

Konzept für einen Web Books Viewer der neuen Generation

Masterarbeit

von

Abdelbasset Amara

Technische Universität Graz

Institut für Informationssysteme und Computer Medien (ICM)

A-8010 Graz

O.Univ.-Prof. Dr. Dr.h.c. Hermann Maurer

Graz, im März 2014



Concept for a Web Books Viewer of the new generation

Master's Thesis

submitted by

Abdelbasset Amara

Technische Universität Graz

Institut for Information Processing and Computer Supported New Media (IICM)

A-8010 Graz

O.Univ.-Prof. Dr. Dr.h.c. Hermann Maurer

Graz, March 2014

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....

Date

.....

(Signature)

Kurzfassung

Durch die Digitalisierung von Büchern und Handschriften wird der Zugriff auf eine enorme Menge an wertvollem Wissen und Informationen ermöglicht. Zahlreiche Projekte starteten den Digitalisierungsprozess mit dem Ziel, viele sowohl historische als auch neue Bücher zugänglich zu machen. Doch die Digitalisierung dieser Bücher und Handschriften erfordert neben technischen vor allem menschliche Bemühungen. Allen voran die Korrektur der resultierenden OCR (*optical character recognition*) Ausgaben und das Hinzufügen bzw. die Extrahierung von Struktur- und Inhaltsinformationen im Rahmen der Erweiterung der aus den Digitalisaten gewonnenen Metadaten. Diese bietet eine Fülle an neuen Möglichkeiten mit Daten umzugehen, sie für die Volltextsuche zu benutzen, mit anderen zu vernetzen und zu konsolidieren.

Um die bereits digitalisierten Materialien für die Öffentlichkeit online bereitzustellen, sind spezielle Viewer notwendig. Diese sollen sowohl die Präsentation verschiedenster Arten digitaler Inhalte gewährleisten als auch die „Lese-Emotionalität“ unterstützen. Dabei werden den Lesern Funktionalitäten wie Durchblättern, Suchen, Annotieren, Verlinken, Teilen von Inhalten in Sozialen Netzwerke etc. angeboten. Weiters soll eine intelligente Suchfunktion basierend auf den extrahierten Metadaten bereitgestellt werden. Dieser Aspekt wird von Bibliothekaren oft übersehen. Sie sind es gewohnt, ein Buch nach Kriterien wie Titel, Autor, Verlag etc. zu suchen. Diese Tatsache spiegelt sich in den oft verwendeten Metadatenformularen wider. Benutzer wollen aber oft alle Informationen zu einem Thema XY anzeigen lassen, ohne jedoch zu wissen, in welchen Büchern diese enthalten sind. Mit Hilfe von *Information Integration*-Ansätzen sollen nur die relevantesten Ergebnisse zu den Suchanfragen zurückgeliefert werden.

In dieser Masterarbeit wird zunächst das Konzept eines Web Books Viewers anhand verschiedener Anforderungen erstellt. Im Anschluss wird ein Prototyp vorgestellt, der speziell für das Austria-Forum entwickelt wurde und die Grundfunktionalitäten anbietet.

Abstract

The digitization of historical books and manuscripts offers an access to a huge amount of valuable knowledge and information. A lot of projects started the process of digitization with the aim to make many, both historic and new books available. However, the digitization of books and manuscripts requires not only technical but also human efforts. Especially the correction of the resulting faults in OCR (optical character recognition) and the adding or extracting of structural and content-related information within the scope of enhancing the extracted metadata from the digitized materials. This provides a wealth of new ways to deal with data, to use it for the full text search, to interconnect it and to consolidate it with other knowledge.

To provide online access to the already digitized materials, special viewers are needed. This should guarantee the presentation of the various types of digital contents as well as ensure the "reading emotions". Thereby a lot of functionalities should be offered to readers such as browsing, searching, annotating, linking, sharing content in social networks, etc. Furthermore, based on the previously extracted metadata, an intelligent search function should be provided. This aspect is often overlooked by librarians. They usually search for a book by criteria such as title, author, publisher, etc. This fact is reflected in the often used metadata forms. However, users often want to see all information about a topic xy without knowing which books contain it. With the help of information integration approaches, only the most relevant results to the search queries will be returned.

In this Master thesis, we propose at first a concept of a Web Books Viewer based on various requirements. After that a demonstration of a prototype is given, which was specially developed for the Austria-Forum and provides the basic functionalities.

Danksagung

An dieser Stelle möchte ich all jenen danken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Masterarbeit beigetragen haben.

Mein Dank gilt Herrn O.Univ.-Prof. Dr. Dr.h.c. Hermann Maurer für das Bereitstellen dieses interessanten Themas der Diplomarbeit und die freundliche Hilfsbereitschaft, die er mir entgegengebracht hat.

Des Weiteren danke ich Herrn Wurzinger, der mich sehr unterstützte und ohne dessen Hilfe und Bemühungen diese Arbeit nicht zustande gekommen wäre. Danken möchte ich außerdem den Mitgliedern des Austria-Forum Teams für ihre Unterstützung und die gute Zusammenarbeit.

Herzlich bedanken möchte ich mich bei meiner Mutter, die mir mein Studium ermöglicht und bei all meinen Entscheidungen unterstützt hat.

Mein besonderer Dank gilt meiner Freundin, die mich immer wieder ermutigte und mit vielen nützlichen Tipps zu meiner Diplomarbeit beigetragen hat.

Gender-Erklärung

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums angewendet. Es wird an dieser Stelle darauf hingewiesen, dass die ausschließliche Verwendung der männlichen Form geschlechtsunabhängig verstanden werden soll.

Inhaltsverzeichnis

Kurzfassung	II
Abstract	III
Danksagung	IV
Gender-Erklärung	V
Inhaltsverzeichnis	VI
Abbildungsverzeichnis	X
Tabellenverzeichnis	XI
1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellung	2
1.3 Ziele.....	3
1.4 Struktur der Arbeit	4
2 Austria-Forum und Digitalisierung	6
2.1 Austria-Forum – Überblick	6
2.2 Ablauf der Digitalisierung	7
2.2.1 Materialaufbereitung	8
2.2.2 Scannen.....	10
2.2.3 Bearbeitung und Speicherung der Materialien	12
2.2.4 Verbesserung der <i>OCR</i> -Ergebnisse.....	13
2.2.5 Veröffentlichung der digitalisierten Materialien	16
2.3 Web Books im Austria-Forum.....	16
3 Analyse des Standes der Technik	21
3.1 Übersicht verfügbarer Viewer	21
3.2 OpenSource	21
3.2.1 Google Books.....	22
3.2.2 Open Library Viewer.....	25

3.2.3	Uni Würzburg Viewer	28
3.2.4	DFG Viewer.....	31
3.3	Kommerzielle Produkte.....	32
3.3.1	Crocodoc.....	33
3.3.2	Scribd.....	34
3.4	Abschließende Auswertung	35
4	Warum einen eigenen E-Book Viewer?	38
4.1	Anforderungsklassen.....	38
4.2	Benutzeranforderungen	39
4.2.1	Funktionale Anforderungen.....	39
4.2.2	Nicht-funktionale Anforderungen.....	41
4.3	Interaktives und Kollaboratives Lesen der Bücher	42
4.3.1	Enhancement Layer.....	43
4.3.2	Communication Layer	45
4.4	Kollaborative OCR-Korrektur / Crowdsourcing	45
4.5	Metadaten	48
4.5.1	Metadaten anhängen.....	48
4.5.2	Buchinhalt indizieren	49
4.5.3	Akquisition der Metadaten	50
4.6	Information Integration	52
4.6.1	Konsolidierung der Informationen.....	53
4.6.2	Transclusion.....	53
5	Konzeption	56
5.1	Bestimmung der Implementierungsumgebung.....	56
5.2	Entwurf der Benutzeroberfläche	58
5.3	Architektur.....	60
5.3.1	Serverschnittstellen (APIs)	60
5.3.2	Clientseite	69

5.4	Ajax Webanwendung	70
6	Implementierung	72
6.1	Überblick	72
6.2	Steuerungsaufteilung	73
6.2.1	Model-Operationen	74
6.2.2	View-Operationen.....	76
6.2.3	Controller- Operationen	78
6.2.4	Kommunikation zwischen <i>Model-View</i> und <i>Controller</i>	80
6.3	Erläuterung der Oberflächenarchitektur und deren Funktionalitäten	81
6.3.1	Grundlegender Aufbau	81
6.3.2	Bibliothekübersicht	82
6.3.3	Ansichtsmodi	83
6.3.4	Layers	85
6.3.5	Andere Funktionalitäten	86
6.4	Image Extractor.....	87
6.4.1	Klassendiagramm.....	87
6.4.2	Funktionsweise	89
6.5	Schwierigkeiten bei der Implementierung	91
6.5.1	MooTools Framework.....	91
6.5.2	Benutzeroberfläche.....	92
6.5.3	Menge der Datenübertragung	92
6.5.4	Größe und Komplexität der Anwendung	92
6.5.5	Sicherheitspezifische Aspekte	93
7	Kritische Evaluierung und Ausblick.....	94
7.1	Abdeckung der Anforderungen	94
7.2	Ausblick.....	98
8	Literaturverzeichnis	99

Abbildungsverzeichnis

Abbildung 1 Einstiegsseite zu den Austria-Forum Web Books	17
Abbildung 2 Erste Ansicht beim Lesen des Buches.....	19
Abbildung 3 Seite mit einem angebrachten Link.....	19
Abbildung 4 Einzelseitenmodus.....	20
Abbildung 5 Vollständige Einzelseitenansicht auf Google Books	23
Abbildung 6 Eingeschränkte Vorschau	24
Abbildung 7 Auszugsansicht	25
Abbildung 8 Doppelseitenansicht	27
Abbildung 9 Einzelseitenansicht.....	27
Abbildung 10 Thumbnailansicht.....	28
Abbildung 11 Einzelseitenansicht.....	29
Abbildung 12 Doppelseitige Ansicht mit Umblätternanimation	29
Abbildung 13 Dreifachlupe.....	30
Abbildung 14 Scan-Textformat-Viewer	30
Abbildung 15 DFG-Viewer	32
Abbildung 16 Crocodoc Einzelseitenansicht	33
Abbildung 17 Crocodoc Doppelseitenansicht.....	34
Abbildung 18 Scribd Demo: Einzelseitenansicht.....	35
Abbildung 19 Layers des Web Books Viewers [Müller & Maurer, 11]	43
Abbildung 20 Metadaten-Suchfunktion im Austria Forum [Maurer, & Mueller, 13]	52
Abbildung 21 Transclusion.....	54
Abbildung 22 RESTful Web Server [Rasberry, 13].....	56
Abbildung 23 Web Books Viewer GUI-Entwurf, erstellt von Dr. Müller	59
Abbildung 24 Web Books Viewer MVC Muster Schichten.....	69
Abbildung 25 Web Books Viewer als Ajax Webanwendung.....	71

Abbildung 26 Klassendiagramm des Web Books Viewers	74
Abbildung 27 Hauptbildschirm der Anwendung	82
Abbildung 28 Abschnitt der Bibliothekansicht des Web Books Viewers	83
Abbildung 29 Einzelseitenmodus.....	84
Abbildung 30 Doppelseitenmodus	84
Abbildung 31 Thumbnailsseitenmodus	85
Abbildung 32 Korrektur-layer	86
Abbildung 33 Klassendiagramm.....	88
Abbildung 34 Image Extractor.....	89
Abbildung 35 Laufsequenz	90
Abbildung 36 Ergebnis einer Bildsuche	91

Tabellenverzeichnis

Tabelle 1 Crowdsourcing checklist	47
Tabelle 2 Vergleich zwischen Anforderungen und realisierten Funktionalitäten.....	94

1 Einleitung

1.1 Motivation

Seit den 1990er Jahren ist das Interesse an Digitalen Bibliotheken gestiegen. Bei den Digitalen Bibliotheken unterscheidet man drei große Kategorien: Bücher, Zeitschriften und Multimediaobjekte. Elektronische Zeitschriften sind inzwischen weit verbreitet. Davon sind manche kostenlos verfügbar wie z.B. J.UCS (Journal of Universal Computer Science)¹. Ein Beispiel für Multimediaobjekte sind Bildsammlungen, die u.a. im Austria-Forum² (AF) unter [Kronprinzenwerk, 14] und [IMAGNO, 14] angeboten werden. Da die verschiedenen Kategorien auf unterschiedlichen Techniken basieren, weist jede ihr eigenen Problematiken auf.

Von diesen drei Arten wird in Folge ausschließlich von Büchern die Rede sein.

In den letzten Jahren, dank der zunehmenden Verfügbarkeit von ökonomischen digitalen Speichermedien, *High-Speed-Scannern* und Netzwerken mit hoher Bandbreite, haben Digitale Bibliotheken sowohl wissenschaftlich als auch beruflich massiv an Bedeutung gewonnen. Mittlerweile ist der Traum, das große Wissen der Menschheit zu digitalisieren und online verfügbar zu machen, ein realisierbares Ziel.

Als Schritt in diese Richtung starteten mehrere Projekte, allen voran jenes an der *Carnegie Mellon University* ([Reddy, 95], [Reddy & Stclair, 01]), den Prozess der Digitalisierung der alten bzw. historischen Bücher. Der Digitalisierungsprozess eines Buches besteht aus verschiedenen Phasen: Materialbeschaffung, Scannen, Bildbearbeitung, Qualitätsprüfung und *Web-Hosting* bzw. Archivierung. Nähere Informationen zu der Erstellung Digitaler Bibliotheken sind in [Lesk, 04] gegeben. Aufgrund ständig laufender industrieller und akademischer Bemühungen auf diesem Gebiet, hat sich die anfängliche Frage, wie man Bücher digitalisiert, zur Frage, was man mit Digitalisaten machen kann, gewandelt [Sankar et al. 06].

Worin liegt nun der Vorteil, über diese Bücher in digitaler Form zu verfügen und wie können sie veröffentlicht bzw. online angeboten werden?

¹ <http://www.jucs.org/>

² <http://austria-forum.org/>

Um diese Fragen zu beantworten, muss man sich intensiv damit beschäftigen, wie man die Vorteile des Internets im Zusammenhang mit dem digitalisierten Schriftgut am besten nutzen kann [Crane, 06].

Das Digitalisieren von Bibliotheksbeständen und Archiven hat das Ziel, Daten bereitzustellen, zu veröffentlichen und mit bereits vorhandenen Ressourcen zu verknüpfen und somit in eine virtuelle Forschungsinfrastruktur zu integrieren.

Die daraus entstandene Infrastruktur macht das Internet zu einem integralen Forschungsgebiet für eine wachsende digital orientierte kulturwissenschaftliche Forschung. Um das Potenzial des Internets vollständig ausnutzen zu können, müssen diese digitalen Dokumente mit anderen verfügbaren Online-Ressourcen vernetzt werden [DFG, 13].

Österreich verfügt mit seinen Archiven, Bibliotheken, Museen und historischen Institutionen über einen unglaublichen Schatz. Dieses hochwertige Gut ist ein bestehender Teil des österreichischen Kulturerbes. Dieses Erbe wird von den kulturellen und wissenschaftlichen Einrichtungen konserviert, bearbeitet und veröffentlicht.

Ziel also wäre, dieses Kulturerbe so professionell wie möglich zu strukturieren und zu bearbeiten, um es allen zugänglich und nutzbereit zu machen. Eine so große Masse an Materialien in Form einer modernen und frei zugänglichen virtuellen Bibliothek, die neben Büchern alle Arten von Kulturbeständen und alten Ressourcen einbindet, bereitzustellen, ist nicht ganz hürdenfrei [Mikhail et al. 13].

In dieser Hinsicht ist das Austria-Forum als eine übergeordnete und verbindende Initiative zu betrachten. Übergeordnet in dem Sinn, dass es an der Beschaffung und dem Schutz der wertvollen alten Bücher, Handschriften und Kultursammlungen teilnimmt und verbindend deswegen, weil es diese Materialien für die Öffentlichkeit zur Verfügung stellt und mit anderen Wissenssammlungen effektiv verknüpft.

1.2 Problemstellung

Wenn man sich intensiv mit einem Thema beschäftigen möchte, z.B. im Rahmen einer Diplom- oder Forschungsarbeit, ist es zuallererst notwendig die relevante Literatur zu beschaffen. Eine wichtige Anlaufstelle dafür ist die Bibliothek, die heutzutage neben dem Internet bei der Wissensvermittlung eine zentrale Rolle spielt.

Schwierig wird es jedoch, wenn man auf der Suche nach historischen Büchern und Handschriften ist, die in Museen oder Archiven konserviert und geschützt, aber keinesfalls ausgeliehen oder kopiert werden können [Brantl & Schäfer, 10].

Natürlich ist der Sinn dahinter, dass solche Quellen in Museen und Kultureinrichtungen zu finden sind, nicht der, den Zugang der Öffentlichkeit zu verhindern, sondern diesen Schatz möglichst lange zu bewahren. Dies schränkt aber den Zugriff massiv ein und stellt damit die Nutzbarkeit solch wichtiger Ressourcen in Frage.

Wie kann man diesen Teufelskreis durchbrechen?

Die Bücher und Handschriftensammlungen sorgfältig zu scannen, zu bearbeiten und sicher zu speichern wird das Problem nicht lösen. Man muss diesen Kulturschatz für all jene, denen ein Interesse an seiner Verwendung liegt, zugänglich machen und zwar in einer Form, welche dem Original entspricht.

Die vorliegende Arbeit beschäftigt sich genau mit dieser Thematik, nämlich dem Veröffentlichen der bereits digitalisierten Bestände.

Wie kann man diese Bücher und Handschriften in digitaler Form einfach online zugänglich machen? Kann man das Lesen der Bücher bei der Veröffentlichung besser gestalten, so dass die „Lese-Emotionalität“ besonders gut unterstützt und präsentiert wird? Wie kann zusätzlich das Lesen von Büchern in virtuellen Gruppen ermöglicht und somit ein Ideenaustausch vorangetrieben werden?

Diese und noch weitere Fragen haben dazu geführt, den im Rahmen der vorliegenden Arbeit umgesetzten Prototyp Viewer als Grundkonzeption eines Web Books Viewers in Richtung Veröffentlichung der verfügbaren Digitalisate zu realisieren.

1.3 Ziele

Ziel dieser Arbeit ist die Realisierung eines Prototyp Viewers, mit dem digitalisierte Kulturgüter und andere verfügbare Bücher im Rahmen einer Onlineplattform mit Österreichbezug veröffentlicht werden können. Diese Plattform gilt mittlerweile als eine der größten Wissenssammlung des Landes und als Informationsportal und nennt sich Austria-Forum.

Dieser Viewer soll Bücher und Handschriften, die bereits gescannt und bearbeitet wurden, so darstellen, dass sie den Originalen entsprechen. Außerdem soll das Suchen, Zoomen, Erstellen von Kommentaren und Notizen, Vormerken von Lesezeichen sowie das Teilen oder Verlinken von Inhalten aus Büchern in Sozialen Netzwerken gewährleistet werden. Diese Punkte und andere werden in Abschnitt 4.2 erläutert.

Aufgrund der zunehmenden Nutzung mobiler Endgeräte wie Tablets und Smartphones, soll der Viewer plattformunabhängig sein, um für alle eine stabile und gleichförmige Verwendung zu garantieren.

Im Rahmen dieser Arbeit wird auch ein Grundkonzept für die Erstellung und Anlegung einer Bilder-Bibliothek erarbeitet. Diese soll das Extrahieren von Bildern aus digitalisierten Bücher und darüber hinaus das Anbieten eines Bilder-Inhaltsverzeichnisses ermöglichen, dank dessen man auch die Möglichkeit hat im Viewer gezielt nach Bildern zu suchen.

Näher untersucht werden soll auch das Konzept des Austria-Forums und der Weg zum Digitalisierungsgut, welcher vom Scannen über das Bearbeiten bis zum Speichern und Bereitstellen der Ressourcen führt.

1.4 Struktur der Arbeit

Die vorliegende Arbeit gliedert sich wie folgt:

Kapitel 2 liefert einen kurzen Überblick über das Austria-Forum und die Ansätze in Richtung Digitalisierung von Kulturressourcen in Österreich. Außerdem wird auf die Rubrik „Web Books“, worin Austria-Forum Bücher veröffentlicht, eingegangen.

Daraufhin folgt im Kapitel 3 eine Analyse der bereits existierenden Book Viewers. Es werden sowohl OpenSource Viewers als auch Shareware Softwares kurz beschrieben. Abschließend werden die bisher verfügbaren Tools im Zusammenhang mit den Erwartungen an einen Viewer ausgewertet.

Kapitel 4 geht der Frage nach, warum die Implementierung eines eigenen Viewers notwendig ist. Im Zuge dessen werden die Anforderungen dargestellt, die sich aus der zuvor präsentierten Analyse ableiten lassen und anschließend nach ihrer Priorität verschiedenen Klassen zugeordnet. Zusätzlich werden die Anforderungen in die Kategorien „funktional“ und

„nicht-funktional“ aufgeteilt und die jeweils dazugehörigen Benutzeranforderungen aufgelistet und definiert. Im Anschluss daran werden jene Anforderungen beschrieben, die in zukünftigen Arbeiten umgesetzt werden sollen.

Das Kapitel 5 beinhaltet eine Beschreibung der wesentlichen Konzeptionsschritte der Anwendung. Hierbei wird die Implementierungsumgebung vorgestellt und auf die Server APIs anhand von Aufrufbeispielen eingegangen. Danach wird die Architektur der Clientseite im Rahmen einer AJAX Anwendung präsentiert.

Aufbauend wird im Kapitel 6 die Implementierung der Schnittstellen erörtert. Dabei wird der Mechanismus der Steuerungsaufteilung erläutert sowie die Komponenten und Dienste des Prototyps vorgestellt. Im Weiteren wird die Software, welche für die Extrahierung der Bilder aus den Bücherseiten zuständig ist, dargestellt und dabei das Klassendiagramm sowie die Ablaufsequenz erklärt. Am Ende dieses Kapitels kommen jene Schwierigkeiten zur Sprache, welche bei der Umsetzung und während der Entwicklung des Prototyps aufgetaucht sind.

Abschließend wird im Kapitel 7 eine kritische Evaluierung und Auswertung der erarbeiteten Funktionalitäten im Hinblick auf die Gesamtheit der Anforderungen diskutiert. Mit einem Ausblick auf zukünftige Arbeiten wird diese Arbeit abgeschlossen.

2 Austria-Forum und Digitalisierung

In diesem Kapitel werden sowohl historische als auch technische Aspekte des Austria-Forums behandelt. Einerseits wird ein Überblick über die Entstehung des Austria-Forums gegeben und andererseits der Digitalisierungsprozess näher betrachtet. Abschließend wird die „Web Books“-Komponente im AF vorgestellt.

2.1 Austria-Forum – Überblick

Das Austria-Forum ist, wie in [Maurer, 14] beschrieben eine umfassende digitale Wissenssammlung mit dem Schwerpunkt Österreich. Es wurde ab 2006 am Institut für Informationssysteme und Computer Medien (IICM) der Technischen Universität Graz, basierend auf dem Vorläufer „AEIOU“³, in mehreren Schritten entwickelt [Trattner, 09]. AEIOU ist das berühmte Kürzel mit vielen Erklärungen, die aber alle nicht dokumentarisch belegbar sind.⁴ Im speziellen Fall stand es für Allgemeines Elektronisches Interaktives Universelles Österreichlexikon. Es wurde aufgrund einer Initiative von Hermann Maurer [AEIOU, 95] als das erste Online-Lexikon der Welt mit Hilfe des Wissenschaftsministeriums ab 1994 aufgebaut, die Text- und Bildteile erschienen aber auch in gedruckter Form, zuletzt dreibändig 2003. Aufgrund gekürzter Unterstützung seitens der Bundesministerien konnte die Aktualität nicht mehr aufrecht erhalten werden, weshalb die Bundesregierung die Einstellung des Betriebs befahl. Es war aber nur ganze 13 Tage eingestellt, wonach es wegen 4.000 Protestbriefen wieder in Betrieb genommen wurde. Es ist noch immer abrufbar: die Daten wurden alle in das Austria-Forum übernommen und werden nur dort gewartet.

Das Austria-Forum, eine auf JSP-Wiki-System basierte Onlineplattform, stellt dessen Nachfolger dar. Es beinhaltet Medien wie z.B. Texte, Bilder, Landkarten, Filme und Musik. Das Wissensnetz, welches das Austria-Forum anbietet, besteht aus den verschiedensten Themen mit dem Schwerpunkt Österreich, die von unterschiedlichen Standpunkten aus umfassend betrachtet werden. Es ist das Ergebnis der Integration und Vereinigung von aufbereiteten Metainformationen zu Interessensgebieten aus Österreichlexika, Büchern, Bildsammlungen und -bänden, Multimediaarchiven und noch vielen anderen Quellen.

³ www.aeiou.at

⁴ <http://austria-forum.org/af/Wissenssammlungen/Symbole/AEIOU>

Außerdem stellt das Austria-Forum eine Diskussionsplattform und redaktionell zitierbare Wissenssammlungen zur Verfügung, die von dem AF-Team laufend erarbeitet wird. Dies geschieht dadurch, dass die Mitglieder des *Editorial Boards*⁵ (EB) die Qualität der vorliegenden Beiträge überprüft, verifiziert (nähere Informationen zur Verifizierung der Beiträge⁶) und anschließend deren Bearbeitung sperrt.

Viele der im AF vorhandenen Beiträge werden im Gegensatz zu Lexika wie z.B. Wikipedia nicht aktualisiert, da eine Aktualisierung zu Informationsverlust führen kann. Allerdings haben die Beiträge ein Erstellungsdatum und können jederzeit durch Kommentare ergänzt werden. Auch kann zum selben Thema ein neuer Beitrag geschrieben werden, der aber den alten nicht zerstört, sondern auf diesen verweist.

Ein wichtiger Bestandteil des AF sind vollständige Bücher, die in Form von „Web Books“ zur Verfügung gestellt werden. Web Books sind nichts anderes als digitalisierte Bücher, welche analogen Büchern ähneln und im AF gelesen werden können. Das Besondere an einem Web Book ist, dass sein Layout der Ansicht eines aufgeschlagenen Buches entspricht und dass man Links von „normalen“ Seiten auf Buchseiten und umgekehrt legen kann, oder von einer Seite eines Buches auf eine andere Seite, gegebenenfalls auch eines anderen Buches.

Im Gegensatz zu eBooks sind diese Bücher nur online lesbar und können weder heruntergeladen noch ausgedruckt werden. Eine Ausnahme stellen „gemeinfreie“ Bücher dar.

Die vorliegende Arbeit beschäftigt sich damit, die Bedienbarkeit dieser Web Books zu optimieren, so dass eine Interaktive-Lese-Emotionalität überall (auf allen Browsern, Gadgets und Endgeräten) gewährleistet wird.

2.2 Ablauf der Digitalisierung

Hinter dem Begriff „Digitalisierung“ steckt ein gesamter Arbeitsvorgang, welcher aus unterschiedlichen Schritten besteht. Als erstes werden die Bücher bzw. Handschriften bereitgestellt und deren Scanbarkeit geprüft. Danach werden sie mit speziellen Scannern gescannt [Dörr & Weber, 97]. Die auf diesem Wege erhaltenen Bilder müssen manuell bearbeitet werden, um die Scanmängel zu beheben. Während dieser Qualitätscheck-Phase

⁵ http://austria-forum.org/af/Infos_zum_AF/Editorial_Board

⁶ http://austria-forum.org/af/Infos_zum_AF/Anleitung/Verifizierung

werden basierend auf diesen Bildern Metadaten erstellt, Volltexte gewonnen und bearbeitet. Anschließend werden die digitalisierten Informationen gespeichert und für die Veröffentlichung sowie die Langzeitarchivierung bereitgestellt.

Weil im AF auch Bücher in PDF-Form vorliegen, entfällt bei diesen der Scanprozess. Es handelt sich dabei um *Born digital* Bücher und solche, die bereits gescannt wurden und daher ebenfalls digital vorliegen.

2.2.1 Materialaufbereitung

Der Büchervorrat des AF umfasst vier Arten von Büchern. Bei einem Teil handelt es sich um gemeinfreie Bücher, die von jedermann frei verwendet werden können. Eine weitere Art von Büchern stellen jene dar, deren Nutzungsrechte auf das AF übertragen wurden. Schließlich umfasst das AF auch vergriffene Bücher und solche, die von den Verlagen nur mehr in einer geringen Anzahl verkauft werden und deshalb dank einer Kooperation mit dem Verlag im AF angeboten werden dürfen. Im Anschluss werden diese Arten näher beschrieben.

Im Rahmen des Digitalisierungsprozesses muss ein besonderes Augenmerk auf jene Rechte gelegt werden, die dadurch empfindlich berührt werden.

Damit ist das Urheberrecht angesprochen, welches das geistige Eigentum der Urheber im weiteren Sinn schützt [Wikipedia, 14a]. Es bestimmt an welchen Materialien Rechte bestehen. Bei der Aufbereitung dieser Materialien für den Digitalisierungsgebrauch muss festgestellt werden, wer die Inhaber der Vervielfältigungsrechte (Entscheidung über die Anfertigung von Kopien) sind und ob für die Digitalisierungszwecke die Zustimmung des Rechtsinhabers erforderlich ist [Hoeren, 05].

Das Urheberrechtsgesetz umfasst unter anderem gemäß § 2 Z 1 Werke der Literatur, zu denen Sprachwerke aller Art gezählt werden.

An gemeinfreien Büchern besteht kein Urheberrecht, weshalb sie von jedermann ohne eine Genehmigung verwendet werden dürfen, ohne dass eine Zahlungsverpflichtung entsteht [Wikipedia, 14b]. Zu den gemeinfreien Büchern zählen einerseits solche, deren Schutzfrist bereits abgelaufen ist. Gemäß § 60 Abs 1 UrhG erlischt das Urheberrecht 70 Jahre nach dem Tod des Urhebers. Andererseits gibt es auch Bücher, die nie urheberrechtlich geschützt waren.

Ist ein Buch vergriffen, fallen die Rechte an den Autor zurück. Ist dies der Fall, nimmt das AF-Team Kontakt mit dem Autor auf und fragt nach, ob ein Interesse daran besteht, das Buch neu auferstehen zu lassen. Der Autor eines vergriffenen Buches ist meist froh, wenn es wieder der Öffentlichkeit zugänglich gemacht wird und willigt ein, obwohl er dafür kein Entgelt erhält. Da gemäß § 23 Abs 1 UrhG das Urheberrecht vererblich ist, sind nach dem Tod des Autors dessen Erben die Ansprechpartner. Diese fordern in der Regel ein Entgelt. Da dies aufgrund der begrenzten finanziellen Ressourcen des AFs nicht möglich ist, stimmen die Erben meist nicht zu und das betroffene Buch kann nicht verwendet werden.

Wie gerade erwähnt, fallen die Rechte an einem vergriffenen Buch an den Autor zurück. Anders verhält es sich aber mit den Bildrechten. Bei den Bildern, die in Büchern enthalten sind, handelt es sich um Lichtbildwerke iSd § 3 UrhG und sie unterliegen als Werke der bildenden Künste ebenfalls dem Urheberrechtsgesetz. Stammen Bilder aus nur wenigen Quellen, ist es möglich, Kontakt zu den Photographen aufzunehmen und deren Einwilligung einzuholen. Sind in einem Buch Bilder aus vielen verschiedenen Quellen vorhanden, stellt diese Vorgehensweise einen zu großen Arbeitsaufwand dar. Deshalb kann es auch vorkommen, dass ein Buch aufgrund der Bildrechte nicht verwendet werden kann.

Wiederum gesondert zu betrachten sind Bücher, von denen der Verlag im Jahr nur noch zwischen 50 und 100 Exemplare verkauft. Stellt der Verlag in diesem Fall dem AF das Buch gratis zur Verfügung, so ist die Veröffentlichung im Austria Forum nicht als eine Konkurrenz zu betrachten, sondern stellt eher eine Werbung für das Buch dar. Eine derartige Kooperation besteht mit mehreren Verlagen wie „Böhlau“, „Brandstätter“, „Stocker“ etc.

