



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

**DEPARTMENT INFORMATION**

## **Bachelorarbeit**

### **Der OPAC aus dem Baukasten**

**Realisierung eines Katalog 2.0 unter Einbeziehung der  
Community**

*vorgelegt von*

*Jörg Schmitt und Marcel Stehle*

**überarbeitete Version  
1.0  
31.12.2010**

**Studiengang Bibliotheks- und Informationsmanagement**

**erste Prüferin: Prof. Ursula Schulz**

**zweite Prüferin: Prof. Christine Gläser**

**eingereicht am 21. September 2010**

## **Abstract**

Die vorliegende Bachelorthesis beschreibt die exemplarische Erstellung des Katalog 2.0 „bachelopac“ unter Verwendung der Open-Source-Software VuFind. Die Frage, inwieweit dies, mit den erlernten Kenntnissen und erworbenen Fähigkeiten während des Studiums Bibliotheks- und Informationsmanagement der HAW Hamburg, realisierbar ist, wird beantwortet. Hierzu werden vom Gemeinsamen Bibliotheksverbund (GBV) zur Verfügung gestellte Metadaten des Bibliotheksystems der HAW Hamburg verwendet. Die Vor- und Nachteile, die durch den Einsatz von Open-Source-Software in Bibliotheken entstehen, werden ebenso erläutert wie die Notwendigkeit zeitgemäßer Kataloge und deren Umsetzung im Hinblick auf Bibliothek 2.0 und Katalog 2.0. Angelehnt an eine Machbarkeitsstudie werden Projektaufgaben und Projektziele sowie ein Kriterienkatalog zur Beurteilung der eigenen Kenntnisse entwickelt und durchgeführt. Der praktische Teil dieser Arbeit befasst sich mit den Problemen und Schwierigkeiten der technischen Umsetzung, beschreibt aber auch Lösungswege und -konzepte. Neben der aktiven Unterstützung durch die Community wird auch die Möglichkeit zur Nachnutzung von bereits bestehenden Lösungen aufgezeigt.

**Schlagworte:** VuFind, Open-Source-Software, Katalog 2.0, OPAC 2.0, Bibliothek 2.0, Discovery Interface, Community, Machbarkeitsstudie.

<http://www.bachelopac.de.vu>

## **Kontakt zu den Autoren**

joerg.schmitt@haw-hamburg.de

marcel.stehle@student.hu-berlin.de

## **Lizenz**

Der Text dieser Arbeit sowie die Abbildungen 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 34, 35, 36, 37 und 38 stehen unter folgender Creative-Commons-

Lizenz: <http://creativecommons.org/licenses/by/3.0/de/>

**Wir bedanken uns bei allen, die uns unterstützt haben!**

# Inhaltsverzeichnis

Tabellenverzeichnis.....	VI
Abbildungsverzeichnis.....	VI
Listingverzeichnis.....	VIII
Abkürzungsverzeichnis.....	IX
1. Einleitung.....	1
2. Notwendigkeit zeitgemäßer Kataloge.....	3
3. Die Idee des Katalog 2.0.....	5
3.1. Anwendung der Prinzipien.....	7
3.2. Funktionen eines Katalog 2.0.....	13
3.3. Problematik und Problembewusstsein.....	16
4. Zum Einsatz von Open-Source-Software im Kontext von Bibliothek 2.0.....	17
5. VuFind – ein Open-Source-Discovery-Interface.....	19
5.1. Technischer Hintergrund .....	19
5.2. Serviceorientierte Architektur.....	21
5.3. Entstehung und Anwendungsszenarium.....	23
5.4. Funktionsumfang aus Nutzersicht.....	25
6. Studium – Inhalte und Ziele.....	29
7. Methodik der Arbeit.....	31
8. Beschreibung der praktischen Vorgehensweise.....	35
8.1. Machbarkeitsprüfung.....	35
8.2. Praktische Umsetzung der Methodik.....	36
8.2.1. Projektaufgaben.....	36
8.2.2. Projektziele.....	39
8.2.3. Einzusetzende Technologien/benötigte Ressourcen.....	39
8.2.4. Projektzeitrahmen.....	39
8.3. Bewertungskriterien.....	40

9. Der Weg zum bachelopac.....	42
9.1. Grundinstallation.....	42
9.1.1. Ubuntu Linux 8.04 LTS.....	42
9.1.2. VuFind 1.0.....	43
9.1.3. Basiskonfiguration .....	44
9.2. Import von Metadaten.....	46
9.2.1. Für den Import geeignete Formate.....	46
9.2.2. Pretest mit Beispieldatensätzen.....	48
9.2.3. Import eines Datengrundstocks des HIBS.....	50
9.3. Automatisierter Metadatenabgleich.....	52
9.3.1. .z3950fetch.....	53
9.3.2. Makulierung.....	55
9.4. Anzeige von Verfügbarkeitsinformationen.....	56
9.4.1. DAIA-Datenmodell.....	57
9.4.2. Funktionsweise der DAIA-API in der VuFind-Umgebung.....	58
9.4.3. Nutzung von DAIA im bachelopac.....	60
9.4.4. Anzeige der Standorte.....	61
9.5. Die Konfiguration von VuFind.....	63
9.5.1. Anpassung der Facetten.....	63
9.5.1.1. Deaktivierung nicht benötigter Facetten.....	64
9.5.1.2. Facette „Erscheinungsjahr“ .....	65
9.5.1.3. Ergänzung einer Checkbox-Facette.....	66
9.5.1.4. Ergänzung der Facette „Basisklassifikation“ .....	67
9.5.2. Suchsyntax von Lucene.....	72
9.5.3. Relevance Ranking.....	76
9.5.4. More Like This.....	78
9.5.5. Boosting von Lehrbüchern.....	79

9.5.6. Stoppworte und Synonyme.....	82
9.5.6.1. Stoppworte.....	82
9.5.6.2. Synonyme.....	83
9.5.7. Fehlertolerante Gestaltung der Suche.....	85
9.5.7.1. Rechtschreibkorrektur und Verweisungen.....	85
9.5.7.2. Reduktion von Wortformen (Stemming).....	86
9.5.7.3. Minimum Should Match.....	88
9.5.8. Personalisierung .....	91
9.5.8.1. Technik.....	91
9.5.8.2. Funktionen.....	92
9.5.9. Kataloganreicherungen und Mashups.....	97
9.5.9.1. Coveranzeige.....	97
9.5.9.2. Wikipedia.....	98
9.5.9.3. Rezensionen.....	98
9.5.9.4. Implementierung der Google Book Search API.....	100
9.5.9.5. Chatauskunft.....	104
9.5.10. Anpassung der Katalogoberfläche.....	110
9.5.10.1. Oberfläche für mobile Endgeräte.....	112
9.5.10.2. Anpassung von Übersetzungen.....	113
10. Nicht gelöste Probleme.....	114
11. Bewertung der Machbarkeit.....	118
11.1. Tabellarische Zusammenfassung.....	118
11.2. Beurteilung der eigenen Kenntnisse.....	121
12. Schlussbetrachtung.....	124
Literaturverzeichnis.....	126
Anhang.....	i

## Tabellenverzeichnis

Tabelle 1: Kodierung der DAIA-Services.....	57
Tabelle 2: Bewertung der Machbarkeit .....	118

## Abbildungsverzeichnis

Abbildung 1: Monolithische Architektur vs. SOA-System.....	22
Abbildung 2: Überblick über die Software-Architektur von VuFind.....	23
Abbildung 3: Mögliche Ergebnisalternativen von Machbarkeitsstudien.....	32
Abbildung 4: Theoretischer Ansatz des Bewertungsschemas.....	33
Abbildung 5: Bewertungsschema zur Lösung der Projektaufgaben.....	41
Abbildung 6: Fehlermeldung: "Solr index is offline".....	44
Abbildung 7: Trefferliste einer „leeren Suche“ nach Einspielung der DNB-Testdaten der Reihe B.....	49
Abbildung 8: Anzeige der Verfügbarkeit eines Exemplars im PICA-OPAC LBS4 der HAW Hamburg.....	56
Abbildung 9: Schematische Darstellung des DAIA-Datenmodells.....	57
Abbildung 10: Schematische Darstellung des Datenflusses zum Auslesen von Verfügbarkeitsinformationen, Standort und Signatur.....	58
Abbildung 11: HTML-Ausgabe des DAIA-Servers zur Prüfung der Verfügbarkeit einer PPN.....	59
Abbildung 12: Facetten Ära und Erscheinungsjahr im direkten Vergleich.....	65
Abbildung 13: Checkbox-Facette zur Einschränkung der Ergebnisse auf Abschlußarbeiten.....	66
Abbildung 14: Darstellung des MARC-Feldes 084 in der „Internansicht“ von VuFind.....	67
Abbildung 15: Vereinfachter Programmablaufplan zur Indexierung der BKL.....	69
Abbildung 16: BKL-Facetten von TUBFind und bachelopac.....	69
Abbildung 17: BKL in der erweiterten Suche.....	70
Abbildung 18: Suche nach „Dostoevskij“ im bachelopac (1 Treffer).....	72

Abbildung 19: Auszug der Trefferliste der unscharfen Suche nach "Dostoevskij~" im bachelopac.....	73
Abbildung 20: Suchanfrage nach „Inkunabel“ ohne Nutzung von Synonymen.....	83
Abbildung 21: Suchanfrage nach „Inkunabel“ mit Nutzung von Synonymen.....	84
Abbildung 22: Rechtschreibkorrektur und automatische Verweisungen im bachelopac.....	85
Abbildung 23: Stemming des Suchterminus "Häuser".....	87
Abbildung 24: Tweet von @bachelopac: "Was bedeutet denn diese Konfig..." ..	89
Abbildung 25: Tweet von @tillk: "@bachelopac Gibt ein Nutzer 3 bis 5 Suchterme ein..." ..	89
Abbildung 26: Treffer-Liste mit Datensätzen, die nicht alle Suchtermini enthalten.....	90
Abbildung 27: Öffentliche Liste zum Thema Web 2.0 und Bibliotheken.....	95
Abbildung 28: RSS-Feed der Katalog-Suche nach "Bibliothekswesen" im Google Reader.....	96
Abbildung 29: GBS API Preview Wizard.....	102
Abbildung 30: GBS-Button und eingebettete GBS-Vorschau im bachelopac.....	103
Abbildung 31: Konfiguration des MeeboMe-Chat-Widgets.....	105
Abbildung 32: Chat-Auskunft auf Basis von MeeboMe in der Null-Treffer-Liste.....	107
Abbildung 33: Chatauskunft auf Basis von LibraryH3lp.....	108
Abbildung 34: Für Smartphones optimierte Katalogoberfläche.....	112
Abbildung 35: Geänderte Übersetzung des Reiters „Beschreibung“ in der Titelanzeige.....	113
Abbildung 36: Unerwünschte Zeichen in Datensätzen.....	115
Abbildung 37: Fehlende Bandangaben in der Trefferliste.....	116
Abbildung 38: Fehlender Ausleihstatus in der Trefferliste.....	117
Abbildung 39: Tweet von @Kaess: "Kann's kaum erwarten, meiner EDV von @SBChemnitz #vufind schmackhaft zu machen..." ..	125

## Listingverzeichnis

Listing 1: Erste Anpassungen der Datei config.ini.....	44
Listing 2: Import von Testdatensätzen der DNB.....	48
Listing 3: Download von Metadaten mit Z3950fetch.php.....	53
Listing 4: Konfiguration von z3950fetch.php.....	54
Listing 5: Anbindung von DAIA an das PICA-LBS (DAIA.ini).....	60
Listing 6: Codierung der Bibliotheksstandorte in der Datei yourlocationfiles.ini..	62
Listing 7: Auszug aus der angepassten Konfigurationsdatei facets.ini.....	64
Listing 8: Solr-Index-Felder für die BKL (schema.xml).....	68
Listing 9: Eintrag des Mapping-Skripts bcl.sh in der Datei marc_local.properties.....	68
Listing 10: Gewichtung von Index-Feldern (searchspecs.yaml).....	77
Listing 11: Gewichtung von Operatoren (searchspecs.yaml).....	77
Listing 12: Felder, die bei More Like This verglichen werden (searchspecs.yaml).....	78
Listing 13: Gewichtung der Felder bei More Like This (searchspecs.yaml).....	78
Listing 14: Berücksichtigung der BKL in den "ähnlichen Treffern" (solrconfig.xml).....	79
Listing 15: Boosting von Lehrbüchern und Werken mit Inhaltsverzeichnissen in beluga (beluga-solrconfig.xml).....	80
Listing 16: Lehrbuchboosting im bachelopac (solrconfig.xml).....	80
Listing 17: Konfiguration der Nutzerauthentifizierung (config.ini).....	92
Listing 18: Einbindung von Rezensionen von Amazon (config.ini).....	99
Listing 19: Anzeige von deutschsprachigen Rezensionen von Amazon (amazon.php).....	99
Listing 20: Ausgabe des GBS-API Preview Wizards für eine eingebettete Buchvorschau der Größe 600 * 500 Pixel.....	102
Listing 21: Angepasstes JavaScript zur eingebetteten Darstellung von Google Books in der Titelanzeige.....	103
Listing 22: Code von MeeboMe, der in Websites eingebunden wird.....	106



## Abkürzungsverzeichnis

AIM	AOL Instant Messenger
AjBD	Arbeitsgemeinschaft für juristisches Bibliotheks- und Dokumentationswesen
APA	American Psychological Association
API	application programming interface
APT	Advanced Packaging Tool
bcl	basic classification
BIM	Bibliotheks- und Informationsmanagement
BKL	Basisklassifikation
bq	boost query
bzw.	beziehungsweise
CANMARC	CANadian Machine-Readable Cataloging
CBS	Central Library System
CC0-Lizenz	creative commons 0
CSS	Cascading Style Sheet
d.h.	das heißt
DAIA	Document Availability Information API
DB	Datenbank
DFG	Deutsche Forschungsgemeinschaft
DIN	Deutsches Institut für Normung
DisMax	Maximum Disjunction
DNB	Deutsche Nationalbibliothek
etc.	et cetera
FRBR	Functional Requirements for Bibliographic Records
GB	Gigabyte
GBV	gemeinsamer Bibliotheksverbund
GNU	GNU is not Unix
GPL	General Public License
HAW	Hochschule für Angewandte Wissenschaften
HeBIS	Hessisches BibliotheksInformationsSystem
HIBS	Hochschulinformations- und Bibliotheksservice der HAW Hamburg
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICQ	Homophon für "I seek you"., Instant Messaging- Dienst
ID / IDs	Identifizier
ILS	Integrated Library System
ISBN	Internationale Standardbuchnummer
ISIL	International Standard Identifier for Libraries and Related Organizations
ISO	International Organization for Standardization
ISSN	Internationale Standardseriennummer
J2SE	Java 2 Platform Standard Edition
JDK	Java Development Kit

J-ID	Jabber-Identifizier
KUG	Kölnener Universitäts-Gesamtkatalog
LBS	lokales Bibliothekssystem
LDAP	Lightweight Directory Access Protocol
LOC	Library of Congress
LTS	Long Term Support
MAB2	Maschinelles Austauschformat für Bibliotheken 2
MARC	Machine-Readable Cataloging
MLA	Modern Language Association
MSN	Windows Live Messenger
MYSQL	My Structured Query Language
OAI	Open Archives Initiative
OAI - PMH	Open Archives Initiative Protocol for Metadata
	Harvesting
OCLC	Online Computer Library Center
OPAC	Online Public Access Catalogue
OSS	Open-Source-Software
OSWD	Onlineschlagwortnormdatei
PHP	rekursives Akronym für "PHP: Hypertext
	Preprocessor"
PICA	Project of Integrated Catalogue Automation
PPN	PICA-Produktionsnummer
RAK	Regeln für die alphabetische Katalogisierung
RAM	Random-Access-Memory
RDF	Resource Description Framework
REST	Representational State Transfer
RSS	Really Simple Syndication
RVK	Regensburger Verbundklassifikation
SMS	Short Message Service
SOA	Serviceorientierten Architektur
Solr	Searching On Lucene w/Replication
SRU	Search/Retrieval via URL
SSH	secure shell
SUB	Staats- und Universitätsbibliothek
SWD	Schlagwortnormdateien
TUB HH	Technischen Universitäts Bibliothek Hamburg
	Harburg
UI	User Interface
UKMARC	United Kingdom Machine-Readable Cataloging
URL	Uniform Resource Locator
USA	United States of America
USMARC	USA Machine-Readable Cataloging
Vgl.	vergleiche
VPN	virtual private network
VZG	Verbundzentrale des Gemeinsamen
	Bibliotheksverbundes
XAMPP	X(Apache, MySQL, PHP, Pearl)

XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
XOPAC	Extendable Online Public Access Catalog
XSL	Extensible Stylesheet Language
z.B.	zum Beispiel
ZDB	Zeitschriftendatenbank

# 1. Einleitung

Der Diskurs über Web 2.0 und Katalog 2.0 begleitet die Verfasser schon seit dem Beginn ihres bibliothekarischen Studiums.

Die Notwendigkeit attraktiver Kataloge sollte eigentlich nicht mehr bestritten werden. Sieht man aber von den ambitionierten, aber wenigen Projekten von wissenschaftlichen Bibliotheken, wie KUG, beluga, xopac oder HeiDi ab und betrachtet die deutsche OPAC-Landschaft etwas allgemeiner, ist in den letzten Jahren wenig passiert.

Glücklicherweise ist die Zeit 1992 nicht stehen geblieben: Open-Source-Software ist auch im Bibliothekswesen immer beliebter geworden. Mittlerweile gibt es eine Reihe freier Alternativen zur kommerziellen Software, die nur darauf warten, ausprobiert und weiterentwickelt zu werden.

In zahlreichen deutschen Veröffentlichungen und wissenschaftlichen Blogs wird immer zunehmend VuFind als eine Alternative zu den jetzt im Einsatz befindlichen Bibliothekskatalogen genannt. In Deutschland wird VuFind bisher nur von der Universitätsbibliothek der Technischen Universität Hamburg-Harburg zur Entwicklung eines zeitgemäßen Bibliothekskataloges eingesetzt.<sup>1</sup>

Den Verfasser stellt sich daher folgende Frage: Inwieweit können Absolventen<sup>2</sup> eines bibliothekarischen Studiums einen zeitgemäßen OPAC auf Basis von VuFind selbst realisieren? Welche Fähigkeiten sind dazu erforderlich? Welche Kenntnisse müssen erworben oder von außen eingeholt werden?

Gemäß des Titels „Der OPAC aus dem Baukasten - Realisierung eines Kataloges 2.0 mit VuFind unter Einbeziehung der Community“ liegt ein Schwerpunkt dieser Arbeit darin, nach bereits vorhandenen „Bausteinen“, die für die Realisierung eines Kataloges 2.0 erforderlich sind, zu suchen und diese, wenn möglich, nachzunutzen.

Der Fokus liegt daher weniger auf der klassischen Literaturrecherche und der Beschreibung und Bewertung einzelner Funktionen sowie den zahlreichen Ideen, die im Kontext des Katalogs 2.0 erarbeitet wurden, sondern vielmehr darin,

---

<sup>1</sup> Vgl. MARAHRENS 2010 – Die Zukunft.

<sup>2</sup> Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit das generische Maskulinum verwendet.

deren mögliche Umsetzung mit einem auf VuFind basierenden Katalog zu erforschen.

Dies geschieht gerade auch unter dem von Wiesenmüller geforderten Paradigmenwechsel, der Bibliothekare auffordert weg von der Einstellung „So etwas geht eben mit unserem System nicht“ hin zu einem „Wir wollen es, wie bekommen wir es hin“<sup>3</sup> zu gehen.

Im ersten Teil dieser Arbeit werden kurz die Notwendigkeit zeitgemäßer Bibliothekskataloge, die Idee des Katalogs 2.0 und die Hintergründe von VuFind erläutert. Dieser erste Teil schließt mit einer Betrachtung der Vor- und Nachteile beim Einsatz von Open-Source-Software in Bibliotheken ab.

Der zweite Teil dieser Arbeit beginnt mit der Beschreibung, der durch das Studium an der HAW Hamburg gewonnenen Kenntnisse und Fähigkeiten, soweit diese für die Erstellung eines Bibliothekskatalogs mit VuFind relevant sind. Es folgt die Erläuterung der dieser Arbeit zugrunde liegenden Methodik, die sich der Mittel des Projektmanagements bedient und an eine Machbarkeitsstudie angelehnt ist, sowie die Beschreibung der praktischen Umsetzung dieser Methodik durch die Verfasser. Vor allem die praktische Umsetzung dieser Methodik berücksichtigt die Prinzipien der Bibliothek 2.0.

Die Beschreibung der Umsetzung des unter dem Projektnamen „bachelorpac“<sup>4</sup> erstellten Bibliothekskatalogs mit VuFind gliedert sich in die Installation von VuFind, die Anbindung an das lokale Bibliothekssystem (LBS) und die Beschreibung der notwendigen Anpassungen und Konfigurationen, die gemacht werden mussten. Die Verfasser nutzen zur Erstellung ihres Kataloges das LBS des Hochschulinformations- und Bibliotheksservice der HAW Hamburg (HIBS) mit einem durch den gemeinsamen Bibliotheksverbund (GBV) zu Verfügung gestellten Datenbestandes des HIBS. Das HIBS nutzt das LBS PICA des Herstellers Online Computer Library Center (OCLC). Mit einer Zusammenfassung der gelösten und gefundenen offenen Probleme endet das Kapitel.

Die Arbeit endet mit der Bewertung des erstellten Bibliothekskatalogs und der Bewertung der durch das Studium vermittelten Fähigkeiten. Eine Schlussbetrachtung schließt diese Arbeit ab.

---

<sup>3</sup> WIESENMÜLLER 2010 – Daten härter.

<sup>4</sup> <http://bachelorpac.de.vu/>.

## 2. Notwendigkeit zeitgemäßer Kataloge

Der vorrangige Sinn und Zweck eines Bibliothekskatalogs ist es, dem Nutzer einen umfassenden und bedürfnisorientierten Zugriff auf den Bestand zu ermöglichen.<sup>5</sup> Waren Bibliotheken und ihre OPACs zu Beginn der 1990er Jahre noch „the Jewel in the Crown“<sup>6</sup> bei der Befriedigung von Informationsbedürfnissen, stehen diese heute immer stärker mit anderen Informationsdienstleistungen aus dem Internet in Konkurrenz. Das Internet hat sich unter dem Schlagwort Web 2.0 in den letzten Jahren in vielerlei Hinsicht weiterentwickelt. Die heutigen „Durchschnitts-OPACs“ oder auch „DinOPACs“ werden den heutigen Anforderungen kaum noch gerecht<sup>7</sup> und setzen auf den überholten Ansatz der Profirecherche.<sup>8</sup> Nutzer unterscheiden nicht mehr zwischen einer Suche im OPAC oder der Recherche bei Google, eBay oder Amazon. Lewandowski beschreibt diesen Umstand wie folgt: „Suchmaschinen „erziehen“ Nutzer zu „schlechtem“ Rechercheverhalten.“<sup>9</sup> Der Versuch, das Suchverhalten der Nutzer aber den Katalogen anzupassen, ist nicht erfolgsversprechend, da freiwillige Nutzerschulungen selten in Anspruch genommen und Hilfetexte nicht gelesen werden. Das Desinteresse an Nutzerschulungen besteht nicht nur aus einem Mangel an Motivation, sondern scheitert auch daran, dass viele Nutzer denken, sie wüssten bereits wie man eine Suchmaschine richtig bedient.<sup>10</sup>

Die größten Schwächen der seit den 1990er Jahren kaum weiterentwickelten Kataloge bestehen laut Wiesenmüller<sup>11</sup> in der Überforderung der Nutzer, eine sinnvolle Eingabe in einem leerem Suchfeld zu formulieren. Lewandowski hingegen meint, dass „[d]ie meisten Nutzer [...] nicht willens [sind], bei der Formulierung ihres Suchziels allzu viel kognitive und zeitliche Energie aufzuwenden.“<sup>12</sup>

Die Problematik der Booleschen Operatoren – eine automatische Verknüpfung zweier oder mehrere Eingaben mit UND – führt häufig zu Null-Treffer-

---

<sup>5</sup> Vgl. DRAUSZ, PLIENINGER 2010 – Nutzerwünsche, S. 40.

<sup>6</sup> Vgl. MARKEY 2007 – End-User Searching, S. 1071.

<sup>7</sup> Vgl. WIESENMÜLLER 2008 – Neue Trends, S. 27.

<sup>8</sup> Vgl. LEWANDOWSKI 2009 – Suchmaschinen vs. OPAC.

<sup>9</sup> LEWANDOWSKI 2009 – Wie Suchmaschinen.

<sup>10</sup> Vgl. MARKEY 2007 – End-User Searching, S. 1078.

<sup>11</sup> Vgl. WIESENMÜLLER 2010 – Daten härter, S. 48.

<sup>12</sup> Vgl. Machill et al. 2003, zitiert nach LEWANDOWSKI 2009 – Suchmaschinen vs. OPAC.

Ergebnissen, die keinerlei Hinweise auf den Fehler oder weiterführende Hilfen anbieten. Es besteht keine Möglichkeit, die Trefferlisten nach bestimmten Kriterien zu sortieren oder weiter einzuschränken, obwohl Nutzer durch Suchmaschinentechnologie daran gewöhnt sind, dass die Suchergebnisse nach Bedeutung angeordnet werden. Kaum ein Nutzer betrachtet mehr als ein oder zwei Trefferlisten; 80% der Nutzer sehen sich lediglich die erste Trefferliste an.<sup>13</sup> Weiterhin bestehen kaum Möglichkeiten, die gefunden Informationen weiterzuverarbeiten, zu speichern oder zu ergänzen. Zudem sind herkömmlichen OPACs „Monolithen“ mit nahezu undurchdringlichen Wänden: Verknüpfungen zu anderen Anwendungen, und damit eine Nachnutzung von OPAC-Daten, sind ebenso schwierig wie umgekehrt das Einbringen von Informationen durch Nutzer.<sup>14</sup>

---

<sup>13</sup> LEWANDOWSKI 2009 – Suchmaschinen vs. OPAC.

<sup>14</sup> Vgl. WIESENMÜLLER 2008 – Der OPAC der Zukunft.

### 3. Die Idee des Katalog 2.0

OPAC 2.0,<sup>15</sup> Katalog 2.0,<sup>16</sup> Bibliothekskatalog 2.0, Rechercheplattform,<sup>17</sup> oder Next Generation Catalog,<sup>18</sup> sind nur einige der Bezeichnungen, die für einen Katalog der Zukunft<sup>19</sup> oder vielmehr zeitgemäßen Katalog<sup>20</sup> in der Literatur Verwendung finden.

Woher stammt aber dieses babylonische Sprachwirrwarr, wenn doch all diese Begriffe einen schlauerem, schönerem und offenerem Bibliothekskatalog<sup>21</sup> beschreiben, oder wie es Drauz und Plieninger 2010 formulieren:

“Letztlich aber handelt es sich als Leitfrage stets darum, den Bedürfnissen der Benutzer zu entsprechen, die Handhabung selbsterklärender, die Information reicher, die Weiterführung der Recherche mit anderen Mitteln einfacher zu machen.“?

Was aber genau zeichnet einen dieser Kataloge aus? Im Folgenden wird der Begriff Katalog 2.0 als Arbeitsgrundlage gewählt.

#### Prinzipien der Bibliothek 2.0

Um diese Frage beantworten zu können müssen, zuerst die Prinzipien und Forderungen, die im Rahmen des Begriffs Bibliothek 2.0 diskutiert, entwickelt und gestellt worden sind, berücksichtigt werden. Danowski und Heller nennen in ihrem Artikel „Bibliothek 2.0: Die Zukunft der Bibliothek?“<sup>22</sup> folgende grundlegende Prinzipien.

- OPAC + Browser + Web-2.0-Eigenschaften + Offenheit für Verbindungen zu Anwendungen Dritter = OPAC 2.0
- Bibliotheksbenutzer an der Gestaltung und an der Implementierung von Dienstleistungen teilhaben lassen

---

<sup>15</sup> Vgl. STEINER 2007 – OPAC 2.0.

<sup>16</sup> HELLER 2009 – Was ist ein Katalog 2.0.

<sup>17</sup> CHRISTENSEN 2007 – Über das Projekt.

<sup>18</sup> Vgl. WIESENMÜLLER 2008 – Neue Trends, S.28.

<sup>19</sup> Schwerpunkt in BuB 62(2010) Nr. 1.

<sup>20</sup> STABENAU 2010 – zeitgemäße Kataloge.

<sup>21</sup> CHRISTENSEN 2007 – Katalog 2.0.

<sup>22</sup> DANOWSKI, HELLER 2006 – Bibliothek 2.0, S. 1261ff.



- Bibliotheksbenutzer sollten in der Lage sein, die zur Verfügung gestellten Dienstleistungen zu benutzen und auf ihre individuellen Bedürfnisse zuzuschneiden
- Offenheit: Bibliothek 2.0 ist kein geschlossenes Konzept
- Permanente Verbesserung anstatt Upgrade-Zyklen („perpetual beta“)
- Kopieren und Integrieren von Programmen und Ideen Dritter in die Bibliotheksdienstleistungen
- Dienstleistungen ständig überprüfen, verbessern und dazu bereit zu sein, diese jederzeit durch neue, bessere Dienstleistungen zu ersetzen

Betrachtet man diese Prinzipien genauer, so erscheint es, dass nur die These „OPAC + Browser + Web-2.0-Eigenschaften + Offenheit für Verbindungen zu Anwendungen Dritter = OPAC 2.0“ einen informationstechnischen Hintergrund hat; alle anderen Forderungen beschreiben eine veränderte Kultur der bibliothekarischen Arbeit. Aber geht es bei der Forderung nach Web-2.0-Eigenschaften nur um den Einsatz von neuen Informationstechnologien und Kommunikationswegen, wie zum Beispiel Blogs und Wikis?

Betrachtet man die sieben von Tim O'Reilly postulierten Prinzipien, die die Bedeutung des Begriffs Web 2.0 geprägt haben,<sup>23</sup> an die sich eben auch die oben genannten Prinzipien der Bibliothek 2.0 anlehnen, ergibt sich eine andere Sicht. So haben z.B. die Forderungen nach „Das Web als Plattform“<sup>24</sup> und „Nutzung kollektiver Intelligenz“<sup>25</sup> ebenfalls eher einen sozialen als einen informationstechnischen Hintergrund.

Der Begriff Web 2.0 beschreibt nicht nur eine Veränderung der technischen Möglichkeiten, er beschreibt auch eine Veränderung des Nutzerverhaltens, die dadurch veränderten Erwartungshorizonte an Internetangebote und die veränderte Rolle des Nutzers, der nicht nur Konsument sondern auch Produzent von Information ist („user-generated content“).

---

<sup>23</sup> Vgl. O'REILLY 2005 – What Is Web 2.0.

<sup>24</sup> O'REILLY 2008 – Was ist Web 2.0.

<sup>25</sup> Ebd.

Danowski und Heller schreiben schon 2007 eben nicht von Web 2.0-Anwendungen, sondern von Web 2.0-Eigenschaften, die einen OPAC zu einem Katalog 2.0 machen.

Zu betonen ist hier, dass der Begriff Bibliothek 2.0 nicht nur den Einsatz von Web 2.0 Anwendungen wie Wikis, Blogs, RSS oder Social-Media Anwendungen wie Twitter oder Facebook beschreibt, sondern vielmehr eine veränderte Kultur der bibliothekarischen Arbeit, die nicht nur bestandsorientiert sondern vor allem auch nutzer- und bedürfnisorientiert ist.<sup>26</sup>

„Zu oft wird Information mit Informationstechnik gleichgesetzt. Bezeichnend ist in diesem Zusammenhang das vor einem Jahr viel diskutierte „Librarian 2.0 Manifesto“ von Laura Cohen. Hier ist nicht die Rede von neuer Informationstechnik, sondern von einer neuen Unternehmenskultur, die auf Kundenorientierung, Kreativität und Vertrauen setzt.“<sup>27</sup>

### **3.1. Anwendung der Prinzipien**

Ein Katalog 2.0 ist also mehr als eine Ansammlung von Funktionen, die durch informationstechnische Innovationen möglich sind. Bei der Entwicklung eines Katalogs 2.0 sollten auch die wesentlichen Prinzipien der Bibliothek 2.0 berücksichtigt werden. Wie können nun die Prinzipien der Bibliothek 2.0 auf einen Katalog 2.0 umgesetzt werden, auch wenn sich diese nur zum Teil und auch nicht eins zu eins übertragen lassen bzw. überschneiden?<sup>28</sup>

#### **OPAC + Browser + Web-2.0-Eigenschaften + Offenheit für Verbindungen zu Anwendungen Dritter = OPAC 2.0**

Vorangestellt sei, dass ein Katalog 2.0 mit Web 2.0- Eigenschaften natürlich auch die Konventionen des „Web 1.0“ erfüllt. Das bedeutet, dass jedes Katalogisat eine eindeutige und dauerhafte URL besitzt. Diese URLs werden auch als Permalink bezeichnet. So können diese per Mail verschickt werden, sind für Suchmaschinen

<sup>26</sup> Vgl. DRAUSZ, PLIENINGER 2010 – Nutzerwünsche S.40.

<sup>27</sup> Vgl. HOBHOM 2007 – Bibliothekswissenschaft 2.0.

<sup>28</sup> So bezieht sich beispielweise die Forderung, Bibliotheksbenutzer an der Gestaltung durch Usability-Tests teilhabenzulassen, auch auf die Forderung der Offenheit von Bibliotheken.

auffindbar und können als Bookmark gespeichert werden<sup>29</sup>. Dies ist ein seit mehr als 15 Jahren gängiger Standard für Webseiten. Permalinks werden in deutschen Bibliotheken so gut wie gar nicht eingesetzt.<sup>30</sup>

Die Forderung nach Offenheit für Verbindungen zu Anwendungen Dritter basiert vor allem auf den fehlenden Schnittstellen der heute im Einsatz befindlichen monolithischen Kataloge.<sup>31</sup> Weder Anwendungen Dritter können dynamisch integriert werden, noch können externe Anwendungen Dritter den Katalog nutzen oder abrufen.

„Softwarehersteller, Verbundzentralen und Bibliotheken sind daher gefordert, ihre Systeme mit Schnittstellen auszustatten, die einen prozessorientierten Zugang zu den bereitgestellten Diensten über eine Portalplattform ermöglichen.“<sup>32</sup>

Nach Sicht der Verfasser unterscheidet sich O'Reillys Definition von Web 2.0, nach der nur eine oder mehrere der sieben Kernkompetenzen und der damit verbundenen Umsetzungen erfüllt sein müssen, um eine erfolgreiche Web 2.0-Anwendung zu entwickeln, von der der Forderung, Web-2.0 Eigenschaften in den Katalog zu integrieren.<sup>33</sup> So zeigen sich doch eben diese Web 2.0- Eigenschaften vielmehr in den Forderungen nach Prinzipien des Begriffs Bibliothek 2.0, der sich an die Prinzipien des Web 2.0 anlehnt.<sup>34</sup> Im Unterschied zu O'Reilly bedarf es aber nicht der Berücksichtigung einer oder mehrere Prinzipien, sondern aller der folgenden Prinzipien der Bibliothek 2.0 im Bezug auf die Entwicklung eines Katalogs 2.0.

## **Teilhabe**

Der Forderung, Bibliotheksbenutzer an der Gestaltung und an der Implementierung von Dienstleistungen teilhaben zu lassen, kann sowohl mit qualitativen oder auch mit quantitativen Methoden der Nutzerforschung entsprochen werden. Zu den quantitativen Methoden der Nutzerforschung zählen der Einsatz von Fragebögen

<sup>29</sup> Vgl. HELLER 2009 – Was ist ein Katalog 2.0.

<sup>30</sup> Vgl. KINSTER 2010 – Kommentar zu Kataloge auf dem Weg ins Internet.

<sup>31</sup> Vgl. WIESENMÜLLER 2008 – Neue Trends, S. 28.

<sup>32</sup> KOSTÄDT 2008 – Der Weg in die Zukunft, S. 113.

<sup>33</sup> Eine deutsche Übersetzung des Artikels „Design Patterns and Business Models for the Next Generation of Software“ von O'Reilly, der die sieben Prinzipien der Idee des Web 2.0 erläutert, ist abrufbar unter [http://www.oreilly.de/artikel/web20\\_trans.html](http://www.oreilly.de/artikel/web20_trans.html).

<sup>34</sup> Eine umfassende Beschreibung der Charakteristika der Eigenschaften von Web 2.0 findet sich in Steiner 2007 - OPAC 2.0.

oder aber auch die Auswertung von Logfiles. An qualitativen Methoden eignen sich Tiefeninterviews oder Fokusgruppen, deren Ergebnisse mit Hilfe von Usability-Tests überprüft werden können.<sup>35</sup> Aber auch ethnographische Verfahren wie Cultural Probes sind denkbar, in denen die Teilnehmer gebeten werden, ihr Rechercheverhalten mittels Tagebüchern, per Foto bzw. Video- oder Tonaufzeichnung zu dokumentieren. Cultural Probes dienen dazu, genauere Erkenntnisse und Einsichten über die Motivation, die Gefühle und eventuell vorhandene latente Bedürfnisse zu erzielen. Konkrete Fragestellungen lassen sich nicht beantworten, aber wie schon die Übersetzung des Begriffs Probes zeigt, der sowohl Probe aber auch Sonde oder Messfühler bedeuten kann, ist dies eine Methode, mit der wirkliche Hintergrundinformationen über eine Zielgruppe gewonnen werden können.<sup>36</sup> Eine erste, auch mit Cultural Probes erstellte Studie, die sich auch mit dem Katalog der Zukunft beschäftigt, ist die aus Dänemark stammende Studie „The hybrid library: from the users’ perspective“.<sup>37</sup>

**Bibliotheksbenutzer sollten in der Lage sein, die zur Verfügung gestellten Dienstleistungen zu benutzen und auf ihre individuellen Bedürfnisse zuzuschneiden**

Berücksichtigt man die im Kapitel „Zur Notwendigkeit zeitgemäßer OPACs“ beschriebenen Erkenntnisse, ergibt sich die zwingende Notwendigkeit des Einsatzes von aktueller Suchtechnologie in einem Katalog 2.0, um die Nutzer von Bibliothekskatalogen wirklich dazu zu befähigen, die zur Verfügung gestellten Angebote zu nutzen. Wiesenmüller fordert in ihrem Vortrag am 10. November 2008 auf einer Fortbildungsveranstaltung der Arbeitsgemeinschaft für juristisches Bibliotheks- und Dokumentationswesen (AjBD) die Integration von Suchmaschinentechologie, mit denen beispielsweise Stemming oder eine Rechtschreibkorrektur möglich ist.<sup>38</sup> Auch Christensen zählt Rechtschreibprüfung und Stemming zu den Eigenschaften eines Kataloges 2.0.<sup>39</sup>

Die eingangs erwähnte Problematik der Verwendung des Begriffs Katalog 2.0 und die daraus resultierende Begriffsvielfalt zeigt sich hier das erste Mal: Wiesenmüller

<sup>35</sup> Vgl. CHRISTENSEN 2009 – Partizipative Entwicklung, S. 528.

<sup>36</sup> Vgl. BRANDES, ERLHOFF ET AL. 2009 – Designtheorie, S. 168.

<sup>37</sup> Vgl. CHRISTENSEN 2009 – Partizipative Entwicklung, S. 528.

<sup>38</sup> Vgl. WIESENMÜLLER 2008 – Der OPAC der Zukunft.

