Change request log

Team

Baradji_Wei

Coding: Baradji Diallo

Documentation: Wei Xi

Change Request

Change Request #2

- The Alternate Mix module allows to merge two or more documents taking pages alternately in natural or reverse order. Of course, this feature does not work when only one document is provided. You are requested to modify this module to allow the reversal of a single document.

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 "Alternate Mix".*	*Because we identified a button in the screen called "Alternate Mix"*
4	*From 139 results, we used eclipse IDE dependency navigator to inspect the class AlternateMixModule.*	*contained the searched keyword.*
5	*We inspected the class AlternateMixParametersBuilder.*	*Builder for the AlternateMixMultipleInputParameters.*

6	*We used eclipse IDE to open the declaration of the class AlternateMixMultipleInputParameters.*	*Was using a third-party library (sejda 3rd party library).*
7	*We inspected the class AlternateMixMultipleInputParameters.*	*We noticed a word stating "@AtLeastTwo". Which, we assumed that must have been the constraint forcing Alternate Mix to have 2 input parameters (sejda 3rd party library).*
8	*We inspected the class AtLeastTwo using eclipse IDE to open where it was declared.*	*To validate our assumption. After seeing the comment "Constraint validating an NotNull element with a minimum size of 2" our assumption was justified.*
9	*We marked the class AlternateMixParametersBuilder as "located".*	*We confirmed this class had to be modified.*

Time spent (in minutes): 300

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 open the call hierarchy of the class AlternateMixParametersBuilder.*	*To track the classes or methods that could be impacted by the change.*
2	*We inspected the class AlternateMixModule.*	*We realized this class had to be changed to display the correct message.*

Time spent (in minutes): 15

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 created the class AlternateMixMultipleInputParameters.*	*The 3rd party library had a constraint causing Alternate Mix to have a minimum number of 2 inputs. Therefore, a new class was needed without the constraint.*
2	*We modified the class AlternateMixParametersBuilder.*	*To have the right import file.*
3	*We created the class AlternateMixTask.*	*To perform the mix of one or more given PdfMixInput.*

4	*We modified the sejda.xml file.*	*To pass in the proper parameters for execution.*

Time spent (in minutes): 45

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 PDFsam* *Step2: click on "Alternate Mix"* *Step3: add a file by clicking on "Add" at the top left corner* *Step4: click on "Reverse" next to the file name* *Step5: click on run*	*To make sure the system is functioning as we expected. (was more convenient at the time)*
2	*Testing multiple file:* *Step1: run PDFsam* *Step2: click on "Alternate Mix"* *Step3: add a file by clicking on "Add" at the top left corner* *Step4: repeat step3 as necessary* *Step4: click on "Reverse" next to the file name you want to reverse.* *Step5: click on run*	*To make sure the original functionalities are still working properly.*

Time spent (in minutes): 15

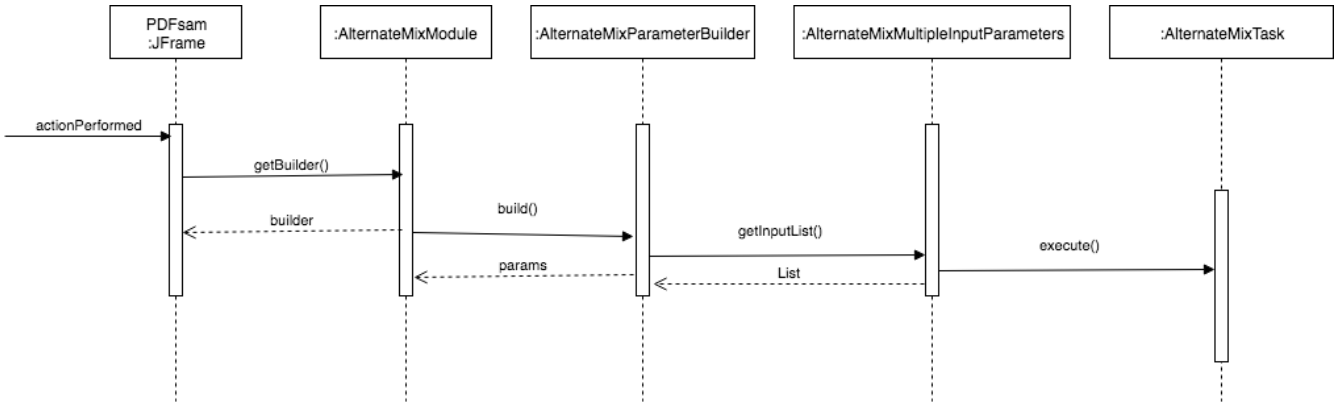
Timing

Summarize the time spent on each phase.