Da Bücher eine starke Verfallsrate haben und sich ein neues Buch durchschnittlich nur zwei Jahre lang gut verkauft, können Bücher voraussichtlich bereits drei Jahre nach ihrer Erscheinung im Web angeboten werden.

Um die historischen Ressourcen zu ergänzen, hat das AF die Initiative „Blogmobil“⁷ ins Leben gerufen. Beim Blogmobil handelt es sich um einen mit “Hightech Equipment ausgestatteten Oldtimer LKW”, der die Möglichkeit anbietet, Objekte vor Ort und in Echtzeit zu digitalisieren. Bereits digitalisierte Güter zu ausgewählten Themen werden im Blogmobil veröffentlicht, den Besuchern gezeigt und mit ihren Geschichten und Erinnerungen bereichert. Dieser hochtechnisch gestaltete LKW reiste mit einem Team von Studenten der Technischen

⁷ <http://austria-forum.org/af/Wissenssammlungen/Blogmobil>

Universität, der Karl-Franzens-Universität und der Fachhochschule Joanneum quer durch die Steiermark, um den Kulturraum möglichst gut abzudecken und das alte Wissen originalgetreu zu sammeln bzw. zu erweitern. Unterstützt wird dieses Projekt vom Zukunftsfonds der steirischen Landesregierung [Blogmobile, 13].

Solche Dienste sind anhand folgender technischer Ausstattung gewährleistet:

- “Traveller’s Conservation Copy Stand 4232 (Buchscanner) mit Canon D7
- Nikon COOLSCAN V ED (Filmscanner)
- Reflecta DigitDia 5000 + ICE (Diascanner, hoher Durchsatz)
- Canon D7 mit Wechselobjektiven und Filmrig
- Allview Fotokopf mit terrestrischer Handsteuerung
- HDCopter Cinema Edition (optional)
- 3D-Kamera (Sony HDR-TD10)
- x AXIS P1344 Netzwerk Kamera (Interviewplatz)
- 3D Display (Samsung UE55ES8080)
- Outdoor Display (Samsung Large Format Display DE55C)
- Bose MA12EX, LT MB12, Powermatch 4500
- Lokaler (Web) Server
- SAN Disk
- Netzwerk (GSM 3G uplink, Kabel, WLAN access point außen)” [ebd.]

2.2.2 Scannen

Der Digitalisierungsprozess beginnt mit dem Scannen. Hierbei kommen spezielle Scanner zum Einsatz um die Buchseiten zu entsprechenden Bildern zu konvertieren.

Weil die Bücher originalgetreu veröffentlicht werden sollen, achtet das AF besonderes darauf, dass die Qualität des Digitalisats mit jener des Originals übereinstimmt. Aus diesem Grund wird ein „semi-automatisches“ Scanverfahren mit dem Ziel durchgeführt, scharfe, originäre und vollständige Bilder zu generieren um das Lesegefühl positiv zu beeinflussen. Um dies zu bewerkstelligen, kooperiert das AF mit der Vestigia Digitalisierungsgruppe der Karl-Franzens-Universität Graz unter der Leitung von Karl Lenger. Dieser Partner gewährleistet durch seine langjährige Erfahrung in diesem Bereich und seiner ausgezeichneten technischen Ausrüstung eine entsprechend hohe Scanqualität [Maurer & Müller, 11].

Das Ergebnis dieses ersten Schrittes sind zwei Formen von digitalisierten Materialien: der digitale Master (Roh-Digitalisat) und die daraus abgeleiteten Derivate. Ein digitaler Master beinhaltet Metadaten, die eine vollständige Beschreibung darstellen und daher die Automatisierung des Prozesses ermöglichen. Im Laufe des Digitalisierungsvorgangs bleibt der

digitale Master fix und kann somit nicht geändert werden. Bei Derivaten handelt es sich um Kopien in verschiedenen Formaten (z.B. JPEG, PNG), die eine deutlich kleinere Dateigröße haben als der digitale Master. Derivate sind also Kopien, welche auf der Präsentation und Qualität des jeweiligen Masters basieren. Im Gegensatz zu den Mastern können Derivate beliebig oft modifiziert werden. Jedoch sind nur Änderungen, die keinen Einfluss auf den Inhalt haben, erlaubt.

Grundsätzlich haben digitale Master im AF eine Auflösung von 300 dpi (Bildpunkte pro inch, engl. *dots per inch*) und liegen unter verlustfreien komprimierten Files als JPEG2000 oder TIFF (*Tagged Image File Format*) vor.

Da die weiteren Digitalisierungsprozesse auf dem digitalen Master basieren, muss ein besonderes Augenmerk auf dessen Herstellung gerichtet werden.

Besonders in Bezug auf die OCR-Erkennung spielt die Scanqualität eine bedeutende und entscheidende Rolle. Um die Qualität der OCR-Ausgaben positiv zu beeinflussen, müssen neben visuellen auch noch andere Kriterien sowohl während des Scanvorganges als auch danach berücksichtigt werden.

Der Scanvorgang soll vollständige Vorlagen liefern. Es dürfen also durch den technischen Prozess keine unerwünschten Informationsverluste entstehen. Jedoch treten besonders bei älteren Materialien gehäuft Scanprobleme auf. Grund dafür sind zum Beispiel unterschiedliche Lichtverhältnisse innerhalb der Seiten, die auf feuchtigkeitsbedingte Welligkeiten zurückzuführen sind. Diese Wölbungen der Textzeilen in den Scandateien machen die OCR-Erkennung besonders schwierig. Des Weiteren können Buchstaben der Rückseite, die auf der Vorderseite erscheinen, nicht als Hintergrund erkannt werden und führen deshalb unkorrekterweise zu einer Textausgabe.

Nur wenn man sich bereits vor dem Scanprozess über die häufig auftretenden Probleme bewusst ist, kann man entsprechende Maßnahmen setzen, um diesen Problembereichen gezielt entgegenzuwirken. Auf diese Weise kann die Qualität der Scans und in weiterer Folge auch jene der OCR-Erkennung verbessert werden.

Schließlich gibt es aber auch Scanprobleme, die trotz eines verbesserten Scanvorganges nicht behoben werden können. Beispiele dafür sind Schmutzflecken, Unterstreichungen oder Anmerkungen von vergangenen Benutzern. In manchen Fällen stellen sich jedoch Anmerkungen als nützlich und sogar erwünscht heraus und zwar dann, wenn sie zusätzliche

Erklärungen oder Annotationen von Personen bzw. Autoren enthalten, die sich mit der Materie besonders gut auskennen. So sind zum Beispiel Anmerkungen in Mathematik- oder allgemeinen Sachbüchern oft von entscheidender Bedeutung. Diese Erläuterungen sollten berücksichtigt und den Lesern in einem zusätzlichen Layer angeboten werden.

Das AF-Team bemüht sich, unerwünschte Mängel, die auf den Scans erscheinen, möglichst effektiv zu beheben. Dabei werden unter der Verwendung gängiger Bildbearbeitungssoftwares beispielsweise Randentfernungen, geometrische Korrekturen und Entfernungen unerwünschter Flecken durchgeführt.

2.2.3 Bearbeitung und Speicherung der Materialien

In der Regel ist die Qualität der aufgenommenen Bilder nicht optimal, weshalb Korrekturen und Bearbeitungen notwendig sind, die jedoch zu keiner Formänderung führen dürfen.

Ein wichtiger Schritt dieser Phase ist die Erzeugung der Metadaten, welche dazu dienen, den digitalen Master zu beschreiben. Sie sind grundsätzlich in Form einer XML-Datei bereitgestellt und werden in den restlichen Ablauf des Digitalisierungsprozesses einbezogen. Ein Roh-Digitalisat wird anhand von verschiedenen deskriptiven Metadaten beschrieben wie z.B. Fileformat, Auflösung, Kompression, Buchscanner etc. Hinzu kommen die in originalen Büchern standardmäßig enthaltenen Metadaten wie Titel, Autoren, Verlag, ISBN-Nummer, Jahr, Anzahl der Seiten usw. Es werden jedoch nicht nur deskriptive, sondern auch strukturelle Metadaten in diesem Schritt erzeugt. Diese beschreiben die Struktur des Digitalisats, wodurch viele zusätzliche Funktionalitäten angeboten werden können. So kann z.B. ein Inhaltsverzeichnis erstellt werden um es dem Leser zu ermöglichen, das Buch einfach und übersichtlich zu lesen und bequem durchzublättern.

Den letzten Schritt dieser Phase stellt die Volltextgenerierung dar. Diese erfolgt durch die Ausführung von OCR (engl. *Optical Character Recognition*). Das AF kooperiert diesbezüglich mit der Karl-Franzens-Universität Graz [Maurer & Müller, 11], an der die Volltextmaterialien mit einer speziell eingesetzten OCR-Software automatisch aus den Bildern extrahiert sowie manuell kontrolliert und ausgebessert werden. Dieses Vorgehen dient dazu, die Textansicht möglichst qualitativ hochwertig zu gestalten und so mit einer gewissen Zuverlässigkeit zu veröffentlichen.

Mit diesen durch OCR erstellten Texten können ausreichend Strukturmetadaten extrahiert und beispielsweise für die Suche nach Wörtern oder sogar vollständigen Sätzen in Büchern benutzt werden.

Schließlich werden diese Informationen durch ihre Speicherung als PDF-Dateien für die Langzeits archivierung bereit gemacht, da sich das PDF-Format für die Langzeitsicherung und -archivierung am besten eignet. Gleichzeitig gewährleistet es eine langfristige und sichere Verwendung des Digitalisats und stellt die Materialien optimal zur Verfügung.

2.2.4 Verbesserung der OCR-Ergebnisse

Obwohl die Bemühungen der Forscher auf dem Gebiet der maschinellen Texterkennung zu erheblichen Fortschritten geführt haben, ist man von einer hundertprozentigen Textübereinstimmung noch weit entfernt [Erkin, 10]. Vor allem im Zusammenhang mit historischen Büchern und Handschriften hat die OCR-Erkennung einen schlechten Ruf bei den Digitalisierungsverantwortlichen. Hier stehen einer sehr geringen Erkennungsgenauigkeit und somit beinahe Unbrauchbarkeit der Ergebnisse enorme Kosten gegenüber [Wegstein, 09].

Zum einen ist die Fehlererkennung bei historischen Materialien auf die Mängel der Ausgangsquellen zurückzuführen. Faktoren wie Schmutz bzw. Flecken, graue Hintergründe, schlechte Kontrastverhältnisse, unebene Seiten, Risse in den Dokumenten sowie schwerleserliche Kopien können die Erkennung des Textes geradezu unmöglich machen. Zum anderen stellt die Schriftart der Altbestände, wie zum Beispiel die Frakturschrift, ein großes Problem dar [Erkin, 10]. Die Frakturschrift gehört zu der Gruppe der gebrochenen Schriftarten und war von der Mitte des 16. bis Anfang des 20. Jahrhunderts die meistbenutzte Druckschrift im deutschsprachigen Raum [Wikipedia, 14c]. Erschwerend hinzu kommt noch, dass sich der in historischen Dokumenten enthaltene Wortschatz und die Rechtschreibung aufgrund der Sprachentwicklung erheblich von jener der modernen deutschen Sprache unterscheidet.

Das AF-Team hat den weiteren Handlungsbedarf in diesem Bereich erkannt und es sich zum Ziel gesetzt, die Qualität der OCR-Ausgaben zu verbessern.

Zurzeit läuft ein Projekt, das sich mit dem „Biographischen Lexikon des Kaiserthums Österreich (Wurzbach)“ befasst. Dieses Werk wurde in den Jahren 1856 bis 1891 von Constant von Wurzbach in 60 Bänden verfasst und veröffentlicht. Es enthält 24.254 Biographien bedeutungsvoller Personen, die von 1750 bis 1850 in den österreichischen Kronländern entweder geboren wurden oder gelebt haben. Dieses Lexikon zählt somit zu den wichtigsten Biographiequellen.⁸

Um die biographischen Informationen optimal nutzbar zu machen, muss es möglich sein, innerhalb des 60-bändigen Wurzbach nach Namen zu suchen. Ausgangspunkt dafür ist das in jedem Band enthaltene Namensregister. Dieses gibt Auskunft darüber, auf welcher Seite der Haupteintrag zu der jeweiligen Person beginnt. Jedoch enthält es keinen Hinweis darauf, auf welchen anderen Seiten bzw. in welchen anderen Biographien diese Person als Nebenperson in Erscheinung tritt. Für Forscher, die sich mit einer bestimmten Person befassen, ergibt sich somit ein grundlegendes Problem. Es ist nämlich geradezu unmöglich alle 60 Bände mit einem Umfang von durchschnittlich ca. 500 Seiten pro Band manuell daraufhin zu prüfen, ob die Person, an der ein Forschungsinteresse besteht, auch noch an anderen Stellen genannt wird.

Wie bereits erwähnt, soll eine Personensuche innerhalb der Wurzbach-Bände durchführbar sein. Dabei handelt es sich im Gegensatz zu einer Volltextsuche, deren Verwirklichung um ein Vielfaches komplexer wäre, um eine Suche unter Verwendungen eines eingeschränkten Wörterbuches. Es stellt sich somit die Frage, wie diese Funktionalität bei einer gleichzeitig hohen Zuverlässigkeit der Ergebnisse gewährleistet werden kann.

Die aus den bereits digitalisierten Bänden generierten OCR-Ausgaben weisen weitgehend schlechte und zum Teil sogar unbrauchbare Texterkennungsergebnisse auf, die einer erfolgreichen Personensuche im Wege stehen. Angesichts der enormen Anzahl an Buchseiten kommt nur eine OCR-Korrektur in Frage, die eine starke Lernkomponente aufweist und in einem völlig automatisierten Workflow durchgeführt werden kann.

Der Standardablauf einer OCR-Erkennung gliedert sich in die folgenden drei Phasen, die aufeinander abgestimmt sind und im Rahmen eines OCR-Verfahrens mehrmals wiederholt werden:

⁸ http://austria-forum.org/af/AustriaWiki/Constantin_von_Wurzbach

- **Strukturerkennung und Segmentierung**

Die Layouterkennung ist der erste und wichtigste Schritt in diesem Prozess. Hier wird der Seiteninhalt aus dem Druckbereich extrahiert. Der Inhalt kann neben dem Text auch Bilder, Tabellen und Graphiken umfassen sowie verschiedene Kombinationen daraus. Für den nächsten Verfahrensschritt werden ausschließlich Textblöcke entnommen und in Zeilen, Worte und Zeichen segmentiert. Damit können im Optimalfall Überschriften, Fußnoten, Anmerkungen und Verzeichnisse im Dokument erkannt werden.

- **Texterkennung mittels eines Mustererkennungsverfahrens**

Nach der Extrahierung der Wörter und Buchstaben folgt nun deren Analyse. Diese wird durch die Anwendung von Klassifikatoren durchgeführt. Es handelt sich bei diesen Klassifikatoren um vordefinierte Muster, die Buchstaben definieren bzw. repräsentieren um die OCR-Engine auf alle vorkommenden Zeichen zu trainieren. Dieser Abgleich liefert mögliche Wortkandidaten für das zu erkennende Wort.

- **Evaluation**

In diesem Schritt kommt ein Wörterbuch zum Einsatz. Das in [Gotscharek et al., 09] von CIS-Gruppe der Ludwig Maximilian Universität München veröffentlichte Testergebnis zeigt, dass man durch die Anwendung eines Wörterbuches während des Erkennungsprozesses eine Fehlerreduktion von 30% bis 60% erreichen kann.

Im Rahmen des Wurzbach-Projektes wird ein OCR-Verfahren verwendet, in dessen Evaluationsphase ein erweiterbares Wörterbuch zum Einsatz kommt. Dies wird dadurch möglich, dass eine fehlerfreie Version des 1. Bandes des Wurzbach-Lexikons zur Verfügung steht, die im Rahmen eines Wiki Commons Projektes mittels manueller OCR-Korrektur erstellt wurde. Neben diesem korrigierten Textdokument bilden zwei weitere, mit ABBYY FineReader⁹ bzw. Tesseract¹⁰ erstellte Textausgaben die Arbeitsgrundlage.

Die zwei OCR-Ausgaben werden mit dem korrigierten Textdokument automatisch verglichen um so deren Fehler zu erkennen. Dieser Vergleich wird mit Hilfe des Levenshtein Distanz Algorithmus durchgeführt, der die Ähnlichkeit zwischen den Wörtern der beiden Texte bestimmt. Die daraus resultierenden Ergebnisse werden in das Wörterbuch eingetragen und

⁹ Abbyy FineReader. <http://finereader.abbyy.de/professional/>

¹⁰ Tesseract-OCR <https://code.google.com/p/tesseract-ocr/>

somit die OCR-Engine auf die neu hinzugefügten Wörter trainiert. Danach wird ein heuristisches Verfahren angewendet, welches die Fehlerquote ermittelt und anhand dessen entschieden wird, wie oft dieser Prozess wiederholt werden soll.

Während des Korrekturprozesses hat sich gezeigt, dass die alte Rechtschreibung mit Tesseract besser erkannt wird als mit dem Abby FineReader, so dass nun nur mehr die von Tesseract erkannten Texte bearbeitet und bewertet werden.

Derzeit liegt die Erkennungsrate bei 85% [Pirker & Müller, 14]. Die verbleibende Fehlerrate ergibt sich jedoch nur bei einer Volltextsuche, weil das entwickelte Programm weder die Fontgröße noch das tatsächliche Vorliegen eines Personennamens berücksichtigt. Daher ergibt sich bei einer ausschließlichen Betrachtung der Personennamen eine höhere Trefferquote.

Um zu noch besseren Ergebnissen zu gelangen, wird die Arbeit fortgesetzt. Da die restlichen Bände einen ähnlichen Inhalt aufweisen, lässt sich aus der Erkennungsgenauigkeit des ersten Bandes eine gute Prognose für die restlichen Bände ableiten.

2.2.5 Veröffentlichung der digitalisierten Materialien

Um den Digitalisierungsprozess zu vollenden, müssen die digitalisierten Güter für die Öffentlichkeit bereitgestellt und zugänglich gemacht werden. Dafür bietet das AF einen auf Flash basierten Web Books Viewer an. Da Apple die Flash-Technologie von Adobe im iOS nicht unterstützt (siehe [Steve Jobs, 10]) und bis dato von Adobe keine gratis Flash Version für iOS veröffentlicht wurde, stellt sich die Herausforderung, eine auf HTML5 basierte Alternative anzubieten.

Mit der vorliegenden Diplomarbeit wird das Ziel verfolgt, einen Web Books Viewer zu erstellen, der genau diese funktionalen Lücken schließt.

2.3 Web Books im Austria-Forum

Neben der umfassenden Wissenssammlung sowie der zahlreichen Beiträge bietet das Austria-Forum eine Bücher-Rubrik an, die eine wichtige Komponente darstellt. Hier können die verfügbaren Bücher sowie Handschriften ähnlich wie konventionelle Bücher, dank eines auf

Flash basierten Readers, gelesen werden. Dieser Reader stellt zahlreiche Funktionalitäten zur Verfügung, die das Lesen der Bücher auf eine attraktive Art und Weise ermöglicht. Bücher durchblättern, in einem Buch suchen und zoomen sind Teil des Angebotes. Am wichtigsten sind aber die Links, welche auf den Buchseiten zu finden sind und sowohl auf weitere Austria-Forum Quellen- und Beitragsseiten als auch auf externe Wissensquellen verweisen. Es ist auch möglich, selbst Links auf den Buchseiten sowie Lesezeichen zu erstellen.

Die Bücher sind über die „Web Books“ Einstiegsseite zu finden. Sie sind verschiedenen Kategorien zugeordnet und erlauben die Suche nach einem bestimmten Titel, nach dem Verfasser oder dem Verlag. Die Abbildung 1 zeigt einen Teil des angebotenen Bücherregals.

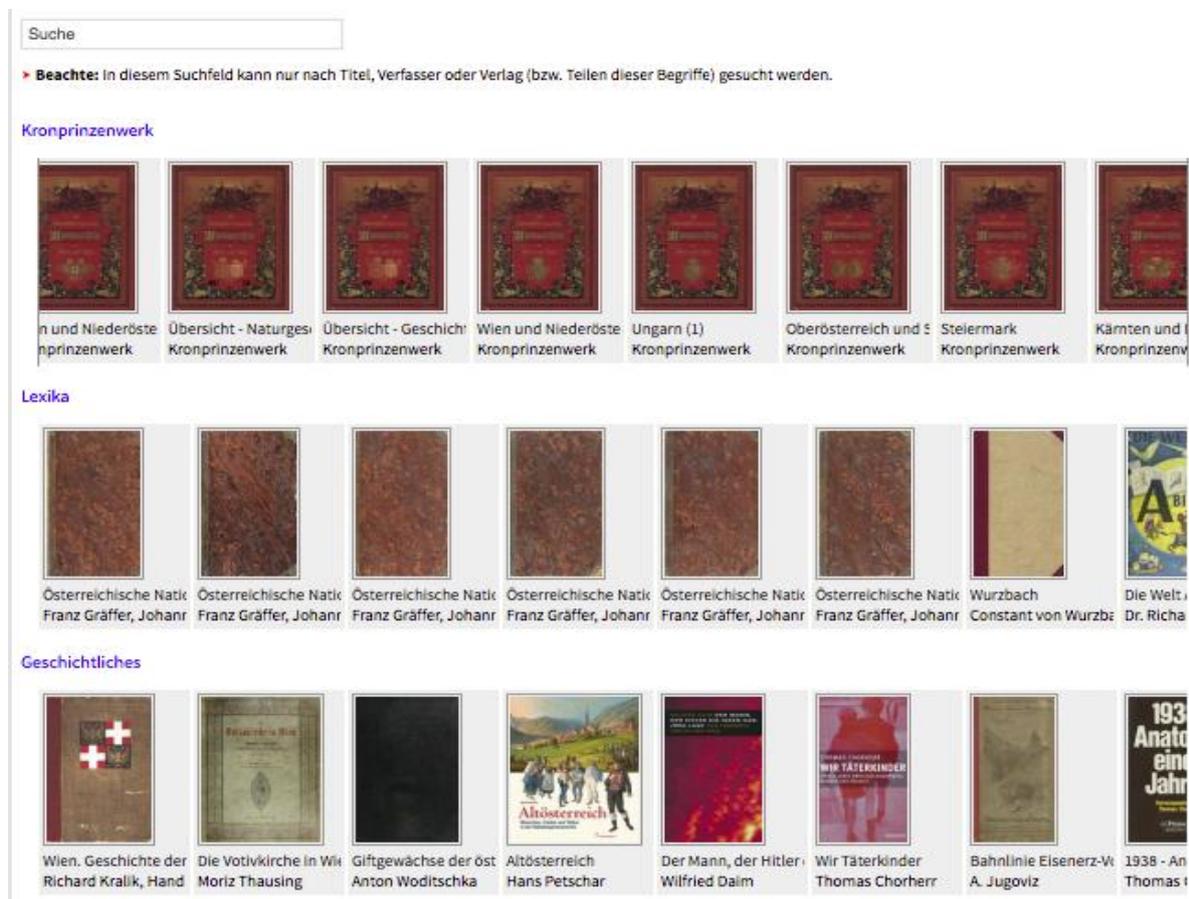


Abbildung 1 Einstiegsseite zu den Austria-Forum Web Books

Um die oben erwähnten Funktionalitäten zu illustrieren, wird ein Lesebeispiel mit Hilfe von Screenshots kurz beschrieben. Dabei handelt es sich um das historische Buch „Die eherne

Mark - Band 2, Ferdinand Krauss", welches in der Austria-Forum Bibliothek zu finden ist.¹¹

Die Abbildung 2 zeigt das mit dem Web Books Reader geöffnete Buch. Diese Ansicht ist in vier Bereiche geteilt:

- **Kopfleiste (Header)**

Am linken Rand sind Symbole für „Links“ und „Lesezeichen“ angebracht, anhand deren Links und Lesezeichen auf den Buchseiten angelegt werden können. In der Mitte der Kopfleiste befindet sich ein Suchfeld und am rechten Rand werden die Leserdaten angezeigt.

- **Fußleiste (Footer)**

In der Fußleiste dient ein „Schieberegler“ dazu, das Umblättern im Buch mit der Computermaus zu ermöglichen. Außerdem ist es ersichtlich, an welcher Stelle des Buches man sich gerade befindet und an welchen Positionen bereits Lesezeichen erstellt wurden.

- **Seitenleiste (Sidebar)**

In dieser Leiste werden die Metadaten des Buches angezeigt, welche die Navigation innerhalb des Buches erleichtern sollen. Thumbnails können einzel- und doppelseitig dargestellt werden. Bei manchen Büchern ist es zusätzlich möglich, das Inhaltsverzeichnis anzeigen zu lassen und ebenso den Inhalt in alphabetischer Reihenfolge zu ordnen.

- **Ansichtsfenster (View port)**

Das Ansichtsfenster stellt jenen Bereich dar, in dem das Buch angezeigt wird. Links neben der Coverseite liefert ein kleines Fenster Auskunft über folgende Metadaten: Autor, Titel, Ort, Verlag, Jahr und Band.

¹¹http://austria-forum.org/ebook/wbin/ambrosius.html#pagenum=0&book=Geschichtliches/Die_eherne_Mark_Band2&layer=Default&pageid=buch_aussen_eherne_mark_01_vs&thumbview=2p

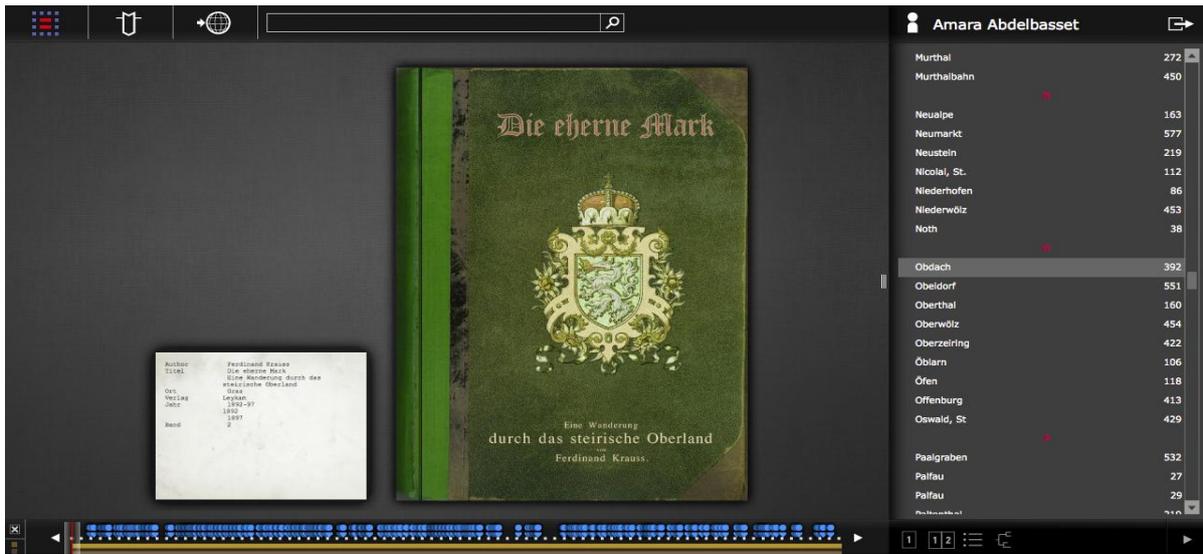


Abbildung 2 Erste Ansicht beim Lesen des Buches

Die Abbildung 3 zeigt eine Beispielseite, an der ein Link angebracht wurde. Beim Anklicken des Linkssymbols wird ein neues Fenster mit dem angezeigten Link im Browser geöffnet.

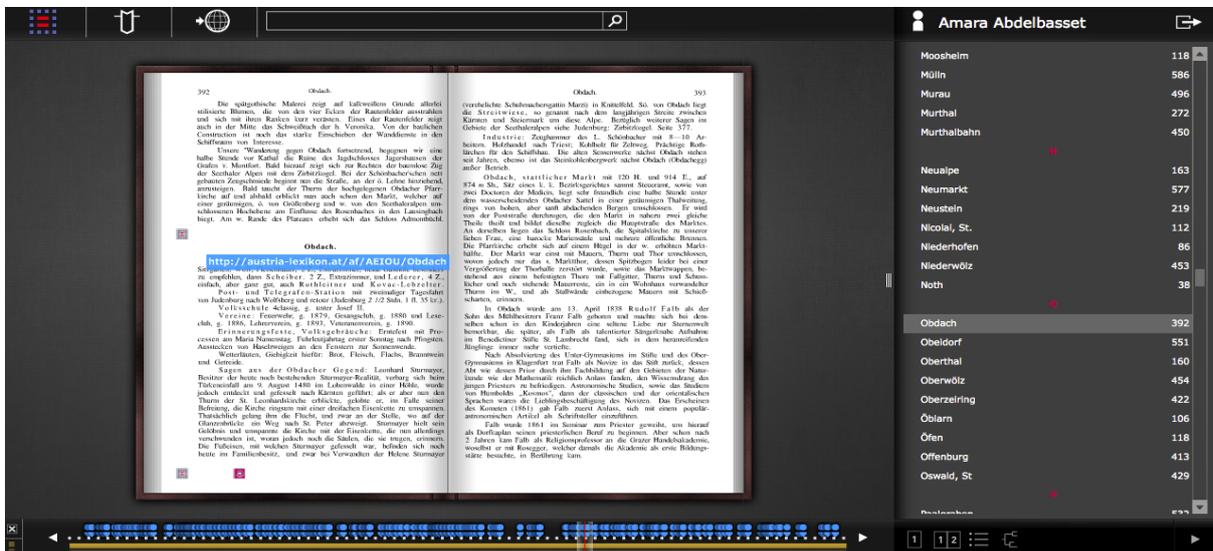


Abbildung 3 Seite mit einem angebrachten Link

Das Lesen des Buches kann auch in einem Einzelseitenmodus stattfinden. Die Abbildung 4 zeigt ein solches Lesebeispiel. Zusätzlich ist in diesem Modus eine Zoom- sowie Umblättereiste vorhanden (in der Abbildung links und rechts neben der angezeigten Buchseite).

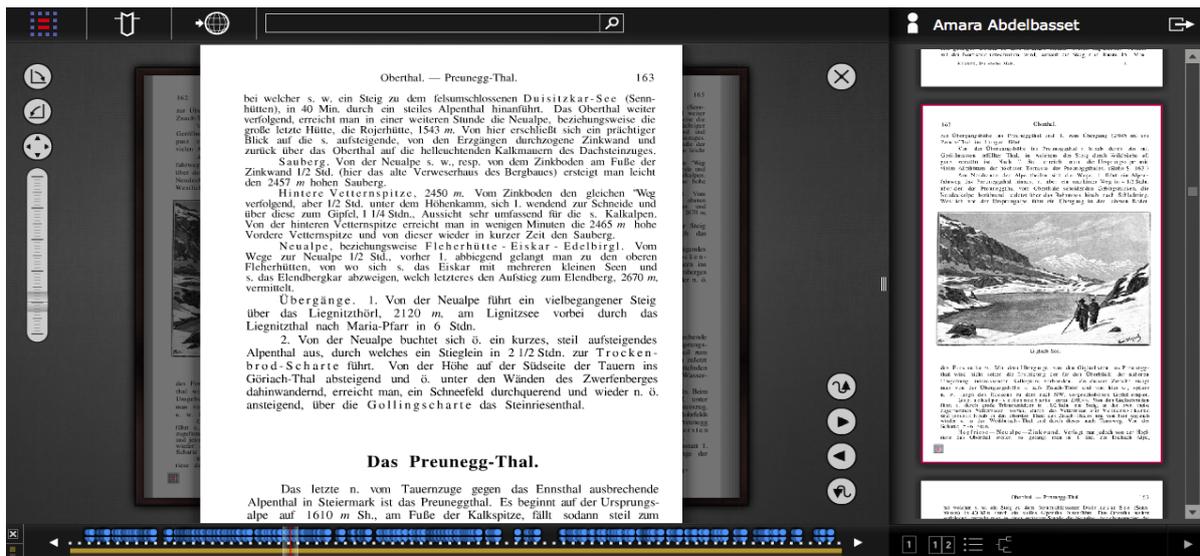


Abbildung 4 Einzelseitenmodus

3 Analyse des Standes der Technik

In diesem Kapitel werden einige Viewers vorgestellt, die im Rahmen verschiedenster Digitalisierungsprojekte für die Veröffentlichung der in digitaler Form vorliegenden Informationen entwickelt und verwendet wurden. Es geht hierbei nicht darum, die unterschiedlichen Anwendungen miteinander zu vergleichen, sondern einen objektiven Überblick über die derzeit wohl bekanntesten Visualisierungssysteme zu liefern.