<sup>39</sup> Vgl. CHRISTENSEN 2007 – Katalog 2.0.

weist auf die Gefahr hin, dass die Idee eines Kataloges 2.0 zu sehr auf Web 2.0 reduziert wird. Die Idee, Suchmaschinentechologie in Bibliothekskatalogen einzusetzen, ist älter als die Idee des Katalogs 2.0. In der Diskussion mit Professoren, Kommilitonen und Bibliothekaren während des Studiums über den Begriff des Kataloges 2.0, wurden die Verfasser oft mit dem Argument konfrontiert, dass dies „alles schon lange bekannt“ sei.

Schon 1994 forderte Schulz in einem Vortrag auf der Fortbildungsveranstaltung für Bibliothekare anlässlich der 18. Jahrestagung der Gesellschaft für Klassifikation e.V. in Oldenburg den Einsatz von Relevance Ranking, Stemming und einer fehlertoleranten Suche in Bibliothekskatalogen.<sup>40</sup>

Die Forderung, dass Bibliotheksbenutzer in der Lage sein sollten, die zur Verfügung gestellten Dienstleistungen auf ihre individuellen Bedürfnisse zuzuschneiden bedeutet aber auch, dass ein Kataloge 2.0 plattformübergreifend aufrufbar sein müssen. Daraus ergibt sich, dass Kataloge nicht nur über einen bestimmten Browser eines Desktopsystems, sondern auch über Smartphones, Tablet-Computer und andere technische [i]Endgeräte aufrufbar sein sollten.<sup>41</sup> Vor allem aber ist eine Personalisierung der Kataloge notwendig, um eine Interaktivität zwischen Katalog und Nutzer zu ermöglichen.

### **Perpetual Beta**

Das Prinzip der permanenten Verbesserung anstatt von Upgrade-Zyklen ist nicht nur angelehnt an die Prinzipien des Web 2.0 bzw. der Bibliothek 2.0, sondern kann direkt auf die Programmierung und Weiterentwicklung, aber auch auf die Pflege eines Kataloges übertragen werden. Der Begriff Perpetual Beta beschreibt die fortwährende Ent- und Weiterentwicklung eines Webangebotes. Software befindet sich also dauerhaft in einer Art Testphase. Der Perpetual Beta-Gedanke spricht vor allem auch eher Communities an, in denen Software gemeinsam und fortwährend weiterentwickelt und verbessert wird, als geschlossene Programmerteams innerhalb eines Unternehmens.<sup>42</sup>

---

<sup>40</sup> Vgl. SCHULZ 1994 – Was wir über OPAC-Nutzer wissen.

<sup>41</sup> Vgl. HELLER 2009 – Was ist ein Katalog 2.0.

<sup>42</sup> Vgl. KOCH, RICHTER 2009 – Enterprise 2.0, S. 19.

Auch Perpetual Beta zielt auf die Teilhabe der Nutzer ab. Jede einzelne Weiterentwicklung des Angebots wird direkt in das laufende System integriert, sodass der Nutzer jederzeit den Entwicklungsprozess nachvollziehen und neue Funktionen nutzen kann. Daher sind Nutzer eher bereit, eventuelle Schwächen zu akzeptieren, aber eben auch, frühzeitig zu bestimmten Entwicklungen und Ideen Feedback zu geben. So ergibt sich für die Bibliotheken die Chance, eventuelle Fehlentwicklungen zu vermeiden und diesen frühzeitig gegenzusteuern.

## **Offenheit**

Unter der Prämisse „Offenheit: Bibliothek 2.0 ist kein geschlossenes Konzept“ sollten Kataloge 2.0 die Möglichkeit bieten, die in ihnen enthaltenen Daten zu exportieren, auch in größeren Mengen herunterzuladen und mit Hilfe dieser Daten neue Anwendungen zu konzipieren und zu entwickeln.<sup>43</sup> „Die Prämisse für diese Art der Innovation lautet: Benutzern des Katalogs werden neue, nützliche und überraschende Anwendungen für die darin enthaltenen Daten einfallen, die weder den Entwicklern noch den Betreibern des Katalogs je in den Sinn gekommen wären.“<sup>44</sup> Das Beispiel der Universitäts- und Staatsbibliothek Köln, die in Kooperation mit dem Hochschulbibliothekszenrum des Landes Nordrhein-Westfalen ihre Metadaten der Öffentlichkeit zur Verfügung stellt,<sup>45</sup> zeigt dass dieses sowohl rechtlich als auch technisch möglich ist. Die Freigabe erfolgt, wie auch Heller fordert, unter der CC0-Lizenz.<sup>46</sup>

Zudem meint Offenheit aber auch die Offenheit für Verbindungen zu Anwendungen Dritter, durch die sich gefundene Titel beispielsweise in Literaturverwaltungsprogramme exportieren lassen.

Die Forderung der Offenheit von Bibliothek 2.0 führt aber auch zum folgenden Abschnitt, der die Übernahme von Ideen und Entwicklungen Dritter thematisiert. Ideen, Konzepte und bereits entwickelte Programme sollten veröffentlicht werden. Anderen Institutionen und interessierten Personen sollte jederzeit auch der Einblick

---

<sup>43</sup> Vgl. HELLER 2009 – Was ist ein Katalog 2.0.

<sup>44</sup> Ebd.

<sup>45</sup> Vgl. FLIMM 2010 – Offene bibliographische Daten.

<sup>46</sup> Informationen über CC0 finden sich auf <http://creativecommons.org/publicdomain/zero/1.0/deed.de>.

in laufende Projekte gewährt werden, um vor allem die Nachnutzung von bereits entwickelten Lösungen zu ermöglichen.

## **Übernahme**

Einer der entscheidendsten und wichtigsten Punkte bei der Entwicklung eines Kataloges 2.0 unter der Berücksichtigung der Prinzipien der Bibliothek 2.0 ist die Bereitschaft, Programme und Ideen Dritter in die Bibliotheksdienstleistungen zu kopieren und zu integrieren. Dazu bedarf es mehr als das passive permanente Screening der relevanten Blogs, Mailinglisten und Veröffentlichungen. Vielmehr bedarf der aktiven Partizipation, beispielsweise durch Teilnahme an Konferenzen bzw. Unkonferenzen,<sup>47</sup> der Kontaktaufnahme zu Personen, die bereits an ähnlichen Projekten arbeiten, und eben auch der Offenheit des eigenen Projekts. Der Bedeutung des sogenannten „Netzwerkens“, also der Aufbau eines Personenkreises mit gleichen Interessen und Absichten, kann an dieser Stelle nicht genug betont werden. Viele Informationen, über die Anwendung bestimmter technischer Entwicklungen, sind nicht oder noch nicht veröffentlicht, sondern können nur über die persönliche Kontaktaufnahme gewonnen werden.<sup>48</sup>

Selbstverständlich meint aber die Forderung nach der Übernahme von Ideen und Programmen Dritter nicht nur solche Ideen, die im bibliothekarischen Kontext entstanden sind. Darüber hinaus geht es auch darum, sich der Konkurrenz von Web-Suchmaschinen, Social-Bookmarking-Diensten und auch Verkaufsplattformen zu stellen, deren Ideen zu überprüfen und falls sinnvoll und möglich zu übernehmen.

## **Dienstleistungen ständig verbessern und dazu bereit zu sein, diese jederzeit durch neue, bessere Dienstleistungen zu ersetzen**

Diese Forderung bringt nun die Beantwortung der zu Beginn gestellten Frage nach den Ursachen der Begriffsvielfalt für den Katalog 2.0. Auch die bereits erwähnte Problematik des Begriffs Katalog 2.0, der eine (zu) starke Ausrichtung auf Web 2.0-Anwendungen implizieren könnte, erklärt sich hier.

---

<sup>47</sup> Unkonferenz meint hier Veranstaltungen, in denen es keine Trennung zwischen Vortragenden und Zuhörern gibt, z.B. das seit 2007 jährlich stattfindende Bibcamp.

<sup>48</sup> Vgl. auch die Kapitel „Umsetzung der Methodik“ und „Schlussbetrachtung“.

Setzt man sich das Ziel, die eigenen Dienstleistungen und damit natürlich auch das Herzstück der bibliothekarischen Arbeit, den Katalog, ständig zu überprüfen und zu verbessern, kann es keine endgültige Definition des Begriffs Katalog 2.0 oder gar eine Beschreibung eines bestimmten Funktionsumfang und zwingend erforderlicher Elemente geben; außer eben der Definition, dass ein Katalog 2.0 ein Katalog ist, der unter Berücksichtigung der hier vorgestellten Prinzipien ständig überprüft und weiterentwickelt wird.

Aus dieser Definition ergibt sich, dass der Begriff des Kataloges 2.0 keine zu einem bestimmten Zeitpunkt aktuelle Version beschreibt, sondern vielmehr beschreibt, wie man einen Katalog immer schlauer, schöneren und offener macht.

Katalog 2.0 heißt demnach, sich auf aktuelle und bereits länger bekannte Erkenntnisse aus der OPAC-Benutzerforschung, den Bibliothekswissenschaften und den in der bibliothekarischen Praxis gesammelten Erkenntnissen zu beziehen und mit den derzeit technisch verfügbaren Möglichkeiten und unter Verwendung der Prinzipien der Bibliothek 2.0 umzusetzen.

### **3.2. Funktionen eines Katalog 2.0**

Wertet man nun die verschiedenen Ideen, Konzeptionen und Erkenntnisse aus, ergibt sich folgender Ideenkatalog ohne auf Anspruch der Vollständigkeit, der als Basis für einen Katalog 2.0 im Jahr 2010 dienen kann:

- Relevance Ranking
  - Mit Hilfe der der Metadaten
  - Mit gemischten Medienarten
  - Auf Basis der Nutzungsdaten
  - Mit Hilfe einer Volltextindexierung
- Browsing
  - Einstieg schon auf der Startseite
  - Facettiertes Browsen durch Drilldowns
  - Browsing unter Verwendung der Metadaten



- Browsing auf Basis eines Thesaurus
- Rechtschreibfunktionen
  - Hinweis auf eine mögliche richtige Schreibweise
  - Automatische Rechtschreikorrektur
  - Autovervollständigung
- Stemming, Boosting und unscharfe Suche
- Retrieval von Datensätzen, die nicht alle Suchtermini enthalten
- Empfehlungsdienste
  - Auf Basis von Nutzungsdaten
  - Auf Basis von Bewertungen
  - Auf Basis von Metadaten
- FRBRisierung (Functional Requirements for Bibliographic Records)<sup>49</sup>
- Personalisierung
  - Listen privat und öffentlich zu erstellen
  - User-generated content (Tags, Kommentare, Rezensionen, Bewertungen)
  - RSS-Feeds
  - Treffer und Suchergebnisse per E-Mail oder SMS verschickbar
- Verlinkung oder Anzeige weiterer Informationen (z.B. Wikipedia)
- Kataloganreicherungen durch Cover, Rezensionen und Inhaltsverzeichnisse
- Integration von audio-visuellen Inhalten
- Zitierfunktion
- Offene und standardisierte Schnittstellen für Im- und Export
- Permalinks

---

<sup>49</sup> Meint hier die Anzeige des Werkes in der Trefferliste. Vgl. DE REUS 2010 – FRBR.

- Suchraumerweiterung auf alle Angebote der Bibliothek („web-scale-index-search“)
- Katalog ist Teil der einzigen Website der Bibliothek
- Mobile Nutzung

Welche dieser Ideen, Eigenschaften und Funktionen für eine einzelne Bibliothek sinnvoll und von den Nutzern gewünscht werden, kann nur wieder über die Beteiligung der Nutzer an der Entwicklung des Katalogs geschehen oder zumindest aus der Auswertung von Nutzerstudien vergleichbarer Zielgruppen. Selbstverständlich ist die Umsetzung an die technischen, finanziellen und personellen Möglichkeiten einer Bibliothek gekoppelt.

### 3.3. Problematik und Problembewusstsein

Anzumerken ist, dass die Notwendigkeit von benutzergerechten, attraktiven und zukunftsfähigen Katalogen eigentlich unbestritten sein sollte, auch wenn vor allem unter Bibliothekaren noch eine unzureichende Wahrnehmung der Probleme herrscht.<sup>50</sup> „Eine solche Haltung scheint bei manchen Diskussionen auf bibliothekarischen Mailinglisten immer noch durch, die korrekte Anwendung der RAK wird mit Benutzerfreundlichkeit gleichgesetzt. „Usability – die Anpassung des Services an die Bedürfnisse der jeweiligen Zielgruppe – ist aber mittlerweile die Messlatte für zukünftige Erweiterungen des Kataloges.“<sup>51</sup> Obwohl viele Anforderungen an einen benutzerfreundlichen Katalog, basierend auf der Nutzerforschung, bereits seit spätestens 1996 bekannt sind, erscheint selbst im Jahr 2010 noch ein Artikel in der BuB über zukunftsfähige Kataloge, mit dem auch „bibliothekarische Vorbehalte beseitigt werden“<sup>52</sup> sollen.

Natürlich war und ist zum Beispiel die Diskussion um Kataloganreicherungen mit Covern, Rezensionen oder anderen externen Inhalten sinnvoll und wichtig, da sie die den Katalog verbessern und dem Nutzer helfen, die Relevanz eines Treffers besser zu beurteilen. Die Kernaufgabe eines Kataloges bleibt aber die Suche, und der Kern jeder Suchanwendung ist die Relevanz der angezeigten Treffer, also das Relevance Ranking.<sup>53</sup>

---

<sup>50</sup> Vgl. WIESENMÜLLER 2008 – Neue Trends, S.28.

<sup>51</sup> DRAUSZ, PLIENINGER 2010 – Nutzerwünsche, S. 40.

<sup>52</sup> Schwerpunkt in BuB 62 (2010) Nr. 1.

<sup>53</sup> Vgl. LEWANDOWSKI 2009 – Wie Suchmaschinen.

## 4. Zum Einsatz von Open-Source-Software im Kontext von Bibliothek 2.0

Der Begriff Open Source fasst eine Reihe von verschiedenen Lizenzen zusammen, bei denen der Quellcode einer Software frei zugänglich ist. Die bloße Offenlegung des Quellcodes alleine macht jedoch noch keine Open-Source-Software (OSS) aus. Entscheidend ist, ob eine Software bzw. deren Lizenz die Open Source Definition<sup>54</sup> der Open Source Initiative<sup>55</sup> erfüllt.<sup>56</sup> Allen Lizenzen gemein ist, dass der Quellcode einer Software nicht nur eingesehen, sondern für beliebige Zwecke benutzt, verändert und Dritten zugänglich gemacht werden kann.

Die bedeutendste Open-Source-Lizenz ist die von Richard Stallmann entwickelte GNU General Public Licence (GPL), unter der die in dieser Arbeit verwendete Katalogoberfläche VuFind, die Document Availability Information API (DAIA) sowie der Linux-Kernel steht. Durch die Verwendung der GPL wird sichergestellt, dass eine Software immer frei bleiben wird, da sämtliche Veränderungen und Weiterentwicklungen ebenfalls unter der GPL stehen müssen.<sup>57</sup> Frei ist hier im Sinne von Freiheit und nicht im Sinne von kostenlos gemeint, denn der Einsatz von OSS ist nicht automatisch gratis, da Ressourcen für eventuell erforderliche Anpassungen und Erweiterungen benötigt werden. Durch die Teilung des Entwicklungsaufwands entstehen jedoch Synergieeffekte, von denen alle Anwender profitieren können.

Open-Source-(Bibliotheks-)Software geht Hand in Hand mit den Prinzipien der Bibliothek 2.0, auch sie ist per Definition „kein geschlossenes Konzept“. Da OSS für die eigenen Bedürfnisse zugeschnitten, erweitert und weitergegeben werden kann, können Nutzer an der Weiterentwicklung mitwirken. Die Trennung zwischen Nutzern und Entwicklern wird aufgeweicht. Durch die Offenheit des Quellcodes - und damit der Offenheit der Programmierleistung – besteht jederzeit die Möglichkeit, die Qualität einer Software zu überprüfen. „Open Source ist unter [a]nderem deshalb oft kommerziellen Produkten überlegen, weil offene Systeme

---

<sup>54</sup> <http://www.opensource.org/docs/osd>.

<sup>55</sup> <http://www.opensource.org/>.

<sup>56</sup> Vgl. RENNER 2005 – Open Source Software 2005, S. 12.

<sup>57</sup> Vgl. ebd., S. 20.

nicht so einfach ihren Dreck unter den Teppich kehren können.“<sup>58</sup> Es erklärt sich von selbst, dass OSS ideale Voraussetzungen bietet, um Programme und Ideen Dritter in eigene Dienstleistungen zu integrieren. Die Realisierung eines Katalogs unter Berücksichtigung aller Prinzipien der Bibliothek 2.0 ist ohne die Grundlage Open Source daher nicht möglich. Ein häufiges Problem beim Einsatz von OSS stellt die mangelnde Interoperabilität mit kommerzieller Software dar. Dies ist aber nicht als Nachteil von OSS, sondern als Nachteil von kommerzieller bzw. Closed-Source-Software zu bewerten, deren Schnittstellen (meist) nicht offengelegt sind.<sup>59</sup> Zu beachten ist, dass aber auch alternative Katalogoberflächen von kommerziellen Anbietern nicht out of the box funktionieren.<sup>60</sup>

Laut Kinstler ist „[m]ake or buy [...] eigentlich gar nicht die Frage. Denn selbst bei “buy” bleibt immer noch genug “make” übrig, wenn man am Ende etwas Brauchbares haben will. Ohne Anpassungen an lokale Gegebenheiten funktioniert im Bibliothekswesen doch keine Software von der Stange. Die Verbundzentrale macht eigentlich zum großen Teil nichts anderes als Anpassungen an gelieferter Software (“Systemintegration”). Und irgendwo stößt man [bei kommerziellen Produkten] dabei immer an Grenzen.“<sup>61</sup>

Da die Entwicklung von OSS nicht an einen bestimmten Hersteller gebunden ist, durch den eine Gewährleistung oder Garantie erfolgen könnte, liegt es an der Community, die Entwicklung ihrer Programme voranzutreiben. Für Bibliotheken birgt dies aber auch eine große Chance, da sie so „Betrieb, Anpassung und Entwicklung ihrer “Informationstechnik”(!) als Teil ihres Kerngeschäfts begreifen und entsprechend professionell(!) betreiben.“<sup>62</sup> können.

---

<sup>58</sup> VOSS 2010 – Qualitätssicherung.

<sup>59</sup> Vgl. RENNER 2005 – Open Source Software 2005, S. 175.

<sup>60</sup> Vgl. KINSTLER 2010 – VuFind.

<sup>61</sup> Vgl. KINSTLER 2010 – Kommentar zu make or buy.

<sup>62</sup> Ebd.

## 5. VuFind – ein Open-Source-Discovery-Interface

### 5.1. Technischer Hintergrund

VuFind war eines der ersten freien Open Discovery Interfaces unter der GPL und wird nach Ansicht von Breeding auch die dominierende Alternative zu proprietären Lösungen bleiben.<sup>63</sup> Open Discovery Interfaces sind keine vollständigen lokalen Bibliothekssysteme (LBS), sie wurden entwickelt, um an ein bereits bestehendes LBS anzudocken und das herkömmliche Kataloginterface zu ersetzen.<sup>64</sup> Daher stammt auch der nicht mehr gebräuchliche Name „next-generation library catalog“.<sup>65</sup>

VuFind kann aber vielmehr auch als zentrales Informations- und Rechercheangebot genutzt werden, da es alle als Text darstellbaren Daten durch die Nutzung von Solr/Lucene indexieren kann.<sup>66</sup> So beinhaltet zum Beispiel TUBFind, der auf VuFind basierende Katalog der Technischen Universität Hamburg-Harburg, nicht nur den Medienbestand, es wurden auch die Webseiten und der Blog der TUB indexiert und so gemeinsam durchsuchbar gemacht.<sup>67</sup> Auch verfügt VuFind über eine OAI-Schnittstelle, mit der Metadaten aus digitalen Repositorien integriert werden können.<sup>68</sup> „The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is a low-barrier mechanism for repository interoperability. Data Providers are repositories that expose structured metadata via OAI-PMH.“<sup>69</sup>

Ebenfalls unterstützt VuFind das Webservice-Protokoll Search/Retrieve via URL (SRU) mit dessen Hilfe es möglich ist, Anfragen an Datenbanken zu schicken und die dort gefundenen Treffer anzuzeigen. „SRU (Search / Retrieve via URL) ist ein standardisiertes Webservice-Protokoll, um Datenbanken im Internet abzufragen. Die Ergebnisse der Suche werden in einem definierten XML-Format zur Verfügung gestellt. „Das SRU-Protokoll ermöglicht die gezielte Suche mittels Suchindizes und Suchbegriffen und die Übernahme der entsprechenden Treffer in

---

<sup>63</sup> Vgl. BREEDING 2009 – Open Discovery Interfaces, S.40.

<sup>64</sup> Vgl. HOUSER 2009 – The VuFind implementation, S. 93 und Breeding 2009 – Open Discovery Interfaces 2009, S.40.

<sup>65</sup> Vgl. BREEDING 2010 – The State of the Art, S. 31.

<sup>66</sup> Vgl. KINSTLER 2010 – VuFind.

<sup>67</sup> Vgl. MARAHRENS 2010 – Änderung.

<sup>68</sup> Vgl. HO, KELLY ET AL. 2009 – Implementing VuFind, S.84.

<sup>69</sup> OPEN ARCHIVES INITIATIVE 2010 – Interoperability.

die eigene Umgebung.“<sup>70</sup> Weitere Informationen über SRU und die Nutzung durch VuFind finden sich in der VuFind-Dokumentation.<sup>71</sup>

Die vorliegende Arbeit beschäftigt sich aber primär mit der Nutzung von VuFind als Bibliothekskatalog, auch wenn mit VuFind mittlerweile Möglichkeiten eröffnet werden, die über eine reine Verwendung als Alternative zu den herkömmlichen im Einsatz befindlichen Katalogen hinaus gehen und mit der sich ein Open Discovery Interface verwirklicht lässt, dessen Ziel es ist „[...] to provide access to all aspects of library collections, not just those managed in the traditional library catalog, which is limited to the content managed by the integrated library system.“<sup>72</sup>

Auch wenn die Anzahl der Bibliotheken, die kommerzielle Lösungen nutzen noch weitaus höher ist, zeigt doch die steigende Anzahl an sehr unterschiedlichen und angesehen Institutionen, die Open Discovery Interfaces einsetzen, dass dies nicht nur ein momentaner Trend sein wird.<sup>73</sup>

Treiber, um die Bestands- und Verfügbarkeitsinformationen aus den lokalen Bibliothekssystemen auszulesen, sind bereits für folgende Systeme verfügbar:

- Aleph
- Evergreen
- Horizon
- Innovative
- Koha
- Unicorn
- Virtua
- Voyager
- XCNCIP
- PICA/DAIA<sup>74</sup>

---

<sup>70</sup> Vgl. HUDE 2010 - SRU.

<sup>71</sup> Vgl. KATZ 2009 - System Classes.

<sup>72</sup> Vgl. BREEDING 2010 - The State of the Art, S. 31.

<sup>73</sup> Vgl. BREEDING 2009 - Open Discovery Interfaces, S.40.

<sup>74</sup> Vgl. VUFIN DOKUMENTATION 2010 - PICA/DAIA.

Für bisher noch nicht unterstützte Bibliothekssysteme können eigene Treiber geschrieben werden. Weitere Informationen zur Programmierung von Treibern finden sich in der VuFind-Dokumentation.<sup>75</sup>

Die Suchmaschinentechologie auf Basis von Solr/Lucene, die in VuFind verwendet wird, ist auch bei vielen anderen kommerziellen und nicht-kommerziellen Lösungen im Einsatz. Durch Solr/Lucene können viele der in der Diskussion um den Katalog 2.0 entwickelten Ideen, aber auch viele seit langem bekannte Forderungen an einen Katalog, der dem Suchverhalten der Nutzer gerecht wird, einfach umgesetzt werden. Beispielsweise können Funktionen wie facetiertes Browsen, Relevance Ranking, Stemming, Vorschlagsfunktionen („Meinten Sie“) realisiert werden.

VuFind hat eine frei anpassbare Oberfläche, die es ermöglicht, das eigene Corporate Design einfach zu integrieren. Das Frontend ist hauptsächlich in PHP und HTML programmiert und kann den lokalen Bedürfnissen leicht angepasst werden. Eine MySQL-Datenbank wird genutzt, um Tags, Kommentare und Rezensionen zu speichern.

## 5.2. Serviceorientierte Architektur

Im Gegensatz zu den monolithischen „DinOPACs“<sup>76</sup> beruht die Softwarearchitektur von VuFind auf dem Prinzip der Serviceorientierten Architektur (SOA). SOA bezeichnet aber nicht den Einsatz einer konkreten Technologie, sondern das Konzept für die Strukturierung und Nutzung verteilter Funktionalität<sup>77</sup>, mit der über ein Netzwerk auf verschiedene (Web-)Services, die unabhängig voneinander betrieben werden, über eindeutig definierte Schnittstellen zugegriffen werden kann. Durch die Zusammenführung einzelner Module können so komplexe Anwendungen erstellt werden.<sup>78</sup>

---

<sup>75</sup> Vgl. VUFINDDOCUMENTATION 2010 - ILS Driver Support.

<sup>76</sup> Vgl. WIESENMÜLLER 2008 – Der OPAC der Zukunft.

<sup>77</sup> Vgl. Mac-Kenzie et al., zitiert nach WUNDER, GROSCHE 2009 – Verteilte Führungsinformationssysteme 2009 S. 48.

<sup>78</sup> Vgl. HAMANN 2006 – Dienstorientierte Architektur SOA, S. 8.



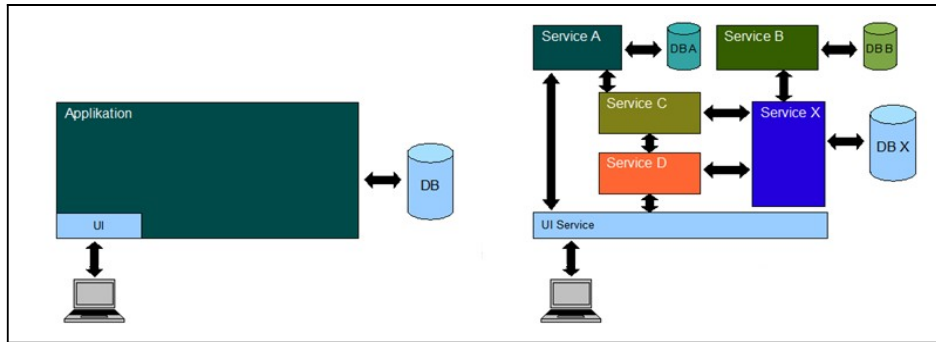


Abbildung 1: Monolithische Architektur vs. SOA-System

(GRINDLE 2010 – SOA 2010, S. 2 und 3)

Erst durch die Verwendung einer Serviceorientierten Architektur und dem daraus folgenden modularen Aufbau können in VuFind ohne großen Aufwand einzelne Funktionen unabhängig voneinander integriert, angepasst und gepflegt werden („Baukastenprinzip“).

VuFind benötigt folgende Systemsoftware:

- Apache HTTP Server 2.2 oder neuer
- PHP 5.1.4 (5.2.0 oder neuer empfohlen)
- MySQL 4.1 oder neuer
- Java J2SE JDK 1.4 oder neuer

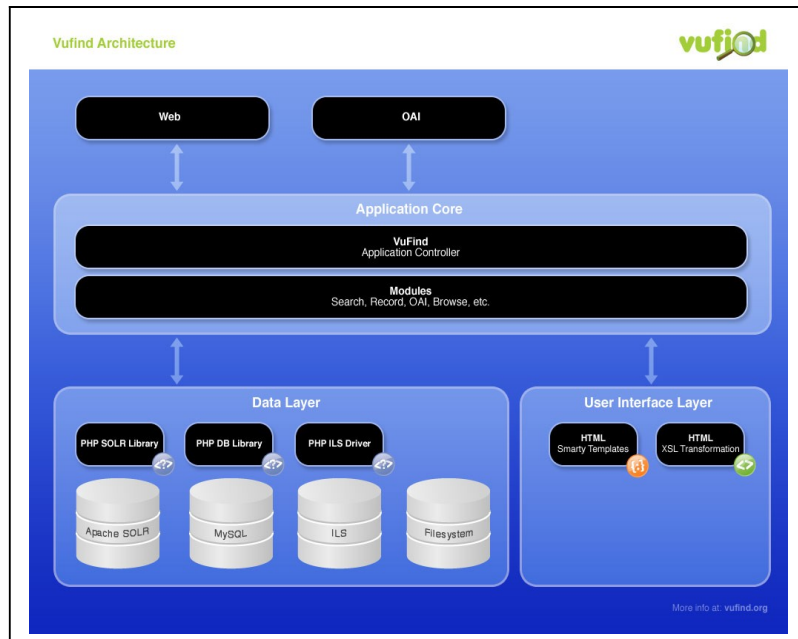


Abbildung 2: Überblick über die Software-Architektur von VuFind  
(VUFIND DOCUMENTATION 2010 - developer manual)

### 5.3. Entstehung und Anwendungsszenarium

VuFind wird an der Falvey Memorial Library der Villanova University (Pennsylvania, USA) und mit der Unterstützung einer stetig wachsenden internationalen Community entwickelt. So wurde zum Beispiel der Indexierungsprozess von Mitarbeitern der Bibliothek der University of Virginia bereits im Jahr 2007 soweit verbessert, dass eine Indexierung von 100000 Metadaten nicht mehr eine Stunde sondern nur noch zehn Minuten dauert.<sup>79</sup> Die Idee, VuFind zu programmieren, entstand während der Arbeit an der Digitalen Bibliothek der Falvey Memorial Library der Villanova University im Jahr 2006. Der erste Prototyp wurde auf der Code4Lib Conference im Februar 2007 noch unter dem Namen MyResearch Portal präsentiert. Im Mai 2008 ging die National Library of Australia als erste Bibliothek mit einem auf VuFind basierenden Katalog noch vor den eigentlichen Entwicklern von VuFind online. Am 18. August 2008 folgte der Verbundkatalog der Minnesota State Colleges and Universities und auch zwei Tage später folgte der Katalog der Bibliothek der Villanova University. Noch ist VuFind vor allem in englischsprachigen Ländern im Einsatz, aber es finden sich mittlerweile auch Projekte aus Belgien, Deutschland und auch aus

<sup>79</sup> Vgl. HOUSER 2009 – The VuFind implementation, S. 97.

Südamerika .Weltweit sind weit über 50 VuFind-Projekte, zumeist an Universitätsbibliotheken, im Einsatz bzw. in der Entwicklung.<sup>80</sup> Die „Suchkiste“ war das erste in Deutschland umgesetzte Projekt, das auf Basis von VuFind entwickelt wurde. Das durch die Deutsche Forschungsgemeinschaft (DFG) geförderte Projekt hat das Ziel, eine zentrale Recherche- und Serviceplattform für national lizenzierte Ressourcen zu sein. Zurzeit sind mehr als 20 Millionen Datensätze recherchierbar.<sup>81</sup> Dieses unter der Leitung des Gemeinsamen Bibliotheksverbunds entstehende Projekt unterstützt die weltweite Community der VuFind Nutzer.<sup>82</sup> Der erste VuFind-Katalog an einer deutschen Bibliothek ging unter dem Namen TUBFind an der Technischen Universität Hamburg Harburg am ersten April 2010 online. Am 15. Juli 2010 verließ VuFind mit der Veröffentlichung der Version 1.0 den beta-Status.

Ursprünglich wurde VuFind nur als Oberfläche für digitale Bibliotheken entwickelt.<sup>83</sup> Heute kann es allerdings auch als gemeinsame Portallösung für alle Arten von Bibliotheksressourcen genutzt werden. Zudem sind Lösungen für Artikeldatenbanken, Repositorien, Digitale Bibliotheken und andere Arten von Bibliothekskollektionen im Einsatz.<sup>84</sup>

---

<sup>80</sup> VUFIN D DOCUMENTATION 2010 – Installations.

<sup>81</sup> <http://finden.nationallizenzen.de/>.

<sup>82</sup> Vgl. KINSTLER 2010 – VuFind.

<sup>83</sup> Vgl. HOUSER 2009 – THE VUFIN IMPLEMENTATION, S. 94.

<sup>84</sup> Vgl. VUFIN D DOCUMENTATION 2010 – UBUNTU.

## 5.4. Funktionsumfang aus Nutzersicht

Ausgehend von der Demo-Version<sup>85</sup> von VuFind wurde als erster Schritt eine Aufnahme der in VuFind theoretisch zu verwirklichenden Elemente und Funktionen aus Sicht eines Anwenders erstellt. Bewusst wurde hier die Demo-Version gewählt, da viele sich bereits im Einsatz befindenden Versionen von VuFind stark modifiziert und mit eigenen Erweiterungen ergänzt wurden.<sup>86</sup>

### Startseite

Die Startseite von VuFind bietet, wie auch Google, einen einzigen Suchschlitz. Die Suche über alle Felder ist voreingestellt, kann aber auf Titel, Verfasser, Schlagwort, Signatur, ISSN/ISBN oder Tag eingeschränkt werden. Zusätzlich befindet sich auf der Startseite auch ein BrowsingEinstieg nach Klassifikation, Sprache und Format.

### Erweiterte Suche

Die erweiterte Suche ermöglicht die kombinierte Suche über mehrere Suchfelder mit Booleschen Operatoren. Voreingestellt sind drei Suchfelder mit der UND-Verknüpfung. Über ein Drop-Down-Menü kann zwischen UND, ODER und NICHT gewählt werden. Eine Kombination verschiedener Boolescher Operatoren kann durch die Hinzufügung von weiteren Suchmengen erfolgen. Auch hier besteht die Möglichkeit die Treffer zusätzlich über ein Auswahlmenü auf Klassifikation, Sprache und Format einzugrenzen. Über eine Checkbox kann auf Medien mit Illustrationen oder ohne Illustrationen eingeschränkt werden.

---

<sup>85</sup> <http://vufind.org/demo/>.

<sup>86</sup> Vgl. VUFIND DOCUMENTATION 2010 – Installations.

## **Trefferanzeige**

In der Trefferanzeige bietet VuFind dem Nutzer die Möglichkeit, die erzielten Treffer nach Relevanz, Datum ab- und aufsteigend, und Verfasser zu sortieren. Die Suche über alle Felder ist voreingestellt. Angezeigt werden die ersten 20 Treffer der Suche, die Anzahl der Treffer und die Suchdauer. Im Falle von orthographischen Fehlern schlägt die Rechtschreibkorrektur mögliche richtige Schreibweisen vor. Weiterhin schlägt VuFind dem Nutzer weitere zu seiner ursprünglichen Suche relevante Schlagwörter vor. Bei einzelnen Titeln innerhalb der Trefferliste werden die Informationen Titel, Autor, Verfügbarkeit, Signatur, Standort (bei nur einem Standort) und, falls vorhanden, ein Link zu einem Inhaltsverzeichnis angezeigt. Soweit Coverabbildungen zu einzelnen Titeln verfügbar sind, werden diese ebenfalls schon in der Trefferanzeige angezeigt. Einzelne Titel können ohne den Aufruf der Titelanzeige direkt den eigenen Favoriten hinzugefügt werden. Unter dem Punkt „Suchwerkzeuge“ besteht die Möglichkeit, die Trefferanzeige per E-Mail zu versenden oder auch auf dem eigenem Account zu speichern. Auch können einzelne Suchanfragen per RSS abonniert werden. RSS ist ein Akronym für Really Simple Syndication und ermöglicht es dem Nutzer, einzelne Suchanfragen über einen FeedReader abzurufen, ohne erneut den Katalog aufsuchen zu müssen. Mit Hilfe des facettierten Browsens lässt sich der Suche nach Institution, Signatur/Klassifikation, Verfasser, Sprache, Genre, Ära und Region einschränken. Mit Hilfe der Blätterfunktion kann sich der Nutzer die ersten 11 Trefferlisten oder auch die letzte Trefferliste anzeigen lassen. Durch die Auswahl des Autors können zusätzlich Informationen über den Autor aus Wikipedia abgerufen werden.

## **Titelanzeige**

In der Titelanzeige werden Titel, Autor, Format, Sprache, Verlag, Schlagworte und falls vorhanden Tags angezeigt: Über verschiedene Reiter können weitere Informationen über Exemplare, Beschreibung, Kommentare, Rezensionen, Ausschnitte und das Internformat angezeigt werden. Voreingestellt ist hier der Reiter „Exemplare“, der Auskunft über den Standort, die Signatur und die Verfügbarkeit liefert. Der Reiter „Beschreibung“ gibt Auskunft über die ISBN, Seitenzahl und eventuell vorhandene Illustrationen. Von anderen Nutzern erstellte

Kommentare können unter dem Reiter „Kommentare“ eingesehen werden. Rezensionen von Amazon und anderen Anbietern befinden sich unter dem Reiter „Rezensionen“. Der Reiter „Ausschnitte“ zeigt Textauszüge oder Abstracts an. Über eine Auswahl des Internformats kann sich der Nutzer das ursprüngliche Metadaten austauschformat anzeigen lassen. Über die Anzeige ähnliche Einträge zeigt VuFind andere zum Medium relevante Treffer an.

Die Titelanzeige bietet dem Nutzer folgende Funktionen: Es besteht die Möglichkeit den Titel in den Zitierformaten APA<sup>87</sup> und MLA<sup>88</sup> anzuzeigen. Der einzelne Titel kann per SMS oder als E-Mail versendet werden. Darüber hinaus bietet VuFind die Möglichkeit, die Titelangaben in die Literaturverwaltungsprogramme RefWorks und EndNote zu exportieren bzw. im MARC oder RDF-Format. Auch können eigene Kommentare zum Titel verfasst oder Tags erstellt werden.

## **Konto**

Über die personalisierte Seite lassen sich Informationen über Favoriten, ausgeliehene Medien, Bestellungen und Vormerkungen, Gebühren, persönliche Angaben und die gespeicherten Suchen abrufen. Der Nutzer kann bei der Erstellung von Listen zwischen den Optionen öffentlich und privat wählen; das bedeutet, dass er selbst die Entscheidung hat, ob er seine Literaturlisten öffentlich einsehbar machen will oder nicht. Es besteht sowohl die Möglichkeit, einzelnen Literaturlisten einem Tag zuzuordnen als auch einzelne Medien zu taggen.

## **Überall**

In der Demo-Version stehen dem Nutzer auf jeder Seite folgende Funktionen zur Verfügung: Unter dem Punkt Suchoptionen können mit dem Link Suchhistory die bisherigen Suchen eingesehen werden oder es kann direkt zur erweiterten Suche gewechselt werden. Weiterhin besteht die Möglichkeit die Stöberfunktion von VuFind aufzurufen, nach studienrelevanten Medien zu suchen oder Neuerscheinungen anzuzeigen. Die studienrelevanten Medien können auf Semester, Dozent und Fachrichtung eingeschränkt werden. Neuerscheinungen seit

<sup>87</sup> Zitierformat der American Psychological Association.

<sup>88</sup> Zitierformat der Modern Language Association.

gestern, seit fünf oder dreißig Tagen können, einschränkbar auf die Fachbereiche, angezeigt werden. Auch besteht auf jeder Seite von VuFind die Möglichkeit die Sprache zu ändern und sich anzumelden, um das eigene Konto einzusehen und zu bearbeiten. Der Nutzer kann über ein Drop-Down-Menü zwischen Deutsch, Spanisch, Französisch, Italienisch, Niederländisch, Japanisch, Niederländisch, brasilianisches Portugiesisch, vereinfachtes Chinesisch, Chinesisch und Türkisch wählen, auch wenn noch nicht alle Hilfetexte übersetzt sind.

