A Description of Hichart Program Diagrams for C

2T－7

°Youzou MIYADERA[1], Masumi OHNO[2], Yuka OCHIAI[3], Takeo YAKU[4]

[1]Tokyo Denki Univ., [2]DEC Japan Co. Ltd., [3]NEC Co. Ltd., [4]Nihon Univ.

1. Introduction

We proposed an implementation of a Hichart generating system for Pascal language in [2]. In our department, we have been utilizing the Hichart system for education of programming language and visually describing structured programs. Recently, C language has become important and popular because of the expansion of Unix workstations. Therefore, we considered the need for the implementation of a Hichart generating system for C language. At first, to create the system, we had to design how to describe Hichart program diagrams to suit the C language.

In this paper, we will propose the description of Hichart program diagrams for C.

2. The present Hichart system

We will consider Hichart program diagrams in this paper, which are the set of tree flowcharts. Hichart has following major characteristics: (1) a tree-flowchart keeps the flow control lines of a Neumann program flowchart. (2) only two classes(iteration and selection) of symbols are added to the Neumann flowcharts to represent control structure. (3) a program is represented by a tree-like structured graph, which is spreaded rightward by the use of those two classes of control symbol. Therefore, Hichart can describe simultaneously both the hierarchy and the flow of control.

Now, in our department, Hichart system was implemented on MS-DOS, IBM VM/CMS and Unix. It is being utilized for education of programming language for Pascal.

Hichart system consists of three tools: a Hichart flowchart editor (HED), a Hichart from Pascal translator (HfromP), a Hichart to Pascal translator (HtoP). There were some problems about the generating and drawing diagrams related to the system, after solving them we got some results [1], [2], [3].

3. Design of Hichart system for C

According to expansion of Unix workstations, C language has become popular and important, so we wanted to develop a Hichart system for C. At first we had to design how to describe a Hichart program diagrams to suit the C language. We run into some points which were not fit to describe a Hichart diagrams for C. Followings are the problems which should be considered.

(1) How to treat #include statement? Should we describe source program on the diagrams, which is called by #include statement from the other files?

(2) How to treat Macro define and replacement? According to replace character in a preprocessor, diagrams are different from original source program.

(3) How to describe relations of functions of target program? In case of Pascal program, functions and procedures express one of sub structure of main program. But functions for C are separated from each other, and main program is also a function.

(4) Should we use special symbols for input/output? Because C program does not support input/output statement such as Pascal, functions of input/output are realized by function call.

(5) How to treat statements for processing files?

We considered about the above problems, and came up with the following improvements and restrictions.

(1) Control statements for preprocessor except for #include, #define are not intended for describing diagrams. #include declaration is expressed by reference symbol, but its source is not. For #define, we treat only replacement characters.

(2) Every function is described in the same hierarchy of a Hichart tree diagrams.

(3) Input/output function calls are described by general data symbol for input/output.

(4) Target source programs have to exist in a file.

Table 1 shows the fundamental flowchart symbols available in Hichart, meanings and examples for C.

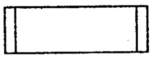
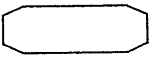
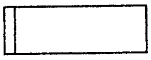
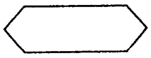
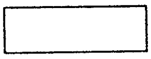
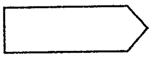
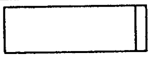
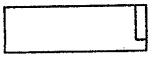
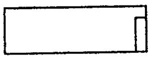
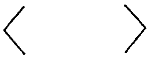
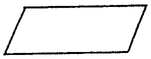
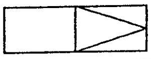
CLASS	symbol	purposes
1: module		module name
2: declaration		declarations of variable and struct
3: function		function name
4: preparation		function head and arguments list
5: process		assignment statement, function call ststement
6: condition		if / switch - case statement
7: iteration		for statement
8: iteration		while statement
9: iteration		do - while statement
10: preprocess		#..... statement
11: in/output		input / output functions
12: jump		goto / return / break statement

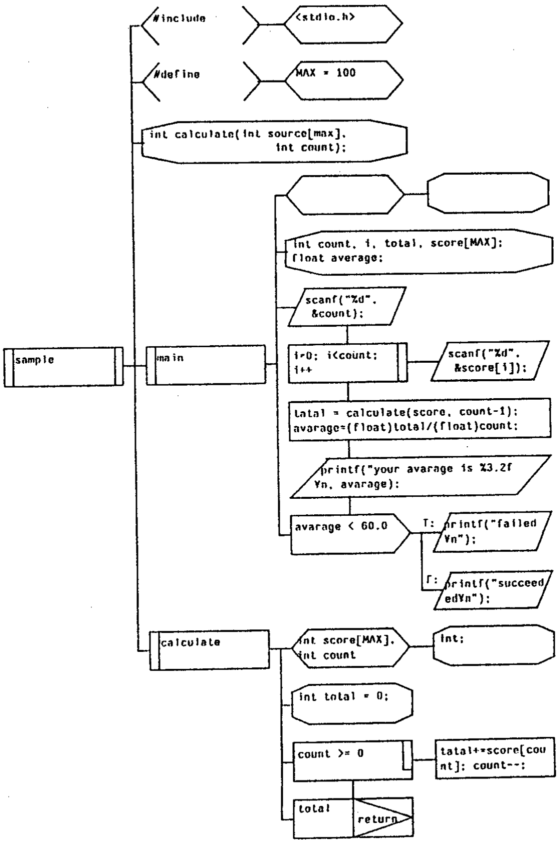
Table 1. The fundamental flowchart symbols in Hichart for C.

And a Hichart flowchart description is illustrated in Figure 2.

4. Conclusion

We proposed a description of Hichart program diagrams. It is variable that control flow, data structure and program source code are described visually in a Hichart. Recently computer graphics have been developed so rapidly on Unix workstations. Therefore, Hichart should become one of important tools for software development support environment.

Some of the problems remained for a future study are as follow.



```
#include  <stdio.h>
#define  MAX = 100
int calculate(int source[max],
              int count);
sample
main
int count, i, total, score[MAX];
float average;
scanf("%d",
 &count);
i=0; i<count;
i++
scanf("%d",
 &score[i]);
tatal = calculate(score, count-1);
avarage=(float)total/(float)count;
printf("your avarage is %3.2f
¥n, avarage);
avarage < 60.0
T: printf("failed
¥n");
F: printf("succeed
edYn");
calculate
int score[MAX],
int count
int;
int total = 0;
count >= 0
tatal+=score[cou
nt]; count--;
total
return
```

Figure 2. An Example of Hichart flowchart description for C.

(1) To implement to a Hichart from C language translator(HfromC).

(2) To improve description to a better one by users opinions.

We would like to thank Mr. Terasawa, I. and Mr. Kaminaga, I. for their valuable suggestions.

References

[1] Ogura, K., Miyadera, Y. Yaku, T. et al., Generation of Hichart Program Diagrams, *Journal of Information Processing*, *vol.15(2)*, (1992).

[2] Go, Y. Miyadera, Y. Yaku, T. at al., Method for generation of Hichart Program Diagrams, *Transaction of Information Processing Society of Japan*, *vol.31(10)*,pp.1463-1473(1990). (in Japanese)

[3] Miyadera, Y. Imai, K., et al., ETA87-An Extension of a Hichart Flowchart Processing System, *35th Conference Processing, Information Society of Japan*, pp.1201-1202(1987).