3.1 Übersicht verfügbarer Viewer

Nachdem die Bücher digitalisiert und für die Visualisierung bzw. Veröffentlichung bereitgestellt worden sind, müssen sie mittels eines Viewers zugänglich gemacht werden.

Unter den verwendeten Viewers, die im Rahmen der zahlreichen Digitalisierungsprojekte der letzten Jahre für die Veröffentlichung der erstellten Digitalisate zum Einsatz kamen, findet man Freie Software (OpenSource), proprietäre Software (ClosedSource) sowie „Shareware“. In der Regel unterstützen die verwendeten Viewers folgende Grundfunktionalitäten: Bildanzeige, Vor- und Zurückblättern, Verkleinern oder Vergrößern des Originalbildes und Anzeige der Metadaten des Buches (z.B Titel, Autoren, Verlag, Jahr, Seitenanzahl etc.).

In Folge werden einige ausgewählte Viewers betrachtet und durch Screenshots kurz beschrieben. Angesichts der mangelhaft verfügbaren Dokumentation kann eine detaillierte Ausführung nicht erfolgen.

3.2 OpenSource

Zu dieser Kategorie gehören Anwendungen, die zur freien Verfügbarkeit veröffentlicht wurden. Die ausgewählten Viewer sind im Rahmen geförderter Digitalisierungsprojekte entstanden, die entweder von Universitäten oder anderen wissenschaftlichen Einrichtungen durchgeführt wurden.

3.2.1 Google Books

Im Jahr 2004 startete Google das Google Books Projekt – damals noch unter dem Namen Google Print – mit dem Ziel, die umfangreichste digitale Bibliothek der Welt zu erschaffen [Zimmer, 12]. Im Rahmen dieses Projektes sollen alle weltweit gedruckten Bücher digitalisiert werden. Diese sollen über das Internet bereitgestellt werden, kostenlos zugänglich sein und eine Volltextsuche ermöglichen [Hetcher, 06].

Bei den Digitalisaten handelt es sich vor allem um Bücher, die entweder sehr selten sind, nicht mehr gedruckt werden oder außerhalb von Bibliotheken nicht erhältlich sind. Deshalb kooperiert Google mittlerweile weltweit mit öffentlichen und privaten Bibliotheken. Nicht bekannt ist allerdings die Anzahl der Bücher, die bereits von Google digitalisiert wurden.

Google bietet ein „Bibliotheksprogramm“ an, das den Zugang zu den bereits digitalisierten Büchern in Form online durchsuchbarer Bücherkataloge ermöglicht. Somit können die Leser nach Eingabe eines Suchbegriffes die bibliographischen Informationen zu den relevanten Büchern erhalten. Je nachdem, ob es sich um gemeinfreie Bücher handelt oder nicht, kann entweder das gesamte Buch oder nur Ausschnitte daraus angezeigt werden [Google Books Library, 14].

Nachfolgend werden anhand von Screenshots die verschiedenen Lese-Modi beschrieben, die von Google Books angeboten werden.

Vollständige Ansicht:

Die Bücher können in dieser Ansicht zur Gänze gelesen werden, wenn sie urheberrechtlich nicht geschützt sind bzw. wenn der Urheberrechtsschutz bereits abgelaufen ist. Weiters wenn der Verlag oder der Autor darum gebeten hat, das Buch vollständig anzuzeigen.

Diese Ansicht ermöglicht es den Lesern, alle Buchseiten anzusehen und falls das Buch gemeinfrei ist, es herunterzuladen, in einer PDF-Version zu speichern oder es auszudrucken.

Die Abbildung 5 zeigt ein Lesebeispiel in dieser Ansicht.



Abbildung 5 Vollständige Einzelseitenansicht auf Google Books

Die Ansicht ist in folgenden drei Teile unterteilt:

- Eine **linksbündige Sidebar**, die neben Titel, Band und Autor des Buches, die Suche ermöglicht und Buchrezensionen einblendet. Weitere Metadaten über das Buch erhält der Leser mit einem Klick auf „Über dieses Buch“. Eine E-PUB Version des Buches kann durch einen Klick auf den roten Button „E-BOOK - KOSTENLOS“ heruntergeladen werden.
- **Buttonsleiste**: Sie beinhaltet (von links nach rechts) Zoom-Funktionen, Buttons für das Umschalten zwischen den Buch-Ansichtsmodi, nämlich Einzelseiten-, Doppelseiten- und Thumbnailmodus, ein Schneiden-Tool, um Inhalte des Buches in Sozialen Netzwerken zu teilen oder auf anderen Webseiten einzubetten. Außerdem einen Link-Button, der einen Link zu der aktuellen Buchseite generiert. Es gibt auch die Möglichkeit, das Buch zu der personalisierten Buchliste hinzuzufügen, als Favorit zu markieren, für eine weitere Lektüre vorzumerken oder als gelesen zu kennzeichnen. Auf der rechten Seite befindet sich neben den Buttons für das Umblättern ein Seitennummern-Inputfeld. Daneben befindet sich eine Schaltfläche, mit der man zu einer erweiterten Suche und auch zu dem angebotenen Hilfesystem gelangt.
- **Content-Bereich**: Hier werden die Buchseiten abgebildet.

Eingeschränkte Vorschau:

Wenn das Buch nicht gemeinfrei ist, aber der Verlag oder der Autor eine Anzeigerlaubnis gegeben hat, kann eine begrenzte Anzahl von Buchseiten als Vorschau angezeigt werden.

Die Abbildung 6 stellt ein Lesebeispiel in dieser Ansicht dar.

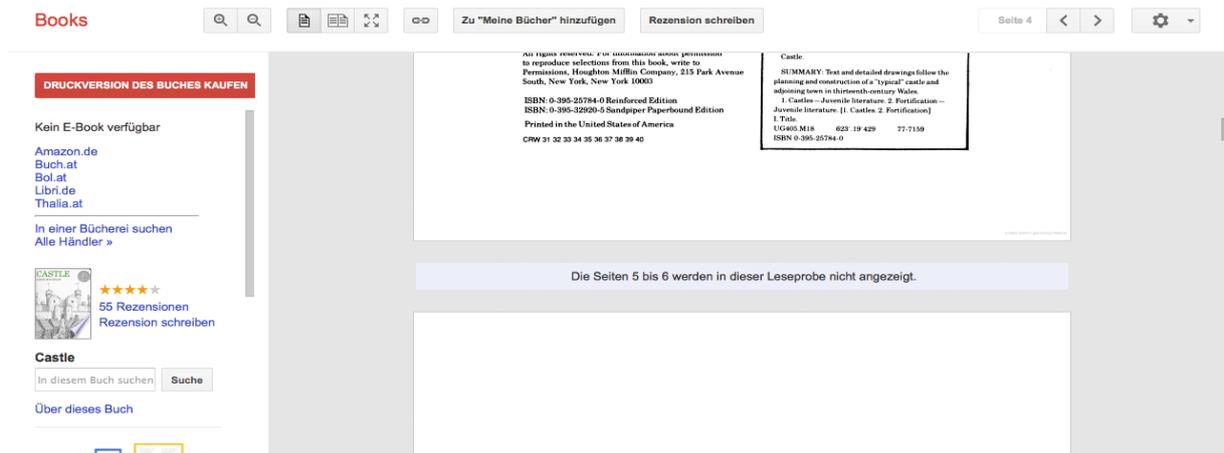


Abbildung 6 Eingeschränkte Vorschau

Unterschiede zu der vorherigen Ansicht ergeben sich in der Sidebar und im Contentbereich. In der Sidebar werden Kaufmöglichkeiten für das Buch angeboten und im Contentbereich finden sich Hinweise zu den ausgeblendeten Buchseiten.

Auszugsansicht:

Diese Ansicht wird wie ein Zettelkatalog (engl. *card catalog*) dargestellt. Sie beinhaltet Informationen über das Buch und zeigt einige Textauszüge an, in denen der eingetragene Suchbegriff vorkommt.

Die Abbildung 7 zeigt Suchergebnisse zu "behavior" in dem Buch "A History of Psychology"

A History of Psychology
Erwin Allen Esper
Saunders, 1964 - 368 Seiten
★★★★★
0 Rezensionen

Neu! Bücher auf Google Play kaufen
Stöbern Sie im größten eBookstore der Welt und lesen Sie noch heute im Web, auf Ihrem Tablet, Telefon oder E-Reader.
Jetzt zu Google Play »

Meine Bücher
Mein Verlauf
Bücher bei Google Play

Im Buch
behavior
79 Seiten stimmen mit dem Suchbegriff "behavior" in diesem Buch überein.

Seite 21
But a great deal of man's routine behavior is "adequate" only because it conforms to social convention and hence has been socially "reinforced"; it would not be adequate in an unrelated social environment. Thus language, attitudes and responses to other members of the family and tribe, and the etiquette of eating and of

Seite 25
Marett can hardly be taken as a strict dichotomy in the life of man. Animistic, magical, and religious behavior and attitudes on the one hand and practical, rational, and scientific behavior and attitudes on the other coexist and interpenetrate one another. Procedures of animistic and magical origin become part of the routine of life and

Seite 270
port published his *Social Psychology*, he introduced a new spirit into what had been largely a "theoretical"—i.e., speculative, mentalistic, and verbose—subject. In his preface he said that "there are two main lines of scientific achievement which I have tried to bring within the scope of this volume. These are the behavior viewpoint

Wo ist der Rest dieses Buches?
Ergebnisse 1-3 von 79

Abbildung 7 Auszugsansicht

Keine Vorschau verfügbar:

Schließlich ist es auch möglich, dass für bestimmte Bücher überhaupt keine Vorschau verfügbar ist und somit keine Inhalte aus dem Buch angezeigt werden. Hier bekommt man lediglich Informationen und Metadaten über das Buch geliefert.

3.2.2 Open Library Viewer

Open Library¹² ist ein Projekt, welches im Jahr 2007 von Internet Archive¹³ ins Leben gerufen wurde, um eine Online-Bibliothek basierend auf einer bibliographischen Datenbank zu erstellen und dessen Inhalt online zu stellen. Laut eigenen Angaben verfügt Open Library über eine Million digitalisierter gemeinfreier Bücher [OL, 13].

¹² <https://openlibrary.org/help/faq/about>

¹³ <http://www.openarchives.org/OAI/OAI-organization.php>

Im Zuge dieser Online-Bibliothek hat Internet Archive einen Book-Reader entwickelt, um die gescannten Bücher und digitalisierten Handschriften, die sowohl in der Open Library selbst vorhanden sind als auch von anderen Organisationen angeboten werden, anhand eines Webbrowsers mit Internetverbindung zugänglich zu machen [IA, 13].

Bei diesem Viewer handelt es sich um eine Webanwendung, die auf HTML und Javascript basiert. Er hat eine einfache und freundliche Benutzeroberfläche und besteht aus drei Teilbereichen: Header, View port und Footer.

Auf der linken Seite zeigt der Header Metadaten des geöffneten Buches an. Auf der rechten Seite befinden sich ein Suchfeld, eine Schaltfläche für eine automatische Umblätterfunktion sowie für die Anzeige der Informationen über das Buch und die zusätzlich zur Verfügung stehenden Formate, in denen es visualisiert werden kann. Daneben befindet sich ein Button für die Funktion „Share this book“, welche die Möglichkeit bietet, entweder einen Link zu dem Buch oder zur aktuell angezeigten Seite zu generieren, der dann in den Sozialen Medien geteilt werden kann. Schließlich befindet sich im Header auch die Schaltfläche „Read this book aloud“ um sich das Buch Abschnitt für Abschnitt laut vorlesen zu lassen.

Im View port werden die Buchseiten entsprechend dem Ansichtsmodus visualisiert.

Im Footer befindet sich ein Navigations-Slider, der ein rasches Auffinden der gewünschten Buchseite erlaubt. Daneben sind Buttons für die Auswahl des Ansichtsmodus (siehe Abbildungen 8–10), für die Funktion „Zoom in/out“ und für das Springen zur nächsten oder vorherigen Seite angebracht. Weiters besteht die Möglichkeit mit einem Klick den Header und Footer gänzlich auszublenden.

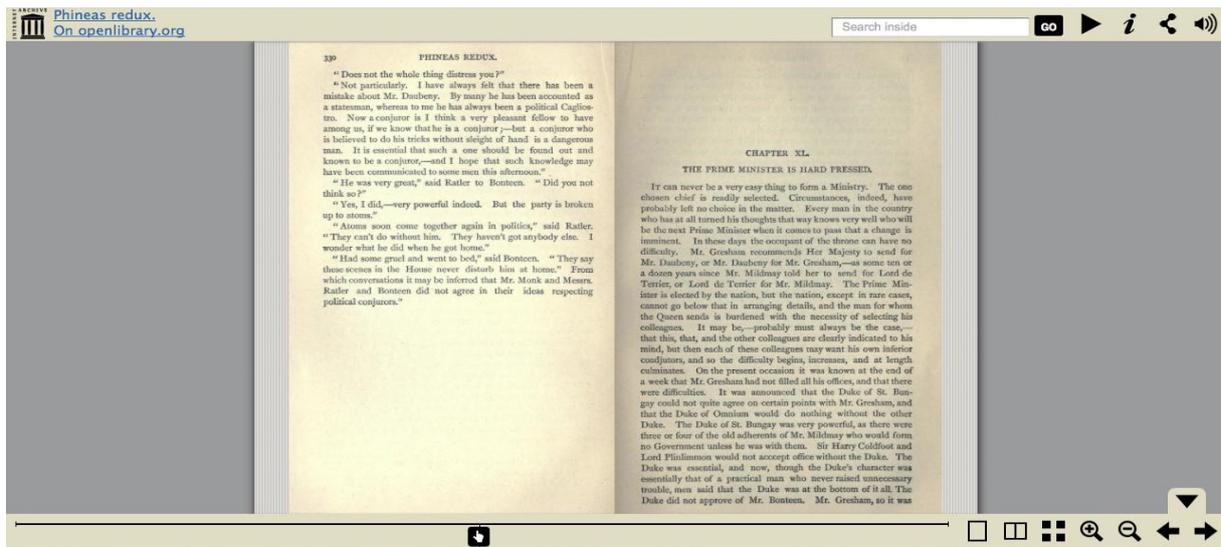


Abbildung 8 Doppelseitenansicht

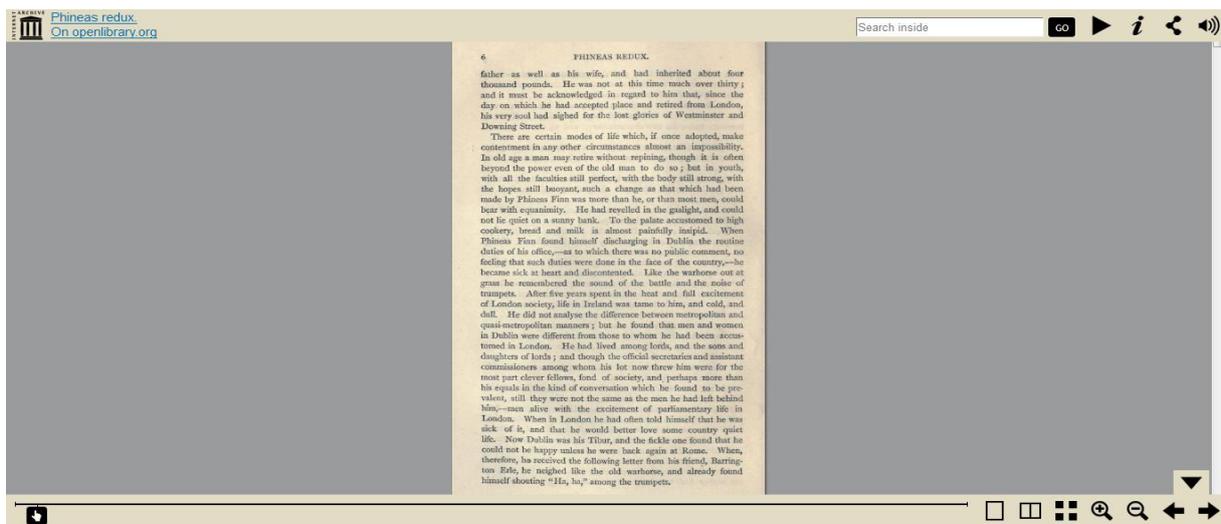


Abbildung 9 Einzelseitenansicht

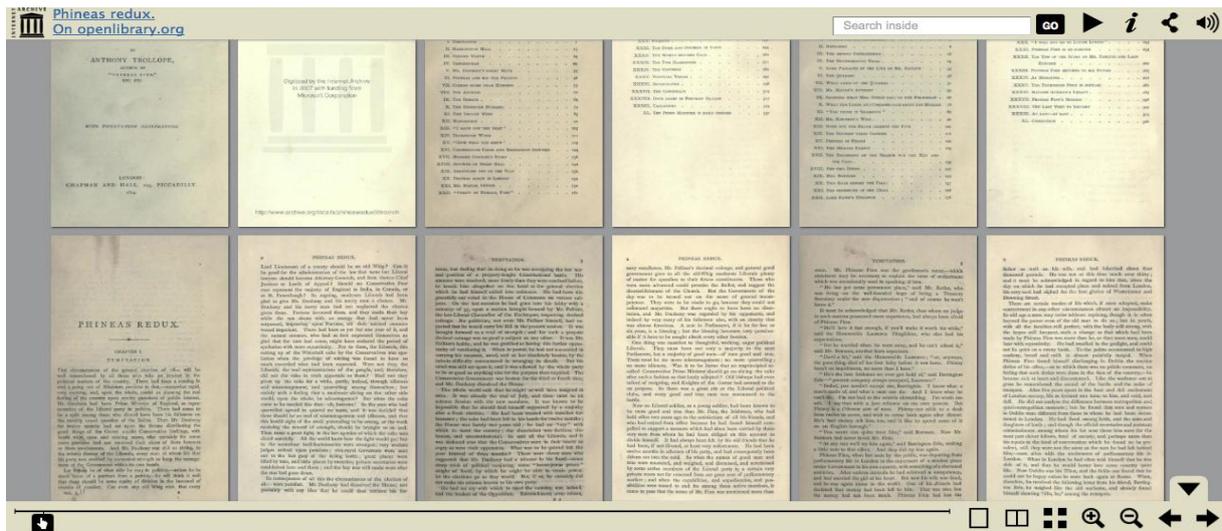


Abbildung 10 Thumbnailansicht

3.2.3 Uni Würzburg Viewer

Die Universitätsbibliothek Würzburg bietet eine virtuelle Bibliothek in Form einer Online-Plattform an. Diese beinhaltet digitalisierte Archivmaterialien, Handschriften und Sonderbestände, die rund um die Uhr online zugänglich sind. Um die Erreichbarkeit der Digitalisate zu gewährleisten, hat die Universität einen webbasierten Viewer entwickelt [Uni-Würzburg, 13].

Der Header beinhaltet die Buttons „erste Seite“, „vorherige Seite“, „nächste Seite“, „letzte Seite“ und dazwischen eine Dropdown-Liste, welche alle Seiten auflistet und das Springen zu einer bestimmten Seite ermöglicht. Daneben befindet sich ein „Vorschau“-Button, der die Thumbnailansicht ein- bzw. ausblendet, ein Button, der die Blätterfunktion aktiviert, eine Zoomoption, eine Dreifachlupe (siehe Abbildung 13), eine Volltextansicht und schließlich noch die Schaltfläche „Services“, die z.B. verschiedene Druck- und Ausgabefunktionen bereithält. Im View port-Bereich werden die Seiten der Digitalisate dargestellt. Der Footer weist im Grunde die gleiche Navigationsleiste wie der Header auf mit der Ausnahme, dass hier kein „Vorschau“- und kein „Services“-Button vorhanden ist.

Der Viewer kann die Digitalisate in zwei verschiedenen Ansichtsmodi darstellen. Beim Öffnen des Digitalisates erscheint zuerst die Einzelseiten-Ansicht (siehe Abbildung 11).



Abbildung 11 Einzelseitenansicht

Ein weiterer Modus stellt die Seiten des Digitalisates in doppelseitiger Form dar. Diese wird durch das Anklicken des „Blättern“-Buttons aktiviert. Diese Ansicht hat im Wesentlichen die Aufgabe, die Umblätternanimation zu ermöglichen. Die Abbildung 12 zeigt ein Lesebeispiel in diesem Modus mit zusätzlichen Navigationsanweisungen zu den verfügbaren Schatflächen.

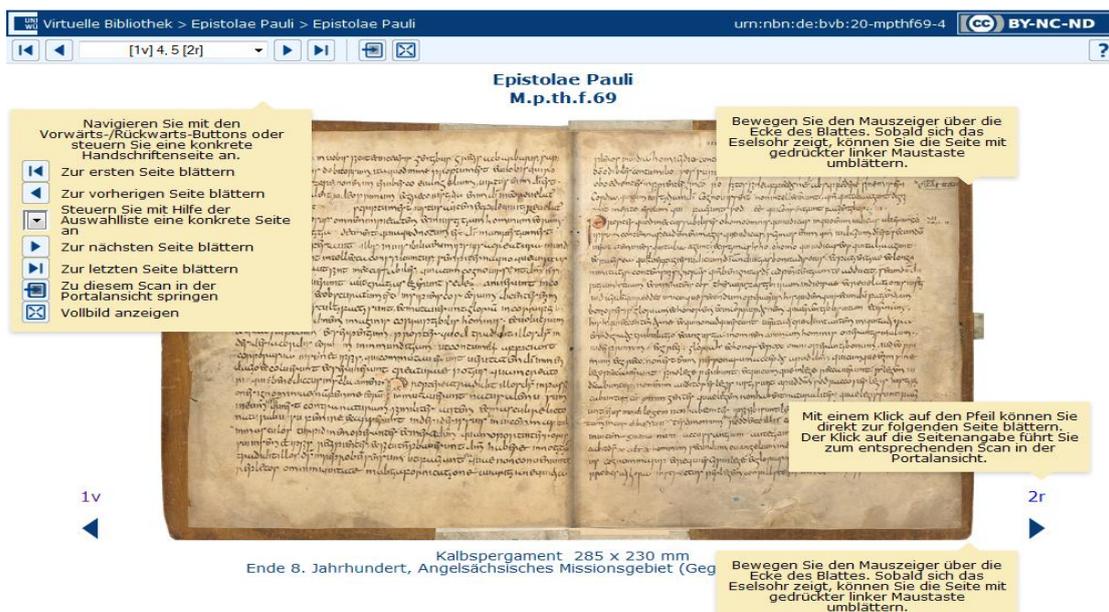


Abbildung 12 Doppelseitige Ansicht mit Umblätternanimation

Die Dreifachlupe ermöglicht ein paralleles Lesen von Originalschrift, stilisierter Ursprungsschrift und Volltextschrift. Zur Zeit verfügen jedoch nur ein paar Handschriften über diese Funktionalität.¹⁴

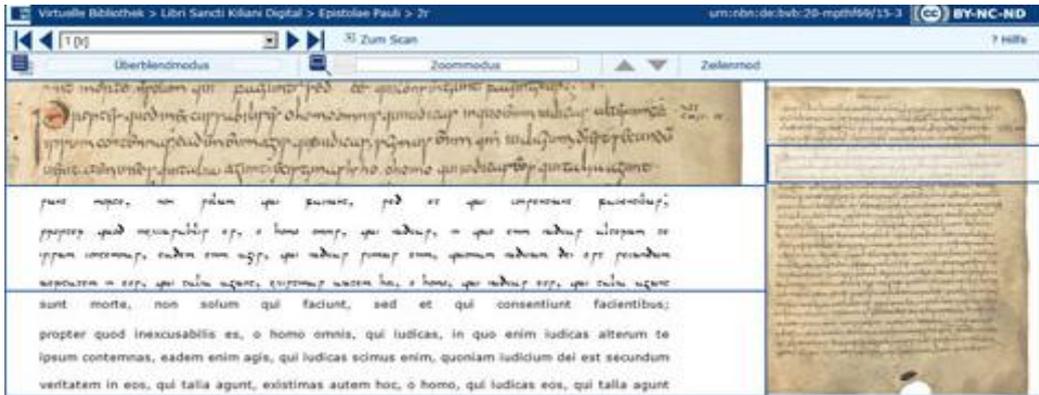


Abbildung 13 Dreifachlupe

Schließlich findet man den „Scan-Textformat-Viewer“ Button. Dieser öffnet in der Form einer Doppelseite links die Scan-Ansicht (ein Facsimile Bild) und rechts die Volltext-Ansicht (siehe Abb. 14).

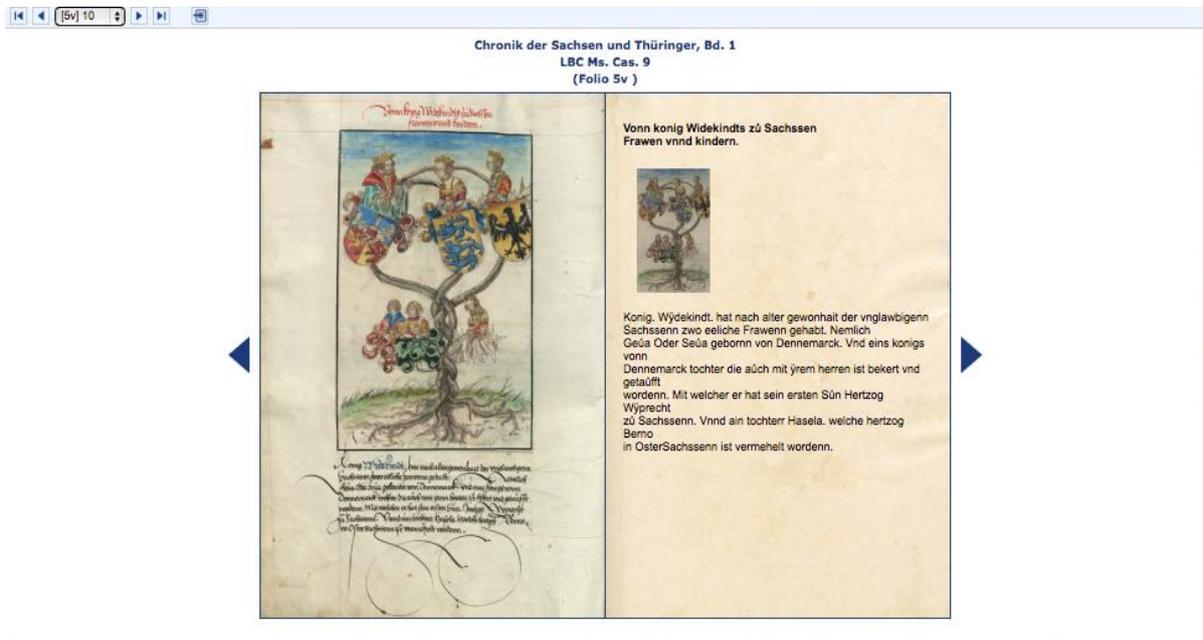


Abbildung 14 Scan-Textformat-Viewer

¹⁴ <http://vb.uni-wuerzburg.de/ub/lskd/dreifachlupe.html>

3.2.4 DFG Viewer

DFG steht für „Deutsche Forschungsgemeinschaft“, eine Selbstverwaltungsorganisation der Wissenschaft in Deutschland, die als privatrechtlicher Verein organisiert ist. Zu ihren Mitgliedern zählen Hochschulen, außeruniversitäre Forschungseinrichtungen sowie die Akademien der Wissenschaften [DFG, 13].

Um eine einheitliche Oberfläche zur Visualisierung der digitalisierten Materialien zu gewährleisten, hat DFG einen Viewer als „Browser-Webdienst“ festgelegt. Der DFG-Viewer basiert auf dem freien Content-Management-System (kurz: CMS) Typo3¹⁵ und kann kostenlos genutzt werden [DFG-Viewer, 13].

Er wurde von der Sächsischen Landesbibliothek – Staats- und Universitätsbibliothek Dresden (SLUB Dresden) in Zusammenarbeit mit drei weiteren Bibliotheken mit Massendigitalisierungsprojekten entwickelt: der Universitäts- und Landesbibliothek Sachsen-Anhalt, der Herzog-August-Bibliothek Wolfenbüttel und der Bayerischen Staatsbibliothek München. Zusätzliche Unterstützung boten die Niedersächsische Staats- und Universitätsbibliothek Göttingen und die Staatsbibliothek zu Berlin-Preussischer Kulturbesitz [ULB Halle, 13].

Ein Lesebeispiel in diesem DFG-Viewer ist in Abbildung 15 angezeigt. Der Headerbereich umfasst Metadaten über das geöffnete Werk (Titel, Jahr, Autor, Verlag etc.) sowie eine Dropdown-Liste mit allen Seitennummern und weist auf die aktuelle Seite hin. Zusätzlich ist eine Buttonleiste angebracht, die aus folgenden Elementen besteht:

- der „Katalog-Nachweis“-Button führt zu zusätzlichen Metadaten,
- rechts davon sind zwei Buttons, die es ermöglichen zwischen einer doppelseitigen und einer einseitigen Ansicht zu wechseln,
- im Anschluss befinden sich Download-Schaltflächen (entweder kann die aktuelle Seite oder das gesamte Werk heruntergeladen werden), die nur dann aktiviert sind, wenn für das jeweilige Buch eine Downloadoption besteht,
- es folgen die Buttons „Zoom in“ und „Zoom out“
- und schließlich findet man die gewöhnliche Navigationsleiste, welche „erste Seite“, „5 Seiten zurück“, „vorherige Seite“, „nächste Seite“, „5 Seiten weiter“ und „letzte Seite“-Buttons beinhaltet;

¹⁵ <http://typo3.org>

Im View port-Bereich werden die Seiten des Digitalisates angezeigt. Rechts ist ein Randfenster (Sidebar), welches das Inhaltsverzeichnis des Werkes auflistet, damit Leser auf beliebige Inhaltspunkte direkt zugreifen können. Der Footer weist die gleiche Navigationsleiste wie der Header auf.



Abbildung 15 DFG-Viewer

3.3 Kommerzielle Produkte

Bei dieser Kategorie handelt es sich um Softwares, die nicht frei verfügbar sind und bei denen auch keine Veränderungen erlaubt sind, mit anderen Worten, fehlt der Zugriff auf den Quellcode um etwaige Änderungen bzw. Erweiterungen vorzunehmen. Sharewares dürfen allerdings vor dem Kauf getestet werden.

Im Anschluss wurde eine Auswahl aus der Vielzahl von Systemen und deren Funktionalitäten getroffen, um einen Überblick zu ermöglichen.

3.3.1 Crocodoc

Crocodoc wurde im Jahr 2007 von MIT Ingenieuren gegründet. Anfangs war Crocodoc eine auf Flash basierte Technologie, die es Lesern ermöglichte, PDF Dateien hochzuladen und eine Browser Version des Dokuments zu erhalten. Diese konnte dann mit anderen geteilt und mittels Annotationen und Kommentaren im Rahmen einer kollaborativen Arbeitsweise benutzt werden.

2011 wurde eine neue, auf HTML basierende Version von Crocodoc vorgestellt. Seitdem bietet Crocodoc einen Cloud-Dienst an, der Office- und/oder PDF-Dokumente in HTML5 umwandelt und u.a. von Dropbox, Microsoft, Box, LinkedIn, der Microsoft-Tochter Yammer und SAP genutzt wird [TC, 13].

Crocodoc verarbeitet sowohl PDF-Dateien als auch Office-Dokumente und sorgt für deren Umwandlung in HTML, CSS und SVG. Die resultierenden Dokumente lassen sich sehr einfach in Webanwendungen integrieren und darüber hinaus wird ihre Funktionalität in allen gängigen Browsern sowie auf mobilen Geräten gewährleistet [Golem, 13].

