

Change request log

Team

Baradji_Wei

Coding: Wei Xi

Documentation: Baradji Diallo

Change Request

Change Request #1

- The Alternate Mix and Merge modules of PDFsam provide buttons to move a document up or down in their respective lists of files. However, these buttons are not ideal when the list of documents is long, and the user wants to move one of them to the top or the bottom. You are requested to add two new buttons in these modules, to allow the user to move a selected document to the top and bottom of the list.

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	<i>We ran the system</i>	
2	<i>We interacted with the system after running it.</i>	<i>To identify the screens or graphical elements we had to change.</i>
3	<i>We used eclipse Search function (ctrl h) to search for files containing the text "Move Up".</i>	<i>Because we identified a button in the screen called "Move Up"</i>
4	<i>From 94 results, we used eclipse IDE dependency navigator to inspect the class SelectionTable.</i>	<i>contained the searched keyword.</i>
5	<i>We inspected the class MoveType.</i>	<i>While inspecting the class SelectionTable, we noticed that different MoveType was being call for each menu item. We realize this class might have something to do with our change request.</i>

6	<i>We inspected the class SelectionTableToolbar.</i>	<i>While inspecting the class MoveType, we used the eclipse IDE to open the call hierarchy of the function UP. While scrolling through the options we had, we noticed there were a MoveUpButton method in SelectionTableToolbar.</i>
7	<i>We marked the class SelectionTableToolbar as "located".</i>	<i>We confirmed this class had to be modified.</i>

Time spent (in minutes): 60

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	<i>We used the eclipse IDE to open the call hierarchy of the method MoveUpButton.</i>	<i>To track the classes or methods that could be impacted when we add 2 new buttons (Move to top, Move to bottom).</i>
2	<i>We inspected the class SelectionTableToolbar. Such class was marked as to be change.</i>	<i>We realized this class had to be changed because it was responsible for displaying the buttons on the screen.</i>

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	<i>We created the class MoveToTopButton.</i>	<i>Allow the user to move a selected document to the top of the list.</i>
2	<i>We created the class MoveToBottomButton.</i>	<i>allow the user to move a selected document to the bottom of the list.</i>
3	<i>We modified the class SelectionTableToolbar.</i>	<i>To make sure it displays the 2 new buttons added.</i>

Time spent (in minutes): 30

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	<i>We did manual testing by interacting with the system. Step1: run PDFsam Step2: click on "Alternate Mix" Step3: click on "Add" and navigate to the document you want to add. Step4: repeat step3 as nessary Step5: click on the document you want to move to the top or bottom Step6: click on "Move to top" or "Move to bottom" base on the action you want to perform</i>	<i>To make sure the system is functioning as we expected. (was more convenient at the time)</i>

Time spent (in minutes): 14

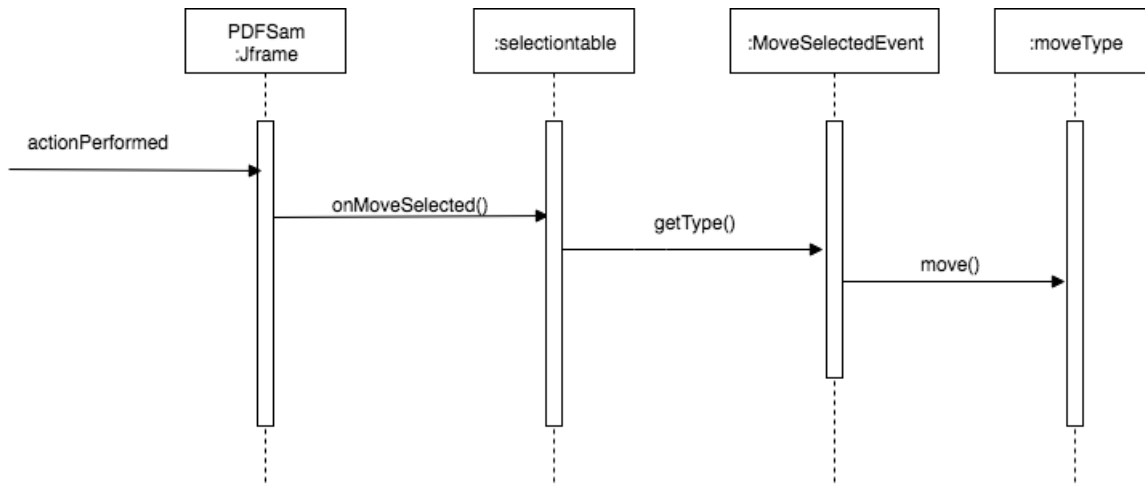
Timing

Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	60
Impact Analysis	12
Prefactoring	n/a
Actualization	30
Postfactoring	n/a
Verification	14
Total	116