### **Browsingestieg**

Der Browsingestieg bietet dem Nutzer die Möglichkeit die Felder Tag, Signatur, Verfasser, Thema, Genre, Region und Ära über verschiedene Filter zweimal einzuschränken und bietet so einen Sucheinstieg ohne eine eigene Eingabe eines Suchwortes.

## 6. Studium – Inhalte und Ziele

Der Bachelorstudiengang Bibliotheks- und Informationsmanagement (BIM) (Library and Information Science) an der Hochschule für Angewandte Wissenschaften Hamburg (HAW) am Department Information hat eine Regelstudienzeit von sechs Semestern. Er wird mit dem akademischen Grad eines Bachelor of Arts abgeschlossen.

Ute Krauss-Leichert, Leiterin des Department Information der HAW Hamburg, fasst die Inhalte des Studiums wie folgt zusammen: „Die Inhalte der bibliothekarisch-orientierten Studiengänge orientieren sich nicht mehr an Institutionen, wie Öffentliche oder [W]issenschaftliche Bibliotheken, sondern an unterschiedlichen Handlungsfeldern. Zu den inhaltlichen Schwerpunkten zählen Informationstechnologie, Information Research, Wissensorganisation, Informations- und Medienmanagement, Public Management, Kultur- und Medienarbeit, Informationsgesellschaft und Informationsstrukturen, zielgruppenorientierte Dienstleistungen und die Vermittlung von Informationskompetenz.“<sup>89</sup>

Innerhalb des Studiums wird ein besonderer Schwerpunkt auf die Praxisnähe gelegt<sup>90</sup>, was sich auch in einem halbjährigen Praxissemester, Kooperationen mit der Medienwirtschaft und einem Studienprojekt ausdrückt.

Das Studium ist in Pflicht- und Wahlpflichtbereiche gegliedert. Im Pflichtbereich werden grundlegende Kenntnisse aus den Bereichen Informationstechnologie, Kultur und Medien, Medienpolitik und sowie des Informations- und Dienstleistungsmanagements vermittelt.

Vom 4. bis zum 6. Semester können Studierende sechs Wahlpflichtseminare aus den Bereichen Informationstechnologie, Informationsmanagement, Informationsmarkt und -dienstleistungen sowie Medien und Kultur belegen.

Der Studiengang soll Absolventen zu einer „fachlich selbstständigen Tätigkeit“, in wissenschaftlichen und öffentlichen Bibliotheken sowie in Institutionen und Unternehmen aus den Bereichen Kultur und Information, befähigen.<sup>91</sup>

---

<sup>89</sup> Vgl. KRAUSS-LEICHERT 2008 - Ausbildung im Wandel.

<sup>90</sup> Vgl. ebd.

<sup>91</sup> Vgl. SCHMIDT 2010 – Modulhandbuch, S. 4.



Der Studienschwerpunkt der Verfasser lag bei der Wahl der Wahlpflichtseminare, Praktika und Studienprojekte, weniger in den Bereichen der Informationstechnologie, sondern vor allem im Bereich der Informationsdienstleistungen. Welche Kenntnisse und Fähigkeiten, die im Rahmen des Studiums an der HAW erlangt wurden, für die erfolgreiche Umsetzung eines Katalog-2.0-Projektes von besonderer Bedeutung waren und welche Kenntnisse durch das Studium nicht ausreichend abgedeckt wurden, wird im Kapitel „Beurteilung der eigenen Kenntnisse“ analysiert.

## 7. Methodik der Arbeit

Die theoretische Vorgehensweise, auf der diese Arbeit beruht, lehnt sich eng an die aus dem Projektmanagement stammende Methode der Machbarkeitsstudie, die auch als Durchführbarkeitsstudie oder Feasibility Study bezeichnet wird, an. Eine Machbarkeitsstudie besitzt eine multifaktorielle Struktur, was bedeutet, dass sie immer aus mehreren Teilstudien besteht. Diese Teilstudien untersuchen Standortbedingungen, Wirtschaftlichkeit, Marktsituation, Zielgruppen, potentielle Nachfrage, Konkurrenz, alternative Konzeptionen und die technische Machbarkeit.

<sup>92</sup>

Durch eine Machbarkeitsstudie soll geklärt werden, ob ein bestimmtes Projekt aus heutiger Sicht realisierbar ist.<sup>93</sup> Zweck einer Machbarkeitsstudie ist es Fehlinvestitionen zu vermeiden, optimale Lösungswege anzuzeigen und Risiken aufzuzeigen.<sup>94</sup> Dieser Ansatz wurde insoweit verändert, dass die vorliegende Arbeit nicht untersucht, ob ein bestimmtes Projekt aus heutiger Sicht realisierbar ist, sondern inwieweit das Projekt technisch, mit den in einem bibliothekswissenschaftlichen Studium gewonnen Fähigkeiten und Kenntnissen, zu realisieren ist.

Der erste und wichtigste Schritt bei der Durchführung einer Machbarkeitsstudie ist die „genaue Beschreibung der Projektzielsetzung, der Projektaufgaben, einzusetzenden Technologien, benötigter Ressourcen sowie des Projektzeitrahmens.“<sup>95</sup> Zu Beginn einer Machbarkeitsstudie müssen ebenfalls Kriterien entwickelt werden, mit denen die Machbarkeit eines Projektes beurteilt werden kann.<sup>96</sup> Anhand dieser Kriterien wird nach Beendigung der Machbarkeitsstudie beurteilt, inwieweit die Zielsetzung des Projekts erfolgreich abgeschlossen werden kann.

Die Ergebnisse, die mit Hilfe einer Machbarkeitsstudie gewonnen werden, zeigen sowohl Entscheidungsmöglichkeiten, geben aber auch Empfehlungen für eine Entscheidung. Sie liefern also eine grundsätzliche Beurteilung, ob, und wenn, wie ein Konzept umgesetzt werden kann. Da Machbarkeitsstudien stets ergebnisoffen

---

<sup>92</sup> Vgl. WIENOLD 2004 - Machbarkeitsstudien 2004, S. 5.

<sup>93</sup> Vgl. BEA, SCHEURER ET AL. 2008 – Projektmanagement 2008, S. 87.

<sup>94</sup> Vgl. ANGERMEIER 2006 – Machbarkeitsstudie.

<sup>95</sup> Vgl. BEA, SCHEURER ET AL 2008 – Projektmanagement 2008, S. 87.

<sup>96</sup> Vgl. ANGERMEIER 2005 – Projektmanagement-Lexikon 2005, S. 221.

sind, kann das Ergebnis also auch sein, dass ein Projekt unter der Berücksichtigung der ursprünglichen Projektzielsetzung nicht umsetzbar ist. Auch ermitteln Machbarkeitsstudien projektgefährdende Risiken. Übertragen auf die vorliegende Arbeit wären eventuell projektgefährdenden Risiken die fehlenden Kenntnisse, die die Verfasser in Lage versetzt hätten, einen Katalog 2.0 zu erstellen. Umgekehrt liefern die Ergebnisse einer Machbarkeitsstudie aber auch die Grundlagen für das Risikomanagement, bilden also in dem hier liegenden Fall die erforderlichen Fähigkeiten und Kenntnisse ab.<sup>97</sup>

„Machbarkeitsstudien können demnach grob drei Richtungen verfolgen. Diese sind in Bewertungskonzepten („do it / don’t do it“), Alternativen („do this/do that“) oder in Realisierungsmöglichkeiten („do it like this“) zu sehen.“<sup>98</sup>

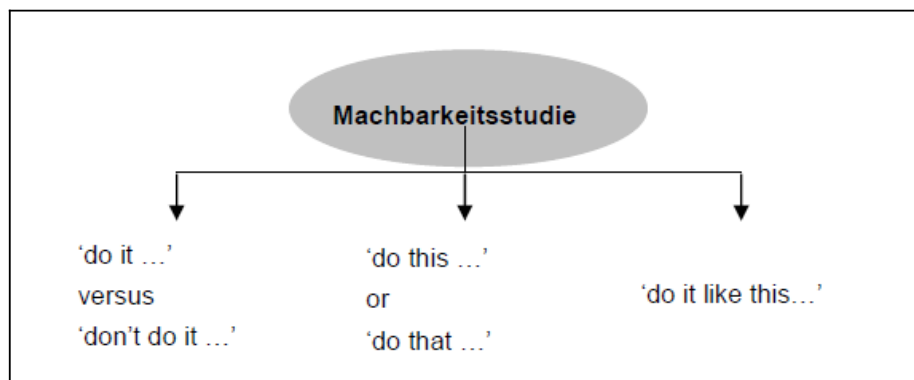


Abbildung 3: Mögliche Ergebnisalternativen von Machbarkeitsstudien  
(WIENOLD 2004 - Machbarkeitsstudien, S. 5.)

In Anlehnung an die Theorie von Wienold, die Bewertungskonzepte, Alternativen und Realisierungsmöglichkeiten als Ergebnisse von Machbarkeitsstudien beschreibt, wurde folgendes Modell zur Beantwortung der Fragestellung entwickelt.

Es zeigt sowohl das Bewertungskonzept („Yes we can“ versus „No we can’t“), differenziert davon ausgehend aber insoweit, dass Realisierungsmöglichkeiten („we got help“ versus „this could be the way“) aufgezeigt werden.

<sup>97</sup> Vgl. ANGERMEIER 2005 – Projektmanagement-Lexikon, S. 221.

<sup>98</sup> Vgl. WIENOLD 2004 – Machbarkeitsstudien, S. 5.

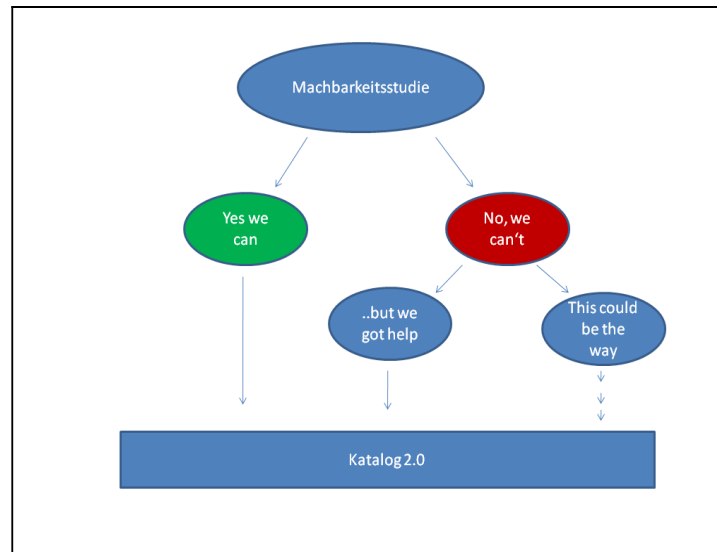


Abbildung 4: Theoretischer Ansatz des Bewertungsschemas

Vor dieser Machbarkeitsstudie wurde eine Machbarkeitsprüfung vorgenommen. Durch eine Machbarkeitsprüfung, auch Feasibility Check genannt, soll überprüft werden, ob das eigentliche Projektziel überhaupt technisch erreichbar ist oder schon die Grundidee verworfen werden sollte.

Der Begriff der Machbarkeitsstudie ist nicht mehr an eine bestimmte Norm gebunden. Die früher gültige DIN 69905:1987, die den veralteten Begriff Projektstudie anstatt Machbarkeitsstudie verwendete, ist nicht mehr gültig.<sup>99</sup> Vorgeschlagen wird in der Literatur, Machbarkeitsstudien mit Hilfe von technisch-wissenschaftlichen Analysen durchzuführen, die auf einer Literaturrecherche, Expertenbefragungen und rechnergestützten Simulationen beruhen.<sup>100</sup> Im Unterschied zur eigentlichen Idee der Machbarkeitsstudie nutzen die Verfasser aber keine Simulationssoftware, sondern nutzen eine bereits verfügbare Software als Grundlage zur Umsetzung. Die Umsetzung folgt den im Kapitel „Katalog 2.0“ beschriebenen Prinzipien der Bibliothek 2.0 und verdeutlicht deren Bedeutung für eine erfolgreiche Durchführung von Katalog-2.0-Projekten.

Auch wenn es keinen konkreten Auftraggeber für das hier durchgeführte Projekt gibt, entspricht es doch der Definition eines Projekts. Dies erklärt die Wahl einer Methode aus dem Projektmanagement.

<sup>99</sup> Vgl. ANGERMEIER 2006 - Machbarkeitsstudie.

<sup>100</sup> Vgl. ANGERMEIER 2005 - Projektmanagement-Lexikon, S. 221.

Die DIN69901 definiert ein Projekt als „[...] ein Vorhaben, das im [W]esentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie z.B.: Zielvorgabe, zeitliche, finanzielle, personelle oder andere Bedingungen, Abgrenzungen gegenüber anderen Vorhaben und projektspezifische Organisation.“

Auf die Erstellung eines Lastenheftes wird verzichtet, da es nicht um die Machbarkeit an sich, sondern um die Machbarkeit in Bezug auf die vorab beschriebenen Kenntnisse von Absolventen eines bibliothekarischen Studiums an der HAW Hamburg geht. Das Ergebnis der Machbarkeitsstudie spiegelt sich daher vielmehr in der Bewertung, inwieweit und wie die in der Projektzielsetzung definierten Ziele erreicht wurden, wieder.

## 8. Beschreibung der praktischen Vorgehensweise

Ausgehend von der oben aufgezeigten Beschreibung der Methodik wurden von den Verfassern folgende Schritte, Überlegungen und Vorgehensweisen umgesetzt.

### 8.1. Machbarkeitsprüfung

Zur Durchführung der Machbarkeitsprüfung, also der Beantwortung der Frage, ob das eigentliche Projektziel überhaupt (technisch) erreichbar ist oder schon die Grundidee verworfen werden sollte, mussten vorab folgende drei Fragen beantwortet werden, deren Beantwortung zwingend ein „Ja!“ erforderte.

- Können die Verfasser eine Grundinstallation mit Testdaten aufsetzen und diese online abrufbar machen?
- Besteht grundsätzlich die Möglichkeit Metadaten des HIBS zu erhalten und zu nutzen?
- Gibt es eine Möglichkeit die Verfügbarkeitsinformationen der Medien anzuzeigen?

Die Antwort auf die erste Frage konnte durch die erfolgreiche Grundinstallation unter Verwendung von Testdaten der DNB bejaht werden.<sup>101</sup>

Auch die zweite Frage konnte positiv beantwortet werden, da Till Kinstler, Mitarbeiter des GBV, die notwendigen Metadaten zur Verfügung stellen würde.

Die dritte Frage konnte durch die Existenz der Document Availability Information API (DAIA)<sup>102</sup> ebenfalls mit Ja beantwortet werden.

Anhand dieser Erkenntnisse wurde die grundsätzliche Durchführbarkeit, also die Verwirklichung der Grundfunktionen eines Kataloges (Import und Anzeige der Metadaten sowie Anzeige der Verfügbarkeit) als positiv beurteilt und daher mit der eigentlichen Machbarkeitsstudie, die die Grundlage dieser Arbeit bildet, begonnen.

---

<sup>101</sup> Vgl. hierzu das Kapitel „Import von Metadaten“.

<sup>102</sup> GBV WIKI – Verfügbarkeitsrecherche mit DAIA.

## 8.2. Praktische Umsetzung der Methodik

Angelehnt an eine Machbarkeitsstudie werden nun also die Projektaufgaben, die Projektzielsetzung, die benötigten Technologien und der Zeitrahmen beschrieben. Die Umsetzung des im Abschnitt „Beschreibung der Methodik“ entwickelten Modells zur Entwicklung des Bewertungskonzepts wird nun spezifiziert und endgültig definiert. Im Hinblick auf die Prinzipien der Bibliothek 2.0 werden alle Änderungen und Fortschritte, die im Rahmen der Projektaufgaben erreicht werden, unter der Webadresse <http://www.bachelopac.de.vu> und zusätzlich im Sinne der Offenheit von Bibliothek 2.0 über ein eigenes Blog<sup>103</sup> bzw. über einen Twitteraccount<sup>104</sup> veröffentlicht. Der Forderung, Bibliotheksbenutzer an der Gestaltung und an der Implementierung von Dienstleistungen teilhaben zu lassen, wollen die Verfasser insoweit gerecht werden, dass die bereits erfolgten Usability-Test und Fokusgruppen, die im Rahmen des beluga-Projekts durchgeführt wurden, berücksichtigt werden.<sup>105</sup> Es kann davon ausgegangen werden, dass sich das untersuchte Nutzerverhalten der Kunden der Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky (SUB Hamburg) und Kunden des Hochschulinformations- und Bibliotheksservice der HAW Hamburg (HIBS) nicht unterscheidet. Ein großes Augenmerk bei der Bewältigung der gestellten Projektaufgaben legen die Verfasser auf die Nachnutzbarkeit von anderen Programmen und Ideen Dritter im Sinne der Bibliothek 2.0.

### 8.2.1. Projektaufgaben

Ausgehend von der bereits erarbeiteten Definition eines Katalog 2.0, der Beschreibung von VuFind aus Nutzersicht im Sinne des Best-Practice-Gedankens und den Ergebnissen der erfolgten Machbarkeitsprüfung, werden folgende Anforderungen an die Funktionen für den zu verwirklichenden Katalog im Rahmen des Projekts „bachelopac“ definiert. Es bedarf daher der Definition der folgenden zu erreichenden Projektaufgaben im Sinne des Projektmanagements.

---

<sup>103</sup> <http://www.bachelopac.wordpress.com>.

<sup>104</sup> <http://twitter.com/bachelopac>.

<sup>105</sup> Vgl. SCHMITT, STEHLE 2009 – Stimmen aus dem Usability-Labor.

### **Datenimport, Datenabgleich und Verfügbarkeit**

Die Datensätze des HIBS können vollständig und fehlerfrei in eine Grundinstallation von VuFind eingespielt werden. Der Datenabgleich soll automatisiert werden, das heißt, dass sowohl neue Titel automatisch in den Index aufgenommen, als auch makulierte Titel aus dem Index gelöscht werden. Die Anzeige der Verfügbarkeit und des Standortes der Exemplare soll im bachelopac korrekt angezeigt werden.

### **Suche und Stöbern**

Facettiertes Browsen soll nicht nur möglich sein, sondern zusätzlich soll eine Facette mit der Basisklassifikation erstellt und angezeigt werden, die das HIBS zur Klassifikation verwendet. Zudem soll eine Eingrenzungsmöglichkeit auf Abschlussarbeiten der HAW Hamburg erstellt werden. Die Anzeige „ähnliche Treffer“ soll verfügbar sein und zusätzlich die Basisklassifikation bei der Auswahl anderer relevanter Treffer berücksichtigen. Die dazu notwendige Indexierung der Basisklassifikation soll exemplarisch die Möglichkeit der Indexierung von MARC-Datenfeldern zeigen, die von VuFind nicht automatisch indexiert werden. Das Relevance Ranking soll insoweit verändert werden, dass Lehrbücher höher gerankt werden. Eine fehlertolerante Suche wird angestrebt und soll unter Verwendung eines Stemmingalgorithmus für die deutsche Sprache, einer Rechtschreibprüfung und einem Retrieval nach dem closest-match-Verfahren erreicht werden. Ebenso soll die von VuFind verwendete Suchsyntax von Lucene/DisMax vollständig anwendbar sein.



### **Personalisierung**

Nutzer können sich bei VuFind anmelden und so folgende Funktionen nutzen: Kommentare können zu einzelnen Titeln in der Titelanzeige erstellt werden, einzelne Titel können mit Tags versehen werden und Listen können sowohl öffentlich als auch privat erstellt, getaggt und gespeichert werden. Die Suchhistorie soll ebenfalls speicherbar sein.

### **Mashups und Kataloganreicherungen**

Sowohl die Trefferlisten als auch die Titelanzeige sollen durch die Anzeige von Covern ergänzt werden. Hierzu sollen die Angebote von Google Books, Librarything, Amazon und OpenLibrary genutzt werden. Die Titelanzeige soll durch die Einbindung von Amazon-Rezensionen erweitert werden. Informationen über Autoren aus der Wikipedia erweitern die Titelanzeige.

### **Funktionen**

Sowohl Suchanfragen also auch einzelne Titel können als E-Mail versendet werden. Die Möglichkeit RSS-Feeds für Suchanfragen zu abonnieren besteht. Die in VuFind integrierte Zitierfunktion, die es erlaubt Zitate für Literaturverzeichnisse zu erstellen, soll im bachelopac verfügbar sein. Die Metadaten einzelner Titel sind in die Literaturverwaltungsprogramme RefWorks und EndNote exportierbar.

### **Anpassung der Katalogoberfläche**

Das Design soll insoweit verändert werden, dass es dem Corporate Design der HAW Hamburg entspricht und die vorgenommenen Modifikationen anzeigt. Fehlerhafte und missverständliche Bezeichnungen sollen verbessert, fehlende Übersetzungen ergänzt werden.

## **Einschränkungen**

Auch wenn sich unter der Verwendung von VuFind noch zahlreiche andere Funktionen und Anwendungen realisieren lassen könnten (siehe Kapitel „VuFind“), müssen die Projektaufgaben auf Grund des kurzen Projektzeitraums auf die Grundfunktionen eines Kataloges reduziert werden. Eine mögliche Anbindung an die Nutzerkonten des ILS ist nicht überprüfbar (siehe Kapitel „Personalisierung“).

### **8.2.2. Projektziele**

Das Projektziel ergibt sich zunächst aus der Fragestellung der vorliegenden Arbeit. Ziel ist es die Machbarkeit der Erstellung eines Kataloges 2.0 unter Verwendung von VuFind unter Berücksichtigung der im Studium erlangten Kenntnissen und Fähigkeiten der Verfasser zu überprüfen. Ressourcen und mögliche Ansprechpartner sollen im Hinblick auf die Nachnutzbarkeit von bereits erarbeiteten Lösungen identifiziert werden im Sinne der schon im Titel dieser Arbeit betonten Einbeziehung der Community.

### **8.2.3. Einzusetzende Technologien/ benötigte Ressourcen**

- Server: Virtual Server, 1GB RAM
- Betriebssystem: Ubuntu Linux 8.04 Server Edition
- Katalogsoftware: VuFind 1.0
- Metadaten: Datensätze des HIBS im Format MARC21

### **8.2.4. Projektzeitrahmen**

Der Projektzeitrahmen wird zunächst durch die Studienordnung der HAW Hamburg begrenzt und umfasst drei Monate zur Erarbeitung der Bachelorarbeit. Nach der 10. Projektwoche werden bis zum Abschluß der Bachelorarbeit keine technischen Veränderungen mehr am Katalog vorgenommen.

### 8.3. Bewertungskriterien

Ausgehend von der im Kapitel „Beschreibung der Methodik“ entwickelten Methodik zur Bewertung der Machbarkeitsstudie, werden nun die folgenden Bewertungskriterien, mit denen die einzelnen im Abschnitt „Projektziele“ festgelegten Teilziele bewertet werden, festgelegt. Die einzelnen Bewertungskriterien wurden auf Basis der Fragestellung dieser Arbeit

*„Inwieweit können Absolventen eines bibliothekarischen Bachelor-Studiums einen zeitgemäßen OPAC auf Basis von VuFind selbst realisieren? Welche Fähigkeiten sind dazu erforderlich? Welche Kenntnisse müssen erworben werden? An welchen Stellen wird Unterstützung von Experten benötigt?“*

erarbeitet und bilden demzufolge auch die Bewertungsgrundlage zur Beantwortung der Fragestellung:

**1.** Das geforderte Teilziel lässt sich „out-of-the-box“ verwirklichen. „Out-of-the-box“ meint hier, dass es keinerlei zusätzlicher Konfiguration durch die Verfasser bedarf, sondern die gewünschten Funktionen direkt nach der Installation von VuFind zur Verfügung stehen.

**2.** Das geforderte Teilziel konnte von den Verfassern selbständig („Yes we can“) erreicht werden. Da die Machbarkeitsstudie aber auch gleichzeitig Realisierungsmöglichkeiten anzeigen soll, bedarf es hier einer weiteren Differenzierung. Diese Differenzierung ist ebenso notwendig, um zwischen den in einem bibliothekarischen Bachelor-Studium erworbenen Kenntnisse und den im Rahmen dieser Arbeit erworbenen Kenntnisse bzw. der Unterstützung durch die Community zu unterscheiden.

**2a** Das Teilziel wird mit den im Studium erworbenen Kenntnissen und Fähigkeiten, der verfügbaren Literatur und durch die Analyse des Quellcodes von VuFind bzw. deren Kommentierung im Sinne von „Yes we can“ gelöst.

**2b** Auch hier wird das Teilziel erreicht, kann aber mit den in 2a beschriebenen Lösungswegen nicht erreicht werden. Zur Lösung des Problems bedarf es der Überlassung von bereits programmierten Skripten, dem Einblick in den Quellcode von anderen Katalogprojekten oder der

Beantwortung spezieller, in der Literatur (noch) nicht dokumentierten Problemen. Aus „No we can't“ wird Dank „but we got help“ ein „Yes we can“.

**3** Das Teilziel kann zwar erreicht werden, dies geschieht aber nur durch die aktive Hilfe von Dritten. Aktive Hilfe von Dritten meint hier, dass andere Personen sich auf den bachelopac-Server einloggen und Probleme lösen, die die Verfasser nicht selbstständig bewältigen können. „No we can't“ führt also unter der Berücksichtigung von „but we got help“ trotzdem einen Schritt näher an die Verwirklichung des geplanten Katalogs 2.0.

**4** Das geplante Teilziel kann im Rahmen der Machbarkeitsstudie nicht erreicht werden. Die aufgezeigten Lösungswege könnten trotzdem einen Schritt näher zum Ziel Katalog 2.0 führen. In Abbildung 5 werden die entwickelten Bewertungskriterien grafisch visualisiert.

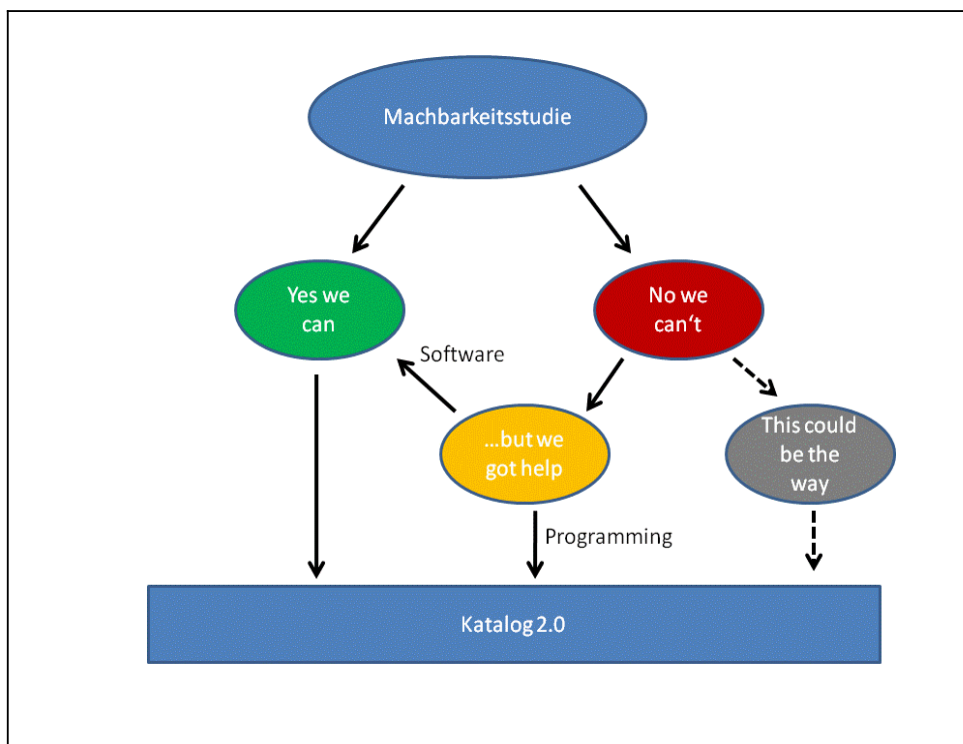


Abbildung 5: Bewertungsschema zur Lösung der Projektaufgaben

## 9. Der Weg zum bachelopac

### 9.1. Grundinstallation

#### 9.1.1. Ubuntu Linux 8.04 LTS

VuFind kann auf vielen verschiedenen Betriebssystemen installiert werden, wobei Linux am geläufigsten ist.<sup>106</sup>

Prinzipiell ist es auch möglich VuFind auf Windows-Systemen zu betreiben. Da VuFind aber für Linux konzipiert wurde, ist für Windows-Installationen ein größerer Installations- und Konfigurationsaufwand erforderlich.<sup>107</sup>

Die Verfasser entschieden sich aus folgenden Gründen für das Betriebssystem Ubuntu Linux:

- Priorisierung von Open-Source-Software
- Ubuntu ist die meist verbreitete Linux-Distributionen<sup>108</sup> und verfügt über eine „exzellente Internet-basierte Infrastruktur mit einem Wiki als Dokumentation“<sup>109</sup>, die im weiteren Verlauf der Arbeit auch als Referenz herangezogen wird.
- Ubuntu basiert technisch auf der Distribution Debian und nutzt deren Software-Paketverwaltung Advanced Packaging Tool (APT). Eine Paketverwaltung ermöglicht es in vielen Fällen, eine Software mit nur einer Kommandozeile herunterzuladen und zu installieren, da abhängige Software nach Möglichkeit automatisch nachinstalliert wird. Für die VuFind Version 1.0, die am 15.07.2010 veröffentlicht wurde, waren bereits im Vorfeld Software-Pakete für Debian-basierte Linux-Distributionen angekündigt.<sup>110</sup>

---

<sup>106</sup> Vgl. VUFINd DOCUMENTATION 2010 – Installations.

<sup>107</sup> Vgl. VUFINd DOCUMENTATION 2010 – Windows.

<sup>108</sup> Unter einer Linux-Distribution wird eine Zusammenstellung des Linux-Kernels mit zusätzlicher Software verstanden.

<sup>109</sup> Vgl. DISTROWATCH 2010 – Ein Wegweiser.

<sup>110</sup> Vgl. NAGY 2008 – VuFind Release.

Das Prozedere der Grundinstallation eines Ubuntu-Servers kann im Ubuntu Users Wiki<sup>111</sup> sowie im Ubuntu Server Tutorial von Zimmer<sup>112</sup> nachgelesen werden. Die Grundinstallation eines Ubuntu-Servers stufen die Verfasser auch für Linux-Neueinsteiger als machbar ein. Weiterhin müssen Grundkenntnisse zur kommandobasierten Benutzung („Shell“)<sup>113</sup> eines Betriebssystems vorhanden sein oder erworben werden.

### 9.1.2. VuFind 1.0

Die Installation von VuFind auf Ubuntu-Systemen ist in der VuFind-Dokumentation<sup>114</sup> detailliert beschrieben. Durch das Paketmanagement-System APT (Advanced Packaging Tool) ist die Grundinstallation mit nur wenigen Kommandozeilen zu bewerkstelligen, da ab VuFind Version 1.0 ein Debian-Software-Paket zur Verfügung steht.

Dieses muss nur noch heruntergeladen und mit der Befehlszeile

```
# sudo dpkg -i VuFind_1.0.deb
```

installiert werden. Sollten die von VuFind benötigten Komponenten (Apache, MYSQL, PHP, Java) noch nicht installiert sein, können diese mit nur einer Befehlszeile

```
# sudo apt-get --fix-missing
```

aus den Paketquellen heruntergeladen und eingerichtet werden.

---

<sup>111</sup> UBUNTU USERS WIKI 2010 - Server Installation.

<sup>112</sup> ZIMMER 2008 - Ubuntu 8.04.

<sup>113</sup> UBUNTU USERS WIKI 2010 - Shell.

<sup>114</sup> VUFIND DOCUMENTATION 2010 - Ubuntu.

### 9.1.3. Basiskonfiguration

Bevor die VuFind-Installation lauffähig ist, müssen einige Anpassungen in der Datei `vufind/web/conf/config.ini` vorgenommen werden. Die Konfigurationsdatei ist durch Kommentarzeilen ausführlich dokumentiert. Folgende Anpassungen wurden zunächst ausgeführt:

<pre>url = http://library.myuniversity.edu/vufind email = support@myuniversity.edu title = "Library Catalog" language = en locale = en_US timezone = "America/New_York"</pre>	<pre>url = http://83.169.45.69/vufind email = marcel.stehle@haw-hamburg.de title = "bachelopac" language = de locale = de_DE timezone = "Europe/Berlin"</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

Listing 1: Erste Anpassungen der Datei `config.ini`

VuFind kann jetzt mit dem Befehl

```
# /usr/local/vufind/vufind.sh start
```

gestartet<sup>115</sup> und im Browser aufgerufen werden.

An dieser Stelle sind die Verfasser auf das Problem gestoßen, dass der Solr-Index nicht verfügbar war.

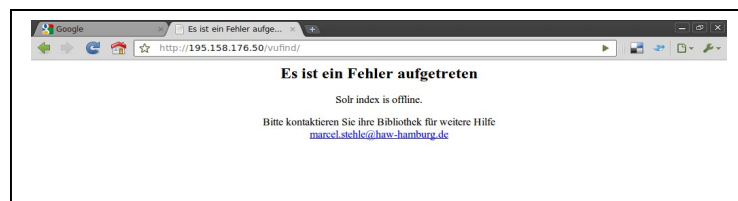


Abbildung 6: Fehlermeldung: "Solr index is offline"

Eine Suche nach der Fehlermeldung im Archiv der Mailingliste VuFind-General führte jedoch schnell zu einer funktionierenden Lösung.<sup>116</sup>

<sup>115</sup> VUFind DOCUMENTATION 2010 - starting and stopping.

<sup>116</sup> KATZ 2010 – Re: Installation problems.

## Absichern der VuFind-Installation

Nach der Grundinstallation sollte VuFind, wie in der Dokumentation beschrieben, abgesichert<sup>117</sup> werden. Dieser Schritt sollte nicht übersprungen werden, da es sonst nicht autorisierten Personen beispielsweise möglich wäre, Delete-Befehle über den Browser zu versenden, um Datensätze zu löschen.<sup>118</sup>

## Erforderliche Kenntnisse zur Grundinstallation

Für die Grundinstallation von VuFind auf Basis von Ubuntu Linux waren folgende Kenntnisse erforderlich:

- Installation eines Betriebssystems
- Konfiguration der Netzwerkverbindung
- Grundkenntnisse der kommandobasierten Nutzung (Shell) eines unix-artigen Betriebssystems
- Eine Konfigurationsdatei anpassen
- Kenntnisse zur Fernsteuerung eines Rechners über SSH (secure shell), falls der Server über ein Netzwerk angesprochen werden soll

## Einstufung der Grundinstallation von VuFind nach den Kriterien der Machbarkeit: 2a

---

<sup>117</sup> VUFind DOCUMENTATION 2010 – security.

<sup>118</sup> KINSTLER 2010 - Re: Deleting records.



## 9.2. Import von Metadaten

### 9.2.1. Für den Import geeignete Formate

Um bibliographische Daten ohne Umwege in einen VuFind-Katalog importieren zu können, müssen diese in einem binären MARC-Format vorliegen. Das Akronym MARC steht für Machine Readable Cataloging. MARC21, eine Synthese der Formate UKMARK, CANMARK und USMARK,<sup>119</sup> ist mittlerweile der weltweit am weitesten verbreitete Standard zum Austausch von Katalogdaten.<sup>120</sup>

In Deutschland und Österreich ist das Maschinelle Austauschformat für Bibliotheken MAB2 für den Datenaustausch gebräuchlich. Der Standardisierungsausschuss hat 2004 den Umstieg von MAB2 auf MARC21 als einheitliches Austauschformat beschlossen, um die Voraussetzungen für die Fremddatenübernahme und den Datenaustausch national und international zu verbessern.<sup>121</sup> Die Umsetzung des Beschlusses dauert allerdings noch an.

VuFind benutzt die Software SolrMarc, um bibliografische Daten im MARC-Format zu indexieren und stellt hierfür das Import-Skript `import-marc.sh` bereit. Damit der Import gelingt, müssen die Datensätze folgende Eigenschaften aufweisen:

- Binäres MARC-Format nach ISO2709
- Dateinamen der Datenpakete müssen mit der Dateierweiterung „.mrc“ enden
- Unterscheidbarkeit durch einen eindeutigen Identifikator<sup>122</sup>

---

<sup>119</sup> Vgl. RUSCH 2008 – MARC 21.

<sup>120</sup> HELMKAMP, OEHLISCHLÄGER 2006 – Internationalisierung, S. 1282.

<sup>121</sup> Vgl. ebd.

<sup>122</sup> Vgl. VUFINDDOCUMENTATION 2010 – Importing Records.

Falls ein Bibliothekssystem keine Datenpakete exportieren kann, die die oben genannten Kriterien erfüllen, können folgende Lösungswege beschritten werden:

### **1. Anpassung von VuFind an ein anderes Datenformat**

Um bibliographische Daten zu importieren, die nicht im MARC-Format vorliegen, muss ein neues Importprogramm geschrieben werden, mit dem die Metadaten indiziert werden können. Zudem muss eventuell ein neuer Treiber zur Anzeige der Datensätze entwickelt werden. Ein „Generic Record Driver“, der sich prinzipiell zur Anzeige aller Datenformate eignet, ist vorhanden. Dieser kann jedoch nicht alle Besonderheiten eines speziellen Datenformats abbilden.<sup>123</sup>

### **2. Die Metadaten in ein MARC-Format konvertieren**

Eine fertige Softwarelösung zur Konvertierung von MAB2 nach MARC21 existiert nicht.<sup>124</sup> Sofern eine Bibliothek an der Verbundkatalogisierung teilnimmt, besteht die Möglichkeit, über die Verbundzentrale die Daten aus deren Zentralsystem in einem MARC-Format zu exportieren. Dabei kann es allerdings zu kleineren Abweichungen zwischen den exportierten Daten aus dem Zentralsystem und denen im lokalen System der Bibliothek kommen.<sup>125</sup>

---

<sup>123</sup> Vgl. VUFINd DOCUMENTATION - other than marc.

<sup>124</sup> Vgl. KINSTLER 2010 – Re: Datenkonvertierung.

<sup>125</sup> Ebd.

## 9.2.2. Pretest mit Beispieldatensätzen

Um die Basisfunktionalitäten von VuFind zu testen, empfiehlt es sich, zunächst mit frei zugänglichen Beispieldaten im MARC21-Format zu arbeiten. Die Verfasser verwendeten im Rahmen der Machbarkeitsprüfung 540 Testdatensätze der Reihe B, die der Datendienst der Deutschen Nationalbibliothek mit der Datei Btestutf8.mrc zum kostenlosen Download bereitstellt.<sup>126</sup> Der folgende Auszug aus dem Terminal zeigt den Datenimport des Metadatenpakets:

```
# /usr/local/vufind/import-marc.sh /home/bachel/Btestutf8.mrc
Now Importing /home/bachel/Btestutf8.mrc ...

INFO [main] (MarcImporter.java:769) - Starting SolrMarc indexing.
INFO [main] (MarcHandler.java:325) - Attempting to open data file:
/home/bachel/Btestutf8.mrc
[...]
WARNUNG: [biblio] Solr index directory '/usr/local/vufind/solr/biblio/index'
doesn't exist. Creating new index...

INFO [main] (MarcImporter.java:266) - Added record 1 read from file:
010101659
[...]
INFO [main] (MarcImporter.java:266) - Added record 540 read from file:
999975358

INFO [main] (MarcImporter.java:516) - Adding 540 of 540 documents to index
INFO [main] (MarcImporter.java:517) - Deleting 0 documents from index
INFO [main] (MarcImporter.java:391) - Calling commit
INFO [main] (MarcImporter.java:402) - Done with the commit, closing Solr
```

Listing 2: Import von Testdatensätzen der DNB

<sup>126</sup> DNB 2010 – Datendienst Biografische Dienstleistungen.