Die Abbildung 16 zeigt eine Einzelseiten-Ansicht. Diese Ansicht bietet einen Zoom, ein Vor- oder Zurückblättern sowie das Hinzufügen von Kommentaren bzw. Annotationen.

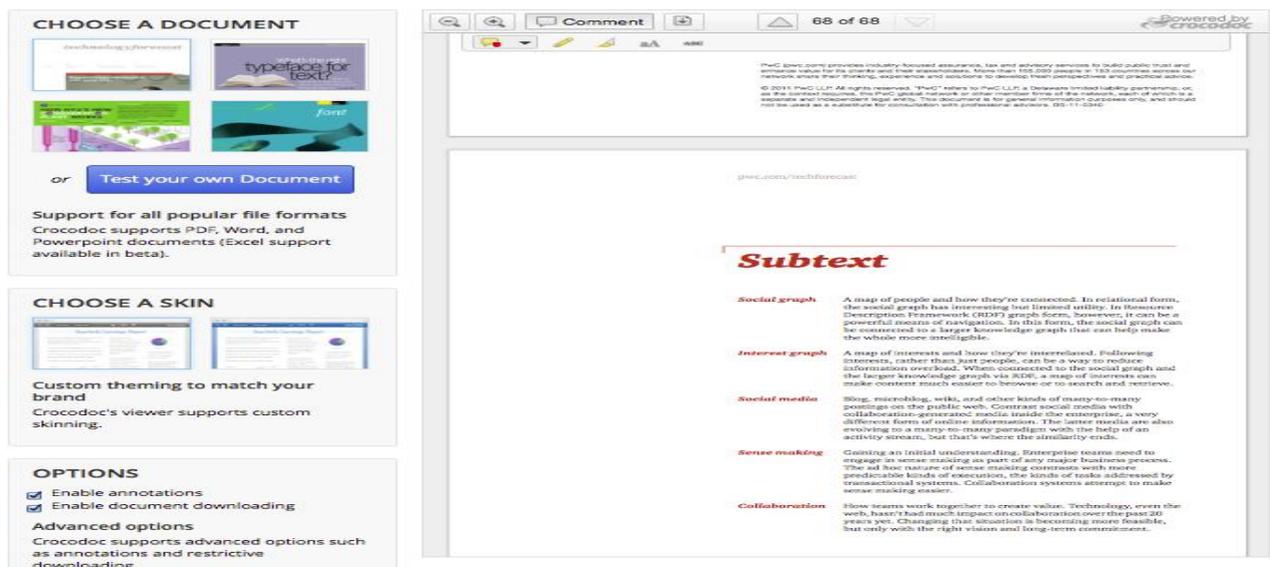


Abbildung 16 Crocodoc Einzelseitenansicht

In der in Abbildung 17 dargestellten Doppelseiten-Ansicht ist hingegen nur das Umblättern möglich (Demo Version).

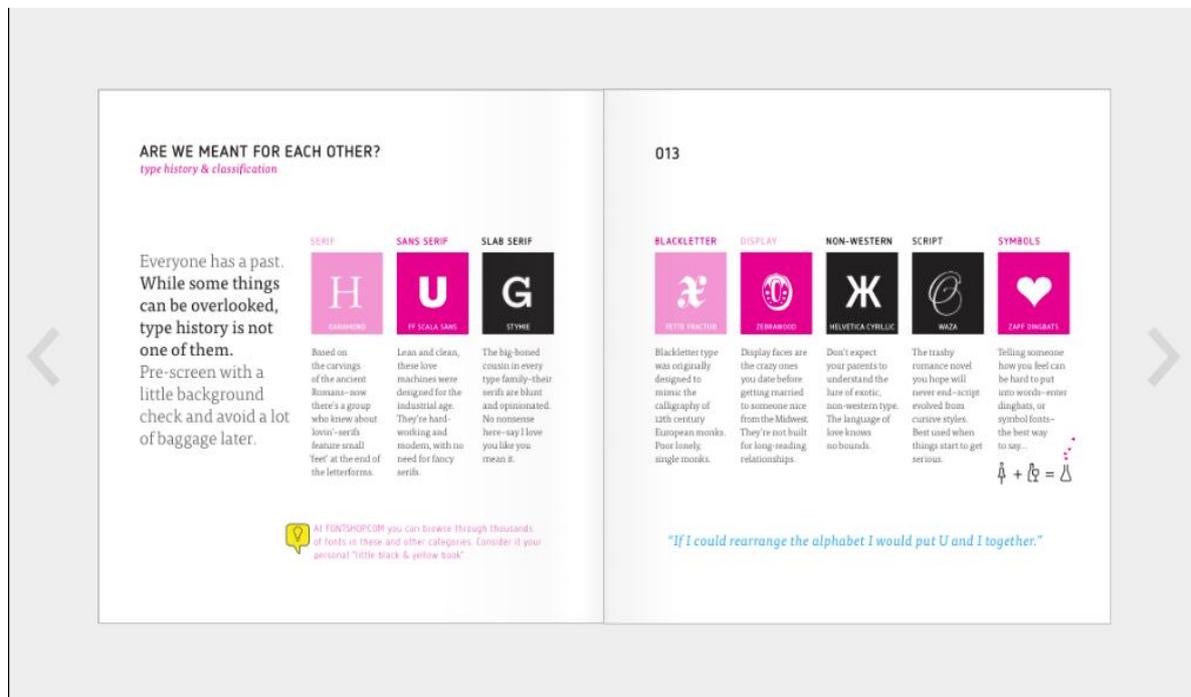


Abbildung 17 Crocodoc Doppelseitenansicht

3.3.2 Scribd

Scribd ist eine Dokumentenaustausch-Plattform, die es Nutzern ermöglicht, Office- und PDF-Dateien hochzuladen, direkt im Browser anzuzeigen und anderen Personen online zur Verfügung zu stellen [ZDNet 2008].

Benutzer dieses Webportals können nicht nur PDF-, Postscript-, Word-, Excel- und Powerpoint-Dateien hochladen, sondern auch RTF-, Open-Office-, ODF-Dokumente etc. sowie verschiedene Bildformate und diese je nach Wunsch entweder veröffentlichen oder ausschließlich für den privaten Gebrauch vorsehen. Die hochgeladenen Dokumente werden in das HTML5-Format umgewandelt und im Browser visualisiert.

Funktionalitäten wie Zoomen, Suchen, Kommentieren sind im Scribd vorhanden, allerdings kann man sie in einer Demo-Version nicht testen.

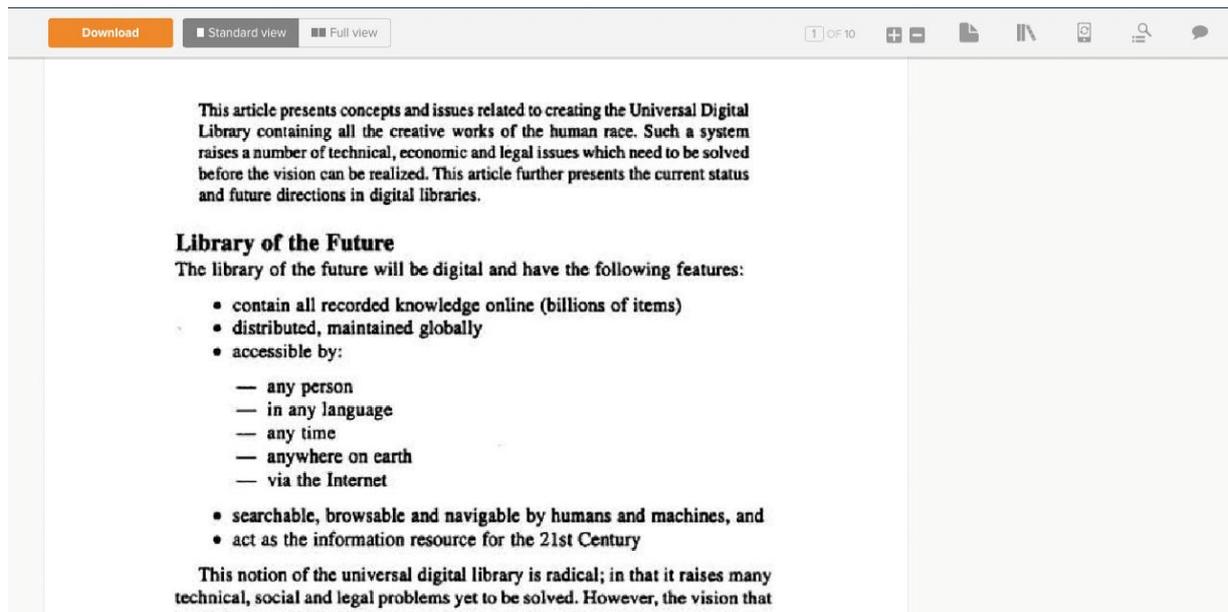


Abbildung 18 Scribd Demo: Einzelseitenansicht

3.4 Abschließende Auswertung

Diese Analyse bezieht sich nur auf die frei verfügbaren Viewer, da bei den „Sharewares“ weder Anpassungen noch Ergänzungen fehlender Funktionalitäten erlaubt sind.

Die oben genannten Viewers werden für Visualisierungszwecke eingesetzt, um die zahlreichen im Zuge der Digitalisierungsprojekte bereitgestellten Digitalisate der Öffentlichkeit zugänglich zu machen.

Diese Softwares erfüllen die Basisanforderungen, indem sie über Grundfunktionalitäten verfügen, die für das Lesen der Digitalisate unabdingbar sind. Sie ermöglichen insbesondere das Umblättern, Zoomen und die Anzeige von Metadaten.

In dem DFG-Viewer wird ein Inhaltsverzeichnis in der Sidebar angezeigt, damit eine übersichtliche Navigation innerhalb des Werkes möglich ist.

Der von der Uni Würzburg verwendete Viewer bietet für einige Digitalisate eine Dreifach-Leselupe an, welche ein paralleles Lesen von Originalschrift, stilisierter Ursprungsschrift und Druckschrift ermöglicht. Weiteres verfügt dieser Viewer über eine „Scan-Textformat“-Ansicht, welche in Form einer Doppelseite links die Scan-Ansicht und rechts die Volltext-Ansicht darstellt.

Trotz der vielen Funktionalitäten, welche die Viewers aufweisen, gibt es noch einige unerfüllte Anforderungen.

- **Soziale Medien**

Nicht nur das gesamte Werk soll anhand eines Hyperlinks in Sozialen Netzwerken geteilt werden können, sondern auch einzelne Seiten sowie bestimmte Teiinhalte (Absätze oder Sätze). Außerdem sollen Annotationen, Kommentare, Lesezeichen, Textmarkierungen und andere Funktionalitäten, die eine kollaborative Leseatmosphäre schaffen, bereitgestellt und unterstützt werden.

- **Formate der Digitalisate**

Die verschiedenen Formate der Digitalisate sollen bei der Visualisierung unterstützt werden, indem für jedes Format ein Lese-Layer angeboten wird.

- **Benutzerfreundlich**

Der Viewer soll unkompliziert bedient werden können. Das soll durch eine intuitive und einfache Benutzeroberfläche sowie Bedienungsanweisungen im Hilfesystem erreicht werden.

- **Volltextsuche**

Der Viewer soll Suchfunktionalitäten unterstützen, sodass die Suche nach dem Volltext oder nach Schlüsselwörtern in Digitalisaten möglich wird.

- **Browser und Mobilgeräte**

Die angebotene Visualisierung soll in allen gängigen Browsern sowie auf mobilen Geräten einheitlich angeboten und gewährleistet werden.

- **Metadaten**

Metadaten sollen an den Buchseiten von Benutzern angehängt werden können. Außerdem soll eine Extrahierung der Content-Metadaten und somit die Inhaltsuche möglich sein.

- **Crowdsourcing**

Die Benutzer sollen die Möglichkeit erhalten, die Fehler in den OCR-Ausgaben auszubessern.

- **Benutzergruppenverwaltung**

Es soll möglich sein eigene Benutzergruppen zu erstellen, damit vorgenommene Annotierungen und Verlinkungen von der jeweiligen Gruppe einsehbar sind. Diese Funktionalität kann beispielsweise im Rahmen von Schulprojekten äußerst hilfreich sein.

4 Warum einen eigenen E-Book Viewer?

Es wurde deutlich gemacht, dass die in Frage kommenden Viewer hinsichtlich ihrer Funktionalität einige Kritikpunkte zulassen. Vor allem im Hinblick auf „soziale Netzwerke“ können sie keine zufriedenstellende Gestaltung anbieten. Aus diesem Grund hat sich das Austria-Forum dazu entschlossen, die Anforderungen neu zu bearbeiten und darauf basierend einen eigenen Web Books Viewer zu erstellen.

In diesem Kapitel werden die einzelnen Softwareanforderungen definiert und hinsichtlich dem Grad ihrer Realisierung klassifiziert.

4.1 Anforderungsklassen

Um den Vorgang bei der Implementierung nachvollziehbar zu machen, werden die einzelnen Anforderungen nach der ihnen zukommenden Priorität bei der Realisierung drei verschiedenen Klassen zugeordnet. Diese wurden mit den folgenden Eigenschaften definiert:

- **Klasse 1:**

Diese Klasse gruppiert jene Eigenschaften, die als elementar zu betrachten sind und deshalb auch ohne Einschränkung implementiert wurden.

- **Klasse 2:**

Diese Klasse beinhaltet alle Funktionalitäten, die optional realisiert wurden.

- **Klasse 3:**

Eigenschaften dieser Klasse sind bei der Implementierung im Rahmen dieser Arbeit nicht vorgesehen, werden aber bei der Beschreibung der Architektur berücksichtigt. Es handelt sich dabei um einige erweiterbare Funktionalitäten.

- **Klasse 4:**

In dieser Klasse befinden sich Anforderungen, die für die Zukunft geplant sind. Sie stellen das Kernstück des Viewers dar. Dieser Klasse sind die in den Abschnitten 4.3 bis 4.6 beschriebenen Anforderungen zuzuordnen.

4.2 Benutzeranforderungen

Einige Anforderungen haben sich durch die Analyse der in Kapitel 3 besprochenen Softwares herauskristallisiert und andere sind durch die Entscheidung für einen eigenen Viewer hinzugegetreten. Diese Anforderungen betreffen sowohl die System- als auch die Benutzerebene.

Weil die Benutzeranforderungen für die Realisierung des in dieser Arbeit präsentierten Viewers eine entscheidende Rolle spielen, wird hier ausschließlich auf sie näher eingegangen. Es kann zwischen funktionalen und nicht-funktionalen Anforderungen unterschieden werden.

4.2.1 Funktionale Anforderungen

- (1) **Authentifizierung:** Der User soll sich in dem System anmelden bzw. abmelden können. (Klasse 3)
- (2) **Bibliothek-Ansicht:** Eine Übersicht in der alle verfügbaren Bücher der virtuellen Bibliothek verschiedenen Kategorien zugeordnet sind. Innerhalb der einzelnen Kategorien soll eine Bücherliste den Suchprozess erleichtern. (Klasse 1) Zusätzlich soll die Suche nach Titel, Autor oder Verlag angeboten werden sowie eine inhaltliche Suche, welche dadurch ermöglicht wird, dass Metadaten aus den Büchern gewonnen werden. (Klasse 3)
- (3) **Buch-Ansicht:** Nachdem der User ein Buch ausgewählt hat, wird dieses im View port angezeigt. Dort soll eine Auswahl zwischen verschiedenen Anzeigemodi möglich sein. (Klasse 1)
- (4) **Doppelseitenmodus:** Die Buchseiten sollen in diesem Modus doppelseitig dargestellt werden. Diese Darstellungsweise ähnelt der Ansicht eines aufgeschlagenen Buches. (Klasse 1)
- (5) **Einzelseitenmodus:** Die Buchseiten sollen wahlweise auch einzeln angezeigt werden können. (Klasse 1)
- (6) **Thumbnailmodus:** Alle Buchseiten sollen als Thumbnails angezeigt werden. Die jeweilige Darstellung in diesem Modus hängt davon ab, ob die Seiten zuvor doppel- oder einzelseitig angezeigt wurden. Der Thumbnailmodus dient dazu, einen Überblick über das gesamte Werk zu liefern. (Klasse 2)

- (7) **Moduswechsel:** Der User soll bequem zwischen den angebotenen Modi wechseln können. Dies kann beispielsweise mit einer Dropdown-Liste realisiert werden. (Klasse 1)
- (8) **Anzeige der aktuellen Buchseite:** Es soll ein Feld vorhanden sein, das die aktuelle Buchseite anzeigt. Der User soll von hier aus auch zu einer bestimmten Buchseite springen können, indem er die entsprechende Seitennummer im Feld eintippt. (Klasse 2)
- (9) **Buchlayers:** Das Buch soll sowohl in dem originalen Scanformat (=Facsimile) als auch in einem OCR-Format (z.B. PDF) gelesen werden können. (Klasse 1)
- (10) **Korrektur-Layer:** Der User soll die beiden Layers der Buchseite (Facsimile /OCR) nebeneinander anzeigen lassen können um zwischen ihnen zu vergleichen und eventuell vorhandene Fehler zu lokalisieren (Klasse 2) und anschließend zu korrigieren. (Klasse 3)
- (11) **Umblättern:** Es soll möglich sein, das geöffnete Buch umzublättern z.B. mit Hilfe eines „Vorherige Seite“ und „Nächste Seite“-Buttons. Auch die Steuerung mit „Key-down Events“ (z. B. Tastaturpfeil rechts/links) soll ermöglicht werden. (Klasse 1)
- (12) **Zoomen:** Der User soll die aktuelle Ansicht des Buches entweder vergrößern, verkleinern oder in der optimalen Größe, die abhängig von der jeweiligen Bildschirmgröße berechnet wird, anzeigen lassen können. (Klasse 1)
- (13) **Metadaten:** Metadaten zu dem Buch wie Titel, Autor, Erscheinungsdatum usw. sollen angezeigt werden. (Klasse 1) Des Weiteren sollen dem Leser Inhaltsdaten bereitgestellt werden (siehe Abschnitt 4.5).
- (14) **Thumbnails in der Sidebar:** Je nach Ansichtsmodus sollen die Thumbnails doppel- bzw. einseitig in der Sidebar angezeigt werden. (Klasse 1)
- (15) **Buchseite drehen:** Im Einzelseitenmodus soll der User die Buchseiten im und gegen den Uhrzeigersinn drehen können. Diese Funktionalität ist deshalb wichtig, weil die angezeigten Bilder teilweise im Querformat dargestellt sind und deshalb gedreht werden müssen. (Klasse 3)
- (16) **Suche im Buch:** Die Suche nach einem bestimmten Wort/Satz soll ermöglicht werden. Der User gibt ein oder mehrere Suchwörter in das Suchfeld ein und erhält eine Liste mit den Suchergebnissen und kann daraus eine Auswahl treffen. (Klasse 1)
- (17) **Vernetztes Lesen:** Der User soll Annotationen bzw. Notizen erstellen und Seiten mit Hilfe von Lesezeichen vormerken können. Darüber hinaus sollen Hyperlinks zu

externen und/oder Austria-Forum-Seiten erstellt und mit dem Inhalt des Buches verknüpft werden können. (Klasse 3)

- (18) **Kollaboratives Lesen:** Es soll möglich sein, dass Mitglieder einer virtuellen Lesegruppe zu einem Buch parallel Annotationen verfassen, Seiten vormerken und Wörter, Sätze, Textstellen usw. highlighten, sodass diese wechselseitig von den Mitgliedern einsehbar sind. Dafür ist eine effiziente Gruppenverwaltungsstrategie erforderlich. (Klasse 3)
- (19) **Contentsharing:** Die Sozialen Netzwerke sollen maßgeblich in die Konzeption des Viewers einbezogen werden. Der User soll Inhalte aus einem Buch z. B. via Facebook oder Twitter teilen sowie ein Buch seinen Freunden weiterempfehlen können. Dies soll nur dann erlaubt sein, wenn damit keine urheberrechtliche Verletzung verbunden ist. (Klasse 3)
- (20) **Bildersuche:** Die Suche nach einem bestimmten Bild im Buch soll eine mögliche Option darstellen. Nachdem der User den gesamten Bildtitel oder Teile davon eingibt, sollen als Suchergebnis jene Buchseiten erscheinen, die Bilder mit den jeweiligen Informationen enthalten. Eine solche Suche benötigt vorerst die Extrahierung der die Bilder betreffenden Metadaten der Buchseiten. Diese umfassen den Buchtitel sowie die relative Positionierung des Bildes in der Buchseite. Im Rahmen dieser Arbeit wurde eine spezielle Software implementiert, welche die Extrahierung solcher Metadaten bewerkstelligt. Diese Software wird im Abschnitt 6.4 erläutert. (Klasse 2)
- (21) **Bilderansicht:** Aufbauend auf der Bildersuche soll ein zusätzlicher Ansichtsmodus bereitgestellt werden, welcher ausschließlich die aus den Buchseiten extrahierten Bilder anzeigt. Die Suche nach einem bestimmten Bild anhand von Suchbegriffen soll auch in dieser Ansicht möglich sein. (Klasse 3)
- (22) **Hilfesystem:** Auf diese Weise soll der User bei der Bedienung des Viewers unterstützt werden. (Klasse 3)

4.2.2 Nicht-funktionale Anforderungen

Den folgenden Anforderungen kommen wegen ihrer Wichtigkeit oberste Priorität zu und werden deshalb gänzlich der Klasse 1 zugeordnet.

- (1) **Benutzbarkeit:** Die Oberfläche des Viewers soll unkompliziert gestaltet und somit intuitiv bedienbar sein. Der User soll ohne jegliches Vorwissen und mit durchschnittlicher Interneterfahrung den Viewer bedienen und Bücher lesen können.
- (2) **Plattformunabhängigkeit:** Der Viewer soll auf allen Plattformen funktionsfähig sein.
- (3) **Browserkompatibilität:** Der Viewer soll mit allen gängigen Browsern und auf allen Mobilgeräten lauffähig sein.
- (4) **Quellcode-Vollständigkeit:** Der Quellcode soll ausreichend dokumentiert und hinsichtlich des Datums des letzten *Commits* möglichst aktuell sein, um den Änderungsprozess zu gewährleisten.
- (5) **Wartbarkeit und Erweiterbarkeit:** Die Wartung und die Weiterentwicklung des Viewers soll ermöglicht werden. Dabei soll sowohl die funktionale Erweiterung als auch die Anpassung an neue Serverschnittstellen, z. B. bei einer Softwareaktualisierung im Zuge der Veröffentlichung neuer Server APIs, berücksichtigt werden.

4.3 Interaktives und Kollaboratives Lesen der Bücher

Das definierte Ziel ist also nicht nur, Bücher mit Österreichbezug zur Verfügung zu stellen, sondern auch die Qualität des Lesens weitgehend zu unterstützen und in einer interaktiven Weise zu ermöglichen. Dies wird dadurch geschehen, dass dem Leser zusätzliche Layers angeboten werden, die sich durch die Kombination einiger Funktionalitäten ergeben.

An dieser Stelle werden zwei neue Layers präsentiert. Der Erweiterungs (engl. Enhancement) Layer soll ein kollaboratives Lesen im Sinne von Einzel- oder Gruppenannotationen ermöglichen sowie das Erstellen und Editieren von Hyperlinks. Der Communication Layer soll ein interaktives Lesen dadurch gewährleisten, dass er eine Verknüpfung des Buches mit den Sozialen Netzwerken sowie eine Verbindung mit anderen Wissensnetzen und -räumen erlaubt.

Die Abbildung 19 zeigt eine Übersicht der erweiterten Layer-Architektur des Viewers.

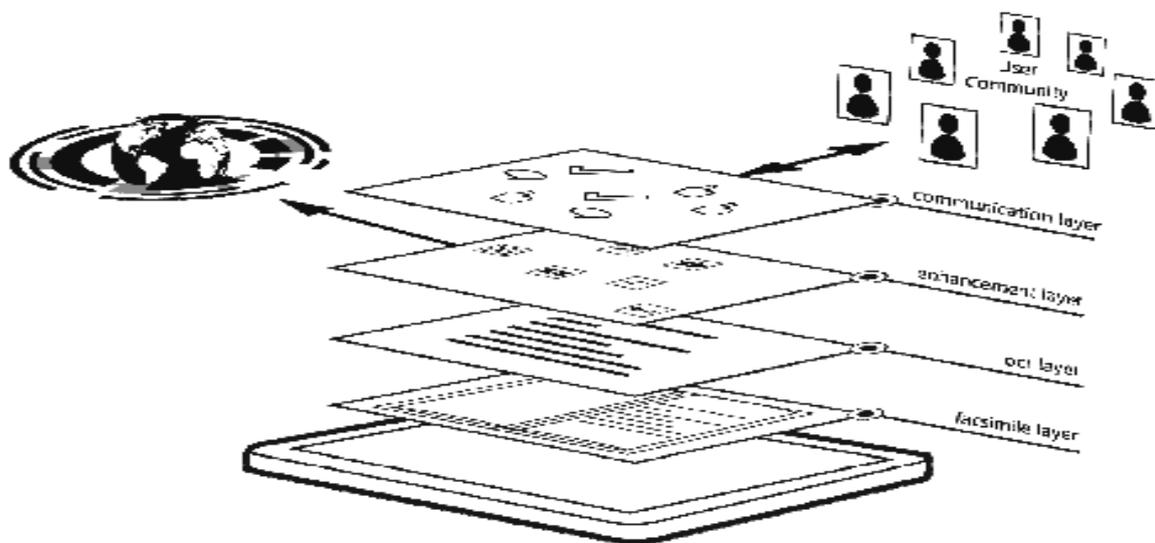


Abbildung 19 Layers des Web Books Viewers [Müller & Maurer, 11]

4.3.1 Enhancement Layer

Dieser Layer soll den Viewer um Funktionalitäten erweitern, die das Annotieren von Büchern für einzelne Leser sowie für Gruppen von Lesern ermöglichen. Außerdem soll der Inhalt des Buches dadurch ergänzbar gemacht werden, dass Hyperlinks im Buch hinzugefügt werden dürfen. Dabei kann es sich sowohl um Hyperlinks zu Websites als auch um Links zu Multimedia-Objekten wie z.B. Video-, Bild- oder Audio- Dateien handeln.

Der Enhancement Layer soll grundsätzlich folgende essentielle Funktionalitäten unterstützen bzw. anbieten:

- **Seiten markieren:**

Der Leser soll die Möglichkeit haben, ein oder mehrere *Post-its* zu erstellen. Diese erscheinen am Rand der Buchseite und ragen auch darüber hinaus, damit es ersichtlich ist, an welchen Seiten im Buch *Post-its* vorhanden sind. Ihnen kann ein optionaler Text und auch verschiedene Farben zugewiesen werden. Diese Markierungen können sowohl von einem einzelnen Leser als auch in einer Lesegruppe hinzugefügt und bearbeitet werden.

- **Links aufsetzen:**

Der Leser soll Hyperlinks zu externen Websites bzw. Wissensnetzen sowie zu den AF-Beitragseiten auf den Buchseiten hinzufügen können. Außerdem soll das Verknüpfen mit Multimediadateien möglich sein.

- **Kommentare und Notizen:**

Der Leser soll auf den Buchseiten beliebig viele Kommentare und Notizen hinzufügen können. Diese sollen ebenso im Rahmen von Lesegruppen von deren Mitglieder erstellt werden können. Sie beinhalten Informationen und Anmerkungen in Textform, die der Leser wie in einem konventionellen Buch verfassen kann. Es soll auch die Möglichkeit geben, diese Notizen zu editieren und/oder mit anderen Lesern zu teilen.

- **Tagging:**

Der Leser soll Tags auf den Buchseiten erstellen können. Diese können sowohl von Gruppenteilnehmern als auch von einzelnen Lesern angehängt werden. Tags haben im Wesentlichen die Aufgabe, Stichwörter, Personen- oder Plätzenamen, Events oder allgemeine Objekte innerhalb des Buches zu klassifizieren und einzubinden. Sie werden auf dem Server mittels URIs indiziert. Sie dienen als zusätzliche Metadaten (siehe Abschnitt 4.5), die von den Lesern ergänzt werden können. Der Leser soll auch die Möglichkeit haben, die vorhandenen Tags zu bearbeiten.

- **Highlighten:**

Dies ist eine digitale Form des Filzstiftes (engl. *felt-tip*). Es soll dafür verwendet werden, die Aufmerksamkeit auf Teile der Buchseiten (oder sogar ganze Seiten) durch Markierung mit einer beliebigen Farbe zu richten. Die auf den Buchseiten enthaltenen Bilder sollen von dieser speziellen Funktion ausgeschlossen bleiben.

- **Bilder markieren und annotieren:**

Der Leser soll einzelne oder mehrere Bilder auf den Buchseiten markieren können. Damit können Bilder mit anderen Objekten verknüpft werden. Der Leser soll also beliebig viele Informationen und Annotationen zu den Bildern hinzufügen können.

- **Nano Publications:**

Der Viewer soll die Definition einer *Nano Publication* durch ein Subjekt, Prädikat und Objekt [Mons, 09] mit den Elementen des Buches ermöglichen sowie dessen Visualisierung mittels RDF-Triple [Müller & Maurer, 11].

4.3.2 Communication Layer

Dieser Layer hat im Wesentlichen das Ziel, das Lesen der Bücher in virtuellen Lesegruppen zu ermöglichen sowie das vernetzende Lesen umfassend zu unterstützen. Dies soll vor allem durch die maßgebliche Mitwirkung der Sozialen Netzwerken gewährleistet werden. Der Communication Layer soll folgende grundlegende Funktionalitäten unterstützen:

- **Inhalt teilen:**

Der Leser soll ein ganzes Buch, eine Seite oder einen Teil einer Buchseite mit anderen Lesern bzw. über bestehende Soziale Netzwerke teilen können. Des Weiteren soll die Möglichkeit gegeben sein, Inhalte des Buches mit den Mitgliedern von virtuellen Gruppen auszutauschen bzw. sie gemeinsam lesen und bearbeiten zu können.

- **Social Tagging:**

Der Leser soll Teile eines Buches klassifizieren, kommentieren und Kategorien zuordnen können [Folksonomy, 09].

- **Diskussionen:**

In dem Viewer soll ein Diskussionsforum und eine Chat-Funktion enthalten sein. Auf diese Weise soll das Buch selbst als Schnittstelle für soziale Interaktionen dienen.

- **Lesechronik:**

Der Viewer soll sowohl die persönliche Lesechronik als auch jene von Gruppen sowie Listen der Anmerkungen und Links anbieten können.

- **Suchagent:**

Der Viewer soll es dem Leser ermöglichen, die gespeicherten Suchabfragen jederzeit mit einem einfachen Klick und ohne alle Suchbegriffe wieder eingeben zu müssen, erneut auszuführen. Des Weiteren wird der Leser von dem Suchagenten benachrichtigt, sobald neue Ergebnisse zu den angeforderten Suchbegriffen erscheinen.

4.4 Kollaborative OCR-Korrektur / Crowdsourcing

Wie in Abschnitt 2.2.4 erwähnt, werden während des Scanvorganges und auch danach einige Maßnahmen gesetzt, um möglichst korrekte OCR-Ausgaben zu erhalten. Die Ergebnisse

hängen aber stark von den Vorlagen ab und sind bei historischen Materialien trotz aller Vorsichtsmaßnahmen fehlerhaft. Derzeit liegt die Fehlerquote nach der automatisch durchgeführten OCR-Korrektur zwischen 10 und 15%.

Das Ziel wäre also, diese automatisch erkannten Texte manuell zu korrigieren. Dies lässt sich heutzutage im Rahmen einer „sozialen Engagement“-Initiative verwirklichen, was als *Crowdsourcing* bezeichnet wird.

"Crowdsourcing ist eine interaktive Form der Leistungserbringung, die kollaborativ oder wettbewerbsorientiert organisiert ist und eine große Anzahl extrinsisch oder intrinsisch motivierter Akteure unterschiedlichen Wissensstands unter Verwendung moderner IuK-Systeme auf Basis des Web 2.0 einbezieht. Leistungsobjekt sind Produkte oder Dienstleistungen unterschiedlichen Innovationsgrades, welche durch das Netzwerk der Partizipierenden reaktiv aufgrund externer Anstöße oder proaktiv durch selbsttätiges Identifizieren von Bedarfslücken bzw. Opportunitäten entwickelt werden." [Martin, Lessmann & Voß, 08]

Crowdsourcing nutzt Techniken sozialen Engagements um ein gemeinsames, großes und signifikantes Ziel durch Zusammenarbeit zu erreichen. In der Regel ist Crowdsourcing gegenüber sozialem Engagement mit einem höheren Maß an Arbeit, Zeit und intellektuellem Input verbunden. Einige interessante Beispiele für Crowdsourcing, die von Web 2.0 Technologien verwendet wurden, sind in [Sharky, 09] beschrieben.

