Change request log

Team

Baradji_Wei

Coding: Baradji Diallo

Documentation: Wei Xi

Change Request

Change Request #1

- In the File » Recent Files main menu of jEdit, the text box on top of the recent files list allows to highlight recent files names that match with a given string (see Figure 1). The string in the text box should match all the files that contain it anywhere in their name. However, the highlight works only when the string matches the beginning of a file name. You are requested to modify this feature so that the highlight occurs for the cases when the string is contained anywhere in the file name.

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.

Step #	Description	Rationale
1	*We ran the system*	
2	*We interacted with the system after running it.*	*To identify the screens or graphical elements we had to change.*
3	*We used eclipse Search function (ctrl h) to search for files containing the text "recent file".*	*Because we identified a menu option on the screen called "Recent Files"*
4	*From 33 results, we used eclipse IDE dependency navigator to check different folders containing the searched keyword.*	
5	*We inspected the first 3 files (Usage Questions, The Buffer Options Dialog Box, Working With Files)*	*The first 3 files had the keyword (recent file) we were looking for; however, they were mostly text documents. Therefore, we concluded these files were not relevant for our change request.*

	We went down the list until we found a class called RecentFilesProvider. We clicked on the class RecentFilesProvider for inspection.	*After reading the tittle, we had a feeling that might be the class we are looking for.*
6	*While inspecting the class RecentFilesProvider by scrolling through it, we noticed a comment that stated, "// Old style (before jEdit 4.3pre18): Match start of file name." This led us to believe we are at the right location.*	
7	*We marked the class RecentFilesProvider as "located".*	*We confirmed this class had to be modified.*

Time spent (in minutes): 45

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 used the eclipse IDE to do a dependency check on RecentFilesProvider.*	*To see what other classes might get affected if we made a change.*
2	*We inspected the class EnhancedMemu*	*The update method in RecentFilesProvider was being called in EnhancedMemu. However, this class was mark as potential impact point, but no change needed. Since it wasn't modifying anything.*
3		
4		

Time spent (in minutes): 12

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 modify the method update(JMenu menu).*	*The old code, highlight works only when the string matches the beginning of a file name. Therefore, we had to modify it; so, highlight occurs for the cases when the string is contained anywhere in the file name*
2		

3		
4		

Time spent (in minutes): 10

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 did manual testing by interacting with the system.* *Step1: run jEdit* *Step2: In the main menu of jEdit, navigate to File >> Open…* *Step3: Navigate to the file you want to open* *Step4: Select the file and press open* *Step5: Repeat steps 2-4 for all the files you want included in your recent files history.* *Step6: Navigate to File >> Recent Files of jEdit* *Step7: In the text box on top of the recent files list, search for any string you want. The matching files containing the string will be highlighted.*	*The program is old, hard to add test cases, and manual testing seems easier.*

Time spent (in minutes): 20

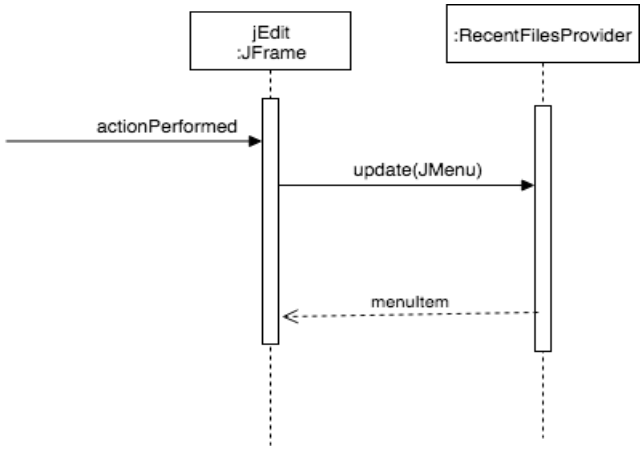
Timing

Summarize the time spent on each phase.

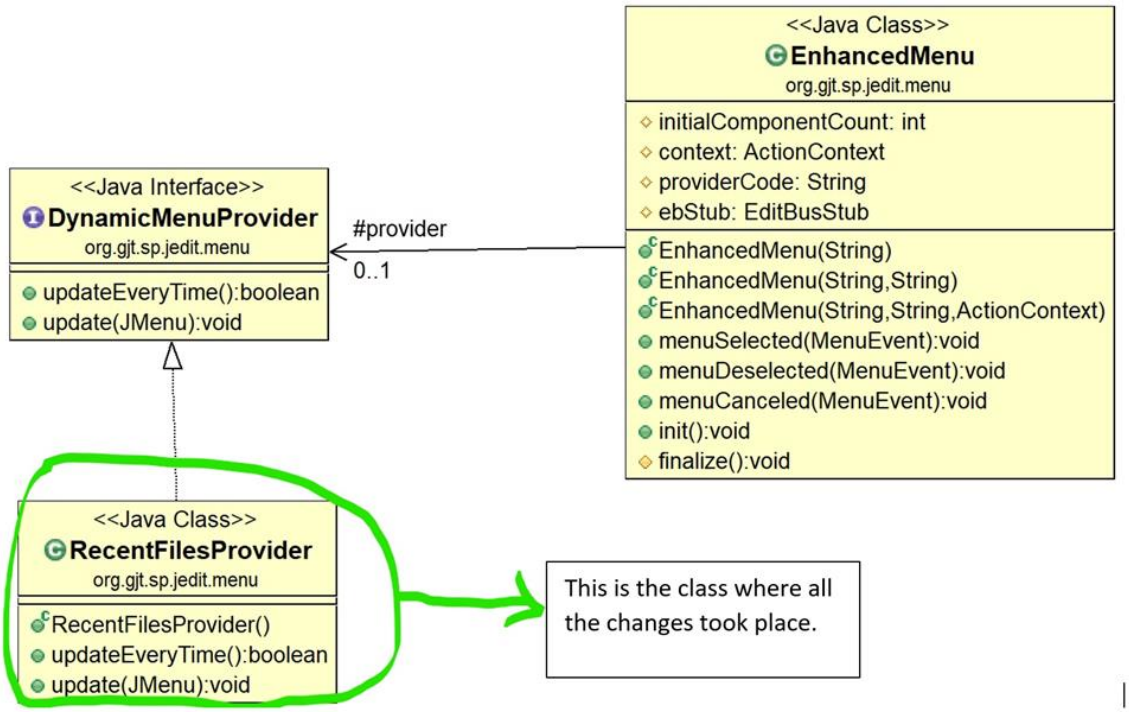
Phase Name	Time (in minutes)
Concept location	45
Impact Analysis	12
Prefactoring	n/a
Actualization	10
Postfactoring	n/a
Verification	20
Total	87

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, building the program was the hardest to do. I wasn't familiar with JEdit; therefore, the errors did not make much sense to me. Thanks to google, I was able to find a way to get it going. Additionally, the concept location was the second hardest because I did not know the program that well. However, the architecture and the code were not complicated which made impact analysis and actualization a little bit easier. Testing was performed manually because the program is old and hard to add tests cases. Therefore, manual testing seems to make more sense.

Classes and methods changed:

- *org/gjt/sp/jedit/memu/RecentFilesProvider*
    - *void update(JMenu menu)*