

A Kind of PAD Including Declarations to Teach C/C++ Programming

Masachika Miyata

Kanazawa Institute of Technology, Ishikawa, Japan.

Index Terms: Pseudo-code, Structured programming chart, Top down analysis.

I. INTRODUCTION

It is well known that several charts like NS chart [1], Jackson's chart [2], and Problem Analysis Diagram (PAD) [3] are much more effective to teach structured programming than a classical flow chart, since each stepwise refinement process by top down should be either concatenation, selection or repetition. Especially PAD is suitable to describe complicated programs directly.

However these charts do not contain declaration. It is important in object oriented programming to see how classes are declared. Then we propose a modified PAD including declaration.

II. INSERTING DECLARATION

In order to introduce PAD, consider the following C program containing pseudo-codes which finds the maximum and the second maximum number from ten input numbers.

```
//Program 1
#include <stdio.h>
int main(){
    int x[10],x1,x2;
    <* input x *>
    <* (x1, x2) from (x[0], x[1]) *>
    for(i=2; i<10; i++)
        <* (x1, x2) from (x1, x2, x[i]) *>
    <* output x1, x2 *> return 0;
}

<* (x1, x2) from (x[0], x[1]) *>:={
    if(x[0]<x[1]){x1=x[1]; x2=x[0];}
    else {x1=x[0]; x2=x[1];}
}

<* (x1, x2) from (x1, x2, x[i]) *>:={
    if(x2<x[i]){x2=x1; x1=x[i];}
    else if(x1<x[i]){x1=x[0];}
}
```

The corresponding PAD is shown in Fig. 1, where $B1 := \{x1=x[1]; x2=x[0];\}$, etc.

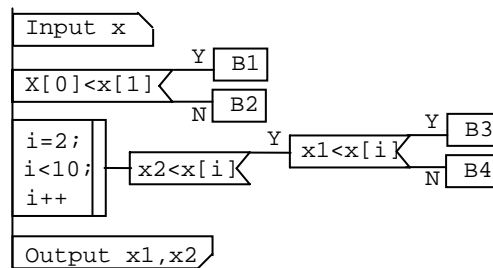


Fig. 1. Problem analysis diagram of Program 1

As shown in Fig. 1, declarations are not contained in PAD. In order to discuss how to include declaration in a structured programming chart, consider the following program.

```
//Program 2
#include <stdio.h>
int sum(int u, int v){return u+v;}
void add(int* p, int* q){*p += *q;}
int main(){
    int x,y; < * input x,y * >
    add(x, y);
    < * output x,sum(&x,&y) * > return 0;
}
```

According to the design policy:

- (1) All of the necessary information is included,
 - (2) Declaration blocks are *not* similar to execution blocks,
 - (3) Pseudo-codes are explicitly distinguished, and
 - (4) Easy to draw with popular application software,
- we propose a chart as shown in Fig. 2.

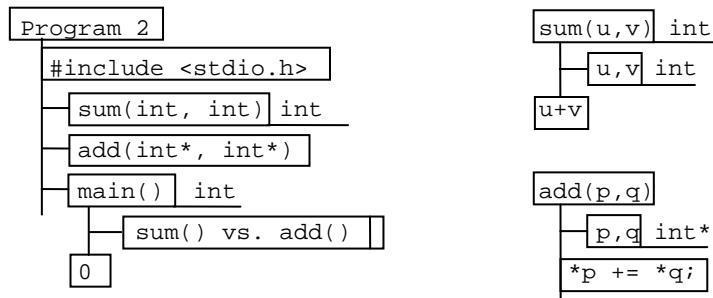


Fig. 2. Declaration and definition of Program 2.

Figure 3 shows a declaration of a class `cmpl` and the definition of its member function `add(cmpl*)`.

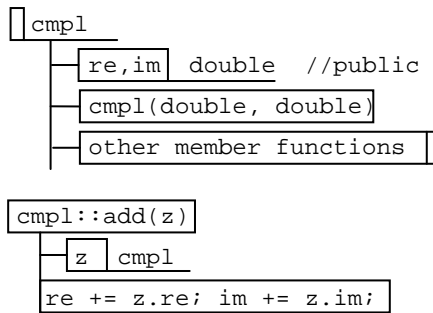


Fig. 3. A class and its member function

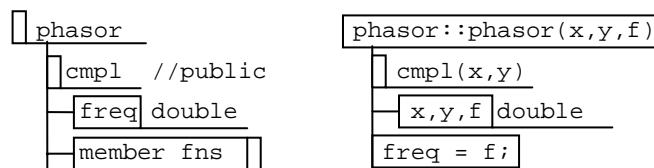


Fig. 4. A derived class and its constructor

A chart of derived classes is shown in Fig. 4. A sample of member functions is `phasor::sequence(FILE*)` which outputs the time-sequence of the sinusoidal wave.

