Study on Program Reading System for Visually Impaired

Ryo Hirakawa*, Yoshihisa Nakatoh*

*Kyushu Institute of Technology
1-1, Sensui-cho, Tobata-ku, Kitakyushu-shi, Fukuoka, 804-0015, Japan

*nakatoh@ecs.kyutech.ac.jp

Abstract
The purpose of this study is to assist the visually impaired to create applications. There is a method for the visually impaired to develop applications by reading program codes with screen reader. In this method, the visually impaired can read code on line by line and select a line by knocking the up and down arrow key on the keyboard. However, such an operation takes them a long time with this method. In this paper, we suggested a new method which converts program code into flow chart and read blocks in the flow chart. In this method, the visually impaired can select blocks by knocking the left, right, up and down arrow key by keyboard. In addition, we proposed a new type flow chart for reading programs, because it is difficult to understand the program by reading the conventional flow chart. We compared the operation time between approach and the proposed method. As a result, it was confirmed that using the proposed method can understand programs about 40 to 50 percent faster than using the conventional method.

Keywords: Visually impaired, Programming, Flow chart

1. Introduction

One of the challenges of modern society is employment status of the disabled[1]. In particular, it is difficult for the visually impaired to be employed because they can get little information. Employers are demanding talent who can develop applications, and the visually impaired hope to be employed as programmer[2]). Therefore, we decided to assist the visually impaired to create applications. The most difficult task in developing applications by the visually impaired is getting information displayed on the editor such as source codes of the program. The visually impaired are dealing with this problem by read information displayed on the editor with screen reader. However, it takes them a long time with this method because it is difficult for them to grasp the source code which they really need. In this paper, we suggested a new method which converts program code into flow chart and read blocks in the flow chart.

2. Outline of the proposed system

In this paper, we suggest the new method for reading the program. In this method, the program is converted into the flow chart and users can read the program by reading flow chart. In addition, we redesign flow chart as reading programs, because it is difficult to understand the program by reading the conventional flow chart.

The visually impaired needed to read all processing of the program with the conventional method in order to chase the processing run. Because they cannot recognize the whole structure of the program visually. As the result, it takes them a long time to chase the processing with the conventional method. The proposed method leads to select the interested block of the program which is read by the visually impaired and they need not to read all the lines of the program. In addition, it is expected that the visually impaired can chase the processing of the program easily.

Furthermore, this method targets on reading the program written by C programming language.

2.1 Flowchart in proposed system

We explain the flow chart which has been redesigned as reading program. This flow chart assist the visually impaired to understand the program with three approaches.

(1) Adding blocks which mean start and end of branching statements.
(2) Integrating blocks of looping statements.
(3) Setting the structure of the program by using comment.
First approach is for converting branching statements into the flow chart. It is general that the program including branching statements like Figure 1(a) is depicted the flow chart like Figure 1(b). However, we redesign the flow chart as Figure 1(c). In this flow chart, blocks were added which mean start and end of branching statements instead of depicting graphically like arrows. Selecting the block before and after branch by knocking arrow keys was set prioritizing the flow of "if" over the flow of "else". In Figure 1(c), this system selects the block of "if" when users input down arrow key at the block of "int a=0". However, users cannot select the block of "else" directly at the block of "int a=0". By the same token, users cannot select the block of "end else" at the block of "printf("\n")", however, they can select the block of "end if" directly.

Second approach is for converting looping statements as Figure 2(a) into the flow chart. We redesign the flow chart as Figure 2(b) as Figure 2(c). In this flow chart, some blocks, from start of loop to end of loop, are integrated as "One of loop block". It assists users to image parts of being repeated easily.

Third approach is for the program set the structure by using comments. Generally, comment as “Declaration” in Figure 3(a) is not converted into the flow chart. Because the purpose of the flow chart is not assist users to understand the structure of the program, comments used for setting the structure on the program are not used in the flow chart. However, the purpose of flow chart for reading the program is to assist users to understand the structure of the program. Therefore, some blocks are integrated in this flow chart as Figure 3(b).

2.2 Operation method and behavior of reading system

In conventional method, the visually impaired can read code on a line-to-line basis and select a line by knocking the up and down arrow key by keyboard. However, the visually impaired can select blocks by knocking the left, right, up and down arrow key by keyboard in proposed method. The system in which is implemented the proposed method runs to repeat the following three steps.

(1) Waiting for knocking with keyboard by users.
(2) Selecting a block in the flow chart in compliance with key which users have inputted.
(3) Reading the processing of the selected block.

I explain the operation of proposed system with some examples. In the flow chart as Figure 1(c), this system reads block of "int a=0" at first. After that, this system waits for knocking with keyboard by users. If down arrow key is inputted, this system selects the block of "if a>5" and reads...
it. After reading blocks, this system waits input for knocking again. If right arrow key is inputted, this system selects the block of "else" and reads it. Next, we explain the flow chart including blocks in which is integrated some blocks as Figure 2(c) and Figure 3(b). If users input ‘z’ key at the block integrated, this system shifts the flow chart in the balloon in this flow chart. The flow chart in the balloon is what is subdivided the block integrated. If users input ‘r’ key, the flow chart which is read with this system shifts the flow chart in which some blocks are integrated.