Die Integration dieses Konzeptes, das als neue Form der Arbeitsteilung gilt, kann folgende Vorteile mit sich bringen [Holley, 2009a]:

- Die Verbesserung der Qualität von OCR-Ausgaben und somit die Möglichkeit einer genaueren Volltextsuche.
- Die Erreichung von Zielen innerhalb einer kürzeren Zeitspanne und unter einem geringeren Einsatz von Finanz- und Personalressourcen.
- Die Entstehung zahlreicher neuer virtueller Gemeinschaften und Benutzergruppen.
- Eine effektive Verwendung von dem Interesse, Wissen und Know-how, das in einer Gemeinschaft vorhanden ist.
- Die Möglichkeit, ein Leser-Feedback zu erhalten.

- Da es sich nicht um ein gewinnorientiertes Projekt handelt, haben die Leser die Gewissheit, dass sie nicht ausgenutzt werden. Somit können sie Vertrauen und Loyalität gegenüber dem Austria-Forum entwickeln.
- Die aktive Mitwirkung der Leser wird deren Eigenverantwortung und Wertschätzung für Sammlungen des Kulturerbes fördern.

Im August 2008 hat die Australische Nationalbibliothek im Rahmen eines Pilot-Projektes ein Korrektur-Modul veröffentlicht, das es Lesern ermöglicht, Fehler in den OCR-Textdateien neben dem Scanbild direkt Zeile per Zeile auszubessern. Im November 2009 haben mehr als 6.000 Leser an der Korrektur teilgenommen und 7 Million Zeilen in 318.000 Artikeln korrigiert [Holley, 2009b].

[Holley, 2009a] beschreibt in der Tabelle allgemeine Tipps für die Umsetzung von Crowdsourcing.

Tabelle 1 Crowdsourcing checklist

The Thing	The System	The People	The Content
👍 Clear goal	👍 Easy and Fun	👍 Acknowledge	👍 Interesting
👍 Big challenge	👍 Reliable and quick	👍 Reward	👍 New
👍 Progress	👍 Intuitive	👍 Team support	👍 Lots
👍 Results	👍 Options	👍 Trust	👍 History/Science

Crowdsourcing soll also dem AF dabei helfen, die Fehler, welche in der OCR-Erkennung enthalten sind, durch die Beteiligung von Lesern auszubessern. Dafür soll in dem Viewer ein Korrektur-Modul umgesetzt werden, das die Scandatei neben der OCR-Erkennung anzeigt und die dafür benötigten Tools in intuitiver Form dem Leser zur Verfügung stellt. Die durch den Einsatz dieses Moduls gesammelten Benutzereingaben werden daraufhin bei der in Abschnitt 2.2.4 erwähnten Lösung verwendet. Dabei wird die OCR-Engine von einem selbstlernenden System durch die neu erkannten und in das Wörterbuch hinzugefügten Wörter trainiert.

4.5 Metadaten

Zu den deskriptiven und formellen Metadaten, welche Informationen wie Autor, Titel, Erscheinungsdatum usw. (siehe Abschnitt 2.2.3) enthalten, sollen strukturelle Metadaten hinzutreten. Sie sollen die verschiedenen Teile des Digitalisates beschreiben und somit die Struktur des Werkes abbilden. Zu den strukturellen Komponenten gehören das Inhaltsverzeichnis, die Seitennummern bzw. die Seitenzahl, die einzelnen Kapitel, Bilder und Tabellenanhänge. Diese Metadaten dienen der Beschreibung des Dokumentenaufbaus und haben grundsätzlich die Aufgabe, eine einfache Navigation und die Volltextsuche innerhalb des Digitalisates zu ermöglichen. Der Leser muss sich nicht mühsam Seite für Seite durch das Buch durcharbeiten um zu den gewünschten Informationen zu gelangen. Darüber hinaus soll durch diese Art von Metadaten eine starke Verknüpfung zwischen den Buchinhalten und den Beiträgen des Austria Forums und auch anderen AF-Objekten ermöglicht werden.

Die Vergabe der Metadaten kann allerdings im Falle historischer Bücher und Handschriften sehr aufwendig sein. Da sie sich hinsichtlich ihres Aufbaus erheblich voneinander unterscheiden können, bedarf beinahe jedes einzelne Werk eines individuellen Lösungsansatzes.

Darüber hinaus soll der Viewer die von den Lesern eingetragenen Informationen klassifizieren und zu den bereits vorhandenen Metadaten hinzufügen.

Um eine effiziente Verwaltung und somit Nutzung der Metadaten gewährleisten zu können, soll sich der Erwerb der Metadaten in dem Web Books Viewer wie folgt gestalten:

4.5.1 Metadaten anhängen

Auf den einzelnen Buchseiten sollen Metadaten angehängt werden können, damit eine automatische Verlinkung sowohl innerhalb des Buches als auch von den Büchern zu den anderen Objekten im AF und umgekehrt möglich ist. Sie sollen z.B. im unteren Bereich der Seite angezeigt und als abstrakte Ressourcen mittels URI identifiziert werden. Dies dient der Verknüpfung der Informationen aus den Büchern mit bereits zu diesem Thema vorhandenen Beiträgen im AF und lässt somit ein Wissensnetz entstehen.

Dabei handelt es sich einerseits um Stichwörter, die in den Buchseiten enthalten sind und andererseits um Kommentare, Annotationen und *Tags*, die von Lesern erfasst und hinzugefügt

wurden. Damit besteht die Möglichkeit *Tagclouds* [Rivadeneira, 07] anzubieten, welche den Inhalt der Bücher mit den Inhalten des AF verbindet und zusammenstellt. Auf diese Weise werden dem Leser verschiedene Themen und Schwerpunkte durch eine visuelle Darstellung einer ausgewählten Reihe von *Tags* angeboten. Mit einem Klick auf den gewünschten *Tag* werden dem Leser als Suchergebnis die jeweiligen Beitragseiten des AF sowie Buchseiten geliefert, auf denen das Stichwort identifiziert wurde.

Die von den Lesern verfassten Kommentare sind jedoch nicht immer als *Tags* zu bewerten und zu verwenden. Es handelt sich in diesem Fall um Kommentare, welche Informationen zu einem bestimmten Thema bzw. Schwerpunkt ergänzen. Sie sollen im Rahmen eines Forschungsmechanismus von anderen Lesern bzw. vom AF-Team geprüft und falls sie sich als nützlich erweisen, für die anderen Leser veröffentlicht bzw. als Metadaten angehängt werden.

4.5.2 Buchinhalt indizieren

Um die Navigation innerhalb der Digitalisate besser zu gestalten, sollen Metadaten über den Inhalt und die Struktur des Dokumentes vergeben werden. Dabei sollen Datenindexe in der Form eines elektronischen Inhaltsverzeichnisses (alphabetisches, Namens-, Titel-, Stichwortverzeichnis) erstellt werden. Diese stellen eine Auflistung der Datensatzgliederung in den Digitalisaten mit den jeweiligen Seitenzahlen dar und bestehen aus Verweisen auf die einzelnen Seiten. Sie sollen dem Inhaltsverzeichnis eines analogen bzw. gedruckten Buches ähneln mit einer zusätzlichen Sortierung nach allen möglichen Suchbegriffen, Schlagwörtern, Stichwörtern oder Namen in einer verlinkten Liste. Das dient einerseits dazu, einen schnellen Zugriff auf den Inhalt der einzelnen Buchseiten zu ermöglichen und andererseits einen Überblick über die Gliederung des gesamten Werkes bereitzustellen. Vor allem dann, wenn keine präzisen Suchbegriffe vorhanden sind bzw. die Suchanfrage aufgrund eines Schreibfehlers ergebnislos verlaufen ist, kann der Leser den Index nach relevanten Informationen durchsuchen.

Diese Metadaten sollen von dem Suchindex indiziert werden und im Sinne einer erfolgreichen Wissensvernetzung auch vom AF-Suchfeld aus abrufbar sein.

4.5.3 Akquisition der Metadaten

Weil die Volltextsuche in den Bücher oft zu sehr vielen Ergebnissen führt, ist es notwendig, dass diese nach ihrer Relevanz gereiht werden. Dies wird durch die Extrahierung relevanter Metadaten aus den einzelnen Buchseiten möglich.

Aufgrund der zunehmenden Menge an digitalen Ressourcen kann die Extrahierung dieser Metadaten nicht manuell durchgeführt werden, sondern muss mittels eines automatisierten Vorganges möglich sein. Dies erfordert effiziente Algorithmen um die Ressourcen möglichst gut zu analysieren und daraus „Content“ bzw. inhaltliche Metadaten zu vergeben [Wendler, 99].

Abgesehen von den Fehlern bei der OCR-Erkennung und deren Folgen für die Qualität der erkannten Texte, stellt sich die automatische Extrahierung der Metadaten aus den Digitalisaten bzw. deren OCR's als ein besonders schwieriges und kompliziertes Unterfangen dar. Grund dafür ist, dass die erfassten Materialien sehr unterschiedlich sind, was keinen einheitlichen Lösungsansatz bzw. Algorithmus erlaubt. Daher müssen Buchbestände einzeln betrachtet werden um für die automatische Extrahierung der Metadaten bestmöglich angepasste Lösungsansätze zu entwerfen.

Wie sich in [Korica-Pehserl & Maurer, 13] herausgestellt hat, werden angesichts der verschiedenen Bücherarten drei Typen der Metadatenextrahierung unterschieden:

- **Metadatenextraktion aus Verzeichnissen**

Aus den verschiedenen Auflistungen der Inhalte wie Inhaltsverzeichnis, Index, Abbildungsverzeichnis und/oder Sachregister sollen Metadaten gewonnen werden. Diese Metadaten dienen dazu, die Navigation innerhalb der Bücher zu verbessern. Des Weiteren stellen sie bei den Materialien, deren OCR-Erkennung eine schlechte Qualität aufweist, die einzige Suchoption dar. Wie in Abschnitt 2.2.4 erwähnt, lassen sich die OCR-Ausgaben der Altenbestände nicht so einfach bearbeiten um eine Extraktion der Metadaten durchzuführen. Deshalb müssen in diesem Fall die Ergebnisse zusätzlich manuell überprüft werden.

- **Metadatenextraktion aus Lexika**

In Lexika sind die Einträge alphabetisch geordnet bzw. aufgelistet. Abhängig von der Gestalt und dem Thema des Lexikons werden alle oder nur einige relevante Begriffe extrahiert.

Da in diesem Fall nur die Form und das Auftreten der Begriffe pro einzelner Seite betrachtet und analysiert wird, kann diese Art der Extrahierung relativ einfach durchgeführt werden. Dafür wurden bereits Algorithmen entworfen und in [Korica-Pehserl & Maurer, 13] beschrieben.

Die gewonnenen Metadaten sollen auf der jeweiligen Buchseite angehängt werden (siehe 4.5.1).

- **Metadatenextraktion aus anderen Büchern**

Es handelt sich in diesem Fall um Bücher, die weder über eine alphabetische Sortierung der Inhalte noch über eine andere Inhaltsauflistung verfügen. Hier wird die Extrahierung keine leichte Aufgabe sein, da jedes Buch einzeln angeschaut werden muss um eine angepasste Extraktionstrategie zu entwickeln, die relevante Metadaten aus dem Inhalt ableitet.

In der Regel bestehen solche Metadaten aus Namen von Plätzen, Personen, Organisationen usw. In diesem Zusammenhang eignen sich am besten *Named Entity Recognition* (NER)-Ansätze. In [Korica-Pehserl & Maurer, 13] wurde NER [Stanford NER, 14], [Finkel et al. 05] verwendet, weil es bereits eine Implementierung für die deutsche Sprache in [Faruqui & Pado, 10] gibt.

Eine alternative Lösung bietet der in [Miner et al. 12] und [Srivastava & Sahami, 09] beschriebene *Text Data Mining*-Ansatz. Dabei wird eine Metadatenextrahierung nach relevanten Suchbegriffen aus den AF-Beitragsseiten als *classifier* durchgeführt. Diese Vorgehensweise kann viele Informationen bzw. Ergebnisse liefern, die sich mit Hilfe überlegter heuristischer Methoden wie in [Grossman & Frieder, 04] bewerten lassen.

Ein Beispiel dafür, wie diese Metadaten zu präzisen Suchergebnisse führen, ist die bereits im AF angebotene Metadaten-Suchfunktion. Abbildung 20 zeigt, wie eine Suche aussehen kann, wenn man sich beispielsweise nicht an den Namen der gesuchten Person erinnern kann, aber sehr wohl an einige Details.

Suche in Biographien:

UND	Wien	Geburtsort	
UND		Geburtsland	-
UND		Geburtsjahr	- Jahr oder zwei Jahre mit - dazwischen eingeben
UND	Physik	Arbeitsort	-
UND		Arbeitsgebiet	-
UND		Todesort	-
UND	Italien	Todesland	-
UND		Todesjahr	- + Jahr oder zwei Jahre mit - dazwischen eingeben

Zeige Suchergebnisse

Suchergebnisse für 'Geburtsort:Wien AND Arbeitsgebiete:Physik AND Todesland:Italien'

Seite	Relevanz
Boltzmann, Ludwig (Biographien)	100

Abbildung 20 Metadaten-Suchfunktion im Austria Forum [Maurer, & Mueller, 13]

4.6 Information Integration

Angesichts der enormen Mengen an Informationen, mit denen wir tagtäglich konfrontiert sind, stellt sich *Information Integration* [Mehmood & Maurer, 13] als eine der größten Herausforderungen dieses Jahrhunderts dar.

Wissen verdoppelt sich in vielen Bereichen (Biologie, Medizin, Informatik, etc.) innerhalb von sechs Jahren, während sich Daten alle acht Monate verdoppeln [Wurzinger, 10]. Das bedeutet, dass das gleiche Wissen in kurzen Zeitabständen mit nur geringen Veränderungen veröffentlicht wird.

Es ist somit erstrebenswert, Informationen aus verschiedenen Quellen mit so wenig Verlusten, Überschneidungen und Redundanzen wie möglich, automatisch zusammenzulegen. Dies könnte einerseits eine völlig neue Art des Umgangs mit Daten in dem Web Books Viewer und andererseits eine intelligente Informationssuche ermöglichen.

In diesem Zusammenhang sollen die aus den verfügbaren Büchern extrahierten inhaltlichen und strukturellen Metadaten mit anderen relevanten Informationen aus den AF-Beitragseiten und anderen AF-Objekten verbunden werden. Daraus soll eine zuverlässige Sammlung aller relevanten Informationen entstehen.

Um dieses Vorhaben zu verwirklichen, müssen neue Wege für die oben erwähnten Anforderungen an die Metadatenextrahierung gefunden sowie intelligente Algorithmen, die

eine einfache und möglichst verlustfreie Zusammenlegung der Daten gewährleisten, entworfen werden.

4.6.1 Konsolidierung der Informationen

Die aus den Büchern gewonnenen Metadaten sollen konsistent und redundanzfrei sein. Konsistent im Sinne von widerspruchsfrei und eindeutig und redundanzfrei in der Hinsicht, dass Datensätze nicht mehrfach vorkommen dürfen bzw. mittels *Heuristics* nach ihrer Relevanz geordnet werden sollen.

Diese Metadaten sollen mit dem im AF vorhandenen Wissen verknüpft und die Daten der beiden Parteien zusammengeführt werden. Datensätzen aus Büchern und aus Beitrags- und Wikiseiten des AF, die sich auf dasselbe Thema, etwa dieselbe Person oder denselben Ort, beziehen, werden identifiziert und daraufhin deren Inhalte konsolidiert. Somit sollen nach einer Suchanfrage die relevanten Ergebnisse in dem Buch geliefert werden sowie Hinweise auf zusammenhängende Informationen in Beitragsseiten, Bildern oder anderen Objekte des AF angeführt werden.

In [Afzal, 09], [Afzal, 10] und [Zaka, 08] hat eine Gruppe von Forschern versucht, wissenschaftliche Arbeiten, welche sich mit ähnlichen Themen beschäftigen, zusammenzuführen. Zusätzlich sind andere interessante Ansätze in Richtung Konsolidierung von Informationen im Rahmen der Plagiaterkennung in Arbeiten von [Maurer, 07a] und [Maurer, 07b] sowie von [Kappe, 06] angewendet worden.

4.6.2 Transclusion

Der Viewer soll einen Transclusion-Mechanismus anbieten, mit dem Teile aus anderen Büchern und/oder anderen AF-Objekten „nahtlos“ in ein Buch eingebunden werden können.

Unter Transclusion versteht man die Fähigkeit, Teile bzw. Abschnitte oder ganze Dokumente in ein bestehendes Dokument einzubinden. Die Transclusion ähnelt einem Unterprogramm bzw. einem Prozeduraufruf in einer Programmiersprache mit einer automatischen Rückkehr zu dem Ausgangspunkt. Die bei diesem Mechanismus verwendete Verlinkung ähnelt dem „goto“ in den Programmiersprachen [Maurer & Mehmood, 14].

Diese Funktion ist im AF auf der Basis der im Rahmen von Hypertext "Xanadu" von Ted Nelson [Neldon, 87] vor langer Zeit vorgeschlagenen Idee bereits implementiert. Dabei wird eine Seite A mit dieser Funktionalität in eine Seite B eingefügt, statt den Inhalt von A nach B zu kopieren. Dies hat den Vorteil, dass Modifikationen der Seite A automatisch in die Seite B übernommen werden. Werden beispielsweise die Lebensdaten eines Erfinders in dessen Biographie korrigiert, erfolgt die Korrektur automatisch auch im AEIOU-Lexikon, falls die Biographie dorthin eingefügt wurde.

Weiters besteht die Möglichkeit nur einen Teil einer anderen Seite einzufügen. Durch zusätzlich spezielle Marker „\$\$\$x“, die in einer Seite gesetzt werden können. Man kann damit einen Teil einer anderen Seite, begrenzt durch zwei Marker, einfügen. Das x hinter \$\$\$ ist eine Zahl (Index), welche dem Plugin als Parameter angegeben wird.

```

    {{InsertPage page='Seite' start='Start-Index' end='Ende-Index'}}
  
```

Wird ein Parameter nicht angeben, so wird die Seite von Beginn an oder bis zum Ende eingefügt.

Abbildung 21 aus [Kolbitsch & Maurer, 05] zeigt wie diese Funktionalität verwendet wurde.

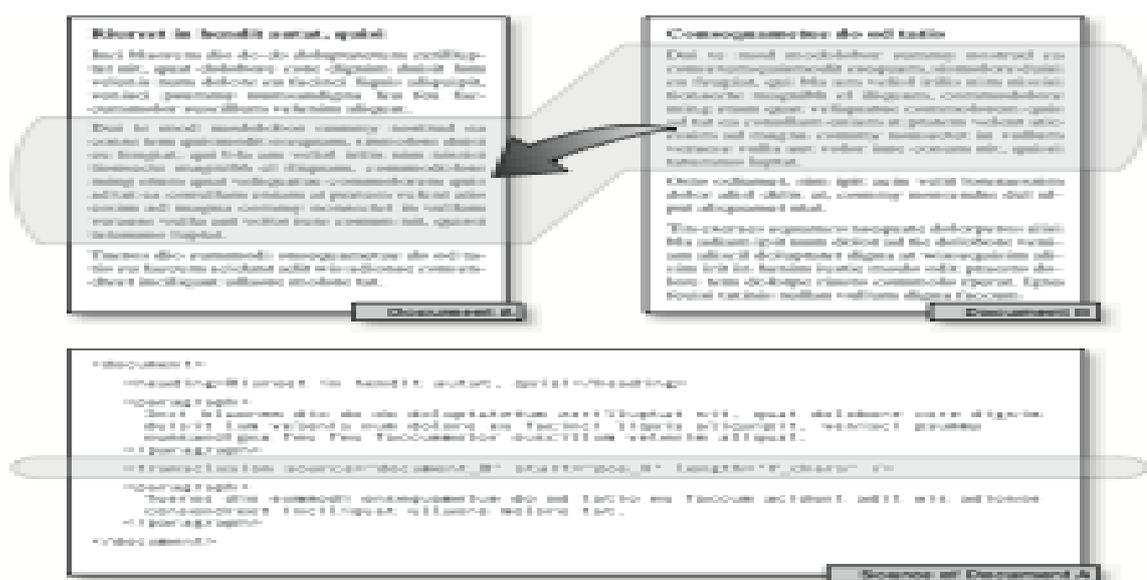


Abbildung 21 Transclusion

In dem Viewer soll es ähnlich wie in der AF-Implementierung die Möglichkeit geben, einen Anfang und ein Ende des Markierungsbereiches zu definieren, um diesen Abschnitt in eine andere Seite einzufügen bzw. zu verlinken. Die Verweise sollen bidirektional sein. Wenn man eine Buchseite in dem Web Books Viewer liest, soll man auch sehen, welche anderen Buchseiten und AF-Beitragsseiten auf diese Seite verweisen.

5 Konzeption

5.1 Bestimmung der Implementierungsumgebung

Basierend auf den im vorherigen Kapitel präsentierten Anforderungen wurde eine Auswahl an Technologien getroffen, die bei der technischen Umsetzung und Implementierung des Web Books Viewers verwendet wird. Wie im Abschnitt 4.2 erwähnt, soll der Viewer plattformunabhängig und webbasierend sein. Deshalb werden für die Entwicklung des Systems aktuelle Webtechnologien verwendet.

Auf der Serverseite wird ein RESTful (REpresentational State Transfer) Webservice [Richardson & Ruby, 07] zur Verfügung gestellt. Dies stellt die APIs bereit, die die Funktionalitäten des Viewers gewährleisten. (siehe Abbildung 22)

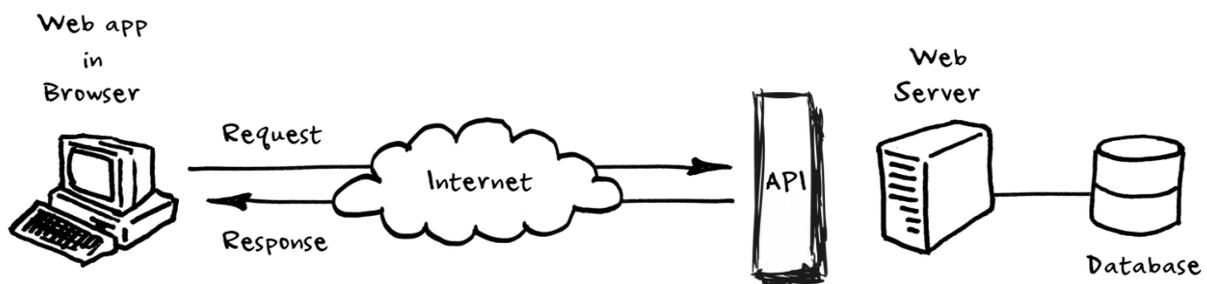


Abbildung 22 RESTful Web Server [Rasberry, 13]

Hier eignet sich der Einsatz eines RESTfull Web Services, weil der Zugriff auf die digitalen Ressourcen über ein einheitliches API standardisiert werden kann. Außerdem können mehrere Clients gleichzeitig die Serverdienste problemlos in Anspruch nehmen.

Der Viewer wird durch das Kommunikationsprotokoll HTTP über die RESTfull-Schnittstellen mit dem Server kommunizieren.

Wie im Buch [Obcena pro, 10] beschrieben, werden folgende HTTP-Methoden dem Client angeboten:

- **OPTIONS:** prüft die Methoden, die auf einer Ressource zur Verfügung stehen
- **POST:** um eine neue Ressource einzufügen

- **GET:** um eine Ressource anzufordern. Diese kann vom Client beliebig oft abgeschickt werden, weil die GET-Requests nur lesend auf den Server zugreifen dürfen.
- **PUT:** um eine bestimmte Ressource zu ändern oder zu ersetzen, wenn sie bereits am Server existiert
- **DELETE:** um eine Ressource auf dem Server zu löschen

Ein nutzbares Merkmal dieses Architekturstils ist die via URIs (Uniform Resource Identifier) bzw. URLs (Uniform Resource Locator) eindeutige Identifizierung der Ressourcen. Daher wird REST als „Resource-Oriented Architecture“ bezeichnet [Deiss et al. 06].

Clientseitig wurde MooTools “My Object Oriented JavaScript Tools” gewählt, die laut [Obcena, 10] als eine JavaScript Library bzw. als ein Framework zu betrachten ist, die JavaScript wie keine andere ergänzt.

Wenn man JavaScript verwendet, muss man üblicherweise alle möglichen Lücken und Eventualitäten abdecken, was jedoch die Größe des Quellcodes beeinflusst. Ein typisches Beispiel dafür ist die Browserkompatibilität. Doch durch die Verwendung eines Javascript Frameworks, das sowohl solche Fälle abdeckt als auch viele andere Vereinfachungen, wie beispielsweise eine Sammlung hilfreicher Funktionen, von Haus aus anbietet, erspart man sich unnötigen Quellcode.

Einer der großen Vorzüge von MooTools ist, dass es sehr viel Wert auf die objektorientierte Programmierung und vor allem auf die Klassenvererbung legt. Dies bietet dem Programmierer die Möglichkeit, das Framework einfach zu erweitern sowie benötigte Anpassungen durchzuführen oder wiederkehrende Abläufe und flexible Funktionen festzulegen.

Man kann von den Stärken von MooTools am besten profitieren, indem man Funktionen und Routineabläufe in Form von Klassen schreibt.

Mittlerweile wird MooTools von vielen Webseiten verwendet wie zum Beispiel von den folgenden:

- Formula1.com
- The Museum of Modern Art
- Stanford University
- Miss Universe
- Vimeo
- Ferrari

- Joomla!
- Chrysler

Um den Code möglichst gut zu strukturieren und zu organisieren, fiel die Entscheidung, den Quellcode entsprechend dem MVC Muster (engl. Model, View, Controller) [Krasner et. All 88] zu schreiben. Weitere Informationen zu dieser Art der Implementierung befinden sich in Kapitel 6.

5.2 Entwurf der Benutzeroberfläche

Das „Graphical User Interface“ (GUI) des Viewers wurde größtenteils von dem bereits existierenden auf Flash basierten Viewer des Austria Forums¹⁶ inspiriert. Die Abbildung 23 stellt eine GUI-Skizze in der Anfangsphase dar. Sie zeigt, dass der Viewer zwei Ansichten anbieten soll: Buch- und Explore-Ansicht. Während die erste Ansicht lediglich der Veranschaulichung des Buches dient, kann der User im Explore Mode das Buch annotieren und Hyperlinks erstellen. Auch die Umschaltung zwischen den verschiedenen Modi wie z.B. Single-/Thumbnailmodus ist in diesem Entwurf vorgesehen.

¹⁶ <http://austria-forum.org/ebook/wbin/ambrosius.html>

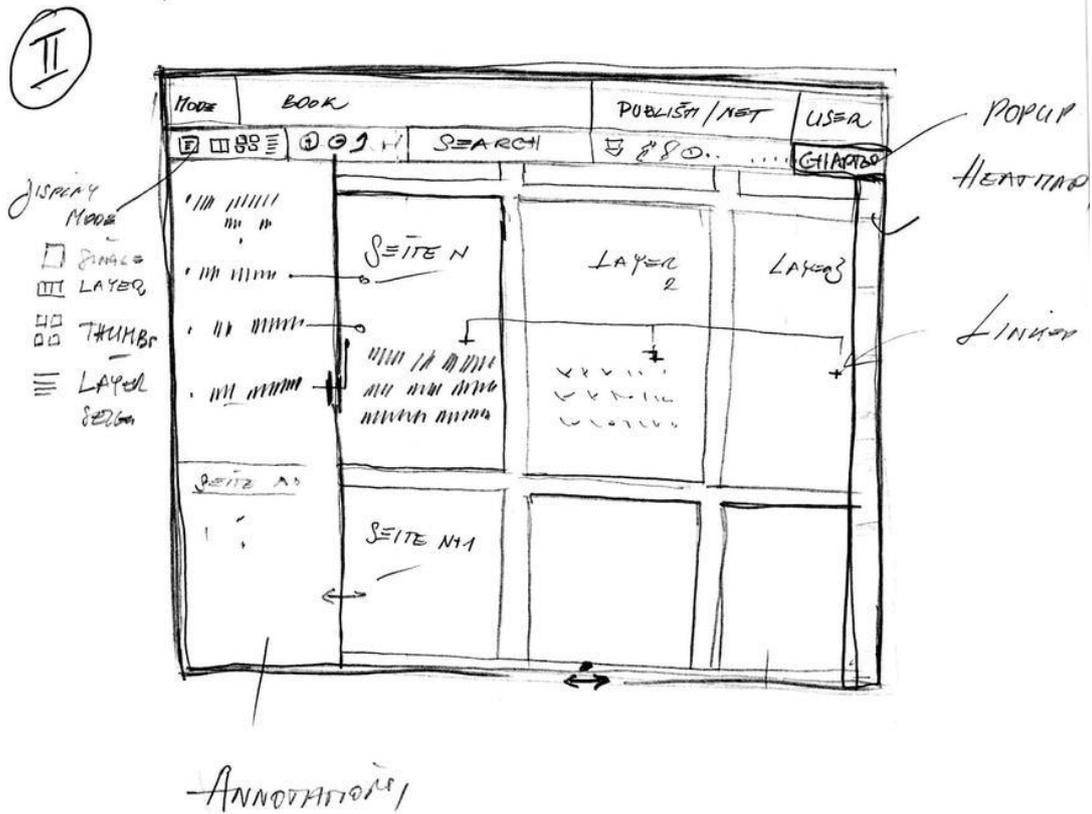
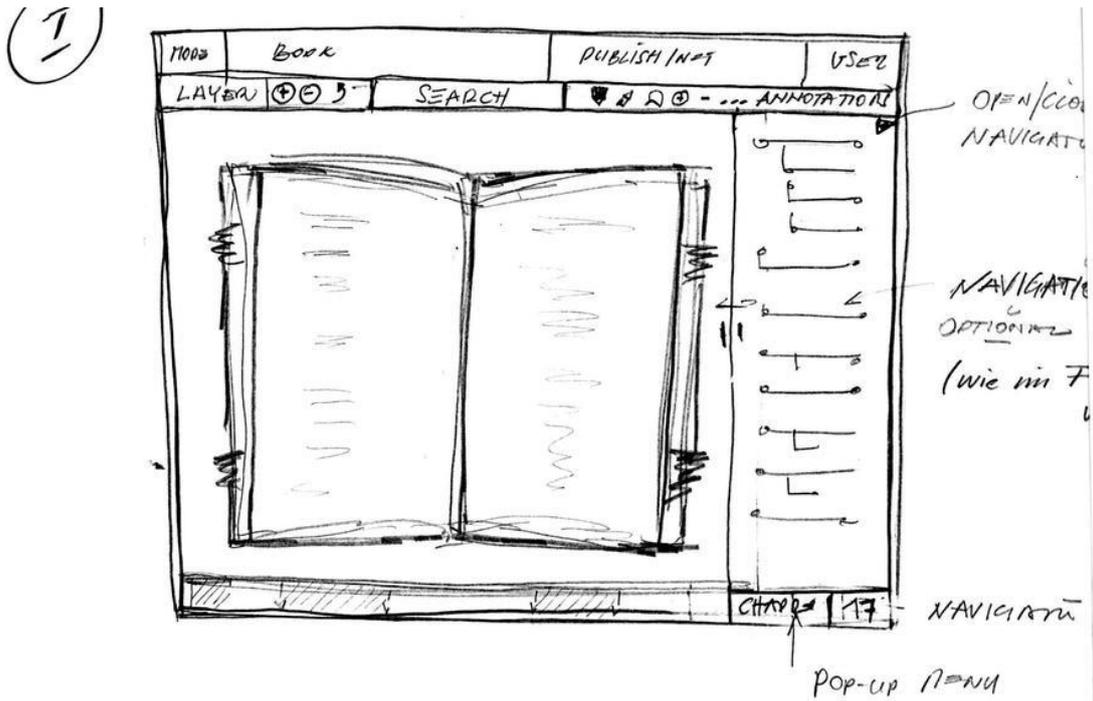


Abbildung 23 Web Books Viewer GUI-Entwurf, erstellt von Dr. Müller

5.3 Architektur

Wie oben schon erwähnt, wird im Rahmen dieser Arbeit ein Viewer unter Verwendung des MooTools Javascript Frameworks umgesetzt.

