Ebook Manufacturing Pipeline

Bachelorarbeit

Studiengang E 033 532

Medieninformatik und Visual Computing

eingereicht von

Natascha Machner 1027745

bei Ao.Univ.Prof. Mag. Dr. Horst Eidenberger

Institute of Software Technology and Interactive Systems

der Technischen Universität Wien

Wien, August 2013

1. Projektbeschreibung

1.1 Aufgabenstellung

Es soll ein konkretes Verfahren mit der dazu benötigten Software bereitgestellt werden, um aus einem realen Buch in genau definierten Schritten ein Ebook herzustellen.

1.2 Umsetzung

Bei der Umsetzung wurden eigene kleine Programme selbst entwickelt sowie auf bestehende Software zurückgegriffen. Die verwendeten Programme sind dabei folgende:

- Calibre[1] (Entwickler: Kovid Goyal , Lizenz: GNU GPLv3)
- FreeOcr[2] (Entwickler: paperfile, Lizenz: freie Lizenz)

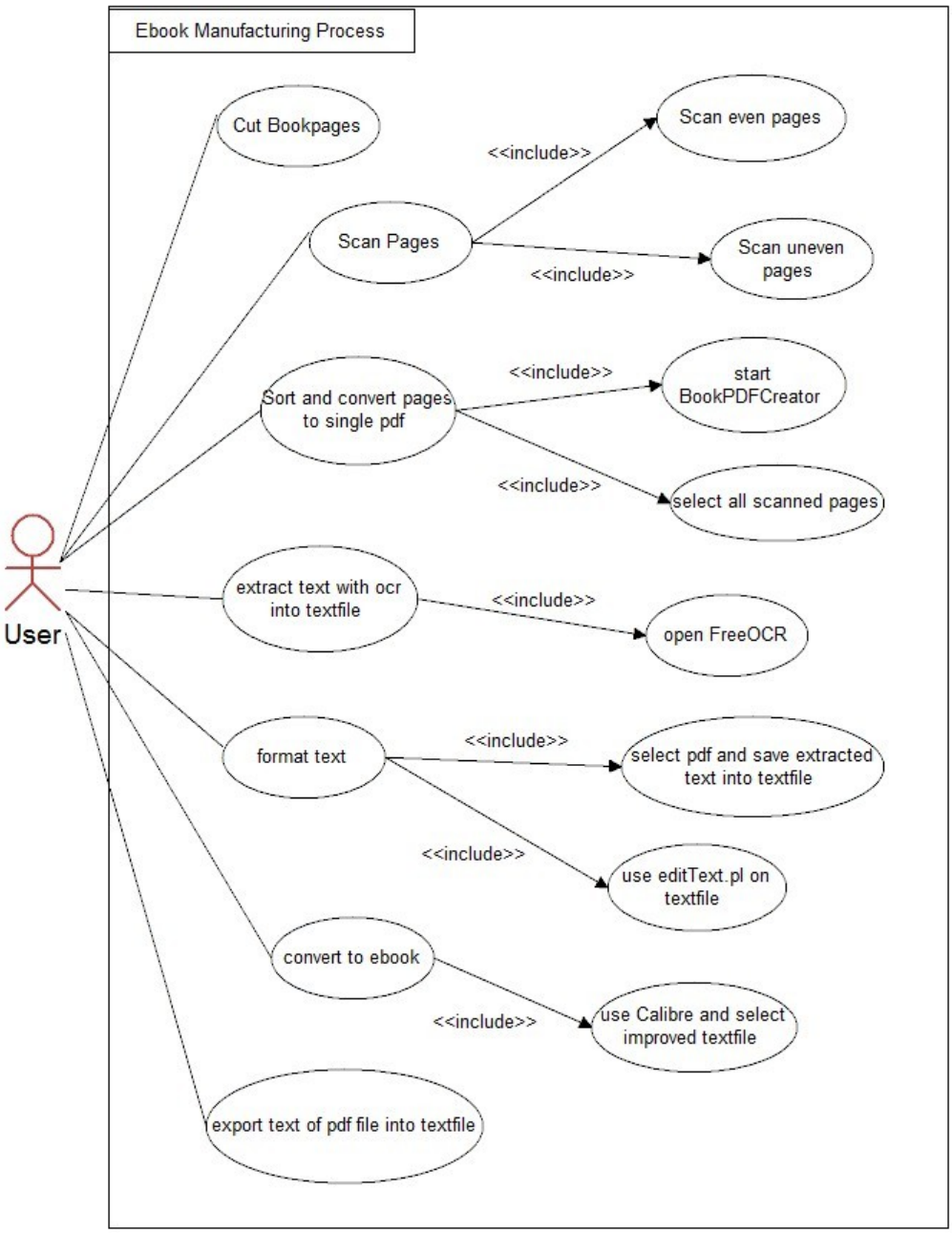
Die einzelnen Schritte des Prozesses sind dabei wie folgt definiert:

