Change request log

# Team

Surface

Hwankook Lee, 831960823

<https://github.com/Martin987999/cs515-s19-surface-PDFsam>

# Change Request

PDFSAM Change Request #2

PDFSAM Change Request #3

# 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.**

PDFSAM Change Request #2

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | I ran the system. |  |
| 2 | For the test in AlternateMix, I added a single document, then ran it. | To figure out an error message indicated. |
| 3 | The error message produced “At least two elements required.”. I found the message using IntelliJ’s all search and the message came from .class file (AlternateMixMultipleParameters) not .java file due to @AtLeastTwo. I cannot modify the library. |  |
| 4 | Using IDE’s file search, I searched a class which is changed the original function slightly because I cannot change AlternateMixMultipleParameters. At “org.pdfsam.task”, There is BulkRotateParameters. | To find out a new class that is modified the original class file. |
| 5 | I inserted a new AlternateMixSingleInputParameters in that point and then debugged it. However, it was not working well. I realized that there are some more files for the task. | There are some files to support the new parameters file. |
| 6 | I found that there are two files more to change them. First one is Task file, and the second one is to add service into validation. | I confirmed that class file and xml file had to be changed. |

**Time spent (in minutes):** 1116

PDFSAM Change Request #3

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | I ran the system. |  |
| 2 | For the test in Merge, I added a document and input page ranges like 1-3, 2, then ran it. | To see an error message. |
| 3 | Using IntelliJ’s search tool, I searched for the class MergeParameters. | Due to the error message came from that point. |
| 4 | I went to the class MergeSelectionPane from 36 matches in 11 files because the error occurred by selection pane. |  |
| 5 | I analyzed the class MergeSelectionPane using ‘Find usages’ with IntelliJ’s function. | I found that the string that I input is manipulated by ConversionUtils class. |
| 6 | I debugged ConversionUtils class through a breakpoint and added it to favorites. | I convinced that the class had to be modified |

**Time spent (in minutes):** 726

# 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.

**Make sure you time yourselves when going through this process and provide the total time spent below.**

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 detail how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

PDFSAM Change Request #2

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | At the class AlternateMixParametersBuilder, I marked as to change. |  |
| 2 | This case is particular because I had to change AlternateMixMultipleInputParameters but I cannot. As a result, I made a new class for a single document. I created AlternateMixSingleInputParameters. Moreover, I added the class AlternateMixTask. Additionally, I changed AlternateMixParametersBuilder and sejda.xml file for those new classes. | The case does not affect other classes |

**Time spent (in minutes):** 48

PDFSAM Change Request #3

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | First, I examined the class ConversionUtils and marked as to change. | I changed the class because there is a bug. When the intersect page ranges were input, it cannot handle it. Therefore, I changed it and marked as changed. |
| 2 | I used IntelliJ’s function that ‘FindUsages’ showing where the class is using and then made a list. | To track each class that could be impacted by the change. |
| 3 | At a list, there are two classes, the first one is ExtractOptionsPane, and the second one is SelectionTableRowDaya. I marked the first as inspected and the second as propagating. | Because these classes do not impact on others so that I discarded to change them. |

**Time spent (in minutes):** 34

# 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.

**Make sure you time yourselves when going through this process and provide the total time spent below.**

PDFSAM Change Request #2

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | I created the class AlternateMixSingleInputParameters and AlternateMixTask. | I cannot directly change AlternateMixMultipleInputParameters so that I created AlternateMixSingleInputParameters to alter it slightly. Moreover, I needed task class to added on the system. |
| 2 | I manually examined functional testing at the first time, after that I created JUnit test for the new classes. | To make sure properly works. |

**Time spent (in minutes):** 135

PDFSAM Change Request #3

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | I modified the class ConversionUtils. | It does not handle to intersect page range set. I added several lines of code for searching intersecting page range. The logic is similar to find max or min value. |
| 2 | I manually tested writing intersect page range, then I made a Junit test. | To make sure properly works. |

**Time spent (in minutes):** 41

# 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.

**Make sure you time yourselves when going through this process and provide the total time spent below.**

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Test case defined: buildWithSingleDoc  Inputs: a single document mock  params.getInputList().size()  Expected output: 1 | This is the regular expected behavior.  The test passed. |
| 2 | Test case defined: intersect  Inputs: "1-3,2,1"  pageSet.size()  Expected output: 1 | This is the regular expected behavior.  The test passed. |
| 3 | Test case defined: intersect  Inputs: "1-3,2,1"  pageSet.stream().findFirst().get().getStart()  Expected output: 1 | This is the regular expected behavior.  The test passed. |
| 4 | Test case defined: intersect  Inputs: "1-3,2,1"  pageSet.stream().findFirst().get().getEnd()  Expected output: 3 | This is the regular expected behavior.  The test passed. |

