COBOLについて*

渡　辺　昭　雄**

事務計算用の共通言語であります COBOL の仕様が，米国の国防総省，ほかの連邦政府諸機関，計算機メーカおよび産業界におけるユーザたちの共同作業の成果としてはじめて出版されましたのは，1960 年4月，ちょうど今から4年まえのことでした．

当時米国におきまして，電子計算機を用いてデータ処理をしておりますユーザ達は，いずれも次のような結論に達しました．それは，電子計算機の事務方面への広範囲の応用を計るためには，プログラムの作成と修正維持に要する時間と費用とをなるべく少なくしなければならないということでして，これがやがて一つの声になって先に申したような共同作業にまで発展したのです．

COBOL の仕様が決まり出版されますと，米国の各計算機メーカは自社の計算機のための COBOL Compiler の製作に着手し，若干の遅れはありましたが，大体予定どおり 1962 年中には完成いたしました．当初の計画では，主要な計算機はすべて含みまして，約 30 機種のための Compiler を作成するということでしたが，実際に完成いたしましたのは，途中で開発されました機種も含めまして，50 機種を上回ったように聞いております．

さて，こうして完成しました COBOL Compiler は，米国におきましてそれぞれのユーザの手で試用され，つづいて実用化の時代にはいっていったのですが，米国における最近の COBOL の使用状況は，と申しますと，実はあまりかんばしくないのです．COBOL と兄弟格にある ALGOL あるいは FORTRAN が，米国におきましては科学技術計算のプログラム言語としては常識になっておりますのに，ひとり COBOL だけは，まだ事務計算のプログラムは 100% COBOL で書く，と宣言する大手ユーザもないままに，1963 年一年間は COBOL に対する批判の嵐の中に立たされていた，といった状態でした．

一方日本におきましては，米国の雑誌や学会誌によって，米国でのその動向はつぎつぎ伝えられたのですが，それらは，大学，研究所，産業界のごく一部の人達に対する情報にすぎませんで，実際に COBOL を使えるような計算機を持っているユーザが理解できるかたちの説明が日本語でなされましたのは，1961 年の7月に出ました学会誌情報処理の記事が最初でした．その後，1962 年6月には，某外国機メーカが COBOL 講習会を開催したのを皮切りに，現在では主要な国産機メーカも含めまして，各メーカいずれも3日間ぐらいの，COBOL の講習会を開催しております．COBOL Compiler の方も，まず外国機の金物について日本にはいってきたのですが，メーカも強力に COBOL の使用をすすめもしないし，ユーザの方も積極的に COBOL を使ってみようとはしなかったため，ごく一部で実験的に使われたにとどまりました．

ところが 1963 年の夏ごろから，日本において，非常に興味ある現象が起きてきました．その第一は，一足とびにある大型機を設置したあるユーザが，事務計算のプログラムはすべて COBOL で書くという方針を打ちだしたのです．私も非常に興味をもってその経過を見守っていたのですが，稼動を開始して数ヵ月になりますが，別段，何の支障があったということも聞いていないばかりか，そのユーザでは昨年11月の衆議院議員選挙の開票速報のプログラムを COBOL で書かれ，そして，立派に計算機を動かされたのを私はテレビで見たのです．このユーザの場合は，設置されました計算機に用意されていたプログラム言語が，アッセンブラとその上は COBOL しかなく，それに優秀なプログラム要員はいるけれども，処理対称業務の量にくらべると要員の絶対数は不足しているから，どちらかといえばてっとり早くプログラムの組める COBOL を選ばれた，と聞いております．

興味ある現象の第二は，これは私どもの例をあげて恐縮なのですが，私どもの会社の場合です．私どもの会社のデータセンタでは，これまで機械語と COBOL とのちょうど中間ぐらいの水準にある Autocoder と

---

* Trends of COBOL in Japan, by Akio Watanabe (Datacenter, Toyo Kogyo Co., LTD.)
情報処理学会関西支部設立記念講演会で講演

** 東洋工業株式会社

いう言語を事務計算の場合の標準プログラム言語と定め，その上にたって分業方法や運用ルールを決めて，百数十人の計算要員の統制をとっております．

事務計算の場合 Autocoder を標準言語と定めましたのは，今使用している計算機を設置する際に，Program の作成と修正維持に要する時間と費用とをなるべく少なくするには，COBOL のまだ完成されていなかった当時としては，機械語と COBOL とのちょうど中間にあるという Autocoder を使うのが最良である，との結論に達したからでして，さらに，Autocoder を使用するからには現在のように分業化するのが最適と考えたのです．こうして，私どもデータセンタを運営しておりますうちに，現在使っております計算機の COBOL Compiler ができてきましたので，早速試用もしてみましたし，また COBOL を標準言語としたとき，データセンタの運用システムはどのように変えるべきか，ということも検討いたしました．従来のシステムと，COBOL を標準言語とするシステムとの利害得失を比較いたしました結果，企業におけるデータセンタ運用上は COBOL の方があらゆる点で優利との確信を得たのですが，ただひとつ非常に基本的なことながら，事務計算のプログラムが機械語で書ければ COBOL でも書くことが可能か，もちろん，技術的に書くことが可能というだけでなく計算実行時間も含めての実用的な意味においても可能か，という疑問だけは心の底に残っておりました．