Table 1 shows keys that are used in this system. If users input ‘x’ key, they can read a currently selected block again.

### 3. Verification of improvement of understanding by proposed system

We compared the time taken to understand programs by using the conventional method and the time taken to understand programs by using the proposed method. We experimented as shown Figure 4. First, we gave subjects the program having error and asked them to select a block which relates error of the program. Subjects read the program with proposed system. After that, we confirmed that they had selected a right block. If the answer is true, experiment is finished. Otherwise, we make them to select a block again. The purpose of this experiment is to verify that the proposed system is useful in debugging. Because the visually impaired read the program in debugging.

#### 3.1 Experiment method

Programs which are read in this experiments are sorted according to structure and difficulty as Table 2. There are three types of structure (included branching statement only, included looping statement only and included both statements). In addition, we setted the difficulty according to the number of step of the program and presence of nested structure. These programs ware prepared for experiment of conventional method and proposed method. The program for proposed method has same structure as he program for conventional method. "Same structure" means that the flow chart has same form as another one as Figure 5. In addition, the block which relates error of the program for the experiments of both methods was setted same hierarchial level like the block of "printf" in Figure 5.

Subjects are sighted persons, however, we made them not to be able to see PC screen in the experiment. We measure the time from start of the experiment to end of the experiment as index and consider the method which had need less time as good in debugging.

<table>
<thead>
<tr>
<th>key</th>
<th>Response of purpose system</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑, ↓, ←, →</td>
<td>Selecting the block in the inputted direction and reading it</td>
</tr>
<tr>
<td>X</td>
<td>Reading the selected block again</td>
</tr>
<tr>
<td>Z</td>
<td>Shifting from the integrated block to the flow chart which is subdivided the block integrated</td>
</tr>
<tr>
<td>R</td>
<td>Shifting from the flow chart which is subdivided the block integrated to the integrated block</td>
</tr>
</tbody>
</table>

#### Table 1. Relationship of inputted key and response

![Fig. 4. Flow of experiment](image)

![Fig. 5. Same structure and same hierarchical level](image)

#### Table 2. Types of programs

<table>
<thead>
<tr>
<th>Level</th>
<th>Branch</th>
<th>Repeat</th>
<th>Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level1</td>
<td>Not nested structure (About 40steps)</td>
<td>Not nested structure (About 40steps)</td>
<td>Not nested structure (About 40steps)</td>
</tr>
<tr>
<td>Level2</td>
<td>Nested structure (About 70steps)</td>
<td>Nested structure (About 70steps)</td>
<td>Nested structure (About 70steps)</td>
</tr>
<tr>
<td>Level3</td>
<td>Nested structure (About 100steps)</td>
<td>Nested structure (About 100steps)</td>
<td>Nested structure (About 100steps)</td>
</tr>
</tbody>
</table>
3.2 Results and discussion

In Figure, the measured time which is averaged for each subject is shown. Subjects could reduce time for understanding the program with proposed method.

Figure 6 is the result of each structure. In particular, time for understanding the program including branching statements decreased about 60%. Time of experiment reading the program including looping statements decreased about 40%. Time of experiment reading the program including both statements decreased about 45%.

Figure 7 is the result of each difficulty. In particular, time for understanding the program having 40 steps decreased about 40%. Time of experiment reading the program having 70 steps decreased about 42%. Time of experiment reading the program having 100 steps decreased about 53%.

Time which need to understand for the program including branching statements is improved greatly. The more complex program, the greater reduction of the time understanding for the program.

This reason is that users can select the processing which is read by the visually impaired and they need not to read all processing. This consideration is led by the result that the program including branching statements was improved greatly. The reason which subjects can understand complex program easier is that the outline of the flow can be understood easily by integrating some processing. Looping statements did not improved greatly. Because they cannot decrease the read processing compared to branching statements. Therefore, it is need to assist the visually impaired to judge easily that they have to read the integrated looping statements. The program which is low level difficulty can be understand easily. Because reading the processing obediently make the visually impaired not to take time. Because of this, it is need to assist the visually impaired to understand the program easily and not to increase futile process in reading simple program.

In this experiment, subjects made some mistake to input key. Therefore, it is need to improve operating the system which is accommodated usability as setting keys not to make mistakes and being able to input keys intuitively.

4. Conclusion

In this paper, we confirmed that proposed system is more useful than conventional method in debugging task because the time for understanding the program decreased. But looping statements and for simple programs are only a little improved.

In the future, we will improve the approach for them. For example, it is necessary to judge easily that they have to read the integrated looping statements. In addition, we will also improve usability in operating the system so that the visually impaired can operate easily.

References
