Externe Indizierung von OPAC-Inhalten

External Indexing of OPAC-contents
Indexation externe des données du catalogue électronique public en ligne (OPAC)

Harald Jele

Mit dieser Arbeit wird gezeigt, dass durch eine externe Indizierung von bibliographischen Daten außerhalb der üblichen OPAC-Systeme sowohl Vorteile im Retrieval als auch deutliche Kostenersparnisse im Bereich der typischen Lizenzkosten erwartbar sind.
Die Rahmenbedingungen – unter denen dieser Ansatz verfolgt wird – sind:
• ein zeitgemäßer Web-OPAC, mit dem die gängigen Funktionen eines solchen angeboten werden
• ein Bibliotheks(verwaltungs)system, in dem bibliographische Daten in standardisierter Form (hier: `MAB2`) angeboten werden
• die externe Indizierung von ca. `370.000` bibliographischen Datensätze durch ein Open-Source-Produkt (hier: `Swish-e` sowie alternativ `Lucene`).

In this paper it is shown that by an external indexing of bibliographic data outside of an typical OPAC-system advantages in the retrieval as well as remarkable cost savings in the field of licenses can be expected.
The set-up for this approach is:
• a modern Web-OPAC which offers the usual functions
• a library system, in which bibliographic data are stored in standardized form (in our case: `MAB2`)
• the external indexing of about `370.000` bibliographic data records by an Open-Source-Software (in our case: both `Swish-e` and `Lucene`).

Ce travail a pour objectif de présenter les avantages résultants d'une indexation externe de données bibliographiques en dehors des systèmes conventionnels de catalogues électroniques publics en ligne (OPAC), aussi bien concernant une meilleure récupération de l'information que du point de vue d'une économisation des frais de concessions.
Les données préalables pour la réalisation de ce projet sont les suivantes:
• un catalogue électronique Web-OPAC actuel, qui offre toutes les fonctions normalement requises par un tel système
• un système de gestion de bibliothèque capable de fournir les données bibliographiques sous forme standardisée (ici: `MAB2`)
• une indexation externe d'environ `370.000` données bibliographiques par un produit logiciel Open-Source (ici: `Swish-e` et ou `Lucene`).

1 Einleitung

Die praktischen Erfahrungen im Einsatz von OPAC-Systemen zeigen, dass sich diese für eine Vielzahl von Anwendungsfällen bestens eignen. Andererseits offenbaren viele Systeme in Hinblick auf die übliche Benutzung eines öffentlich zugänglichen Online-Kataloges leider auch „überraschende“ und mitunter unüberwindbar scheinende Schwächen.

Um diesen Schwächen adäquat zu begegnen, bietet sich an, OPAC-Daten außerhalb des zugehörigen Bibliothekssystems zu indizieren und mit den Ergebnissen einer entsprechenden externen Suche auf das Bibliothekssystem (im einfachsten Fall durch einen Web-Link) zurückzuverweisen.[1]

[1] auf den Umstand, dass der Einsatz externer Indizierungsverfahren bibliographischer Daten ein durchaus gängiger werden kann, weisen exemplarisch jene Ansätze hin, die in Kostädt (2003) und

Neben den sich daraus günstiger Weise ergebenden Vorteilen ist vielfach der finanzielle Aspekt eines solchen Ansatzes nicht zu vernachlässigen:
Die Lizenzkosten der verbreiteten OPAC-Systeme richten sich wesentlich nach der Anzahl der zugelassenen, gleichzeitigen Systemabfragen. Mit der externen Indizierung kann die Anzahl der gleichzeitigen Systemabfragen, die letztendlich im Bibliothekssystem durchgeführt werden, drastisch reduziert werden. Wenn bei der Realisierung dieses Vorhabens zudem auf ein Open-Source-Produkt zurückgegriffen wird, kann bei gleichzeitigem Wegfall von Unwägbarkeiten die jährliche Kostenersparnis deutlich ausfallen.

2 Relevante Stärken und Schwächen von OPAC-Systemen

Zu den **Stärken** zählen zweifellos all jene Anwendungen, mit denen ein/e Benutzer/in aufgefundene Werkdaten in irgendeiner Form nutzt und einer individuellen oder gar personalisierten Bearbeitung zuführt. Das meint typischerweise das Vormerken, Bestellen oder Reservieren von Werken, das Verlängern von bestehenden Ausleihen, die Durchführung von Bestellungen in anderen Bibliotheken über Fernleihfunktionen etc. Das Gemeinsame an diesen Anwendungen ist ihr administrativer Charakter.
In OPAC-Systemen werden diese Funktionen zumeist durch eine (geschickte) logische Verknüpfung, Prüfung und Speicherung von benutzer/innen-, werk- und bibliotheksabhängigen Daten erreicht. In den allermeisten Fällen werden diese (zumindest auszugsweise) relational vorgehalten und gespeichert – und entsprechen somit den üblichen Anforderungen typischer, leistungsfähiger Datenbanksysteme. Diese Entsprechung führt im günstigen Fall auch dazu, dass diese Anwendungen performant umsetzbar und für den/die Benutzer/in ebenso zügig vonstatten gehen.

Die wesentlichen **Schwächen** vieler OPAC-Systeme liegen dagegen zumeist im Retrieval. Dies erscheint natürlich aus dem Grund als eine „überraschende“ Schwäche, da das Retrieval im üblichen Fall ja als *die* Kernaufgabe eines OPAC-Systems angesehen wird. Dabei liegen die gewöhnlich vorhandenen Schwächen vielfach nicht in einer fehlenden oder gar zu wenig ausgefeilten Suchsprache, in fehlenden – und für eine bibliothekarisch oder bibliographisch arbeitende Einrichtung individuell nicht passenden – Suchformularen oder in den Retrieve- und Browse-Möglichkeiten, sondern überwiegend in der Schwierigkeit, sehr performante Suchanfragen durchzuführen.

