Change request log

Team
Baradji_Wei

Coding: Wei Xi

Documentation: Baradji Diallo

Change Request
Jedit----Change Request #2
Currently, jEdit displays the horizontal and vertical scroll bars whenever the content of the opened document exceeds the size of the editor. Implement an option in the View menu that allows to hide the scrollbars.

Concept Location
Use the table below to describe each step you follow when performing concept location for this change request. In your description, include the following information when appropriate:

- IDE Features used (e.g., searching tool, dependency navigator, debugging, etc.)
- Queries used when searching
- System executions and input to the system
- Interactions with the system (e.g., pages visited)
- Classes visited
- The first class found to be changed (this is when concept location ends)

When there is a major decision/step in the process, include its rationale, i.e., why that decision/step was taken.

Make sure you time yourselves when going through this process and provide the total time spent below.

The following is an example of a concept location process for the change request "Color student schedule":

Step #	Description	Rationale
1	*We ran the system*	
2	*We generated a file with a lot of context*	*To have enough words so that we can see the scroll bar and see if it's working*
3	*We expand view->scrolling menu*	*We want to see all the existed function with scrolling and decide which keyword to search for*
4	*We decide to search for the keyword "scrolling"*	
5	*From 1639 results, expanded the package org.git.sp.jedit*	
6	*We inspected the actions.xml*	*We notice that it has the highest match results*
7	*We inspected the jedit_gui.props*	*Since we need to add another option, we marked this as a potential starting point.*
8	*We went to org.git.sp.jedit.textarea*	*We found a lot of the actions in actions.xml calling the method TextArea.xx()*
9	*We inspected TextArea class*	
10	*We marked the class TextArea as "located".*	*We confirmed this file had to be modified.*

Time spent (in minutes): 15

Impact Analysis

Use the table below to describe each step you follow when performing impact analysis for this change request. Include as many details as possible, including why classes are visited or why they are discarded from the estimated impact set.

Do not take the impact analysis of your changes lightly. Remember that any small change in the code could lead to large changes in the behavior of the system. Follow the impact analysis process covered in the class. Describe in details how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

Step #	Description	Rationale
1	*We inspected the file jedit.gui.props and found that that's the file where we can add another menu option*	*To see if there are any classes that will be affected by our changes.*
2	*We inspected the file actions.xml and textarea.action.xml. these files were marked as "to change" as well. We found the actions are called from TextArea class*	*We realized all of the file had to be changed because we need to create new methods in TextArea for disabling/enabling the scrollbar which will be called in actions.xml and textarea.actions.xml*
3		

Time spent (in minutes): 10

Actualization

Use the table below to describe each step you followed when changing the code. Include as many details as possible, including why classes/methods were modified, added, removed, renamed, etc.

Step #	Description	Rationale
1	*We created a menu option called scroll-disable in jedit_gui.props and made a label for it called "Disable the scroll bar"*	*The current program does not have a menu option for disabling the scrollbar.*
2	*We created the method disableScrollbar in textArea.java*	
3	*We created new action in actions.xml and textarea.acitons.xml.*	*These actions calls disableScrollbar.*
4		

Time spent (in minutes): 20

Validation

Use the table below to describe any validation activity (e.g., testing, code inspections, etc.) you performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

Step #	Description	Rationale
1	*We created a file with really large context*	*Easier to observe the scrollbar*
2	*We went to (View->Scrolling->Disable the scroll bar), when we click it, there's a check mark before it and the scrollbar was hided*	*The test passed.*

3	*We click "Disable the scroll bar" again, the check mark is gone and the scrollbar appears again*	

Time spent (in minutes): 5

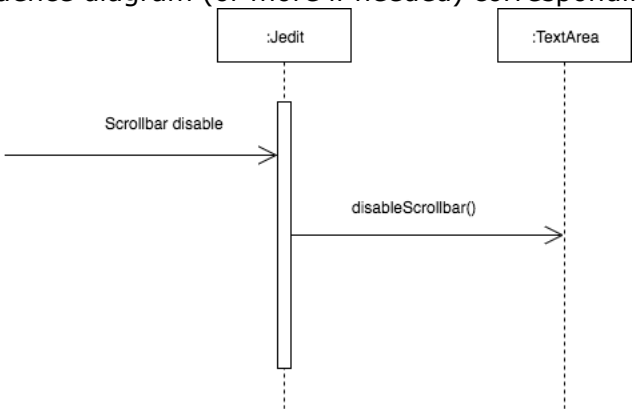
Timing

Summarize the time spent on each phase.

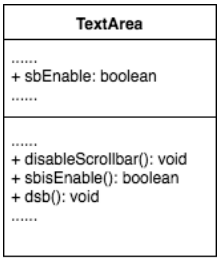
Phase Name	Time (in minutes)
Concept location	15
Impact Analysis	20
Prefactoring	NA
Actualization	20
Postfactoring	NA
Verification	5
Total	60

Reverse engineering

Create a UML sequence diagram (or more if needed) corresponding to the main object interactions affected by your change.



Create a partial UML class diagram of the classes visited while navigating through the code. Include the associations between classes (e.g., inheritance, aggregations, compositions, etc.), as well as the important fields and methods of each class that you learn about. The diagram may have disconnected components. Use the UML tool of your preference. When a significant fact about a class or method is learned, indicate it via annotations on the diagram. **For each change request, start with the diagram produced in the previous change request. For the first, you will start from scratch.**



Conclusions

Provide a set of conclusions about the change request and the change process. List the major challenges this change request posed.

List all the classes and methods you have changed.

For this change request, concept location was pretty easy because we just needed to find the Jscroll pane or Jscroll bar. Impact analysis took a little more time than expected because we were not very familiar with the system. There's a wired thing, we found that actions.xml and textarea.actions.xml is similar but it seems like the changes we made in textarea.actions.xml does not influence the whole software. We manually tested the functions by creating large files and click the new menu option and we think it works well.

Classes and files changed:

`org/gjt/sp/jedit/actions.xml`

`org/gjt/sp/jedit/jedit_gui.props`

`org/gjt/sp/jedit/textarea/TextArea.java`

- `void disableScrollbar()`
- `boolean sbisEnable()`
- `void dsb(Boolean sbEnable)`

`org/gjt/sp/jedit/textarea/textarea.actions.xml`