談　話　室

ある Symposium より

山　本　欣　子*

“第1回目は主として内科関係，第2回目は外科関係のシンポジウム”

というと，誰しもお医者さんの集まりだと思うでしょうが，さにあらず．この集りの正式名称は ALGOL-FORTRAN Compiler Symposium といいます．不定期ながら，ほぼ年1回，電子協の主催で開かれるこの symposium は，都会からの雑音を遮断する目的で2回とも箱根の山奥でひらかれて来ました．参加者は毎回 30 人前後で，かつて自分で ALGOL または FORTRAN の Compiler を作った経験のある人というのが参加資格となっています．そして出席者全員が何らかの経験あるいは研究発表を行なうべしというノルマが与えられます．

さて内科というのは，聴診器をあてたり病人の顔色や発熱のぐあいで病状を打診する．つまり Compiler の文法をどうきめるか，どういう機能を持っているかといった外側からみた Compiler のあり方です．それに対してメスで切り刻んで内部を解剖するやや残酷な分野を外科と称します．つまり Compiler の中身をさらけ出して細かい Compile technique を検討するのが外科の領分です．これに対し次回は Compiler の知性について論議したいという人がいます．source program の処理には多分に知性が必要です．おせっかいにならないように気をくばりながら error の処理，コンパイル時間，object の能率，memory の割当てなど，臨機応変に最適の処置をしなければなりません．もっとも現実には何事かが起ると突如として気が狂って暴走する Compiler もあるようですから，こうなると恐らく第3回目は精神科とでもいうことになりそうです．

さて，この symposium に参加されたのは多士済々の面々でしたから各回ともそれぞれ3日間にわたって大へん有意義な議論が戦わされたのはもちろんですが，専門的な内容はこの symposium の報告書にゆずるとして，ここでは放課後の話，つまりやや無責任な放談をとり上げることとしました．

——AとBとCとの対話——

A：ずいぶん多くの人が沢山の Compiler を作っているようだが，本当によく使われているのはどのくらいあるんだろうか．

B：Compiler なんていうものは作るためのものじゃなくて使うためのものだ．しかし現実には作るために作った Compiler なんていうのもあるんじゃないか．

C：始めから研究的という意味ならそんなのが幾つかあっても決して無駄なことではないと思うが，そうではなくて強い要求があって苦労して作ってはみたが，結局は使われなかったというのがあると思う．使われない理由というのは何だろう．

A：Compiler ができたということと，実用になるということとは全然別だ．user の要求はコンパイル時間，error の処理，object の能率など少しずつちがうだろうが，まず第一は Compiler system の信頼性ということだろう．user は少なくとも hardware と同じレベルの信頼性を Compiler に要求している．

C：たとえば見当ちがいの error message が出たりしても，もう信用はなくなる．まして本当は error のある source program であるにもかかわらず，何も教えてくれなかったりしたら絶対だめだ．コンパイルは最後までちゃんとやった．演算も一見無事にやったように見える．にもかかわらず答は正しくなかった，なんていうのは最低．

A：Compiler の error のために被害を受けて自分の仕事に支障をきたしたりすると，誰しもこんな Compiler は二度と使ってやるものかという気になるのは人情だろう．

B：相等長い期間，広く使われたある Compiler で非常に特殊なケースだったが数年後に始めて見つかった error があったという話を聞いたが，そういうことは止むを得ないのだといっても user には通用しない．たとえ Compiler にとっては些細なミスでも使

* （社）日本電子工業振興協会

う方にとっては大きくクローズアップされるんだから.

C：Compiler の中にじっとひそんでいる虫を徹底的に探し出す方法なんてあるものだろうか. たとえば hardware ではあらゆる論理的組み合わせを作ってみてシラミつぶしに check するというフォームが何かできているんだろうと思うが, Compiler でも, たとえば arithmetic なんか, variable, constant, function, すべての operator とその順序等, あらゆる組み合わせを計算機ででも作らせて片っぱしから check してみるなんていうことは無理だろうか.

A：Arithmetic だけなら, やってできないことはないかもしれないが, 無駄が多い. それに実際には procedure の parameter や他の statement との複合作用によるミスということが多いんだろうから完全な check は無理だろう.

B：現状では, やはり数多く使ってみて虫をあばき出すより仕方がないような状態だが, user's test とは称しても普通の user に頼むわけにはいかないだろう. user は user であって Compiler checker じゃないんだから.

C：しかし始めはどこかの user の協力を得ないとできないことだろう. たとえば大学あたりに依頼して講習会をして戴く. どんどん練習問題をかけてみると, 思いもかけぬ使い方が出て来て, かくれた虫が発見される.

B：一般的にいって Compiler user は Compiler のミスに対しては非常に神経質になっていることはたしかだ. hardware の一部だという考えだ. それに説明書の不備ということもかなり非難される材料になっている.

C：Compiler の説明書というのは ALGOL なり FORTRAN なり, その language をかなりよく知っている人, ないしは programming の知識をかなり持った人を対象とすべきなのか, あるいは計算機そのものの知識もあまり持たぬ人を対象にして, プログラムとはこういう組み方をするものですよというプログラミング入門書を兼ねたものにすべきなのかがどうも判然としないので, どっちつかずのものになり勝ちでよけい批判があると思うんだが.

