
 報 告

ECMA の COBOL 委員会に出席して**

大 駒 誠 一**

1. はじめに

ECMA の COBOL 委員会に出席し、日本の COBOL の標準化がとて遅れているのを感じた。

ECMA (European Computer Manufacturers Association, 欧州電子計算機製造業者協会) というのは、1960年に計算機関係の標準の作成と普及を主たる目的として組織された欧州のメーカーの団体で、現在正会員 17 社、準会員 11 社で構成され、本部はスイスのジュネーブにある。

この中に、COBOL, FORTRAN, PL/I, ディスクパック, 磁気テープカセットなど 19 の委員会 (Technical Working Committee, 普通は COBOL を TC 6, FORTRAN を TC 10 などのように番号で呼ぶ) があって、これまでに FORTRAN, Subset of ALGOL 60—ECMALGOL, 流れ図など ECMA 規格を作っているが COBOL 規格はない。この 19 の委員会のうち ALGOL, 流れ図, 数値制御など 5 つはその作業を完了してすでに解散している。

COBOL 委員会 (TC 6) は J. Bourgain (Honeywell, フランス) 氏が議長, F. Chasles (IBM, フランス) 夫人が副議長で、他に 10 人の委員がいる。この議長と副議長は今年から交代した。

この ECMA の COBOL 委員会と情報処理学会の中の規格委員会 COBOL 作業グループとはこれまでに公式、非公式の接触を保ってきたが、いつも文書によるものだけだったので、具体的な活動状況を知りたいと思い、筆者の滞欧中にオブザーバーとして出席したい旨を申し入れておいたのが実現したのである。

出席してみても、一番強く感じたのが、冒頭の COBOL の規格化については日本は遅れているということである。ただこれは、誰かの怠慢によるのではなくて、日

本の地理的条件や日本語の問題など避けたい原因によるものであらうと思われた。

2. COBOL 規格化の現状

米国の COBOL 規格¹⁾ が 1968 年に制定されたが、現在米国ではこの改訂案を作成中で、ECMA TC 6 が積極的に協力している。ECMA では米国の COBOL 規格をそのまま採用することにきまっているので、ECMA の COBOL 規格というのではないわけである。そのかわり、米国規格作成の段階で ECMA が協力することになっている。

いま米国で COBOL 規格の改訂を担当しているのは米国規格協会 (ANSI) の中の X3J4 というグループで、すでに

Working Document for American National Standard COBOL

という文書を二つ出して、1971 年 12 月発行のものを表紙が赤いので通称 “Red Book” と呼び、1972 年 2 月発行のものを表紙が黄色なので “Yellow Book” と呼んでいる。この両方とも表紙に「これは X3J4 内部の作業用の文書であって、外部に配布するためのものではない」ということわり書きがついている。ECMA の委員は X3J4 の会議に出席した委員が持ち帰っているのではほとんどの人が持っていた。筆者は持っていないが会議中にパラパラと見せてもらったかぎりでは、デバッグとデータ通信の機能が追加され、CODASYL COBOL と相違のあった EXAMINE, NOTE, REMARKS が INSPECT, 注記行に変わっている。データベース、非同期処理はまだ入っていない。

そしてこれが 1972 年中には米国規格 ANS COBOL になる予定と聞いている。米国規格の改訂作業が始まっていることは、この会議のすこし前に知ったが²⁾、もうまとまった文書が出るほどに進行しているのを知ってびっくりしたわけである。

日本の JIS COBOL³⁾ が制定されるのは今年の 8 月

* COBOL Working Committee of ECMA, by Seiichi Okoma

** 慶応義塾大学工学部、在スイス

ごろと聞いているが、これは現在の米国規格と同じ ISO 国際推薦規格案をもとにして作ったもので、制定されたとたんに世界の規格から見ると古いものになってしまうわけである。今後ソフトウェアの国際的流通は多くなる一方なので、日本独自の規格は言語や習慣など日本の特殊な事情をもつものを除いては意味がないので、JIS はいつも国際規格と同じというのがよいと思われる。したがって ECMA と同じように COBOL の JIS は米国規格をそのまま採用するときまれば、規格の面では世界の場合からの遅れは解消できるわけである。