Monat	Hits	Search & Browse Requests	Anteil in %
Sep	530874	461860	87
Oct	1120440	862739	77
Nov	897472	735927	82
Dec	668082	627997	94
Jan	742245	668021	90
Feb	571954	486161	85
Mrz	1253609	1128248	90
Apr	844592	751687	89
May	768604	607197	79
Jun	693931	624538	90
Jul	578136	491416	85
Aug	455300	355134	78
		Durchschnitt	85,5

Abbildung 1: Wertetabelle der Zugriffsstatistik: ausgewählte Daten für den Zeitraum `Sept. 2003` – `Aug. 2004` des lokalen Web-OPACs der UB Klagenfurt

Die Ursachen dafür zeigen im Detail wohl sehr verschiedene Gründe und sind im Wesentlichen vom spezifischen Systemdesign der jeweiligen Hersteller abhängig.

Die Auswirkungen wenig performanter Retrieval-Systeme sind jedoch vielfach sehr ähnlich und äußern sich letztlich auch in der Unzufriedenheit und im Unverständnis der Benutzer/innen, die gewohnt sind, aus erheblich umfangreicheren Beständen in wesentlich kürzerer Zeit brauchbare Ergebnisse zu erzielen. Prominente Beispiele, deren häufige Benutzung zu einer solchen Wahrnehmung verführen sind natürlich vorwiegend aus den Erfahrungen im Umgang mit den beliebten und sehr leistungsfähigen Suchmaschinen des World Wide Web zu finden.

Ein weiterer Umstand, der verdeutlicht, wie wichtig die Performanz des Retrievals eines OPAC-Systems genommen werden muss, liegt darin verborgen, dass OPAC-Systeme wesentlich häufiger für Suchanfragen als für administrative Tätigkeiten von den Benutzer/n/innen verwendet werden.
Das heißt gleichzeitig aber auch, dass gerade jene Funktionen, die typischerweise weniger gut umgesetzt sind, wesentlich häufiger benutzt werden und auch aus diesem Grund deutlicher im Mittelpunkt der Wahrnehmung stehen.

Diese Aussagen lassen sich durch eine Auswertung der entsprechenden System-Log-Files untermauern (siehe *Abb. 1* und *2*):
Im Jahresdurchschnitt sind ausschließlich `14.5%` aller Systemanfragen durch *administrative* Tätigkeiten hervorgerufen worden. Die (relativ große) messbare Spanne

`http://www.contentmanager.de/magazin/news_h5612-print_fast_und_universitaet_bielefeld_foerdern.html` für den deutschsprachigen Raum sowie in `http://www.at-web.de/newsletter/archiv/news107.htm` für den nordamerikanischen publiziert/festgehalten wurden

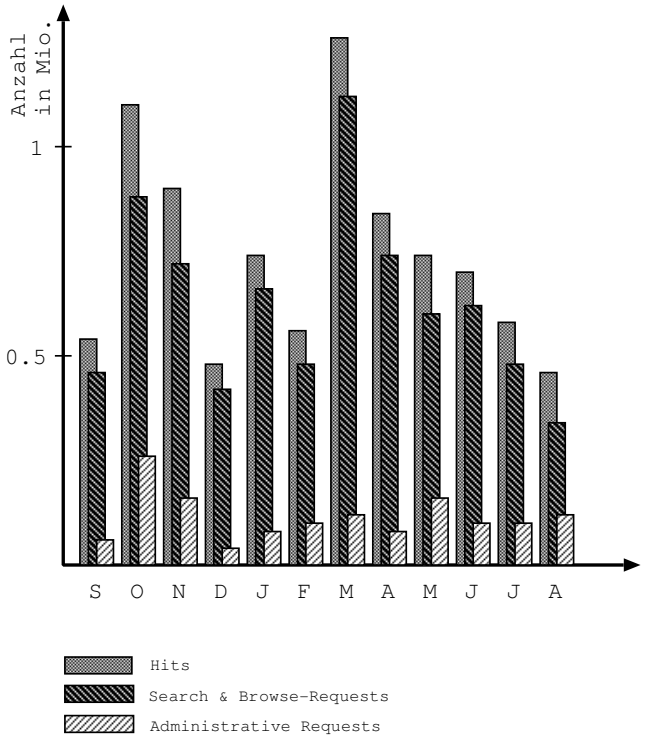

Abbildung 2: Graphische Darstellung der Zugriffsstatistik: ausgewählte Daten für den Zeitraum `Sept. 2003` – `Aug. 2004` des lokalen Web-OPACs der UB Klagenfurt

dieses Wertes reicht dabei von 22% im `Aug. 2004` bis 6% im `Dez. 2003`.

Bei der Interpretation dieser Zahlen bleibt weiters zu bedenken, dass das lokale Bibliothekssystem der UB Klagenfurt Teil des Österreichischen Verbundsystems ist – und die praktische, für eine Vielzahl von Benutzer/n/innen relevante Recherche immer eine vom Verbundsystem ausgehende ist. Unter Berücksichtigung dieses Umstands ist eigentlich anzunehmen, dass in anderen Lokalsystemen – die nicht Teil eines integrierten Verbundes sind – das Verhältnis zwischen administrativen Anfragen und Rechercheanfragen noch wesentlich deutlicher zugunsten der Anzahl der Rechercheanfragen verschoben ist.[2]

Performanzverbesserungen des Retrievals lassen sich in vielen Fällen durch Optimierung des zugrunde liegenden Index-Systems erreichen.
Hersteller von OPAC-Systemen legen bei der Parametrisierung der bibliographischen Indizes meist Wert auf eine größtmögliche Kompatibilität mit dem Durchschnitt aller (möglichen) Suchanfragen und stellen somit sicher, dass „typischerweise“ Gesuchtes auch gefunden werden kann.
Eine Reduzierung der möglichen Indexeinträge auf die unbedingt notwendigen sowie eine sinnvolle Gliederung in mehrere Indexregister[3] sind vielfach einfache aber lohnenswerte und wirkungsvolle Ansätze dazu.