## Ergebnisse des Pretests

Der Import der Testdaten verlief ohne Probleme. Durch eine „leere Suche“ waren alle 540 Datensätze recherchierbar. Die Darstellung der Beispieldatensätze war nahezu fehlerfrei. Vorkonfigurierte Features wie facettiertes Browsen, Relevanzsortierung, Rechtschreibkorrektur oder die Zitierfunktion konnten mit den Beispieldatensätzen sofort ausprobiert werden. Das Metadatenpaket Btestuf8.mrc des DNB-Datendienstes kann für die ersten Gehversuche mit VuFind weiterempfohlen werden.

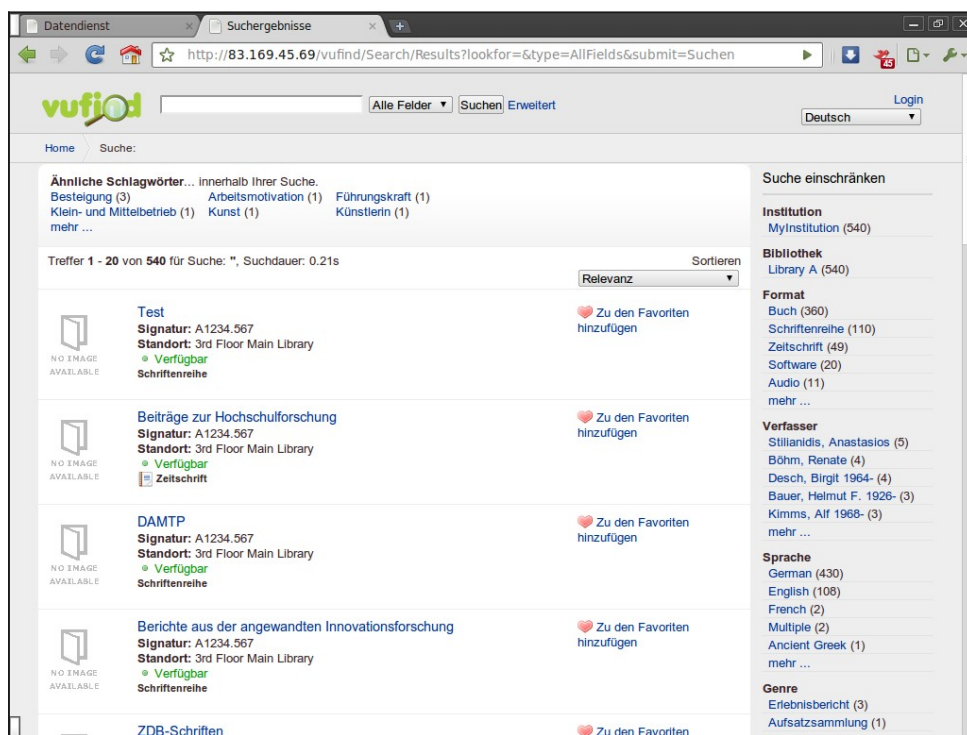


Abbildung 7: Trefferliste einer „leeren Suche“ nach Einspielung der DNB-Testdaten der Reihe B

### 9.2.3. Import eines Datengrundstocks des HIBS

Till Kinstler, Mitarbeiter der Verbundzentrale des Gemeinsamen Bibliotheksverbundes (VZG), erstellte freundlicherweise für die Verfasser ein Metadatenpaket im MARC21-Format, in dem nahezu alle Titel der Bibliothek der HAW Hamburg enthalten sind, die zwischen dem 01.01.1990 und dem 11.05.2010 geändert wurden. Der Datendump wurde über die Z39.50-Schnittstelle<sup>127</sup> des GBV erstellt und entspricht laut Kinstler nicht exakt einem sogenannten „Abzug“.<sup>128</sup>

Bevor die Daten des HIBS eingespielt wurden, musste der Index, in dem noch die Testdaten der DNB enthalten waren, mit dem Befehl

```
# sudo rm -rf /usr/local/vufind/solr/biblio/index
```

gelöscht werden.

Da beim Import des 177 MB großen Datendumps mit mehr als 160000 Datensätzen eine umfangreiche Bildschirmausgabe zu erwarten war, wurde diese in die Logdatei haw-import.log umgeleitet, um die spätere Analyse des Importvorgangs zu erleichtern.

Der Befehl dazu lautet:

```
# /usr/local/vufind/import-marc.sh haw_dump_2010_05_11.mrc > haw-import.log
```

#### Auswertung des Importvorgangs

Im Gegensatz zum Import der Beispieldaten der DNB verlief der Import der Datensätze des HIBS nicht ohne Fehler. Von den 167253 Datensätzen waren in VuFind nur 152714 (92%) recherchierbar.

---

<sup>127</sup> Siehe auch Kapitel „Automatisierter Metadatenabgleich“.

<sup>128</sup> Vgl. KINSTLER 2010 – Re: Datenkonvertierung.

Eine Auswertung der Datei haw-import.log ergab, dass die Importfehler im Wesentlichen auf zwei verschiedene Fehlertypen zurückzuführen waren, die besonders häufig aufgetreten sind: „ERROR: multiple values encountered for non multiValued field author-letter“ und “Error reading record: unable to parse record length”.

Den Verfassern war es nicht möglich, die ausgegebenen Fehler zu interpretieren. Daher wurde Till Kinstler, der VuFind im Rahmen des „Suchkiste“-Projekts einsetzt und bereits 2009 Tests mit Daten einer deutschen Universitätsbibliothek durchführte<sup>129</sup>, um Unterstützung gebeten.

### **Error: multiple values encountered for non multiValued field**

Laut Kinstler beruht der Fehler "multiple values encountered for non multiValued field" darauf, dass „in einem Datensatz Felder mehrfach auftraten, die in Solr nur als "darf nur einmal vorkommen" definiert sind (vereinfachte Erklärung).“<sup>130</sup> Um dieses Problem zu lösen, empfahl er, die mehrfache Belegung eines Indexfeldes pro Datensatz im Index-Schema<sup>131</sup> von Solr zu erlauben. MARC21 unterscheidet nur zwischen dem ersten Autor und weiteren Verfassern. „In unseren Daten kann es aber durchaus mehrere "erste Autoren" geben, weil beim Export des Picaplus-Formats in MARC21 nicht unbedingt klar ist, welches nun der erste Autor ist.“<sup>132</sup>

Dank dieser Hilfestellung konnte das Import-Schema entsprechend angepasst werden, sodass bei einer erneuten Indexierung des Datenpakets der Fehler nicht mehr aufgetreten ist.

### **Error reading record: unable to parse record length**

Eine Lösung für das Problem bezüglich der Fehlermeldung „Error reading record“ wurde aus zeitlichen Gründen nicht weiter verfolgt, zumal laut Kinstler die Wahrscheinlichkeit, dass derartige Lesefehler bei der Verwendung eines Abzugs des GBV auftreten, als „deutlich geringer“<sup>133</sup> einzuschätzen ist.

---

<sup>129</sup> Vgl. KINSTLER 2010 – Hacking VuFind

<sup>130</sup> Vgl. KINSTLER 2010 – Re: Fragen über Fragen.

<sup>131</sup> Die entsprechende Konfigurationsdatei ist vufind/solr/biblio/conf/schema.xml.

<sup>132</sup> Vgl. KINSTLER 2010 – Re: Fragen über Fragen.

<sup>133</sup> Ebd.

## **Erforderliche Kenntnisse zum Import von bibliographischen Daten im MARC-Format**

- Grundkenntnisse zur kommandobasierten Benutzung (Shell)<sup>134</sup> eines Linux-Systems, um den Datenimport auslösen und protokollieren zu können
- Grundverständnis von Katalogisierungsformaten
- Evtl. Grundverständnis von Markup-Sprachen, um Änderungen am Index-Schema vornehmen zu können

### **Einstufung nach den Kriterien der Machbarkeit: 2a**

## **9.3. Automatisierter Metadatenabgleich**

Im Rahmen eines Experteninterviews<sup>135</sup> mit Oliver Marahrens, der in der Abteilung Digitale Dienste der Universitätsbibliothek der Technischen Universität Hamburg-Harburg (TUB HH) an der Realisierung des auf VuFind basierenden Katalogs TUBFind arbeitet, informierten sich die Verfasser über deren Lösungsweg zum automatischen Datenabgleich. Die dabei zum Einsatz kommenden selbstgeschriebenen Skripte wurden dankenswerterweise zur Nachnutzung überlassen. Diese Skripte sind mittlerweile auch online zugänglich.<sup>136</sup>

Da das Bibliothekssystem OCLC PICA LBS4 keine offene XML-Schnittstelle zum Datenexport in ein MARC-Format anbietet,<sup>137</sup> wird auf die Z39.50-Schnittstelle des GBV zurückgegriffen. Die Verbundzentrale des GBV definiert die Schnittstelle Z39.50 dabei wie folgt:

„Die Z39.50-Schnittstelle ist ein international standardisiertes Protokoll zur Kommunikation zwischen bibliothekarischen Datenbanksystemen (Servern) und Zugriffsprogrammen (Clients). Über Z39.50 können Recherchen in den Datenbanken durchgeführt und bibliographische Daten übermittelt werden.“<sup>138</sup>

---

<sup>134</sup> UBUNTU USERS WIKI 2010 - Shell.

<sup>135</sup> MARAHRENS 2010 - Experteninterview.

<sup>136</sup> <http://www.tub.tu-harburg.de/tubfind-blog/2010/09/hilfreiche-VuFind-skripte/>.

<sup>137</sup> Vgl. MARAHRENS 2010.

<sup>138</sup> GBV 2010 – Z39.50.

### 9.3.1. z3950fetch

Um die Metadaten aus dem CBS des GBV im MARC21-Format herunterzuladen, kommt der von Heiko Weier<sup>139</sup> entwickelte Z39.50-Client z3950fetch zum Einsatz.

Das in der Webserver-Sprache PHP geschriebenes Skript z3950fetch nutzt die PHP-Erweiterung PHP/YAZ. PHP/YAZ ist wiederum Teil des freien Z39.50-Toolkits YAZ und ermöglicht die Implementierung von Z39.50-Client-Funktionalitäten in PHP.<sup>140</sup>

Hinweis: In der von den Verfassern verwendeten Ubuntu-Version 8.04 waren die Paketquellen für YAZ veraltet. Daher wurde eine aktuelle Version verwendet, die von der Homepage von PHP/YAZ bezogen wurde.<sup>141</sup> Die PHP-Erweiterung yaz.so musste noch in die Datei /etc/php5/cli/php.ini eingetragen werden.

Das Skript z3950fetch ist für folgende Aufgaben konzipiert:

- Download von Datensätzen, die einer bestimmten PPN, ISBN oder ISSN zugeordnet sind
- Download aller Datensätze, die an einem bestimmten Datum erstellt wurden
- Download aller Datensätze, die am Vortag erstellt wurden

Der folgende Auszug aus dem Terminal zeigt exemplarisch, wie mit dem Skript z3950fetch 15 Datensätze, die am 03.07.2010 erstellt wurden, heruntergeladen und in der Datei day-2010-07-03.mrc gespeichert wurden.

```
# ./z3950fetch.php -day 2010 07 03
15 found for @attr 1=5043 @attr 5=1 "2010 07 03" Job done in 8.88635706902
sec.
# ls
day-2010-07-03.mrc
```

Listing 3: Download von Metadaten mit Z3950fetch.php

<sup>139</sup> Heiko Weier ist Fachreferent für Informatik an der TUB HH.

<sup>140</sup> Vgl. INDEX DATA 2010 – PHP/YAZ.

<sup>141</sup> <http://www.indexdata.com/phpyaz>.



## Ablauf der automatisierten Dateneinspielung

Das Shell-Skript CBS-Cron.sh wird mit Hilfe von Cron<sup>142</sup> täglich zu einer vorgegebenen Uhrzeit automatisch gestartet und arbeitet folgende Befehle ab:

- Das Skript z3950fetch.php wird aufgerufen, welches die am Vortag erstellten Datensätze herunterlädt und diese in einer MARC21-Datei nach dem Muster day-JJJJ-MM-TT.mrc speichert. Dieser Vorgang wird in der Datei z3950fetch.log protokolliert.
- Die heruntergeladene Datei wird vom VuFind-Importer in den Solr-Index importiert. Dieser Prozess wird in Datei import.log protokolliert.
- Die importierte MARC21-Datei wird in ein Archiv-Verzeichnis für importierte Datensätze verschoben.
- Der Solr-Index wird durch das Skript optimize.php, welches im Verzeichnis vufind/util liegt, optimiert.<sup>143</sup>

## Nachnutzung von z3950fetch

Die Zugangparameter zum Z39.50-Servers des GBV sowie eine Übersicht mit allen erreichbaren Datenbanken sind auf den Webseiten des GBV veröffentlicht.<sup>144</sup>

Das Z39.50-Target-Profil des Bibliothekskatalogs der HAW Hamburg lautet:

„HH\_HAW, hh\_haw; 2@lhopc4.rz.uni-hamburg.de“.<sup>145</sup>

Ausgehend von der Konfiguration der TUBHH musste zur Anpassung des Skriptes für die Daten des HIBS lediglich harb\_opc durch hh\_haw ersetzt werden.

```
$user="999";  
$passwd="abc";  
$host="z3950.gbv.de:20012/harb_opc"; // host port database  
$syntax="marc21"; // marc21 sutrs, usmarc, grs1, xml, opac?
```

Listing 4: Konfiguration von z3950fetch.php

<sup>142</sup> Cron ist ein Dienst auf unix-artigen Betriebssystemen, um Programme und Skripte automatisch zu festgelegten Zeiten auszuführen (UBUNTU USERS WIKI 2010 - Cron).

<sup>143</sup> VuFind Documentation 2010 – Performance.

<sup>144</sup> GBV 2010 - Z39.50 Target-Profile.

<sup>145</sup> GBV 2010 - Z39.50 HAW.

### **9.3.2. Makulierung**

Für die VuFind-Installation der TUB HH war zum Zeitpunkt des Experteninterviews bisher noch keine Lösung zur automatischen Löschung von Titeln realisiert. Mittlerweile existiert auch für dieses Problem eine Lösung, die von Oliver Marahrens im Rahmen der GBV-Verbundkonferenz am 09.09.2010 vorgestellt wurde. Durch die Auswertung des Online-Update-Fetch kann abgeglichen werden, welche Datensätze nur in VuFind, nicht aber im lokalen Katalog vorhanden sind; die entsprechenden Datensätze werden anschließend gelöscht.<sup>146</sup> Die Nachnutzung war im gesetzten Projektzeitrahmen jedoch nicht mehr zu realisieren.

#### **Erforderliche Kenntnisse zum automatisierten Datenimport**

Um die Lösung des automatischen Datenimports der TUB HH nachnutzen zu können, waren folgende Kenntnisse erforderlich:

- Installation von PHP-Erweiterungen
- Programme zeitgesteuert per Cron starten<sup>147</sup>
- Anpassung der Skripte CBS-Cron.sh und z3950fetch.php an die lokalen Gegebenheiten

#### **Einstufung nach den Kriterien der Machbarkeit: 2b**

---

<sup>146</sup> Vgl. MARAARENS 2010 – TUBFind.

<sup>147</sup> UBUNTU USERS WIKI 2010 – Cron.

## 9.4. Anzeige von Verfügbarkeitsinformationen

Im Kapiteln „Import von Metadaten“ und „Automatisierter Metadatenabgleich“ wurde beschrieben, wie statische Katalogdaten in den bachelopac importiert werden. Diese Katalogdaten beinhalten jedoch noch keine aktuellen Verfügbarkeitsinformationen zu einzelnen Exemplaren.

<b>Standort:</b>	<a href="#">Fachbibliothek Technik Wirtschaft Information 2 (Berliner Tor 7)</a>
<b>Signatur:</b>	<b>Dat 855 50/5.A.</b>
<b>Ausleihstatus:</b>	Ausleihbestand ausgeliehen → <a href="#">Vormerken</a>

Abbildung 8: Anzeige der Verfügbarkeit eines Exemplars im PICA-OPAC LBS4 der HAW Hamburg

Das Auslesen von Verfügbarkeitsinformationen aus proprietären Bibliothekssystemen ist aufgrund der (meist) fehlenden Schnittstellen ein generelles Problem.<sup>148</sup> Hinzu kommt, dass der Begriff „Verfügbarkeit“ in fast allen Katalogen unterschiedlich definiert wird.<sup>149</sup> Ein Lösungsansatz zur Einbindung von Verfügbarkeitsinformationen in externe Recherchewerkzeuge ist die Document Availability Information API (DAIA). Ziel von DAIA ist es, „eine einfache und einheitliche Methode zu schaffen, um bei [einem] beliebige[n] Bibliothekssystemen anzufragen, ob ein bestimmter Titel zur Zeit verfügbar, d.h. einsehbar oder ausleihbar, ist.“<sup>150</sup> Neben der Verfügbarkeit eines Mediums kann mit DAIA auch der Standort, die Signatur, sowie das voraussichtliche Rückgabedatum eines ausgeliehenen Exemplars und das voraussichtliche Bereitstellungsdatum eines bereits bestellten Exemplars dynamisch abgefragt werden. DAIA wurde von Anne Christensen (SUB Hamburg/beluga), Jakob Voß (GBV) und Uwe Reh (HeBIS) entwickelt.<sup>151</sup>

Da das PICA-LBS keine DAIA-Schnittstelle besitzt, muss eine Art Adapter (Wrapper) verwendet werden, der die DAIA-Anfragen entgegennimmt, diese an den PICA-Katalog- und das Ausleihsystem weiterleitet, und die Ergebnisse an die DAIA-API zurückgibt.<sup>152</sup> „Langfristig soll DAIA soweit etabliert werden, dass alle

<sup>148</sup> Vgl. STEHLE 2010 – OPAC mit VuFind.

<sup>149</sup> REH 2009 – DAIA, S. 5.

<sup>150</sup> GBV Wiki 2010 – Verfügbarkeitsrecherche mit DAIA.

<sup>151</sup> REH 2009 – DAIA, S. 6.

<sup>152</sup> GBV WIKI 2010 – DAIA-Wrapper.

Hersteller DAIA in ihre Kataloge integrieren können.“<sup>153</sup> Sämtliche DAIA-Implementierungen sind unter der GPL lizenziert.

### 9.4.1. DAIA-Datenmodell

„Der Begriff „Verfügbarkeit“ ist in DAIA so definiert, dass ein Exemplar (item) eines Mediums (document) jeweils unabhängig voneinander für verschiedene Dienstleistungen (service) momentan entweder verfügbar (available) oder nicht verfügbar (unavailable) sein kann.“<sup>154</sup>

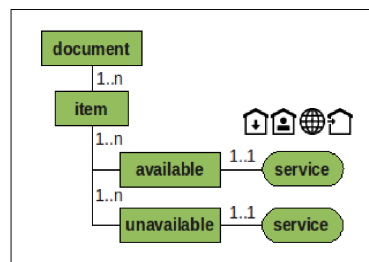


Abbildung 9: Schematische Darstellung des DAIA-Datenmodells  
(Voss 2010 – Verfügbarkeitsinformationen)

Dabei werden folgende Services bzw. Arten der Verfügbarkeit unterschieden:





Art der Verfügbarkeit	Codierung	Symbol
Lokale Einsehbarkeit/Präsenzbestand	presentation	
Ausleihbarkeit	loan	
Freier Zugriff	openaccess	
Per Fernleihe bestellbar bzw. vermittelter Zugriff	interloan	

Tabelle 1: Kodierung der DAIA-Services  
(in Anlehnung an Voss 2010 - Verfügbarkeitsinformationen)

<sup>153</sup> REH 2009 – DAIA, S. 6.

<sup>154</sup> GBV WIKI 2010 – Verfügbarkeitsrecherche mit DAIA.

### 9.4.2. Funktionsweise der DAIA-API in der VuFind-Umgebung

Oliver Marahrens entwickelte zur Implementierung der DAIA-API in TUBFind einen DAIA-Treiber, den Server phpDAIA<sup>155</sup>, sowie einen Wrapper für das PICA-System der TUB HH.

Der DAIA-Treiber, der als Client fungiert, stellt über die PICA-Produktionsnummer (PPN) eine Anfrage an den DAIA-Server. Der DAIA-Server liest über einen Wrapper die Verfügbarkeitsinformationen, sowie die Signatur und den Standort der zur angefragten PPN zugehörigen Exemplare aus dem lokalen PICA-System aus und gibt diese im DAIA/XML-Format an den DAIA-Treiber zurück. Der DAIA-Treiber wandelt die Informationen wiederum so um, dass VuFind diese anzeigen kann.

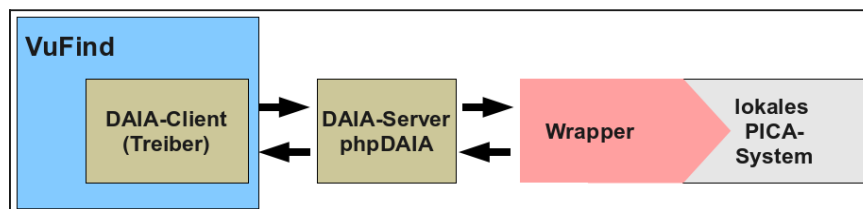


Abbildung 10: Schematische Darstellung des Datenflusses zum Auslesen von Verfügbarkeitsinformationen  
(in Anlehnung an Voss 2009 - Einheitlichen Kodierung von Verfügbarkeitsinformationen)

<sup>155</sup> MARAARENS 2010 – Verfügbarkeitsinformationen.

Eine Abfrage an den DAIA-Server ist nach dem Muster

`http://server/pfad-zur-DAIA-API/daia_ws.php?ppn=<PPN-Nummer-aus-dem-Katalog>+ [&format=xml]`

aufgebaut. Es besteht auch die Möglichkeit, Verfügbarkeitsabfragen über einen Browser auszuführen.

**Document Availability Information API**

This page displays a DAIA response to report availability information of documents. The document has timestamp 2010-08-07T22:55:01+02:00 and it is encoded in DAIA/XML version 0.5. The full XML source is [shown below](#).

**Document**

The response contains information about 3 item[s] of document [ask:ppn:509407382](#).

**Availability**

Document	Item	Location	local presentation	loan	open access	Interloan
<a href="#">ask:ppn:509407382</a>	<a href="#">Dat+855+29/2.A</a> eIn:89353207x	<a href="#">Fachbibliothek TWI 2 (Berliner Tor 7)</a>	● unavailable 23-08-2010	● unavailable 23-08-2010		● unavailable 23-08-2010 201%40/01 aAusleihbestand%7Causgeliehen b2 e89353207x mmon f2 http://kataloge.uni-hamburg.de:80/loan/RES%3FEPN%3D893532 uAusleihbestand vausgeliehen
	<a href="#">Dat+855+29/2.A+8</a> eIn:898146283	<a href="#">Medienservice Finkenu/TWI1</a>	● available	● unavailable Semesterapparat Schulz / Usability Labor	● unavailable	● unavailable 201%40/02 aPraesenzbestand%2C+Ausleihmoeglichkeit+erfragen%7Cverfu b0 e898146283 mmon f2 uPraesenzbestand%2C+Ausleihmoeglichkeit+erfragen vverfuegbar.+Befinden+Sie+sieh+vor+Ort%2C+bitte+Buch+selbst wBestellen+ist+nicht+moeglich%2C+wenden+Sie+sieh+an+das+F
	<a href="#">Dat+855+29/2.A+h</a> eIn:1102924639	<a href="#">Fachbibliothek TWI 2 (Berliner Tor 7)</a>	● unavailable 16-08-2010	● unavailable 16-08-2010 Semesterapparat		● unavailable 16-08-2010 201%40/03 aAusleihbestand%7Causgeliehen

Abbildung 11: HTML-Ausgabe des DAIA-Servers zur Prüfung der Verfügbarkeit einer PPN

### 9.4.3. Nutzung von DAIA im bachelopac

Die Verfasser konnten die Anzeige der Verfügbarkeitsinformationen im bachelopac mit DAIA nicht ohne Hilfe realisieren. Daher wurde Oliver Marahrens um aktive Unterstützung bei Umsetzung gebeten.

#### Installation von phpDAIA

Um das Paket phpDAIA auf dem Server zu installieren, musste das Paket `phpdaia-0.1.tar.gz`<sup>156</sup> nur in einem von außen aufrufbaren Verzeichnis des Webservers entpackt werden. Im Fall des bachelopacs wurde phpDAIA im Verzeichnis `/var/www/phpDAIA` installiert.

#### Konfiguration der Datei DAIA.ini

Damit der DAIA-Server mit dem PICA-System kommunizieren kann, musste die Konfiguration der Datei DAIA.ini an das LBS der HAW Hamburg, welches von der SUB Hamburg gehostet wird, angepasst werden. Der Aufbau der URLs ist in der VuFind-Dokumentation beschrieben.<sup>157</sup> Die Parameter für die HAW Hamburg konnten von der Beispielkonfiguration für die TUB HH abgeleitet werden. Diese Aufgabe hätte im Nachhinein betrachtet auch ohne Unterstützung gelöst werden können.

```
[Catalog]
basicUrl = "http://kataloge.uni-hamburg.de/DB=2/PPN?PPN="
picaPlusUrl = "http://kataloge.uni-hamburg.de/DB=2/SET=1/TTL=1/PRS=PP
%7F/PPN?PPN="
```

Listing 5: Anbindung von DAIA an das PICA-LBS (DAIA.ini)

<sup>156</sup> Mittlerweile (Stand: 30.08.2010) ist die Version 0.2 verfügbar unter <http://sourceforge.net/projects/daia/files/phpdaia-0.2.tar.gz/download>.

<sup>157</sup> Vgl. VUFind DOCUMENTATION 2010 – PICA/DAIA.

### **Anpassung des Wrappers**

Der Wrapper `pica2daia.php` wurde für den Betrieb an der TUB Hamburg-Harburg konzipiert. Zur Nachnutzung mussten Anpassungen im PHP-Quellcode vorgenommen werden. Dies betraf beispielsweise das Auslesen des Rückgabedatums ausgeliehener Exemplare mit der Screen-Scraping-Methode.<sup>158</sup> Diese Aufgabe hätte von den Verfassern ohne die aktive Unterstützung von Oliver Marahrens nicht gelöst werden können.

### **Setzen eines „Vormerken“-Links in den PICA-Katalog**

Die Integrierung der PICA-Nutzerkonten in den `bachelopac` wurde im Rahmen des Projekts nicht angestrebt. Es wurde behelfsmäßig ein dynamischer Link zur Vormerk-Funktion in den PICA-Katalog gesetzt, um den Nutzern des `bachelopacs` die Vormerkung eines entliehenen Exemplars ermöglichen zu können.<sup>159</sup> Hierzu musste die von DAIA zur Verfügung gestellte Variable `recallhref` in das Template `view-holdings.tpl` integriert werden. Auch diese Aufgabe hätte ohne die aktive Unterstützung von Oliver Marahrens nicht gelöst werden können.

### **9.4.4. Anzeige der Standorte**

Die Angabe der Standorte der Medien liegen in PICA nicht im Klartext, sondern nur in Form von Bibliothekssigeln<sup>160</sup> vor. Um diese Sigel in eine für Nutzer lesbare Information zu übertragen, müssen entsprechende Übersetzungen angelegt werden. Die Bibliothekssigel der Teilbibliotheken konnten durch im Bibliotheksführer der SUB Hamburg<sup>161</sup> bzw. dem ISIL- und Sigelverzeichnis der ZDB<sup>162</sup> ermittelt werden. Nicht-recherchierbare Bibliothekssigel konnten durch eine Nachfrage bei der Abteilung Digitale Dienste des HIBS zugeordnet werden.

---

<sup>158</sup> Die HTML-Ausgabe des PICA-Katalogs der HAW Hamburg muss als „Schnittstelle“ zur Extraktion der benötigten Information benutzt werden.

<sup>159</sup> Die Zuordnung erfolgt über den Barcode eines Exemplars.

<sup>160</sup> Eine Ausnahme bildet der Code HIBS-E, der elektronische Medien kennzeichnet, die nur im IP-Netzwerk der HAW Hamburg zugänglich sind.

<sup>161</sup> <http://www.sub.uni-hamburg.de/bibliotheksfuehrer.htm>.

<sup>162</sup> <http://dispatch.opac.d-nb.de/DB=1.2/>.



Die Struktur der Übersetzungsdatei yourlocationfiles.conf lautet dabei wie folgt:

Code ; deutsche Übersetzung ; englische Übersetzung ; URL

```
18/302-a;Fachbibliothek TWI 1 (Berliner Tor 5);Fachbibliothek Technik  
Wirtschaft Information 1 (Berliner Tor 5);http://www.sub.uni-  
hamburg.de/bibitem.shtml?id=83  
  
18/302-b;Medienservice Finkenau/TWI1; Media Service Finkenau / TWI1;  
http://www.bui.haw-hamburg.de/;  
  
18/287; Fachbibliothek Soziale Arbeit und Pflege (Alexanderstr. 1);  
Fachbibliothek Soziale Arbeit und Pflege (Alexanderstrasse);  
http://www.sub.uni-hamburg.de/bibitem.shtml?id=89  
  
18/302; Fachbibliothek TWI 2 (Berliner Tor 7);Fachbibliothek Technik  
Wirtschaft Information 2 (Berliner Tor 7); http://www.sub.uni-  
hamburg.de/bibitem.shtml?id=95;  
  
18/285; Fachbibliothek Design (Armgartstraße); Fachbibliothek Design  
(Armgartstrasse); http://www.sub.uni-hamburg.de/bibitem.shtml?id=87;  
  
18/284; Fachbibliothek Life Sciences (Bergedorf);Fachbibliothek Life  
Sciences (Bergedorf); http://www.sub.uni-hamburg.de/bibitem.shtml?id=85;  
  
HIBS-E; Zugriff nur im Netzwerk der HAW-Hamburg bzw. via VPN; Access only  
via HAW-Network or VPN; http://www.haw-hamburg.de;
```

Listing 6: Codierung der Bibliotheksstandorte in der Datei yourlocationfiles.ini

### **Erforderliche Kenntnisse und Fähigkeiten zur Nutzung von phpDAIA**

- Entpacken von Archiven zur Installation von phpDAIA
- Abstraktion von Beispielkonfigurationen zur Anpassungen der URLs
- Kenntnisse über Bibliothekssigel bzw. ISIL zur Decodierung der Standorte

### **Einstufung nach den Kriterien der Machbarkeit: 3**

Um die beschriebenen Anpassungen ohne aktive Unterstützung lösen zu können, wären PHP-Kenntnisse zur Anpassung des Wrappers und zur Anpassung der VuFind-Templates view-holdings.tpl erforderlich gewesen.

## 9.5. Die Konfiguration von VuFind

### 9.5.1. Anpassung der Facetten

Facettiertes Browsen ist eine Navigationsform, die es den Nutzern ermöglicht, eine unübersichtliche Treffermenge nach formalen und inhaltlichen Kriterien sukzessive auf das gewünschte Ziel einzuschränken. Die einzelnen Facetten bieten jeweils einzelne, aber wesentliche Aspekte des gesuchten Ergebnisses ab und setzen somit einen assoziativen Suchprozess in Gang.<sup>163</sup>

Auf der technischen Ebene liegt jeder Facette ein Solr-Index-Feld zugrunde. Anzuzeigende Facetten werden in der Datei `vufind/web/conf/facets.ini` konfiguriert und werden mit folgender Struktur eingebunden:

Name des Solr-Index-Feldes = Angezeigter Name der Facette<sup>164</sup>

Die Facetten werden zunächst englischsprachig benannt. Übersetzungen in verschiedene Sprachen werden in den jeweiligen Sprachdateien unter `vufind/web/lang/` abgelegt, um eine dynamische Übersetzung, abhängig von der vom Benutzer ausgewählten Sprache der Katalogoberfläche, zu ermöglichen.

Die Facetten Institution, Standort, Autor, Format, Genre, Sprache, Library-of-Congress- Klassifikation und Ära sind bereits vorkonfiguriert und funktionieren out of the box, sofern die entsprechende Datengrundlage im Index vorhanden ist. Nicht benötigte Facetten werden durch ein vorangestelltes Semikolon auskommentiert. Die Reihenfolge der Zeilen in der Datei `facets.ini` entspricht der Facettenanordnung in der der Katalogoberfläche.

---

<sup>163</sup> Vgl. CHRISTENSEN 2009 - Partizipative Entwicklung, S. 532 sowie und IMHOF 2006 - RSWK/SWD und Faceted Browsing, S. 4.

<sup>164</sup> Vgl. VUFIND DOCUMENTATION 2010 - adding facets.

```

[CheckboxFacets]
;edition:1st* = "First Edition"      ; Contrived hypothetical example
hochschulschriften:Hochschule für Angewandte Wissenschaften = "theses only"

[Results]
;institution      = Institution
;building        = Library
authorStr        = Author
publishDate      = Publish Date
bklname = "Library classification"
;callnumber-first = "Call Number"
language         = Language
format          = Format
genre_facet     = Genre
geographic_facet = Region
;era            = Era

```

Listing 7: Auszug aus der angepassten Konfigurationsdatei facets.ini

### 9.5.1.1. Deaktivierung nicht benötigter Facetten

Da der bachelopac nur für die HAW Hamburg konzipiert ist, erübrigt sich die Anzeige der Facette „Institution“; hier ergeben sich keine Einschränkungsmöglichkeiten. Für eine Standortfacette („Library“) gab es im Solr-Index noch keine direkt verwendbare Datengrundlage. Die Idee, eine solche (möglicherweise auf Basis des MARC21-Feldes 900\$d), zu schaffen, wurde aufgrund des begrenzten Projektzeitrahmens nicht weiterverfolgt. Nur ein kleiner Teil der Bestände weist eine LOC-Klassifizierung auf, daher wurde auf die LOC-Facette, auch im Hinblick auf die später erfolgende Implementierung der Facette „Basisklassifikation“, verzichtet.

### 9.5.1.2. Facette „Erscheinungsjahr“

Aufgrund von mangelnden Daten, Inhomogenitäten und Überschneidungen erwies sich die vorkonfigurierte Facette „Ära“ auf Basis der MARC21-Felder 600\$d, 610\$y, 611\$y, 630\$y, 648\$a, 648\$y, 650\$y, 651\$y und 655\$y<sup>165</sup> als unbrauchbar. Als Alternative wurde eine neue Facette „Erscheinungsjahr“ auf Basis des bereits vorhandenen Solr-Index-Feldes publishDate erzeugt.

Ära	Erscheinungsjahr
20th century (155)	2008 (6633)
21st century (47)	2007 (6274)
1990- (31)	2006 (5863)
19th century (28)	1998 (5647)
18th century (9)	2009 (5622)
1927-1998 (9)	2004 (5591)
1945- (9)	1999 (5476)
1939- (6)	2003 (5354)
1945-1990 (6)	2005 (5346)
1960- (5)	2002 (5338)

Abbildung 12: Facetten Ära und Erscheinungsjahr im direkten Vergleich

Eine einfache Konfigurationsoption, um die Inhalte der Facetten alphabetisch absteigend anstatt nach der Trefferanzahl darstellen zu können, existiert leider noch nicht. Dieses Problem wurde im Kontext der Verfasser-Facette auf der Mailingliste VuFind-General diskutiert. Die Verbesserung der Facettenfunktionalitäten in zukünftigen VuFind-Releases hat laut der VuFind Projektmanagement- und Fehlerverwaltungsplattform VuFind Jira hohe Priorität.<sup>166</sup>

<sup>165</sup> Das Metadatenmapping wird in über die Dateien vufind/import/marc.properties und marc\_local.properties konfiguriert.

<sup>166</sup> Vgl. VUFIND JIRA #177 2010 – facet behavior.

### 9.5.1.3. Ergänzung einer Checkbox-Facette

Um die gezielte Suche nach Diplom-, Bachelor- und Masterarbeiten der HAW zu vereinfachen, wurde eine Checkbox-Facette realisiert, mit der die Treffermenge entsprechend eingegrenzt werden kann.



Abbildung 13: Checkbox-Facette zur Einschränkung der Ergebnisse auf Abschlußarbeiten

Datenbasis dieser Facette ist MARC21-Feld 502\$a („Dissertation Note“).<sup>167</sup> Da Feld 502\$a standardmäßig nicht indiziert wird, waren als Vorarbeit die Schritte 1 und 2 notwendig:

#### 1. Erweiterung des Index-Schemas

Zunächst wurde das Index-Schema in der Datei vufind/solr/biblio/schema.xml um das Feld „hochschulschriften“ wie folgt erweitert:

```
<field name="hochschulschriften" type="text" indexed="true" stored="true"
multiValued="true"/>
```

#### 2. Metadatenmapping

Um das Solr-Index-Feld hochschulschriften beim Import mit dem MARC21-Feld 502\$a zu befüllen, wurde die Datei usr/local/vufind/import/marc\_local.properties um folgende Zeile ergänzt:

```
hochschulschriften = 502a
```

#### 3. Aktivierung der Facette

Nach einer Neuindexierung aller Datensätze<sup>168</sup> konnte die Checkbox-Facette in der Datei facets.ini mit der Zeile

```
hochschulschriften:Hochschule für Angewandte Wissenschaften = "theses only"
```

<sup>167</sup> LIBRARY OF CONGRESS 2010 - MARC 21.

<sup>168</sup> VUFIND DOCUMENTATION 2010 - re-indexing.

im Abschnitt [CheckboxFacets] der Datei facets.ini aktiviert werden.

#### 9.5.1.4. Ergänzung der Facette „Basisklassifikation“

Um die (niederländische) Basisklassifikation, die im Bereich des GBV und damit auch vom HIBS verwendet wird, als Facette abzubilden, muss das MARC21-Feld 084 („Other Classification Number“) ausgewertet werden. 084 ist für alle Klassifikationen reserviert, für die keine eigenen MARC21-Felder vorgesehen sind.<sup>169</sup> Das Feld kann in einem Datensatz von verschiedenen Bibliotheksklassifikationen mehrfach belegt werden. Während das Subfeld 084\$a die Notation sowie die Klassenbenennung enthält, ist im Subfeld 084\$2 der Name der jeweiligen Bibliotheksklassifikation codiert.

084	1a 50.31 ; Technische Mechanik	12 bcl
084	1a UF 1500	12 rvk

Abbildung 14: Darstellung des MARC-Feldes 084 in der „Internansicht“ von VuFind

Der in Abbildung 14 gezeigte Ausschnitt eines Datensatzes aus dem bachelopac enthält zwei verschiedene Klassifikationen: Die Basisklassifikation (bcl, engl. basic classification) und die Regensburger Verbundklassifikation (rvk).

Um die Basisklassifikation für eine Facette zu nutzen zu können, muss eine Fallunterscheidung erfolgen: Bei der Indexierung muss anhand des Feldes 084\$a zunächst geprüft werden, ob im Feld 084\$a überhaupt eine Notation und eine Klassenbenennung der BKL vorliegt.

Den Verfassern war es mangels ausreichender Programmierkenntnisse nicht möglich, eine solche Fallunterscheidung bei der Indexierung selbst zu realisieren. Deshalb wurde Oliver Marahrens zu Nachnutzungsmöglichkeiten des Codes der BKL-Facette des TUBFind-Katalogs befragt. Den Verfassern wurde das Mapping-Script bcl.sh zur Nachnutzung überlassen, mit dem die BKL aus den MARC21-Datensätzen beim Import gefiltert und in den Solr-Index importiert werden kann.<sup>170</sup>

<sup>169</sup> Vgl. LIBRARY OF CONGRESS 2010 - MARC 21.

