Discussion and Correspondence

One-day Symposium on PL/1

held at National Physical Laboratory, 18 May 1967.

This symposium, organized by the British Computer Society, was attended by 330 delegates from 150 organizations. The chair was taken by Professor Gordon Black (*Director, National Computing Centre*), who commented upon NPL's long connection with computers, the confusion of abbreviations when PL/1 had been called New Programming Language, and the contribution being made by U.K. staff of IBM to its development. Later in the day, he stimulated discussion from the floor by urging delegates to express their views freely.

Mr. J. E. Nicholls (*IBM United Kingdom Laboratories Ltd.*) reported briefly on the facts to date. The language is based in large part on the efforts of the SHARE Committee during 1963 and 1964. Joint proposals were drafted by 3 users and 3 system-programming organizations for a new programming language (*npl*) to bring a single high level language into use to exploit the full capabilities of new computer systems: in both hardware and applications, the earlier dichotomy between scientific and business usage, more pronounced in USA than in UK, was disappearing. The new language was announced by IBM in April 1964.

The *F. Compiler Version I* was issued in August 1966 and *Version II* in January 1967: this could be used on a 64K byte store with disks and operating system/360. This compiler had been developed at Hursley (Hants).

The *D. Compiler Version I* was due to be issued in July 1967: this could be used on a 16K byte store with disk and tapes and DOS/TDS/360. It was being developed in Germany.

The following facilities were available in the compilers, but the five in italics will not be available in *DI*:

Allocate; Assignment; Begin; Call; Close; Declare; *Delay*; *Delete*; Display; Do; End; Entry; *Exit*; Format; *Free*; Get; Go to; If; On; Open; Procedure; Put; Read; Return; Revert; Rewrite; Signal; Stop; Write.

Directives not yet available in either compiler were:

Locate; Unlock; Wait,

and the Tasking and List-processing facilities were yet to be implemented.

The users' views

Mr. J. W. Lewis (*B.O.A.C.*) spoke on a user's requirements, the language, its implementation, limitations and results to date. His Corporation required a computer with an operating system giving accurate control, and modification of programs with minimum effort. They wanted to get applications working with a minimum of detailed programming and, once working, they wanted to leave the running of them to the operating section. In the past, too much time had been spent on reprogramming. The Computer Project Officers would deal with all Systems problems, and programming of both data processing and mathematical work in a single application would be facilitated by a single language: a higher level language would give automatic documentation, easier debugging during development, and easier conversion to later computers. A decision was taken in December 1965 to use PL/1 alone, for all batch-processing applications.

In expressing views, he wished to be objective and he began by paying tribute to the hard work that had been done by all concerned. He and his colleagues liked the language and felt that it achieved the aims set. It was easy to write and, in time, the reluctance of experienced programmers wore off: COBOL and FORTRAN programmers easily adapted themselves, but O.R. mathematicians had some difficulties. New staff were trained in an 8-day course of which 3 were spent on a model job, coding it, compiling it and getting it running. After 3 months practice a 2-day advanced course was held to fill in details. The *Condition Statements* had been improved since Version I, and comprehensive diagnostics assisted development.

Implementation, from the practical viewpoint, as the compiler changed, was not yet complete. The programmers' guide was incomplete and BOAC had produced their own *Newsletter* notifying changes and differences. The facilities listed above for *FII* were working well, but BOAC were encouraged by a speed-up promised for *FIII*. In the speaker's opinion, efficiency and economy in use of core was not important at this stage; the implementation of a high-level language was as important as a central processor, yet it was not available to be implemented until 3 years after the announcement of the System 360. Core is cheap and, once purchased, is permanent: programmers have to be paid, a continuing expense, and they move from task to task and job to job. At one stage, they had hoped to develop a subset of PL/1 for use as a BOAC internal standard, but it had not been practicable to define an adequate subset for the majority of problems. They were therefore using the full language and their programming standards manual defined the conventions, with input and output rules defined on training courses.