Darüber hinaus wird zur Verbesserung der Performanz vielfach die Anschaffung neuerer Hardware empfohlen, bei deren Auswahl eine deutliche Leistungssteigerung im Mittelpunkt stehen sollte.
Dass diesem Umstand nicht immer Rechnung getragen werden kann, ist allein durch die Kalkulation unter ständig sinkenden Budgets gegeben. Der Gesichtspunkt der Neuanschaffung ist für viele Einrichtungen nur unter Berücksichtigung einer fünfjährigen, steuerlichen Anlagenabschreibung möglich.

Auch unter dem Gesichtspunkt optimierter Index-Systeme und/oder erneuerter Hardware zeigen OPAC-Systeme vielfach Schwächen in der Leistungsfähigkeit des Retrievals.
Als Beispiel dafür ist hier eine typische Recherche in solchen Systeme angeführt. Gesucht wird dabei über den jeweils günstigsten Indexeintrag (unter Berücksichtigung der entsprechenden Möglichkeiten einer Expertensuche) jene Menge an Diplomarbeiten zu einem bestimmten Fachgebiet aus einem bestimmten Erscheinungsjahr.

Diese Fragestellung wird im vorliegenden System[4] durch eine Schnittmengenbildung von vier Teilmengen realisiert. Ausschließlich einer der zur Abfrage notwendigen Einträge wird trunkiert gesucht[5]:

- Teilmenge 1:
  alle Einträge, die den Indexeintrag zur lokalen Notation für „Klagenfurter Hochschulschriften“ (=`40A-`) tragen. Die sich ergebende Teilmenge entspricht `5612` Titeln
- Teilmenge 2:
  alle Einträge, in denen das Titelwort „`Dipl.-Arb.`“ vorkommt (der Abkürzungspunkt am Ende ist im aktuellen Index nicht vorhanden und wird aus diesem Grund im u.a. Suchstring nicht mitgeführt). Die sich ergebende Teilmenge entspricht `5456` Titeln
- Teilmenge 3:
  alle Einträge, die dem Fachgebiet „Wirtschaftswissenschaften“ (=`10-?`) zugeordnet sind. Die sich ergebende Teilmenge entspricht `23383` Titeln.
  Alternativ wurde das Ergebnis mit einer zweiten, ähnlichen Suchanfrage (zur einfachen Überprüfung)

[2] . . . und die Performanz der Recherche damit auch deutlicher wahrgenommen wird

[3] das sind in relationalen Datenbanksystemen wohl in den meisten Fällen „Tabellen“

[4] die hier angeführten Werte beziehen sich auf Suchanfragen, die am `1. Februar 2005` um `21 Uhr` über den Web-OPAC des Systems `ALEPH500` (Version `14.02.06`) bei einer Systemlast („Load“) von `0.4` über den Link
`http://opac.uni-klu.ac.at/`
durchgeführt wurden. Alle weiteren ermittelten Systemkennzahlen deuten auf ein unbelastetes System hin

[5] das im Suchstring erkennbare Fragezeichen („`?`“) dient dabei zur Trunkierung

produziert. Dabei wurden alle Einträge zum Fachgebiet „Geschichte“ (=16-?) gesucht. Als Ergebnis haben sich 22972 Treffer qualifiziert

- Teilmenge 4:
  alle Einträge, deren Titel die gesuchte Jahreszahl (=2000) beinhalten. Die sich ergebende Teilmenge entspricht 11141 Titeln

Die Schnittmengenbildung wurde im System über folgende Suchstrings realisiert:

```
WNO=40A- AND WTI=Dipl.-Arb AND WNO=10-? AND
WJA=2000
```

bzw.

```
WNO=40A- AND WTI=Dipl.-Arb AND WNO=16-? AND
WJA=2000
```

Dabei ist zu beachten, dass vor den eigentlichen Datenbankabfragen kein Prozess gestartet wird, der die für die jeweilige Suchanfrage günstigste Teilmengenbildung ermittelt und anschließend das Retrieval anhand dieser Erstanalyse in optimierter Form steuert.
Durch Auswertung der zugänglichen Datenbank- und Webserver-Logfiles konnte ermittelt werden, dass die Teilmengenbildung vom System in dieser Reihenfolge durchgeführt wurde:

```
(((WNO=40A- AND WTI=Dipl.-Arb) AND WNO=10-?)
   AND WJA=2000)
```

bzw.

```
(((WNO=40A- AND WTI=Dipl.-Arb) AND WNO=16-?)
   AND WJA=2000)
```

Die Ergebnisse bei notwendiger Trunkierung sind, dass das System für die Anfrage zu den Treffern aus den Wirtschaftswissenschaften (=10-?) 30s und für jene zur Geschichte (=16-?) 94s benötigt. Bei komplexeren Abfragen, die die Verarbeitung einer größeren Anzahl an Teilmengen bedingen oder die im Index eine große Anzahl an vorhandenen Zeichen-Permutationen ansprechen, gerät das System sehr leicht ins sog. Timeout.[6]

Möglicherweise kann eine Verbesserung des schlechten Antwortverhaltens für diese konkrete Abfrage durch Veränderungen im bibliographischen Index herbeigeführt werden, mit denen eine Trunkierung evtl. vermieden werden kann. Trunkierungen sind im Informationsretrieval generell jedoch häufig notwendig und in vielen weiteren Situationen oftmals unvermeidbar.

Zudem bleibt zu bedenken, dass bei üblichen Suchanfragen neben der Trunkierung häufig auch der Einsatz von Bereichsoperatoren[7] Verwendung findet.