<sup>170</sup> Das Script bcl.sh und eine Kurzanleitung zur Indexierung der BKL ist seit dem 14.09.2010 auch im TUBFind Blog zugänglich unter <http://www.tub.tu-harburg.de/tubfind-blog/2010/09/hilfreiche-vufind-scripte/>.

Um die Facette Basisklassifikation zur Anzeige zu bringen, waren folgende Schritte notwendig:

### 1. Erweiterung des Index-Schemas

Um eine spätere Aufspaltung des MARC21-Feldes 084\$a vorzubereiten, mussten im Index-Schema drei neue Felder angelegt werden: `bklnumer` soll die Notation enthalten, `bklname` die Klassenbenennung und `bkl` sowohl die Notation als auch die Klassenbenennung.

```
<field name="bklnumer" type="string" indexed="true" stored="true"
multiValued="true"/>
<field name="bklname" type="string" indexed="true" stored="true"
multiValued="true"/>
<field name="bkl" type="string" indexed="true" stored="true"
multiValued="true"/>
```

Listing 8: Solr-Index-Felder für die BKL (schema.xml)

### 2. Metadatenmapping

Um das Metadatenmapping mit dem Skript `bcl.bsh` behandeln zu können, waren folgende Eintragungen in der Datei `vufind/import/marc_local.properties` notwendig:

```
bkl = script(bcl.bsh), getBkl
bklnumer = script(bcl.bsh), getBklNumber
bklname = script(bcl.bsh), getBklName
```

Listing 9: Eintrag des Mapping-Skripts `bcl.sh` in der Datei `marc_local.properties`

Das Skript `bcl.bsh` prüft, ob `084$2` den Eintrag `bcl` aufweist. Falls dies der Fall ist, wird Feld `084$a` in das Solr-Index-Feld `bkl` indiziert. Der Inhalt des MARC21-Feldes `084$a` wird anhand des darin enthaltenen Semikolons aufgesplittet. Die Notation wird auf den Solr-Index `bklnumer`, die Klassenbenennung auf den Solr-Index `bklname` importiert.

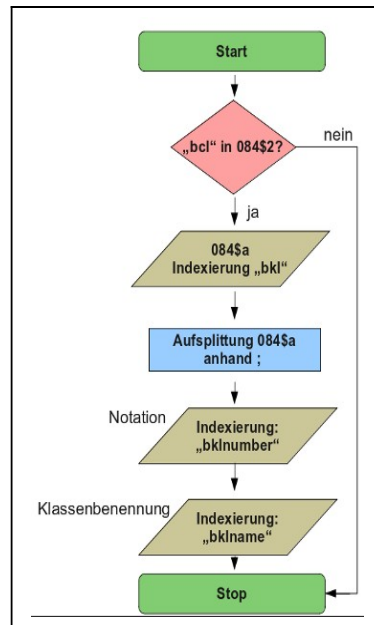


Abbildung 15: Vereinfachter Programmablaufplan zur Indexierung der BKL

### 3. Aktivierung der Facette

Nach einer Neuindexierung aller Datensätze wird in der Datei facets.ini die Zeile  
bkname = "Library classification"

im Abschnitt [Results] eingetragen, um die Facette auf Basis der  
Klassenbenennungen der BKL in der Trefferliste zu aktivieren.

<p><b>BKL</b></p> <ul style="list-style-type: none"> <li>Programmiersprachen (1717)</li> <li>Rechnerkommunikation (1466)</li> <li>Künstliche Intelligenz (1326)</li> <li>Technische Mechanik (1267)</li> <li>Technische Strömungsmechanik (1063)</li> <li>mehr ...</li> </ul>	<p><b>Basisklassifikation</b></p> <ul style="list-style-type: none"> <li>Bildende Künstler (940)</li> <li>Marketing (625)</li> <li>Unternehmensführung (555)</li> <li>Programmiersprachen (547)</li> <li>Modedesign (443)</li> <li>mehr ...</li> </ul>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Abbildung 16: BKL-Facetten von TUBFind und bachelopac



## Exkurs: Aktivierung der Basisklassifikation in der erweiterten Suche

Da die Basisklassifikation inklusive der Notation auch in der erweiterten Suche verfügbar sein soll, wird in der Datei facets.ini im Abschnitt [Advanced] der Datei facets.ini die Zeile

bkl = "Library classification"

eingetragen.

The screenshot shows a search filter interface with three main sections: 'Klassifikation:', 'Sprache:', and 'Medientyp:'. Each section contains a list of options with a small 'iii' icon to the right of the list. Below these sections are radio buttons for 'Abbildungen:' and a 'Suchen' button.

Klassifikation:	Sprache:	Medientyp:
18.04 ; Englische Sprache	Russisch	Audio
20.00 ; Kunstwissenschaften: Allgemeines	Salishan	Buch
20.31 ; Bildende Künstler	Schwedisch	Elektronisch
21.37 ; Graphikdesign	Slovenisch	Karte
21.42 ; Geschichte der Photographie	Slowakisch	Mikrofilm
21.82 ; Industriedesign	Spanisch	Objekt
21.85 ; Modedesign	Sundanese	Paket
31.00 ; Mathematik: Allgemeines	Swahili	Partitur
31.73 ; Mathematische Statistik	Türkisch	Schriftenreih
31.76 ; Numerische Mathematik	Tigrinya	Software

Abbildungen:  Mit Abbildungen  
 Ohne Abbildungen  
 Keine Vorgabe

Suchen

Abbildung 17: BKL in der erweiterten Suche

## 5. Anordnung, Benennung und Übersetzung der Facetten

Da die Anordnung der Zeilen in der Datei facets.ini der Reihenfolge der Facetten in der Katalogoberfläche entspricht, kann durch ein Verschieben der Zeilen die Anzeigeposition leicht verändert werden. Die Anpassung der Facettenbenennungen erfolgte in Anlehnung an die Redesign-Empfehlungen für die Rechercheplattform beluga im Rahmen der Usability-Tests am Department Information der HAW Hamburg im Juli 2009.<sup>171</sup> Da mehrere Probanden Schwierigkeiten mit den Facettenbenennungen „Format“ und „Genre“ (O-Ton einer Testperson: „Da kann ich mir gar nichts drunter vorstellen“) hatten, wurden diese im bachelopac in „Medientyp“ und „Literatur-Genre“ umbenannt.<sup>172</sup>

Die Inhalte der Facetten Sprache, Medientyp und Literatur-Genre wurden zunächst nur in englischer Sprache angezeigt und mussten daher noch übersetzt werden. Um eine dynamische Übersetzung, abhängig von der vom Benutzer ausgewählten

<sup>171</sup> SCHMITT, STEHLE 2009 – Stimmen aus dem Usability-Labor.

<sup>172</sup> Die Verständlichkeit dieser alternativen Bezeichnungen müsste in weiteren Usability-Tests überprüft werden.

Sprache, zu gewährleisten, wurde die Übersetzungsdatei für die deutsche Sprache vufind/web/lang/de.ini entsprechend erweitert. In derselben Datei wurden auch die deutschsprachigen Übersetzungen der Facettenbenennungen angepasst.

### **Erforderliche Kenntnisse zur Anpassung der Facetten**

Um die Facetten wie beschrieben zu erweitern und zu konfigurieren, waren folgende Kenntnisse erforderlich:

- Kenntnisse über den Aufbau und die Bedeutung von Klassifikationen
- Verschieben und Auskommentieren von Zeilen der Konfigurationsdatei facets.ini , um die Position von vorhandenen Facetten zu ändern oder diese zu deaktivieren. Um neue Facetten einzutragen, muss der Solr-Index-Name bekannt sein, der der Facette zugrunde liegt.
- Kenntnisse des MARC21-Regelwerks, um die Datengrundlagen für Facetten ausmachen zu können.

### **Einstufung nach den Kriterien der Machbarkeit: 2a bzw. 2b (BKL)**

## 9.5.2. Suchsyntax von Lucene

Die Lucene-Suche unterstützt vielseitige Suchoptionen, kann aber bei Bedarf auch um eine eigene Suchsyntax ergänzt werden.<sup>173</sup> Suchsyntax meint hier die Such- oder Abfragesprache. Einzelne oder mehrere Suchbegriffe werden nach bestimmten Regeln zu Suchformulierungen zusammengesetzt und in die Suchmaske eingegeben. Eine Suchanfrage besteht aus Ausdrücken und Operatoren.

Die Standard-Suchsyntax von Lucene, die auch in VuFind eingesetzt wird, umfasst neben der Verwendung von Boolesche Operatoren (AND, OR, NOT) und der Phrasensuche mit Anführungszeichen, folgende Suchoptionen:

### Fuzzy-Suche oder unscharfe Suche

Unter der Verwendung des Levenshtein-Distanz-Algorithmus besteht die Möglichkeit mit Lucene eine unscharfe Suche durchzuführen. Vereinfacht gesagt berechnet der Levenshtein-Distanz-Algorithmus wie viele Operationen es in einer Zeichenkette durch Einfügen, Löschen und Ersetzen bedarf, um mit einer anderen Zeichenkette übereinzustimmen.

Eine unscharfe Suche kann mit Hilfe des Operators ~ (Tilde) durchgeführt werden, der am Ende des Ausdrucks hinzugefügt wird. So lassen sich durch die Suche nach Dostoevskij~ auch die Schreibweisen Dostoiewsky, Dostojewski, Dostojewskij etc. finden.



Abbildung 18: Suche nach „Dostoevskij“ im bachelopac (1 Treffer)

<sup>173</sup> HEIDENREICH 2003 – Lucene



Abbildung 19: Auszug der Trefferliste der unscharfen Suche nach "Dostoevskij~" im bachelopac

Treffer, die unter Verwendung der Fuzzy-Suche gefunden werden, sind mit dem Verstärkungsfaktor 0,2 gewichtet, das heißt, es werden erst die Treffer angezeigt, die exakt mit dem gesuchten Ausdruck übereinstimmen und dann erst die Treffer der Fuzzy-Suche.

### Feldsuche

Suchanfragen, die mit Lucene durchgeführt werden, lassen sich auch auf einzelne Felder einschränken. Die Syntax lautet hierfür

Feldname: Suchbegriff.

Sucht man nach Titeln, in denen der Begriff Bibliothek vorkommt, lautet die richtige Anfrage „title:digitale“. Eine Feldsuche lässt sich für jedes in der Datei vufind/solr/biblio/conf/schema.xml indexiertes Feld durchführen.

Zu beachten ist aber, dass immer nur der erste Ausdruck hinter einem Doppelpunkt einer Feldsuche unterzogen wird. Sucht man beispielhaft nach einem Titel in dessen Namen sowohl „digitale“ als auch „bibliothek“ vorkommen sollen, lautet die richtige Suchanfrage „title:digitale AND title:bibliothek“ und nicht „title:digitale bibliothek“.

### Distanzsuche

Auch eine Distanzsuche ist unter Verwendung des Operators ~ (Tilde) und der Angabe der maximalen Distanz, die zwischen zwei Ausdrücken bestehen, die durch Anführungszeichen umschlossen werden, möglich. Die Phrasensuche wird also insoweit verändert, dass die gesuchten Ausdrücke nicht mehr unmittelbar nebeneinander stehen müssen sondern in einem frei zu definierendem Abstand.

## Verstärkungsfaktoren

Lucene erlaubt es auch aktiv in das Relevance Ranking einzugreifen. Bestimmten Suchbegriffen kann ein zusätzlicher Verstärkungsfaktor zugeordnet werden. Dies geschieht durch den Operator ^ (Caret) und der Angabe des Verstärkungsfaktors.

Eine Suchanfrage nach „html css^4“ würde den Ausdruck „css“ viermal so stark gewichten wie den Ausdruck „html“. Da Verstärkungsfaktoren aber auch einen Wert annehmen können, der kleiner als 1 ist, ist auch eine negative Gewichtung einzelner Suchbegriffe möglich. Die Suchanfrage „html css^0.5“ wertet den Ausdruck „html“ also zweimal so stark wie den Ausdruck „css“.

## Trunkierung

Lucene unterstützt den Einsatz von Wildcards. Zu unterscheiden ist hier, ob nur eine Trunkierung für ein einzelnes Zeichen vorgenommen werden soll oder für mehrere Zeichen.

Der Operator für ein einzelnes Zeichen ist das ? (Fragezeichen), der Operator für mehrere Zeichen ist das \* (Asterisk). Einzuschränken ist aber, dass weder der Operator ? (Fragezeichen) noch der Operator \* (Asterisk) am Anfang eines Suchbegriffes stehen können.

Ein Problem entsteht aber bei der Trunkierung von Suchbegriffen, die Umlaute enthalten, da Lucene diese nicht modifiziert und direkt an den Index übersendet. Bei der Indexierung werden aber eben diese Umlaute so modifiziert das aus „ä“ ein „a“ wird, aus „ü“ ein „u“ etc. Mögliche Lösungen beschreibt Maharens im Blog des Katalogprojekts TUBfind.<sup>174</sup>

## Bereichssuche

Mit der Bereichssuche lassen sich zum Beispiel nur Titel finden, die zwischen den Jahren 2004 und 2007 veröffentlicht worden sind. Die korrekte Suchsyntax dafür wäre [2004 TO 2007]. Auch besteht beispielsweise die Möglichkeit mit der Suchsyntax {A TO C} nur nach Begriffen zu suchen, die mit A, B oder C beginnen.<sup>175</sup>

---

<sup>174</sup> MARAHRENS 2010 – Umlaute und Trunkierung.

<sup>175</sup> Siehe hierzu auch die Hilfefunktion von VuFind.

### **Syntaxzeichen ausschließen**

Sollte es erforderlich sein, die durch die Suchsyntax verwendeten Operatoren als Ausdruck zu verwenden, so besteht die Möglichkeit durch das Zeichen \ (Backslash) die verwendeten Operatoren auch als Ausdruck einzubeziehen.

### **DisMax Suchsyntax**

DisMax verwendet eine sehr vereinfachte Lucene-Suchsyntax. Lediglich die beiden Operatoren + (plus) und – (minus) können verwendet werden. Der Operator + (plus) vor einem Suchausdruck bedeutet, dass dieser zwingend in einem Datensatz vorkommen muss. Der Operator – (minus) bedingt, dass ein Suchausdruck nicht in einem Datensatz vorkommen darf.

Zusätzlich besteht die Möglichkeit der Phrasensuche durch die Verwendung von Anführungszeichen. Ab der Version 1.5 von Solr soll DisMax aber auch Trunkierungen verarbeiten können. Damit könnte das im Abschnitt Trunkierung beschriebene Problem der fehlenden Modifikation von Umlauten durch Lucene gelöst werden, da DisMax Suchanfragen, die Umlaute enthalten, vorher modifiziert und erst dann an den Index sendet.<sup>176</sup>

### **Erforderliche Kenntnisse zur Bereitstellung der Lucene-Suchsyntax**

Keine. Mit Ausnahme des beschriebenen Trunkierungsproblems von Suchtermini mit Umlauten funktionieren im bachelopac sämtliche Suchfunktionen out of the box. Das Trunkierungsproblem soll sich ab der Solr-Version 1.5 von selbst lösen.

<sup>177</sup>

### **Einstufung nach den Kriterien der Machbarkeit: 1**

---

<sup>176</sup> MARAHRENS 2010 - Umlaute und Trunkierung.

<sup>177</sup> Vgl. ebd.

### 9.5.3. Relevance Ranking

Das Relevance Ranking von Solr/Lucene bzw. die Trefferliste von VuFind basiert neben den vergebenen Gewichtungsfaktoren hauptsächlich auf folgenden Prinzipien:

- Je seltener ein Suchwort im Index vorhanden ist, desto höher ist sein Beitrag zum Ranking.
- Dokumente, in denen die Suchworte häufiger vorkommen, werden höher bewertet.
- Dokumente, in denen mehr Suchworte vorkommen, werden höher bewertet.
- Je länger der Feldinhalt ist, in dem ein Suchwort gefunden wurde, desto geringer ist sein Beitrag zum Ranking.<sup>178</sup>

VuFind nutzt sowohl DisMax als auch Lucene für Suchanfragen. DisMax wird für die einfache Suche genutzt. Lucene hingegen kommt zum Einsatz, wenn Boolesche Operatoren, Trunkierungen oder die erweiterte Suche genutzt werden. Die Konfiguration des Relevance Rankings von VuFind erfolgt in den Dateien `vufind/web/conf/searchspecs.yaml` und `vufind/solr/biblio/conf/solrconfig.xml`. In der Datei `searchspecs.yaml` lässt sich sowohl das Ranking der DisMax-Suche als auch der Lucene-Suche konfigurieren.<sup>179</sup> Die Gewichtung einzelner Treffer geschieht über das Zeichen ^ (Caret) und der Vergabe eines Boostingfaktors im Vergleich zu anderen von Solr indizierten Feldern der Metadaten.

---

<sup>178</sup> LANGENSTEIN, MAYLEIN 2009 – Relevanz-Ranking, S. 408, S. 2.

<sup>179</sup> Daher kann eine feldübergreifende Suche über die einfache Suche zu einem anderen Ranking als eine feldübergreifende Suche über die erweiterte Suche führen.

```

AllFields:
  DismaxFields:
    - title_short^750
    - title_full^400
    - title^500
    - title_alt^200
    - title_new^100
    - series^50
    - series2^30
    - author^300
    - contents^10
    - topic^500
    - geographic^300
    - genre^300
    - allfields

```

Listing 10: Gewichtung von Index-Feldern (searchspecs.yaml)

Es besteht die Möglichkeit, die Syntax der Standardsuche in der feldübergreifenden Suche bzw. auch in der Suche in einzelnen Feldern zu definieren. Die Booleschen Operatoren AND und OR sowie die Phrasensuche können gewichtet werden. Die Syntax der Konfiguration lautet „eckige Klammer, Operator, Kommata, Wert, eckige Klammer“.

```

QueryFields:
- series:
  - [onephrase, 500]
  - [and, 200]
  - [or, 100]

```

Listing 11: Gewichtung von Operatoren (searchspecs.yaml)



### 9.5.4. More Like This

Die VuFind-Funktion „ähnliche Einträge“ in der Titelanzeige von der Katalogoberfläche von VuFind, die andere relevante Medien zum ausgewählten Titel vorschlägt, funktioniert out of the box. Dies geschieht über die „MoreLikeThis“-Funktion von Solr, die eine Suchanfrage mit Lucene ausführt und verschiedene Solr-Index-Felder vergleicht und gewichtet.<sup>180</sup>

Vorkonfiguriert vergleicht VuFind nach dem Best-Match-Verfahren die Metadaten title, title\_short, callnumber-label, topic, language, author und publishDate. In der Datei solrconfig.xml können die zu vergleichenden Felder verändert werden oder auch deren Gewichtung. Der Parameter mlt.fl gibt die zu vergleichenden Felder an. Diese können nach Bedarf durch den Eintrag von weiteren Feldern ergänzt werden.

```
<str name="mlt.fl">title,title_short,callnumber-  
label,topic,language,author,publishDate</str>
```

Listing 12: Felder, die bei More Like This verglichen werden (searchspecs.yaml)

Mit Hilfe des Parameter mlt.fl lässt sich die einzelne Gewichtung der Indizes zueinander ändern.

```
<str name="mlt.qf">  
title^75  
title_short^100  
callnumber-label^400  
topic^300  
language^30  
author^75  
publishDate  
</str>
```

Listing 13: Gewichtung der Felder bei More Like This (searchspecs.yaml)

<sup>180</sup> Vgl. SOLR WIKI 2010 – MoreLikeThis.

Für die weitere Verbesserung der Relevanz der angezeigten vorgeschlagenen Medien, wurde die Liste der Felder um die BKL ergänzt. Gewählt wurde der Faktor 400, der auch bei der Library of Congress Call Number voreingestellt war. Die BKL wurde bereits zur zusätzlichen Erstellung einer Facette indexiert. Die Datei solarconfig.xml wurde also wie folgt verändert:

```
<str
name="mlt.fl">title,title_short,callnumber,label,topic,language,author,publishDate,bkl</str>

<str name="mlt.qf">

title^75

title_short^100

callnumber-label^400

topic^300

language^30

author^75

publishDate

BKL^400

</str>
```

Listing 14: Berücksichtigung der BKL in den "ähnlichen Treffern" (solrconfig.xml)

### 9.5.5. Boosting von Lehrbüchern

Die Verfasser waren während ihres Studiums immer wieder mit dem Katalogprojekt beluga, das unter der Führung der Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky entsteht, beschäftigt. So war die Idee bekannt, die Trefferliste insoweit zu beeinflussen, dass einführende Literatur höher gerankt wird. Die gleiche Funktion wollten die Verfasser auch in ihrem VuFind-Katalog integrieren. Suchanfragen der Nutzer müssen daher, zusätzlich zu den eingegebenen Suchtermen, mit bestimmten Begriffen erweitert werden.

Nach der Durchsicht des Quellcodes von beluga<sup>181</sup> in der Datei beluga-solrconfig.xml wurde folgende Zeile gefunden.

<sup>181</sup> Der Quellcode von beluga ist (noch) nicht frei zugänglich. Die Verfasser erhielten Einblick auf Anfrage. Siehe dazu CHRISTENSEN 2010 – Kommentar zu make or buy.

```
<str name="bq"> dc\\\:title:(Arbeitsbuch OR Arbeitshilfen OR Arbeitshilfen
OR Arbeitsmethoden OR Basiswissen OR einführende OR Einführung OR
Einführungskurs OR Einführungstext OR Einführungstexte OR Einstieg OR
Farbatlas OR Grundbegriffe OR Grundkurs OR Grundlagenwissen OR
Grundpraktikum OR Grundriss OR Grundstudium OR Grundwissen OR handbook OR
Handbuch OR Intensivkurs OR introducing OR introduction OR introductory OR
kompakt OR Kurzlehrbuch OR Kurzpraktikum OR Lehrbuch OR Leitfaden OR
Praxishandbuch OR Praxisleitfaden OR Prüfungswissen OR Repetitorium OR
Student's book OR Studienbuch OR Taschenatlas OR textbook OR Übungsaufgaben
OR Übungsbuch OR Übungsfragen OR Workbook)^400 OR
dcterms\\\:tableOfContents:[* TO *]^400 </str>
```

Listing 15: Boosting von Lehrbüchern und Werken mit Inhaltsverzeichnissen in beluga (beluga-solrconfig.xml)

Nach weiterer Recherche wurde der Parameter bq (boost query), der dazu führt, dass Suchanfragen erweitert<sup>182</sup> werden können, in den bachelopac integriert.<sup>183</sup> Dieser beeinflusst aber nur das Suchverhalten von DisMax, d.h., die Suchmodifikation erfolgt nur bei der einfachen Suche. Die in beluga gefundene Zeile wurde von den Verfassern folgendermaßen angepasst und in die Datei vufind/solr/biblio/conf/solrconfig.xml des bachelopacs integriert, sodass nicht nur das Solr-Feld title berücksichtigt wird, sondern allfields. Das Boosting von Titeln, die über ein Inhaltsverzeichnis verfügen, wurde nicht übernommen. Daher wurde die Datei solrconfig.xml des bachelopacs wie folgt angepasst:

```
<str name="bq">(Arbeitsbuch OR Arbeitshilfen OR Arbeitsmethoden OR
Basiswissen OR einführende OR Einführung OR Einführungskurs OR
Einführungstext OR Einführungstexte OR Einstieg OR Farbatlas OR
Grundbegriffe OR Grundkurs OR Grundlagenwissen OR Grundpraktikum OR
Grundriss OR Grundstudium OR Grundwissen OR handbook OR Handbuch OR
Intensivkurs OR introducing OR introduction OR introductory OR kompakt OR
Kurzlehrbuch OR Kurzpraktikum OR Lehrbuch OR Leitfaden OR Praxishandbuch OR
Praxisleitfaden OR Prüfungswissen OR Repetitorium OR Student's book OR
Studienbuch OR Taschenatlas OR textbook OR Übungsaufgaben OR Übungsbuch OR
Übungsfragen OR Workbook)^8000</str>
```

Listing 16: Lehrbuchboosting im bachelopac (solrconfig.xml)

<sup>182</sup> Diese Erweiterung geschieht mit einer OR-Verknüpfung, wobei der ursprünglich eingegebene Suchterminus zwingend in den Datensätzen der Suchergebnisse enthalten sein muss.

<sup>183</sup> SOLR WIKI 2010 – DisMaxQParserPlugin.

**Erforderliche Kenntnisse zur Anpassung des Relevance Rankings**

- Verständnis von Markup-Sprachen
- Verständnis der Bedeutung von Relevance Ranking
- Basiswissen zur Konfiguration von Solr/Lucene
- Grundkenntnisse über Funktionen der Parameter von DisMax
- Wissen, in welchen Dateien man die Parameter einstellt

**Einstufung nach den Kriterien der Machbarkeit: 2a**

## 9.5.6. Stoppworte und Synonyme

### 9.5.6.1. Stoppworte

VuFind bietet durch den Einsatz von Solr die Möglichkeit, mit Stoppwortlisten zu arbeiten. Stoppworte sind Worte innerhalb eines Dokuments bzw. innerhalb einer Anfrage, die keine semantische Bedeutung haben und das Retrieval nicht beeinflussen.<sup>184</sup> Stoppwortlisten können in der Datei `vufind/solr/biblio/conf/stopwords.txt` gepflegt werden. Diese Stoppworte werden dann bei der Indexierung von Metadaten nicht in den Index von Solr aufgenommen, ebenso werden Wörter, die auf dieser Stoppwortliste stehen, nicht in die Abfragen von Nutzern miteinbezogen. Ursprünglich war es Ziel der Verfasser, exemplarisch eine Stoppwortliste des HeBis nachzunutzen, um so die Performance von Solr durch einen kleineren Index zu verbessern und den Recall zu erhöhen. Da aber bei der Suche nicht das Verfahren des „exact match“ sondern das Verfahren „minimum should match“ unter Verwendung von DisMax verwirklicht werden sollte (siehe Kapitel „Fehlertolerante Gestaltung der Suche“), führte die Verwendung von Stoppwortlisten häufig fälschlicherweise zu keinen Treffern.<sup>185</sup> Um dieses Problem zu lösen, müssten alle Felder des Solar-Index bei der Indexierung die Stoppwortliste berücksichtigen.<sup>186</sup> Auf Grund der häufigen Null-Treffer-Ergebnisse wurde vorerst von der Nutzung von Stoppwortlisten Abstand genommen. Die Nachfrage an die TUB HH ergab, dass diese zurzeit ebenso auf die Verwendung einer Stoppwortliste verzichten,<sup>187</sup> ebenso wie auch der auf Lucene basierende Katalog HEIDI der Bibliotheken der Universität Heidelberg.<sup>188</sup>

---

<sup>184</sup> Vgl. STOCK 2007 – Information retrieval.

<sup>185</sup> Vgl. ROCHKIND 2010 – Stop Words.

<sup>186</sup> Vgl. VUFIN DOCUMENTATION - subject fields.

<sup>187</sup> Vgl. MARAHRENS 2010 – Tweet

<sup>188</sup> Verschiedene Suchanfragen mit typischen Stoppworten führten in HEIDI zu Treffern.

### 9.5.6.2. Synonyme

Die Kontrolle von Synonymen, also der gleichen Bedeutung von unterschiedlichen Begriffen, ist in VuFind sehr einfach zu handhaben. In der Textdatei `vufind/solr/biblio/conf/synonyms.txt` können Begriffe in einer Art Wörterbuch zusammengeführt werden. Synonyme werden in dieser Datei nur durch Kommata getrennt. So könnte man beispielsweise Teile der in der Schlagwortnormdatei in (SWD) enthaltenen Synonyme in einen auf VuFind basierenden Katalog übernehmen und das Retrieval durch die Verbesserung des Recalls weiter optimieren. Zusätzlich auch die verwandten Begriffe und die Unterbegriffe aus der SWD zu übernehmen, würde aber zu hohem Informationsballast führen.<sup>189</sup> Zu Testzwecken wurde der Begriff „Inkunabel“ (zwei Treffer im bachelopac) in der Synonymliste noch mit den in der Onlineschlagwortnormdatei (OSWD)<sup>190</sup> angezeigten Synonymen „Wiegendruck“, „Inkunabelkunde“ und „Inkunabeln“ verknüpft. Nach der Verknüpfung wurden elf der vierzehn relevanten Titel im bachelopac gefunden. Dies verbesserte den Recall von 14% auf 78%.




Abbildung 20: Suchanfrage nach „Inkunabel“ ohne Nutzung von Synonymen

<sup>189</sup> Vgl. WIESENMÜLLER 2010 – Daten härter, S.51.

<sup>190</sup> Vgl. BZB - OSWD 2010.

Treffer 1 - 11 von 11 für Suche: 'Inkunabel', Suchdauer: 0.05s

---

 **Inkunabelkunde** : eine Einführung in die Welt des frühesten Buchdrucks / von Geldner, Ferdinand, erschienen 1978  
**Signatur:** Kom+303+008  
**Standort:** Fachbibliothek TWI 1 (Berliner Tor 5)  
● Verfügbar  
■ Buch

---

**Inkunabel- und Einbandkunde** : Beiträge des Symposions zu Ehren von Max Joseph Husung am 17. und 18. Mai 1995 in Helmstedt / erschienen 1996  
**Signatur:** BID+006+004+Bd.+29  
**Standort:** Fachbibliothek TWI 1 (Berliner Tor 5)  
● Verfügbar  
■ Buch

---

**Handbuch der Inkunabelkunde** / von Haebler, Konrad, erschienen 1966  
**Signatur:** Kom+303+007/2.A.  
**Standort:** Fachbibliothek TWI 1 (Berliner Tor 5)  
● Verfügbar  
■ Buch

---

**Der Wiegendruck im Kartenbild** / von Teichl, Robert, erschienen 1964  
**Signatur:** Unknown  
**Standort:** Unknown  
● Ausgeliehen  
■ Unbekannt

---

**Katalog der Inkunabeln des Stadtarchivs Heilbronn** : mit einer Liste der Handschriften und einem Abriß der Heilbronner Buch- und Bibliotheksgeschichte / von Hummel, Heribert, erschienen 1981  
**Signatur:** BID+415+002  
**Standort:** Fachbibliothek TWI 1 (Berliner Tor 5)  
● Verfügbar  
■ Buch

---

**Zur Arbeit mit dem Gesamtkatalog der Wiegendrucke** : Vorträge der internationalen Fachtagung vom 26. bis 30. November 1979 in Berlin; [Gesamtkatalog der Wiegendrucke] / erschienen 1989  
**Signatur:** Z+bitb+955/14+IX  
**Standort:** Fachbibliothek TWI 1 (Berliner Tor 5)  
● Verfügbar  
■ Buch

Abbildung 21: Suchanfrage nach „Inkunabel“ mit Nutzung von Synonymen

### **Erforderliche Kenntnisse zur Nutzung von Stoppworten und Synonymen**

- Theoretischer Hintergrund zur Funktionsweise von Stoppwort –und Synonymlisten
- Grundkenntnisse über das Zusammenspiel von Solr/Lucene
- Kenntnisse über den Solr-Index
- Grundverständnis des Information Retrieval und der Wissensorganisation

### **Einstufung nach den Kriterien der Machbarkeit: 2a**

## 9.5.7. Fehlertolerante Gestaltung der Suche

Seit über 20 Jahren ist bekannt, dass Nutzer von Bibliothekskatalogen erwarten, ihre Suchanfragen in natürlicher Sprache formulieren zu können ("... whatever terms popped into their minds").<sup>191</sup> Bereits im Kapitel „Notwendigkeit zeitgemäßer Kataloge“ wird die Problematik beschrieben, dass Boolesche Operatoren von Nutzern kaum verwendet oder falsch verstanden werden und auch die Möglichkeit zur Trunkierungen kaum genutzt wird.

Ausgehend von diesen Erkenntnissen ist es das Oberziel der Verfasser, den kognitiven Aufwand der Nutzer zu senken. Durch die Erhöhung des Recalls sollen Null-Treffer-Ergebnisse vermieden werden. Die von Schulz 1994 bzw. 1998 genannten Lösungen sind auch heute nicht überholt:

- Automatische Rechtschreibkorrektur und automatische Verweisungen
- Reduktion von Wortformen (Stemming)
- Retrieval von Datensätzen, die nicht jeden einzelnen der eingegebenen Suchtermini enthalten (closest-match-Verfahren)<sup>192</sup>

In diesem Kapitel wird untersucht, inwieweit diese umsetzbar sind.

### 9.5.7.1. Rechtschreibkorrektur und Verweisungen

Die automatische Rechtschreibkorrektur sowie automatische Verweisungen auf alternative Suchbegriffe funktionierten out of the box bereits weitgehend zufriedenstellend, sodass auf eine Optimierung der Konfiguration im Rahmen dieser Arbeit verzichtet werden konnte.



Abbildung 22: Rechtschreibkorrektur und automatische Verweisungen im bachelopac

Bedingung für das zuverlässige Arbeiten dieser Funktionen ist lediglich, dass nach einer Aktualisierung des Datenbestandes, z.B. durch Import neuer Datensätze, der Solr-Index optimiert wird.

<sup>191</sup> Hildreth 1989, S. 17, zitiert nach SCHULZ 1998 – Wie der Schnabel, S. 345.

<sup>192</sup> SCHULZ 1998 – Wie der Schnabel, S. 347.



### 9.5.7.2. Reduktion von Wortformen (Stemming)

In der Standard-Konfiguration von VuFind ist der für die englische Sprache optimierte Porter-Stemming-Algorithmus in der Datei `vufind/solr/biblio/schema.xml` vorkonfiguriert:

```
<filter class="solr.SnowballPorterFilterFactory" language="English"
protected="protwords.txt"/>
```

Ziel der Verfasser war es einen für die deutsche Sprache angepassten Stemming-Algorithmus zu verwenden. Solr unterstützt insgesamt fünf verschiedene Stemming-Algorithmen<sup>193</sup> für die deutsche Sprache, von denen zwei auf dem Snowball-Projekt<sup>194</sup> von Martin Porter beruhen. Die Verfasser verfügten jedoch über keine Entscheidungsgrundlage und entschieden sich daher für eine modifizierte Version des für die deutschen Sprache optimierten Snowball-Porter-Stemming-Algorithmus, der auch in beluga zum Einsatz kommt.<sup>195</sup> Aus

```
<filter class="solr.SnowballPorterFilterFactory" language="English"
protected="protwords.txt"/>
```

wurde

```
<filter class="solr.SnowballPorterFilterFactory" language="German2"
protected="protwords.txt"/>.
```

Zu beachten war, dass das Porter-Stemming sowohl bei der Indexierung<sup>196</sup> als auch bei der Suche angewendet werden muss.<sup>197</sup> Daher mussten die Anpassungen in den Abschnitten `<analyzer type="index">` und `<analyzer type="query">` vorgenommen werden.

Nach einer Neuindexierung aller Datensätze stand der für die deutsche Sprache angepasste Stemming-Algorithmus zur Verfügung, der auch die Reduktion von Wortformen mit Umlauten ermöglicht.

<sup>193</sup> SOLR WIKI 2010 – LanguageAnalysis.

<sup>194</sup> <http://snowball.tartarus.org/>.

<sup>195</sup> Dies wurde durch Analyse der Datei `schema.xml` von beluga ermittelt.

<sup>196</sup> Die exakte Suche nach einer bestimmten Flexion ist daher in Feldern, auf die der Stemming- Algorithmus angewendet wurde, nicht mehr möglich.

<sup>197</sup> SOLR WIKI 2010 – AnalyzersTokenizersTokenFilters.

## Anwendungsbeispiel

Durch die Reduktion des Suchterminus „Häuser“ auf die Grundform „Haus“ werden auch Datensätze gefunden, die die Flexionen wie „Hause“ und „häuslich“ enthalten.

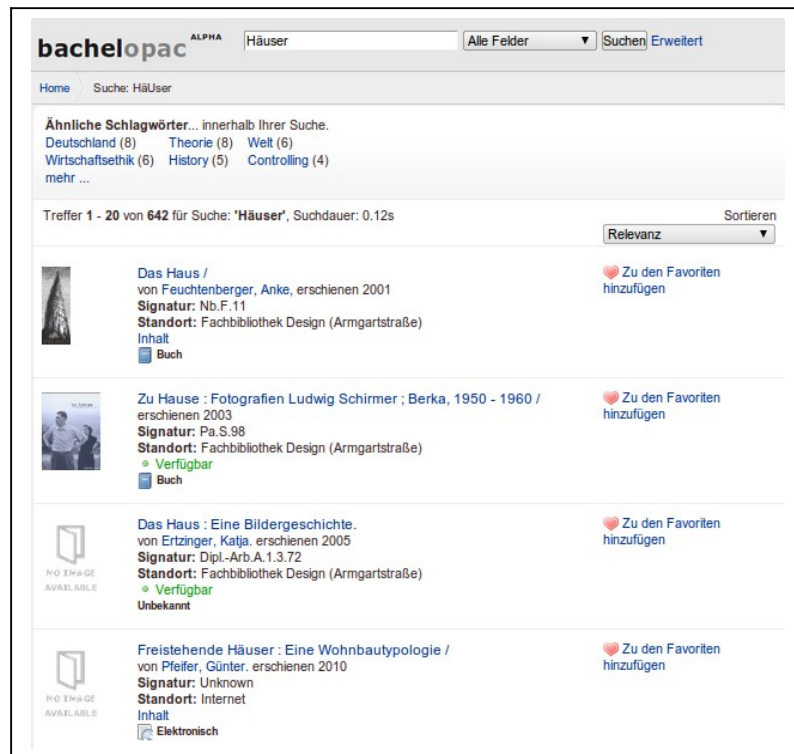


Abbildung 23: Stemming des Suchterminus "Häuser"

## „Protected Words“ - Wörter, die nicht gestemmt werden sollen

Es war den Verfassern nicht möglich, durch Erweiterung der Datei protwords.txt bestimmte Suchtermini vor dem Stemming zu schützen. Katz schrieb auf der Mailing-Liste VuFind General zu diesem Problem: „Unfortunately, I don't think the "protwords.txt" solution will actually work. As far as I know, protwords is an option for the EnglishPorterFilterFactory, not the SnowballPorterFilterFactory. You could always change the stemmer class. I'm not sure why we have a protwords setting in our default configuration, but I'm guessing it may be an artifact from a previous change of class.“<sup>198</sup>

<sup>198</sup> KATZ 2010 – Re: Autostemming.

Es ist zu beachten, dass der Katalog der HAW Hamburg signifikante Anteile an fremdsprachiger Literatur aufweist. Inwieweit es insbesondere in diesem Bereich zu Over- bzw. Understemming<sup>199</sup> kommt, wird im Rahmen dieser Arbeit nicht untersucht.

### 9.5.7.3. Minimum Should Match

DisMax stellt mit dem Parameter mm (Minimum Should Match) eine Option zur Verfügung, mit der eingestellt werden kann, in welchem Mindestumfang die eingegebenen Suchtermini mit einem indexierten Datensatz übereinstimmen (matchen) müssen, um zu einem Treffer zu führen. Diese Angaben können in einfachen relativen Werten (z.B. mindestens 80% der eingegebenen Suchtermini müssen matchen) erfolgen; es sind jedoch auch komplexe Abstufungen, abhängig von der Anzahl der eingegebenen Suchtermini möglich.<sup>200</sup> Der Parameter mm wird, wie auch der im Kapitel „Lehrbuchboosting“ vorgestellte Parameter bq (Boost Query) über die Datei vufind/solr/biblio/confsolrconfig.xml gesteuert.

Minimum Should Match ist jedoch nach der Grundinstallation von VuFind nicht vorkonfiguriert; alle Suchbegriffe müssen mit einem indexierten Datensatz übereinstimmen.