Unter der Architektur versteht man einerseits die Aufteilung der Viewer Module in mehrere Programmteile sowie deren Abhängigkeit und Kommunikation untereinander und andererseits die Serverschnittstellen bzw. das Zugreifen auf die benötigten Ressourcen.

Diese Architektur ist durch die im vorherigen Abschnitt angeführten Prinzipien und Organisationsformen charakterisiert.

5.3.1 Serverschnittstellen (APIs)

Es handelt sich dabei um jene Schnittstellen, die sowohl für die Kommunikation zwischen Client und Server als auch für den Zugriff auf die digitalisierten Ressourcen zuständig sind. Diese wurden speziell für die Fertigung dieses Viewers zur Verfügung gestellt.

Die folgenden Abschnitte sind ein Bestandteil der Web Books API Version 2 und zeigen die verfügbaren Dienste. Die Rückgabewerte erfolgen als JSON-Response.

Im Anschluss findet eine kurze Beschreibung der angebotenen Server APIs mittels Aufrufbeispielen statt.

1.1.1.1 API Library Server

Diese API ist grundsätzlich für die Liste aller verfügbaren Bücher sowie deren Metadaten und Beschreibungen zuständig.

- **Liste aller Bücher**

Liefert ein Array mit allen Büchern, die in der Library enthalten sind.

Route:

GET austria-forum.org/wbv2/api/lib/book

Beispiel:

<http://austria-forum.org/wbv2/api/lib/book>

```
1 [
2   {
3     "title": "100 Jahre Anglistik an der Universität Graz",
4     "subTitle": "",
5     "shortTitle": null,
6     "band": "27",
7     "language": "DE",
8     "location": "Graz",
9     "height": 26.05,
10    "width": 17.99,
11    "date": "1993",
12    "identifier": "100jahreanglist27de1993kfu",
13    "coverUrl": "http://austria-forum.org/wbv2/api/file/100jahreanglist27de1993kfu
14    /cover",
15    "covers":
16    {
17      "eh": "http://austria-forum.org/wbv2/api/file/100jahreanglist27de1993kfu/eh0002
18      /Print/JPG",
19      "cover": "http://austria-forum.org/wbv2/api/file/100jahreanglist27de1993kfu
20      /cover",
21      "ev": "http://austria-forum.org/wbv2/api/file/100jahreanglist27de1993kfu/ev0001
22      /Print/JPG",
23      "br": "http://austria-forum.org/wbv2/api/file/100jahreanglist27de1993kfu/br00001
24      /Print/JPG"
25    },
26    "bookUrl": "http://austria-forum.org/wbv2/api/lib/book/100jahreanglist27de1993kfu",
27    "publisher":
28    {
29      "name": "Akademische Druck- u. Verlagsanstalt Graz"
30    },
31    "authors":
32    [
33      {
34        "name": "Alois Kernbauer"
35      },
36      {
37        "name": "Alwin Fill"
38      }
39    ]
40  }, ...
41 ]
```

- **Metadata eines Buches**

Um die Metadata eines bestimmten Buches anhand seiner “identifier” zu bekommen.

Route:

GET austria-forum.org/wbv2/api/lib/book/identifier

Beispiel:

<http://austria-forum.org/wbv2/api/lib/book/osterreichlexik02de1995kfu>

```

1 {
2   "identifizier": "osterreichlexik02de1995kfu",
3   "source":
4   {
5     "name": "KFU",
6     "description": "Universitätsbibliothek der Uni Graz"
7   },
8   "metadata":
9   {
10    "title": "Österreich Lexikon",
11    "subTitle": "Buchstabe M-Z",
12    "shortTitle": null,
13    "band": "2",
14    "language": "DE",
15    "location": "Wien",
16    "date": "1995",
17    "identifizier": "osterreichlexik02de1995kfu",
18    "publisher":
19    {
20      "name": "Verlagsgemeinschaft Österreich-Lexikon"
21    },
22    "authors":
23    [
24      {
25        "name": "Richard und Maria Bamberger"
26      }, ... ]],
27    "pages":
28    [
29      {
30        "label": "Einband vorne",
31        "reference": "00000001",
32        "width": 0,
33        "height": 0
34      }, ...
35
36    "specialPages":
37    [
38      ],
39    "layers":
40    [
41      {
42        "name": "Facsimile",
43        "formats" [ { "name": "JPG", ..." } ] },
44      {
45        "name": "Ocr",
46        "formats": [ { "name": "JPG", " " }, { "name": "PDF".., } ] ] },
47

```

- **Kategorienliste**

Liefert eine Liste aller vorhandenen Kategorien.

Route:

GET austria-forum.org/wbv2/api/lib/category

Beispiel:

<http://austria-forum.org/wbv2/api/lib/category>

```

1 [
2   {
3     "name": "Kronprinzenwerk",
4     "description": "alle deutschen und ungarischen Kronprinzenwerke"
5   },
6   {
7     "name": "Sachbuch",
8     "description": "allerlei Sachbücher"
9   },
10  ...
11 ]

```

Man kann ebenfalls eine Liste mit allen Büchern samt Metadaten einer bestimmten Kategorie abfragen.

Route:

GET austria-forum.org/wbv2/api/lib/category/identifizier

Beispiel:

<http://austria-forum.org/wbv2/api/lib/category/Handschriften>

```
1  [
2  {
3      "title": "Ms.1",
4      "subTitle": "Pergament 470 Bl.",
5      "shortTitle": null,
6      "band": null,
7      "language": "NONE",
8      "location": "St. Georgs-Ritterorden in Millstatt",
9      "height": 57,
10     "width": 41,
11     "date": "Um 1480",
12     "identifizier": "ms.0001.kfu",
13     ...
14 }
15 ]
```

- **Coverbild**

Um das Coverbild eines Buches zu erhalten.

Route:

GET austria-forum.org/wbv2/api/lib/book/identifizier/cover

Beispiel

<http://austria-forum.org/wbv2/api/lib/book/osterreichlexik02de1995kfu/cover>

1.1.1.2 API File Server

Dieses API ist für den Aufruf einzelner Buchseiten zuständig.

- **Seite Aufrufen**

Liefert eine bestimmte Seite zurück.

Route:

GET austria-forum.org/wbv2/api/file/identifizier/reference/layer/format

Beispiel

<http://austria-forum.org/wbv2/api/file/osterreichlexik02de1995kfu/00000722/Facsimile/JPG>

- **Skalierte Seite aufrufen**

Hier werden Seiten zusätzlich mit einem Skalierungsparameter aufgerufen, das als Faktor zur Größenangabe herangezogen wird.

Dieser Skalfaktor hat folgende Abstufungen bzw. Einschränkungen:

- scale \geq 1.0: max. Größe von 1000*1500 Pixel
- scale $>$ 0.75: 100% von scale=1
- scale $>$ 0.5: 75% von scale=1
- scale $>$ 0.25: 50% von scale=1
- scale \leq 0.25: max. Größe von ?*300 Pixel

Route:

GET austria-forum.org/wbv2/api/file/identifizier/reference/layer/format/scale

Beispiel

<http://austria-forum.org/wbv2/api/file/osterreichlexik02de1995kfu/00000722/Facsimile/JPG/0.25>

1.1.1.3 API Content Server

Dieses API ermöglicht es Operationen am Inhalt des Buches vorzunehmen.

- **Schlüsselwörter einer Buchseite**

Der Aufruf liefert einige wichtige Wörter, welche einen Bezug zum Thema bzw. der Kategorie des Buches aufweisen und deren Position auf einer bestimmten Buchseite.

Route:

GET austria-forum.org/wbv2/api/cont/text/identifizier/reference

Beispiel

<http://austria-forum.org/wbv2/api/cont/text/osterreichlexik02de1995kfu/00000015>

```

1 ▾ [
2 ▾ {
3     "word": "Malerei",
4     "bounds":
5     {
6         "left": 0.7486592,
7         "top": 0.02860179,
8         "width": 0.050475396,
9         "height": 0.0072526457
10    }
11 },
12 ▾ {
13     "word": "(P.",
14     "bounds":
15     {
16         "left": 0.051438324,
17         "top": 0.06312984,
18         "width": 0.016466815,
19         "height": 0.007508697
20    }
21 },
22 ▾ {
23     "word": "Fendi,",
24     "bounds":
25     {
26         "left": 0.07533833,
27         "top": 0.06312984,
28         "width": 0.041265093,
29         "height": 0.007508697
30    }
31 }, ...

```

- **Text eines Bereiches einer Buchseite**

Der Aufruf liefert den Text des ausgewählten Bereiches. Diese „Bounding-Box“ ist mit „left“, „top“, „width“ und „height“ definiert.

Route:

GET austria-forum.org/wbv2/api/cont/text/identifier/reference/left/top/width/height

Beispiel

[http://austria-](http://austria-forum.org/wbv2/api/cont/text/osterreichlexik02de1995kfu/00000029/0.1/0.1/0.3/0.2)

[forum.org/wbv2/api/cont/text/osterreichlexik02de1995kfu/00000029/0.1/0.1/0.3/0.2](http://austria-forum.org/wbv2/api/cont/text/osterreichlexik02de1995kfu/00000029/0.1/0.1/0.3/0.2)

```

1 ▾ {
2     "content": "hl. Sigmund auf dem Sigmundsberg (nach Kapellen und Brunnen an den...",
3     "bounds":
4     {
5         "left": 0.1,
6         "top": 0.1,
7         "width": 0.3,
8         "height": 0.2
9     }
10 }

```

- **Lesezeichenliste**

Liefert entweder eine allgemeine Lesezeichenliste, die eines bestimmten Benutzers oder einer bestimmten Gruppe.

Route:

GET [austria-forum.org/wbv2/api/cont/bookmark\[/identifizier/reference\]](http://austria-forum.org/wbv2/api/cont/bookmark[/identifizier/reference])

Wobei `[/identifizier/reference]` sind optional.

Beispiel

<http://austria-forum.org/wbv2/api/cont/bookmark/osterreichlexik02de1995kfu>

```
1  [
2  {
3      "identifizier": "osterreichlexik02de1995kfu",
4      "reference": "00000203",
5      "bookmark": "00001234",
6      "title": "interessante Stelle",
7      "user" : null,
8      "group" : "Admin"
9  },
10 {
11     "identifizier": "osterreichlexik02de1995kfu",
12     "reference": "00000202",
13     "bookmark": "00001231",
14     "title": null,
15     "user" : null,
16     "group" : null
17 } , ...
18 ]
```

- **Lesezeichen bearbeiten**

Um ein bzw. mehrere Lesezeichen auf einer Buchseite zu bearbeiten.

Route:

POST | PUT | DELETE [austria-forum.org/wbv2/api/cont/bookmark/identifizier/reference\[/bookmark\]](http://austria-forum.org/wbv2/api/cont/bookmark/identifizier/reference[/bookmark])

Wobei `[/bookmark]` ist optional.

Beispiel:

<http://austria-forum.org/wbv2/api/cont/bookmark/osterreichlexik02de1995kfu/00000015/00001234>

Payload / Response

```
1  {
2      "title": "interessante Stelle",
3      "user" : null,
4      "group" : "Admin"
5  }
```

- **Liste der Hyperlinks**

Liefert eine Liste der allgemein erstellten Hyperlinks, der eines bestimmten Benutzers oder einer bestimmten Gruppe.

Route:

GET [austria-forum.org/wbv2/api/cont/link\[/identifizier/reference\]](http://austria-forum.org/wbv2/api/cont/link[/identifizier/reference])

Wobei `[/identifizier/reference]` sind optional.

Beispiel

<http://austria-forum.org/wbv2/api/cont/link/osterreichlexik02de1995kfu>

```

1 [
2   {
3     "identifizier": "osterreichlexik02de1995kfu",
4     "reference": "00000203",
5     "link": "00001234",
6     "title": "interessante Stelle",
7     "user": null,
8     "group": "Admin",
9   },
10  {
11   "identifizier": "osterreichlexik02de1995kfu",
12   "reference": "00000202",
13   "link": "00001231",
14   "title": null,
15   "user": null,
16   "group": null
17  }, ...
18 ]

```

- **Hyperlinks bearbeiten**

Um die Hyperlinks auf einer Buchseite zu bearbeiten.

Route:

POST | PUT | DELETE [austria-forum.org/wbv2/api/cont/link/identifizier/reference\[/link\]](http://austria-forum.org/wbv2/api/cont/link/identifizier/reference[/link])

Wobei [/link] ist optional.

Beispiel

Payload / Response

```

1 {
2   "title": "interessante Stelle, Link zum AF",
3   "user": null,
4   "group": "Admin",
5   "link": "http://austria-forum.org/af/AEIOU/Graz",
6   "type": "austria-forum",
7   "bounds":
8     {
9       "left": 0.6626538,
10      "top": 0.16569853,
11      "width": 0.051397607,
12      "height": 0.0075087026
13    }
14 }

```

1.1.1.4 API Search Server

Dieses API ist für die Suche innerhalb eines Buches bzw. einer Seite zuständig.

- **Allgemeine Suche**

Sucht in allen Büchern nach dem Suchbegriff und liefert als Ergebnis die Seiten-Referenzen.

Route:

GET austria-forum.org/wbv2/api/search?query='suchbegriff'

Beispiel

<http://austria-forum.org/wbv2/api/search?query=malerei>

```
1 {
2   "neuzeitlichesha00de1934iicm":
3   [
4     "00000747"
5   ],
6   "osterreichlexik02de1995kfu":
7   [
8     "00000299",
9     "00000051",
10    "00000278", ...
11  ]
12 }
```

- **Suche in einem Buch**

Sucht in einem bestimmten Buch nach dem Suchbegriff und liefert als Ergebnis eine Liste von Seiten, in denen der Suchbegriff vorkommt und innerhalb dieser eine weitere Liste mit seinen Positionen auf der jeweiligen Buchseite.

Route:

GET austria-forum.org/wbv2/api/search/identifier?query=query

Beispiel

<http://austria-forum.org/wbv2/api/search/osterreichlexik02de1995kfu?query=malerei>

```
1 {
2   "00000307":
3   [
4     {
5       "word": "Malerei",
6       "bounds":
7       {
8         "left": 0.6626538,
9         "top": 0.16569853,
10        "width": 0.051397607,
11        "height": 0.0075087026
12      },
13      "context": "... <span class=\"search-highlight\">Malerei</span> ..."
14    }
15  ],
16   "00000606":
17   [
18     {
19       "word": "Malerei",
20       "bounds":
21       {
22         "left": 0.62726605,
23         "top": 0.073550604,
24         "width": 0.042446736,
25         "height": 0.0060069575
26      },
27      "context": "... <span class=\"search-highlight\">Malerei</span> ..."
28    }
29  ], ...
30 }
```

1.1.1.5 API Authentication Server

Dieses API ist für die Benutzerauthentifizierung zuständig. Mit diesem API können Benutzer und Gruppen verwaltet sowie Anmelde- und Abmeldeoperationen ausgeführt werden.

1.1.1.6 Viewer URLs

Dieses API verwaltet den Aufbau und die Formatierung der URLs des Web Books Viewers. Mit Hilfe dieses API's wird der Viewer die aufgerufene URLs auswerten und anzeigen können. Es sorgt auch dafür, dass die Adresszeile des Browsers immer dem aktuellen Zustand des Viewers entspricht.

5.3.2 Clientseite

Auf der Clientseite wird der Quellcode mit Hilfe von MooTools objektorientiert nach dem MVC Muster konzipiert und organisiert. Die Funktionalitäten werden dabei auf drei Ebenen aufgeteilt. Diese werden im Anschluss kurz beschrieben und deren Implementierung in Kapitel 6 erläutert.

Das MVC-Muster bietet architektonische Vorteile gegenüber dem Standard-JavaScript. Es ermöglicht das Schreiben eines besser organisierten und wartbaren Quellcodes. Die Abbildung 18 zeigt die drei Komponenten des MVC.

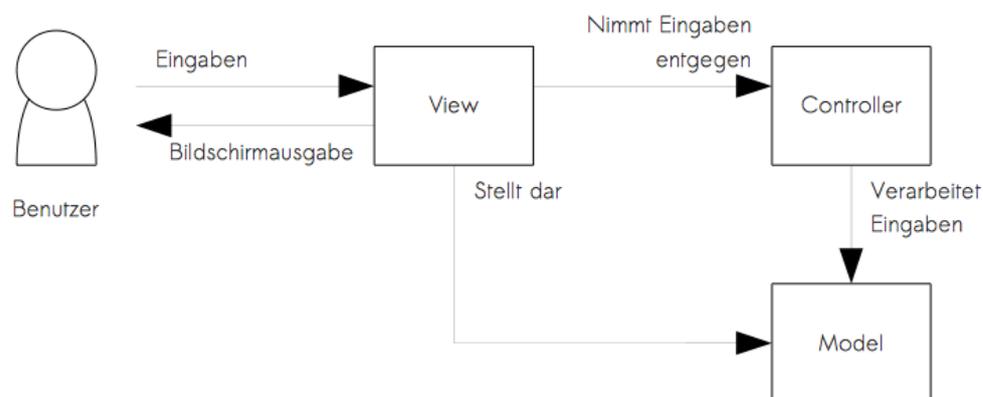


Abbildung 24 Web Books Viewer MVC Muster Schichten

1.1.1.7 Model

Das Model ist für die darzustellenden Daten, unabhängig von der View und von dem Controller, zuständig. Wenn eine Änderung im Model stattfindet, wird es typischerweise seine Beobachter darüber benachrichtigen, dass eine Änderung aufgetreten ist.

In unseren Fall kümmert sich das Model um die AJAX-Aufrufe bzw. um den Datenaustausch zwischen dem Viewer und dem Server. Das erfolgt mittels den APIs, die im Abschnitt 5.3.1 präsentiert wurden.

1.1.1.8 View

Die View hat im wesentlichen die Aufgabe, die benötigten Daten aus dem Model darzustellen. Sie nimmt außerdem die Benutzerinteraktionen entgegen jedoch ohne sie zu bewerten. Diese Präsentation wird mittels HTML, CSS und JavaScript realisiert.

Obwohl sie das Model und den Controller, dessen Daten sie präsentiert, kennt, kann sie die vom Benutzer übergebenen Daten nicht weiterverarbeiten.

In der Regel wird die View-Schicht von dem Model benachrichtigt, falls Datenänderungen stattfinden und kann somit die angezeigten Daten beim Abrufen aktualisieren.

Beim unserem Viewer handelt sich bei der View um einen generierten HTML Code, welcher zur Darstellung der Bücher sowie der Oberfläche dient.

1.1.1.9 Controller

Der Controller ist der Entscheidungsträger und der „Klebstoff“ zwischen Model und View. Der Controller aktualisiert die View, sobald sich die Daten im Model ändern. Zu jeder View existiert ein Controller. Er fügt „Event-Listeners“ zu der View hinzu und aktualisiert die entsprechenden Daten im Model, sobald der Benutzer die View manipuliert.

Im Web Books Viewer behandelt der Controller alle View-Ereignisse, um die entsprechenden Datenänderungen am Model durchzuführen.

5.4 Ajax Webanwendung

Weil es sich bei dem, in dieser Arbeit präsentierten Web Books Viewer, um eine Ajax Webanwendung handelt, werden wir dieses Konzept kurz erläutern. AJAX (Asynchronous JavaScript and XML) ist ein Konzept für die Erstellung von Webanwendungen, das eine asynchrone Kommunikation zwischen dem Server und dem Client möglich macht sowie eine

effizientere Datenübertragung gewährleistet. Diese wird anhand verschiedener Technologien ermöglicht. Bei dieser asynchronen Kommunikation zwischen Server- und Clientseite wird keine Unterbrechung seitens des Lesers zu spüren sein, wenn eine Benutzerinteraktion stattfindet.

Wie [Garett, 05] beschrieben hat, wird die Kommunikation zwischen dem Server und dem Client in dem, in dieser Arbeit umgesetzten Prototyp Viewer, wie in der Abbildung 25 angezeigt.

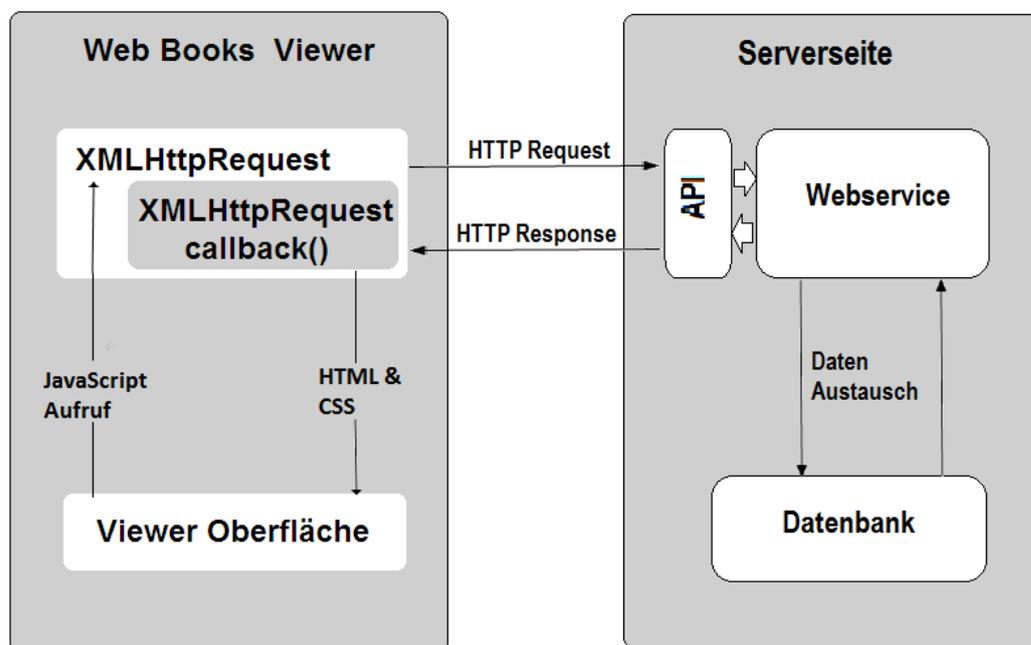


Abbildung 25 Web Books Viewer als Ajax Webanwendung

6 Implementierung

Bei dem vorliegenden Prototyp Viewer handelt es sich um eine auf Internettechnologien basierende Anwendung, die vom Austria-Forum für die Veröffentlichung von Büchern verwendet werden soll. Während der serverseitige Teil auf Java basiert, kommt im clientseitigen Teil HTML/CSS und JavaScript Skripten zum Einsatz, welche von gängigen Webbrowsern verwendet werden.

Der in dieser Arbeit umgesetzte Prototyp bietet vielfältige Grundfunktionalitäten, welche für die Veröffentlichung der Digitalisate notwendig sind. Manche Funktionalitäten wurden nur zum Teil und manche noch nicht implementiert. Dies ist die Folge zahlreicher Schwierigkeiten, die teilweise im 6.5 erläutert werden. Es besteht aber die Möglichkeit diesen Protyp zu erweitern und die offenen Funktionalitäten zu ergänzen.

6.1 Überblick

In dem vorangegangenen Kapitel wurden Ansätze für die Umsetzung des Viewers präsentiert und zusätzlich erläutert, wie der Quellcode strukturiert und organisiert wird. Wie bereits erwähnt, wird die in dieser Arbeit präsentierte Webanwendung in eine Server- und eine Clientseite geteilt. Die Kommunikation erfolgt über das standardisierte HTTP und durch die vom Webserver angebotenen APIs. Um die Trennung von Datenlieferung, Datendarstellung und Logik zu gewährleisten, haben wir uns für den Einsatz des MVC Musters entschieden. Darüber hinaus ermöglicht dieser die Erstellung eines „guten Codes“¹⁷. Diesem Ansatz ist zu entnehmen, dass die Funktionalitäten in drei große Bereiche geteilt werden. Ein Teil kümmert sich um die Datenbereitstellung, ein weiterer um die Darstellung der erhaltenen Daten und der letzte ist für die Logik zuständig. In Abschnitt 6.2 werden die Komponenten des MVC und ihre Verwendung im Web Books Viewer beschrieben.

Der Web Books Viewer wird als ein integrierter (interner) Dienst des Austria Forums angeboten, um die vorhandenen Digitalisate online zu veröffentlichen. Der Abruf des Viewers wird sowohl innerhalb der Austria-Forum Webseite möglich sein, als auch über Hyperlinks, die während einer Lesession geteilt wurden oder in den Beitragsseiten des Austria-Forums

¹⁷ <http://de.wikipedia.org/wiki/Codequalität>

eingebunden sind. Er basiert auf HTML und ist somit, im Gegensatz zu dem jetzigen von Austria-Forum verwendeten auf Flash basierten Book Reader (siehe Abschnitt 2.3), plattformunabhängig. Der Viewer ermöglicht das Lesen eines Buches in verschiedenen Lesemodi (Doppelseite, Einzelseite, Thumbnails) und unterstützt alle vorhandenen Buchlayers. Abhängig davon, in welchen Formaten das Buch zum Lesen angeboten wird, kann der Leser ebenso zwischen Layers wie z.B. OCR- und Scan-Format umschalten. Nicht alle Bücher im Austria-Forum verfügen derzeit über einen OCR Layer, aber es wird bereits daran gearbeitet.

6.2 Steuerungsaufteilung

Wie in Abschnitt 5.1 erwähnt, wurde für die Umsetzung des Viewers das JavaScript Framework MooTools verwendet. Das Kernkonzept dieses Frameworks bildet die Vererbung, die in anderen Programmiersprachen der objektorientierten Programmierung entspricht. Mit Hilfe der Vererbung kann die Stärke des Frameworks hinsichtlich der Wiederverwertung des Codes genutzt werden.

Da bei der Umsetzung nach dem MVC Muster (siehe Abschnitt 5.3.2) vorgegangen wird, werden die Komponenten des Viewers entsprechend den drei Ebenen des Musters aufgegliedert und der Klassendefinition des Viewers zugrunde gelegt.

Bei der Realisierung des Web Books Viewers wurden drei Hauptklassen implementiert: Loader, View und WebBooksReader „*Controller*“ (siehe Abbildung 26).

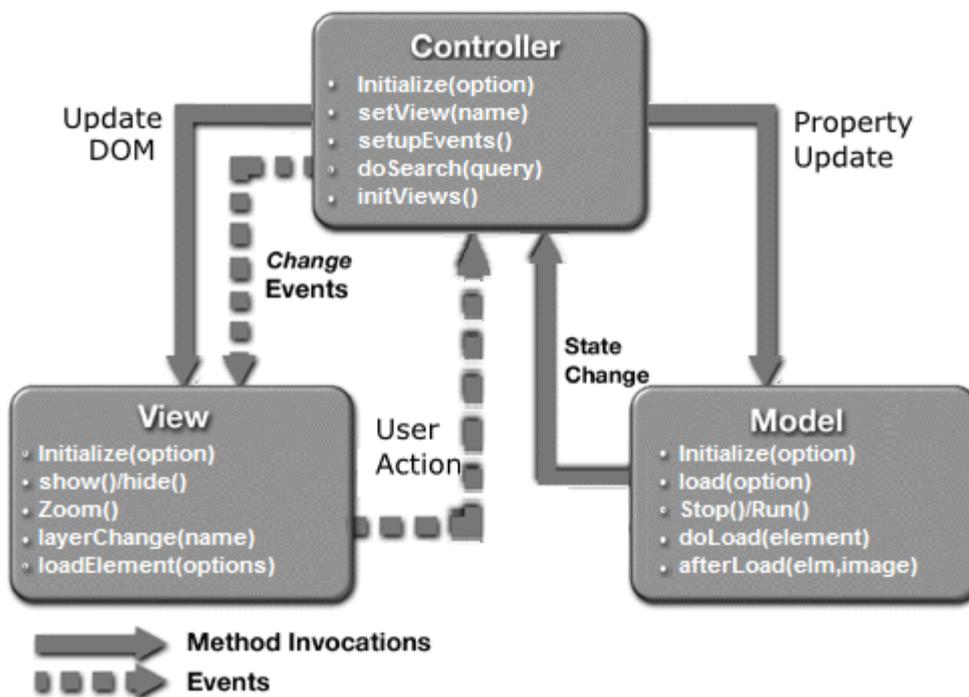


Abbildung 26 Klassendiagramm des Web Books Viewers

Die Loader-Klasse entspricht der Model-Ebene des MVC und ist für alles rund um die Datenhaltung zuständig. In diesem Fall handelt es sich bei den Daten um Digitalisate (Bücher und Handschriften), die das Austria-Forum der Öffentlichkeit zur Verfügung stellt. Die View-Klasse ist für die Darstellung der vom Loader erhaltenen Daten zuständig. Allen voran für die unterschiedlichen Ansichtsmodi, welche die verfügbaren Digitalisate darstellen sollen. Die Controller-Ebene des MVC wird durch die WebBooksReader-Klasse repräsentiert und übernimmt die Steuerung und die Logik des Viewers. Sie beinhaltet Methoden, die beispielsweise auf Benutzerinteraktionen wie das Umblättern reagieren und die View aktualisieren, sobald neue Daten vom Loader erhalten werden.

Im Anschluss werden die Methoden und somit die Funktionalitäten beschrieben, die von jeder einzelnen Klasse gewährleistet werden.

6.2.1 Model-Operationen

Die Loader-Klasse wurde anhand des Models implementiert und ist hauptsächlich für das Laden der Daten zuständig. Dies geschieht durch HTTP Anfragen. Wie in Abschnitt 5.4 beschrieben, werden die benötigten Daten durch eine asynchrone Kommunikation zwischen

dem Server und dem Client geliefert. Diese Daten, sprich Bücher- und Handschriftseiten, werden via URLs abgefragt. In dieser Klasse werden Request-Objekte konstruiert, die URLs für die Request-Daten enthalten. Dadurch wird es möglich, Request-Abfragen zum Server zu schicken. Der „Server Response“ mit den abgefragten Daten wird daraufhin empfangen und zur View weitergeleitet.

Die Loader-Klasse beinhaltet folgende Methoden:

- **Initialize(options)**

Dient als Konstruktor um die Erzeugung eines *Loader* Objekts zu ermöglichen. Als Parameter kann man die „Root URL“, in unserem Fall `http://austria-forum.org/wbv2/api/file/` (siehe Abschnitt 5.3.1.2) sowie ein „Booch Id“ angeben. Diese Parameter sind allerdings optional.

- **Load(options)**

Innerhalb dieser Methode wird die URL basierend auf der „Root URL“ weiter aufgebaut. Diese wird mit „reference/layer/format/scale“ ergänzt. Sie bietet die Möglichkeit, ein Array von Parametern hinzuzufügen. Man kann somit eine bestimmte Seite mit einem gewünschten Layer, Format und Scale-Faktor abrufen. Die Parameter sind wiederum optional.

- **Stop()**

Diese Methode stoppt das Laden der Buchseiten, wenn die Zahl der bisher geladenen Seiten eine bestimmte Menge erreicht.

- **Run()**

Diese Methode startet das Laden von Seiten, wenn die Zahl der geladenen Seiten nicht vollendet ist.

- **doLoad(element)**

Diese ladet eine bestimmte Seite. Der Parameter „element“ ist ein Pflichtparameter. Diese Funktion wird nur aufgerufen, wenn man eine Seite „element“ abrufen will.

- **afterLoad(element, image)**

Diese Funktion fügt die geladenen Seiten zum DOM ein und benötigt zwei Pflichtparameter. Während die erste die Stelle repräsentiert, an der die Seite eingefügt werden soll, entspricht das zweite Parameter der Buchseitenreferenz.

