

Change request log

1 Team

Team name:

Fari-Marylou

Team members and roles:

- Marylou Nash

Role: Investigate & fix **ps3**. Document the investigation phase and steps taken to solve the problem. Complete the change request log form.

- Farzaneh Kadkhodaie

Role: Investigate the fix for **ps3**. Provide support. Answer questions and help evaluate ideas. Review and verify fix.

GitHub location:

The change request was implemented by creating a branch off the v4.0.5 tag of PDFSam code. The branch was named v4.0.5-master. The final version of the code containing all PDFSam change requests was merged into the v4.0.5-master branch.

2 Change Request

The Rotate module allows the user to rotate individual pages of multiple PDF documents in two ways:

- Providing a range of pages to rotate, which can be done for each document; or
- Selecting even, odd or all pages to rotate for those documents whose range was not defined.

Currently, these two options are mutually exclusive although this is not obvious to the user just looking at the GUI and menu choices. Once the user has selected a specific range of pages, the menu settings for EVEN_PAGES and ODD_PAGES are not applied within the selected rotation ranges.

The **ps3** change request wants to alter this behavior and have the EVEN_PAGES/ODD_PAGES apply to the custom ranges selected in the “Page ranges” column of the menu.

3 Concept Location

- IDE Features used (e.g., searching tool, dependency navigator, debugging, etc.)

- The search feature in IntelliJ was helpful locating words and phrases. However, the naming conventions and small number of classes and methods that were focused on the Rotation task made it easy to just look through the code and hone in on the methods that were useful in implementing this change request.
- PDFSam relies on the open source library, Sedja, to accomplish each task with the PDF document. It wasn't very easy to find detailed documentation about Sedja.
- We found IntelliJ's debugger to be a quick reliable way to determine the contents of objects and what information was being passed in from the calling routine.
- The IDE also provided a list of methods available for each object. This was a quick way to determine what methods were available in the Sedja and other objects.

Step #	Description	Rationale
1	<i>The four classes associated with the rotate task located in the pdfsam-rotate directory were – RotateModule, RotateParameterBuilder, RotateOptionsPane, and RotateSelectionPane.</i>	<i>PDFSam offers 7 different tasks that it will perform for the user. Much of the code is divided into classes that are very obviously specific to each of these tasks. Since this change request is unique to the Rotate task we began by looking through the four classes that were obviously related to the Rotate task.</i>
2	<i>The RotateModule and RotateParameterBuilder classes were focused on actually performing the task.</i>	<i>Comments and code in the classes RotateOptionsPane and RotateSelectionPane (as well as the class names) made it obvious that these two classes were focused on implementing the Rotation task menu page.</i>
3	<i>Enabled the debugger and created several test cases to watch and see the contents of the objects PdfSource and Set<PageRange> to learn what values were being passed to Sedja routines. Each of the page ranges as specified by the user were simply passed to Sedja. The</i>	<i>The method addInput() with its two parameters – source and pageSelection looked like a good method to examine and see what values were getting passed to the Sedja routines to perform this task.</i>

	<i>EVEN_PAGES and ODD_PAGES was also set correctly based on the menu. From this it was determined that Sedja was simply ignoring the EVEN/ODD setting.</i>	
4	<i>It was decided to add a method in the RotateParameterBuilder class that could be called to adjust the pages.</i>	<i>A quick, simple way to fix this would be to prefilter the pages before sending them to Sedja.</i>
5	<i>Use the IDE's debugger to determine what objects on the stack contain the information we are seeking about the total quantity of pages. The RotateSelectionPane contained information about the total quantity of pages. Using the debugger again, we were able to find a method from the Sedja library to access and extract this information so we could send it to the parameter builder.</i>	<i>To filter page ranges that were specified to the end of the file (e.g. 5-), we would need to know the total number of pages.</i>

Time spent (in minutes): 30

4 Impact Analysis

Step #	Description	Rationale
1	<i>A new method would need to be developed to filter the page ranges. This could be instantiated in RotateParameterBuilder and called just prior to sending info to Sedja. It would modify the SET of pages that was sent to Sedja.</i>	<i>The page range sent to the Sedja library routines could be altered to fix this and obtain the correct behavior.</i>