Ziel der Verfasser war es, die Funktion „Minimum Should Match“ mit einer für Bibliothekskataloge sinnvollen Konfiguration in den bachelopac zu integrieren. Aufgrund fehlender eigener Entscheidungsgrundlagen wurde auch hier die Konfiguration von beluga übernommen und in den bachelopac integriert. Allerdings konnte diese zum damaligen Zeitpunkt von den Verfassern nicht eindeutig interpretiert werden, weshalb sie Unterstützung auf Twitter suchten und fanden.

---

<sup>199</sup> D.h. zu starke bzw. zu schwache Wortformenreduktion.

<sup>200</sup> Vgl. SOLR 2010 – DisMaxQParserPlugin.



Abbildung 24: Tweet von @bachelopac: "Was bedeutet denn diese Konfig..."  
SCHMITT, STEHLE 2010 – Tweet



Abbildung 25: Tweet von @tillk: "@bachelopac Gibt ein Nutzer 3 bis 5 Suchterme ein..."  
KINSTLER 2010 – Tweet

### Anwendungsbeispiel

Eine Suche im bachelopac nach „pflagemanagement altenheim krankenhaus“ ohne „Minimum Shoud Match“ ergab keine Treffer, da kein Datensatz alle drei Suchbegriffe enthält. Da durch die Verwendung des mm-Parameters jetzt nur noch zwei der drei Suchbegriffe matchen müssen, erhält der Benutzer neun Treffer, die sein Informationsbedürfnis befriedigen könnten. Er wird nicht dazu gezwungen, mehrere Suchanfragen durchzuführen oder Boolesche Operatoren zu verwenden.

Home Suche: Pflegemanagement Altenheim Krankenhaus

Ähnliche Schlagwörter... innerhalb Ihrer Suche.  
 Altenheim (2) Krankenhaus (1) Management (1)  
 Organisationssoziologie (1) Organisationsstruktur (1) Organisatorischer Wandel (1)  
 mehr ...

Treffer 1 - 9 von 9 für Suche: 'pflegemanagement altenheim krankenhaus', Suchdauer: 0.05s  
 Sortieren Relevanz

Andere Suchmöglichkeiten:  
 pflegemanagement altenheim » pflegemanagement altenheimen » pflegemanagement aktuell

	<b>Pflegemanagement in Altenheimen : Grundlagen für Konzeptentwicklung und Organisation /</b> erschienen 1995 Signatur: Ges+C+4+-+2.1 Standort: Mehrere Standorte Verfügbar Buch	Zu den Favoriten hinzufügen
	<b>Grundlagen des Pflegemanagements im Krankenhaus /</b> von Keck, Alfred, erschienen 1995 Signatur: Ges+C+4+-+3.1 Standort: Fachbibliothek Life Sciences (Bergedorf) Verfügbar Buch	Zu den Favoriten hinzufügen
	<b>Pflegemanagement in Altenpflegeeinrichtungen /</b> erschienen 2008 Signatur: Pf.45.Pf.3 Standort: Fachbibliothek Soziale Arbeit und Pflege (Alexanderstr. 1) Inhalt Buch	Zu den Favoriten hinzufügen
	<b>Pflegemanagement in Alteinrichtungen : Grundlagen für Konzeptentwicklung und Organisation /</b> erschienen 2000 Signatur: Pf.45.Pf.1 Standort: Fachbibliothek Soziale Arbeit und Pflege (Alexanderstr. 1) Inhalt Buch	Zu den Favoriten hinzufügen
	<b>Betriebswirtschaft und Management im Krankenhaus /</b>	Zu den Favoriten hinzufügen

**Suche einschränken**  
 nur Abschlußarbeiten

**Verfasser**  
 Gronemeyer, Reimer, (1)  
 Heimerl, Peter, (1)  
 Keck, Alfred (1)

**Erscheinungsjahr**  
 1995 (2)  
 2005 (2)  
 2007 (2)  
 2000 (1)  
 2008 (1)  
 mehr ...

**Klassifikation**  
 Stationäres Gesundheitswesen (3)  
 Deutsche Geschichte des 19. und 20. Jahrhunderts (1)

**Sprache**  
 Deutsch (9)

**Medientyp**  
 Buch (9)

Abbildung 26: Treffer-Liste mit Datensätzen, die nicht alle Suchtermini enthalten

## Erforderliche Kenntnisse und Fähigkeiten

- Kenntnisse über das Nutzerverhalten von Bibliothekskatalogen
- Grundkenntnisse über Funktionen der Parameter von DisMax
- Wissen, in welchen Dateien man die Parameter einstellt

## Einstufung nach den Kriterien der Machbarkeit: 2b

## 9.5.8. Personalisierung

### 9.5.8.1. Technik

VuFind bietet mehrere Möglichkeiten, die Authentifizierung der Nutzer zu ermöglichen. Vorkonfiguriert sind die Anbindung an LDAP<sup>201</sup> bzw. das ILS, die Nutzung der internen MySQL-Datenbank von VuFind oder die Authentifizierung mittels Shibboleth. Die einzig für die Verfasser mögliche Option war die Authentifizierung der Nutzer über die interne MySQL- Datenbank von VuFind, da die Verfasser keinen direkten Zugriff auf das ILS der Bibliothek der HAW Hamburg haben.

Es kann pro E-Mail-Adresse nur ein Konto angelegt werden. Aufgrund der fehlenden Anbindung an die PICA-Nutzerkonten der HAW Hamburg konnten aber die Ansicht der ausgeliehenen Medien, der Bestellungen und der Vormerkungen sowie der Gebühren nicht realisiert werden. Dass die Anbindung an das ILS von PICA möglich ist, zeigt der von Oliver Marahrens entwickelte Treiber für PICA, der bereits im TUBFind-Katalog eingesetzt wird. Auch diese Authentifizierung geschieht über LDAP<sup>202</sup>. Da PICA die Authentifizierung über LDAP nicht unterstützt, wird hier ein eigener LDAP-Server benötigt. Weitere Informationen über den Einsatz und die Konfiguration des PICA-Treibers finden sich in der VuFind-Dokumentation.<sup>203</sup>

Um die Authentifizierung der Nutzer über die interne MySQL- Datenbank von VuFind zu ermöglichen, muss lediglich in der Datei vufind/web/conf/config.ini der Eintrag `method = DB` aktiviert werden.

---

<sup>201</sup> Acronym für Lightweight Directory Access Protocol.

<sup>202</sup> Vgl. MARAARENS 2010 – Experteninterview.

<sup>203</sup> VUFIN D DOCUMENTATION 2010 – PICA/DAIA.

```

; This section allows you to determine how the users will authenticate
; You can use an LDAP directory, the local ILS or the VuFind database.

[Authentication]
;method          = LDAP
;method          = ILS
method          = DB
;method          = Shibboleth

```

Listing 17: Konfiguration der Nutzerauthentifizierung (config.ini)

Die Nutzung der internen MySQL- Datenbank von VuFind erlaubt es dem Nutzer, einzelne Titel zu kommentieren, diese zu taggen, Listen zu erstellen und Suchen zu speichern. Die Nutzerkonten können zum Beispiel mit Hilfe des Programms phpMyAdmin<sup>204</sup> eingesehen und administriert werden. Die Zugangsdaten für die MySQL-Datenbank via phpMyAdmin entsprechen denen, die man bei der Installation von VuFind angegeben hat.

### 9.5.8.2. Funktionen

VuFind bietet mehrere dieser Funktionen an, die zum Teil erst nach einer Anmeldung der Nutzer verfügbar und anwendbar sind. Hierzu zählen die Möglichkeit Titel zu kommentieren, diese zu taggen, Listen zu erstellen, Suchen zu speichern und Suchanfragen per RSS zu abonnieren. Um ein persönliches Konto einzurichten, müssen die Eingaben Vor- und Nachname, Emailadresse, Benutzername und Passwort angegeben werden.

#### Kommentarfunktion

Mit Möglichkeit, Titel im bachelopac zu rezensieren, wird ein expliziter Recommenderdienst geschaffen. Da die Verwendung der Kommentarfunktion nur durch registrierte Nutzer möglich ist, und somit auch jeder Eintrag einem einzelnen Nutzer zuzuordnen ist, sollte sich das Problem des Missbrauchs der Funktion im

<sup>204</sup> phpMyAdmin ist eine Open-Source-Software zur Administration von MySQL-Datenbanken unter der GPL-Lizenz.

Sinne von Spam, „übler Nachrede und verfälschenden Rezensionen“<sup>205</sup> in Grenzen halten. Sollte die Authentifizierung der Nutzer nur über das LBS möglich sein, würde dieses Problem noch weiter eingeschränkt. Trotzdem erfordert der erfolgreiche Einsatz dieser Funktion eine Überwachung durch die Bibliothek und die Schaffung von Anreizen, die Nutzer dazu veranlasst Rezensionen zu schreiben.<sup>206</sup> Eigene erstellte Rezensionen können von Nutzern auch wieder durch diese gelöscht werden. Die Kommentarfunktion konnte ohne gesonderte Konfiguration im bachelopac verwirklicht werden.

### **Tagging**

Tagging bedeutet, dass der Nutzer den Titel selbst verschlagworten kann, also eine eigene Taxonomie verwenden darf, die es ihm ermöglicht für ihn relevante Medien schneller und einfacher wiederzufinden.

Da bei VuFind alle direkt einem Medium zugeteilten Tags per se öffentlich sind (im Gegensatz zu Tags, die innerhalb eigener Listen vergeben werden), können diese Tags auch anderen Nutzern bei ihrer Recherche eine zusätzliche Hilfe sein.

Ein weiteres Anwendungsszenario ist auch die gemeinsame Nutzung von Tags innerhalb von Arbeitsgruppen, die ein gemeinsames Tag verwenden und so ihre gemeinsamen Rechercheergebnisse teilen können.<sup>207</sup> Ein Tag hat eine maximale Länge von 25 Zeichen. Soll ein Tag aus mehr als einem Wort bestehen, müssen Anführungszeichen bei der Eingabe verwendet werden. Zu beachten ist, dass die Suche nach Tags nicht Teil der feldübergreifenden Suche ist, da Tags nicht Teil des Solr-Indexes sind, sondern in der MySQL-Datenbank von VuFind gespeichert werden; es besteht nur die Möglichkeiten, die Suche auf Tags zu beschränken.

Die Vor- und Nachteile von Tagging in Bibliothekskatalogen, wie die fehlende Synonym-Kontrolle, fehlende Unterscheidung von Homonymen und Polysemen, beschreibt Kneifel sehr detailliert in „Mit Web 2.0 zum Online-Katalog der nächsten Generation“<sup>208</sup>.

---

<sup>205</sup> DIEROLF 2007 – Der Uni-Katalog Karlsruhe, S. 131.

<sup>206</sup> DIEROLF, MÖNNICH 2006 – Recommendersysteme , S. 28.

<sup>207</sup> DRAUSZ, P LIENINGER 2010 – Nutzerwünsche.

<sup>208</sup> KNEIFEL 2009 – Mit Web 2.0



## **Listen**

VuFind bietet dem Nutzer die Möglichkeit, beliebig viele Listen zu erstellen. Einzelne Titel können mit der Funktion „Zu den Favoriten hinzufügen“ in Listen eingefügt werden. Diese Listen können sowohl rein privat genutzt als auch öffentlich gemacht werden. Jede Liste erhält ihre eigene, permanente URL. Zusätzlich besteht die Möglichkeit, einzelne Titel mit einem Kommentar bzw. einer Bemerkung zu versehen. Auch können Listen für den eigenen Gebrauch getagt werden, diese Tags sind aber nicht für andere Nutzer einsehbar. Ein denkbare Anwendungsszenario ist ein von Dozenten erstelltes und kommentiertes Literaturverzeichnis, welches über einen permanenten Link abrufbar ist. Andererseits besteht die Möglichkeit, die eigene Literaturrecherche als privat zu kennzeichnen und Literaturlisten nur für den eigenen Gebrauch zu erstellen.

**bachelopac** ALPHA  Alle Felder Suchen Erweitert

[Home](#) [Ihr Konto](#) [Liste](#)

**Seminar Web 2.0**

Seminar Web 2.0

Treffer **1 - 6** von **6** Sortieren Relevanz

	<b>Web 2.0 and libraries : impacts, technologies and trends /</b> Buch	Bearbeiten Löschen
	<b>Web 2.0 : Konzepte, Anwendungen, Technologien /</b> von: Alby, Tom. Buch	Bearbeiten Löschen
	<b>How to use web 2.0 in your library /</b> von: Bradley, Phil. Buch	Bearbeiten Löschen
	<b>Searching 2.0.</b> von: Sauers, Michael P.. Buch	Bearbeiten Löschen
	<b>Mit Web 2.0 zum Online-Katalog der nächsten Generation : Innovationspreis 2009 /</b> von: Kneifel, Fabienne. Buch	Bearbeiten Löschen
	<b>Bibliotheken im Web-2.0-Zeitalter : Herausforderungen, Perspektiven und Visionen /</b> von: Kaiser, Ronald. Buch	Bearbeiten Löschen

Abbildung 27: Öffentliche Liste zum Thema Web 2.0 und Bibliotheken

## RSS

Der Nutzer kann Suchanfragen per RSS-Feed abonnieren, ohne erneut VuFind aufrufen und manuell nach den gewünschten Informationen suchen zu müssen. Diese Funktion bedarf keiner weiteren Konfiguration und war direkt einsetzbar.

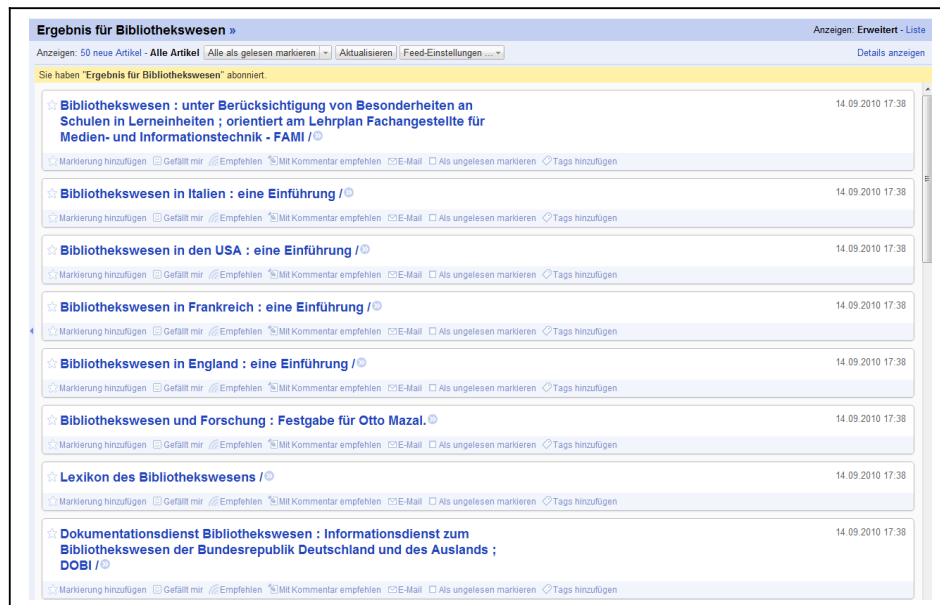


Abbildung 28: RSS-Feed der Katalog-Suche nach "Bibliothekswesen" im Google Reader

## 9.5.9. Kataloganreicherungen und Mashups

### 9.5.9.1. Coveranzeige

Die Anzeige von Covern in der VuFind-Trefferliste bzw. der VuFind-Titelanzeige ist sehr einfach zu konfigurieren/einzurichten. Die Zuordnung der Cover erfolgt über die ISBN. Vorkonfiguriert sind die Anzeige der Cover von Amazon<sup>209</sup>, Syndetic Solutions<sup>210</sup>, OpenLibrary<sup>211</sup>, LibraryThing<sup>212</sup>, Google Book Search<sup>213</sup>. Die Nutzung der Cover von GBS und Openlibrary kann ohne Registrierung erfolgen. Für die Nutzung von LibraryThing und Amazon ist eine kostenlose Registrierung erforderlich. Syndetic Solutions hingegen ist ein kommerzieller Anbieter von Kataloganreicherungsdaten und konnte daher nicht getestet werden.

In der Datei config.ini befindet sich folgende Zeile,

```
;Coverimages= Syndetics:MySyndeticsId, Amazon:MyAccessKeyId,  
LibraryThing:MyLibraryThingId, Google,OpenLibrary
```

die durch die Entfernung des Semikolons aktiviert werden muss bzw. in der die für registrierungspflichtige oder kommerzielle Anbieter zugehörigen IDs bzw. Passwörter eingetragen werden müssen.

Die Anzeige von Covern ist mehr als eine optische Verschönerung des Katalogs, sondern spielt eine wichtige Rolle bei der richtigen Entscheidungsfindung und Medienauswahl des Nutzers. Dies hat auch Christensen sehr praxisnah am Beispiel einer Nutzerin, die sich ein Kinderbuch über Vulkane als Literatur für ein Studium der Geowissenschaften ausleihen wollte, auf dem Bibcamp 2010 beschrieben. Eine zusätzliche Abbildung des Covers hätte eine Missinterpretation bzw. der Metadaten verhindert bzw. das Übersehen des Forms Schlagworts „Kindersachbuch“ ausgeglichen.<sup>214</sup>

---

<sup>209</sup> <http://aws.amazon.com/de/>

<sup>210</sup> <http://www.bowker.com/syndetics/>

<sup>211</sup> <http://openlibrary.org/>

<sup>212</sup> [http://www.librarything.com/wiki/index.php/Free\\_covers](http://www.librarything.com/wiki/index.php/Free_covers)

<sup>213</sup> <http://books.google.de>

<sup>214</sup> CHRISTENSEN 2010 – Der beratende Katalog.

### **9.5.9.2. Wikipedia**

VuFind bietet die Möglichkeit Inhalte der Wikipedia direkt im Katalog anzuzeigen zu lassen. Dies geschieht über die REST-API, die Informationen über den Autor in der Wikipedia abrufen. REST ist ein Akronym für Representational State Transfer. Die Wikipedia-Informationen werden durch einen Klick auf den Namen des Autors in der Titelanzeige aufgerufen.

Diese zusätzlichen Informationen über den Autor dienen nicht nur der eigenen Entscheidungsfindung und zur Bewertung der Relevanz des Autos, so geben sie auch viele Anregungen zu einer möglichen weiteren Recherche bzw. zu Begriffen die für eine mögliche weitere Recherche im Katalog sinnvoll sein könnten. Die im ursprünglichen Wikipedia-Artikel verlinkten Begriffe sind auch in VuFind anklickbar und führen eine Suche nach diesem Begriff im Katalog aus.

Die Anzeige der Informationen über Autoren aus Wikipedia funktioniert in VuFind out of the box. Die Auswahl der deutschen Wikipedia wurde bereits in der Konfiguration der config.ini durch die Auswahl der Spracheinstellung de\_DE getroffen und es bedarf keiner weiteren Konfiguration. Das Problem der fehlenden Anzeige von Bildern aus der Wikipedia im bachelopac konnte nicht gelöst werden. (siehe Kapitel „Offene Probleme“).

### **9.5.9.3. Rezensionen**

VuFind bietet die Möglichkeit, Rezensionen oder Kundenbewertungen in die Titelanzeige einzubinden. In der config.ini von VuFind sind die Angebote von Syndetics und Amazon bereits vorkonfiguriert. Das kostenpflichtige Angebot von Syndetics wurde von den Verfassern nicht getestet. Bei der Verwendung des Amazon-Angebots sind die genauen Geschäftsbedingungen von Amazon zu prüfen.

Die Einbindung von Rezensionen von Amazon erfordert keine zusätzliche Anmeldung, sondern kann mit der gleichen Registrierung wie bereits bei der Coveranzeige erfolgen. In folgenden zwei Einträgen in der config.ini muss der MyAccessKeyId und der MyAmazonSecretKey eingetragen werden:

```

;reviews      =
Syndetics:MySyndeticsId,AmazonEditorial:MyAccessKeyId,Amazon:MyAccessKeyId
;amazonsecret = MyAmazonSecretKey

```

Listing 18: Einbindung von Rezensionen von Amazon (config.ini)

Zusätzlich muss allerdings noch in der Datei amazon.php unter vufind/web/sys der webservice von Amazon auf das deutschsprachige Angebot von Amazon umgestellt werden, um deutsch- statt englischsprachige Rezensionen anzuzeigen.

```

class AWS_Request
{
    private $endpoint = 'webservices.amazon.com';
    private $method = HTTP_REQUEST_METHOD_GET;
    private $requestURI = '/onca/xml';
    private $url;
}

```

Listing 19: Anzeige von deutschsprachigen Rezensionen von Amazon (amazon.php)

Aus 'webservices.amazon.com' wird 'webservices.amazon.de'

Anzumerken ist, dass Fokusgruppeninterviews, die im Rahmen des beluga-Projekts durchgeführt wurden, das Ergebnis brachten, das Studierende gegen eine Anreicherung von Katalogen durch Rezensionen von Amazon sind, da Studierenden von einem Katalog Neutralität und Wissenschaftlichkeit erwarten.<sup>215</sup> An dieser Stelle wurde lediglich die Machbarkeit der Einbindung von Amazon-Rezensionen überprüft.

### **Erforderliche Kenntnisse und Fähigkeiten**

Nach der beschriebenen Aktivierung der Benutzer-Authentifizierung über die Datei config.ini waren keine weiteren Kenntnisse erforderlich.

### **Einstufung nach den Kriterien der Machbarkeit: 2a**

<sup>215</sup> CHRISTENSEN 2010 – Fokusgruppen.

#### 9.5.9.4. Implementierung der Google Book Search API

Die Programmierschnittstelle (API) von Google Book Search bietet die Möglichkeit, auf Buchvorschauen oder Volltexte, die im Rahmen von Google Book Search digitalisiert wurden, zu verlinken oder diese direkt auf der eigenen Website einzubinden. Anhand der ISBN wird überprüft, ob eine Buchvorschau oder ein Volltext via Google Book Search verfügbar ist. Unter Verwendung des GBS-API-Preview Wizard lassen sich vier verschiedene Arten der Einbindung von GBS realisieren:

1. Eine Buchvorschau, die direkt in die Titelseite eingebaut werden kann
2. Ein Google-Vorschau-Button, der auf Knopfdruck eine in die Trefferseite eingebettete Ansicht öffnet
3. Ein Google-Vorschau Button, der direkt zum entsprechendem Titel auf der Google Books Website verlinkt
4. Einen Google Button, der die integrierte Buchvorschau auf einer anzugebende Website öffnet<sup>216</sup>

Die Einbindung von Google Book Search ist in der von den Verfassern verwendeten VuFind Version 1.0 noch nicht enthalten, soll aber in der Version 1.1 sowohl in der Trefferliste als auch in der Titelanzeige auch realisiert werden.<sup>217</sup>

Um die Google-Books-Vorschau in der verwendeten Version 1.0 zu ermöglichen, wurde die GBS-API von den Verfassern nachträglich im bachelopac implementiert. Auch hier zeigt sich die Bedeutung der Vernetzung und Offenheit des Projekts für dessen Entwicklung und Entstehung. In einem Blogeintrag haben die Verfasser eine Möglichkeit zur Einbindung von GBS in VuFind vorgestellt.<sup>218</sup> Dieser Blogeintrag führte zu einer Diskussion über die Art der graphischen Einbindung von GBS in den Katalog. Sollte man auf GBS über einen Button verlinken oder wäre es sinnvoller GBS direkt in einen Katalog einzubinden? Elmar Haake von der Staats- und Universitätsbibliothek Bremen fragte: „Wäre es auch

---

<sup>216</sup> GOOGLE 2010 – GBS Preview Wizard.

<sup>217</sup> VUFINd JIRA #303 2010.

<sup>218</sup> STEHLE 2010 – GBS API.

nicht eine Option, bei der Vollansicht der Metadaten den GBS-Inhalt gleich einzubauen?“ unter Hinweis auf einen Beispieldatensatz der E-LIB Bremen.<sup>219</sup>

Ausgehend von der durch die Usability-Tests von beluga gewonnene Erkenntnis, dass der GBS-Button sehr oft übersehen werden kann, wurde der GBS Button von den Verfassern zunächst an möglichst zentraler Stelle in die Titelanzeige eingebunden.<sup>220</sup> Die von Elmar Haake vorgeschlagene Möglichkeit, GBS direkt im Katalog direkt anzuzeigen, wurde zusätzlich umgesetzt, um die Machbarkeit zu überprüfen.<sup>221</sup>

Der Vorteil der Vorschau in der Titelanzeige ist es, dass es keinen weiteren Button mehr gibt, der übersehen oder falsch verstanden werden könnte. Allerdings führt die Einbindung von GBS in die Titelanzeige dazu, dass vermehrt gescrollt werden muss und die Titelanzeige unübersichtlicher wirkt. Die Frage, auf welche Art Google Book Search in einem Bibliothekskatalog dargestellt werden sollte, kann jedoch nur mit weiteren Usability-Tests beantwortet werden.

### **Technische Umsetzung**

Da in der verwendeten VuFind-Version 1.0 die Google Book Search API noch nicht implementiert ist, wurde diese von den Verfassern manuell in die Titelanzeige eingefügt.

Hierzu wurden die folgenden drei Schritte durchgeführt:

1. Unter Verwendung des GBS-API-Preview Wizard wird ein Java-Script durch die Angabe einer gültigen ISBN generiert. Die ISBN ist an dieser Stelle beliebig, da sie im weiteren Verlauf durch eine Variable ersetzt wird. Weiterhin wird konfiguriert, auf welche Art die GBS-API eingebunden werden soll.

---

<sup>219</sup> Ebd.

<sup>220</sup> SCHMITT, STEHLE 2009 – Stimmen aus dem Usability-Labor.

<sup>221</sup> Ebd.



**1 Enter book identifiers**  
Tell us all the ISBNs or other identifiers you know of for this book. If we can find a preview associated with any of these numbers, we can make it available to your site's users:

ISBN   
 ISBN   
 ISBN   
[Enter more identifiers...](#)

**2 Choose how to show the book preview**  
If a preview is available for this book, what should the user see on your page?

An embedded book viewer.  
Viewer size:  pixels wide,  pixels high

A button that opens a floating book viewer.

A button that links to the book's preview page on Book Search.

A button that goes to a custom preview page on your site.  
Preview URL:

If no preview can be found for the specified book(s), the preview or button will be automatically disabled and nothing will appear.

**3 Select your language**

**4 Generate code for your web page**

Copy the code below and paste it on your page where you would like the viewer to appear.

```
<script type="text/javascript" src="http://books.google.com/books/previewlib.js"></script>
<script type="text/javascript">
  GBS_setLanguage('de');
  GBS_insertEmbeddedViewer('ISBN:3827262062', 600, 500);
</script>
```

Abbildung 29: GBS API Preview Wizard

2. Die eindeutige ISBN, die der GBS-API-Preview Wizard verwendet, um das JavaScript zu erzeugen, muss nun durch die Variable \$isbn, die von VuFind bereits für die ISBN verwendet wird, ersetzt werden.

```
<script type="text/javascript"
src="http://books.google.com/books/previewlib.js"></script>

<script type="text/javascript">

GBS_setLanguage('de');

GBS_insertEmbeddedViewer('ISBN: 3827262062', 600, 500);

</script>
```

Listing 20: Ausgabe des GBS-API Preview Wizards für eine eingebettete Buchvorschau der Größe 600 \* 500 Pixel

```

script type="text/javascript"
src="http://books.google.com/books/previewlib.js"></script>

<script type="text/javascript">

GBS_setLanguage('de');

GBS_insertEmbeddedViewer('ISBN: {$isbn}', 600, 500);

</script>

Code: Ausgabe des GBS-API Preview Wizards für eine eingebettete Buchvorschau
der Größe 600 * 500 Pixel

```

Listing 21: Angepasstes JavaScript zur eingebetteten Darstellung von Google Books in der Titelanzeige

3. Dieser Code wird in die Datei `vufind/web/interface/themes/[Name des Verwenden Themes]/Record/view.tpl` des aktiven Templates integriert.

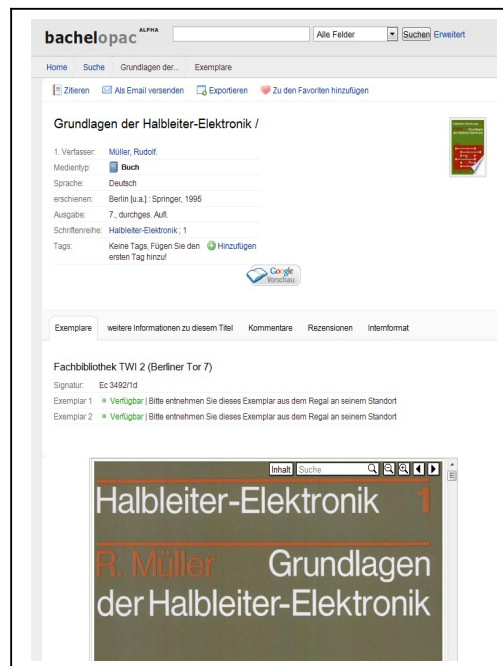


Abbildung 30: GBS-Button und eingebettete GBS-Vorschau im bachelopac

### Erforderliche Kenntnisse und Fähigkeiten

- Setzen einer Variable im vom von GBS-Wizard ausgegebenen JavaScripts.
- Wissen über die grundsätzliche Funktion der VuFind-Templates.

In ab VuFind 1.1. ist GBS bereits vorkonfiguriert, sodass die beschriebenen Schritte entfallen können.

### Einstufung nach den Kriterien der Machbarkeit: 2a

#### 9.5.9.5. Chatauskunft

Gemäß des Bibliothek-2.0-Prinzips „Kopieren und Integrieren von Programmen und Ideen Dritter in die Bibliotheksdienstleistungen“ soll beispielhaft eine im netbib weblog vorgestellte Idee - die Integration einer Chatauskunft auf Basis von MeeboMe in die Null-Treffer-Seite des Bibliothekskatalogs<sup>222</sup> – aufgegriffen und umgesetzt werden.

„This way your users have on-screen access to live help from library staff when and where they often need it.“<sup>223</sup> Das Potential von MeeboMe wird nach Sarah Houghton-Jan nicht ausgeschöpft, wenn Bibliotheken ihre Chat-Widgets „nur“ auf deren Website einbinden. „Library catalogs are our single biggest point of contact with our users. They get used much more than our library websites, which is where most libraries have placed their MeeboMe Widgets.“<sup>224</sup>

#### Meebo und MeeboMe

Meebo<sup>225</sup> ist ein kostenloser Instant-Messaging-Dienst, der mit verschiedenen Chat-Protokollen wie AIM/ICQ, MSN, Google Talk oder Jabber/XAMPP benutzt werden kann. Da Meebo browserbasiert ist, können dessen Benutzer Nachrichten über die Meebo-Website verschicken, ohne vorher ein Instant-Messenger-Programm installiert zu haben. Die Erweiterung MeeboMe ermöglicht es, ein Chat-Widget auf Websites einzubetten. Dazu wird auf der MeeboMe-Website ein Programm-Code für das Widget generiert, der in den Quellcode von beliebig vielen Internetseiten integriert werden kann.

MeeboMe ist in zahlreichen amerikanischen und australischen Bibliotheken im Einsatz.<sup>226</sup> In der deutschen Bibliothekswelt wird MeeboMe nach Kenntnisstand der Verfasser nur im Weblog der Stadtbibliothek Salzgitter<sup>227</sup> verwendet.

Ziel der Verfasser war es also, im bachelopac eine Chatauskunft auf Basis von MeeboMe anzubieten, sobald eine Suche keine Treffer hervorbringt. Zudem sollte die Chatauskunft über die Fußzeile von jeder Seite des Katalogs aus zugänglich sein.

---

<sup>222</sup> HELLER 2007- Drei Beobachtungen zu Chat-Widgets.

<sup>223</sup> HOUGHTON-JAN 2007 – Adding MeeboMe Widget.

<sup>224</sup> Vgl. ebd.

<sup>225</sup> <http://www.meebo.com>.

<sup>226</sup> Vgl. LIBRARY SUCCESS WIKI 2010 - IM

<sup>227</sup> <http://stadtbibliotheksalzgitter.wordpress.com/>.

Umsetzung

## 1. Einrichtung eines Meebo-Kontos

Zunächst wurde ein Meebo-Benutzerkonto auf der Website meebo.com angelegt und mit einem XMPP-Konto verknüpft. XMPP ist ein von der Jabber-Community entwickeltes Open-Source-Internetprotokoll, welches unter anderem für Instant Messaging-Dienste eingesetzt werden kann.<sup>228</sup> Als Benutzerkennung im XMPP-Netzwerk werden sogenannte Jabber-Identifizier (J-ID) benutzt, die wie eine E-Mail-Adresse aufgebaut sind. Das Jabber/XMPP-Protokoll wird auch von vielen populären E-Mail-Anbietern unterstützt, sodass ein bereits im Vorfeld eingerichtetes E-Mail-Konto als J-ID benutzt werden konnte.

## 2. Erstellung eines MeeboMe-Chat-Widgets

Im nächsten Schritt wurde auf der MeeboMe-Website das Chat-Widget konfiguriert und mit dem MeeboMe-Account verbunden. Größe, Form, Farben und Name können hierbei individuell gestaltet werden.

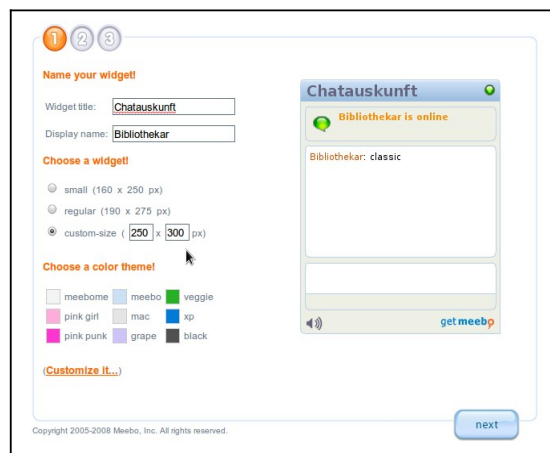


Abbildung 31: Konfiguration des MeeboMe-Chat-Widgets

<sup>228</sup> Vgl. UBUNTU USERS WIKI 2010 – Jabber.

Nach der Anpassung gibt die Meebo-Website den Code des konfigurierten Meebo-Widgets aus, der im nächsten Schritt weiterverwendet wurde.

```
<!-- Beginning of meebo me widget code.
Want to talk with visitors on your page?
Go to http://www.meebome.com/ and get your widget! -->
<embed src="http://widget.meebo.com/mm.swf?IGHeDAkoXW" type="application/x-
shockwave-flash" width="250" height="300"></embed>
```

Listing 22: Code von MeeboMe, der in Websites eingebunden wird

### 3. Einbindung des Widgets in die Datei chat.html

Theoretisch ließe sich mit dem ausgegebenen Code das Chat-Widget direkt in das „Null-Treffer“-Template oder an jede andere Stelle des VuFind-Katalogs einbinden. Dies würde jedoch Usability-Probleme mit sich bringen, da das Chatfenster verschwinden würde, sobald ein Nutzer eine neue Suche abschickt und damit eine neue Trefferliste generiert. Deshalb wurde der Code in die Datei chat.html ausgelagert, sodass das Chat-Widget in einem separaten Browser-Fenster geöffnet werden kann. So wird später die unabhängige Nutzung von Katalog und Chat-Auskunft gewährleistet.

### 4. Einbindung der Links zur chat.html im bachelopac

Da die Chatauskunft, analog zu dem standardmäßig in der Fußzeile<sup>229</sup> vorhandenen Link „Suchtipps“, immer in einem separatem Fenster mit vordefinierter Größe geöffnet werden soll, wurde der entsprechende Link kopiert und für die Chatauskunft wie folgt angepasst:

```
<a href="http://83.169.45.69/chat.html"
onClick="window.open('http://83.169.45.69/chat.html', 'Help', 'width=605,
height=499'); return false;">{translate text='Chatauskunft'}</a>
```

Listing 22: Code von MeeboMe, der in Websites eingebunden wird

Um die Chatauskunft auch von der Null-Treffer-Liste aus verfügbar zu machen, wurde derselbe Link in das Template list-none.tpl eingebunden.

<sup>229</sup> Zuständig ist das Template footer.tpl.

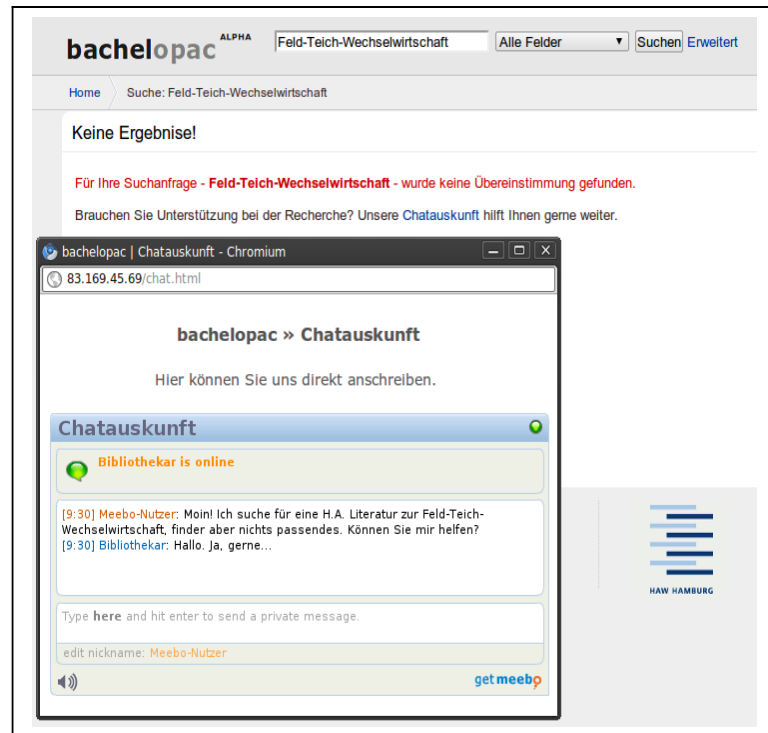


Abbildung 32: Chat-Auskunft auf Basis von MeeboMe in der Null-Treffer-Liste

## 5. Bedienung des Chats durch das Bibliothekspersonal

Sofern der Auskunftsbibliothekar auf der Meebo-Website eingeloggt ist, ist der Chat jetzt bereits funktionsfähig. Es ist für den Auskunftsbibliothekar jedoch komfortabler, den Chat über einen Instant-Messaging-Client wie Pidgin<sup>230</sup> zu bedienen; so können auch Chatlogs automatisch mitgeschnitten werden. Ausführliche Konfigurationsanleitungen zur Einrichtung des freien Chat-Clients Pidgin, in Kombination mit MeeboMe, finden sich in den Blogs Library Voice<sup>231</sup> und dem der Universitätsbibliothek der Universität von North Carolina.<sup>232</sup>

<sup>230</sup> <http://www.pidgin.im/>.

<sup>231</sup> BOENNINGER 2009 – Meebo

<sup>232</sup> UNC LIBRARIES 2008 – Pidgin.

