A Conversational Processor for a Structuring Language

KAZUO USHIJIMA*, ITUZIRO ARITA* AND SETSUKO OTSUKI*

1. *Introduction*

In general the man machine interaction is one of the most important problems in time sharing system. This paper discusses about a conversational use of a language with block structure.

Although the language with block structure is excellent for procedure description, it is an open question whether it is satisfactorily adaptable to conversational use. As a familiar example, ALGOL is treated in this paper. A conversational processor discussed in this paper has been inplemented as an experimental project at Kyushu University using OKITAC 5090.

2. *Man-Machine Conversation*

An indispensable condition for a man-machine conversational processor is that in accordance with man's changeable and careless mood, the error indication and partial execution of a program can be easily carried out at his descretion.

We adopt the following methods:

(i) In order to carry out and output an arbitrary part of the program at an arbitrary time, the "compiler" for translation and the "executor" for execution are separated from each other.

(ii) The smallest unit for processing (SUP) is introduced. This serves to localize effects of the correction and to point out the range of the execution clearly. The unit is so determined that it may not contradict with the block structure of the language.

(iii) A program, "execution designator", is introduced which determines both the mode and the order of the execution and can be modified and copied at will.

Fig.1 shows an example of the conversation. The command such as ☆ALGOL and ☆EXECUTE changes the phase between the compiler and the executor. This change means, from the side of the processor, an exchange of the base register, at the top of the executable program area (X-region) and at the top of data area, which consists of the read write area (W-region) and the read only area (R-region).

This paper first appeared in Japanese in Joho Shori (the Journal of the Information Processing Society of Japan), Vol. 9, No. 1 (1968), pp. 7~13.

* Faculty of Engineering, Kyushu University.

```
        Your name please
1       Taro Yamada
        New or experienced
2       New
        Your number 0003
3       ☆ALGOL
3       begin real A,B,C; array D[1:5];
6       procedure P(X,Y,Z);
7       value Y; array X; real Y,Z;
10      X[1]:=Y+Z;
11      A:=B*0.5+C*A;
12      P(D,A+1.0,B);
13      ☆EXECUTE
13      STEP(11,12)11,A; 12,D[1];
14      ☆ASSIGN 11,A:=1.5,B:=2.3,C:=3.8;
14      ☆START
        11 A= .6850000000₁₀1
14      ☆RESTART
        12 D[1]= .1015000000₁₀2
14      ☆ALGOL
14      if D[1]=0 then A:=B*0.5-C*A;
16      ☆EXECUTE
                .
                .
                .
```

Underlined parts are output from the computer.

Fig. 1. An example of man-machine coversation.

3. *ALGOL Compiler*

In order to imprement a conversational compiler which processes a block-structuring languagc under a time sharing system, the following principles are taken into account:

(i) The source program must be sequentially translated because any part of the input program (unless it is grammatically completed) must be executable whenever the user will execute partly.

(ii) In order to diminish interpretive part and to increase efficiency at execution time as far as possible, the most part of the source program must be directly put into machine codes.

(iii) The object program must be so constructed as to minimize influences by correcting or changing a source program partly.

(iv) The object program must be relocatable, because it is compiled in the data area of the user (W-region).

(v) The important restrictions to the revised ALGOL 60 are the following: 1) a block and a compound statement are not distinguished and 2) a declaration can only appear by the time before the very name in the declaration is used in some statement.

In order to satisfy above principles (i) and (iii), SUPs in ALGOL are specified as follows: 1) ⟨label⟩ 2) *begin* 3) ⟨declaration⟩ 4) executable statements or clauses (⟨assignment statement⟩, ⟨go to statement⟩, ⟨procedure statement⟩, ⟨for clause⟩ and ⟨if clause⟩), 6) *end* and 7) *comment*.