2	<i>addInput() will be modified to call the new method that filters the EVEN or ODD pages located in the page ranges specified by the user in the menu.</i>	<i>The parameter builder (i.e. addInput()) will be impacted since it needs to fix the page formatting.</i>
3	<i>The apply() method in RotateSelectionPane invokes the addInput() method in RotateParameterBuilder and will need to be modified.</i>	<i>RotateSelectionPane would be impacted since it needed to send info about the document size.</i>
4	<i>Fix any affected junit tests that fail as a result of the interface change in addInput().</i>	<i>Since the interface of the addInput() method in RotateParameterBuilder is changing, some junit tests may fail.</i>

Time spent (in minutes): 60

5 Prefactoring (optional)

Step #	Description	Rationale
1	<i>No prefactoring was done.</i>	<i>This change should just require a simple method to adjust the pages only when there are custom page ranges and EVEN or ODD is selected.</i>

Time spent (in minutes): 0

6 Actualization

Step #	Description	Rationale
1	<i>Created a new method to alter the contents of the SET containing the page ranges based on the EVEN/ODD setting in addInput() in the RotateParametersBuilder class.</i>	<i>Sedja would be able to handle the custom page range 3-9, if it was sent as 3,5,7,9. So, we can adjust the representation of the page range to get Sedja to produce the correct output.</i>

2	<p>No alteration is needed for a single page.</p> <p>Alteration for a page range is straightforward.</p> <p>Alteration for the opened ended case (7-) proved to be more difficult. It would require knowing the total pages.</p>	<p>There were three cases that could appear in the <code>pageSelection</code>. These were a single page (7), a range of pages (7-11), or an opened ended range (7-) that indicated all pages to the end of the document were to be included.</p>
3	<p>In the <code>addInput()</code> method the total quantity of pages is unknown. We had to pass that information in from the calling routine.</p>	<p>To filter pages ranges beginning with a specific page to the last page in the document required that we know how many pages exist in the document.</p>
4	<p>The <code>RotateSelectionPane</code> class contained an object which held the total quantity of pages. Using the debugger again, we were able to find a method to access and extract this information so we could send it to the parameter builder.</p> <p>The lambda call in <code>apply()</code> was altered to pass the total pages.</p>	<p>The total quantity of pages was known in <code>RotateSelectionPane</code>. This info needed to be passed to the parameter builder.</p>
5	<p>Passed in a new parameter to existing <code>jUnit</code> tests so they could compile and pass. Tests were located in the classes <code>RotateParametersBuilderTest</code> and <code>RotateSelectionPane</code>.</p>	<p>Adding a new parameter to <code>addInput()</code> caused existing <code>jUnit</code> tests to fail.</p>

Time spent (in minutes): 60

7 Postfactoring (optional)

Step #	Description	Rationale
--------	-------------	-----------

1	<i>No postfactoring was done.</i>	<i>Changes were fairly simple and required no big modifications of existing code.</i>
---	-----------------------------------	---

Time spent (in minutes): 0

8 Validation

Step #	Description	Rationale
1	<p><i>Ran test cases to check page ranges with</i></p> <ul style="list-style-type: none"> <i>• single pages,</i> <i>• a specific range (1-7)</i> <i>• and opened ended ranges (20-)</i> <p><i>on documents of varying lengths up to 49 pages</i></p> <p><i>and various settings of</i></p> <ul style="list-style-type: none"> <i>• EVEN_PAGES</i> <i>• ODD_PAGES</i> <i>• ALL_PAGES.</i> <p><i>One example test - For a 49 page document with custom page range (1-7, 20-) and ODD_PAGES, the tool rotated pages 1,3,5,7,21,23,25,27,29,31,33,35,37,39,41,43,45,47 and 49. EVEN_PAGES was selected and pages 2,4,6,20,22,24,26,28,30,32,34,36,38,40,42,44,46 and 48 were rotated.</i></p>	<p><i>We could quickly test all three cases manually and use the debugger to monitor the info sent to Sedja as well as check the resulting document.</i></p>
2	<p><i>Merged files together to create a 245 page document and rotate all ODD_PAGES from "6-".</i></p>	<p><i>Attempt to test longer files to make sure we don't have any size limitations.</i></p>