## Erforderliche Kenntnisse und Fähigkeiten

Um eine Chatauskunft auf Basis von MeeboMe in den bachelopac zu integrieren, waren folgende Kenntnisse und Fähigkeiten notwendig:

- Grundlegende HTML-Kenntnisse zur Gestaltung der chat.html Seite mit implementiertem Chat-Widget
- Rudimentäre JavaScript-Kenntnisse, um die Eigenschaften zu definieren, wie die Datei chat.html geöffnet werden soll
- Ein Instant-Messaging-Programm für die Verwendung von MeeboMe konfigurieren

## Einstufung nach den Kriterien der Machbarkeit: 2a

### LibraryH3lp – eine Alternative zu MeeboMe

Der Webmessenger LibraryH3lp, funktioniert ähnlich wie MeeboMe und wird ebenfalls durch ein Widget eingebunden. Im Gegensatz zu MeeboMe ist LibraryH3lp speziell für bibliothekarische Bedürfnisse konzipiert und bietet einen größeren Funktionsumfang, wie etwa die Weiterleitung von Anfragen an andere Bibliothekare oder eine Warteschlangenfunktion. Da LibraryH3lp ohne Flash programmiert wurde, kann der Chat im Gegensatz zu MeeboMe leichter von Screenreadern ausgelesen werden.<sup>233</sup>

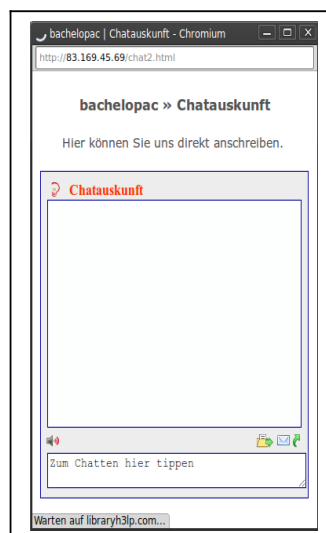


Abbildung 33: Chatauskunft auf Basis von LibraryH3lp

<sup>233</sup> Vgl. JUHST 2009- Kompetent IM Netz, S. 63f.

Die Verwendung von Library3lp ist für Bibliotheken jedoch kostenpflichtig, wobei die Preisstruktur von der Größe des Einzugsgebietes der Bibliothek abhängt. Solange das System nur getestet wird und sich noch nicht im produktiven Einsatz befindet, fallen jedoch keine Kosten an.<sup>234</sup>

### **Testinstallation**

Da Library3lp ebenfalls mit Widgets und Jabber/XMPP arbeitet, verliefen die Schritte zur Einrichtung ähnlich wie bereits bei MeeboMe beschrieben; die Testinstallation, die nur vorübergehend in Betrieb war, konnte ohne Schwierigkeiten bewerkstelligt werden.

Die für MeeboMe aufgeführten Kenntnisse gelten analog.

---

<sup>234</sup> Vgl. LIBRARYH3LP (Hg.) 2010 – Pay for LibraryH3lp.



### **9.5.10. Anpassung der Katalogoberfläche**

Das Aussehen der Benutzeroberfläche wird über sogenannte Themes gesteuert. VuFind bietet für die Browser-Ansicht am PC die Themes „default“ und „classic“ an. Für die Bedienung des Katalogs über ein Smartphone steht das Theme „mobile“ zur Verfügung.

Die Themes sind modular aufgebaut und bestehen aus sogenannten SMARTY-Templates (\*.tpl-Dateien) die vorwiegend PHP- und HTML-Code enthalten. Sämtliche Themes sind unter vufind/web/interface/themes/ abgelegt. Die Auswahl des Themes wird über die Datei vufind/web/conf/config.ini gesteuert.

Für den bachelopac wurde das neue Theme „bacheltheme“ auf Basis des Themes „default“ erstellt. Ziel war es, das ursprüngliche Design von VuFind dem Corporate Design der HAW Hamburg anzugleichen und dem in den Projektaufgaben definiertem Funktionsumfang anzupassen, um beurteilen zu können, inwieweit dies mit den vorhandenen Kenntnissen und Fähigkeiten erreichbar ist.

Hierzu wurde der Ordner web/interface/themes/default in den Ordner web/interface/themes/bacheltheme kopiert. Die Möglichkeit, nur die geänderte Templates in einem separaten Ordner zu verwalten („Theme Inheritance“<sup>235</sup>), und dadurch die Wartbarkeit zu vereinfachen, wurde erst im späteren Verlauf des Projekts wahrgenommen und daher nicht mehr umgesetzt.

#### **Farbgestaltung - bacheltheme/css/styles.css**

Die Oberfläche des bachelopacs konnte an die Farbgestaltung des Corporate Designs der HAW Hamburg angelehnt werden. Hierzu mussten lediglich die entsprechende Farbenwerte im Stylesheet angepasst werden.

#### **Seitenübergreifende Fußzeile – bacheltheme/footer.tpl**

Die Fußzeile wird seitenübergreifend definiert. Ergänzt wurden Links zur Chatauskunft sowie Links zu weiterführenden Informationen über das Projekt bachelopac. Nicht genutzte Funktionen wie der Semesterapparat wurden

---

<sup>235</sup> VUFIND DOCUMENTATION 2010 – customization.

auskommentiert. Zusätzlich wurde hier auch ein Logo der HAW Hamburg eingebunden.

#### **Startseite – bacheltheme/Search/home.tpl**

Das VuFind-Logo über dem Suchschlitz wurde durch ein eigenes bachelopac-Logo ersetzt. Zudem wurde ein kurzer Erklärtext „bachelopac ist ein experimentelles Projekt [...]“ ergänzt.

#### **Layout aller andern Seiten – bacheltheme/layout.tpl**

Um das VuFind-Logo links oben auf allen anderen Seiten gegen ein Logo des bachelopacs auszutauschen, wurde in der Datei layout.tpl eine entsprechende Bilddatei eingebunden.

#### **Titelseite – bacheltheme Record/view.tpl**

Die „Titel als SMS versenden“-Funktion wurde auskommentiert, da diese nur für US-amerikanische Anbieter vorkonfiguriert ist und eine Konfiguration auf eventuelle deutsche Anbieter nicht Teil der Projektaufgabe war. Darüber hinaus wurde im auch die Google-Book-Search-API implementiert.

### 9.5.10.1. Oberfläche für mobile Endgeräte

Beim Aufruf von VuFind über mobile Endgeräte wird anhand der Browserkennung automatisch eine für Smartphones optimierte, auf die wesentlichen Funktionen reduzierte Oberfläche gestartet.

Das Theme für mobile Endgeräte musste lediglich in der Datei vufind/web/config.ini aktiviert werden; es bedurfte keiner weiteren Anpassungen.

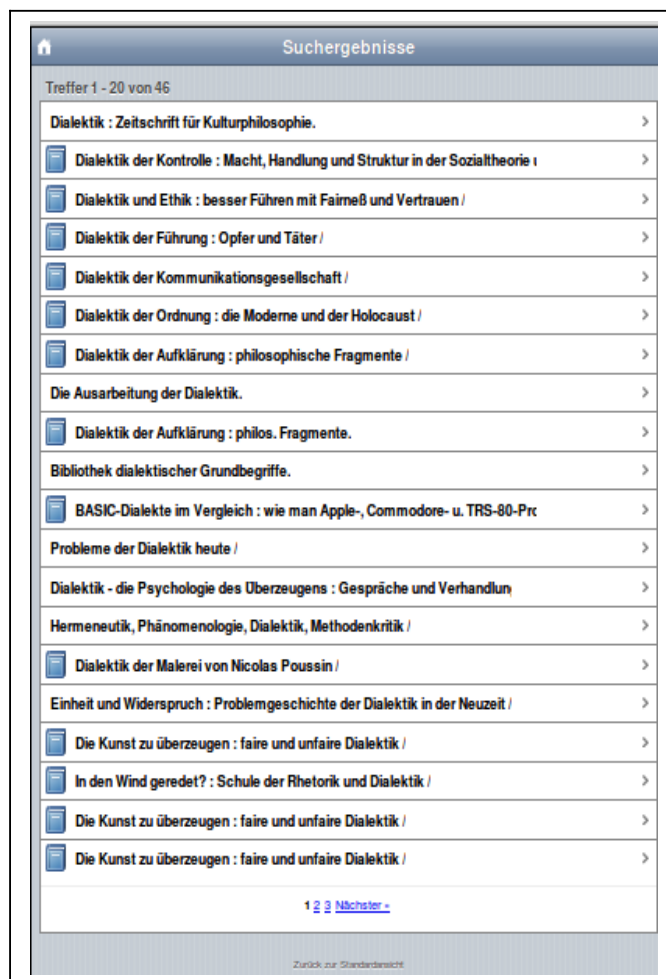


Abbildung 34: Für Smartphones optimierte Katalogoberfläche

### 9.5.10.2. Anpassung von Übersetzungen

In Ergänzung an die im Kapitel 9.5.1 genannten Umbenennungen und Übersetzungen der Facetten und deren Inhalte wurde der Reiter „Beschreibung“ in der Titelanzeige, über den Illustrationsvermerke sowie Angaben zum Umfang und der ISBN eines Titels einsehbar sind, in „weitere Informationen zu diesem Titel“ umbenannt.<sup>236</sup>

Exemplare	weitere Informationen zu diesem Titel	Kommentare	Rezensionen	Internformat
Beschreibung: 96 S : Ill.				
ISBN: 978-3-86828-118-7 :				

Abbildung 35: Geänderte Übersetzung des Reiters „Beschreibung“ in der Titelanzeige

### Kenntnisse und Fähigkeiten

Für die beschriebenen Anpassungen der Katalogoberfläche waren einfache HTML- und CSS-Kenntnisse ausreichend.

### Einstufung nach den Kriterien der Machbarkeit: 2a

<sup>236</sup> Änderungen des Wordings sind von der benutzten Katalogoberfläche („Theme“) unabhängig, sofern die entsprechenden Übersetzungsdateien unter vufind/web/lang verwendet werden.

## 10. Nicht gelöste Probleme

Der innerhalb von 10 Wochen entstandene bachelopac kann – mit einigen Abstrichen – bereits zur Literaturrecherche benutzt werden. Fast alle der Projektaufgaben konnten vollständig gelöst werden. Im Folgenden werden die den Verfassern bekannte Probleme zusammengefasst, die im Rahmen des Projektzeitraums nicht behoben werden konnten. Die Sortierung erfolgt in absteigender Reihenfolge der Dringlichkeit. Dabei decken sich einige der beschriebenen Probleme mit der „Baustellenliste“ von TUBFind.<sup>237</sup>

### Importfehler beim Datengrundstück

Knapp 8% der Datensätze, die zwischen dem 01.01.1990 und dem 11.05.2010 angelegt wurden, konnten nicht oder nicht korrekt importiert werden. Beim regelmäßigen Datenimport neuer Katalogdaten traten aber keine Importprobleme mehr auf. Es wäre zu überprüfen, ob die Importfehler durch Verwendung eines MARC21-Abzugs von der Verbundzentrale, der nicht über die Z39.50-Schnittstelle erstellt wurde, vermieden werden könnten, da das MARC21-Format eines Abzugs nicht exakt dem eines Z39.50-Dumps entspricht. Zudem bedarf es eine weitere Analysen der Metadaten, deren Ergebnisse in einer angepassten Importroutine umgesetzt werden müssten.<sup>238</sup>

### Keine Verknüpfungen von Datensätzen

Mehrbändige Werke sowie Zeitschriftenbände werden im bachelopac nicht mit den jeweils zugehörigen Exemplaren verknüpft. Die TUBHH hat für TUBFind bereits einen Lösung für Zeitschriftenbände entwickelt,<sup>239</sup> die jedoch für nicht für andere Verknüpfungen angewendet werden kann.<sup>240</sup>

---

<sup>237</sup> Vgl. TUB HAMBURG-HARBURG 2010 – TUBFind Baustellenliste.

<sup>238</sup> Vgl. KINSTLER 2010 – Re: Datenkonvertierung.

<sup>239</sup> siehe z.B. die Zeitschrift „Nature“ in TUBFind <http://katalog.tub.tu-harburg.de//Record/129292834>.

<sup>240</sup> Vgl. MARAARENS 2010 – Experteninterview.

### Keine Treffer bei trunkierten oder phrasierten Suchbegriffen, die Umlaute enthalten

Die Trunkierung oder Phrasierung von Suchtermini führt in VuFind mitunter zu keinen Treffern. Den technischen Hintergrund des Problems und mögliche Lösungswege beschreibt Oliver Marahrens im TUBFind-Blog.<sup>241</sup>

### Keine Unterscheidung zwischen Präsensts- und Ausleihbestand

Die Anzeige der Verfügbarkeitsinformation im bachelopac bezieht sich nur auf den DAIA-Service „lokale Einsehbarkeit“. Eine Unterscheidung zwischen Präsensts- und Ausleihbestand findet daher nicht statt. Obwohl Hinweise wie „Semesterapparat“ oder „Dauerausleihe Verwaltung“ via phpDAIA zur Verfügung stehen<sup>242</sup>, werden diese im Katalog nicht zur Anzeige gebracht.

### Datensätze enthalten unerwünschte (Steuer-)Zeichen

Die von PICAPlus nach MARC21 konvertierten Datensätze enthalten als letzte Zeichen der MARC-Felder 100\$a (Personenname) und 245\$b (Zusatz zum Titel) unerwünschte Punkte, Kommata und Schrägstriche, die zu Inhomogenitäten führen. So werden in der Verfasser-Facette für manche Autoren fälschlicherweise mehrfach aufgeführt angezeigt.



Abbildung 36: Unerwünschte Zeichen in Datensätzen

<sup>241</sup> MAHRAHRENS 2010 – Umlaute undTrunkierung.

<sup>242</sup> Beispiel: [http://83.169.45.69/phpDaia/daia\\_ws.php?ppn=389290750](http://83.169.45.69/phpDaia/daia_ws.php?ppn=389290750)

Ein Lösungsansatz wäre, die Indexierungssoftware SolrMarc so konfigurieren, dass die ungewünschten Zeichen bei der Indexierung entfernt werden.<sup>243</sup>

### Zahlreiche "unbekannte" Medientypen

Im bachelopac sind über 37700 Titel (vorwiegend Bücher) verzeichnet, deren Medientyp nicht erkannt wurde.

### Fehlende Bandangaben in der Trefferliste

Die MARC21-Felder 245\$n (Zählung eines Teils/einer Abteilung eines Werkes) und 245\$p (Titel eines Teils/einer Abteilung eines Werkes) werden zwar auf der Titelseite, nicht aber nicht in der Trefferliste angezeigt. Dies führt dazu, dass mehrbändige Werke in der Trefferliste nicht voneinander unterscheidbar sind.



Abbildung 37: Fehlende Bandangaben in der Trefferliste

### Informationsballast durch das Matchen von Suchtermini gegen elektronische Adressen

In der feldübergreifenden Suche werden elektronische Adressen des MARC21-Feldes 856\$u mit einbezogen. Dies führt mitunter zu Informationsballast, da beispielsweise eine Suche nach „GBV“ sämtliche Titel aufgelistet, für die der GBV ein Inhaltsverzeichnis bereitstellt.

---

<sup>243</sup> Vgl. KINSTLER 2010 – Re: Fragen über Fragen.

### Keine Bilder der deutschen Wikipedia

Die Einbindung von Bildern aus der Wikipedia funktioniert zwar mit den englischen, jedoch nicht mit der deutschen Ausgabe. Die technische Ursache hierfür konnte von den Verfassern nicht ermittelt werden. Als „Quickhack“ könnten möglicherweise Bilder der englischen Wikipedia benutzt werden. Sofern das Problem nach dem nächsten Update noch besteht, werden die die Verfasser das Problem den Entwicklern melden.

### Keine Anzeige der Ausleihstatus in der Trefferliste bei physischen Medien mit Inhaltsverzeichnissen

Die Unterdrückung des Ausleihstatus in ist nur bei elektronischen Medien erwünscht. Hier bedarf es einer Anpassung des entsprechenden Templates.



Abbildung 38: Fehlender Ausleihstatus in der Trefferliste



## 11. Bewertung der Machbarkeit

### 11.1. Tabellarische Zusammenfassung

Ausgehend von den im Kapitel „Bewertungskriterien“ entwickelten Kriterien um die Machbarkeit zu bewerten, der im Kapitel „Projektaufgaben“ definierten Anforderungen und Funktionen an den Katalog und des im Rahmen dieser Arbeit realisierten Kataloges erfolgt nun eine zusammenfassende tabellarische Be- und Auswertung der eigenen Kenntnisse und Fähigkeiten.

<b>Funktionen und Eigenschaften</b>	<b>Bewertung</b>	<b>Kommentare</b>
<b>Datenimport, Datenabgleich und Anzeige Verfügbarkeit</b>		
Import von Metadaten	1	Einschränkung: Nicht alle Datensätze wurden fehlerfrei importiert
Automatisierte Einspielung neuer Titel	2b	Nachnutzung TUB HH
Automatisierte Makulierung	4	War innerhalb des Projektzeitraums nicht mehr zu realisieren. Nachnutzung TUB HH denkbar.
Anzeige der Verfügbarkeit	3	Aktive Unterstützung durch die TUB HH. Funktionseinschränkung: Keine Unterscheidung Präsensts-/ Ausleihbestand
Anzeige des Standorts	2a	

Funktionen und Eigenschaften	Bewertung	Kommentare
<b>Suche und Stöbern</b>		
Facettierte Browsen <ul style="list-style-type: none"> <li>• Anzeige der Basisklassifikation</li> <li>• Anpassung der Reihenfolge</li> <li>• Wording</li> <li>• Checkbox zur Eingrenzung auf Abschlussarbeiten der HAW</li> </ul>	1 2b 2a 2a 2a	Nachnutzung TUB HH orientiert an beluga
Anzeige „Ähnliche Treffer“ <ul style="list-style-type: none"> <li>• unter Berücksichtigung der BKL</li> </ul>	1 2a	
Lehrbuchboosting	2b	auf Basis des beluga-Quellcodes
Fehlertolerante Suche <ul style="list-style-type: none"> <li>• Rechtschreibprüfung</li> <li>• Suchergebnisse nach dem closest match verfahren</li> <li>• Verwendung eines Stemmingalgorithmus für die deutsche Sprache</li> </ul>	1 2b 2b	Übernahme der Konfiguration von beluga  Konfiguration von beluga
Funktionierende Suchsyntax	1	Einschränkung: Trunkierung mit Umlauten. War innerhalb des Projektzeitraums nicht mehr zu realisieren. Nachnutzung der TUB HH denkbar.

<b>Funktionen und Eigenschaften</b>	<b>Bewertung</b>	<b>Kommentare</b>
<b>Personalisierung</b>		
Nutzer können Konten anlegen, um ...	2a	
<ul style="list-style-type: none"> <li>• Listen privat und öffentlich anzulegen</li> </ul>	1	
<ul style="list-style-type: none"> <li>• Titel zu kommentieren</li> </ul>	1	
<ul style="list-style-type: none"> <li>• Titel zu taggen</li> </ul>	1	
<ul style="list-style-type: none"> <li>• ihre Suchhistorie zu speichern</li> </ul>	1	
<b>Mashups und Kataloganreicherungen</b>		
Anzeige der Cover von		
<ul style="list-style-type: none"> <li>• Google Books</li> </ul>	2a	
<ul style="list-style-type: none"> <li>• Librarything</li> </ul>	2a	
<ul style="list-style-type: none"> <li>• Amazon</li> </ul>	2a	
<ul style="list-style-type: none"> <li>• OpenLibrary</li> </ul>	2a	
Anzeige von Amazon-Rezensionen	2a	
Anzeige von Informationen über Autoren aus der Wikipedia	2a	
Einbindung von Google Book Search	2a	
Einbindung einer Chatauskunft	2a	
<b>Weitere Funktionen</b>		
Suche und Titel können als E-Mail versendet werden	1	
RSS-Feeds für Suchergebnisse	1	
Zitierfunktion	1	
Exportfunktion von Metadaten in Literaturverwaltungsprogramme	1	
<b>Anpassung der Katalogoberfläche</b>		
Design	2a	
Benutzeroberfläche vollständig in deutscher Sprache, angepasstes Wording.	2a	

Tabelle 2: Bewertung der Machbarkeit

## 11.2. Beurteilung der eigenen Kenntnisse

Ausgehend des bereits beschriebenen Studienschwerpunktes der Verfasser, der eben nicht nur auf Informationstechnologien, sondern eher in dem Bereich der Informationsdienstleistungen liegt, ist es schwierig zu beurteilen, ob Absolventen eines bibliothekarischen Studiums einen zeitgemäßen OPAC auf Basis von VuFind realisieren können. Den im Rahmen dieser Arbeit entstandenen, nur bedingt einsetzbaren Katalog (siehe Kapitel „offene Probleme“), als einzige Referenz zu nehmen zur Beurteilung zu nehmen, greift als Bewertungsansatz bzw. als alleiniges Bewertungskriterium zu kurz.

Vielleicht wären einerseits einige der noch offenen Probleme mit fundierten Kenntnissen in der Programmiersprache PHP lösbar gewesen, so findet sich doch der Ursprung dieser Arbeit in eben den Wahlmodulen zur Informationsdienstleistung. Ohne das Bewusstsein für die Bedeutung zeitgemäßer Kataloge, die hands-on-Erfahrung, die während der für das Projekt beluga durchgeführten Usability-Studien, die den Verfassern die Schwierigkeiten von Nutzern mit herkömmlichen OPACs buchstäblich vor Augen führte, wäre diese Arbeit andererseits sicher nicht entstanden.

Berücksichtigt werden muss auch, dass ursprünglich nur einer der Verfasser (Marcel Stehle) über Erfahrung im Umgang mit dem Betriebssystem Ubuntu Linux verfügte, der andere Verfasser (Jörg Schmitt) sich aber im Rahmen dieser Arbeit Basiskenntnisse erarbeiten musste, die nicht innerhalb des Studiums erlangt worden sind bzw. nicht erlangt werden können. Diese und weitere Kenntnisse können aber auch kurzfristig erlernt bzw. vertieft werden in Abhängigkeit von der eigenen Motivation<sup>244</sup> bzw. der Möglichkeit zur Fortbildung<sup>245</sup> innerhalb eines Katalog-Projekts.

---

<sup>244</sup> oder wie es CHRISTENSEN 2010 – IT-bibliothekarische Kernkompetenzen in einem Blogeintrag zu IT-bibliothekarische Kernkompetenzen beschreibt:  
„Es ist ziemlich charakteristisch für viele Aufgaben im IT-Bereich, dass sie sich am besten in einer Art Schaffensrausch erledigen lassen. Und der endet halt manchmal nicht im Gleitzeitrahmen. Natürlich liegt es mir fern, ein grundsätzliche Bereitschaft zu Heimarbeit und endlosen Überstunden einzufordern. Aber die meisten, die ich kenne, machen so etwas von Zeit zu Zeit. Unter anderem bestimmt deswegen, weil Flow auch Spaß bedeutet.“

<sup>245</sup> Auch wenn CHRISTENSEN 2010 – IT-bibliothekarische Kernkompetenzen schreibt: „Es gibt wenig bis keine formale Aus- und Fortbildung für das, was IT-BibliothekarInnen machen.“

Das im Rahmen der Pflichtveranstaltungen zur Wissensorganisation und Datenstrukturierung erworbene Wissen über Klassifikationen, Metadaten und Information Retrieval versetzt Absolventen aber in die Lage, die in dieser Arbeit vorgenommenen Schritte vorzunehmen oder nachzuvollziehen. Zwar wurden in den Lehrveranstaltungen zur Datenstrukturierung keine Kenntnisse über das Metadatenformat MARC21 vermittelt, so reichen jedoch die erlernten Fähigkeiten zur Katalogisierung in PICA aus, um grundlegenden Kenntnisse über den Aufbau und die Funktion von Metadaten zu erlangen und diese auf andere Formate zu abstrahieren. Auch versteht man die Problematik, die im Rahmen der Metadatenkonvertierung und des Metadatenmappings entsteht.

Auch wenn die Verfasser im Rahmen ihres Studiums keine expliziten XML- und PHP-Kenntnisse erworben haben (was jedoch im Rahmen der Wahlpflichtmodule möglich gewesen wäre), war doch das grundsätzliche Verständnis von Markup-Sprachen und Skriptsprachen, das durch die gewonnenen Kenntnisse von HTML, CSS und JavaScript erlangt worden sind, einer der entscheidenden Faktoren für das Gelingen dieser Arbeit. Ohne diese Fähigkeiten wären auch die Änderungen am Design und die Integration einer Chatauskunft nicht möglich gewesen. Neben den im Studium erworbenen Kenntnissen und erlernten Fähigkeiten, waren aber vor allem auch die durch das Studium gewonnenen Kontakte innerhalb der Hamburger Bibliothekslandschaft von entscheidender Bedeutung.

Der Kontakt zu Anne Christensen, die uns einen Einblick in den Quellcode von beluga gewährte, entstand schon bereits im ersten Semester im Rahmen der Veranstaltung Informationsdienstleistungen, die von ihr durchgeführt wurde. In dieser Veranstaltung entstand auch der Kontakt zu Edlef Stabenau, der den Verfassern den Kontakt zu Oliver Marahrens vermittelte. Dies ist insoweit zu betonen, dass die Verfasser auch verschiedene, ihnen nicht näher bekannte Personen aus dem bibliothekarischen Umfeld angeschrieben und um Hilfe oder Informationen gebeten haben, diese aber meist gar nicht geantwortet haben.

Hier zeigt sich auch das Besondere des Bachelorstudiengangs Bibliotheks- und Informationsmanagements, das die Leiterin des Departments Information der HAW Hamburg Krauß-Leichert wie folgt beschreibt:

„Besonders stolz bin ich auf die Praxisnähe unserer Studiengänge und auf unsere gute Vernetzung. Im Bereich Bibliotheken verfügen wir weit über den Hamburger Raum hinaus über sehr gute Kontakte. Und wir haben sehr, sehr gute Kontakte zur Hamburger Medienwirtschaft.“<sup>246</sup>

---

<sup>246</sup> KRAUSS-LEICHERT 2008 – Hochschulausbildung.

## 12. Schlussbetrachtung

Ziel dieser Arbeit war es, herauszufinden, inwieweit Absolventen des bibliothekarischen Bachelorstudiengangs an der HAW Hamburg in der Lage sind, einen zeitgemäßen Katalog mit der Open-Source-Software VuFind zu realisieren.

Die in dieser Arbeit beschriebenen Erkenntnisse und Ergebnisse ergeben, dass es Bachelorabsolventen auf Basis der im Studium erworbenen Kenntnisse und erlernten Fähigkeiten möglich ist, einen Katalog 2.0 unter Verwendung von VuFind zu realisieren. Erfolgreich kann dies aber nur geschehen, wenn bereits bestehende Lösungen zugänglich und nachnutzbar sind und in Problemfällen auf die aktive Unterstützung anderer Anwender oder im Besonderen der Verbände zurückgegriffen werden kann. Wie die Verfasser selbst erfahren haben, ist es daher zwingend notwendig, aktiv auf ihnen fremde Personen zuzugehen, diese anzusprechen und um Hilfe zu bitten. Die Bedeutung der sozialen Komponente ist nicht zu vernachlässigen, sondern hervorzuheben. Durch die im Bachelorstudium vermittelten Kenntnisse können zwar neu auftretende Probleme und Schwierigkeiten bei der Umsetzung erkannt, analysiert und benannt, aber nur teilweise auf Basis der im Studium vermittelten Kenntnisse gelöst werden. Spezialisiert man sich während des Studiums an der HAW ausschließlich auf Informationstechnologie, wären eigenständige Problemlösungen vielleicht in größerem Umfang möglich. In diesem Zusammenhang zeigt sich vor allem ein Problem der im Einsatz befindlichen proprietären Systeme: Gäbe es offene Schnittstellen, wären viele Probleme, auf die die Verfasser im Rahmen der Umsetzung gestoßen sind, einfacher zu lösen oder wären nicht entstanden.

Will man aber einen Katalog mit VuFind umsetzen, muss vor allem die Bereitschaft vorhanden sein, sich in Thematiken einzulesen, Mailinglisten zu verfolgen und sich trotz fehlender Programmierkenntnisse in den Code einzuarbeiten.

Geht man aber von den fehlenden bzw. nur rudimentär vorhandenen Programmierkenntnissen der Verfasser und den mit Hilfe der Community in kurzer Zeit erreichten Ergebnissen aus, also eines bereits in weiten Teilen funktionsfähigen Katalogs, ergibt sich nur das Ergebnis: **You can do it, too!**



Abbildung 39: Tweet von @Kaess: "Kann's kaum erwarten, meiner EDV von @SBChemnitz #vufind schmackhaft zu machen..."

KROPF 2010 – Tweet



## Literaturverzeichnis

### **ANGERMEIER 2005 - Machbarkeitsstudie**

ANGERMEIER, Georg: *Projektmanagement-Lexikon : 769 deutsche Begriffserklärungen, 589 englische Begriffe, 214 Abkürzungen, Volltextsuche, 8213 Querverweise, Stichwortverzeichnis*. 1. Ausg. München : Projekt Magazin, 2005. – ISBN 3000181148

### **ANGERMEIER 2006 - Projektmanagement-Lexikon**

ANGERMEIER, Georg: *Machbarkeitsstudie, Definition im Projektmanagement-Glossar*. URL <http://www.projektmagazin.de/glossar/gl-0044.html>. – Überprüfungsdatum 2010-09-20

### **BEA, SCHEURER, HESSELMANN 2008 - Projektmanagement**

BEA, Franz Xaver ; SCHEURER, Steffen ; HESSELMANN, Sabine: *Projektmanagement*. Stuttgart : Lucius & Lucius, 2008 (Grundwissen der ÖkonomikBetriebswirtschaftslehre 2388). – ISBN 3000181148

### **BZB - OSWD 2010**

BIBLIOTHEKSSERVICE-ZENTRUM BADEN-WÜRTTEMBERG: *Online-Schlagwortnormdatei (OSWD)*. URL <http://swb.bsz-bw.de/DB=2.104/?COOKIE=U998,Pbszgst,117,B0728+,SY,NRecherche-DB,D2.104,E368cb19d-0,A,H,R195.158.176.108,FY> – Überprüfungsdatum 2010-09-20

### **BOENINGER 2009 - Meebo**

BOENINGER, Chad: *MeeboMe and Pidgin is like Reese's Peanut Butter Cups*. In: *Library Voice*. URL <http://libraryvoice.com/instant-messaging/meebome-and-pidgin-is-like-reeses-peanut-butter-cups>. – Aktualisierungsdatum: 2009-09-23 – Überprüfungsdatum 2010-09-20

### **BRANDES, ERLHOFF, SCHEMMANN 2009 - Designtheorie**

BRANDES, Uta ; ERLHOFF, Michael ; SCHEMMANN, Nadine: *Designtheorie und Designforschung*. Paderborn : Fink, 2009 (UTB Design 3152). – ISBN 9783770546640

### **BREEDING 2009 - Open Discovery Interfaces**

BREEDING, Marschall: *Open Discovery Interfaces*. In: *American Libraries* 40 (2009), 6/7, S. 40–41. URL <http://americanlibrariesmagazine.org/columns/open-discovery-interfaces>

### **BREEDING 2010 - The State of the Art**

BREEDING, Marschall: *The State of the Art in Library Discovery 2010*. In: *Computers in Libraries* 30 (2010), 1/2, S. 31–35

### **CHRISTENSEN 2007 – Katalog 2.0**

CHRISTENSEN, Anne: *Katalog 2.0 – ein guter Begriff?* In: *beluga-blog*. URL <http://beluga-blog.sub.uni-hamburg.de/blog/2007/09/13/katalog-20-ein-guter-begriff/>. – Aktualisierungsdatum: 2007-09-13 – Überprüfungsdatum 2010-09-19

**CHRISTENSEN 2007 - Über das Projekt**

CHRISTENSEN, Anne: *Über das Projekt*. In: beluga-blog. URL <http://beluga-blog.sub.uni-hamburg.de/blog/ueber-das-projekt/>. – Aktualisierungsdatum: 2007-09-06 – Überprüfungsdatum 2010-09-19

**CHRISTENSEN 2009 - Partizipative Entwicklung**

CHRISTENSEN, Anne: *Partizipative Entwicklung von Diensten in der Bibliothek 2.0 : Methoden und Ergebnisse aus Katalog-2.0-Projekten*. In: *Bibliotheksdienst* 43 (2009), Nr. 5, S. 527–537. URL [http://www.zlb.de/aktivitaeten/bd\\_neu/heftinhalte2009/Erschliessung010509BD.pdf](http://www.zlb.de/aktivitaeten/bd_neu/heftinhalte2009/Erschliessung010509BD.pdf) – Überprüfungsdatum 2010-09-20

**CHRISTENSEN 2010 - Der beratende Katalog**

CHRISTENSEN, Anne: *Der beratende Katalog*. Bicamp<sup>3</sup> Hannover 2010 . URL <http://www.slideshare.net/xenzen/der-beratende-katalog> . Überprüfungsdatum 2010-09-20

**CHRISTENSEN 2010 - Fokusgruppen**

CHRISTENSEN, Anne: *Die, die mal drunter leiden werden fragen: Fokusgruppen zum Katalog 2.0 mit Studierenden*. In: beluga blog. URL <http://beluga-blog.sub.uni-hamburg.de/blog/2009/01/31/die-die-mal-drunter-leiden-werden-fragen-fokusgruppen-zum-katalog-20-mit-studierenden/> – Überprüfungsdatum 2010-09-20

**CHRISTENSEN 2010 - IT-bibliothekarische Kernkompetenzen**

CHRISTENSEN, Anne: *IT-bibliothekarische Kernkompetenzen*. In: A growing organism Blog. URL <http://xenzen.wordpress.com/2010/09/11/it-bibliothekarische-kernkompetenzen/>. – Aktualisierungsdatum: 2010-09-11 – Überprüfungsdatum 2010-09-20

**CHRISTENSEN 2010 – Kommentar zu make or buy**

CHRISTENSEN, Anne: *Kommentar zu : Bibliothekssoftware im Eigenbau: "Make or buy", revisited*. In: beluga-blog. URL <http://beluga-blog.sub.uni-hamburg.de/blog/2010/05/28/make-or-buy/#comment-422>. – Aktualisierungsdatum: 2010-05-31 – Überprüfungsdatum 2010-09-20

**DANOWSKI, HELLER 2006 - Bibliothek 2.0**

DANOWSKI, Patrick ; HELLER, Lambert: *Bibliothek 2.0 : Die Zukunft der Bibliothek*. In: *Bibliotheksdienst* 40 (2006), Nr. 11, S. 1259–1271. URL [http://www.zlb.de/aktivitaeten/bd\\_neu/heftinhalte2006/DigitaleBib011106.pdf](http://www.zlb.de/aktivitaeten/bd_neu/heftinhalte2006/DigitaleBib011106.pdf) – Überprüfungsdatum 2010-09-19

**DIEROLF 2007 - Der Uni-Katalog Karlsruhe**

DIEROLF, Uwe: *Der Uni-Katalog Karlsruhe : Eine Bibliothekskatalog im Wandel*. In: *B.I.T. Online* 10 (2007), Nr. 2, S. 130–134. – Aktualisierungsdatum: 2007-06-04 – Überprüfungsdatum 2010-09-21

### **DIEROLF, MÖNNICH 2006 - Recommendersysteme**

DIEROLF, Uwe ; MÖNNICH, Michael: *Einsatz von Recommendersystemen in Bibliotheken*. In: *B.I.T. Online* 9 (2006), Nr. 1, S. 27–30. URL <http://www.b-i-t-online.de/archiv/2006-01/nach1.htm>.

