# BCS FORTRAN SPECIALIST GROUP

Minutes of a Meeting held on 1st December 1975
at the Staff Common Room at the Polytechnic of
Central London, 115 New Cavendish Street,
London W.1.

Present:

| | |
|---|---|
| Mr. D. T. Muxworthy (Chairman) | University of Edinburgh |
| Mr. E. O. Bodger (Vice Chairman) | IBM WT Systems Aid Centre |
| Dr. A. W. Bishop | Hawker Siddley Aviation Ltd. |
| Mr. P. D. Bond | Philips Industries |
| Mr. O. Branley | London Borough of Redbridge |
| Mr. M. R. Dolbear | B.P. London |
| Mr. J. Dyke | Huntingdon Research Centre |
| Dr. J. S. Hutton | Rutherford Laboratory |
| Mr. M. J. King | B.B.C. |
| Mr. D. J. Maisey | I.C.L. Bracknell |
| Dr. J. D. Murchland | University College London |
| Mr. K. Normington | Lanchester Polytechnic Coventry |
| Mr. T. D. Palmer | Computer Power, Cannock |
| Mr. T. L. van Raalte | M.O.D |
| Dr. J. K. Reid | A.E.R.E. Harwell |
| Mr. R. W. S. Rodwell | I.C.L. Euston |
| Mr. A. J. H. Walter | Rutherford Laboratory |
| Mr. P. A. Clarke (Secretary) | Rothamsted Experimental Station |

Apologies for Absence

| | |
|---|---|
| Dr. J. C. Baldwin | University College Cardiff |
| Dr. J. Morice | Mullard ResearchLaboratories |

## 1. Approval-of Minutes

The minutes of the meeting of 2nd October 1975 were approved subject to the following correction: In section 5. for 'Dr. Day' read 'Alan Shaw of University College London'.

## 2. Matters Arising

(a) Mr. Muxworthy said that he had written to the organisers of Datafair 75 to complain about the accommodation provided for the group. The room in question was a converted bedroom and the temperature was in excess of 100 degrees F.

(b) Section 5. Correspondence had been received from Mr. Patel indicating that Compulogic Ltd. have an instruction monitoring program which can be leased for £20 per month.

Mr. Walter said that there was a similar program written in Fortran, at the Atlas Laboratory.

Dr. Reid said that he had a similar program.

Mr. Dolbear was concerned that such programs should have as few restrictions as possible and should be able to monitor the number of times the statement on a logical IF was executed.

For example, the monitor should indicate the number of calls to SUB made during execution of a statement of the form:

```
IF (---) CALL SUB.
```

Some mention was made that a Fortran compiler called SOFOR available at the University of Southampton on an ICL 1900 series computer had such a monitoring option.

(c) Section 3. Arrangements were being made to get speakers from Computer Manufacturers. Univac and Digital Equipment Company had agreed to provide speakers.

A request f or members of the group to provide accommodation for meetings, particularly in the London area was made. Currently the meeting rooms have to be rented and this money comes from the Group's budget.

Several people offered accommodation outside London.

The number of people on the Group's mailing list has increased by about 50% in the last year, and this together with increased postal charges may force us to consider some form of restrictions. As a first measure, it vas decided to ask all members who no longer required minutes to inform one of the Group's officers.

## 3. Fortran Group and Committee Activities

### 3.1 ANSI X3J3

A letter had been received from Dr. Meissner confirming that the draft revised standard would be published in the ACM SIGPLAN Journal in February, assuming its approval.

Extra copies will be printed and a provisional order for 50 of these for BCS Fortran Specialist Group members has been made. We have suggested that these are delivered by air freight to avoid undue delay.

The BCS has agreed to underwrite this publication to the extent of £75 (see minutes of June 1974 meeting). The cost per copy is not yet known, but is expected to be about £3. To ensure a copy please contact the Secretary in writing. To date about 20 orders have been received.

A letter from Frank Engel to Peter Hammersley (editor of the Computer Journal) indicated that a summary article by Frank Engel about the revisions should appear in the May 1976 issue of the Computer Journal.

A request for more members to join the working party to review the ANSI revisions led to a provisional party consisting of Messrs. Day (Chairman), Clarke, Dolbear, Maisey, Reid and Walter.

Any other members wishing to join this party should contact Dr. A. C. Day (at the Computer Centre, University College London) as soon as possible.

T'he minutes of the August 1975 meeting of X3J3 were discussed. It was noted that two more in compatibilities with the 1966 standard had been approved: (1) It is not allowed to have two or more unnamed block data subprograms in an executable program. (2) A '*' in column 1 now indicates a comment line. Existing programs with continuation lines which happen to have a '*' in column 1 may now fail because that line will now be taken as a comment line.

A straw vote showed that those present were in favour of type DDUBLE PRECISION COMPLEX by eight votes to two.

The minutes of the October 1975 meeting of X3J3 were discussed. Comments were sent to X3J3. See Appendix A.

It was noted that meetings in the United States to discuss the revised standard were planned for 9 February 1976 at Fortran Forum West as part of the Computer Science conference at Anaheim and later at Fortran Forum East in Washington, an ACM meeting.

