

# A Summary Construct and Multiple Level Viewers in Natural Language Programming

Author : Weihan Huang

Master of Computer Science Department, State University of New York, at Buffalo, U.S.A.

Master of Physics Department, National Hsing Hua University, Taiwan

Email :weihanh@yahoo.com.tw

DOI: 10.26821/IJSHRE.10.1.2022.100111

## ABSTRACT

Natural Language Programming(NLP) is good for programming codes maintenance and understandability, however it cannot automatically summarize paragraphs of codes. Summary of paragraphs of codes can increase the code understandability a lot. Therefore in this article, a new Summary Construct is proposed, and then we compare it with two classical ways of summary expression. Next an idea of multiple level viewers is given, and this can hide details in multiple levels of program codes. Lastly, I write an algorithm of a multiple level viewer using the new Summary Construct.

**Keywords:** language, level, multiple, natural, programming, summary, viewer

## 1. REVIEW OF NATURAL LANGUAGE PROGRAMMING(NLP) [1]

### 1.1 Definition of Natural Language Programming

Natural Language Programming allows the variable place be in any position of the NLP sentence. And this makes NLP sentence be like natural language sentence. For example

fill \$array\$ with \$object\$

Because they are like natural language sentences, it is easy to understand, and easy to maintain.

### 1.2 Less Comments Need To Be Written.

In Java[2]/C++[3]/Prolog[4], we often write natural language documentations beside the codes. But in natural language programming, it merges the comments to the programming language syntax. For

example:

Java,C++, Prolog :  
father(x,y) // x is father of y

NLP :  
\$x\$ is father of \$y\$

### 1.3 Still Need Summary Of Codes Paragraph

Summary of codes paragraph shows what this paragraph does. It is very important for code maintenance and code understandability. However, as same as all other programming languages, NLP cannot automatically generate summary of codes paragraph. Programmers must add the summary by hands. In the following, I will show two classical ways of expressing summary of codes paragraph, and then I show how NLP gives a new construct of the summary of codes paragraph.

## 2. TWO CLASSICAL WAYS OF WRITING THE SUMMARY OF A PARAGRAPH

### 2.1 Comment

The first way of writing the summary of a paragraph of codes is to write a comment line right on the top of the paragraph. For example:

```
// paint the door  
paint the knobs of the door;  
paint the faces of the door;
```

The comment line starting with // gives the summary of what the following codes do. On the contrary, if we remove the summary line, then it would be hard to figure out what the goal of this paragraph is going to reach. So this again shows the importance of summary for code

understandability.

## 2.2 Function Call

The second way of writing the summary of a paragraph of codes is to write a function. The function name is just the summary of a paragraph of codes.

```
[function]
paint the door {
  paint the knobs of the door;
  paint the faces of the door;
}
```

Note that the summary should be a natural language sentence, and in NLP the function name is also a natural language sentence. So this just means that summary and NLP could coexist with each other.

## 3. A NEW WAY OF WRITING SUMMARY OF A PARAGRAPH

### 3.1 A New Summary Construct

```
do "summary" {
  sentence1;
  sentence2;
  sentence3;
}
```

or

```
do "summary" {
  sentence1;
  sentence2;
  sentence3;

  sentence4;
  sentence5;
  sentence6;
}
```

For example of "paint the door", it can be summarized as

```
do "paint the door" {
  paint the knobs of the door;
  paint the faces of the door;
}
```

This is a new way to write summary. i.e. A new Summary Construct in NLP.

### 3.2 Compare With The First Classical Way, Comment

The first classical way Comment can only take care of one paragraph. So if two paragraphs or more

paragraphs are needed to be summarized in one comment, then it cannot work.

On the contrary, the new Summary Construct can allow more than one paragraph inside the {} block.

### 3.3 Compare With The Second Classical Way, Function Call

Suppose the summary is used only once, then we don't need the function call, hence we can directly use the Summary Construct. Function call is for those codes repeated twice or more than twice. And also using the Summary Construct can be shown in multiple level viewers, which will be discussed in part 5.