### **DISTROWATCH 2010 - Ein Wegweiser**

DISTROWATCH: *Ein Wegweiser zur Auswahl einer Distribution : Ubuntu*. URL <http://distrowatch.com/dwres.php?resource=major&language=DE> – Überprüfungsdatum 2010-09-20

### **DNB 2010 - Datendienst Biografische Dienstleistungen**

DNB: *Datendienst Biografische Dienstleistungen*. URL <http://www.d-nb.de/service/zd/datendienst.htm>. – Aktualisierungsdatum: 2010-09-17 – Überprüfungsdatum 2010-09-19

### **DRAUSZ, PLIENINGER 2010 – Nutzerwünsche**

DRAUSZ, Susanne ; PLIENINGER, Jürgen: *Nutzerwünsche sind nur bedingt RAK-kompatibel : So wird der Katalog zukunftstauglich: Recommenderdienste – Anreicherungen – Katalog 2.0 – Table of Contents*. In: *BuB Forum Bibliothek und Information* 62 (2010), Nr. 1, S. 40–48. URL [http://www.b-u-b.de/cgi-local/byteserver.pl/pdfarchiv/Heft-BuB\\_01\\_2010.pdf](http://www.b-u-b.de/cgi-local/byteserver.pl/pdfarchiv/Heft-BuB_01_2010.pdf) – Überprüfungsdatum 2010-09-19

### **FLIMM 2010 - Offene bibliographische Daten**

FLIMM, Oliver: *Offene bibliographische Daten: Es ist soweit!* In: *OpenBibBlog*. URL <http://blog.openbib.org/2010/03/12/offene-bibliographische-daten-es-ist-soweit/>. – Aktualisierungsdatum: 2010-06-28 – Überprüfungsdatum 2010-09-20

### **GBV 2010 – Target Profile**

GBV: *Z39.50 Target-Profile*. URL [http://www.gbv.de/de/services/info/z39\\_50?opac:int=1](http://www.gbv.de/de/services/info/z39_50?opac:int=1). – Aktualisierungsdatum: 2010-09-20 – Überprüfungsdatum 2010-09-20

### **GBV 2010 – Z39.50**

GBV: *Informationen zum Z39.50 Zugang des GBV*. URL [http://www.gbv.de/vgm/info/benutzer/04extras/extras\\_0065](http://www.gbv.de/vgm/info/benutzer/04extras/extras_0065). – Aktualisierungsdatum: 2010-08-03 – Überprüfungsdatum 2010-09-20

### **GBV 2010 – Z39.50 HAW**

GBV: *Hochschule fuer angewandte Wissenschaften Hamburg : Z39.50-Profil*. URL [http://www.gbv.de/de/services/info/z39\\_50/viewProfile?zTolk=2@lhopc4.rrz.uni-hamburg.de](http://www.gbv.de/de/services/info/z39_50/viewProfile?zTolk=2@lhopc4.rrz.uni-hamburg.de) – Überprüfungsdatum 2010-09-20

### **GBV Wiki 2010 – DAIA-Wrapper**

VOSS, Jakob; MARAHRENS, Oliver: *Erstellung und Wartung eines DAIA-Wrappers*. URL [http://www.gbv.de/wikis/cls/Erstellung\\_und\\_Wartung\\_eines\\_DAIA-Wrappers](http://www.gbv.de/wikis/cls/Erstellung_und_Wartung_eines_DAIA-Wrappers). – Aktualisierungsdatum: 2010-05-11 – Überprüfungsdatum 2010-09-20

**GBV WIKI 2010 – Verfügbarkeitsrecherche mit DAIA**

VOSS, Jakob; REH, Uwe: *Verfügbarkeitsrecherche mit DAIA*. URL

<http://www.gbv.de/wikis/cls/index.php?title=Verf>

%C3%BCgbarkeitsrecherche\_mit\_DAIA&action=edit. – Aktualisierungsdatum: 2010-05-20 –  
Überprüfungsdatum 2010-09-20

**GOOGLE 2010 – Preview Wizard**

GOOGLE: *Preview Wizard - Google Book Search APIs*. URL <http://code.google.com/intl/de-DE/apis/books/docs/preview-wizard.html> – Überprüfungsdatum 2010-09-20

**GRINDLE 2010 - SOA**

GRINDLE, Ronald D.: *SOA testen : Der inkrementelle fachliche Integrations-Test (IFIT)*. URL [www.qa-navigation.com/Downloads/QANav-IFIT-MW.pdf](http://www.qa-navigation.com/Downloads/QANav-IFIT-MW.pdf). – Aktualisierungsdatum: 2010-06-11 –  
Überprüfungsdatum 2010-09-20

**HAMANN 2006 - Dienstorientierte Architektur SOA**

HAMANN, Kristof: *Dienstorientierte Architektur (SOA)*. URL

<http://daten.korelstar.de/dokumente/soa.pdf>. – Aktualisierungsdatum: 2006-10-21 –

Überprüfungsdatum 2010-09-20

**HEIDENREICH 2003 - Lucene**

HEIDENREICH, Andreas: *Lucene - Eine einfache Suchmaschine für Freitextsuche : Die Standard-Suchsyntax von Lucene*. URL <http://www.fh-wedel.de/~si/seminare/ws02/Ausarbeitung/e.lucene/2.html>. – Aktualisierungsdatum: 2003-02-18 –  
Überprüfungsdatum 2010-09-20

**HELLER 2007 - Drei Beobachtungen zu Chat-Widgets**

HELLER, Lambert: *Drei Beobachtungen zu Chat-Widgets*. In: netbib weblog. URL

<http://log.netbib.de/archives/2007/12/07/drei-beobachtungen-zu-chat-widgets/>. –

Aktualisierungsdatum: 2007-12-07 – Überprüfungsdatum 2010-09-20

**HELLER 2009 - Was ist ein Katalog 2.0**

HELLER, Lambert: *Was ist ein Katalog 2.0? : Versuch einer Definition*. In: *Biblionik*. URL

<http://biblionik.de/2009/12/21/katalog-20/>. – Aktualisierungsdatum: 2009-12-21 –

Überprüfungsdatum 2010-09-19

**HELMKAMP, OEHLISCHLÄGER 2006 - Internationalisierung**

HELMKAMP, Kerstin ; OEHLISCHLÄGER, Susanne: *Die Internationalisierung deutscher Standards : auf dem Weg von MAB2 zu MARC 21*. „Access“ – der Schlüssel zum Erfolg. In: *Bibliotheksdienst* 40 (2006),

Nr. 11. URL [http://www.zlb.de/aktivitaeten/bd\\_neu/heftinhalte2006/Erschliessung011106.pdf](http://www.zlb.de/aktivitaeten/bd_neu/heftinhalte2006/Erschliessung011106.pdf). –

Aktualisierungsdatum: 2007-01-24 – Überprüfungsdatum 2010-09-20

**Ho, KELLY, GARRISON 2009 - Interoperability**

Ho, Birong ; KELLY, Keith ; GARRISON, Scott: *Implementing VuFind as an alternative to Voyager's WebVoyage interface : One library's experience*. In: *Library Hi Tech* 27 (2009), Nr. 1, S. 82–92. URL [www.emeraldinsight.com/0737-8831.htm](http://www.emeraldinsight.com/0737-8831.htm) – Überprüfungsdatum 2010-09-19

**HOBOHM 2007 - Bibliothekswissenschaft 2.0**

HOBOHM, Hans-Christoph: *Bibliothek(swissenschaft) 2.0 : Neue Auflage oder Wende in Forschung und Lehre?* Vortrag auf dem 2. gemeinsamen Bibliothekstag Berlin/Brandenburg am. In: *Libreas* (2007), Nr. 10. URL <http://www.libreas.eu/ausgabe10/003hob.htm>

**HOUGHTON-JAN 2007 - Adding MeeboMe Widget**

HOUGHTON-JAN, Sarah: *Adding MeeboMe Widget to the library catalog*. In: Librarian In Black blog. URL <http://librarianinblack.typepad.com/librarianinblack/2007/12/adding-meebome.html>. – Aktualisierungsdatum: 2007-12-04 – Überprüfungsdatum 2010-09-20

**HOUSER 2009 - The VuFind implementation**

HOUSER, John: *The VuFind implementation at Villanova University*. In: *Library Hi Tech* 27 (2009), Nr. 1, S. 93–105. URL [www.emeraldinsight.com/0737-8831.htm](http://www.emeraldinsight.com/0737-8831.htm) – Überprüfungsdatum 2010-09-19

**VON DER HUDE 2010 - SRU**

HUDE, Nicole von der: *SRU im Überblick* <http://www.d-nb.de/service/zd/sru.htm>. – Aktualisierungsdatum: 2010-09-02 – Überprüfungsdatum 2010-09-20

**IMHOF 2006 - RSWK/SWD und Faceted Browsing**

IMHOF, Andres: *RSWK/SWD und Faceted Browsing : neue Möglichkeiten einer inhaltlich-intuitiven Navigation*. URL <http://www.zib.de/Publications/Reports/ZR-06-34.pdf>. – Aktualisierungsdatum: 2006-06-27 – Überprüfungsdatum 2010-09-20

**INDEX DATA 2010 – PHP/YAZ**

INDEX DATA: *PHP/YAZ*. URL <http://www.indexdata.com/phpyaz> – Überprüfungsdatum 2010-09-20

**JUHST 2009 – Kompetent IM Netz**

JUHST, Sabrina: *Kompetent IM Netz: Machbarkeitsstudie zum Einsatz von Instant Messaging in der digitalen Auskunft an der Bibliothek der Helmut-Schmidt-Universität (Hamburg)*. URL <http://opus.unibw-hamburg.de/opus/volltexte/2009/2139/pdf/Schriftenreihe-Band1.pdf>. - Aktualisierungsdatum: 2009-10-02 – Überprüfungsdatum 2010-09-20

**KATZ 2009 - System Classes**

KATZ, Demian: *System Classes*. URL [http://vufind.org/wiki/system\\_classes](http://vufind.org/wiki/system_classes). – Aktualisierungsdatum: 2009-08-14 – Überprüfungsdatum 2009-09-20

**KATZ 2010 - Re: Autostemming**

KATZ, Demian: *Old Nabble - Re: Autostemming*. In: VuFind-General-Mailing-List. URL <http://old.nabble.com/Re%3A-Autostemming-p28341317.html>. – Aktualisierungsdatum: 2010-04-23 – Überprüfungsdatum 2010-09-20

**KATZ 2010 - Installation problems**

KATZ, Demian: *Re: Installation problems and MARC upload problems*. In: VuFind General Mailing-List. URL <http://old.nabble.com/Re:-Installation-problems-and-MARC-upload-problems-p27624633.html>. – Aktualisierungsdatum: 2010-02-17 – Überprüfungsdatum 2010-09-20

**KINSTLER 2010 – Hacking VuFind**

KINSTLER, Till: *Hacking VuFind into german libraries*. URL [http://vufind.org/wiki/hacking\\_vufind\\_into\\_german\\_libraries](http://vufind.org/wiki/hacking_vufind_into_german_libraries). – Aktualisierungsdatum: 2009-07-03 – Überprüfungsdatum 2010-10-20

**KINSTLER 2010 – Re: Datenkonvertierung**

SCHMITT, Jörg ; STEHLE, Marcel: *Datenkonvertierung*. E-Mail-Interview mit Till Kinster. 06.05.2010

**KINSTLER 2010 – Re: Fragen über Fragen**

SCHMITT, Jörg ; STEHLE, Marcel: *Fragen über Fragen*. E-Mail-Interview mit Till Kinster. 28.05.2010

**KINSTLER 2010 - Kommentar zu Kataloge auf dem Weg ins Internet**

KINSTLER, Till: *Kommentar zu : Kataloge auf dem Weg ins Internet*. In: netbib weblog. URL <http://log.netbib.de/archives/2010/04/19/kataloge-auf-dem-weg-ins-internet/#comment-45239>. – Aktualisierungsdatum: 2010-04-19 – Überprüfungsdatum 2010-09-19

**KINSTLER 2010 - Kommentar zu make or buy**

KINSTLER, Till: *Kommentar zu : Bibliothekssoftware im Eigenbau: "Make or buy", revisited*. In: beluga-blog. URL <http://beluga-blog.sub.uni-hamburg.de/blog/2010/05/28/make-or-buy/#comment-421>. – Aktualisierungsdatum: 2010-05-31 – Überprüfungsdatum 2010-09-20

**KINSTLER 2010 - Re: Deleting records**

KINSTLER, Till: *Re: Deleting records*. In: VuFind General-Mailing-List. URL <http://old.nabble.com/Re:-Deleting-records--p27562530.html>. – Aktualisierungsdatum: 2010-02-12 – Überprüfungsdatum 2010-09-20

**KINSTLER 2010 – Tweet**

KINSTLER, Till: *Twitter / Till Kinster: @bachelopac Gibt ein Nutzer 3 bis 5 Suchterme ein, darf 1 in den Treffern fehlen, bei 6 dürfen 2 fehlen, bei mehr als 6 müssen 90% matchen*. URL <http://twitter.com/tillk/status/21295472256> – Überprüfungsdatum 2010-09-20

**KINSTLER 2010 – VuFind**

KINSTLER, Till: *VuFind - Ein Open Source Projekt*. 4. Leipziger Bibliothekskongress. Leipzig. URL [http://www.gbv.de/vgm/info/biblio/01VZG/06Publikationen/2010/pdf/pdf\\_4270.pdf](http://www.gbv.de/vgm/info/biblio/01VZG/06Publikationen/2010/pdf/pdf_4270.pdf). – Aktualisierungsdatum: 2010-03-17 – Überprüfungsdatum 2010-10-20

**KNEIFEL 2009 – Mit Web 2.0**

KNEIFEL, Fabienne: *Mit Web 2.0 zum Online-Katalog der nächsten Generation : Innovationspreis 2009*. Humboldt-Univ., Master-Arbeit u.d.T.: Kneifel, Fabienne: Welche Funktionen und Inhalte sollte ein Bibliothekskatalog im Zeitalter des Web 2.0 bieten?--Berlin, 2008. Wiesbaden : Dinges & Frick, 2009 (B.I.T. onlineInnovativ 23). – ISBN 978-3-934997-26-4

**KOCH, RICHTER 2009 - Enterprise 2.0**

KOCH, Michael ; RICHTER, Alexander: *Enterprise 2.0 : Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen*. 2., aktualisierte und erw. Aufl. München : Oldenbourg, 2009. – ISBN 9783486593648

**KOSTÄDT 2008 - Der Weg in die Zukunft**

KOSTÄDT, Peter: Der Weg in die Zukunft. In: HUTZLER, Evelinde; SCHRÖDER, Albert; SCHWEIKL, Gabriele; GEISSELMANN, Friedrich (Hrsg.): *Bibliotheken gestalten Zukunft : Kooperative Wege zur Digitalen Bibliothek /// Bibliotheken gestalten Zukunft : Kooperative Wege zur Digitalen Bibliothek ; Dr. Friedrich Geißelmann zum 65. Geburtstag*. Göttingen, Göttingen : Univ.-Verl. Göttingen; Niedersächsische Staats- und Universitätsbibliothek., 2008. – ISBN 978-3-940344-43-4, S. 113

**KRAUSS-LEICHERT 2008 - Ausbildung im Wandel**

KRAUSS-LEICHERT, Ute: *Bibliothekarische Ausbildung im Wandel*. URL <http://www.goethe.de/wis/bib/fdk/de60683.htm>. – Aktualisierungsdatum: 2010-09-18 – Überprüfungsdatum 2010-09-20

**KRAUSS-LEICHERT 2008 - Hochschulausbildung**

KRAUSS-LEICHERT, Ute: "Nur was sich ändert bleibt..." : *Zehn Jahre Hochschulausbildung im bibliothekarischen Bereich*. In: *BIT online* 11 (2008), Nr. 1, S. 4–6. URL <http://www.b-i-t-online.de/heft/2008-01-edit-krauss.htm>

**KRAUSS-LEICHERT 2008 - Praxisnahe Ausbildung**

KRAUSS-LEICHERT, Ute: *Praxisnahe Ausbildung zum Informationsspezialisten*. URL <http://www.goethe.de/wis/bib/fdk/de3222990.htm>. – Aktualisierungsdatum: 2008-04-01 – Überprüfungsdatum 2010-09-20

**KROPF 2010 – Tweet**

KROF, Katrin : *Twitter /Kaes : Kann's kaum erwarten, meiner EDV von @SBChemnitz #vufind schmackhaft zu machen.. Wenn das sogar angehende Bibliothekare schaffen ;) #os\_htwk*. URL <http://twitter.com/Kaess/status/16849419930> – Überprüfungsdatum 2010-06-23

**LANGENSTEIN, MAYLEIN 2009 - Relevanz-Ranking**

LANGENSTEIN, Annette ; MAYLEIN, Leonhard: *Relevanz-Ranking im OPAC der Universitätsbibliothek Heidelberg*. In: *B.I.T.online* 12 (2009), Nr. 4, S. 408–413. URL [http://archiv.ub.uni-heidelberg.de/volltextserver/volltexte/2010/10343/pdf/Langenstein\\_Maylein\\_aus\\_BIT\\_4\\_09\\_kpl\\_kl.pdf](http://archiv.ub.uni-heidelberg.de/volltextserver/volltexte/2010/10343/pdf/Langenstein_Maylein_aus_BIT_4_09_kpl_kl.pdf). – Aktualisierungsdatum: 2010-01-26 – Überprüfungsdatum 2010-09-20

**LEWANDOWSKI 2009 - Suchmaschinen vs. OPAC.**

LEWANDOWSKI, Dirk: *Suchmaschinen vs. OPAC*. Arbeitstagung der Fraunhofer-Bibliotheken 2009. URL [http://www.bui.fh-hamburg.de/fileadmin/user\\_upload/lewandowski/vortraege/Bremen2009\\_Lewandowski.pdf](http://www.bui.fh-hamburg.de/fileadmin/user_upload/lewandowski/vortraege/Bremen2009_Lewandowski.pdf). – Aktualisierungsdatum: 2009-10-15 – Überprüfungsdatum 2010-09-19

**LEWANDOWSKI 2009 - Wie Suchmaschinen**

LEWANDOWSKI, Dirk: *Wie Suchmaschinen (immer noch) die Bibliotheken herausfordern*. URL [http://www.bui.haw-hamburg.de/fileadmin/user\\_upload/lewandowski/vortraege/Regensburg2009\\_Lewandowski.pdf](http://www.bui.haw-hamburg.de/fileadmin/user_upload/lewandowski/vortraege/Regensburg2009_Lewandowski.pdf). – Aktualisierungsdatum: 2009-09-18 – Überprüfungsdatum 2010-09-19

**LIBRARY OF CONGRESS 2010 – MARC21**

LIBRARY OF CONGRESS NETWORK DEVELOPMENT AND MARC STANDARDS OFFICE: *MARC 21 Format for Bibliographic Data : Update No. 11*. URL <http://www.loc.gov/marc/bibliographic/>. – Aktualisierungsdatum: 2010-04-06 – Überprüfungsdatum 2010-09-20

**LIBRARY SUCCESS WIKI 2010 – IM**

LIBRARY SUCCESS: A BEST PRACTICES WIKI: *Libraries Using IM Reference*. URL [http://www.libsuccess.org/index.php?title=Libraries\\_Using\\_IM\\_Reference](http://www.libsuccess.org/index.php?title=Libraries_Using_IM_Reference). – Aktualisierungsdatum: 2010-08-31 – Überprüfungsdatum 2010-09-20

**LIBRARYH3LP 2010 - Pay for LibraryH3lp**

LIBRARYH3LP: *Pay for LibraryH3lp*. URL <http://libraryh3lp.com/static/payment.html>. – Aktualisierungsdatum: 2010-08-26 – Überprüfungsdatum 2010-09-20

**MARAHRENS 2010 - Änderung**

MARAHRENS, Oliver: *Große Änderung, (erstmal) kleine Auswirkung*. In: TUBfind Blog. URL <http://www.tub.tu-harburg.de/tubfind-blog/2010/08/groese-anderung-erstmal-kleine-auswirkung/>. – Aktualisierungsdatum: 2010-08-25 – Überprüfungsdatum 2010-09-20

**MARAHRENS 2010 - Experteninterview**

SCHMITT, Jörg ; STEHLE, Marcel: VuFind in der TUB HH. Interview mit Oliver Marahrens.23.06.2010.

**MARAHRENS 2010 - Die Zukunft**

MARAHRENS, Oliver: *Die Zukunft kommt manchmal schneller als man denkt*. In: *Weblog der Universitätsbibliothek der TUHH*. URL <http://www.tub.tu-harburg.de/blog/2010/04/01/die-zukunft-kommt-manchmal-schneller-als-man-denkt/>. – Aktualisierungsdatum: 2010-04-01 – Überprüfungsdatum 2010-09-19

**MARAHRENS 2010 – TUBFind**



MARAHRENS, Oliver: *TUBFind : Open Source Alternative für den Katalog*. Verbundkonferenz des GBV. URL [http://www.google.de/url?sa=t&source=web&cd=1&ved=0CBYQFjAA&url=http%3A%2F%2Fwww.tub.tu-harburg.de%2Ftubfind-blog%2Fwp-content%2Fuploads%2F2010%2F09%2Fvufind\\_gbv\\_vk\\_2010.ppt&rct=j&q=VuFind\\_gbv\\_vk\\_2010.ppt&ei=e5-XTJXXIYz84Abm5uyKBA&usq=AFQjCNFjg2OEMRDA\\_y1HCpAwP0Cs97p2dw](http://www.google.de/url?sa=t&source=web&cd=1&ved=0CBYQFjAA&url=http%3A%2F%2Fwww.tub.tu-harburg.de%2Ftubfind-blog%2Fwp-content%2Fuploads%2F2010%2F09%2Fvufind_gbv_vk_2010.ppt&rct=j&q=VuFind_gbv_vk_2010.ppt&ei=e5-XTJXXIYz84Abm5uyKBA&usq=AFQjCNFjg2OEMRDA_y1HCpAwP0Cs97p2dw). – Aktualisierungsdatum: 2010-09-09 – Überprüfungsdatum 2010-09-20

#### **MARAHRENS 2010 – Tweet**

MARAHRENS, Oliver: *Twitter / Olli: @bachelopac ja, wir benutzen derzeit keine Stopworte*. URL <http://twitter.com/ollmii/status/21922313826>. – Aktualisierungsdatum: 2010-08-23 – Überprüfungsdatum 2010-09-21

#### **MARAHRENS 2010 - Umlaute und Trunkierung**

MARAHRENS, Oliver: *Umlaute und Trunkierung*. In: TUBfind Blog. URL <http://www.tub.tu-harburg.de/tubfind-blog/2010/08/umlaute-und-trunkierung/>. – Aktualisierungsdatum: 2010-08-19 – Überprüfungsdatum 2010-09-20

#### **MARAHRENS 2010 - Verfügbarkeitsinformationen**

MARAHRENS, Oliver: *Verfügbarkeitsinformationen aus dem Bibliothekssystem extrahieren*. In: Weblog der Universitätsbibliothek der TUHH. URL <http://www.tub.tu-harburg.de/blog/2010/05/26/verfuegbarkeitsinformationen-aus-dem-bibliothekssystem-extrahieren/> – Überprüfungsdatum 2010-09-20

#### **MARKEY 2007 - End-User Searching**

MARKEY, Karen: *Twenty-Five Years of End-User Searching : Part 1: Research Findings*. In: *Journal of the American Society for Information Science and Technology* 58 (2007), Nr. 8, S. 1071–1081

#### **NAGY 2008 - VuFind Release**

NAGY, Andrew: *VuFind Release 1.0RC1*. URL <http://old.nabble.com/VuFind-Release-1.0RC1-to-20003617.html#a20003617>. – Aktualisierungsdatum: 2008-10-15 – Überprüfungsdatum 2010-09-20

#### **OPEN ARCHIVES INITIATIVE 2010 - Interoperability**

OPEN ARCHIVES INITIATIVE: *Interoperability through Metadata Exchange*. URL <http://www.openarchives.org/pmh/> – Überprüfungsdatum 2010-09-20

#### **O'REILLY 2005 - What Is Web 2.0**

O'REILLY, Tim: *What Is Web 2.0 : Design Patterns and Business Models for the Next Generation of Software*. URL <http://oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=5>. – Aktualisierungsdatum: 2005-09-30 – Überprüfungsdatum 2010-09-19

### **O'REILLY 2008 - Was ist Web 2.0**

O'REILLY, Tim: *Was ist Web 2.0? : Entwurfsmuster und Geschäftsmodelle für die nächste Software Generation*. URL [http://www.oreilly.de/artikel/web20\\_trans.html](http://www.oreilly.de/artikel/web20_trans.html). – Aktualisierungsdatum: 2008-07-21 – Überprüfungsdatum 2010-09-20

### **Reh 2009 - DAIA**

REH, Uwe: *DAIA : Ein neuer Baustein für Bibliotheken im Web 2.0*. Bibliothekartag 2009. In: *HeBIScocktail* (2009), S. 5–6. URL [http://www.hebis.de/de/1publikationen/hebis-cocktail/bibtag2009/sonderausgabe\\_2009.pdf?PHPSESSID=d7952cd686ae1cce879122c33a9f090b](http://www.hebis.de/de/1publikationen/hebis-cocktail/bibtag2009/sonderausgabe_2009.pdf?PHPSESSID=d7952cd686ae1cce879122c33a9f090b)

### **RENNER 2005 - Open Source Software**

RENNER, Thomas: *Open Source Software : Einsatzpotenziale und Wirtschaftlichkeit ; eine Studie der Fraunhofer-Gesellschaft*. Stuttgart : IRB-Verl.; Fraunhofer-Inst. für Arbeitswirtschaft u. Organisation, 2005. – ISBN 978-3-8167-7008-4

### **DE REUS 2010 - FRBR**

REUS, Irena de: *Functional Requirements for Bibliographic Records : Der Weg zum neuen Katalog?* Fachhochschule Köln : Seminar vom 26.05.2010. URL [http://www.fbi.fh-koeln.de/institut/personen/de\\_reus/material/26\\_05\\_2010\\_Server.pdf](http://www.fbi.fh-koeln.de/institut/personen/de_reus/material/26_05_2010_Server.pdf). – Aktualisierungsdatum: 2010-05-20 – Überprüfungsdatum 2010-09-20

### **ROCHKIND 2010 - Stop Words**

ROCHKIND, Jonathan: *solr stop words/dismax gotcha*. In: Bibliographic Wilderness blog. URL <http://bibwild.wordpress.com/2010/04/14/solr-stop-wordsdismax-gotcha/>. – Aktualisierungsdatum: 14.042010 – Überprüfungsdatum 2010-09-20

### **RUSCH 2008 – MARC21**

RUSCH, Beate: *MARC 21 kommt nach Deutschland : Eine Einführung*. MARC-Workshop, 26. Februar 2008 im DAI. URL [http://www.kobv.de/fileadmin/download/vortraege/rusch\\_20080226\\_marc.pdf](http://www.kobv.de/fileadmin/download/vortraege/rusch_20080226_marc.pdf). – Aktualisierungsdatum: 2008-02-27 – Überprüfungsdatum 2010-09-20

### **SCHMIDT 2010 - Modulhandbuch**

SCHMIDT, Ralph: *Modulhandbuch : Bibliotheks- und Informationsmanagement (Bachelor of Arts)*. URL <http://www.bui.haw-hamburg.de/fileadmin/redaktion/DeptInfo3.pdf>. – Aktualisierungsdatum: 2007-06-04 – Überprüfungsdatum 2010-09-20

### **SCHMITT, STEHLE 2009 - Stimmen aus dem Usability-Labor**

SCHMITT, Jörg; STEHLE, Marcel: *“Da oben habe ich nie wirklich hingeguckt” : Stimmen aus dem Usability-Labor*. In: beluga blog. URL <http://beluga-blog.sub.uni-hamburg.de/blog/2009/08/07/usability-2009/>. – Aktualisierungsdatum: 2009-08-07 – Überprüfungsdatum 2010-09-20

**SCHMITT, STEHLE 2010 – Tweet**

SCHMITT, Jörg; STEHLE, Marcel: *Twitter / Jörg und Marcel : Was bedeutet denn diese Konfig. auf deutsch?* <str name="mm">2<-1 5<-2 6<90%</str> #Solr #MinimumShouldMatch #Followerpower. URL <http://twitter.com/bachelopac/status/21257807972>. – Aktualisierungsdatum: 2010-08-15 – Überprüfungsdatum 2010-09-20

**SCHULZ 1994 - Was wir über OPAC-Nutzer wissen**

SCHULZ, Ursula: *Was wir über OPAC-Nutzer wissen: Fehlertolerante Suchprozesse in OPACs*. In: *ABI-Technik* 14 (1994), Nr. 4, S. 299–310. URL [http://www.bui.haw-hamburg.de/pers/ursula.schulz/publikationen/opac\\_fehlertolerant.pdf](http://www.bui.haw-hamburg.de/pers/ursula.schulz/publikationen/opac_fehlertolerant.pdf) – Überprüfungsdatum 2010-09-20

**SCHULZ 1998 - Wie der Schnabel**

SCHULZ, Ursula: *"Wie der Schnabel gewachsen ist" : über die Qualität von OPACs*. In: *BuB Forum Bibliothek und Information* (1998), Nr. 5, S. 345–351. URL [http://www.bui.haw-hamburg.de/pers/ursula.schulz/publikationen/opac\\_qualitaet.pdf](http://www.bui.haw-hamburg.de/pers/ursula.schulz/publikationen/opac_qualitaet.pdf). – Aktualisierungsdatum: 2007-02-06 – Überprüfungsdatum 2010-09-21

**SNOOPY, SCHMITZ, MÜLLER 1999 - Open source**

SNOOPY ; SCHMITZ, Sebastian Snoopy ; MÜLLER, Martin: *Open source : Kurz & gut*. 1. Aufl. Köln : O'Reilly, 1999 (O'Reillys TaschenbibliothekSonderband). – ISBN 978-3-89721-222-0

**SOLR 2010 - DisMaxQParserPlugin**

SOLR: *DisMaxQParserPlugin - Solr Wiki*. URL <http://wiki.apache.org/solr/DisMaxQParserPlugin> – Überprüfungsdatum 2010-09-20

**SOLR WIKI 2010 - AnalyzersTokenizersTokenFilters**

SOLR WIKI: *AnalyzersTokenizersTokenFilters*. URL <http://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters>. – Aktualisierungsdatum: 2010-08-28 – Überprüfungsdatum 2010-09-20

**SOLR WIKI 2010 - DisMaxQParserPlugin**

SOLR WIKI: *DisMaxQParserPlugin*. URL <http://wiki.apache.org/solr/DisMaxQParserPlugin>. – Aktualisierungsdatum: 2010-07-30 – Überprüfungsdatum 2010-09-21

**SOLR WIKI 2010 - LanguageAnalysis**

SOLR WIKI: *LanguageAnalysis : German*. URL <http://wiki.apache.org/solr/LanguageAnalysis#German>. – Aktualisierungsdatum: 2010-07-14 – Überprüfungsdatum 2010-09-20

**SOLR WIKI 2010 - MoreLikeThis**

SOLR WIKI: *MoreLikeThis - Solr Wiki*. URL <http://wiki.apache.org/solr/MoreLikeThis> – Überprüfungsdatum 2010-09-20

**STABENAU 2010 - zeitgemäße Kataloge**

STABENAU, Edlef: *zeitgemäße Kataloge*. In: netbib weblog. URL <http://log.netbib.de/archives/2010/06/14/zeitgemasse-kataloge/>. – Aktualisierungsdatum: 2010-09-14 – Überprüfungsdatum 2010-09-19

**STEHLE 2010 - OPAC mit VuFind**

STEHLE, Marcel: *OPAC mit VuFind : Zusammenfassung der Diskussion. Bibcamp<sup>3</sup> Hannover 2010*. URL <http://bibcamp.pbworks.com/OPAC-mit-VuFind> – Aktualisierungsdatum: 2010-05-10 – Überprüfungsdatum 2010-09-20

**STEHLE 2010 – GBS API**

STEHLE, Marcel: *Kurz notiert: Google Book Search API einbinden*. In: bachelopac | Weblog. URL <http://bachelopac.wordpress.com/2010/06/14/kurz-notiert-google-book-search-api-einbinden/> – Überprüfungsdatum 2010-09-20

**STEINER 2007 - OPAC 2.0**

STEINER, Esther Susanne: *OPAC 2.0 : Mit Web 2.0-Technologie zum Bibliothekskatalog der Zukunft?* Bachelorarbeit. Stuttgart, Hochschule der Medien. Bachelorarbeit. 2007. URL [http://opus.bsz-bw.de/hdms/volltexte/2007/624/pdf/steiner\\_bachelorarbeit.pdf](http://opus.bsz-bw.de/hdms/volltexte/2007/624/pdf/steiner_bachelorarbeit.pdf). – Aktualisierungsdatum: 2007 – Überprüfungsdatum 2009-09-19

**STOCK 2007 - Information retrieval**

STOCK, Wolfgang G.: *Information retrieval : Informationen suchen und finden ; [Lehrbuch]*. München : Oldenbourg, 2007 (Einführung in die Informationswissenschaft 1). – ISBN 3486581724

**TUB HAMBURG-HARBURG 2010 - Baustellenliste**

TUB HAMBURG-HARBURG: *TUBFind Baustellenliste*. URL <http://katalog.tub.tu-harburg.de/Help/Home?topic=TODO> – Überprüfungsdatum 2010-09-20

**UBUNTU USERS WIKI 2010 - Cron**

UBUNTU USERS WIKI: *Cron*. URL <http://wiki.ubuntuusers.de/Cron>. – Aktualisierungsdatum: 2010-08-28 – Überprüfungsdatum 2010-09-20

**UBUNTU USERS WIKI 2010 – Server Installation**

UBUNTU USERS WIKI: *Server Installation*. URL [http://wiki.ubuntuusers.de/Server\\_Installation](http://wiki.ubuntuusers.de/Server_Installation). – Aktualisierungsdatum: 2010-02-18 – Überprüfungsdatum 2010-09-20

**UBUNTU USERS WIKI 2010 - Shell**

UBUNTU USERS WIKI: *Shell*. URL <http://wiki.ubuntuusers.de/Shell>. – Aktualisierungsdatum: 2010-08-02 – Überprüfungsdatum 2010-09-20

**UBUNTU USERS WIKI 2010 – Terminal**

UBUNTU USERS WIKI: *Terminal*. URL <http://wiki.ubuntuusers.de/terminal> – Überprüfungsdatum 2010-09-20

**UBUNTU USERS WIKI 2010 - Jabber**

UBUNTUUSERS WIKI: *Jabber*. URL <http://wiki.ubuntuusers.de/Jabber>. – Aktualisierungsdatum: 2010-08-24 – Überprüfungsdatum 2010-09-20

**UNC LIBRARIES 2008 - Pidgin**

UNIVERSITY OF NORTH CAROLINA UNIVERSITY LIBRARIES: *Pidgin Setup for Library IM Services - Meebo Me Widget for Anonymous Users*. URL <http://www.lib.unc.edu/reference/eref/pidgin/meebomewidget.html>. – Aktualisierungsdatum: 2008-04-17 – Überprüfungsdatum 2010-09-20

**Voss 2010 - Verfügbarkeitsinformationen**

Voss, Jakob: *Einheitlichen Kodierung von Verfügbarkeitsinformationen mit DAIA : Verfügbarkeitsnachweise, Authentifizierung und Autorisierung/Möglichkeiten des Web 2.0*. vascoda Workshop am 25.8.2009. URL <http://www.slideshare.net/nichtich/einheitlichen-kodierung-von-verf-gbarkeitsinformationen-mit-daia> – Überprüfungsdatum 2010-09-20

**Voss 2010 - Qualitätssicherung**

Voss, Jakob: *Qualitätssicherung von Freien Daten*. In: Jakobblog. URL <http://jakoblog.de/2010/04/14/qualitatssicherung-von-freien-daten/>. – Aktualisierungsdatum: 2010-04-14 – Überprüfungsdatum 2010-09-20

**VuFIND JIRA #177 2010 – facet behavior**

VuFIND ISSUE MANAGEMENT: [#VUFIND-177] *More flexible facet behavior - VuFind Issue Management*. URL <http://vufind.org/jira/browse/VUFIND-177> – Überprüfungsdatum 2010-09-20

**VuFIND JIRA #303 2010 – Google Book Search**

VuFIND ISSUE MANAGEMENT: [#VUFIND-303] *Google Book Search & Open Library previews*. URL <http://vufind.org/jira/browse/VUFIND-303> – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - adding facets**

STEPHENS, Owen: *VuFind Documentation: adding facets*. URL [http://vufind.org/wiki/adding\\_facets](http://vufind.org/wiki/adding_facets). – Aktualisierungsdatum: 2010-08-20 – Überprüfungsdatum 2010-09-21

**VuFIND DOCUMENTATION 2010. - customization**

KATZ, Demian: *VuFind Documentation: customization*. URL <http://vufind.org/wiki/customization>. – Aktualisierungsdatum: 2010-11-01 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - DAIA/PICA Driver**

MARAHRENS, Oliver: *VuFind Documentation: DAIA/PICA Driver*. URL [http://vufind.org/wiki/daia\\_pica\\_driver](http://vufind.org/wiki/daia_pica_driver). – Aktualisierungsdatum: 2010-08-16 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - developer**

KATZ, Demian: *VuFind Documentation: developer manual*. URL [http://vufind.org/wiki/developer\\_manual](http://vufind.org/wiki/developer_manual). – Aktualisierungsdatum: 2010-03-16 – Überprüfungsdatum 2010-09-21

**VuFIND : DOCUMENTATION 2010 - ILS Driver Support**

SIDDARTHA, E. M.: *VuFind Documentation - ILS Driver Support*. URL [http://vufind.org/wiki/ils\\_support\\_list](http://vufind.org/wiki/ils_support_list). – Aktualisierungsdatum: 2010-08-12 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - installation\_windows**

KATZ, Demian: *VuFind Documentation - installation\_windows* : [http://vufind.org/wiki/installation\\_windows](http://vufind.org/wiki/installation_windows). URL [http://vufind.org/wiki/installation\\_ubuntu?rev=1280929921](http://vufind.org/wiki/installation_ubuntu?rev=1280929921). – Aktualisierungsdatum: 2010-09-10 – Überprüfungsdatum 2010-09-19

**VuFIND : DOCUMENTATION 2010 – Installations**

BROWN RIGG, Steward J.: *VuFind Installations*. URL [http://vufind.org/wiki/installation\\_status](http://vufind.org/wiki/installation_status). – Aktualisierungsdatum: 2010-09-17 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - Importing Records**

KATZ, Demian: *MARC Records*. URL [http://vufind.org/wiki/importing\\_records](http://vufind.org/wiki/importing_records). – Aktualisierungsdatum: 2010-09-10 – Überprüfungsdatum 2010-09-19

**VuFIND DOCUMENTATION 2010- other than marc**

KATZ, Demian: *VuFind Documentation: other than marc*. URL [http://vufind.org/wiki/other\\_than\\_marc](http://vufind.org/wiki/other_than_marc). – Aktualisierungsdatum: 2010-07-15 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - performance**

KATZ, Demian: *VuFind Documentation: performance*. URL <http://vufind.org/wiki/performance>. – Aktualisierungsdatum: 2009-11-18 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 – re-indexing**

KINSTLER, Till: *VuFind Documentation: re-indexing*. URL <http://vufind.org/wiki/re-indexing>. – Aktualisierungsdatum: 2010-01-19 – Überprüfungsdatum 2010-09-21

**VuFIND DOCUMENTATION 2010 – Ubuntu**

KATZ, Demian: *VuFind 1.0 on Ubuntu*. URL [http://vufind.org/wiki/installation\\_ubuntu?rev=1280929921](http://vufind.org/wiki/installation_ubuntu?rev=1280929921). – Aktualisierungsdatum: 2010-09-10 – Überprüfungsdatum 2010-09-19

**VuFIND DOCUMENTATION 2010 - subject fields**

N.N.: *VuFind Documentation: subject fields*. URL [http://vufind.org/wiki/subject\\_fields?s=stopwords#topic\\_fields](http://vufind.org/wiki/subject_fields?s=stopwords#topic_fields). – Aktualisierungsdatum: 2008-08-13 – Überprüfungsdatum 2010-09-21

**VuFIND DOCUMENTATION 2010 - security**

KATZ, Demian: *VuFind Documentation: security*. URL <http://vufind.org/wiki/security>. – Aktualisierungsdatum: 2010-05-20 – Überprüfungsdatum 2010-09-20

**VuFIND DOCUMENTATION 2010 - starting and stopping**

KATZ, Demian: *VuFind Documentation: starting and stopping*. URL [http://vufind.org/wiki/starting\\_and\\_stopping\\_vufind](http://vufind.org/wiki/starting_and_stopping_vufind). – Aktualisierungsdatum: 2009-12-22 – Überprüfungsdatum 2010-09-20

**WIENOLD 2004 - Machbarkeitsstudien**

WIENOLD, Kirsten: *Machbarkeitsstudien von ELearning gestützten Bildungsangeboten*. URL [http://www.oleimeulen.info/pdf/artikel\\_2.pdf](http://www.oleimeulen.info/pdf/artikel_2.pdf). – Aktualisierungsdatum: 2004-03-16 – Überprüfungsdatum 2010-09-20

**WIESENMÜLLER 2008 - Neue Trends**

WIESENMÜLLER, Heidrun: *Neue Trends und alte Desiderate: Der OPAC der Zukunft (Teil 2)*. In: *VDB-Mitteilungen* (2008), Nr. 1, S. 29. URL <http://www.vdb-online.org/publikationen/vdb-mitteilungen/vdb-mitteilungen-2008-1.pdf> – Überprüfungsdatum 2010-09-18

**WIESENMÜLLER 2008 - Der OPAC der Zukunft**

WIESENMÜLLER, Heidrun: *Der OPAC der Zukunft : Trends und Desiderate*. Arbeitsgemeinschaft für juristisches Bibliotheks- und Dokumentationswesen (AjBD) Fortbildung 10.11.2008. URL <http://www.ajbd.de/veranst/2008Muenchen/2008Muenchen-Wiesenmueller.pdf>. – Aktualisierungsdatum: 2009-05-17 – Überprüfungsdatum 2010-09-19

**WIESENMÜLLER 2010 - Daten härter**

WIESENMÜLLER, Heidrun: *Daten härter arbeiten lassen und besser präsentieren : Sacherschließung und Normdaten in Online-Katalogen*. In: *BuB Forum Bibliothek und Information* 62 (2010), Nr. 1, S. 48–54. URL [http://www.b-u-b.de/cgi-local/byteserver.pl/pdfarchiv/Heft-BuB\\_01\\_2010.pdf](http://www.b-u-b.de/cgi-local/byteserver.pl/pdfarchiv/Heft-BuB_01_2010.pdf) – Überprüfungsdatum 2010-09-19

**WUNDER, GROSCHKE 2009 - Verteilte Führungsinformationssysteme**

WUNDER, Michael ; GROSCHKE, Jürgen: *Verteilte Führungsinformationssysteme*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2009. – ISBN 364200508X

**ZIMMER 2008 - Ubuntu 8.04**

ZIMMER, Benjamin: *Ubuntu 8.04 : office-server guide*. URL <http://files.zeroathome.de/ubuntu804-server-guide0.7beta-www.zeroathome.de.pdf>. – Aktualisierungsdatum: 2008-12-15 – Überprüfungsdatum 2010-09-20

## **Anhang**

Die beiliegende CD-ROM enthält

- Eine elektronische Ausfertigung dieser Arbeit
- Sämtliche VuFind-Dateien des bachelopacs im Ordner vufind (ohne Index)
- Den verwendeten DAIA-Server phpDAIA mit angepasster Konfiguration für den PICA-OPAC LBS4 der HAW Hamburg im Ordner phpDAIA
- Angepasste Skripte z3950fetch.php und CBS-Cron.sh der TUB HH sowie exemplarisch drei Metadatenpakete im Ordner Datenabgleich



### **Eidesstattliche Erklärung**

Wir versichern, die entsprechend gekennzeichneten Teilen der vorliegenden Arbeit selbständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangabe kenntlich gemacht.

Hamburg, den

J. Schmitt

M. Stehle