They wanted a program once written to be usable for evermore! Current limitations required ingenuity and skill to overcome deficiencies. *Operating Control* in OS/360 was not yet as secure as they would like. They were concerned about:

—absence of full facilities in OS/360;
—*ident* option, access to volume labels; not yet available;
—parameters for sort program; a detailed study of the structure of the record is first necessary, and some improvement is wanted here, as it was difficult to process files of mixed structure;
—no facilities for variable-length working;
—not yet possible to process sequentially, add and delete in same run;
—half-word binary format is accepted but a full word of store is allocated; this had serious consequences for the O.R. team who have reverted to FORTRAN for some jobs.

There was already a danger that the process of implementing PL/1 on System 360 would make it machine dependent; for example, a calculation in floating point expected to give

0·300, which is not an exact binary fraction, gave 0·299 on reconversion, because no rounding-off is automatically applied before output.

The compiler is being used: they are writing, compiling and running programs written in PL/1. Several dozen programs are working. The first programs were completed on a target set over a year ago, despite certain frustrations with the first version of the compiler in the Summer of 1966. It is difficult to compare object programs with experience on other machines. BOAC felt happy that PL/1 would meet their needs and in the light of the last year's experience, the same decision would be taken again now.

Imperial Chemical Industries Ltd.

Mr. J. M. Sykes (*ICI*) began with a brief history of computing in ICI during the past nine years: each division had selected its own computer, yielding diversified experience. Computers used earlier had included Mercury, Elliott 402, IBM 650, HEC 4, Pegasus and EMIDEC 1100. Mercury Autocode had helped achievement of a high level of technical work on the central scientific computer at Wilton (N. Yorks). Commercial work was, however, at a lower level and varied locally in the divisions.

Second generation machines installed included a KDF9 and several IBM 1401's. The pattern of development was the same and a *K Autocode* had been written for the 360 to give compatibility with the past. ICI had ordered 8 System 360 computers on day of issue.

Some time after IBM had announced their new language to supersede COBOL, ALGOL and FORTRAN, ICI, early in 1965, appointed a study committee under the chairmanship of Mr. R. A. Brooker (*Manchester University*) who had earlier been the principal architect of Mercury Autocode. This Committee included various ICI people and was given the following tasks:

1. To assess and compare the 3 IBM languages (FORTRAN, COBOL and *npl*) with ALGOL and K Autocode.
2. To consider desirability of a single language for data processing and mathematical work: would *npl* meet these requirements?
3. In making recommendations, consideration must be given to
   (*a*) Flexibility in advanced applications in a multi-programming environment: this implied open-endedness of design to permit extensions; but ICI were not now likely to extend PL/1 themselves.
   (*b*) Whether implementation was dependent on specific hardware.
   (*c*) Logical simplicity and compactness of rules for ease of learning and efficiency in application.

The verdict of the Committee, given in April 1965 was unanimous:

1. PL/1 had a wider range of facilities than any one of the other languages considered.
2. The desirability of a single language was emphasized by the convergence of data processing and scientific programming, as later stated in the preface to the *PL/1 Student Text*.
3. (*a*) PL/1 was as flexible as any current language.
   (*b*) Some features of PL/1 were hardware dependent, but the main problem will be to write compilers for later machines.
   (*c*) While breadth of features was incompatible with compactness, there seemed to be no intrinsic difficulty in teaching.

The Committee therefore recommended the use of PL/1 if and when compilers had been shown to be satisfactory, and that ICI should move towards its general use with deliberate and cautious speed.

Subsequently they had encountered surprises from implications of small print in the manuals. By July, ICI expected to have 8 System 360s installed. The KDF9 would handle the bulk of the scientific computing in K code until it could be taken over by the 360, and hence it would be some time before K code ceased to be used.

As regards other software, the Report Program Generator was little used in ICI; FORTRAN had not been used for Commercial work; COBOL had been used as the principal language in one installation and Assembly Language had been used in some places, the 1401 Emulator in others and PL/1 was mixed with other languages in certain applications.