[6] beim hier angesprochenen „Timeout“ sind die entsprechenden Systemparameter so eingestellt, dass Suchprozesse, die innerhalb von 120s kein Ergebnis bringen, abgebrochen werden

[7] mit denen z.B. die Ergebnisse aus einem Bereich von Jahreszahlen (wie sämtliche Titel, deren Erscheinungsjahr in den Bereich von „1999 bis 2004“ fallen) erschlossen werden

Monat	Search & Browse Requests	Anzahl Titel-Vollanz	Anteil in %
Sep	461860	124702	27
Okt	862739	284704	33
Nov	735927	228137	31
Dez	627997	263759	42
Jan	668021	167005	25
Feb	486161	116679	24
Mrz	1128248	338474	30
Apr	751687	187922	25
Mai	607197	206447	34
Jun	624538	168625	27
Jul	491416	162167	33
Aug	355134	120746	34
		Durchschnitt	30,4

Abbildung 3: Wertetabelle: Die Anzahl der Such- und Browse-Anfragen im Verhältnis zu den Titel-Vollanzeigen: ausgewählte Daten für den Zeitraum Sep. 2003 – Aug. 2004 des lokalen Web-OPACs der UB Klagenfurt

Bei der konkreten Realisierung von Funktionen, die sich aus Bereichsoperatoren ergeben, ist die spezifische Methode zur Bildung von Gesamtmengen (aus mehreren, sich durch eine Suche qualifizierenden Teilmengen) von bedeutender Relevanz. Eine ungünstige Vorgehensweise in der Gesamtmengenbildung hat dabei einen deutlichen Einfluss auf die sich ergebende, messbare Systemperformanz bzw. zudem auf mathematische Fehler, die sich daraus ergeben können.[8]

Neben der Verbesserung der Systemperformanz verspricht der hier beschriebene Ansatz zur externen Indizierung bibliographischer Daten auch eine deutliche Reduzierung der Abfragen im Bibliothekssystem insgesamt.
Diese werden (wie beschrieben) durch eine Einschränkung einerseits auf die administrativen Anfragen, andererseits durch eine Einschränkung auf die Titel-Vollanzeigen bzw. die Exemplaranzeigen erreicht. Die sich daraus ergebende Situation führt gleichzeitig nur dann zu einer deutlichen Reduktion der bereitzustellenden OPAC-Lizenzen[9], wenn nicht jede Suchanfrage zu

[8] als ein häufiger Fehler in der Gesamtmengenbildung kann beispielhaft jener angeführt werden, der zu einer Nullmenge führt, wenn eine der Teilmengen innerhalb einer Bereichssuche zu einer Nullmenge geführt hat.
Konkret meint dies, dass Titel aus den Erscheinungsjahren 1885-1889 durch eine Bereichssuche über diese Jahre gesucht werden und die Suche keine Treffer ergibt, obwohl sich in den Teilmengen aus vier der betroffenen fünf Jahre Treffer qualifizierten

[9] mit dem Begriff der „OPAC-Lizenz“ ist in diesem Fall die

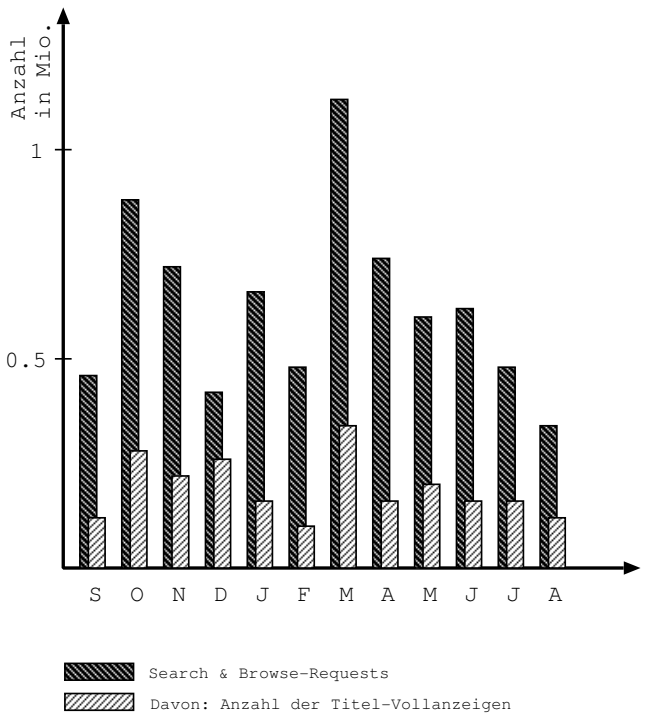

Abbildung 4: Die Anzahl der Such- und Browse-Anfragen im Verhältnis zu den Titel-Vollanzeigen: ausgewählte Daten für den Zeitraum `Sep. 2003` – `Aug. 2004` des lokalen Web-OPACs der UB Klagenfurt

mindestens einer Titelvollanzeige führt.
Bei jeder Vollanzeige von Titel- und Exemplardaten wird ja weiterhin auf den Datenbestand des Bibliothekssystems zurückgegriffen, um die Datenspeicherung möglichst wenig redundant (und damit weniger fehleranfällig) zu halten. Dies bedingt in letzter Konsequenz jedoch auch, dass für diese Anzeigen OPAC-Lizenzen notwendig sind.
Gelingt beim Datenexport bzw. im Zuge der Aufbereitung der exportierten Titeldaten ein günstiges (das ist ein für die Benutzer/innen praktikables) Anzeigeformat, ist mitunter eine Titelvollanzeige, die vom Bibliothekssystem generiert wird, zudem eher selten notwendig. Allein die Exemplaranzeige muss in diesem Fall vom Bibliothekssystem geleistet werden.