## 4. PROBLEM OF SUMMARY OF VARIABLE DECLARATION

Suppose we have a summary of variable declaration as below:

```
do "declare variables" {
  new number $x$;
  new number $y$;
}

do "print variable values" {
  write a row $x$;
  write a row $y$;
}
```

Then there is a problem of variable scope. The scope of \$x\$ and \$y\$ are inside the summary construct of (do "declare variables"). The variables cannot be seen in the scope of (do "print variable values"). So compiler will give an error message: Undeclared variables \$x\$ and \$y\$. The way I solve this problem is by adding variables declared inside the summary construct to the outside scope (the parent block) in compile time. Therefore the variables inside summary construct of "print variable values" can reach their declarations in outside scope. And note that this is also valid in run time.

## 5. HIDING DETAILS IN MULTIPLE LEVELS

### 5.1 Example of Painting A House

Code P

```
do "paint the house" {
  do "paint the door" {
    do "paint the knob" { .. }
    do "paint the faces" { ... }
  }

  do "paint the walls" { .. }
```

```
do "paint the windows" {  
  do "paint the glasses" { .. }  
  do "paint the frames" { ... }  
}  
}
```

### 5.2 A Multiple Level Viewer

A multiple level viewer can show each level of codes, so that we can understand the codes by hiding details in multiple levels. Examples show level N expansion in the following.

#### 5.2.1. The level 1 expansion of the Code P is the Code P1 below

Code P1 :

```
do "paint the house" {}
```

#### 5.2.2. The level 2 expansion of The Code P is the Code P2 below

Code P2 :

```
do "paint the house" {  
  do "paint the door" {}  
  do "paint the walls" {}  
  do "paint the windows" {}  
}
```

#### 5.2.3. The level 3 expansion of the Code P is the Code P3 below

Code P3 :

```
do "paint the house" {  
  do "paint the door" {  
    do "paint the knob" {}  
    do "paint the faces" {}  
  }  
  
  do "paint the walls" {}  
  
  do "paint the windows" {  
    do "paint the glasses" {}  
    do "paint the frames" {}  
  }  
}
```

Therefore different levels of details can be hidden. In the code P2 of level 2, it is clear to see the high level abstract procedures or steps for a function. This is especially useful for algorithm illustration, in which we don't want to show too much detailed implementation. In natural language programming, the codes written in this way can hide the detailed implementation, and also the codes are still compliant and runnable.

## 6. ALGORITHM OF MULTIPLE LEVEL VIEWER

```
new number $target level$ is 2;
```

```
new editor $editor$ is(read editor from file);  
new number $current level$ is 1;
```

```
for each string $line$ in $editor$, do {  
  do "compute number of curly bracket in $line$" {  
    new number $left bracket number$ is 0;  
    new number $right bracket number$ is 0;  
    for each $character$ in $line$, do {  
      if($character$='{'), then {  
        $left bracket number$ increases by 1;  
      }else if($character$='}'), then{  
        $right bracket number$increases by 1;  
      }  
    }  
  }  
}
```

```
$current level$ increases by ($left bracket  
number$ - $right bracket number$);
```

```
if($current level$ <= $target level$), then {  
  write a row $line$;  
}
```

## 7. CONCLUSION

Summary of codes is important for increasing understandability of programming codes. In this article, I provide Natural Language Programming with a new Summary Construct, and show that this new Summary Construct is better than two classical ways of writing the summary of codes. Secondly, I give an idea of multiple level viewers of programming codes, which can hide details of different level of codes. Lastly, I write a NLP program of multiple level viewer that also demonstrates the use of Summary Construct.

## 8. REFERENCES

[1] Weihang Huang, (2021), "Natural Language Programming", International Journal of Advances in Electronics and Computer Science-IJAECS, Volume-8,Issue-9 (Sep,2021) pp15-17 Link to the paper :  
[http://ijaecs.iraj.in/paper\\_detail.php?paper\\_id=18189&nameNatural\\_Language\\_Programming](http://ijaecs.iraj.in/paper_detail.php?paper_id=18189&nameNatural_Language_Programming)

[2] [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

[3] C++  
<https://en.wikipedia.org/wiki/C%2B%2B>

[4] Prolog  
<https://en.wikipedia.org/wiki/Prolog>