Change request log

Team & role

Kartikay Sharma: Implemented the change

Swetha Varadarajan: Wrote the document, Verified the process.

Change Request

Currently, Jedit displays the horizontal and vertical scroll bars wherever the content of the document exceeds the size of the editor. Implement an option in the View menu that allows to hide the scrollbars.

Concept Location

Step #	Description	Rationale
1	We ran the system using the Eclipse IDE	
2	We interacted with the system: after logging in we entered the schedule screen.	To get familiar with some of the features of the system, and identify the screens or graphical elements we had to change.
3	When implementing Change request #1, we looked at other toggle options in View menu for help and somehow stumbled upon toggle gutter. This helped in concept location and actualization of Change request #2	
4	Used the keyword “scroll” in Instasearch tool of Eclipse IDE	
5	Out of 20+ listed files, we picked ScrollLayout.java and inspected the code.	This file name made more sense to the change request we are planning to make.
6	We noticed that this class is using the package textarea. So, we ran a search query using “textarea” as the keyword in instasearch tool.	
7	There were about 15+ files enumerated out of this search. The main file textarea.java was very lengthy to examine, So, to prune the search, we used the keyword “jedittextarea”	Although the keyword matches the concept we want, it didn’t make sense to inspect such huge files. So, we pruned the search space by using alternate keyword. We did spend some time inspecting few of the files and ignoring them before making another query.
8	On inspecting JeditTextArea, we saw that it extended TextArea. We also realized that TextArea class has so many dependencies (shown in UML diagram in Figure 1)	This is because, the text area class is been used to define all the features of jedit editor window. It makes sense to include feature in this one file.
9	We marked the class TextArea as "located".	We confirmed this class had to be modified.
10	We had to reiterate the concept location after the first actualization phase. Using our lucky moment from step 3, we searched for “toggle gutter”. We came across three files actions.xml , jedit_gui.props and jedit.props	We confirmed that these three classes also need to be changed.
11	We had to reiterate the concept location after second actualization phase. Because, the jedit had features supportive of various international language and this	In general, we marked “jedit_*.props” files for change.

	change has to be reflected in files created for different languages.	

Time spent (in minutes): 60

Impact Analysis

For this exercise, we had to do impact analysis iteratively. The first impact analysis was simple to justify.

Impact Analysis 1: Since TextArea file is a file that includes all the properties, adding a class in this file is not going to impact any other file.

Impact Analysis 2: In the iterative process, we identified three other files to make change: actions.xml , jedit_gui.props and jedit.props. Again, since these were property files, we concluded that changing in these files will not make any modification to the entire project.

Impact analysis 3: Changing property files of individual language again doesn't have much impact on the whole system.

Time spent (in minutes): 25

Prefactoring (optional)

We concluded that there is no prefactoring required for this change request.

Time spent (in minutes): 5

Actualization

Step #	Description	Rationale
1	Actualization phase 1: Including the functionality in Action file for toggle_scrollbar	We ran the code with unit test. The build was successful. But, we realized that the functionality was not seen to be implemented. We restarted with the concept location phase.
2	Actualization phase 2: Included the necessary changes in action.xml, jedit.props and jedit_gui.props file. We adopted the code that was used for other features.	We ran the code with unit tests. The build and running was successful.
3	Actualization phase 3: We included the necessary changes in other language files.	We ran the code with unit tests. The build and running was successful.

Time spent (in minutes): 25

Post-factoring (optional)

We decided that this change request doesn't need any post-factoring.

Time spent (in minutes): 5

Validation

Step #	Description	Rationale
1	Test case defined: Manual test cases Input: Click on the toggle scroll bar Output: Scroll bar should appear or disappear according to its initial state.	This is the regular expected behavior. The test passed.

Time spent (in minutes): 5

Timing

Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	60
Impact Analysis	25
Prefactoring	5
Actualization	25
Post-factoring	5
Verification	5
Total	125

Reverse engineering

Figure 1 shows the complex dependency structure involved when editing the TextArea file. But, with intuition, we can tremendously reduce this complexity. Figure 2 shows the classes that we modified for both change request #1 and #2. It shows that these two classes don't interact.