しかしながら、いまのところ日本語で書いてない JIS 規格というのは認められないので、翻訳して何段階の審査を経て JIS として正式に制定されるまでに数年の遅れがいつも生じてしまうのである。この言語の問題ではフランスも同じ悩みをかかえていると聞いていた。ECMA がこれができるのはメーカーの集まりであることと、ECMA の公用語が英語だけとしまっているためではないかと思われる。

いま世界で電子計算機を作っているのは、共産圏を除けば米国と欧州とアジアの日本だけといえるだろう。前二者は規格作成の段階で緊密な連絡をとっているのである。ひとり日本だけが COBOL 規格作成という場からとり残されてつんば枝敷（いつもほとんどできあがってから内容がわかる）におかれているわけなので、日本ももっと規格作成の段階から積極的に参加すべきだと思う。しかしながら実際問題としては困難な点がたくさんある。まず米国の規格作成に日本からのこの出かけていってそう簡単に口が出せるとは思えないし、ECMA はメーカーの集まりであるから情報処理学会の COBOL 委員会は正式メンバーにはなれないし、一年に数回あるいは十数回も米国や欧州で開かれる会議に日本からいちいち出かけるのは、出席する人間も費用もたいへんである。何かうまい手はないだろうか。

さて、いままで米国規格が世界の規格であるかのような書き方をしてきたが、米国規格はあくまでも米国内だけの規格で、ECMA がそれを ECMA の規格として認めているだけで、国際規格ではないのである。

正式な国際規格は ISO (International Organization for Standardization, 国際標準化機構) 規格である ISO には日本もちろん加盟しているので、この ISO 規格ができるのを待って JIS 規格を作ればよいのであるが、この ISO の動きがまたおそろしくのろい。

米国規格で用が足りているからだろうか。

1967 年パリでの ISO/TC 97/SC 5 の会議では「COBOL の ISO 規格は米国規格と同一であることが望ましい」と決議されて、その後実質的には米国規格と同じものが ISO 推薦規格案として提出された。そして 1970 年 11 月に加盟国の郵便投票が行なわれ、反対は 1 カ国ぐらいしかなかったのにまだ ISO 規格または ISO 推薦規格という正式な文書は出ていない。ただ最近になって「ISO Recommendation 1989」というのが発行されたが、まだこれも Proof (校正刷) となっている。したがって、形式的には日本の COBOL の標準化は国際標準からは遅れていないわけである。しかし、日本を除く全部（といってもよいと思う）が米国規格を使っているので、日本も遅れをとらないように何か考えなければいけないのではないだろうか。

現実問題を考えてみると、いま COBOL の翻訳ルーチンを作ろうというとき、たとえば桁調べ用の動詞として、JIS にあわせて EXAMINE の方を採用するか、あるいは将来の変更しにそなえて米国規格の INSPECT (いずれ JIS も変更されるだろう) の方を採用するか、または両方ともコンパイルできるようにしておくかの選択をしなければならないわけである。

3. COBOL の改訂提案

さて会議の方は、COBOL Expert meeting, ANSI X3J4, ECMA のデータベースの会議その他の報告のあと、ANSI の改訂案と ECMA の希望とが異なっている点の処置、CODASYL に出されている提案の検討などが行なわれた。

ここでは、このうち二、三のおもしろい提案を述べるにとどめる。

1) LINAGE 句の変更

一般形式

LINAGE IS $\left\{ \begin{array}{l} \text{データ名-1} \\ \text{整数-1} \end{array} \right\}$ LINES

$\left[\text{, WITH FOOTING AT } \left\{ \begin{array}{l} \text{データ名-2} \\ \text{整数-2} \end{array} \right\} \right]$

$\left[\text{, LINES AT TOP } \left\{ \begin{array}{l} \text{データ名-3} \\ \text{整数-3} \end{array} \right\} \right]$