Why, in the light of the Committee's report, is only one ICI installation at present making extensive use of PL/1? The answer is that people had got used to using other languages before PL/1 was working. Most people, in the speaker's opinion, looked forward to the day when they could change over. Two obstacles, however, were the temporary absence of the *Locate* (in buffer) facility and the apparent permanent absence of the ability to process variable-length records. Some features had been found difficult to implement efficiently and others seem to have "quietly disappeared" from the language since it was first specified. Those lost included *Call* (Routines) and *Double-length Floating Point* working for all variables in scientific work. There were examples of low efficiency of core utilization in compiled programs, particularly the storage of subroutines. The time efficiency at running stage was almost acceptable, broadly comparable to FORTRAN, but input/output operations were much too slow when compiled. Improvements were expected in later versions of the compiler.

There was a general problem in how to educate programmers to use PL/1 efficiently without knowing the object-code produced. A high-level language of this kind would not enable users to employ low-level programmers. The advantages would stem from increased productivity of source language coding, once conventions were established to avoid difficulties.

Despite the erosion of original intentions it could safely be said that:

(1) PL/1 had most of the facilities desired.
(2) The problem for a single language was "What does the Compiler Optimize?" An expert on the compiler was needed at each installation to reduce running time: list processing was at present rudimentary.
(3) (*a*) more open-ended than 2 years ago?
   (*b*) more hardware dependent now, rather than less.
   (*c*) Simplicity was indexed by Mr. Brooker as equal to "Number of Facilities/Pages in Manual".

It was difficult to calculate this index. The first PL/1 course lasted a week in June 1965; the latest took 3 weeks, and was incomplete by modern experience: there was thus a feeling that too much detail was coming into the final manual. *Subsets* were needed for a 2-day training course which could be expanded after a short practical period of training. When implementing, keep to the minimum as a standard: thus list processing might be a subset. Could we avoid the confusion

of FORTRAN before standardization, or the elective features of COBOL-61?

The above were largely personal views of the speaker. What did the programmers feel about it? A sample of 35 had been sounded with the following results. They included both scientific and commercial programmers ranging from 10 years' experience (40 programs) downward.

Answers given to questions asked were (as percentages):

		Commercial			*Technical*		
Is PL/1 easy:	*Answers:*	*Yes*	*Qualified Yes*	*No*	*Yes*	*Qualified Yes*	*No*
to learn?		35	55	10	21	22	57
to use (write)?		58	42	0	30	20	50
to debug?		31	42	27	0	27	73

The commercial programmers found the debugging facilities speeded up work but were expensive in core. The technical and scientific programmers were all firmly rooted in K code, some had also used FORTRAN and they had experienced difficulties with early compiler's compile-time diagnostics.

Some of the qualifications to "yes" were:

—language lacks simplicity and obviousness.
—too many ways of doing same things.
—difficult to write efficient programs; e.g. card reader does not exceed 200 c.p.m. after compiling from PL/1.
—exploration is wrought with pitfalls; e.g. fixed precision arithmetic is governed by rules in small print and a better manual is needed.
—when compiler tries to correct an error, the result may be misleading; for example, in one case an identifier for an array was confused with a function call.
—run-time diagnostic facilities were at present less than K code, and facilities for handling conditional jumps with unequal arithmetic expressions, also a pseudo-random number generator for statistical work appeared to be missing.

What was the future of PL/1?

In the speaker's view it might be accepted as a Standard since it appeared to be more general purpose than any other high-level language they had met. It would certainly become acceptable to IBM users. There was a good chance that it would become so popular that other manufacturers would need to implement compilers for their machines: it might be a good thing if FORTRAN and COBOL could be displaced. Development of applications using the language had not, however, been as easy as they would have been if the compiler had been available earlier. Other points were:

(1) A reasonable subset was an essential for small configurations of System 360.
(2) The missing facilities such as variable-length working were needed and unnecessary duplication of ways of doing things ought to be trimmed.
(3) PL/1 seemed to be the best contender for place of a general-purpose language, and one ICI Division at least would be prepared to switch to it, given a period of stability in the language with successful development of compilers.