Outline of compilation The compiler translates every SUP of the source program which is conversationally typed in and combines each unit into a list structure according to the tree structure of ALGOL language. The compiled SUP as an object program is composed of both "head" and "body". The former includes several pointers of the list structure and pieces of information about internal structure of the "body" with a fixed length of words. The latter is composed of the source image, the corresponding object codes and arrays of both names and constants used in the very SUP. As for examples of "body", the declaration real A, B, C; in Fig. 1 is embodied to Fig. 2 and an assignment statement 11 in Fig. 1 is shown in Fig. 3, where parentheses ( ) and brackets [ ] mean indirect addressing and SCC modification respectively. BOX 4 is the initial address of the compiled SUP of Fig. 2. R-, W- and X-register modifications are specified by R, W and X.

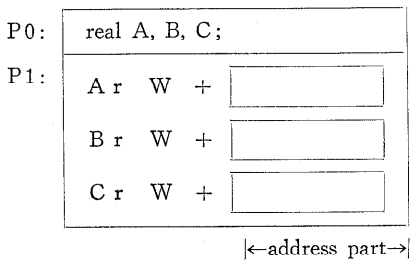

Fig. 2. An example of "body" of a declaration.

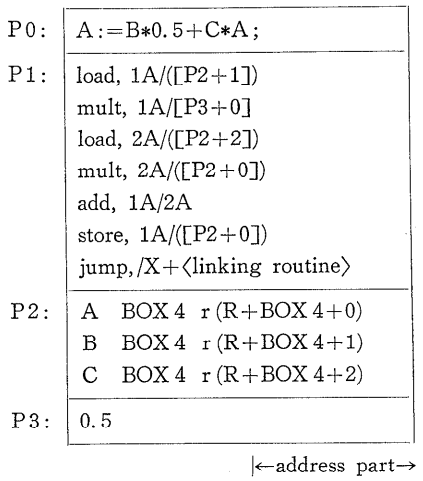

Fig. 3. An example of "body" of an assignment statemnt.

There are many difficult problems to design a conversational compiler, one of which is to deal with ⟨designational expression⟩, because it cannot be sequentially determined while compiling.[1)]

In order to correct and change the source program conversationally, three commands, ☆DELETE, ☆INSERT and ☆EXCHANGE are supplied.

4. *Executor*

Execution designator In order to have conversation with machine at execution time, the concept of "execution designator" is introduced. "Execution designator" is a kind of program which specifies the mode and the range of execution, and points out the variable names to be output for the sake of checking (see SUPs 13 and 14 in Fig. 1).

The mode "FULL" means full speed mode, so the specified range of the source program is executed sequentially in this mode. On the other hand, in the mode "STEP", execution

of the program is forced to stop at the end of each SUP and the next operation is determined by user's command. "Execution designator" is not a part of ALGOL source program but it is interpreted by the executor.

Commands of execution phase For the man machine conversation at execution time, following commands are provided:

(i) ☆ASSIGN : To assign values to varibles.
(ii) ☆TYPE : To type out values of variables.
(iii) ☆START : To start from the initial SUP.
(iv) ☆STOP : To stop at the end of SUP.
(v) ☆RESTART : To start from the next SUP.

Outline of executor Roughly speaking, "executor" consists of two parts. The one is a routine to interprete "execution designator". When one statement of "execution designator" is typed in, this routine refers to the list of compiled SUPs and adds them some records (mode tags, output tags, etc.). The other is a routine to manage execution as follows:

1) To determine the initial SUP.
2) To find the "head" of the corresponding SUP from the list.
3) To carry out the "body" of the corresponding SUP, if mode tag is found in the "head".
4) To output the corresponding value, if output tag is found in the "body".
5) To determine the next SUP by both information in the "head" and the consequence of execution of the "body".

Standard functions and standard procedures are contained in this routine.

Reference

[1] Ushijima, K., I. Arita and S. Otsuki, A Conversational Processor for a Structuring Language (in Japanese), *Joho Shori*, *9*, 1 (1968), 7–13.