1. Die Seiten werden aus dem Buch gelöst.
2. Die Seiten, auf denen sich der relevante Text befindet, werden eingescannt, erst aufsteigend alle Seiten mit einer ungeraden Seitennummer, dann ebenfalls aufsteigend die Rückseiten mit geraden Seitennummern.
3. Das entwickelte Programm BookPDFCreator.jar wird aufgerufen und die zuvor eingescannten Bilddateien ausgewählt. Diese müssen entsprechend zur Scanreihenfolge benannt sein (z.B. Scan1.jpg). Das Programm bringt die Seiten in die richtige Reihenfolge und erzeugt aus allen Seiten ein einziges PDF-File.
4. Mit Hilfe von FreeOcr kann nun aus diesem PDF-File der Text ausgelesen und in eine Textdatei gespeichert werden.
5. Durch das entwickelte Perl-Script werden Seitennummern, Worttrennungen und ungewollte Absätze entfernt.
6. Mit Hilfe von Calibre kann die so verbesserte Textdatei in das gewünschte Ebook-Format konvertiert werden.

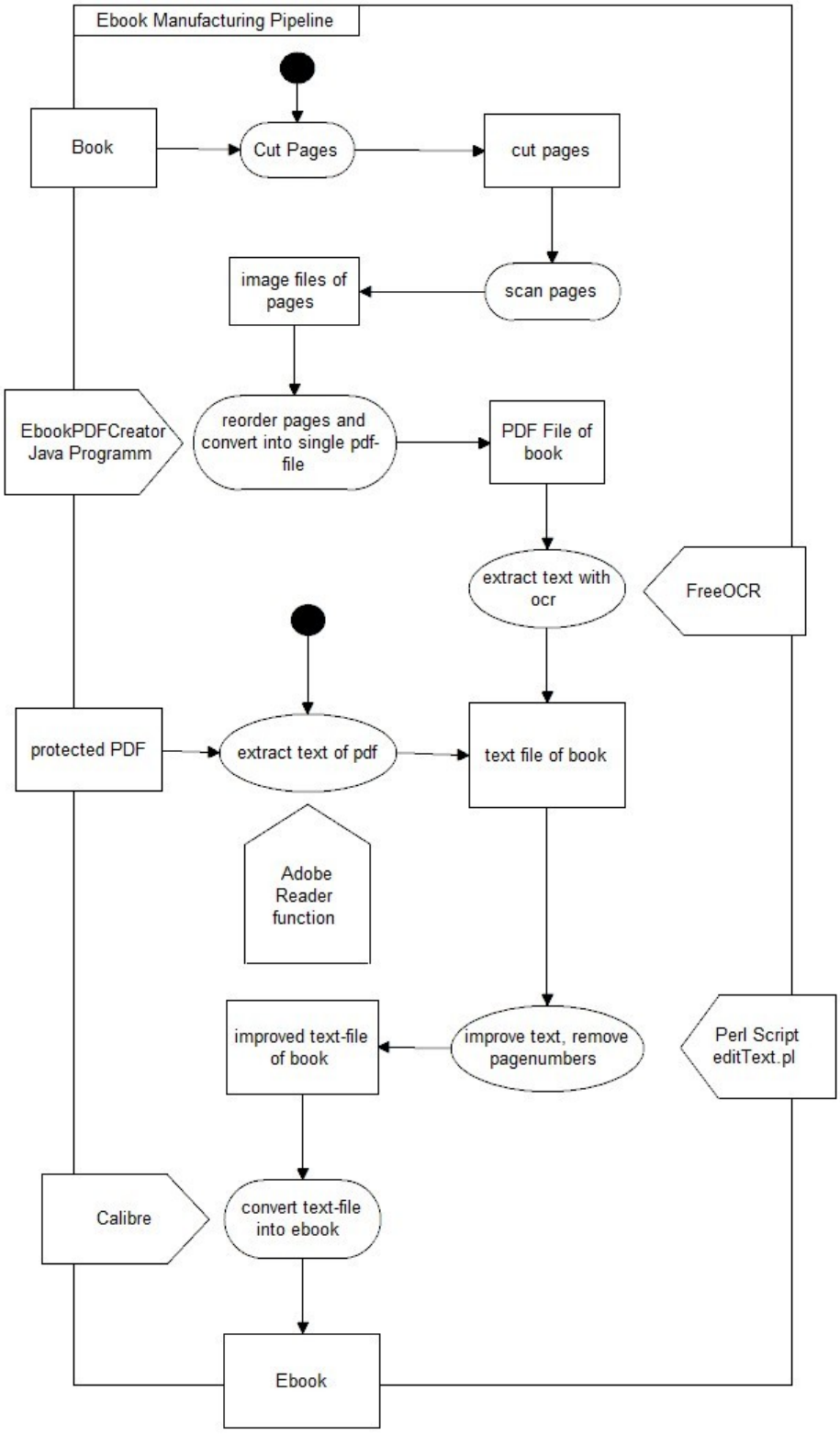
1 http://calibre-ebook.com
2 http://www.paperfile.net

