COMPUTE（すなわち数式）のないものが多いが，逆に，演算用動詞は COMPUTE だけというものもある．COBOL の演算項目は最大 18 桁ということになっているが 14 桁から 64 桁まで，さまざまである．

EXAMINE(桁調べ), MOVE CORRESPONDING（名前による対応）は作るのが大変なので書けるようになっているのは少なく，DECLARATIVES USE や INCLUDE（ライブラリの引用）の使えるのもわずかであり，DEFINE（マクロの定義）にいたってはまったくない．もっとも外国製の大型コンパイラでも，DEFINE が使えるのは見たことがない．

この他，クラスのテスト，条件名のテストもコンパイラ作成の面倒さあるいは金物の性質上厳密にはテストできないなどの理由で省略しているものが多い．

4. その他

コンパイルの方法については細かい調査は行なわなかったが，（チ）と（リ）は他とかなり異なっているので簡単に報告すると，この二つはいずれももとは一つで，すでに完成していたほぼ COMPACT COBOL 程度のコンパイラにつけ加えてカナコボル，日本語コボルとしたものである．すなわち，本来 RENAMING や修飾のできなかったものをパスの数を増やして，修飾してある名前やカナの名前を英字の一意の名前につけかえるなどの事前処理によって COMPACT COBOL に変換し，これを従来のコンパイラに引きつぐというやり方である．これはコンパイルの能率という点では問題があるが，ともかくも動くコンパイラを早く作るという意味でひとつの方法である．

また，COBOL ができているのは固定語長計算機（word machine）よりも可変語長計算機（character machine）に多いのは，COBOL のとりあつかうデータの最小単位が 1 文字なので，語を分割する必要のない後者向きでしかも SYNCHRONIZED は無用だし，機械によっては USAGE も無関係になるなどコンパイラを作りやすいためであろう．

5. むすび

本調査項目の中に，コンパイラ作成の費用，デバッグに使用した計算機時間も入っていたが，具体的な数字に関する回答は得られなかった．この種のデータが精確に得られていないのか，わかっていても公表できないのかは不明である．

COBOL コンパイラ作成にあたって，各社とも一様に困ったことは，COBOL 文法にあいまいな点が多く，どう解釈して良いかわからなかったことである．（だいたいは計算機に都合のよい方をとって作成したようであるが）しかし，これら，不明な点の大半は，COBOL-65[3] で明確になった．

またエラーメッセージについては，番号で出すもの，英文，ローマ字，カナといろいろあるが，必ずしもそれが適切でない，あるいは見当違いであるという使用者からの非難の声が多い．しかし実際問題として

```
MOVE ALPHA TO BETA ADP GAMMA
DELTA.
```

では，「ADP という data-name はない」とか，「MOVE に何らかの文法上の誤りがある」といった意味のエラーメッセージが出て，「ADP は ADD の書

き間違い？」というようなメッセージはそう簡単には出ないだろう．

計算機そのものもそうであるが，コンパイラが最初にユーザに納入されたときは「虫」が多く，なかなか使えるようにならない．作成者側はできるだけのテストはしているはずなのに，いったん実用に供すると，予期しない誤りがかなり出てくるのが普通である．そこで，COBOL とか ALGOL とかいった共通言語にはコンパイラ検査用のすなおな例題，いやらしい例題いろいろな誤りのある例題などの標準プログラムがたくさん用意されていて，これらをコンパイルして所定の結果が得られれば，そのコンパイラはほぼ大丈夫というようなものがほしいものである．

最後に，進んでこの面倒な調査に協力をして下さった各メーカのかたがたに感謝します．本稿の執筆は，幹事大駒誠一が担当した．

参考文献

1) 情報処理学会：和訳 COBOL, 1963 年
2) アメリカ国防総省：COBOL 61 Extended. 1963 年
3) アメリカ国防総省：COBOL Edition 1965, 1965 年
4) CODASYL：COMPACT COBOL CIB＃5, Oct. 1964
5) 西村恕彦：日本語による COBOL，情報処理 8 巻，3 号（1967）
6) 西村恕彦：ECMA における COBOL の活動，情報処理 8 巻，3 号（1967）

（昭和 42 年 2 月 7 日受付）