University of London

Mr. David F. Hendry (*Institute of Computer Science*), discussing data-processing requirements and alternative approaches, expressed the view that the average commercial user wanted to define the shape of his input and the format of his output, and expected a software "black box" to deal with everything in between. In notes issued (at the registration table) he had focused attention on the central problem of handling data-structures. He posed a set of ideal requirements with which the facilities of PL/1 might be compared; these included:

(1) The use of character literals as identification and control symbols on input.
(2) References to data elements previously defined as components of larger data structures.
(3) Data elements to include arithmetic expressions.
(4) Directives to facilitate Editing on input.
(5) Listing of all items already defined as being stored sequentially (in juxtaposition), or as mutually exclusive alternatives.
(6) Repetition and compounding of data structures as if each structure were a data element.

Some of these facilities were already available in existing languages, (*such as Nebula—Ed.*): they could be implemented as extensions to FORTRAN or ALGOL and had already been implemented in the *BCL* data-processing language developed at his Institute for their Atlas computer. There was no indication that any of this work would be undertaken on PL/1: the BCL compiler occupied 6–7K Atlas (48-bit) instructions. The speaker pointed out that the availability of a standard language with adequate facilities would have a crucial effect on work of systems analysts.

Mr. B. Higman (*Institute of Computer Science*) deputized, at short notice, for Mr. E. Nixon, and expressed the view of members of the Institute concerned with scientific computing. Could PL/1 become a standard? It was a great improvement on any predecessor but as a potential standard it should be judged on its character set, its facilities compared with earlier languages, and whether its conventions were convenient to conform to. He had heard of difficulties in evaluating conditional expressions; e.g. "If 2 greater than x greater than 0, then . . ." had been compiled to test only the value of the bit patterns; "If not . . ." had also given rise to miscoding; in another case a sequence of operators, punched in error, had been accepted as an arithmetic expression, the even-numbered characters being treated as data and the odd-numbered as operators: these faults would need to be corrected.

The 600 man years reputed as being needed for the FII compiler was to be compared with something under 10 man years for the BCL compiler, which required less than 32K bytes without tapes and had a manual of 20–30 pages.

Dr. P. A. Samet (*University College, London*) and another delegate spoke on their experience. They could now run

PL/1 programs: it seemed unlikely that any one language could suit all applications, so why not aim at a family of compatible languages? Programming languages (problem oriented) would have to be taught to students in large numbers. At present, compiling time was much too long for educational use; only 10 programs were being compiled in an hour. How long would the 16K D compiler take on smaller college machines if the 64K F Compiler took so long? The diagnostic output at present took the form of a coded message and one had to search 100 pages of manual to find the meaning. Most students doing mathematical work did not want to become professional programmers and were not interested in elegance, only in speed. Any new language claiming to supplant existing languages must be seen to be better and there was time only to teach a cut-down subset in a general university syllabus.

The speaker needed 6 accesses to computer to compile his first PL/1 program, some difficulty or misunderstanding having arisen on use of *I . . . N* as integers. Language design was perhaps far too important to be left to manufacturers. For research and educational establishments the criterion was "How many jobs can we handle in a day?"

Audience experience

At this stage a question from the Chair elicited from a show of hands the information that about one-third of the people present had been concerned with trying to compile PL/1 programs. Later, there were indications that about two-thirds would still be using FORTRAN, COBOL etc. in 1970.

I.C.T. Policy

Mr. A. L. Vann (*I.C.T.*) asked the question: if other manufacturers agreed to implement PL/1 would IBM then come forward with PL/2? To this, Mr. Nicholls replied that PL/1 was the only development at present contemplated. Mr. D. Pearson later made a statement indicating that I.C.T. are watching development and studying the design of the language and would consider writing compilers for it, if it showed signs of becoming a standard.

