Change request log

Team
Max Roselius

Change Request
#1. This change request aims to modify the recent files search filter such that the string entered in the search box should match all files that contain it anywhere in their name. Currently, only files that begin with the search string will be matched.

Concept Location

Step #	Description	Rationale
1	*Build the system from its current state, and examine the current functionality.*	*To gain familiarity with the system, and look for potential search terms that will lead to the concept*
2	*Create a few sample files and visit them to populate the Recent Files list*	*Need to have some files to test searches against*
3	*Using Eclipse search tool, searched for term "search"*	*Get a starting point of places to search for concept location*
4	*Got over 2400 matches, need to narrow search space*	*2400 matches is way too many to manually investigate*
5	*Inspect search bar, find a tip pop-up saying "a filter prefix or glob-pattern may be entered here". Search workspace for this term*	*This term should serve as a starting point for an object that displays this string*
6	*Found jedit_en.props with some terms including "`recent-files.textfield.tooltip`". Search for "`recent-files`"*	*"recent-files" may lead to a comment or reference to a class that links to this line in the configuration file*
7	*Search returned 3 matches in the RecentFilesProvider class.*	
8	*Examined the class code and found search logic in the update() method*	*This appears to be the concept location as this method is called each time a new character is entered in the filter search box, and looks through all the files in the recent files list to compare*
9	*Marked the class RecentFilesProvider as "located".*	

Time spent (in minutes): 55

Impact Analysis

Step #	Description	Rationale
1	*Examined the search logic for any method calls on other objects*	*If any other objects are used, need to ensure they are not affected by a change*
2	*Examined keyReleased() method, which is called any time a new character is entered in the search box.*	*This is exact the location where the change should be made, need to ensure that modifying the filter string will not propagate changes to other system objects*
3	*The only objects touched during this method*	

	call are Jmenu items, but they are not modified in any way aside from being highlighted if their name matches the user-supplied filter.	
4	*This class is the only one that requires a change.*	*No other objects are touched or modified.*

Time spent (in minutes): 10

Actualization

Step #	Description	Rationale
1	*Current code version had a convenient comment stating how appending * to the current search term matches any file starting with the term.*	
2	*Modify line 109 of the file to prepend * to the front of the search term as well.*	*Temporary search term now contains wild card at beginning and end of it, causing the pattern to match any string (file name) that contains the search term anywhere.*
3	*Recompile source to test change.*	*To make sure everything works.*

Time spent (in minutes): 10

Validation

Step #	Description	Rationale
1	*Populate the recent files list with a few files. Namely, "sampleFavorite, sampleFavorite2, results.ods, and Untitled-2"*	*Need files to test on*
2	*Enter search term "avor". sampleFavorite and sampleFavorite2 should match.*	*This term is contained in the middle of two file names, and those files should match the search term. The test passed.*
3	*Enter search term "-2". Only Untitled-2 should match.*	*A possible edge case containing non alphanumeric character. Test passed.*
4	*Enter search term "*s". Only results.odt should match.*	*Need to test that user-defined wild-card character search is not affected by change. Test passed.*
5	*Enter search term "s*". Only sampleFavorite and sampleFavorite2 should match.*	*Same reasoning as above. Test passed.*