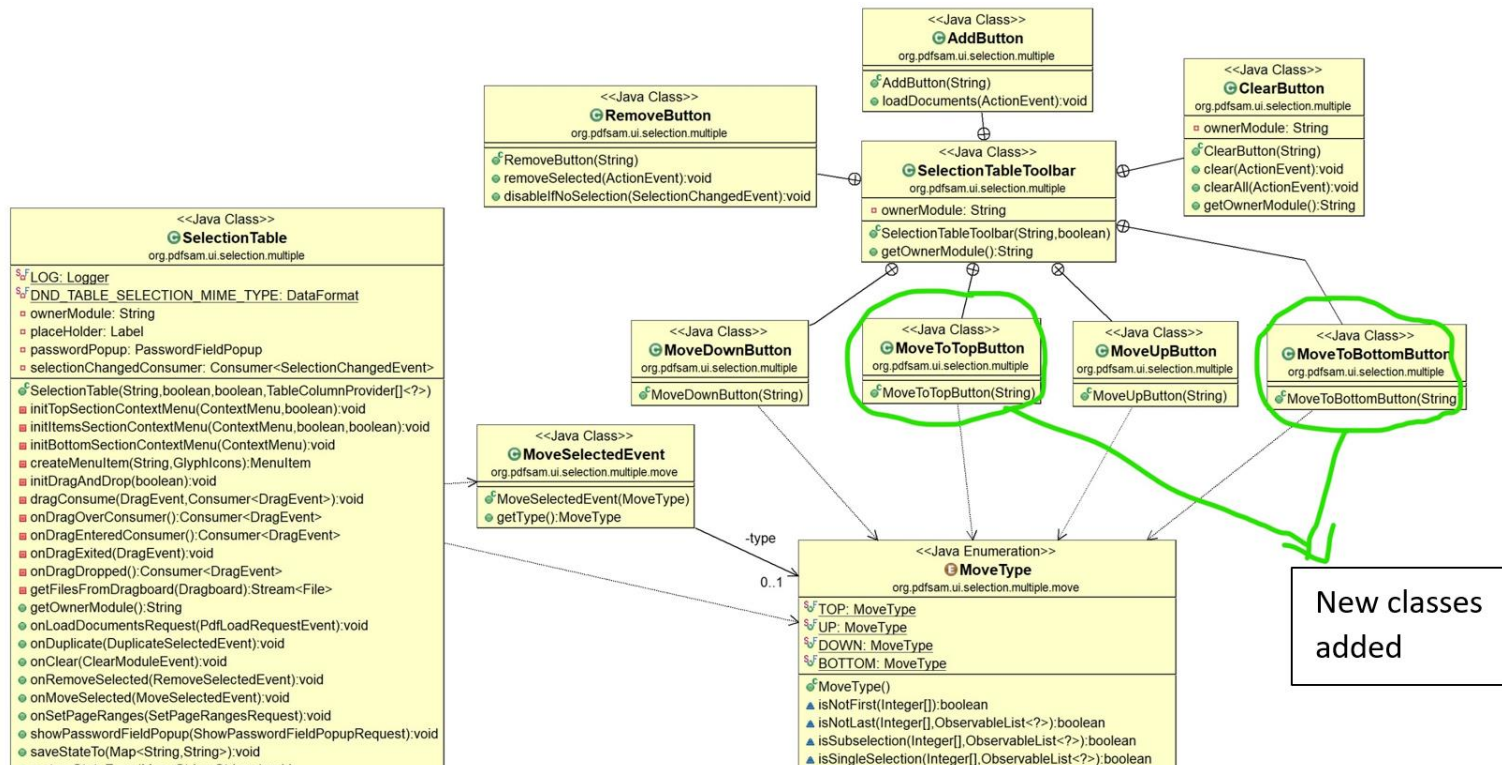
Reverse engineering

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 very difficult because the error messages did not make much sense. 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 it was more convenient at the time.

Classes and methods changed:

- pdfsam-fx/src/main/java/org/pdfsam/ui/selection/multiple/SelectionTableToolBar
 - SelectionTableToolBar(String ownerModule, boolean canMove)
 - MoveToTopButton(String ownerModule)
 - MoveToBottomButton(String ownerModule)