この点に関しましては，理論的な考察よりも，ある期間を定めて実際に COBOL を使って相当量のプログラムを組んでみれば経験的に確かめられることなのですが，新しいプログラム言語の評価というものは，4カ月ぐらいその言語ばかりを使ってみないと正確にはいえない，その4カ月というのは，新しいプログラム言語になれるということと，それ相応の新しい分業組織，運用ルールにおいて考えてみるための期間なのですが，プログラミングに追われるデータセンタではそのために一人戦列から離すということもできず，そのうちにどこかで使うところがでてくるからその様子を見ようと，一部の者が運用システムは従来のままでコーディングのみ COBOL でやってみて，Compile のための時間も変らないし，計算実行時の速度も変わらない，どうやら間違いはなさそうだ，との見とおしをたてていたにすぎなかったのです．

以上述べました二つの例，先のはそこに COBOL しかなかったので使ったところ，何の支障もないというもの，後のは，従来の最良と思われるプログラム言語と比較して少なくとも運用上は COBOL がよい，と判断したもの，これが非常に興味ある現象であるというゆえんは，もし，この二つの判断が正しいとするならば，米国の学会誌や雑誌上をにぎわしております COBOL に対する批判をほとんどつぶしてしまうことができる，ということなのです．

米国での COBOL に対する批判を分類してみますと，大きく三つに分けることができます．その第一は，COBOL の本質的なものについてですが，COBOL などではプログラムは組めない，あるいは，組めるには組めてもあらゆる点で不経済，といったたぐいのものです．これは，磁気テープもつかぬような小型の計算機の場合は別として，いちおう，自動プログラムを採用できるような大きさ，構成の計算機であればさきにあげた二社の例でつぶせます．

第二は，COBOL を使うことは，プログラムの作成と修正維持に要する時間と費用とを最少にするものではない，たとえば，Data Division をながながと書くのにたえきれないというたぐいのものです．これは，私ども自信をもってそんなことはありません，と申しあげることができます．詳しい説明は，また別の機会にゆずりますが，現在，機械語を使っている世界へそのまま COBOL を持ってきても，だめかもしれませんが，運用組織をそれ相応に考えてやれば，現在の言語をそのまま使い続けるよりたくさん利益があると考えるのです．

第三の批判というのは，これは全然的はずれなのですが，COBOL を生みだした人達が COBOL のねらいとして考えもしなかったことを特長としてふれ回るある人達がいる，そのある人達の意見に対する批判ですから，これは別物と考えてよいでしょう．

このように考えてきますと，COBOL は輸入先の日本では割合評判がよいが，本場の米国ではあまり歓迎されていないということになるのですが，事実，そのとおりであろうと私は思います．この矛盾は，実は COBOL を基礎にした運用システムを考えるということと，その運用システムを実施に移すということと

が，日本では同じなのですが，米国では別物であることによるものと思います．つまり，日本ではまだどこのデータセンタも小規模で，あるルールを変えるにしても数人が集まって打ち合わせをする程度で，はい明日からはこうしましようとできるのでしようが，米国のデータセンタのように世帯が大きく，数十人，数百人の，それぞれ経験もあり，どれも一家言持つような計算要員をある一つの運用ルールのもとで働かせているところでは，おいそれと運用ルールを変えることができないと思うのです．

私どものような世帯の小さいところでも思い当たる例がいくつもあるのですが，たとえば事務計算の場合，対象業務の処理内容が変ったためにおこるプログラムの訂正の場合なのですが，機械語かその少し上の Autocoder で，他人がその人流に書いたプログラムを読まされるのは，はじめから自分でプログラムを組まされるより時間がかかります．そこで，すこしでもプログラムの Man to Man の Compatibility をよくするために，フローチャートまたはブロックダイヤグラムの種類を定めたり，使用するワクの型を決めたり，あるいは専用の用紙を作ったりして，極力個人差が無くなるようにしています．ところが，COBOL といった新しいプログラム言語を採用すると，フローチャートは全くいらなくなってしまうしそのほかの運用ルールも現在よりうんと簡素化されるのですが，やはり全廃するわけにはいきません．簡素化されればそれなりの新しい規定を作り，その徹底方法，教育方法まで考えねばならないが，それがめんどうなのです．

