|  |
| --- |
|  |
|  |
|  |
| **Simple Merge**  **Software Requirement Specification** |
| Team number **#5** Team member 김성재, 박준영, 이현재, 주현준, 전진우 |

Content

1. Analysis
2. Design
3. Implementation

IV. Test Report

1. Analysis
2. Design
3. MVC modeling

* We design MVC(Model, View, Control) model.

1. Implementation
2. Control

* EditPanelController

In editPanelController, we control EditPanelModel(Model), EditPanel(View). Main function is Load, Save, Edit and this function is interact with MainWindowController.

•Save File

If click ‘Save’ button, we can save new file as original file or another named file. We get the string on the ‘EditPanelModel’(M of MVC), and copy this to new file or original file.

•Load File

In Load button’s ‘LoadActionListener’, when we choose a file and click ‘open’, system read one line by one line until meet ‘EOF’. And set file name on panel, set readed string on panel and other buttons(save, edit) is activated. After two file is loaded successfully, ‘merge’ button is activated. We modeled button action in ‘Control’, and set the contents in ‘View’ using ‘Model’.

•Edit File

Edit button is activated when file is loaded successfully. When file is loaded first, panel is uneditable. When edit button is clicked, we can edit the text on panel. And click again, panel is uneditable. Edit button control is in ‘Control’ area and ‘Control’ manipulate ‘editable’ and ‘uneditable’ function which is in ‘Model’ area.

* MainWindowController

Main is in this controller. All Swing component is designed not ‘thread-safe’. So approaching to swing component is acted on single-thread. This is called as ‘event-dispatch thread’. When we don’t result value and don’t care when task is ended. So we use EventQueue.invokeLater(Runnable runnable) method. When program is started, thread call the MVC. Main function is control the merge panel’s function(compare button, merge button, search button) and interact with EditPanelController.

•Compare

Compare the two loaded text file using ‘LCSubsequence’ and highlight the different section.

•Select(Up, Down)

When click ‘Up’ or ‘Down’ button, we search the highlighted section and highlight another color again.

•Merge(To Left, To Right)

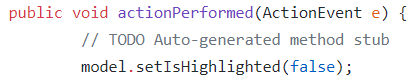
*\*In this section, all steps in Copy to Right is as same as steps in Copy to Left excepts only the index orders, so the steps in Copy to Left is described as detail as possible.*

After select the node to change, when the user clicks ‘Copy to Left’, there are two cases. 1. merge the selected node, which is colored as green in right section, to the section where correspond to positions that the node points in left section.

2. delete the selected node, which is colored as green in left section that means the context the node has is only in the left section, and re-compare it.

To be precisely, the steps are followed the below.

1. As soon as this button is clicked, the model’s highlighted is initialized.



1. Only after that the button “Compare” is pressed, the button “Copy to Left” or “Copy to Right” activates. If they are activated, and the user presses it, A node where is pointed by movements of UP/DOWN button is popped. Then, it is divided into three parts as string. The 1st one is head, the 2nd one is mid and the 3rd one is tail.

|  |
| --- |
|  |

1. Depends on where the selected node is in whether the left panel or the right panel, it computes differently.
   1. If the selected node is in the left panel, it means that the context of the node only exists in the left panel. It should be deleted even if the user presses the button “Copy to Left”
   2. If the selected node is in the right panel, it means that the context of the node only exists in the right panel. It should be inserted into the right position on the left panel even if the user presses the button “Copy to Left”
2. After finishing copying contexts, the nodes and highlights should be recomputed.

|  |
| --- |
| This is the same procedure of merging. |

* LCSubsequence

1. Model

1. Test Report