**Time spent (in minutes):** 24

# 

# Timing

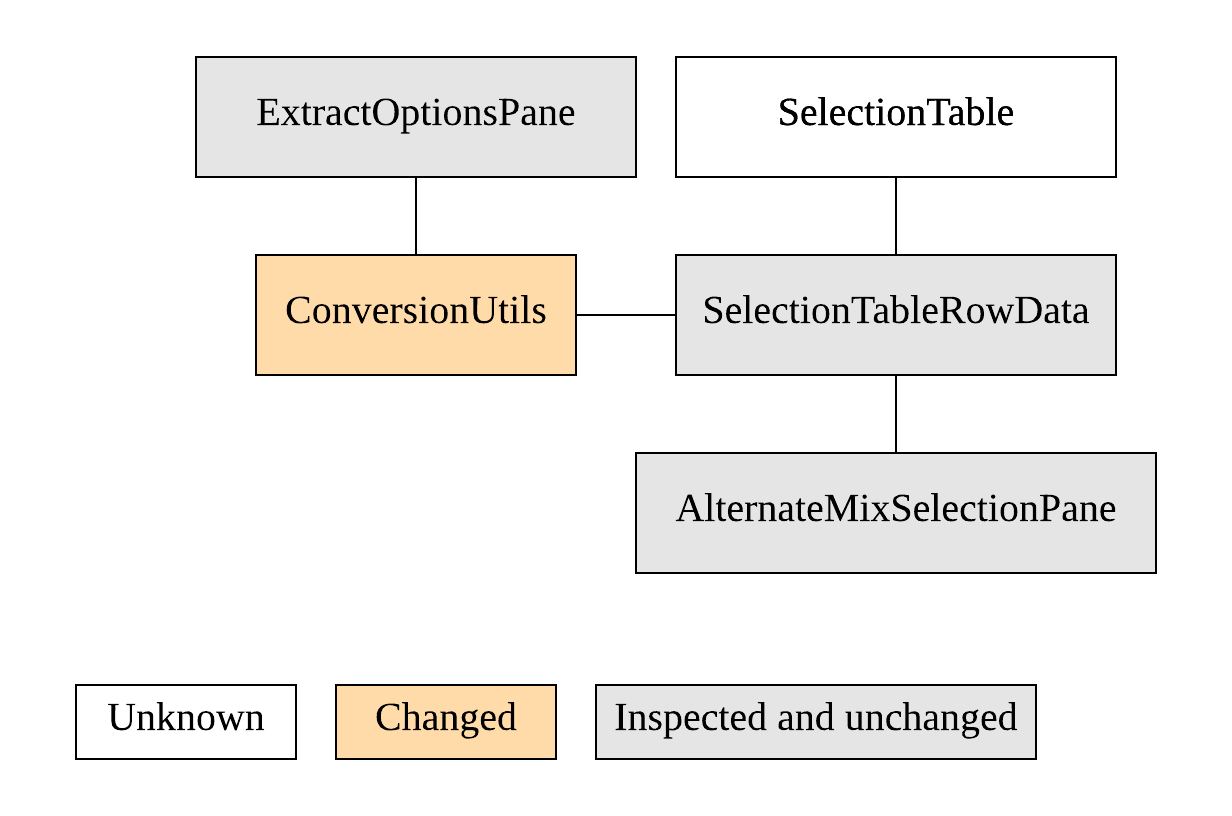
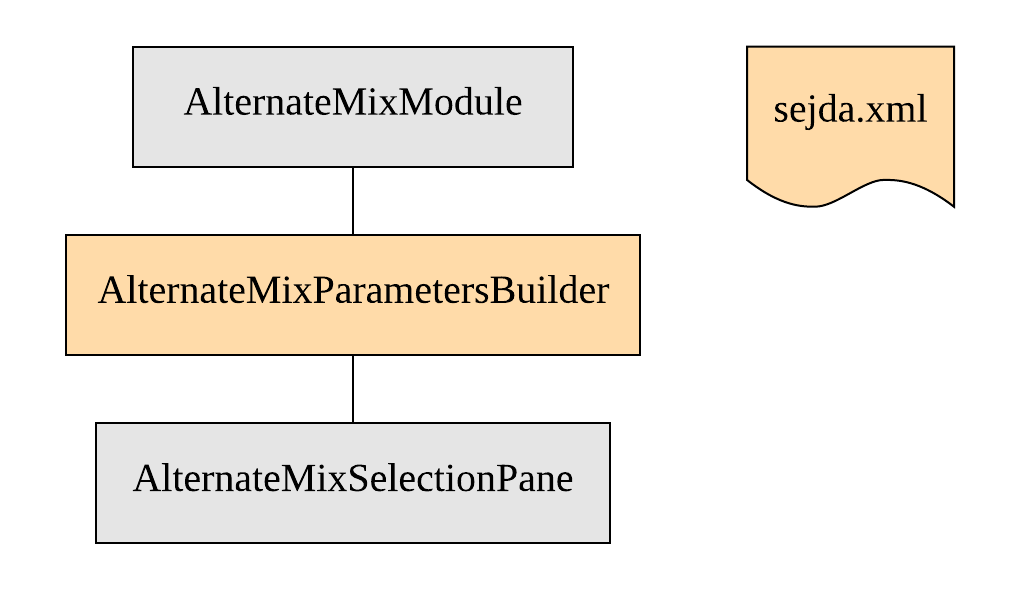
Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 1842 |
| Impact Analysis | 82 |
| Prefactoring |  |
| Actualization | 176 |
| Postfactoring |  |
| Verification | 24 |
| Total | 2124 |

# 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.

Through the whole progress, concept location was the hardest part because the system is too complex to find which I have to change. The system was written object-oriented design so that I hardly understand the system because most interface and abstract class indicated implicitly. Impact analysis is relatively easy to analyze. The reason is that I can use IDE’s functions and noticed which part affects other classes. Testing was performed manually for debugging to understand whole system. After I checked it works well, I created Junit test for easy checking.

Classes and methods changed:

* pdfsam-alternate-mix/src/main/java/org/pdfsam/alternatemix/AlternateMixParametersBuilder.java
* pdfsam-core/src/main/java/org/pdfsam/task/AlternateMixSingleInputParameters.java
* pdfsam-service/src/main/java/org/pdfsam/task/AlternateMixTask.java
* pdfsam-basic/src/main/resources/sejda.xml
* pdfsam-service/src/test/resources/sejda.xml
* pdfsam-core/src/main/java/org/pdfsam/support/params/ConversionUtils.java