1.3 Entwurfsdiagramme

Anwendungsfalldiagramm



Aktivitätsdiagramm mit den verwendeten Ressourcen



1.4 Zeitliste

Aktivität	**Datum**	**Zeit (h)**
Einlesen in die Aufgabenstellung und erste Recherche	13.06.2013	3
Recherche über AdobeScript	15.06.2013	2
Recherche über PDF-Creator Programme und APIs	15.06.2013	2
Installieren und Testen von ausgewählten Programmen (PDFCreator usw.)	17.06.2013	3
Besorgen, Zerschneiden und Einscannen von Büchern	17.06.2013	2
Informationen über PDF zu Plain Text Anwendungen	17.06.2013	3
Installieren und Testen von FreeOCR	17.06.2013	1
Einlesen in Perl	22.06.2013	4
Einarbeiten in Perl	23.06.2013	4
Erstellen des ersten Script-Prototypen zum Entfernen der Seitennummern in Perl	27.06.2013	5
Recherche über Ebook Konvertierer	28.06.2013	2
Installieren und Testen von Calibre	28.06.2013	1
Rechereche über APIs im Bereich OCR und Ebook Konvertierung	30.06.2013	4
Recherche über itext zum Vereinfachen des Sortierens und der PDF-Erstellung	01.07.2013	2
Einbinden und Einarbeiten in itext	01.07.2013	3
Erstellen des Java Programms EbookPDFCreator	02.07.2013	8
Build Prozess mit Maven	03.07.2013	5
Verbessern des Perl-Scripts	05.07.2013	5
Einscannen einen gesamten Buches zum Testen	08.07.2013	4
Bugfix im JavaProgramm EbookPDFCreator und testen	08.07.2013	6
Testen der Adobe Export Funktion in PlainText	09.07.2013	1
Erstellen des Abgabedokuments (Aufgabenstellung, usw)	10.07.2013	7
Javadoc im Programm	11.07.2013	2
Erstellen des Abgabedokuments (Diagramme)	11.07.2013	3
Verbessern des Perl-Scripts, Recherche über Codierungsprobleme	12.07.2013	4
Überarbeiten und Verbessern	03.08.2013	6
Zusammenstellung der Abgabedateien	04.08.2013	4

2. Mechanischer Part

Um die Seiten möglichst effizient aus dem gebundenen Buch zu lösen, wurden mehrere Herangehensweisen getestet.