もう一つ，簡単に運用ルールを変えられない例をあげますと，大規模なデータセンタでは，新しくはいってきた人達に対して，機械語の説明と Software の使用法だけメーカにやってもらって，それがすめば直ぐ戦列に加えるというわけにはいきません．分業化された組織の中で働くのですから，プログラマに対しては，そのデータセンタの運用ルールにそって，与えられたフローチャートの読み方，プログラム言語の使い方，プログラム・カードの穿孔の依頼方法，プログラム・テストの依頼方法，Debugging のやり方，完成したプログラムの整理の仕方，これだけをなるべく短時間に，しかし，それだけの専門知識は完全にマスタさせなければならないのです．システム分析者に対してはまた別の教育内容と方法がありマシン・オペレータに対しては，それ向きの教育内容と方法があるわけですから，分業方法や運用ルールの変革は，教育用のテキストや資材にまで影響をおよぼします．本当をいえば，計算機メーカの方で，そのユーザの望むようなテキストを作ってくれて，データセンタ単位に講習会でも開いてくれるのであれば助かるのですが，ことに米国の場合のように各データセンタに歴史があると，その間の発展の歴史的な事情も手伝って，大きくは似ていても細部では各社各様の運営システムをとっているので画一的なテキストでは役に立たぬということになります．

このように，プログラム言語を途中で変えるということは，いま，日本語をやめて英語にしようということと，その規模の違いこそあれそう変わらないことが，おわかりいただけたと思います．

日本の場合をみますと，計算機械化のテンポが比較的早かったために，運用システムなど考えなくても少人数で家内工業的に運用できた小型電子計算機が設置されてから大型計算機が設置されるまでの期間が短かつたので，運用ルールもないままに大型機の時代に突入した場合が多いように見受けられます．したがっていま，その大型機についている COBOL を見て，運用ルールを変えねばならないという苦痛はあまりなくて，かえって，これから運用ルールを作るところだから，COBOL を標準言語として使って，それに適した簡単な運用ルールを作っておこうというところなのでしょう．これらが輸入先である日本の方が，COBOL を素直に受け入れているという理由であると私は考えています．

さて次に，COBOL の採用が，ユーザ側のデータセンタの管理者，第一線で働く計算機要員，メーカ側の金物の設計者，Software の設計製作者に，それぞれ，どんな影響を及ぼすか考えてみたいと思います．

まず，ユーザ側のデータセンタ運用管理者にとっては，これはよいことばかりになってしまいますが，COBOL の採用はプログラムの作成と修正保持に要する時間と費用とを最小にしてくれますし，運用ルールは簡単になり，プログラマが特殊技能職から一般職にうつるので養成獲得が楽になり，FORTRAN や ALGOL による技術計算のオープンショップ制ほどまではいかないにしても，素人に対して，COBOL を手が

かりにして電子計算機のアウトラインを要領よくつかませることが間違いなく可能になり，幅広い計算機の活用が期待できるようになります．計算機要員にとっては，それを，システム分析者，プログラマ，オペレータと分けてオペレータは，変りありませんが，システム分析者は，これまでプログラマに処理内容を説明していた，その半分ぐらいの時間で，COBOL を使ってシステム分析者自身でプログラムが書けてしまうでしょう．したがって，非常にせまい意味においてのプログラマの仕事は，ユーザ側には無くなってしまいます．

一方，計算機メーカ側にとっては，まず，金物の設計屋さんですが，これは事務用をねらう計算機なら，COBOL の Compilation に要する単位問題当たりの費用が最小になるような，そういう回路構成と入出力構成とを持つ計算機を考えればよいのです．といいますのは，COBOL の Compilation の途中では，分類や照合や数表索引など，事務計算の場合の基本形が繰りかえし使われているので，COBOL の Compilation が経済的に行なえるということは，事務計算を実行させたときにも経済的であることを意味しますし，また，磁気テープを何台付けるとか，内部記憶の容量をいくつにするかといった計算組織としての構成も，COBOL の Compilation が経済的に行なえる規模のものにしておけば，いかなる規模の事務計算でもなんとか処理できるからなのです．従来，金物の設計屋さんは，最小規模の構成を本命とするのか，あるいは入出力装置を全部実装したときの構成を本命とするのか，はっきりした指令のないまま，価格の面からみれば非常に中途半端な計算機を設計しておられたのではないか，と推測するのですが，こんどは COBOL を経済的に使える機械構成と，目標がはっきり与えられるのですからやりやすくなる反面，ユーザは同一問題を異なる計算機で Compile してみることも簡単にできますから内部回路と入出力の速度とのつり合いの問題など，COBOL を使わねば表面にでて来なかったところが欠点としてめだちはじめたりして，営業からつつかれることもあるでしょう．