A：両方の目的を一つのものでいい尽くそうと思うから無理がいく. 望ましいのは2種類作って, その一つは本当の入門者向きで, 簡単な例題を中心にして説明する. variable とか array とかいう用語の説明などもちゃんとやっておかねばならない. しかしその Compiler の機能を必ずしも全部網羅する必要はないわけで, これだけ知っていれば, ともかくも一とおりの仕事はできるという程度にとどめておく. そしてもう一つの方は, やや専門家向きのものとして複雑な使い方もできるよう細かい厳密な注意も書きそえる. あるいは Compile technique の上から, こんな時はこういう書き方をした方が早くなるとか object が短かくなるとかいうこともつけ加えてもいいだろう. 人によってどちらの説明書を使うかは自由だ.

B：7090 の FORTRAN System は, ごく最近まで修正改良が行なわれ続けてきたそうだが, 日本では作ったら作りっぱなしが多いんじゃないか.

C：ミスでもあればもちろんすぐ直すだろうが, コンパイル時間を短かくするとか, object をより能率よくするとか, 機能をふやすとかという改良はあまりね. 大体やりたくても そんな余裕がないんじゃないか.

A：これは世界的の事情だろうが, 今や Computer の戦国時代で, 次から次と新機種を出さなきゃならないようになっている. 大体一つの計算機の社会的寿命は5年くらいだから, 古い機械の Compiler なんかいつまでもいじっているヒマも経済的余裕も労力もない. そんな余力はすっかり新機種の方に向けなけりゃならない.

B：とはいえ after service がいらないという理由にはならないだろう. userにもいろいろあって適当に自分で改良して使っているところもあるし, 既製のものにいろいろ注文をつけて直させるところもある.

C：あるメーカの話だと, 注文が多くて結局 1 user 1 compiler というようなことになってしまったといっている.

A：他の機械との compatibility という問題もからんで来るが, 小さな機械に大形並みの Compiler 機能を要求したがる user もあるが, 機械の能力に見合った分相応のものがよいのだということを理解してもらわないと.

B：大体においてこれからは Compiler を使う人が program の専門家でない場合が増えるだろうから, そういう人達の多くは文法的に沢山の機能を持つことなんか少しも望んでいない. 覚えることは少なければ少ないほどいい. しかし, その少ない機能の範囲では十二分に面倒を見てほしい.

そして Compile time もできるだけ早くしてほしい. というようなことが共通した要望であるらしい.

C：FORTRAN は IF statement だけあればいいという話があるそうだが，ややもすると Compiler maker 側では内科的機能の拡張をやりたがり，こんな書き方もできます，こんな statement も使えますといいたくなる．この種の language を使う人でいろいろな機能を使って非常に凝った program を作りたがる人は全体の何%いるだろうか．おそらく Compiler を自分で組むような人達が案外そんな希望を持っていて，しかし現実には彼等はめったに自分の作った Compiler を使うチャンスのない人達だというようなことではないだろうか．

A：しかし Compiler を Compiler で組むなんていうことも考えられている世の中だから，あまり近視眼的なのはまずい．その機能を入れても system の能率もスピードもあまり変らないなら入れた方がいい．必要のない人はそこを使わなければいいんだから．

B：使う立場と作る立場では少しねらいがくい違うということはいくらかあるだろう．たとえば作る時に非常に苦労したところが，案外使う時には効果を発揮していないとか，逆にちょっと手を抜いたところが実は引っかかって文句が出たりする．

C：Compiler を作る人は，あまり Compiler を使ったことのない人だというのは事実らしいが，ある程度統計的にどういう機能はどのくらい使われるか，なんていうことは少し念を入れて調べれば出て来ることだろう．ごく稀にしか出て来ないケースのために非常に頻繁に使われる部分が時間的，能率的に犠牲になるなんていうのはバカバカしい話だ．

A：さっきの話で十二分の面倒をみるという中には，error や warning の message のほかに，この種の language では logical な debug に関してもこの system しか頼れないわけだから，source language の level だけで完全な能率のよい debug ができることを望んでいる．

B：object の running speed もさることながら，ともかくもコンパイル時間だけは短かくしてほしいという希望はあるようだ．そして望むらくは，Compiler は神様か仏様のようであってほしい．どんな error のある source program でもがっちり受けとめてもらいたい．

C：Compiler maker の悩みは十分の時間が与えられないことだ．精神的にも余裕がないし，やってやれないことはないのだが，早くでき上るという至上命令のためには，ここのところはサボッちゃえということになる．余裕さえくれたら理想的な Compiles を作ってみせる．

A：いずれにせよ user の無理な注文も聞き，system programmer としての自尊心も満足させ，たえず精神的に追い立てられながら，まだあまり完全に整備されていない hardware を使って無理な debug を重ね，非常によくできて当り前，ちょっとでも不備な点があると鬼の首でもとったように攻め立てられる．全く Compiler maker とはつらい商売である．