Die Methoden *stop()* und *run()* stellen das Entwurfsmuster *Lazy loading* dar. Es handelt sich dabei um einen *Performance*-Ansatz. Dabei wird beim Buchöffnen bzw. beim Laden der einzelnen Buchseiten nur eine bestimmte Seitenanzahl gleichzeitig vom Server abgerufen und im Viewer angezeigt. Will der Leser weitere Seiten lesen, wird eine entsprechende Menge an neuen Seiten geladen. Dieser Prozess wiederholt sich so lange, bis das gesamte Buch geladen wurde. Das erneute Laden soll im Hintergrund ablaufen, damit der Leser kaum etwas davon bemerkt.

6.2.2 View-Operationen

Unter *View* werden unter anderem die verschiedenen Ansichtsmodi verstanden, welche der *Web Books Viewer* beim Lesen eines Buches anbietet. Dabei handelt es sich um einen Doppelseiten-, Einzelseiten- und *Thumbnail*modus, welche beim Lesen jedes Buches und jeder Handschrift zur Verfügung stehen sollen.

Beim Öffnen des Buches soll der Doppelseitenmodus standardmäßig erscheinen, der das Digitalisat analog zu einem geöffneten Buch anzeigt. Es erscheinen mit Ausnahme des Buchumschlages zwei Seiten nebeneinander (*side by side*). Gleichzeitig werden in einer *Sidebar* alle Buchseiten als Miniaturbilder aufgelistet und es erfolgt ein Hinweis auf die aktuelle Seite.

Im Einzelseitenmodus erscheinen nur einzelne Buchseiten, während die *Thumbnails* in der *Sidebar* bestehen bleiben, um einen Überblick zu liefern.

Bei aktivierter *Thumbnail*-Ansicht erscheinen im *View port* die einzelnen Buchseiten in Miniaturansicht.

Aus der *View*-Klasse leitet sich eine spezielle Ansicht ab, die als Korrektur-Ansicht bezeichnet wird. Sie ähnelt dem Doppelseitenmodus, aber unterscheidet sich dadurch, dass inhaltlich gesehen die selbe Seite nebeneinander erscheint. Links wird jedoch die Seite in dem *OCR Layer* dargestellt und rechts in dem *Faksimile Layer*. Dieser Modus dient dazu, den *OCR Layer* anhand eines direkten Vergleiches auf Fehler hin zu untersuchen und somit dessen Qualität schrittweise zu erhöhen.

Allerdings verfügen derzeit noch nicht alle vorhandenen Bücher und Handschriften über einen *OCR Layer*. Außerdem ist zu diesem Zeitpunkt noch keine Serverschnittstellen vorhanden,

die es erlaubt, die entsprechenden Änderungen durchzuführen. Dieser Mängel sollen jedoch bald behoben werden.

Die *View*-Klasse verfügt über verschiedene Methoden, die nachstehend erläutert werden:

- **Initialize(options)**

Dient als Konstruktor um ein *View* Objekt zu erzeugen. Optional kann man einige Parameter eingeben. Diese sind: Name der Ansicht, id, css-Klasse, Name des *Controllers*, Seitenanzahl, Zahl der *Layer*, Zoomfaktor, eine Boolean Variable, die besagt ob die erzeugten Ansicht nach angezeigt werden soll oder nicht, minimale Größe, in der die Seiten angezeigt werden sollen, Format, der in dieser Ansicht angezeigten Digitalisate;

Mit dieser Methode werden die Bausteine der Ansicht festgelegt. Innerhalb der erzeugten *View* wird ein *Loader* Objekt instanziiert und beide anschließend einem *Controller* zugewiesen. Außerdem wird die HTML Grundstruktur dynamisch generiert, um die Darstellung der von dem *Loader* abgeholten Daten anzuzeigen.

- **Show()**

Da die Ansicht nach ihrer Erzeugung meistens verborgen bleibt, wird diese Methode aufgerufen um sie anzeigen. Diese Methode wird ebenfalls aufgerufen um die gewünschte Ansicht am Bildschirm anzuzeigen, wenn der Leser zwischen verschiedenen Ansichten wechselt.

- **Hide()**

Diese Methode hat genau die gegenteilige Funktion der vorangegangenen. Sie wird aufgerufen um eine Ansicht zu verbergen.

- **Resize()**

Notwendige Größenänderungen werden mit dieser Methode durchgeführt. Sie kommt hauptsächlich dann zur Anwendung, wenn die Größe des *View port's* bzw. der Anzeigefläche des Browsers geändert wird und dient dazu, die Größe der Ansicht an die Größe des aktuellen *View port's* anzupassen.

- **Zoom(faktor)**

Diese Methode benötigt einen Zoomfaktor als Pflichtparameter und ändert die Größe der Ansicht um diesen Faktor.

- **LayerChange(layerName)**

Diese Methode wechselt den Ansichtslayer zu dem eingegebenen Pflichtparameter „layerName“.

- **loadElement(options)**

Diese Methode hat im Wesentlichen die Aufgabe, dem *Loader* die eingegebenen Pflichtparameter weiterzureichen, damit eine bestimmte Buchseite geladen werden kann.

- **showSearchResult(data)**

Diese Methode gibt der *View* die Suchergebnisse weiter und benötigt „data“ als Pflichtparameter.

Bei der Konstruktion dieser Klasse wurden alle Eigenschaften jeder einzelnen Ansicht berücksichtigt.

6.2.3 *Controller*- Operationen

Der *Controller* ist die wichtigste Komponente des MVC Musters. Er hat im Wesentlichen die Aufgabe, die Daten des *Model*'s zu bearbeiten und an die *View* weiterzureichen. Des Weiteren nimmt er die Benutzerinteraktionen entgegen, bewertet sie und agiert entsprechend.

In der Regel ist ein *Controller* für eine *View* und ihren *Model* zuständig.

In dem *Web Books Viewer*, der in dieser Arbeit beschrieben wird, wurde das *Controller* Konzept mittels der Klasse „WebBooksReader“ umgesetzt. Diese Klasse wird als Entscheidungsträger bzw. Vermittler betrachtet und übernimmt die Steuerung des Viewers.

Innerhalb dieser Klasse werden die *View*-Objekte mit dem dazugehörigen *Model* initialisiert. Daraufhin werden die Benutzerereignisse (engl. *Events listener*) gesetzt und in deren Folge Aktionen ausgelöst. Die Rolle des *Controller*'s in dem Viewer wird in Abschnitt 6.3 näher dargestellt.

Die Operationen der „WebBooksReader“-Klasse werden wie folgt definiert:

- **Initialize(options)**

Diese Methode erzeugt ein Web Books Viewer Objekt. Beim Aufruf dieses Konstruktors können einige Parameter optional angegeben werden. Diese sind: „library-URL“, „File-URL“ und „search-URL“. Weitere sind „Name“ des *Layer*‘s, „Name“ der *View*, „Seitennummer“, „Zoomfaktor“ und der „Container“, an dessen Stelle die Buchansicht in DOM hinzugefügt werden soll sowie der „Thumbnails-container“, an dessen Stelle die Thumbnail-Bilder eingelegt werden sollen.

- **setView(name)**

Diese Methode aktiviert eine *View* und zeigt sie an. Sie benötigt als Pflichtparameter „name“, den Namen der gewünschten Ansicht.

- **setupEvents()**

Diese Methode setzt *Events listener* auf Benutzerinteraktionen, um sie abzufangen und dem Event (z.B. Umblättern einer Seite) entsprechend auszulösen.

- **getBook()**

Liefert das Buch Objekt zurück, das alle Informationen über das geöffnete Buch beinhaltet.

- **getCurrentPage()**

Die Seitennummer der aktuellen Buchseiten wird von dieser Methode zurückgeliefert.

- **setCurrentPage(index)**

Diese Methode zeigt die Buchseite an, die das angegebene Pflichtparameter „index“ als Seitennummer aufweist.

- **setCurrentPageOffset(offset)**

Diese Methode wird aufgerufen, um die als „offset“ angegebene Zahl nach vorne oder nach hinten zu springen und diese Seite dann anzuzeigen.

- **getCurrentLayer()**

Von dieser Methode wird der *Layer*-Name der angezeigten Buchseite zurückgeliefert.

- **setCurrentLayer(Name)**

Diese Methode setzt den Namen des *Layer*‘s auf das angegebene Pflichtparameter „Layername“.

- **doSearch(query)**

Diese Methode ist für die Suche im Buch zuständig. Sie benötigt die Suchanfrage „query“ als Pflichtparameter. Das *Model* der aktuellen *View* wird daraufhin aufgefordert einen AJAX Aufruf zu betätigen. Der *Controller* erhält die *Server-Response* und bearbeitet sie. Anschließend werden die Suchergebnisse der *View* weitergegeben um sie anzeigen zu lassen..

- **initViews()**

Diese Methode ist eine der wichtigsten in dieser Klasse. Es handelt sich dabei um die Instanziierung und die Initialisierung aller Ansichten. Innerhalb dieser Methode werden die drei Ansichtsmodi erzeugt. Sie weisen die selbe Struktur auf und bieten die gleichen Funktionalitäten an, jedoch ähneln sich weder die Gestaltung noch die Implementierung der Methoden. Deshalb wird jede Ansicht einzeln von der Klasse *View* abgeleitet und entsprechend erweitert.

6.2.4 Kommunikation zwischen *Model-View* und *Controller*

In Abschnitt 6.2 wurde der Ansatz für die Steuerungsaufteilung des Web Books Viewers präsentiert. Ausgangspunkt ist seine Realisierung bei der fast alle Operationen vom *Controller* gesteuert werden. Bei diesem Ansatz wird dem *Controller* eine Visualisierungskomponente (*View*) und eine Datenhaltungskomponente (*Model*) zugewiesen.

Der typische Ablauf stellt sich so dar, dass eine Instanziierung der „WebBooksViewer“-Klasse, die als *Controller* zu betrachten ist, beim Starten des Viewers stattfindet. Der *Controller* leitet die Ansichtsmodi sowie ihre *Models* ab und ergänzt ihre einzelnen Methoden. Standardmäßig ist die Doppelseitenansicht aktiviert und die Coverseite wird am Bildschirm im Anzeigefenster angezeigt. Im Abschnitt 6.3 werden mit Hilfe von Screenshots die Oberfläche und deren Funktionalitäten eingehender erklärt.

Der *Controller* nimmt die Schaltflächenklicks sowie die *Keydown*-Ereignisse des Lesers als Benutzerinteraktionen entgegen und agiert entsprechend darauf. Als Benutzerinteraktionen gelten z.B. das Umblättern, der Wechsel der Ansicht oder des Layers oder das Zoomen. Der *Controller* steuert auch die anderen Schichten und fordert anhand der Geschehnisse Anpassungen an und führt Aktionen durch.

Des Weiteren gibt er die von dem Leser durchgeführten Datenänderungen an das *Model* weiter, welches sie wiederum durch eine entsprechende Serverschnittstelle an den Server weiterreicht. Die Kommunikation erfolgt durch Methodenaufrufe des jeweiligen Objektes. Oftmals sind Parameterangaben erforderlich. Diese werden entweder vom Leser angegeben bzw. ausgewählt oder vom *Controller* selbst eingereicht.

Die *View*- und die *Model*-Ebene können nicht direkt miteinander kommunizieren, sondern ausschließlich über den *Controller*, der für beide Objekte zuständig ist.

6.3 Erläuterung der Oberflächenarchitektur und deren Funktionalitäten

Wie in Abschnitt 4.2.1 erwähnt wurde, soll eine intuitive und übersichtliche Benutzeroberfläche umgesetzt werden. Zur Gestaltung des *Graphical User Interfaces* wurde sowohl der in Abbildung 5.2 dargestellte Entwurf von Dr. Müller als auch aktuelle *e Book Reader*-Oberflächen in Betracht gezogen. Ausgangspunkt ist die GUI des auf Flash basierten Viewers, der zurzeit vom Austria-Forum für die Veröffentlichung der Digitalisate verwendet wird.

Im Anschluss werden die Funktionalitäten, die im Zuge dieser Arbeit umgesetzt werden, mittels Screenshots beschrieben.

6.3.1 Grundlegender Aufbau

Die Oberfläche des Viewers teilt sich in drei Hauptbereiche:

- **Funktionsleiste (*Toolbar*):** Diese befindet sich im oberen Bereich des Bildschirms und ermöglicht mit Schaltflächen und Eingabefeldern den Zugriff auf die Viewer-Funktionalitäten. Sie beinhaltet *Dropdown*-Listen für die *Layers* und die Ansichtsmodi, ein Suchfeld, *Zoomtools* sowie Buttons für die Seitennavigation und das Verbergen/Anzeigen der *Sidebar*.
- **Randfenster (*Sidebar*):** Die *Sidebar* umfasst einen schmalen Bereich auf der rechten Seite des Anzeigefensters. Sie stellt dem Leser ansichtsspezifische Informationen wie Doppelseithumbnails, Einzelseithumbnails oder Metadaten des geöffneten Buches

zur Verfügung. Sie weist eine eigene Funktionsleiste auf, welche es ermöglicht, zwischen der Thumbnail- und der Metadatenansicht umzuschalten.

- **Anzeigefenster (*View port*):** Hier werden die Buch- bzw. Handschriftseiten abgebildet.

Abbildung 27 zeigt die Ansicht, die unmittelbar nach der Buchauswahl auf dem Screen erscheint.



Abbildung 27 Hauptbildschirm der Anwendung

6.3.2 Bibliothekübersicht

Der Inhalt des Austria-Forums ist verschiedenen Wissenssammlungen zugeordnet.

Eine Kategorie bilden die Web Books (siehe Abschnitt 2.3). Innerhalb dieser Kategorie sind die Bücher und Handschriften wiederum verschiedenen Kategorien zugeordnet, aus denen der Leser das gewünschte Buch bzw. die gewünschte Handschrift auswählen kann.

Das ausgewählte Buch wird anschließend mit dem Web Books Viewer geöffnet. Um ein anderes Buch zu lesen, ist es nicht nötig, wieder in die vorherige Browseransicht zurückzukehren, da der Viewer eine eigene Bibliothekübersicht anbietet (siehe Abbildung 28).



Abbildung 28 Abschnitt der Bibliothekansicht des Web Books Viewers

6.3.3 Ansichtsmodi

Der Viewer bietet drei Lesemodi, in denen jedes Buch bzw. jede Handschrift angezeigt werden kann. Im Einzel- und Doppelseitenmodus wird das Buch im Hauptbereich des Anzeigefensters angezeigt und die entsprechenden Thumbnails in der Sidebar. Im Thumbnailmodus erscheinen in der Sidebar die Metadaten des jeweiligen Digitalisats.

- **Einzelseitenmodus**

In diesem Modus werden einzelne Buchseiten im *View port* und Einzelseitentumbnails in der *Sidebar* angezeigt. Der Leser kann die Buchseiten umblättern. Die Abbildung 29 zeigt ein in diesem Modus geöffnetes Buch.

- **Doppelseitenmodus**

Dieser Modus ist beim Öffnen des Buches aktiviert. Er zeigt zwei Buchseiten nebeneinander an und entspricht der Ansicht eines aufgeschlagenen Buches. Zusätzlich erscheinen Doppelthumbnails in der *Sidebar*. Der Leser kann gleich wie im Einzelseitenmodus die Buchseiten umblättern. Bei dem im Rahmen dieser Arbeit umgesetzten Prototyp ist noch keine fortgeschrittene Umblätteranimation vorgesehen. Die Abbildung 30 zeigt ein Lesebeispiel in diesem Modus.

- **Thumbnailmodus**

Bei diesem Modus handelt es sich um eine Ansicht, die alle Buchseiten in Form von *Thumbnails* im *View Port* anzeigt. Parallel dazu erscheinen die Metadaten des Buches in der *Sidebar*. Die Navigation erfolgt über das Scrollen. Nachdem der Leser die gewünschte Seite angeklickt hat, erfolgt ihre Anzeige im Einzelseitenmodus. Die Abbildung 31 zeigt den Thumbnailmodus.

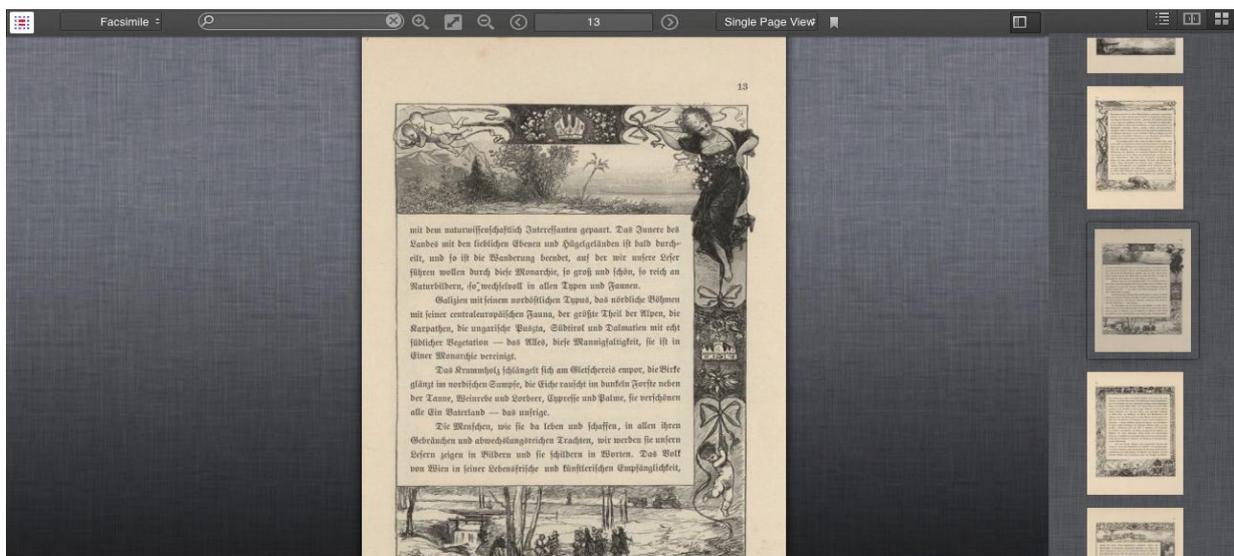


Abbildung 29 Einzelseitenmodus



Abbildung 30 Doppelseitenmodus

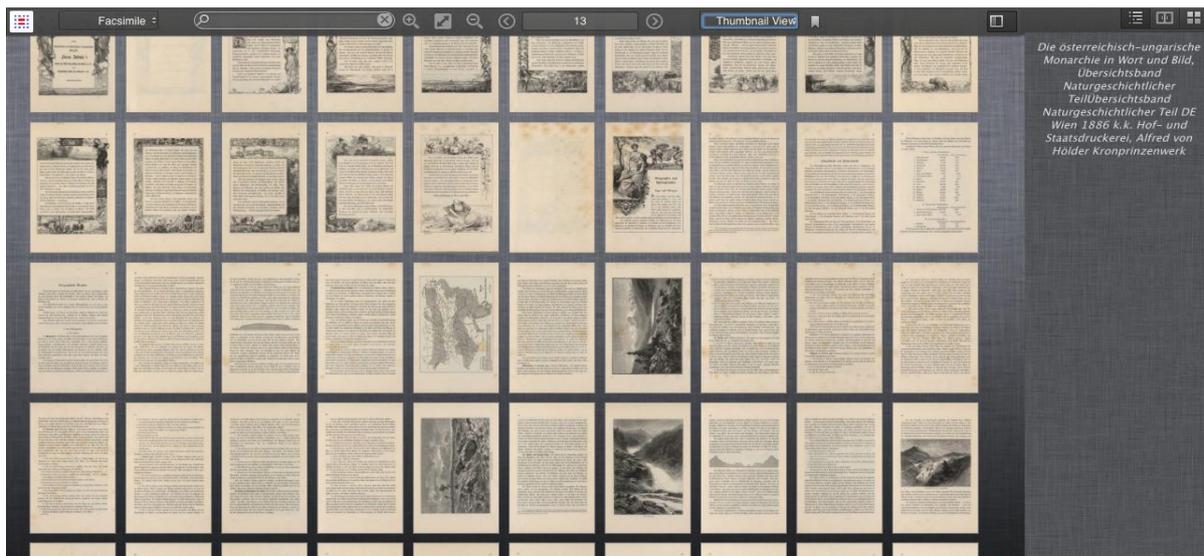


Abbildung 31 Thumbnailsseitenmodus

6.3.4 Layers

Der Viewer unterstützt alle *Layers*, in denen die Bücher und Handschriften im Austria Forum verfügbar sind: *OCR-Layer*, *Print* bzw. *Facsimile-Layer* und *Korrektur-Layer*. Der Leser kann zwischen den vorhandenen *Layers* wechseln.

- **Facsimile-Layer**

Es handelt sich dabei um das Scanformat des Digitalisats (siehe Abschnitt 2.2.2). In diesem Layer werden meistens *JPEG* Bilder veröffentlicht. Im Prototyp Viewer werden die Digitalisate zuerst in diesem *Layer* angezeigt, es sei denn, es handelt sich um *born-digitals*.

- **OCR-Layer**

Dieser Layer ist das Ergebnis der Durchführung einer *Optical character recognition*. Er dient dazu, dem Leser die Buchseiten in einem Textformat (entweder PDF oder sogar JPEG) anzubieten. Dies ermöglicht diverse Suchfunktionen und gewährleistet bei allen Digitalisaten ein problemloses Lesen. Jedoch kann dieser Layer Erkennungsfehler beinhalten, die aufgrund vieler Faktoren wie Schriftart oder Schriftgröße nicht gänzlich ausgeschlossen werden können.

- **Korrektur-Layer**

Dieser wird wie eine Doppelseitenansicht dargestellt. Die Buchseite wird links im *OCR-* und rechts im *Facsimile-Layer* abgebildet. Somit kann zwischen den beiden *Layers* verglichen und eventuell vorhandene Fehler lokalisiert werden. Die Fehlerkorrektur ist im Prototyp Viewer noch nicht realisiert, wird aber im Rahmen der Anforderungsklasse 3 berücksichtigt.

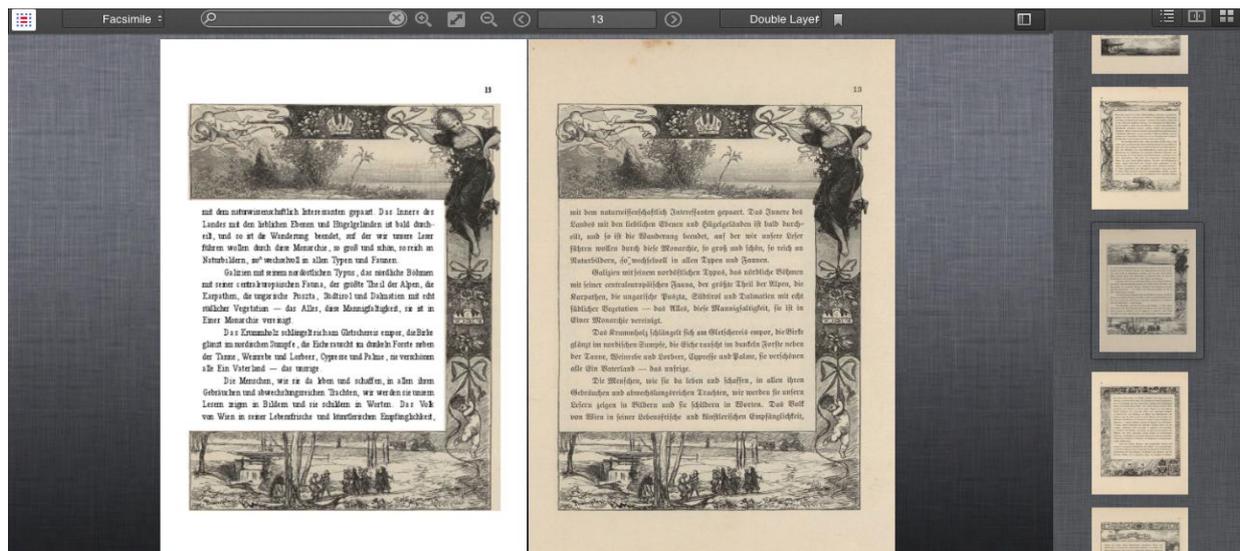


Abbildung 32 Korrektur-layer

6.3.5 Andere Funktionalitäten

Der Viewer bietet noch viele Standardfunktionalitäten an wie z.B. die Volltextsuche in Digitalisaten. Wenn der Leser einen Suchbegriff eingibt, ruft der *Controller* die Methode des *Models* jener Ansicht auf, die den *AJAX*-Aufruf mit der Suchanfrage (siehe 5.3.1) an den Server schickt. Dieser antwortet mit einer *JSON-Response*. Der *Controller* agiert dementsprechend und reicht der *View* die Liste der Seitenergebnisse (falls vorhanden) weiter. Daraufhin zeigt diese die Ergebnisse im Ansichtsfenster an.

Eine weitere Funktionalität stellt das Zoomen dar. Der Leser kann im Doppel- und Einzelseitenmodus die angezeigten Seiten vergrößern bzw. verkleinern. Am Anfang wird die Buchansicht angesichts der Ansichtsfenstergröße optimal berechnet. Diese optimale Größe steht dem Leser auch für einen spätere Auswahl zur Verfügung.

Andere Funktionalitäten wie das Aufsetzen von Lesezeichen oder das Selektieren von Textstellen aus dem Buch, um diese in Sozialen Netzwerken zu teilen, sind in diesem Prototyp noch nicht vorhanden.

6.4 Image Extractor

Im Rahmen dieser Masterarbeit wurde eine Software für die Extrahierung von Metadaten umgesetzt, welche die Anforderung „Bildersuche“, die in Abschnitt 4.2.1 beschrieben wurde, erfüllt. Diese Software stellt Informationen zu den in den Buchseiten eingebundenen Bildern bereit. Dies ermöglicht die Suche nach einem bestimmten Bild. Im Suchfeld ist entweder der vollständige Bildtitel einzugeben oder einzelne darin enthaltene Wörter. Diese Funktionalität bildet die Basis dafür, in einer erweiterten Version des Viewers eine eigene Bilderansicht bereitzustellen.

6.4.1 Klassendiagramm

Die Software wurde mit Java umgesetzt. Es handelt sich dabei um eine GUI Software, die nach der Eingabe des Pfades eines PDF-Dokuments die vorhandenen Bilder und deren Metadaten zurückliefert.

Die Abbildung 33 zeigt ein Klassendiagramm, das die Klassen des Softwaresystems und deren Beziehungen zueinander darstellt.

Die *Image Extractor Software* wurde mit Hilfe der *Apache PDFBox Library*¹⁸ umgesetzt und beinhaltet folgende Klassen:

- **Gui**: Dies ist die Hauptklasse und für die Generierung des GUI's sowie für die Benutzerinteraktionen zuständig. Nachdem ein Pfad zu der PDF-Datei ausgewählt wurde, wird diese der Klasse *PDFImageExtractor* weitergegeben.
- **PDFImageExtractor**: In dieser Klasse werden die einzelnen Seiten der PDF-Datei nach Bildern durchsucht. Sobald ein Bild gefunden wird, wird es extrahiert und in dem Zielordner gespeichert. Anschließend wird ein *PrintImageLocations*-Objekt instanziiert.

¹⁸ <http://pdfbox.apache.org/>

- **PrintImageLocations:** Diese Klasse kümmert sich um die Extrahierung der Bildposition. Diese wird aus der Darstellungsmatrix geparkt, in Prozentwerte konvertiert und gespeichert. Danach wird ein *TextStripper*-Objekt instanziiert.
- **TextStripper:** In dieser Klasse wird die Suche nach dem Titel (engl. *caption*) durchgeführt. Es wird rund um das Bild nach dem Titeltext gesucht. Entscheidend ist die Konfiguration, die beim Starten des Extrahierungsprozesses ausgewählt wurde. Standardmäßig wird die Suche unter dem Bild gestartet. Dabei wird der Inhalt einer dynamischen *Bounding-Box* Wort per Wort extrahiert und anhand der *Font*-Eigenschaften miteinander verglichen. Die fehlende Übereinstimmung stellt eine Abbruchbedingung dar. Der resultierende Text wird als Titelbild zurückgegeben.

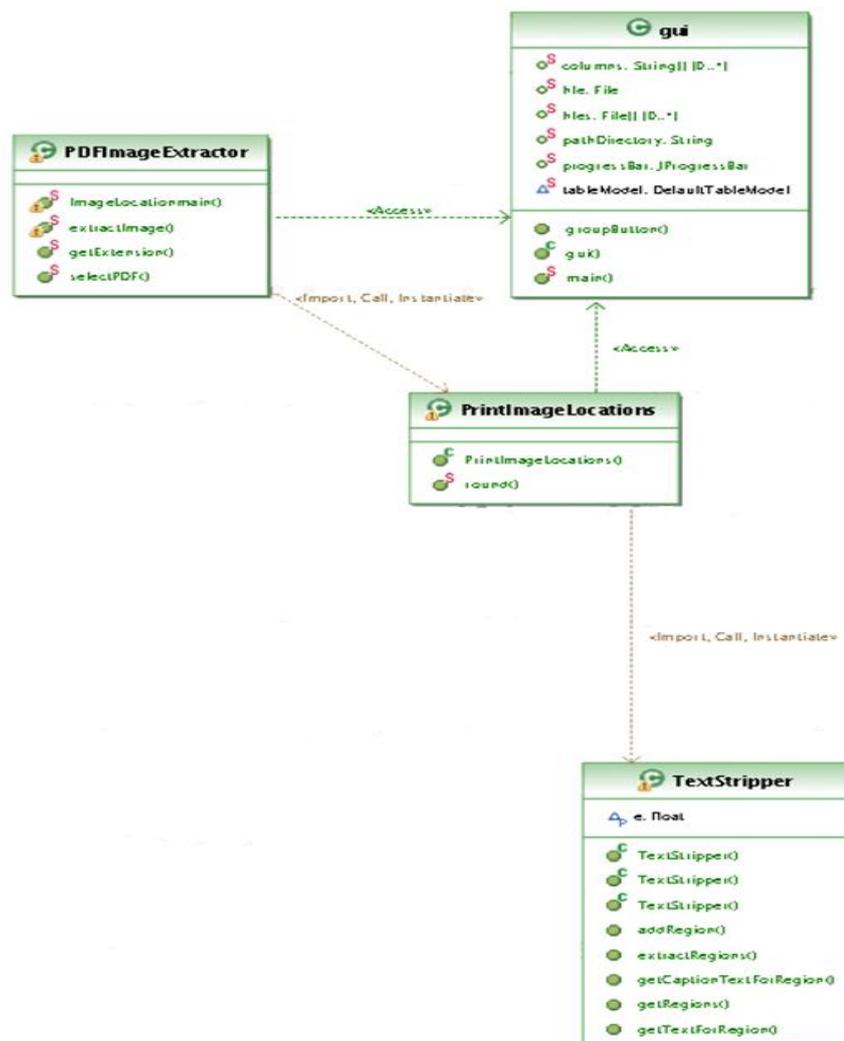


Abbildung 33 Klassendiagramm

6.4.2 Funktionsweise

Wie in der Abbildung 34 zu sehen ist, wurde eine einfache GUI mit Hilfe von *Java Swing*¹⁹ erstellt. Dabei kann man entweder den Pfad zu einer einzelnen PDF-Datei (*Use File path*) oder zu einem Folder (*Use Directory path*) eingeben bzw. auswählen. In einem tabularen Fenster werden die Informationen zu den einzelnen Bildern fortlaufend angezeigt. Eine Reihe von optionalen Konfigurationen wie z.B. die Position des Bildtextes können beim Anklicken des *Settings-Buttons* angepasst werden. Mit einem Klick auf den *Extract-Button* wird der Extrahierungsprozess gestartet und dessen Fortschritt in einer *progress bar* angezeigt.