$\left[\text{, LINES AT BOTTOM } \left\{ \begin{array}{l} \text{データ名-4} \\ \text{整数-4} \end{array} \right\} \right]$

これはファイル記述に

LINAGE IS a FOOTING b TOP c
BOTTOM d

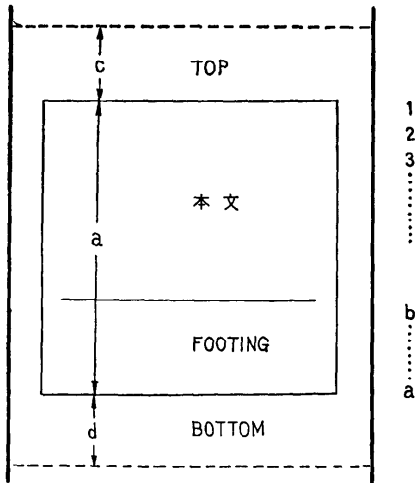


図1 1ページの体裁

のように書いて、図1のように報告書機能を使わないときの1ページの体裁をきめる。

1ページの大きさは $a+c+d$ である。FOOTING から BOTTOM までの間に脚書きが書け、FOOTING の位置は WRITE 命令の END-OF-PAGE 句で検出する。TOP の上、BOTTOM の下は自動的にスキップされる。

2) COPY 句の変更

一般形式

```
COPY 登録名 [ { OF } 呼び名 ]
          [ { IN } ]

REPLACING { (Pseudo-Text-1)
             語-1
             定数-1
             一意名-1 }

BY { (Pseudo-Text-2)
     語-2
     定数-2
     一意名-2 } ...
```

これは Pseudo-Text というものを導入して、プログラムのどんな部分でも置き換えることができるようにしようというものである。Pseudo-Text というのはめんどうな定義があるが、簡単にいえば、語や定数の並びを次のように二重の引用符で囲んで示す。

```
" " A OF B " "
" " MOVE ALL * " " "
```

そしてたとえば次のように書く。

```
COPY TOROKU
REPLACING " " IS EQUAL TO "
```

" BY " " = " "

文字定数の中以外の区切りとしての空白は数を問題にしない。または

REPLACING " ", " BY " " " "

と書いて複写の際にコンマを消去できる。Pseudo-Text-2 の方の中味は空でもよい。いずれも従来の COPY 句ではできなかったものである。

しかし、注記行も変えるのか行のつながりをどうするか、何かほかに変なことが起こりそうな気がする批判的な空気が多かった。

3) 前処理

```
{ REPLACE [ { BEFORE }
            { AFTER } ] COPYING }

{ (Pseudo-Text-1)
  語-1
  定数-1 }

BY { (Pseudo-Text-2) } ...
   語-2
   定数-2 }
```

これは見出し部の前に置いて、前処理ルーチンにより、プログラムの一部を変更しようというものである。BEFORE か AFTER で COPY 句による複写をする前に置き換えるか、後で置き換えるのかを指定する。Pseudo-Text は前述のものと同じである。

一度これを置き換えたものを (BEFORE)、複写した後また必要なら置き換える (AFTER) のか、いずれにしても翻訳ルーチンの負担が大きくなりすぎるという意見が多かった。

この他、宣言部分の節にも優先番号をつけて区分化する案とか MOVE, COMPUTE などすべてオペランドは少なくとも 10 個は書けなければならないとしようとかたくさんの提案、問題点が検討された。

このような様子であるから COBOL の機能はまだまだどんどん増えていきそうな気配である。

最後にこの会議に出席するにあたって電子技術総合研究所の西村恕彦氏と ECMA の J. Besse 氏にお世話になりました、謝意を表します。

参考文献

- 1) "USA Standard COBOL", X3.23-1968, ANSI.
- 2) 西村恕彦: "コボル短信(9)", 情報処理, Vol. 13, No. 1, p. 59 (1972).
- 3) "日本工業規格案電子計算機プログラム言語 COBOL," 日本規格協会, 1971年.

(昭和47年4月10日受付)