In *Abb. 4* ist das Verhältnis von Such- und Browseanfragen zu der Anzahl der daraus resultierenden Vollanzeigen von Titel- bzw. Exemplardaten dargestellt.
Daraus erkennt man, dass nur ca. ein Drittel aller Suchanfragen auch zu einer oder mehreren Vollanzeigen führt.

Somit kann angenommen werden, dass neben der Reduzierung der direkten OPAC-Anfragen auf die oben beschriebenen, rein administrativen Tätigkeiten sich auch die Systemabfragen – und damit einhergehend das nötige Lizenzaufkommen – durch eine Einschränkung bei notwendigen Titelanzeigen aus dem Bibliothekssystem insgesamt deutlich reduzieren müssen.

3 Ein möglicher Lösungsansatz

Sind sämtliche systemspezifische Optimierungsstrategien ausgeschöpft und konnten gleichzeitig die erhofften Verbesserungen der Performanz nicht erreicht werden, kann mitunter die Verwendung eines externen Systems zur Indizierung der bibliographischen Daten und zu deren weiterem Retrieval herangezogen werden.

Dabei werden die in der systemeigenen Datenbank gespeicherten bibliographischen Daten extrahiert, aufbereitet und extern gespeichert. Die so gewonnene Datenmenge wird anschließend durch ein Retrieval-System indiziert und dieser Index den Benutzer/n/innen mit einer üblichen Suchmaske zugänglich gemacht.
Die darüber erzielten und in weiterer Folge angezeigten Treffer verweisen mit ihren Einträgen (z.B. durch Web-Links) wiederum direkt ins OPAC-System, sodass alle administrativen Tätigkeiten dort vorgenommen werden können. Im besten Fall bemerken die Benutzer/innen dabei nicht, dass sie mit zwei oder mehreren Systemen arbeitsteilig kommunizieren (vgl. *Abb. 5*).

Zwei Lösungsansätze wurden dabei näher betrachtet:

1. das Generieren von externen `XML`-Dateien, die anschließend durch ein Open-Source-Produkt indiziert wurden. Dieser Ansatz wurde vor allem aus Gründen schlechterer Performanz und wegen des erheblich größeren Aufwands in der konkreten Realisierung nicht weiterverfolgt (siehe dazu auch die Ausführungen im Text in *Abschnitt 3.1* sowie in *Fußnote 21*)
2. der Export der bibliographischen Daten direkt in `xHTML`-Dateien, deren Strukturierung durch die Beifügung der für das weitere Retrieval relevanten Dublin-Core-Kategorien innerhalb der sog. Meta-Tags erhalten bleibt. Dieser Lösungsansatz wird hier in weiterer Folge diskutiert und ausgeführt.

3.1 Datenextraktion und Datenaufbereitung

Bei der Umsetzung des gewählten Lösungsansatzes werden die bibliographischen Daten online extrahiert.[10] Al-

Anzahl der gleichzeitig auf den OPAC zugreifenden Benutzer/innen gemeint

[10] der Unterschied, der hier einerseits zwischen dem „Export“ und andererseits der „Extraktion“ gesehen wird, ist folgender:
Der Export erfolgt im ersten Schritt. Dabei werden die bibliographischen Einträge in der vorhandenen Form aus der Datenbank übernommen.
Bei der Extraktion hingegen werden den – durch den Export gewonnenen – Einträgen (nach formaler Prüfung) spezifische

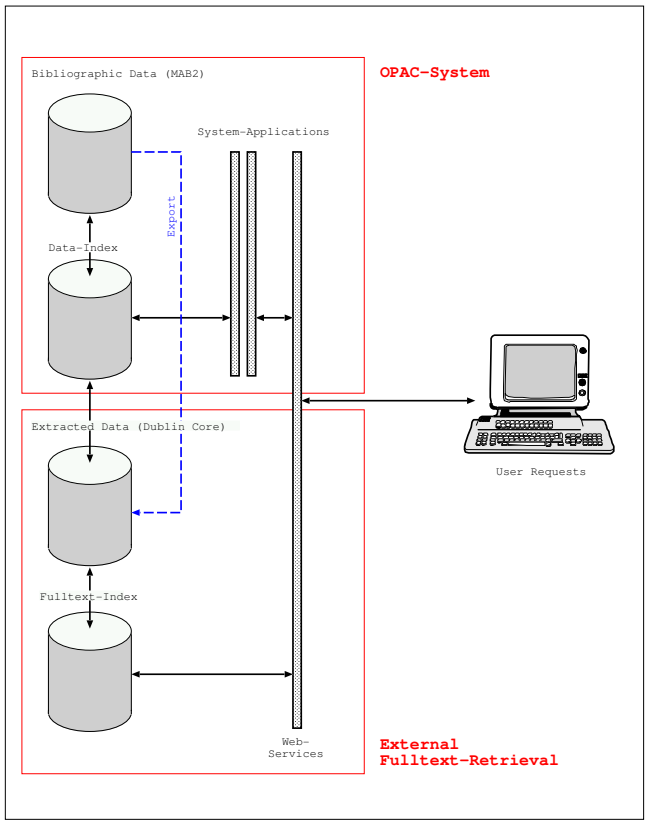

Abbildung 5: Schematische Darstellung des Datenzugriffs

ternativ können diese offline durch einen Export (der betreffenden Tabellen) in eine Datei und anschließender Extraktion aus dieser gewonnen werden.

Die sich aus der Online-Gewinnung der Daten ergebenden Vorteile liegen dabei auf der Hand:

- durch die zeitnahe Ermittlung der bibliographischen Einträge sind die gewonnenen und im Weiteren extern gespeicherten Daten fast deckungsgleich zu den Einträgen im Bibliothekssystem (d.h., dass die externen Daten sehr aktuell sind)
- Änderungen, die in den Ausgangsdaten zu einem späteren Zeitpunkt erfolgen, können durch relativ einfache Mechanismen auch nach der Extraktion geprüft und übernommen werden.