	<i>Expected behavior was observed and final document had all odd pages from 7 to 245 rotated 90 degrees.</i>	
3	<i>Merged files together to create a 735 page document and rotate all EVEN_PAGES from "7-". Expected behavior was observed and final document had all even pages from 8 to 734 rotated 90 degrees.</i>	<i>Next tried an even longer file of 735 pages.</i>
4	<i>Altered junit test, buildRanges(), to account for and ODD_PAGES setting on a custom page range.</i>	<i>A setting of ODD_PAGES instead of ALL_PAGES causes a reduced number of pages.</i>

Time spent (in minutes): 60

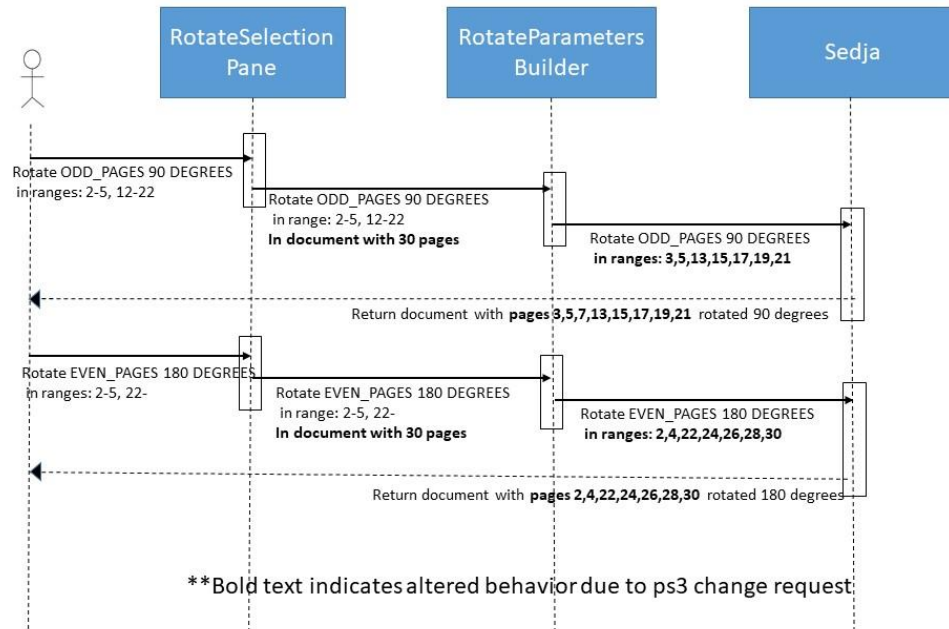
9 Timing

Summarize the time spent on each phase.