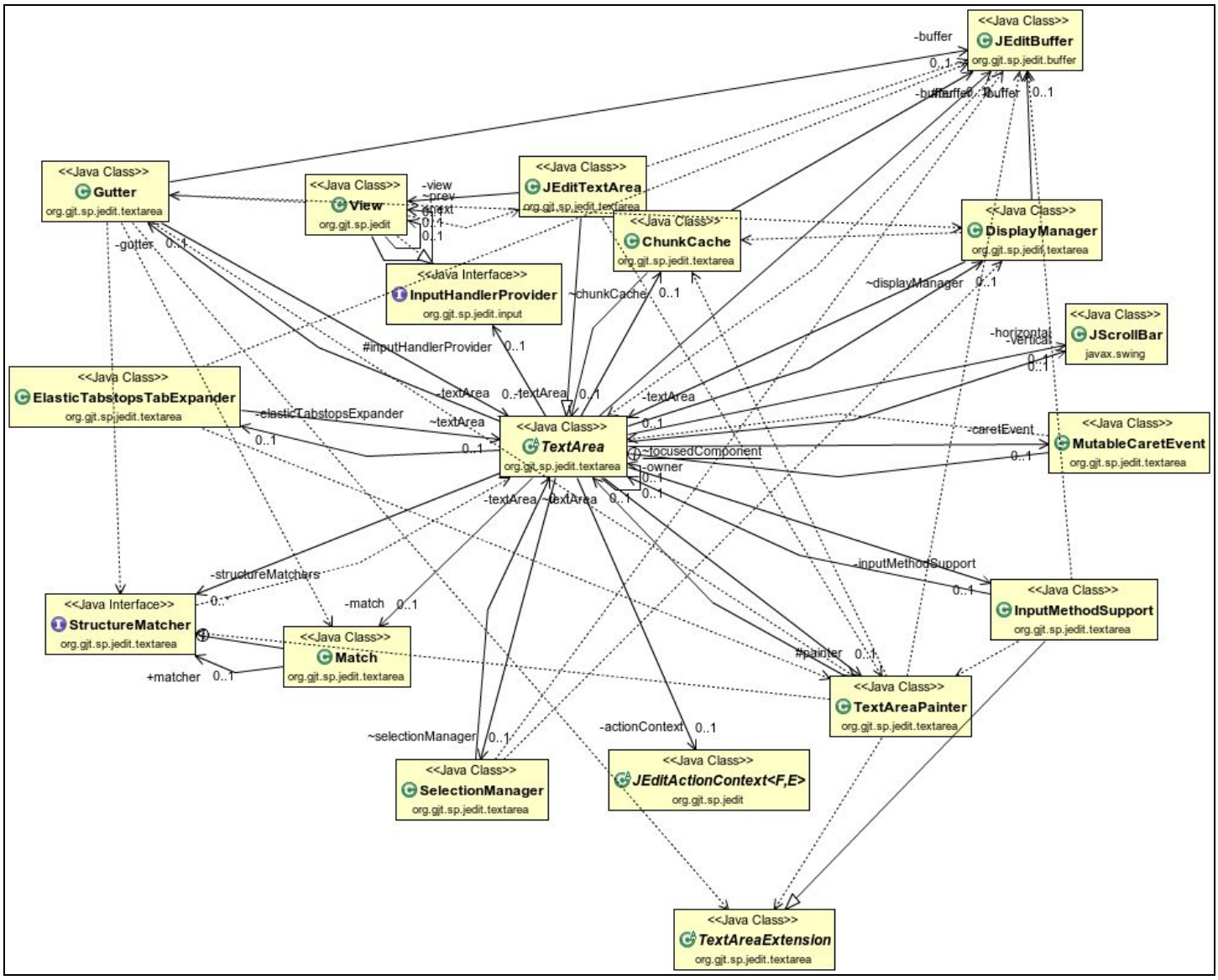

Figure 1: UML showing the complexity involved when editing the TextArea class.

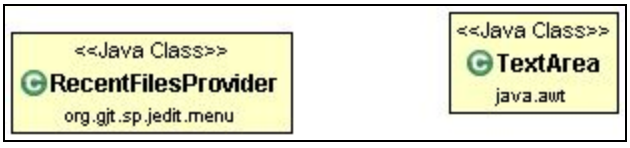

Figure 2: UML showing no interaction between the classes that we used for Change request #1 and #2

Conclusions

For this change, concept location took some time as the classes were lengthy. The Impact Analysis was very complex. It can be seen from the first UML diagram. But, intuitively, this step can be made easy and we saw that it worked. We had to iterate the concept location, impact analysis and actualization steps as it was not very obvious to locate all the places to change in the first phase. Although the change was easy after we completed it, the dependency in this system was very cryptic and we believe that the software change process had to be done iteratively rather than strictly following the steps.

Classes and methods changed: Here we mention only the classes that we changed. One can refer the github commits to follow which method/ line in each class was modified.

- org/gjt/sp/jedit/actions.xml
- org/gjt/sp/jedit/jedit.props
- org/gjt/sp/jedit/jedit_gui.props
- org/gjt/sp/jedit/textarea/TextArea.java
- org/jedit/localization/jedit_cs.props
- org/jedit/localization/jedit_de.props
- org/jedit/localization/jedit_en.props
- org/jedit/localization/jedit_fr.props
- org/jedit/localization/jedit_ja.props
- org/jedit/localization/jedit_ko.props
- org/jedit/localization/jedit_ru.props
- org/jedit/localization/jedit_zh.props