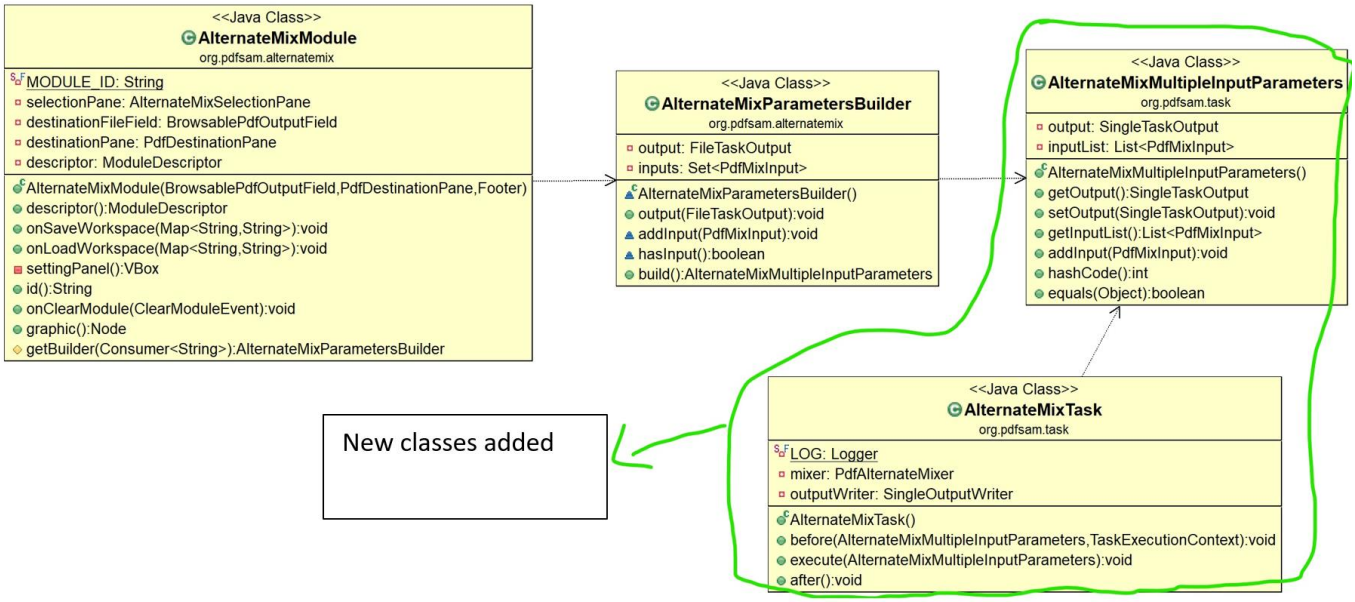
Phase Name	Time (in minutes)
Concept location	300
Impact Analysis	15
Prefactoring	n/a
Actualization	45
Postfactoring	n/a
Verification	15
Total	375

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, concept location was very hard because the system is large and searching for keywords was not helping at all. Additionally, the program was using a 3rd party library (sejda); which made it hard to track down where to start. Impact analysis and actualization was not too difficult because once we know how the 3rd party library was being used, it made it a little easier to correlate things and make changes. Testing was performed manually because it was more convenient at the time.

Classes and methods changed:

- *pdfsam-alternate-mix/src/main/java/org/pdfsam/alternatemix/AlternateMixModule*
- *pdfsam-alternate-mix/src/main/java/org/pdfsam/alternatemix/ AlternateMixParametersBuilder*
- *pdfsam-core/src/main/java/org/pdfsam/task/AlternateMixMultipleInputParameters*
- *pdfsam-service/src/main/java/org/pdfsam/task/AlternateMixTask*

- *pdfsam-community/src/main/resources/sejda*