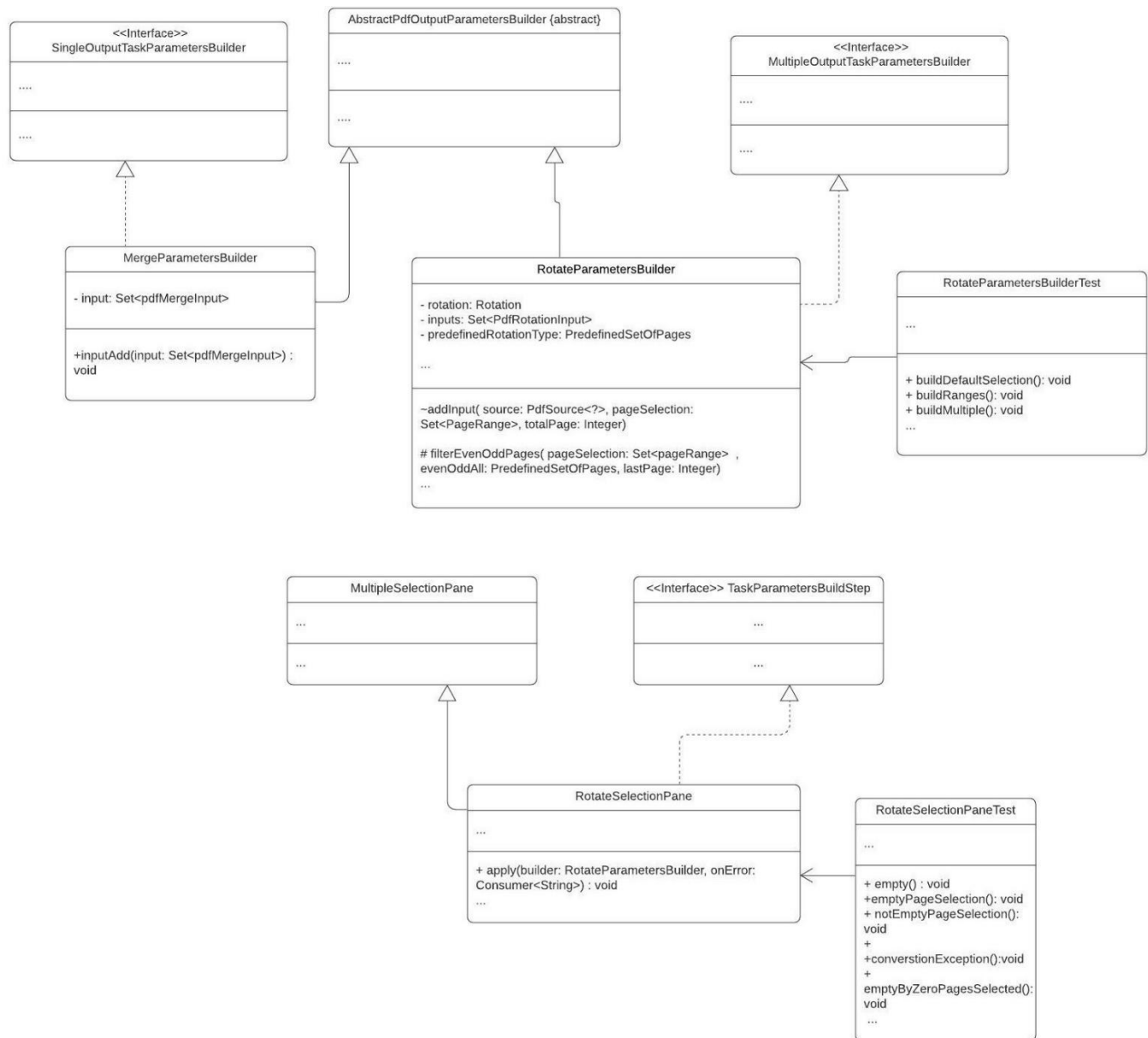
Phase Name	Time (in minutes)
Concept location	30
Impact Analysis	60
Prefactoring	0
Actualization	60
Postfactoring	0
Verification	60
Total	210

10 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.



11 Conclusions

The code for PDFSam is very well organized and unique to each task. The formatting of each task and communication with the Sedja open source library routines is individual to each task and there is very little overlap. Because of this, concept location was quickly and easily performed.

It was a little more difficult to interact with the Sedja objects and methods, but IntelliJ's debugger proved invaluable determining their content and structure. Once we extracted that information, things went quite smoothly.

Classes and methods changed:

- **pdfsam-rotate/src/main/java/org/pdfsam/rotate/RotateParametersBuilder.java**
 - `addInput(PdfSource<?> source, Set<PageRange> pageSelection, Integer totalPages)`
/* added new parameter to provide number of pages in document */
 - `filterEvenOddPages(Set<PageRange> pageSelection, PredefinedSetOfPages evenOddAll, Integer lastPage)` /* new method to remove EVEN or ODD pages from page ranges */
- **pdfsam-rotate/src/main/java/org/pdfsam/rotate/RotateSelectionPane.java**
 - `apply(RotateParametersBuilder builder, Consumer<String> onError)`
/* altered method to supply page size of document to addInput() */
- **pdfsam-rotate/src/main/java/org/pdfsam/rotate/ RotateParametersBuilderTest.java**
 - `buildDefaultSelection()` /* updated to reflect new parameter in addInput() */
 - `buildMultiple()` /* updated to reflect new parameter in addInput() */
 - `buildRanges()` /* updated test assertion to reflect new behavior of rotation task */
- **pdfsam-rotate/src/main/java/org/pdfsam/rotate/ RotateSelectionPaneTest.java**
 - `conversionException()` /* updated to reflect new parameter in addInput() */
 - `emptyByZeroPagesSelected()` /* updated to reflect new parameter in addInput() */
 - `empty()` /* updated to reflect new parameter in addInput() */
 - `emptyPageSelection()` /* updated to reflect new parameter in addInput() */
 - `notEmptyPageSelection()` /* updated to reflect new parameter in addInput() */