Time spent (in minutes): 15

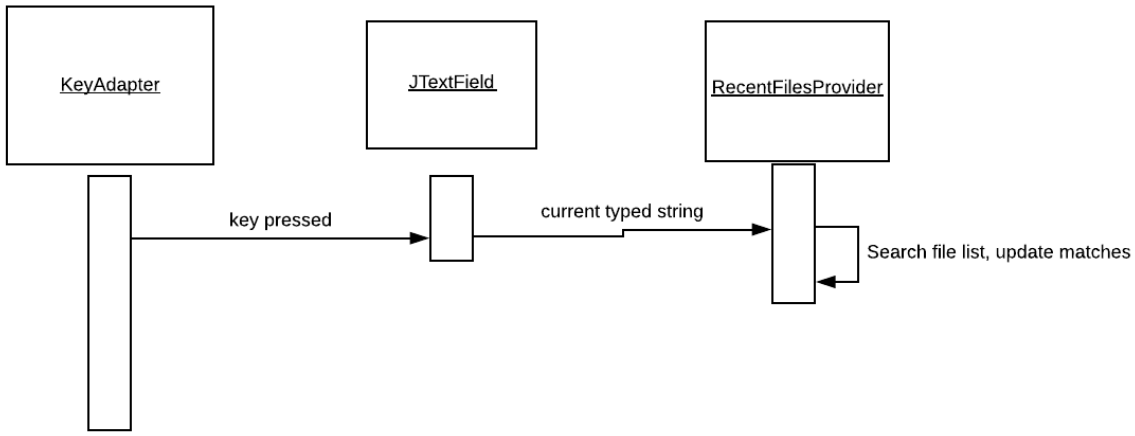
Timing

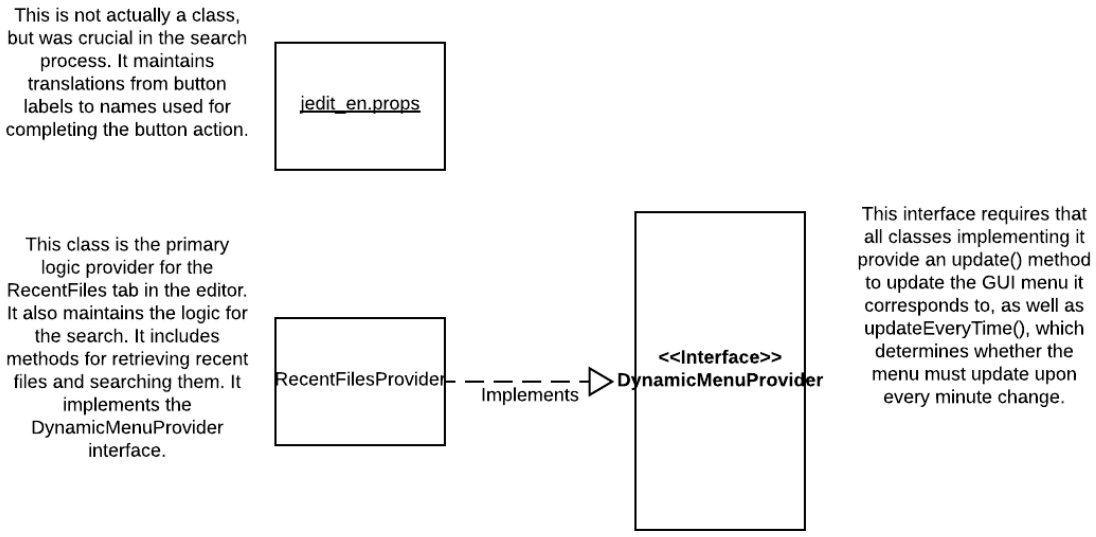
Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	55
Impact Analysis	10
Prefactoring	NA
Actualization	10
Postfactoring	NA

Verification	15
Total	90

___Reverse engineering





___Conclusions

For this change, concept location took the longest. This was due to a couple of searches for terms that were not helpful (“search”). I ended up browsing through quite a few classes that contained the term “search” in hopes of finding something relevant, and this turned out to be a waste of time. Once I examined the text box where searches can be entered (in the Jedit text editor), I found the tip text, which

gave me a great search term and led me to the concept fairly quickly. The change itself was very easy to implement, and only involved changing a single line. This made impact analysis rather easy as well, since no other classes or methods were used/called that could propagate the change.

Classes and methods changed:

- *org/gjt/sp/jedit/menu/RecentFilesProvider*
  - *keyReleased()*
  - *Technically changed update(), since keyReleased is defined within update()*