III. ADDITIONAL MODIFICATION

The advantage to use a chart compared with a colored text is reduction of reserved words as shown in fig. 5, which implies large characters can be used in a lecture using MS Power Point slides.

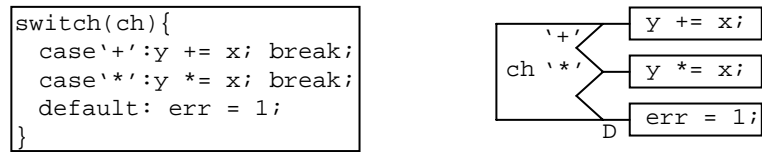


Fig. 5. Effect of eliminating reserved words

Although PAD is a compact expression of an algorithm as shown in Figs. 1 and 5, alternative expression in Fig. 6 is more compact and easy to draw, since every chart can be drawn by *superposing polylines on a uniformly spaced plain text*.

Remark `<* input x *>` in Program 1 represents each `x[i]` ($0 \leq i < 10$) should be filled by an integer using a certain method. An example of specified input/output is shown in Fig. 7 which correspond to `<* input(fscn):x *>` and `<* printf("%8d\n"):x1,x2 *>`.

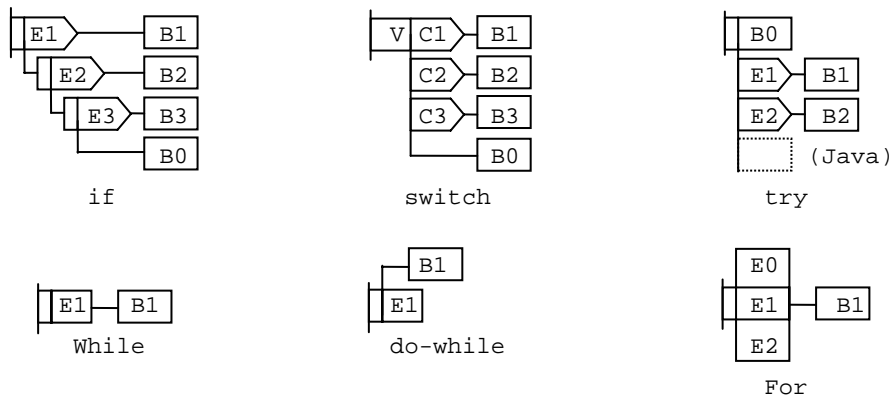


Fig. 6. Alternatives for compound statements

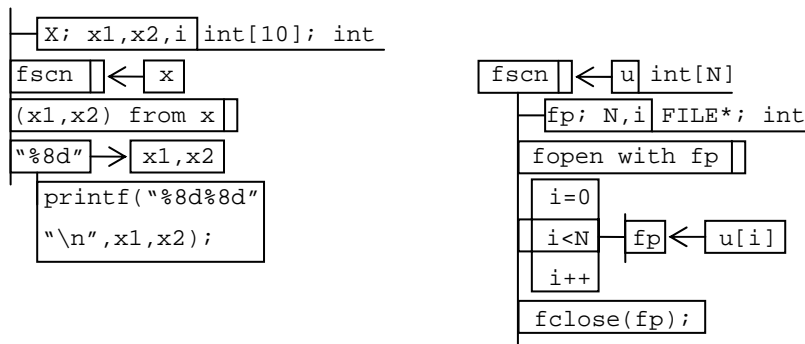


Fig. 7. An example of input and output for Program 1.

IV. CONCLUSION

The proposed chart is an object oriented programming chart rather than a structured programming chart. Since reserved words are reduced in such a chart, it is easier than a colored text to see how the algorithm is implemented.

REFERENCES

- [1] Weiss, E. H., "Visualizing a Procedure with Nassi-Schneiderman Charts", *Journal of Technical Writing and Communication*, Vol.20, No.3, 1990, pp. 237-254.
- [2] <http://www.cbu.edu/~lschmitt/I351/Nassi%20Schneiderman.htm> (Mark Kelly, *Structured Design with Nassi-Schneiderman Charts*)
- [3] Jackson, M. A., *Principle of Program Design*, Academic Press, 1975.
- [4] <http://www.informingscience.org/proceedings/IS2003Proceedings/docs/091Ourus.pdf> (N. Ourusoff, *Using Jackson Structured Programming (JSP) and Jackson Workbench to Teach Program Design*)
- [5] Futamura, Y., Kawai, T., Tsutsumi, M., and Horikoshi, H., "Development of computer programs by Problem Analysis Diagram (PAD)", *Proc. 5th Int'l Conf. on Software Engineering*, New York, IEEE Computer Soc., 1981, pp.325-332.
- [6] <http://fi.ftmr.info/PapersToRead/PAD-JARECT.PDF> (Y. Futamura and T. Kawai, *Problem Analysis diagram*)