Beim Export werden die bibliographischen Daten in einem ersten Schritt in der vorliegenden Form der Datenbank unverändert entnommen (siehe auch *Fußnote 10*). Als Export-Schnittstellen dienen dazu primär natürlich jene, die das Datenbanksystem nativ bereitstellt.[11] Beim sog. „Ur-Lauf“[12] werden zusätzlich zu den bibliographischen Daten auch die entsprechenden Datums- und Zeitstempel des Systems pro Datensatz mit übernommen. Mit diesen kann in weiterer Folge geprüft werden, ob Änderungen in den bibliographischen Ausgangsdaten (gegenüber den exportierten) erfolgt sind.[13]

Bevor die durch den Export gewonnenen Daten jedoch extern gespeichert werden, durchlaufen diese eine Folge von Routinen, die hier zu einer funktionellen Einheit zusammengefasst als „Extraktion“ bezeichnet werden. Dabei werden aufgrund von Zuordnungstabellen die Inhalte einer Reihe von definierten Kategorien im Ausgangsformat `MAB2` ermittelt und in jenem Kategorienschema (Dublin-Core) hinterlegt, das zur externen Speicherung verwendet wird. Der Extraktionsprozess erfolgt durch ein sequenzielles Lesen sämtlicher, vorhandener Kategorieneinträge.[14] Bei der Zuordnung muss beachtet werden, dass die Inhalte mehrerer, mitunter auch verschiedener `MAB2`-Kategorien einer einzelnen Kategorie des Dublin-Core-Formates[15] zugewiesen werden können (vgl. dazu auch die Tabelle in *Abb. 6*).
Die Speicherung der so gewonnenen Daten erfolgt anschließend relational in Tabellen.

Der darauf folgende Schritt ist jener der „Datenaufbereitung“. Dieser ist technisch durch mehrere, unabhängig voneinander arbeitende Programme realisiert.
Eine überaus wichtige Funktion dabei ist die sog. „Anreicherung“ der Titeldaten mit weiteren bibliographischen Einträgen. Diese werden überwiegend aus an-

Inhalte entnommen und entsprechend den Erfordernissen des externen Index-Systems zur weiteren Verarbeitung aufbereitet

[11] im vorliegenden Fall wurden nicht die nativen Datenbankschnittstellen wie z.B. jene über ODBC (=Online Database Connector) oder `SQL-Net` (=Oracle `SQL`-Netzwerkprotokoll) verwendet, sondern – überwiegend aus Gründen einer effizienten Programmierung – die `Perl`-Datenbank-Schnittstellen `DBD/DBI`

[12] das ist der erstmalige, vollständige Datenexport, der in weiterer Folge durch Datenergänzungen und -korrekturen am laufenden Stand gehalten wird

[13] die optionale Verwendung bzw. der Einsatz von Datenbank-„Triggern“ ist für diesen Fall – sofern das verwendete Datenbankmanagementsystem solche unterstützt – natürlich vorzuziehen

[14] inklusive der vorhandenen Indikatoren und Teilfelder

[15] zur standardkonformen Verwendung der Kategorien vgl. vor allem
`http://dublincore.org/` sowie innerhalb der Meta-Tags von `HTML`
`http://selfhtml.org/`

Das in *Abb. 8* angeführte Beispiel beinhaltet in den Meta-Tags folgende Kategorieneinträge:

```
<meta name="DC.Creator" content="Jele, Harald">
<meta name="DC.Title" content="Wissenschaftliches
  Arbeiten: Zitieren">
<meta name="DC.Publisher" content="
  Oldenbourg, München ; Wien">
<meta name="DC.Date" content="2003">
<meta name="DC.Description"
  content="3-486-27506-2">
<meta name="DC.Subject" content="Wissenschaftliches
  Arbeiten, Zitat, Richtlinie, Veröffentlichung,
  Zitat, Richtlinie, Bibliographieren, Zitat,
  Richtlinie">
<meta name="DC.Identifier"
  content="http://opac.uni-klu.ac.at/ALEPH/-/ext-
  find?base=ubk01&find=WID=AC03842486">
<meta name="DC.Source" content="http://ubdocs.uni-
  klu.ac.at/open/texte/AC03842486.pdf">
```

```
Dublin-Core     : MAB2 Kategorie
-Kategorie      : (KAT+Indikator+Teilf.)
------------------------------------------------
                :
DC.Creator      : 100_ a Personennamen
                : 100b a
                : 104a a
                : 108a a
                : 200_ a Körperschaftsnamen
                : 200b a
                : 204b a
                : 208b a
DC.Title        : 331_ a Titelangaben
                : 331a a
                : 451_ a (1. Gesamttitel in Vorlageform)
DC.Publisher    : 412_ a Verlage
                : 410_ a Orte
DC.Date         : 425a a Erscheinungsjahr
                : 425_ a
DC.Description  : 089_ a Bandangaben
                : 540a a ISBN
                : 542a a ISSN
                : 433_ a Umfangsangabe
                : 512_ a Kollationsvermerke
                : 001_ a ID-Nummer
DC.Subject      : 902_ s Schlagwörter
                : 902_ f
                : 905_ s
                : 905_ f
                : 912_ s
                : 912_ f
```

