61　PASCALを使用してみたら

和田英一・武市正人（東京大学工学部）

はじめに　これまで「Pascal[1]を使用してみたら」どうであったかを報告する。

経過　我々がPascalを最初に聞いたのは1971年春，英国WarwickのIFIP, WG2.3の会合でであり，その後Acta Informatica誌で言語仕様を知った。1972年秋，Poland ZakopaneのIFIP, TC-2, Programming Teaching Techniqueの会合では，Lecarme[2]を始めPascal愛好者の少くないことを知った。一方東大計数工学科の講義で，algorithmや特にdata構造の説明にad hocな言語を使用していたが，それにPascal程度が手頃ではないかと思いたち，1972年度から，できるだけPascalに準ずることにした。1974年度からはPascalの処理系もいくつか使用できるようになったので，数理courseでは算法の演習もPascalを使っている。

結果　結論をいえば次のようである。

- array, record, pointer typeを適当に使うとdata構造が説明し易い。しかも言語はPL/Iほどには複雑ではない。
- Fortranでは許されないrecursive callの面白さを学生が覚える殆ど初めての機会となる。Lispは必ずしも毎年教えるとは限らない。
- Fortranよりcompilerが作り易いので（特にPascal-S[3]の場合）compilerの講義の対象言語としても使える。
- 処理系の増殖に熱心なので，interpreter版なら比較的容易に処理系ができ，すぐに実際のprogramを走らせてみることができる。
- interpreter版もcompile and go版も処理系のerror messageが非常によくできているし，表示も親切で，他の処理系がこれ程でないのが不思議である。
- 最大の問題は文法書で，よくもこれで「5年も使ってきた」と驚くばかり不備な箇所がめだつ。正確な文法書を急いで用意する必要がある。文法の変更がすでに2回あったのも多すぎる。
- 言語のPRに熱心なのは大いに評価できるが，Pascalを使った（と称する）論文や教科書の例が文法に正確には合っていないことが多く，気になる。

言語批判　Pascalの批判はHabermann[4]のが有名だが，それもそれへの反論[5]もどうも噛み合っていず，設計者自身があまり応答していないのでから振りに終った観がある。実際に使用して感じた具合いの悪いと思われる点は次のようである。しかしPL/Iの一利用者のように，だから機能を追加せよというのではない。

- 一番困りそうなのは恐らくarrayが上下限つきでtypeになることで，これはHabermannもふれ，Wirthも意見[6]をのべている。・Pascalで面白いのは利用者に定義できるscalar type　例えばtype state=(run, ready, wait)であるが，この識別名がprogram中に書けるだけで，入出力できず，Wirth自身，Pascal-Sの処理系の中で苦心している。このscalar typeにrealを入れてしまったが，それは到るところ例外として扱わざるを得ず，しかし文法は呑気にもほとんどこのことに触れない。realの扱いほどの言語でも苦労で，Pascalのaxiomatic definition[7]でもrealは避けて通った。・record typeには途中から別の構造を

るvariantが認められ，例えば type link = ↑sexpr ; sexpr= record case isatom: boolean of true:(pname: alfa); false : (car, cdr : link) end と書けるがこのisatom のような tag field の使い方が不徹底である．ここには今どのvariantになっているかの情報が入っているのだが，その代入は利用者まかせ，また処理系もcheckを省略してよいことになっている．また最新版ではtag fieldはなくてもよいことになった．Pascal-Sの処理系には明らかにこの使い方に誤りがある．・settypeはstructured typeの仲間だが，他のstructuredtypeはvalueの書き方がなく，componentがとれるのに，これはvalueが書け，componentがとれない．中途半端で何とかして欲しい．・宣言は使われる前に必ず書いておく必要があるが，上例の如くpointerを使ったrecursive recordの場合だけ例外をゆるし，文法，処理系ともきたなくなっている．・入出力は初の改行の文字をもっていたが，最近Fortranの行control方式を採用した．逆courseも甚だしい．・goto文は存在するが，labelがunsigned integerだけなのはなぜか．最初の版ではcase文のlabelと一緒になってambiguousなprogramも書けたがそれは改訂された．しかしstatementにlabelをつけたので，structured statementの構成statementにもlabelがつけられるようになったが，これはやはりAlgol N流にlabelをblockの構成要素と考える方がずっとすっきりする．

その他まだまだあるが，少し我慢すれば使い勝手は比較的よいといえる．我々の次期の操作システムcos®もPascal[7]で書くことにしている．

処理時間　Pascal compilerの論文[3]にあったPascal program例を近くの処理系にかけてみた処理時間（比）を表1に示す．

おわりに　文法書を除いてprogramming teaching用としてPascalは一応成功した言語といえよう．Utopia 84の到来する前に，Pascalより更に好ましい言語の出現が望まれるが，当面Pascalをよく検討して次の言語に備えるべきであろう．そのために皆さんも「Pascalを使用してみたら」いかがであろうか．

参考文献　1] N. Wirth : The Programming Language Pascal, Acta Informatica 1, 35-63 (1971); 2] O. Lecarme : Structured Programming, Programming Teaching and the Language Pascal, SIGPLAN Notices 9, No.7, 15-21 (1974); 3] N. Wirth: Pascal-S : A Subset and its Implementation, Institut für Informatik ETH (1975); 4] A. N. Habermann : Critical Comments on the Programming Language Pascal, Acta Informatica 3 47-57 (1973); 5] O. Lecarme & P. Desjardins: Reply to a Paper by A. N. Habermann on the Programming Language Pascal, SIGPLAN Notices 9, No.10, 21-27 (1974); 6] N. Wirth : An Assessment of the Programming Language Pascal, IE³ Trans on SE 1 192-198 (1975); 7] C.A.R. Hoare & N. Wirth : An Axiomatic Definition of the Programming Language Pascal, Acta Informatica 2 335-355 (1973); 8] N. Wirth : The Design of a Pascal Compiler, Software Practice and Experience 1 309-333 (1971).

表1　時間比較　total time(=compile, link and go, i/o wait) Melcom 7700 BPM

program	Algol 60	Fortran IV	Pascal
1 Matmult n=100	> 5 min.	1.297 min. (1)	2.410 min. (1.86)
2 Sort n=2000	> 5	1.312 (1)	2.280 (1.74)
3 Partition n=30	4.325 (1.82)	2.377 (1)	1.364 (0.574)
4 Charcount	—	3.277 (1)	1.373 (0.419)