At present it did not seem to justify the cost of implementation for I.C.T. machines, from manufacturer or user economic viewpoint. Evidence of demand seemed to be restricted to a small number of large users in UK. The bulk of I.C.T. customers appeared to be satisfied with languages already issued. It is difficult to get standardization, if the language development remains under the control of a single manufacturer.

English Electric Computers/ECMA

Mr. R. P. Scull (*English Electric*) referred to the work of ECMA Technical Committee 10 from December 1964 onwards, to standardize ALGOL and COBOL. Some progress had been made in devising subsets, but it would take 3 years to get an ECMA standard. The 3 major British Manufacturers were co-operating with the Ministry of Technology and the National Computing Centre to formulate a statement of policy which is expected to be available towards the end of 1967. The first design study for a subset of PL/1 on EE System 4 was in progress at English Electric.

Other users' views—discussion

Mr. Leigh (*Cheshire County Council*) referred to the use of PL/1 to compile bi-variate tables from a traffic census: he could find no improvement over FORTRAN for this work.

Mrs. M. M. Barritt (*Edinburgh Regional Computing Centre*) felt that a language could not be evaluated in the abstract. A good operating system and experienced programmers were also necessary. In her opinion PL/1 was 3 years too late for scientific research and some 3 years too early for commercial users. When would British Manufacturers have a system, including operating system, available if PL/1 adopted as standard? Would there be a revolution or an evolution and would it be suitable for multi-access computing centres? The Chairman here expressed the view that progress would certainly be evolutionary in view of the preferences of two-thirds of the audience for existing languages!

Mr. J. Murphy (*National Computing Centre*) commented that the Centre would like to receive views on measures of efficiency of languages and would like to experiment with standardizing the several dialects of the three principal systems.

The Chairman then called for a contribution from an engineering-company user, known to have contributed significantly to the development of PL/1: but it appeared that delegates had departed and the audience regretted that no contribution from them was made.

A delegate from *A.E.I.* said that much work had been done in his group in Mercury Autocode. PL/1 had been used since January 1967 but they were somewhat disappointed with the present implementation. Non-programmers seemed happy with the language and the first trial usually sorted out syntax errors: he agreed with a previous speaker that compiling took too long, and they had had some odd experiences in sorting out hexadecimal dumps.

A program written in ALGOL was compiled on a KDF9, using the Whetstone compiler, in 40 seconds. With the Kidsgrove compiler it was a little slower. A similar job in PL/1 had taken 30 minutes to compile, but running time was satisfactory after compilation. All facilities required by his group were present in PL/1, but improved efficiency was wanted in compiling, execution time and core usage: perhaps two compilers would be necessary one giving speed of compiling at expense of elegance in execution.

Mr. E. B. Fossey (*Atlas Computer Laboratory*) referred to the work of a BCS study group in which various languages were compared. Their provisional conclusions were:

(1) They did not like the PL/1 character set: many characters required were missing.
(2) A FORTRAN user would find himself at home in PL/1 with few exceptions, for example, when dealing with *equivalent* statements. Existing FORTRAN programs could be converted semi-automatically.
(3) An ALGOL programmer would also be happy subject to certain omissions, notably the inability to call in a routine by name and the absence of conditional expressions.
(4) The *structuring* facility was liked, and good use made of *Allocate* and *Free*.

Mr. B. D. Johnson (*Henry Wiggin & Co. Ltd.*) commented on his experience with a 16K 1401 computer with disks and a model 360/30 (64K) with 5 disks and 4 tapes. They had relied on IBM's assurance of PL/1 being viable, but of 40 programs written for the new machine, only one was at present working. Another program using disks, tapes, cards and printing with 300 statements had not yet been fitted into their configuration.

Mr. G. Aiyer of *Unilever Ltd.* warned users to note carefully the reserved words in the language, when creating their own lists of identifiers.