Abbildung 6: Zuordnungsschema von exportierten MAB2-Kategorien zu den entsprechenden Dublin-Core-Einträgen

deren (und möglicherweise inhaltlich abhängigen) Datensätzen ermittelt und in den letztlich zu indizierenden Datensatz mit aufgenommen. Darunter fallen vor allem Einträge, die durch die hierarchischen Ordnungen der Katalogisierung nach RAK-WB entstehen.[16]
Entsprechend den Link-Informationen zu den übergeordneten Datensätzen eines jeweiligen Titels werden die Informationen innerhalb eines einzelnen Titel-Datensatzes (=eines Katalogisats) zusammengeführt. Dies ist allein schon deshalb notwendig, um nach der anschließenden Indexierung Titel wieder auffindbar zu machen, die aufgrund der vorhandenen Datenkonsistenz[17] in Folge nur unzureichend zugänglich wären.
Im vorliegenden Fall wurde entschieden, Informationen der übergeordneten Datensätze in den jeweiligen Titel zu integrieren. Das heißt jene Angaben, die für ein Reihenwerk im hierarchisch übergeordneten Datensatz nur einmal vorhanden sind, redundant in jedem betroffenen untergeordneten Datensatz mit aufzunehmen.[18]

[16] das sind für das vorliegende Problem im Wesentlichen die Einträge zu den Personennamen und Titelinformationen innerhalb von Katalogisaten zu Werken mit Reihen- und Bandangaben

[17] siehe dazu z.B. das sog. „Schiller-Räuber-Problem“ (vgl. Eversberg (1997) http://www.biblio.tu-bs.de/allegro/news/acn972.htm sowie Jele (2001, S.58))

[18] der dazu gegensätzliche Ansatz (Informationen aus den sog. Stücktitel-Datensätzen im Reihentitel-Datensatz zusammenzuführen) ergibt sehr umfangreiche und vor allem äußerst unübersichtliche Datensätze

```
FMT   L MH
LDR   L 00393pM2.01200024------h
001   L $$aAC03842486
030   L a|1dcz|z|||17
036a  L $$aDE
037b  L $$ager
050   L a|a||||||||||
051   L mz||w||
070   L $$aVBK
070a  L $$a292
070b  L $$aUBK
076s  L $$a01
100   L $$aJele, Harald
331   L $$aWissenschaftliches Arbeiten: Zitieren
359   L $$avon Harald Jele
410   L $$aMünchen ; Wien
412   L $$aOldenbourg
425a  L $$a2003
433   L $$a145 S.
434   L $$aIll., graph. Darst.
435   L $$a24 cm
517   L $$aLiteraturverz. S. 141 - 144. - Link-Verz. S. 145
540a  L $$a3-486-27506-2$$bgeb. : EUR 18,30 ; EUR 17,80 (D)
655e  L $$uhttp://ubdocs.uni-klu.ac.at/open/texte/AC03842486.pdf
700g  L $$aAK 39580
902   L $$sWissenschaftliches Arbeiten$$94066571-9
902   L $$sZitat$$94067889-1
902   L $$fRichtlinie
907   L $$sVeröffentlichung$$94187925-9
907   L $$sZitat$$94067889-1
907   L $$fRichtlinie
912   L $$sBibliographieren$$94112753-5
912   L $$sZitat$$94067889-1
912   L $$fRichtlinie
```

Abbildung 7: Katalogisierte MAB2-Kategorien einer Monographie im Bibliothekssystem

Abbildung 8: Exportiertes Kategorienschema einer Monographie nach Dublin-Core (Anzeige des Datensatzes im Web-OPAC in der Vollanzeige). Die zugehörigen Dublin-Core-Einträge innerhalb der Meta-Tags dieses Datensatzes sind in *Fußnote 15* angeführt

Weiters können bei der Anreicherung prinzipiell Informationen sowohl aus den titelrelevanten Bestandsangaben als auch aus den Exemplardaten[19] berücksichtigt und mit in den zu erstellenden Datensatz aufgenommen werden.

[19] zu den relevanten Exemplardaten zählen in unserem Fall hauptsächlich die unterschiedlichsten Standortinformationen inkl. Systematikvermerken und Signaturen. Wesentlich sind diese für das weitere Retrieval im vorliegenden Ansatz jedoch nicht. Aus diesem Grund bleibt die Beschreibung von Methoden der Anreicherung mit Exemplardaten in weiterer Folge auch unberücksichtigt. Vielmehr wird eine vollständige Exemplar- bzw. Bestandsangabe durch Rückgriff auf das OPAC-System realisiert

Der letzte zu berücksichtigende Vorgang innerhalb der Datenaufbereitung ist wesentlich von der Entscheidung geprägt, ob die bibliographischen Datensätze, die in Folge indiziert werden, ausschließlich als Datensätze in einer Datenbank gespeichert bleiben – oder ob diese weiter in Dateien exportiert werden.

Für die ausschließliche Speicherung innerhalb einer Datenbank sprechen die Vorteile, dass die üblichen SQL/XQuery-Werkzeuge zur Bearbeitung der in dieser Form abgelegten Inhalte im Grunde bestens dafür geeignet und frei verfügbar sind.[20] Dagegen spricht, dass die beobachtbare Performanz, die bei der Gewinnung des bibliographischen Index erzielt wird, oft sehr enttäuschend (und zumindest in vielen Fällen schwierig im Vorfeld abzuschätzen) ist.[21]

Für die Speicherung in einzelnen Dateien spricht, dass gerade im Bereich der Performanzsteigerung beim Volltextretrieval umfangreiche Erkenntnisse sowie praktische Anleitungen, die aufgrund von Erfahrungen erzielt wurden, zugänglich und nachvollziehbar umzusetzen sind. Zudem spricht für diesen Ansatz, dass die zu indizierenden Dateien bereits in einem anzeigbaren Format vorliegen können[22], wodurch sich eine entsprechende Aufbereitung bei der Anzeige erübrigt. Dieser Umstand macht sich gerade bei intensiver Benutzung des Systems in positiver Weise deutlich bemerkbar.