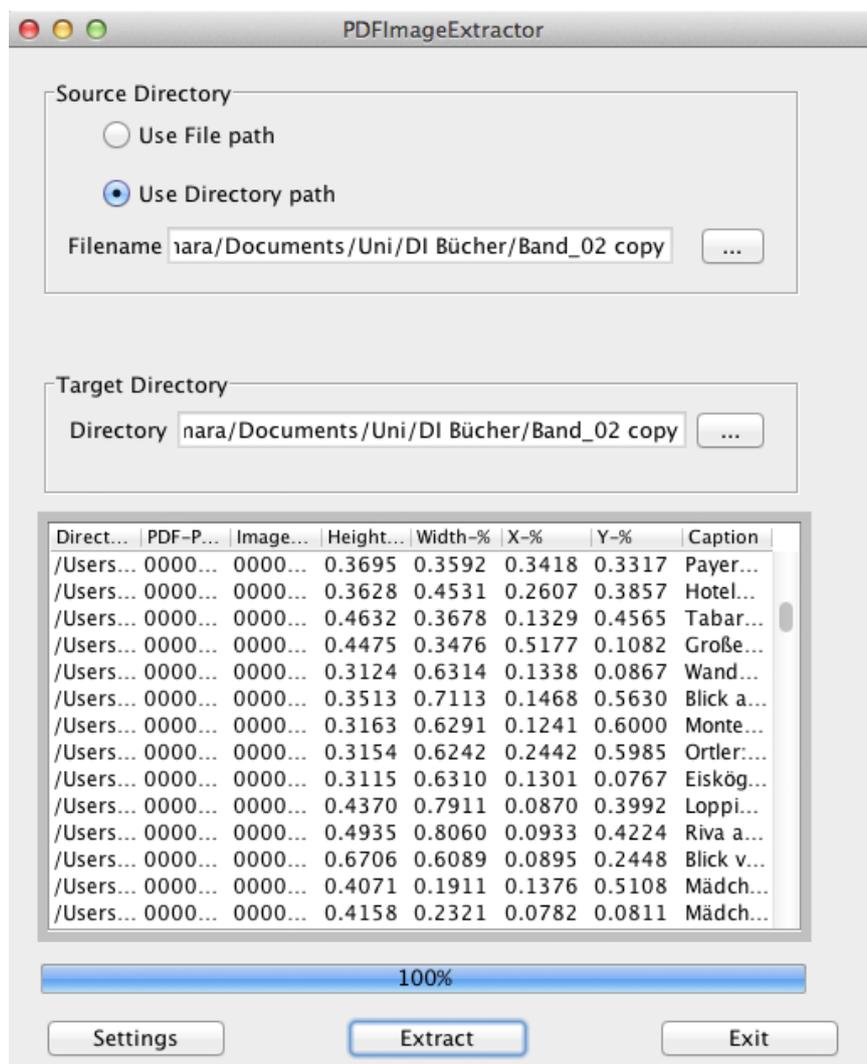


Abbildung 34 Image Extractor

¹⁹ <http://docs.oracle.com/javase/tutorial/uiswing/>

Die Operationen, welche die Bilder und dessen Metadaten extrahieren, werden wie in der in Abbildung 35 angezeigten Laufsequenz durchgeführt. Dabei laufen folgende Schritte ab:

- Aufruf der `extractImage()` Methode durch den *Extract-Button* auf der GUI
- der *Extension Name* der Datei wird überprüft, um sicherzustellen, dass es sich um eine PDF-Datei handelt
- Durchführung der Bildersuche auf jeder einzelnen Seite der PDF-Datei
- Extraktion des Bildes sowie dessen Größe und relative Positionierung auf der Seite
- Berechnung der Größe der *BoundingBox*, die den Titeltext des Bildes enthalten soll (standardmäßig unterhalb des Bildes)
- Extraktion und Speicherung des Textes

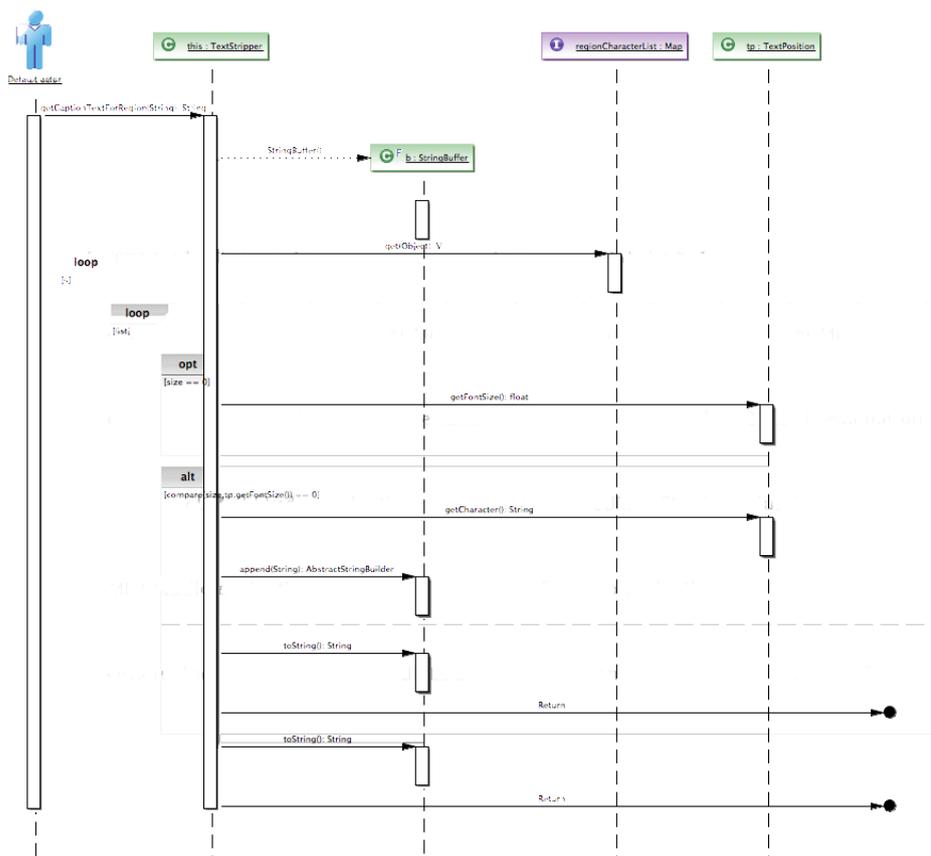


Abbildung 35 Laufsequenz

Abbildung 36 zeigt das Ergebnis der Suche nach dem Bild mit dem Titel „Der Bisamberg an der Donau bei Wien“. Im *View port* wird jene Buchseite angezeigt, die dieses Bild enthält und in der Sidebar erscheint die Buchseite als Einzelthumbnaill.



Abbildung 36 Ergebnis einer Bildsuche

6.5 Schwierigkeiten bei der Implementierung

Während der Implementierung des Web Books Viewers zeigte sich, dass die Umsetzung der formulierten Anforderungen schwieriger als zuvor angenommen ist. Deshalb konnte aus Zeitgründen nur ein Teil davon tatsächlich im Prototyp Viewer realisiert werden.

Im Anschluss werden jene Punkte herausgegriffen, die sich als besonders problemhaft herausstellten.

6.5.1 MooTools Framework

MooTools verfügt nicht nur über viele Vorteile, sondern weist auch ein paar Nachteile auf. Verglichen mit anderen Frameworks wie z.B. *jQuery*²⁰ ist die Hilfscommunity deutlich kleiner. Es wird auch eine geringere Anzahl von *Plugins* zur Verfügung gestellt. Während *jQuery* zahlreiche Umblätternanimationen anbietet, wird man bei MooTools nicht fündig. Die

²⁰ <http://jquery.com/>

Implementierung einer eigenen Umblätteranimation war deshalb erforderlich und stellte einen zusätzlichen Aufwand dar und kostete Zeit.

6.5.2 Benutzeroberfläche

Die Realisierung einer intuitiven und unkompliziert bedienbaren Benutzeroberfläche stellte eine große Herausforderung dar. Die Konzeption gestaltete sich schon deshalb schwierig, weil das Austria-Forum von unterschiedlichen Altersgruppen genutzt wird, die zum Teil divergierende Anforderungen an die Benutzeroberfläche stellen.

Für den im Rahmen dieser Diplomarbeit umgesetzten Prototyp Viewer wurde vorübergehend eine Benutzeroberfläche umgesetzt, die anderen zurzeit verfügbaren Viewern ähnelt und die Implementierung einer professionellen Benutzeroberfläche als spätere Option vorgesehen. Diese müsste allerdings von einem Profigrafiker designed werden.

6.5.3 Menge der Datenübertragung

Der Web Books Viewer stellt eine AJAX Webanwendung dar. Die Digitalisate und deren einzelne Seiten werden via AJAX vom Server abgerufen und im Browser via *JavaScript (MooTools)* ausgetauscht. Diese Kommunikationsweise verursacht eine massive Datenübertragungsrate. Deshalb fiel bei der Konzeption des Viewers die Entscheidung, dass das Anfordern der Daten vom Server nach dem Entwurfsmuster *Lazy loading* erfolgen soll. Dies hat den Vorteil, die Laufzeit bei der Datenübertragung gering zu halten und die Zahl der auf einmal geschickten *Server-Requests* zu kontrollieren. Jedoch führt dieses Entwurfsmuster beim Lesen der Digitalisate zu Performanceeinbußen. Das liegt daran, dass ein weiterer Request notwendig ist um weitere Seiten abzurufen. Es entsteht ein zusätzlicher Kommunikationsweg und daraufhin das Bearbeiten und Weiterreichen der angeforderten Daten, was zu einer zusätzlichen Wartezeit führt.

6.5.4 Größe und Komplexität der Anwendung

Entscheidend für die Umsetzung des *MooTools Frameworks* und dessen Vererbungskonzept war die Größe und Komplexität der Webanwendung. Das System des in dieser Arbeit

präsentierten Web Books Viewers wurde jedoch hinsichtlich seiner Business-Logik im Laufe der Entwicklung immer komplexer. Daher wäre eine neuerliche Projektplanung notwendig um die noch offenen, im Rahmen dieser Arbeit noch nicht implementierten Funktionalitäten zu einem späteren Zeitpunkt problemlos umsetzen zu können.

6.5.5 Sicherheitspezifische Aspekte

Einen wesentlichen Punkt bei diesem Web Books Viewer stellt die Sicherheit dar. Da die Webanwendung auf Datenübertragung basiert, stellt sich die Herausforderung, den Kommunikationsweg zwischen *Server* und *Client* zu sichern. Sicherheitstechnisch bedeutet das, dass der Gefahr des Abhörens und Abfangens der Daten entgegengewirkt werden muss. Angriffe wie beispielsweise *Man-in-the-Middle* oder *SQL-Injection* sollen gänzlich unmöglich sein. Das könnte mit Hilfe einer Verschlüsselung der Kommunikation zwischen den Parteien über SSL erreicht werden.

7 Kritische Evaluierung und Ausblick

7.1 Abdeckung der Anforderungen

Ziel der Implementierung des in dieser Arbeit präsentierten Prototyps war die Umsetzung des erarbeiteten Konzepts für die Visualisierung der Digitalisate, die vom Austria-Forum für die Öffentlichkeit zur Verfügung gestellt werden.

Im Rahmen dieser Arbeit konnten jedoch nicht alle Anforderungen an Funktionalitäten umgesetzt und realisiert werden. Die Grundfunktionalitäten, die für ein einwandfreies Funktionieren des Viewers essentiell sind, wurden allerdings zur Gänze implementiert. Eine Erweiterung sowie Weiterentwicklung der offenen Funktionalitäten sollte angesichts der angewandten Implementierungsstrategien durchaus möglich sein.

In der Tabelle x werden die in Abschnitt 4.2 präsentierten Anforderungen an den Viewer hinsichtlich des Grades ihrer Realisierung beschrieben.

Tabelle 2 Vergleich zwischen Anforderungen und realisierten Funktionalitäten

Anforderung	Realisierung
Authentifizierung	Diese Funktionalität wurde nicht realisiert. Das in Abschnitt 5.3.1.5 beschriebene API kann jedoch im prototypischen Viewer aufgerufen werden.
Bibliothek-Ansicht	Entsprechend der Austria-Forum Web-Books-Seite ermöglicht der Viewer eine Bibliothek-Ansicht, in der alle verfügbaren Bücher verschiedenen Kategorien zugeordnet sind.
Buch-Ansicht	Das Buch, das gerade gelesen wird, wird im Ansichtsfenster dargestellt. Zuerst erscheint die Doppelseitenansicht, wobei das Wechseln zwischen den verschiedenen Modi möglich ist.
Doppelseitenmodus	Das Buch wird doppelseitig zum Lesen geöffnet. Die

	Buchseiten sind nebeneinander dargestellt. Ein Buchrand sowie Seitenschatten werden abgebildet. Das Buch erscheint in diesem Modus wie ein analoges Buch.
Einzelseitenmodus	Die Seiten werden einzeln im Ansichtsfenster angezeigt.
Thumbnailmodus	Alle Seiten des Buches erscheinen im Ansichtsfenster als <i>Thumbnails</i> .
Moduswechsel	Das Umschalten zwischen den Ansichtsmodi erfolgt durch eine Dropdown-Liste, in der alle Ansichtsmodi aufgelistet sind. Nach getroffener Auswahl wird das Buch bzw. die Handschrift im gewünschten Ansichtsmodus angezeigt.
Anzeige der aktuellen Buchseite	Die Nummer der aktuellen Buchseite ist in einem <i>input</i> -Feld im <i>Header</i> abgebildet. Der Leser kann eine beliebige Seitennummer eingeben um die entsprechende Buchseite anzeigen zu lassen.
Ansichtslayer	Das Buch kann in allen vorhandenen Layers gelesen werden (<i>Facsimile</i> oder <i>OCR</i>).
Korrektur-Layer	Der Korrektur-Layer ähnelt der Doppelseitenansicht. Links wird die Buchseite im OCR-Layer und rechts im Facsimile-Layer angezeigt. Dies dient der Lokalisation von Fehlern im OCR-Layer.
Umblättern	Das Vor- und Zurückblättern wurde implementiert, allerdings handelt es sich um eine einfach umgesetzte MooTools Animation. Das Umblättern wird mittels Keydowns sowie Buttons ermöglicht.
Zoom	Im Viewer wird die Navigation innerhalb der Buchseiten durch mehrere Zoomfunktionen unterstützt.
Metadaten-Ansicht	Die Metadaten jedes Digitalisats werden mit der Cover- bzw. ersten Seiten sowie in der Thumbnailansicht angezeigt.

Suche im Buch	Eine Volltextsuche wurde dank der Serverschnittstelle realisiert. Geliefert werden jene Buchseiten, welche die Suchbegriffe beinhalten sowie die relative Positionierungen der Wörter auf den Ergebnisseiten.
Vernetzendes Lesen	Diese Funktionalität wurde nicht realisiert. Das in [Müller &Hermann 11] beschriebene Konzept ist in den prototypischen Viewer zu implementieren.
Kollaboratives Lesen	Diese Funktionalität wurde noch nicht implementiert. Das Buch soll gruppenweise gelesen und annotiert werden können.
Contentsharing	Diese Funktionalität wurde nicht implementiert. Inhalte des Buches sollen in Sozialen Netzwerken wie Facebook und Twitter geteilt werden können.
Hilfesystem	Diese Funktionalität wurde nicht realisiert. Ein Bedienungshilfesystem ist in den prototypischen Viewer zu integrieren.
Intuitive GUI	Die Benutzeroberfläche wurde sehr intuitiv und durchschaubar realisiert. Das Ansichtsfenster teilt sich in drei Hauptbereiche. Im Header befindet sich die Menüleiste, im View port werden die Buchseiten angezeigt und die Sidebar stellt Thumbnails bzw. Metadaten zum Buch bereit.
Plattformunabhängigkeit	Dieser Prototyp Viewer ist eine webbasierte Anwendung und somit plattformunabhängig.
Browserkompatibilität	Der Viewer funktioniert auf allen gängigen Browsern.
Quellcode-Vollständigkeit	Der Code wurde gut strukturiert geschrieben und in regelmäßigen Abständen refaktoriert.
Wartbarkeit und Erweiterbarkeit	Da der Code in MooTools nach dem <i>Model-View-Controller</i> -Muster geschrieben wurde, ist eine Erweiterung und Änderung dieses Prototyp Viewers problemlos möglich.

Bilderansicht	Diese Funktionalität wurde im Prototyp Viewer nicht realisiert. Die Bilder im Buch sollten in einer eigenen Ansicht dargestellt werden können.
Bildersuche	Diese Funktionalität wurde implementiert, aber in den erstellten Prototyp nicht integriert.
Drucken	Diese Funktionalität wurde nicht realisiert.
Tooltips	Die Implementierung von Tooltips ist von den angebotenen Diensten abhängig und deshalb nicht realisiert.
Mehrbenutzerfähigkeit	Diese Funktionalität wurde nicht realisiert.
Drag-and-Drop-Unterstützung	Diese Funktionalität wurde nicht implementiert.
Navigation in den Viewer Komponenten	Die Navigation in den Komponenten erfolgt im Browser.
Code-Kommentare	Im Programm sind Kommentare integriert.
Mehrsprachigkeit	Diese Funktionalität wurde nicht realisiert
Thumbnails in der Sidebar	Die Buchseiten werden entsprechend der aktuellen Ansicht parallel in der Sidebar angezeigt.
Umschalten zwischen Layers	Das Umschalten zwischen den Buchlayers wurde mittels einer Dropdown-Liste realisiert. Aufgelistet sind die vorhandenen Layers, in denen das Buch gelesen werden kann.

7.2 Ausblick

Im Hinblick auf die in dieser Arbeit definierten Anforderungen an einen Web Books Viewer sowie die im vorherigen Abschnitt beleuchteten offenen Funktionalitäten, können noch einige Verbesserungen und Erweiterungen an dem bestehenden Prototyp Viewer vorgenommen werden.

Geplant ist die Hinzufügung weiterer *Server-APIs*, die zusätzliche Funktionalitäten ermöglichen sollen. Außerdem ist die Implementierung der im Kapitel 4 präsentierten Anforderungen an den Viewer zu verwirklichen. In künftigen Arbeiten hoffen wir, die positiven Effekte der implementierten *Layer* und Funktionalitäten auf das Leseverhalten evaluieren zu können.

8 Literaturverzeichnis

- [AEIOU, 95] AEIOU Oesterreich-Lexikon. <http://austria-forum.org/af/AEIOU> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Afzal, 09] M. T. Afzal, W. T. Balke, N. Kulathuramaiyer, H. Maurer (2009). Improving Citation Mining: Proc. of International Conference on Networked Digital Technologies (NDT 2009) Ostrava, The Czech Republic, July 29 - 31. pp. 116-121. Publisher: IEEE.
- [Afzal, 10] M. T. Afzal, W. T. Balke, N. Kulathuramaiyer, H. Maurer; (2010). Rule based Autonomous Citation Mining With TIERL; JDIM (Journal of Digital Information Management), Volume 8, Issue 1, 2010, pp. 196-204
- [Blogmobile, 13] Das Blogmobil. <http://www.blogmobil.at/blogmobil/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Brantl & Schäfer, 10] M. Brantl and I. Schäfer (2010). Massendigitalisierung von Handschriften und Alten Drucken, Herausforderungen und Erfahrungen an der Bayerischen Staatsbibliothek, 175-197
- [Crane, 06] Crane, G. (2006). What do you do with a million books?. D-Lib Magazine, 12(3), 1.
- [Deiss et al. 06] A. Deiss, B. Druckenmüller, A. Melle (2006). SOA vs. REST basierend auf Google, eBay, Amazon, Yahoo! Fachstudie Nr. 56. http://elib.uni-stuttgart.de/opus/volltexte/2006/2618/pdf/FACH_0056.pdf
- [DFG, 13] Praktische Hinweise zur Digitalisierung von Bibliotheksbeständen. (2013) "DFG-Praxisregeln Digitalisierung". DFG-Vordruck 12.151-02/13
http://www.dfg.de/formulare/12_151/12_151_de.pdf
- [DFG-Viewer, 13] Der DFG-Viewer. <http://dfg-viewer.de/ueber-das-projekt/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Dörr & Weber, 97] Marianne Dörr und Hartmut Weber:(1997). „Digitalisierung als Mittel der Bestandserhaltung? Abschlußbericht einer Arbeitsgruppe der Deutschen Forschungsgemeinschaft“, in: ZfBB 44 1, S. 55-78
- [Erkin, 10] Erkin, B. (2010). Statistikbasierter Lösungsansatz für die Fehlerkorrektur in OCR-

- Systemen: Auf Grundlage der n-Gramm Häufigkeitstabellen von Wolfgang Schönplflug. GRIN Verlag.
- [Faruqui & Pado, 10] Faruqui, M., Pado S. (2010). Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. Proc. KONVENS 2010. Saarbrücken, Germany.
- [Finkel et al. 05] Finkel, J. R., Grenager, T., Manning. C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proc. 43rd Annual Meeting of the Association for Computational Linguistics ,, 363-370.
- [Folksonomy, 09] Isabella Peters, Folksonomies. (2009) Indexing and Retrieval in Web 2.0, Walter de Gruyter & Co., Hawthorne, NJ
- [Garrett, 05] Garrett, J. J. (2005). Ajax: A new approach to web applications.
- [Golem, 13] Golem.de. (2013) Crocodoc macht aus Office- und PDF-Dokumenten HTML5. <http://www.golem.de/news/box-uebernahme-crocodoc-macht-aus-office-und-pdf-dokumenten-html5-1305-99175.html> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Google Books Library, 14] Google Books Library. <http://books.google.at/googlebooks/library/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Gotscharek et al., 09] Gotscharek, Annette, et al. (2009) ."On lexical resources for digitization of historical documents." Proceedings of the 9th ACM symposium on Document engineering. ACM,.
- [Grossman & Frieder, 04] Grossman, D. A., Frieder, O. (2004): „ Information Retrieval: Algorithms and Heuristics“, 2nd Edition, Springer
- [Hetcher, 06] Steven Hetcher, (2006). The Half-Fairness of Google's Plan to Make the World's Collection of Books Searchable, 13 MICH. TELECOMM. & TECH. L. REV. 1, 65-66.
- [Hoeren, 05] Hoeren, T. (2005). Urheberrecht und Wissenschaft-am Beispiel der Bibliotheksfreiheit. Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, 2, 69.
- [Hoeren, 05] Hoeren, T. (2005). Urheberrecht und Wissenschaft-am Beispiel der Bibliotheksfreiheit. Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, 2, 69.
- [Holley, 2009a] Holley, R. (2009, November). Crowdsourcing and social engagement: potential, power and freedom for libraries and users. In Proceedings of Pacific Rim Digital Library Alliance (PRDLA) Annual meeting and Conference: Libraries at the End of the World: Digital Content and Knowledge Creation (pp. 1-28).

- [Holley, 2009b] Holley, R. (2009) Many Hands Make Light Work: Public Collaborative OCR Text Correction in Australian Historic Newspapers, National Library of Australia, ISBN 9780642276940.
http://www.nla.gov.au/ndp/project_details/documents/ANDP_ManyHands.pdf
- [IA, 13] Internet Archive BookReader. <https://openlibrary.org/dev/docs/bookreader> (Datum des letzten Zugriffs: 27. Februar 2014)
- [IMAGNO, 14] Historische Bilder (IMAGNO) http://austria-forum.org/af/Bilder_und_Videos/Historische_Bilder_IMAGNO (Datum des letzten Zugriffs: 27. Februar 2014)
- [Kappe, 06] F. Kappe, H. Maurer, B. Zaka (2006) .Plagiarism - A Survey;; J.UCS 12,8
- [Kolbitsch & Maurer, 05] Kolbitsch, J., Maurer, H. (2005). Transclusions in an HTML-Based Environment. Journal of Communication and Information Technology, Volume 14, Issue 2, pp. 161-174.
- [Korica-Pehserl & Maurer, 13] Petra Korica-Pehserl and Hermann Maurer. (2013). Metadata Extraction from Books with Facts about Austria. In Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS '13). ACM, New York, NY, USA, , Pages 615 , 4 pages.
- [Krasner et. All 88] G. Krasner, S. Pope (1988): A Cookbook for Using Model-View-Controller User Interface Paradigm in Smalltalk-8, erschienen in Journal of Object-Oriented Programming, Ausgabe1(3),S.26–49,
- [Kronprinzenwerk, 14] Kronprinzenwerk Bildersammlung http://austria-forum.org/af/Bilder_und_Videos/Kronprinzenwerk (Datum des letzten Zugriffs: 27. Februar 2014)
- [Lesk, 04] Lesk, M.E (2004).: Understanding Digital Libraries, 2nd ed. Morgan Kaufmann, San Francisco, CA
- [Martin, Lessmann & Voß, 08] Nicole Martin, Stefan Lessmann und Stefan Voß (2008): Crowdsourcing: Systematisierung praktischer Ausprägungen und verwandter Konzepte. Institut für Wirtschaftsinformatik, Universität Hamburg.
- [Maurer & Mehmood, 14] H. Maurer, & R. Mehmood (2014). Merging Image Databases as an example for Information Integration.Institute for Information Systems and Computer Media. Central european Journal of Operations Research, to appear (2014)

- [Maurer & Müller, 11] Maurer, H., & Müller, H. (2011) Das Austria-Forum und die Informationswissenschaft. http://austria-forum.org/attach/Infos_zum_AF/Gesamtdarstellung_des_AF.pdf
- [Maurer, & Mueller, 13] Maurer, H., & Mueller, H. (2013). Can the Web turn into a digital library?. *International Journal on Digital Libraries*, 13(2), 65-75.
- [Maurer, 07a] H. Maurer, N. Kulathuramaiyer; (2007). Fighting plagiarism and IPR violation: why is it so important? *Information Services & Use*, vol.27, no. 4, IOS Press, 185-191
- [Maurer, 07b] H. Maurer, B. Zaka; (2007). Plagiarism - a problem and how to fight it;; *Proc. of ED-MEDIA 2007*, 4451-4458
- [Maurer, 14] H. Maurer (2014). Austria-Forum and beyond; *IPSI Journal for Internet Research*; ISSN 1820 – 4503, Vol. 10, No.2 to appear (2014)
- [Mehmood & Maurer, 13] Mehmood, R., & Maurer, H. (2013, December). Aspects of Information Integration in a Wiki. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services* (p. 575). ACM.
- [Mikhail et al. 2013] Youssef Mikhail, Mohamed Abou Ouda, Ahmed Omar. (2013) A Book, a Web Browser and a Tablet: How Bibliotheca Alexandrina's Book Viewer Framework Makes It Possible. *Open Repositories 2013 (To Appear)*
- [Miner et al. 12] Miner, G., Elder, J., Hill, T, Nisbet, R., Delen, D. and Fast, A. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Elsevier Academic Press. ISBN 978-0-12-386979-1
- [Mons, 09] Mons, B., Velterop, J. (2009). Nano-Publication in the e-Science Era, *Workshop on Semantic Web Applications in Scientific Discourse (SWASD 2009)*.
- [Müller & Maurer, 11] Müller, H., & Maurer, H. (2011, October). How to carry over historic books into social networks. In *Proceedings of the 4th ACM workshop on Online books, complementary social media and crowdsourcing* (pp. 25-34). ACM.
- [Neldon, 87] Nelson, T. H., & Nelson, T. H. (1987). *Computer lib: Dream machines* (p. 31). Redmond: Tempus Books of Microsoft Press.
- [Obcena pro, 10] *Pro JavaScript with MooTools Learning Advanced JavaScript Programming*. Mark Joseph Obcena
Foreword by Valerio Proietti, MooTools Founder and Lead Developer (2010)

- [Obcena, 10] Mark Joseph Obcena Foreword by Valerio Proietti, MooTools Founder and Lead Developer. (2010) Pro JavaScript with MooTools Learning Advanced JavaScript Programming. 3-171
- [OL, 13] Open Library. <https://openlibrary.org> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Pirker & Müller, 14] J. Pirker, H. Müller (2014). Web Books and Wurzbach; Journal of Internet Research, Vol. 10, No.2
- [Rasberry, 13] How to build a RESTful Web API on a Raspberry PI in Javascript. <http://thefloppydisk.wordpress.com/2013/05/08/how-to-build-a-restful-web-api-on-a-raspberry-pi-in-javascript/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Reddy & , Stclair 96] Reddy, R., & Stclair, G. (2001). The Million Book Digital Library Project. On-line paper.
- [Reddy, 96] Reddy, R. (1996). The universal library: Intelligent agents and information on demand. In Digital Libraries Research and Technology Advances (pp. 27-34). Springer Berlin Heidelberg.
- [Reddy, 96] Reddy, R. (1996). The universal library: Intelligent agents and information on demand. In Digital Libraries Research and Technology Advances (pp. 27-34). Springer Berlin Heidelberg.
- [Richardson & Ruby, 07] L. Richardson & S. Ruby (2007) RESTful Web Services. p. 53-141
- [Rivadeneira, 07] Rivadeneira, A. W., Gruen, D. M., Muller, M. J., & Millen, D. R. (2007). Getting our head in the clouds: toward evaluation studies of tagclouds. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 995-998). ACM.
- [Sankar et al. 06] Sankar, K. P., Ambati, V., Pratha, L., & Jawahar, C. V. (2006). Digitizing a million books: Challenges for document analysis. In Document Analysis Systems VII (pp. 425-436). Springer Berlin Heidelberg.
- [Sharky, 09] Sharky, Clay (2009) Here Comes Everybody: How Change Happens When People Come Together, Penguin Books, ISBN 9780141030623.
- [Srivastava & Sahami, 09] Srivastava, A., and Sahami. M. (2009). Text Mining: Classification, Clustering, and Applications. Boca Raton, FL: CRC Press. ISBN 978-1-4200-5940-3
- [Stanford NER, 14] Stanford Named Entity Recognizer. <http://nlp.stanford.edu/software/CRF-NER.shtml>

- (Datum des letzten Zugriffs: 27. Februar 2014)
- [Steve Jobs, 10] Steve Jobs Brief April 2010, <http://www.apple.com/hotnews/thoughts-on-flash/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [TC, 13] Leena Rao. (2013). Box Acquires Crocodoc To Add HTML5 Document Converter And Sleek Content Viewing Experience To Cloud Storage Platform. <http://techcrunch.com/2013/05/09/box-acquires-crocodoc-to-add-html5-document-converter-and-sleek-content-experience-to-cloud-storage-platform/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Trattner, 09]. Christoph Trattner (2009): Vom Austria-Forum zum Wiki Konzept. Dipl.-Arbeit TU-Graz, IICM
- [ULB Halle, 13] Martin-Luther-Universität Halle-Wittenberg. <http://bibliothek.uni-halle.de/projekte/dfg-viewer/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Uni-Würzburg, 13] Die Virtuelle Bibliothek Würzburg. <http://vb.uni-wuerzburg.de/ub/index.html> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Wegstein, 09] Wegstein, W., Mirjam Blümm, Dietmar Seipel, and Christian Schneiker. (2009). Digitalisierung von Primärquellen für die TextGrid-Umgebung: Modellfall Campe-Wörterbuch. ([http:// www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_R4_1.pdf](http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_R4_1.pdf))
- [Wendler, 99] Robin Wendler. (1999). Metadata in the library. (LDI Update), Harvard University Library Notes, No. 1294, July/August 1999, p.4-5
- [Wikipedia, 14a] Urheberrecht in Österreich http://de.wikipedia.org/wiki/Urheberrecht_%28%C3%96sterreich%29 (Datum des letzten Zugriffs: 27. Februar 2014)
- [Wikipedia, 14b] Gemeinfreiheit. <http://de.wikipedia.org/wiki/Gemeinfreiheit> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Wikipedia, 14c] Fraktur-Schrift http://de.wikipedia.org/wiki/Fraktur_%28Schrift%29 (Datum des letzten Zugriffs: 27. Februar 2014)
- [Wurzinger, 10] Wurzinger, G. (2010). Information Consolidation in Large Bodies of Information. Journal of Universal Computer Science. 16, 21 (2010), 3314-3323.
- [Zaka, 08] B. Zaka, N. Kulathuramayer, H. Maurer, W.-T. Balke (2008). Topic-Centered Aggregation

of Presentations for Learning Object Repurposing, Proc. of E-Learn 2008, 3335-3342

- [ZDNet 2008] Britta Widmann (2008). Scribd bringt webbasierten Viewer für Office-Dateien. <http://www.zdnet.de/39187355/scribd-bringt-webbasierten-viewer-fuer-office-dateien/> (Datum des letzten Zugriffs: 27. Februar 2014)
- [Zimmer, 2012] Michael Zimmer. (2012). The ethical (re)design of the Google Books project. In Proceedings of the 2012 iConference (iConference '12). ACM, New York, NY, USA, 363-369.