Dr. Murchland made a proposal that "in order to facilitate reference to statements in a Fortran program by the compiler and in documentation, there shall be a standard numbering of the statements in the program."

The members of the Group had discussed this before (see minutes of meeting of June 1974).

There was some discussion on whether numbering should be based on lines or on statements.

There seemed to be increasing need for such numbering because of the increasing use of preprocessors of many types - structured preprocessors (such as SHELTRAN and the 60 odd others recently listed), macro preprocessors, efficiency monitors, PFORT verifier, database DML host language preprocessors and some editors. Numbering would also be useful in referencing statements in published algorithms.

It was decided to leave this to another meeting.

## 3.2 CODASYL Fortran DBMLC

A letter had been received from Mr. J. S. Knowles of Aberdeen University relating to the correspondence with the CODASYL committee. He confirmed that the use of pre-processors to 'host' the DML could cause restrictions on interactive use.

Mr. Clarke had received a copy of the first draft of the FDBMLC's Journal of Development 16 September 1975, version 0.2 together with a copy of a working paper on a DDL for Fortran by J. P. Rundell.

These documents were being circulated to members and interested persons should ask to be included on the circulation list.

## 3.3 Fortran Development Committee

The fifth FORWARD Fortran Development Newsletter dated October 1975 had been received.

The survey of Structured pre-processors was continuing and copies of the questionnaire forms can be obtained on request from the Secretary.

On the subject of structured pre-processors, an internal report on the talk given by Mr. Croes at the last meeting on the subject of SHELTRAN had been written by Dr. Murchland for use at UCL.

This gave a broader view than the report by the Secretary attached to the last set of minutes (as Appendix E).

Dr. Murchland agreed to let us publish his report and it is attached as Appendix B to these minutes.

## 3.4 ECMA Activities

Mr. Muxworthy drew attention to the minutes of the meeting of the TC8 committee. He expressed concern about a comment made by TC8 that the ANSI X3J3 revisions work was not serving European interests. Mr. Maisey said that this referred to a comment made by a group studying real-time and process control applications of Fortran and the comment applied to this area on1y.

Mr. Clarke said that there was interest in this problem in the United States too and that he was awaiting information from those concerned - centred at the Purdue International Workshop, Purdue University.

## 3.5 Dutch Fortran Study Group

A letter of introduction inviting an exchange of information on Fortran matters, and in particular, on the ANSI Fortran revisions had been received from Mr. Heyns, Secretary of the Dutch Fortran Study Group.

## 4. Any Other Business

Mr. Muxworthy drew the group's attention to a method of reducing the number of disc transfers for IBM 360/370 Fortran programs which made many overlay changes during execution and used two or more regions.

It was found that overlay calls across regions could cause a minimum of four disc transfers (reference R. A. Buhler, in the conference report 'Comp Stats 1974') and typically 8 to 32.

By including the names of these called subprograms in an EXTERNAL statement in one of the subprograms in the root segment the number of disc transfers is reduced typically by about eight percent. However, each name appearing in the EXTERNAL statement increases the core requirements by about 20 bytes.

Following the survey of members, it was realised that there was interest in topics other than Fortran Standards, and recently our meetings have been covering some of these topics. The Steering Committee have been aware that this conflicts with the Group's original document of objectives and a request has been made to draft a revised document for discussion by members of the Group with a view to submitting it to the Specialist Group Committee of the BCS. (See Appendix D).

5. Mr. Walter gave a talk on Optimising Compilers for Fortran (see Appendix C for details) which was followed by questions and discussion.

6. The next meeting will be held on 2nd February 1976.

## Appendix A

### MEMORANDUM

From: BCS Fortran Specialist Group To: ANSI X3J3

Date: 30th December, 1975

Subject: Comments on items under discussion

The following points arose at our meeting of 1st December 1975 and relate to FORTREV/71 (75-09-26).

1. A majority of those present were in favour of a type DOUBLE PRECISION COMPLEX. The view was expressed that for many applications and on many processors, double precision was necessary.

2. In section 19.1.1 item (10)

   The wording 'Redundant type specifications' was not specific enough. It might be taken to mean that a variable whose type was defined implicitly by its name or by an IMPLICIT statement should not also be defined in a type statement. We would prefer a wording such as 'Duplicate type definition in type statements' if, as we suspect, this is what was intended.

## Appendix B

## Report on SHELTRAN by Dr. J. D. Murchland

Dr. Croes gave a talk to the British Computer Society Fortran Group on 1975 October 2, and the summary of his remarks below supplement the article, which is no longer quite up-to-date.

She11 has about 50 computer centres, and 5,000 people programming. Seventy percent of this programming effort is devoted to 'maintenance', including program adaptation and elaboration, of course. Hence the acute interest in program readability and documentation.

In devising this Fortran dialect they were acutely conscious that anything they provided would have to be supplied for evermore.

The pre-compiler, not the user, is responsible for the nicely indented listing, with note-type comments copied to the right-hand positions of the statements to which they refer. It is also possible to get an output consisting only of the statement type and notes, so recovering a kind of high-level flow diagram.