Die einfachste Lösung, gerade auch wenn mehrere Bücher verarbeitet werden sollen, ist die Nutzung einer Schneidemaschine, wie sie in vielen CopyShops verwendet wird. Dabei können in einem ersten Schritt per Hand die Seiten in einem aus dem Bucheinband gelöst werden, was schnell und ohne großen Kraftaufwand möglich ist. Danach kann die Klebestelle an der Rückseite der Seiten mit der Schneidemaschine durch Einstellen eines geringen Abstandes entfernt und die Seiten somit voneinander getrennt werden.

Der Prozess zusammengefasst wäre also:

1. Heraustrennen des geklebten Seitenblocks aus dem Einband per Hand
2. Abtrennen der Klebestelle mit Hilfe der Schneidemaschine

Sollte ein solches Werkzeug nicht zugänglich sein, gestaltet sich das schnelle Trennen der Seiten schwieriger.
Die beste Variante wäre dann wohl, die Seiten mit einem Teppichmesser direkt aus dem Buch zu schneiden, ohne diese vorher daraus auszulösen, um dadurch eine stabilere Schneidefläche zu gewährleisten.
Dies ist zwar mit einigem Kraftaufwand verbunden und meist muss mehrmals mit dem Messer angesetzt werden, doch für den Gebrauch durch Privatpersonen ohne Zugang zu speziellen Maschinen ist dies wohl die effizienteste und schnellste Methode.

3. Implementierung

Das Programm BookPDFCreator wurde in Eclipse entwickelt. Als zusätzliche Library zur Konvertierung von Bilddateien zu PDF wurde itext[3] verwendet sowie das Buildtool Maven[4].
Das Programm ist sehr knapp gehalten und dient lediglich dazu, die Schritte des Umordnens der eingescannten Seiten und das Konvertieren in ein einziges PDF-Dokument zusammenzufassen und somit zu vereinfachen.
Der User muss nur noch die Seiten in einem Dialog auswählen, woraufhin das Programm die weitere Bearbeitung übernimmt und das fertige PDF-Dokument im selben Ordner wie die Bilddateien der Seiten ausgibt.
Dabei musste besonders darauf geachtet werden, dass Strings mit enthaltenen Zahlen nicht intuitiv nach eben diesen Zahlenwerten geordnet werden. Dies wurde aber beim Testen erkannt und konnte korrigiert werden, indem die Zahlenwerte in den Benennungen der Seiten (z.b. Seite10.jpg) extrahiert, in Integer konvertiert und erst dann sortiert wurden.

Die Komponente editText.pl wurde in der Scriptsprache Perl entwickelt. Mit Hilfe von regulären Ausdrücken wird hier nach Mustern gesucht, die nach dem Einscannen von Seiten und dem Auslesen des Textes durch FreeOCR entstehen.
Es werden allein stehende Zahlen (Seitennummern) entfernt, Absätze am Ende der Zeilen in Leerzeichen umgewandelt, falls diese nicht mit einem schließenden Satzzeichen enden, sowie getrennte Worte wieder zusammengefügt.
Bei Header und Footer ist das Erkennen nicht ganz so einfach. Diese können nicht von Überschriften unterschieden werden. Insofern müssen auch Überschriften entfernt werden, um eine Unterbrechung des Textes durch HeaderInhalte zu verhindern. Dafür ist die spezielle Variante „editText for Footers and Headers.pl“ vorgesehen, da die meisten deutschen Romane ohne Header oder Footer gedruckt werden und somit eine Behandlung dieser eher störend ist.

3 Itext, Entwickler: 1T3XT BVBA, Lizenz: GNU Affero General Public License, http://itextpdf.com/
4 Apache Maven , Entwickler: Apache Software Foundation, Lizenz: Apache-License 2.0, http://maven.apache.org/

4. Ausblick

Insgesamt wäre es wünschenswert, die Trennung der Schritte mehr und mehr zusammen zu fassen. Allerdings müssten so mehrere Programmiersprachen gleichzeitig verwendet werden, da die Textverbesserung und Formatierung über Perl funktioniert, gute OCR-Software wie Tesseract[5] aber in C++ vorliegt.
Der Vorteil an der Zergliederung in mehrere Schritte allerdings liegt darin, dass der User so auch in diese eingreifen kann und selbst optional eine manuelle Sprachverbesserung am Text vornehmen kann. Das würde bei einem vereinten Prozess größtenteils entfallen.

5 http://code.google.com/p/tesseract-ocr/