Maintenance and Refactoring of Information Systems

Design of my planned contribution to the PDFBox Project

Communication with Mr. Litchfield (1)

Me: …I am writing to you asking for your advice and help…if I could commit some useful changes to your project… Have you any ideas?... what new elements do you want to add?...

Mr. Litchfield: …I would love some help…A couple that would be pretty interesting would be:

- Extract Images
- support color space
- PDF->HTML conversion
- HTML->PDF conversion
- adding support for PDF 1.5 object streams
- FDF export/import

Communication with Mr. Litchfield (2)

Mr. Litchfield:

…I have started some work on a PDFViewer…. I would like to have it be able to change values and write the PDF back to disk…

…I would also be interested in creating an Acrobat Reader like application in Java…**This is no way you could complete this** but if you started it maybe other people could work on it as well…

…There are some people asking for a PDF->XML as well, **that might be a little easier** than PDF-> HTML to implement. It would need to contain font/positioning information…

My Final Decision

Conversion of a PDF document to XML or HTML

Class PDFTextStripper (1)

Class PDFTextStripper

org.pdfbox.util

Hierarchy:

Java.lang.Object

|

+--org.pdfbox.util.PDFStreamEngine

|

+--org.pdfbox.util.PDFTextStripper

- public class PDFTextStripper extends PDFStreamEngine
- Version: $Revision: 1.33 $
- Author: Ben Litchfield (ben@csh.rit.edu)
- PDFTextStripper: 555 lines of code
- PDFStreamEngine: 864 lines of code

Class PDFTextStripper (2)

- The org.pdfbox.util.PDFTextStripper is the class that extracts the text out of the PDF.
- This class takes a PDF document, strips out all of the text and ignores the formatting.
- This class give us only the text and font information.
- There is no image extraction right now so we have to ignore images.

Official documentation:

This class runs through a PDF content stream, executes certain operations and provides a callback interface for clients that want to do things with the stream.

Method Summary (1)

- **Methods inherited from class org.pdfbox.util.PDFStreamEngine:**

protected void processOperator(PDFOperator operator, List arguments)
This is used to handle an operation.
void processStream(COSStream cosStream, Map fontMap)
This will process the contents of the stream.

- **Methods inherited from class java.lang.Object:**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- protected void flushText()

This will print the text to the output stream.

- int getEndPage()

This will get the last page that will be extracted.

Method Summary (2)

- String getLineSeparator()

This will get the line separator.

- String getPageSeparator()

This will get the page separator.

- int getStartPage()

This is the page that the text extraction will start on.

- String getText(COSDocument doc)

Deprecated.

- String getText(PDDocument doc)

This will return the text of a document.

- void writeText(COSDocument doc, Writer outputStream)

Deprecated.

- void writeText(PDDocument doc, Writer outputStream)

This will take a PDDocument and write the text of that document to the print writer.

Method Summary (3)

- protected void processPage(PDPage page, COSStream content) This will process the contents of a page.
- protected void processPages(List pages)

This will process all of the pages and the text that is in them.

- void setEndPage(int endPage)

This will set the last page to be extracted by this class.

- void setLineSeparator(String separator)

Set the desired line separator for output text.

- void setPageSeparator(String separator)

Set the desired page separator for output text.

- void setStartPage(int startPage)

This will set the first page to be extracted by this class.

- protected TextPosition showString(byte[] string)

This will show a string.

←←← (This method will be useful!)

Method showString

showString

protected TextPosition **showString**(byte[] string) throws IOException

This will show a string.

Overrides:

showString in class PDFStreamEngine

Parameters:

string - The string to show.

Returns:

A description of the text being shown.

Throws:

IOException - If there is an error showing the string.

Implementation Problems

- I was not able to set my classpath yet, so I have not run the application.
(This is the first problem to solve!)
- I know just the basics of XML.
- Do I have to create a new class or just to extend the existing PDFTextStripper?
- Is this work enough for our course?
- I don't know how much time I really need!