As well as this documentation, the pre-compiler makes an exhaustive check of the specific1y Sheltran statements. The target Fortran compiler is relied on for the rest of the diagnostics required. Correspondence is established by the pre-compiler's generation of the same ISN numbers that the Fortran compiler will use (as in the p.129 example). Unfortunately-this means a different numbering algorithm for each Fortran compiler! The generated Fortran is never seen or used by the programmer.

No features of Fortran, other than statement numbers, are deleted - for example, multiple Entries remain.

One distinction between Sheltran and the more than 60 other Fortran pre-compilers which have been described is its universality: it operates for all the Fortran compilers with which Shell is concerned (IBM, CDC, XDS, PDP, ...). Thus the pre-compiler is written in ANSI standard Fortran.

Another distinction is that there is an exactly similar Shelbol for Cobol, so similar indeed that careful scrutiny of the input is needed to tell whether the target is Fortran or Cobol.

Programmers seem to like Sheltran, since although they lose the label feature they more than make up through the 'if ...else', 'perform' and documentation facilities. In practice, it seems that program development time is reduced by a quarter, the generated Fortran programs are usually slightly shorter and somewhat faster. The main reason for the latter is the tendency of programmers to replace subroutines by 'performs' (i.e. code copying). The latter is implemented by Assigned Goto's, which are much faster than subroutine Calls. The main justification for Sheltran remains the maintenance facilitation, however.

There is some associated software. Shell programmers have a 'Librarian' facility for managing the texts of programs, variable name replacement and so on (which Dr. Croes thought too obvious an aid to be worth mentioning in his talk!). For those who want it, there is a (Sheltran) macro system MAGIC. Another system is BEEF, which provides a universal correspondence between characters and integer values, and input-output conversion, across the range of machines which Shell.use. Because of the slowness of Fortran input-output, use of BEEF can save one-third of execution time.

Accepting the claims that Sheltran is a better and more efficient programming language than ordinary Fortran, and that it is universal across different machines, how can it be obtained?

Here is the catch. Shell do not wish to enter the software marketplace themselves, and are looking for a software house to market it worldwide for them. This means the price will be high, and also the house has not yet been selected. What Dr. Croes could offer, in the immediate future, was a corrected listing of Sheltran 1. Sheltran 3 is the next version, Sheltran 2 the one in current use. The Sheltran 1 version has the same features as Sheltran 2, but is not so efficient. The listing would cost £45 from Dr. Croes at the address given, and the user would have to master it and provide his own implementation and maintenance. The program is about 4,000 Fortran statements long.

J. D. Murchland 75:10:9

## Appendix C

## Summary of a Talk by Mr. A. Walter on Optimising Compilers for Fortran

Mr. Walter described how the IBM 704 Computer had influenced the design of the first Fortran systems in 1958. At that time, one of the design aims was to produce code as good as assembler. Examples of 704 features were the arithmetic IF, which was orientated to the machine instructions. In subscripting, the variable had to be non-negative and the fact that DO loops could only be nested to 10 deep, led to the 'extended range' implementation.

He then considered the impact of standards on Fortran from two viewpoints; that of the compiler writer and that of the user. He concluded that more restrictions had been put on the user and less on the compiler writer by standardisation. He gave examples of how compiler writers could interpret the standard to permit optimisations. One commonly used technique for moving code outside loops depends on the assumption that the loop will be executed at least one time. This assumption could be made because in the case where the initial value was greater than the final. value, the outcome was 'undefined' by the standard.

This would not be possible if the new standard allowed 'zero times though the loop' in this case. In fact moving code outside the loop could cause it to be executed at least once and leaving it inside would cause it to be executed zero times unless care were taken to avoid this. Compilers such as the IBM H level Fortran tended to avoid some of these problems by converting DO constructs into IF and GOTO constructs before optimising.

One problem associated with 'code moving' was that of error messages. When code was moved, and perhaps a single piece of code used to replace duplicate code in the original source, there was a problem in producing execution-time error messages that relate to the original source.

The relative place of manual optimisation was discussed in terms of questions such as: Is it productive of programmers time? Is it generally possible? Is it conducive to clarity and ease of debugging? Can the programmer do it consistently and with certainty? And having concluded that the answer is no! Mr. Walter indicated that the programmer could make best use of his special knowledge of the problem in the areas of improvement of data structures and algorithms.

He concluded with a brief summary of the performances that could be expected from some optimising compilers and gave some examples of specific optimisations and false optimisations.

## Appendix D

## BCS Fortran Specialist Group Proposed.Revision of Objectives

The existing objectives published in the March 1970 Computer Bulletin are:

a) To form a Forum for the discussion of problems concerned with establishing and maintaining FORTRAN standards.

b) To work in association with the BCS Standards Committee and through them with national and international standardisation bodies.

The following additional objectives are proposed:

c) To obtain and distribute information concerning developments in, and extensions to the Fortran Language as defined by existing standards.

d) To obtain and distribute information concerning implementations of Fortran processors and supporting systems.

e) To obtain and distribute information relating to programming techniques and methods employed in the application of Fortran to specific problems.

[*Produced from the original typescript using OCR with typos corrected, September 2014*]