IBM reply

Mr. J. E. Nicholls replied that the objective of PL/1 becoming a standard did not mean that it would supplant all other languages. IBM were aware of some diagnostic difficulties, but nevertheless the language was widely applicable. Such a language was needed to increase the productivity of programmers whose costs were rising when hardware (system) costs were falling, and to decrease the cost of getting debugged programs completed. There was a firm belief, expressed in the manuals, that programming problems in scientific work, data processing and real time applications had a unity which could usefully be exploited for training purposes. Finally they had demonstrated that it was feasible to include all necessary facilities in a common language and get acceptable implementations.

A comprehensive language would be a major language in a complete program: they recognized the need for subsetting and extension (open-endedness). System dependencies were localized so that a programmer could quickly replace relevant routines. The PL/1 functions taken over from existing languages included extended data types, data aggregates into arrays etc., extended operations and built-in functions, generalized arithmetic expressions, generalized file-handling, essential input/output facilities for editing and listing data, input and output of complete records, storage allocation (extending to dynamic allocation).

Significant extensions, not in earlier languages, were *Macros* introduced at compiling time, very powerful interruption handling facilities for real-time work etc., and their intention to provide at a later date facilities for far-reaching extensions into *Multi-tasking* and *List-Processing*. Discussions with customers had confirmed the need for these facilities.

To date, implementation had taken the form of a single (F) compiler with reasonable compiling speed and little optimization, but good diagnostics. (There followed some examples.) The great variety of means of procedure description gave great sensitivity to programming habits; but a better understanding of PL/1 would give a better program, when, for example, translating COBOL.

One example was of a program to process small files: the PL/1F version occupied 2·22 times the space of a COBOL OS/360 program; this ratio was further analysed into Library codes, called in for all programs, 6·51, and procedure codes 1·02. For larger programs the 2·22 ratio would be reduced. Compiling time ratio was 1·25, link-edit 1·54 and execution 1·56.

A comparison with FORTRAN of 5 scientific programs showed a 1%–5% reduction of space. Compiling time ratio was 0·51 and 0·74, link-edit 1·47 to 1·54, and execution time 1·11 to 2·44.

Considerable analysis would be necessary to improve these figures, but they demonstrated that the PL/1 compiler was broadly comparable with COBOL and FORTRAN compilers. Time for debugging was very favourable and programmers knew that they could complete debugging in 2–3 compiling runs. Hence programmers' productivity was good.

They recognized the importance of the points made by Dr. P. Samet on compilers for educational work, and there was the alternative question, how many people were prepared to bear the overheads in compiling time etc., which a high-level general-purpose language involved.

Comments on publications were acknowledged and would be dealt with: an improved *Manual* would be issued and formal definitions were being released as work-in-progress reports, to give users clearer view of obscure parts. It was believed that PL/1 was well specified compared with other languages at a comparable stage of development. They were convinced it could cover all applications, believed it could become a standard, and were concentrating effort on it.

Professor Black reported that the Government had asked the British manufacturers to talk among themselves on standards. They were also getting groups of users together, under the auspices of the NCC, to discuss standard languages, to get some compatibility within each group. The experience earlier in the day, when some 200 people had indicated their continued preference for ALGOL, FORTRAN, COBOL for some years to come, confirmed that we were unlikely to attain a general standard language in the United Kingdom.

The meeting closed with a vote of thanks to NPL and to all those who had contributed, proposed by the Chairman; a word of appreciation for the Chairman's efforts, spoken by Mr. H. Voysey, on behalf of the Committee which had organized the meeting, and generally applauded, terminated the proceedings.

A rather nostalgic feature for older BCS members had been the service of the after-lunch coffee in the ACE room, where the large cabinets and heavy ducting of the now idle cooling system ("fans the property of M.P.B. and W.") emphasized the changes that had taken place in *hardware* in less than a decade. One delegate was heard to remark that this machine might have been able itself to advise on such topics of the day as "rounding errors in algebraic processes", had it been able to speak!

Editors' note:

The above report has been compiled by an editor from notes made at the meeting: the assistance of Mr. and Mrs. H. Voysey is acknowledged. Short contributions are invited for Discussion and correspondence *detailing experience, as the compilers for this (and other programming languages) are developed.*