最後に，Software の設計製作屋さんにとってですが，COBOL を使うことが決っているということで金物に用意すべきプログラム言語の段階は非常に定めやすくなります．機械語の代りに記号を使うようなSymbolic 言語を最低のレベルとして，最高の水準が COBOL となるのですから，その中間の水準に一ぱつおくにしても，プログラム言語体系を考えることは非常に楽になるというわけです．だいたい，Compiler とか，Utility Program などというものは，機能とその使用方法を決めるのが大変で，その仕様さえ完全にできてしまえば，あとのコーデングは仕様を作ることに比べるとそれ程問題ではありません．私どもの経験では，仕様を決めて仕様書を作るまでの労働が 70%，残り 30% がそのコーディングと見当をつけているのですが，よく失敗するのはその機能と使用方法とをコーディングしながら考えるやり方で，これではいくら有能な人が何日かかってもきりがありません．その点，COBOL という文法体系ができ上っているのですから，あとは，コーダをうまく分業させてコーディングするシステムを作ればよいのです．Software はプログラム言語だけではないのですが，COBOL Compiler を作る前に，分類，照合，それに Symbolic 言語などの基本的なプログラムはでき上っていなければなりませんから，事務計算の場合のプログラムを得るための Software 体系は，COBOL を頂点とするピラミッド型を構成することになります．このようなわけで，製作すること，つまりコーディングをすることをのぞけば，Software の設計屋にとって，COBOL を使うと決めることは非常にありがたいものなのです．しかし，そのコーディングにはありがたさの数十倍の苦痛をともなうだけに，公平に考えて，COBOL を使うことによる一番の被害者は Software の製作屋ということになるかもしれません．しかし，COBOL Compiler を作るためのコーディングは，たとえそれがどんなに大きな苦労であっても，いったんできてしまえばその COBOL が使われるたびに，延べ何千人，何万人というユーザたちに利益として還元されるものですから，十分むくわれると私は考えます．それに第一，Software を作るということで禄をはんでおられるのですから，コーディングがむつかしいからというだけの理由で COBOL をやめるわけにはいきません．

問題なのは，ユーザ側です．COBOL の出現によって，プログラマの名人芸がいらなくなる，プログラマの経験が従来ほど重要ではなくなる，コーディングの仕事量が大幅に軽減されるということは，なにかプログラムというものの価値が他の仕事に比べて小さくなるとか，プログラマの威厳がきずつけられたように受

けとられがちで，米国における COBOL に対する批判のなかにも，“COBOL は，プログラマを Sloppy（なまけもの）にする以外の何ものでもない”ときめつけているのがあります．

電子計算機を動かすには，実にいろいろの仕事が必要なことは皆様よくご承知のとおりですが，その中でもプログラミングというのはたしかに重要な仕事であり，特に日本におきましては，ろくな計算機も持たぬのに適用例だけは米国の最先端のを見るものですから，まともに考えればとても処理できそうもないような大規模な業務をおしつけられては，プログラマが奇抜な着想と緻密な計画を行なうことによって規模の小さい計算機でなんとか処理してきた，という実績がありますだけにちょっと考えますと COBOL の採用によってプログラマの名人芸がいらなくなるとか，経験がものをいわなくなるということは，COBOL を使うと計算機の能力をフルに発揮できなくなるような感じを受けるのですが，これはとんでもない誤りでして，本当は計算速度が速くなり，そして入出力装置もたくさん付くような大規模な計算機になったから，かっての小型機の時代のようなプログラマの名人芸の必要性がなくなって，その代わり COBOL の助けを借りて，経験の浅い人でも非常に効率のよいプログラムを組めるようになった，ということなのだと私は考えております．

今日まで，日本の電子計算機の発展の原動力となった多くのプログラマたちの英智が，計算機メーカ側においては，COBOL Compiler を頂点とする Software の開発の面に集結され，ユーザ側においては，プログラミングより一つ前段階のより広範囲の適用業務の開発，すなわちシステム分析という面にふり向けられるなら，日本における電子計算機の活用状況は，やがて米国におけるそれをしのぐことも可能であると確信するものであります．それには，メーカ側にある人も，ユーザ側にある人も，新しい COBOL をいれるためにどんな器を用意すればよいか，おのおのの職務の中で冷静に考えてみなければならない，その重要な時期に，いま，私どもはさしかかっているのではないかと思うのです．

（昭和 39 年 1 月 27 日受付）