Nachteilig wirkt sich dbzgl. der Umstand aus, dass die dabei entstehenden Dateien bei jeder relevanten Änderung innerhalb der im Bibliothekssystem gespeicherten Daten neu erstellt und indiziert werden müssen.

```
FMT   L MH
LDR   L 00835nM2.01000024------h
001   L $$aAC00748398
030   L z|1dcr|||||27
036a  L $$aDE
037b  L $$ager
050   L a|a||||||||||
051   L no|||||
070   L $$aUBI
076f  L $$aWUW alle U-Sätze gelöscht
076s  L $$a10
100b  L $$aAllmendinger, Jutta
200   L $$aZentrum für Umfragen, Methoden und Analysen
        <Mannheim>$$9117597-X
204b  L $$aInformationszentrum Sozialwissenschaften
        <Bonn>$$92053774-8
331a  L $$aZUMA-Handbuch sozialwissenschaftlicher Skalen
359   L $$aZUMA, Zentrum für Umfragen, Methoden u. Analysen ;
        Informationszentrum Sozialwissenschaften. Wiss. Bearb.:
        Jutta Allmendinger... Dokumentar. Bearb.: Theodor
        Eikelmann ; Peter Ohly. Fortgef. von Dagmar Krebs (wiss.
        Bearb.) ...
370a  L $$aHandbuch sozialwissenschaftlicher Skalen
370a  L $$aSkalen-Handbuch
410   L $$aBonn
412   L $$aInformationszentrum Sozialwiss.
432   L $$aLfg. 1(1983) - 4(1988)
433   L $$aLosebl.-Ausg.
435   L $$a33 cm
505   L $$pNebent.$$aSkalen-Handbuch
540a  L $$a3-8206-0019-1
574   L $$a86,B08,0042
902   L $$sEinstellungsmessung$$94151420-8
902   L $$sSkala$$94317067-5
902   L $$fTabelle
903   L $$a213
```

Abbildung 9: MAB2-Kategorien zum ersten Band eines mehrbändigen Werkes im Bibliothekssystem (=übergeordneter Datensatz)

Abbildung 10: Exportiertes Kategorienschema zum ersten Band eines mehrbändigen Werkes nach Dublin-Core (=Titeldaten aus dem übergeordneten Datensatz, ergänzt durch die Bandangaben aus dem entsprechenden abhängigen Satzes. Anzeige des Datensatzes im Web-OPAC in der Vollanzeige)

3.2 Datenindizierung

Um den Benutzern/innen die außerhalb des Bibliothekssystems gehaltenen Daten zugänglich zu machen, werden diese mit einem externen Programm indiziert. Über

[20] eine brauchbare Liste relevanter Open-Source- wie Closed-Source-Produkte zur Indizierung von XML-Daten ist unter folgendem Link zugänglich
http://www.w3.org/XML/Query
Ohne genauer auf diesen Umstand einzugehen, sollte daraus erkennbar sein, dass für die Speicherung
- sowohl eine „klassische“ relationale Speicherung (ausschließlich in einem entsprechenden relationalen Datenbanksystem) und ein daraus resultierendes SQL-Retrieval
- als auch eine XML-basierende Speicherung als Textobjekte (innerhalb eines Filesystems oder in gleicher Weise auch in einem entsprechend ausgeführten Datenbanksystem) und ein darauf basierendes XQuery-Retrieval

möglich sind. Die Entscheidung für eine der beiden Methoden ist wohl im Wesentlichen vom gegebenen Arbeitsumfeld sowie den sich daraus ergebenden typischen Fragestellungen und Anforderungen vorgegeben. Bei der Verarbeitung von sehr großen Datenmengen bleiben daneben die bereits erwähnten Anforderungen an die Systemperformanz zu berücksichtigen

[21] dieser Hinweis ist besonders bei der Verwendung von umfangreichen XML-Attributen zu beachten, die via XQuery selektiert werden müssen.
Als Beispiel dafür kann die Open-Source-Implementierung „Galax“
vgl. http://www.galaxquery.org/
genannt werden, die sich besonders dadurch auszeichnet, dass ihre Programmierer aktuelle Vorgaben des Web-Konsortiums (W3C) rasch, vollständig und standardkonform umsetzen

[22] z.B. als xHTML-Dateien, deren Layout zentral durch eine externe CSS-Datei geregelt ist. Die Attribute des Dublin-Core-Formats sowie deren Inhalte sind dabei in den „Meta-Tags“ standardkonform hinterlegt

eine – den gängigen Kriterien entsprechend gestaltete – Web-Suchmaske sind die aufbereiteten Inhalte wieder auffindbar.

Bei der Auswahl eines geeigneten Programms zur Indexierung wurde vor allem auf folgende Bedingungen Rücksicht genommen:

- da die extern gehaltenen, bibliographischen Daten in strukturierter Form gespeichert werden, sollte sowohl bei der Indizierung als auch beim Retrieval auf die (sinngebenden, bedeutungstragenden) Eigenschaften der Datenstruktur Rücksicht genommen werden können.
  Das heißt, eine Suche muss unter Berücksichtigung der (=durch Einschränkung auf die) Dublin-Core Kategorieneinträge durchführbar sein
- (zumindest) die gängigen Boole'schen Operatoren sollen zur Mengenbildung im Suchvorgang realisiert sein. Auf den Einsatz von Proximity-Operatoren oder Gewichtungsfaktoren wird kein besonderer Wert gelegt. Beim Retrieval sehr großer Datenmengen können diese jedoch für eine sinnvoll gestaltbare Sortierung von Ergebnismengen von praktischem Nutzen sein[23]
- bibliographische Daten, die „Einwort-Titel" beinhalten und zudem keinen zugehörigen Personennamen (das sind üblicherweise die Namen der/des Autor/in/s oder Herausgeber/in/s) tragen, sind im Bereich der Zeitschriftenliteratur gängig. Zudem ist zu beachten, dass die davon betroffenen Titelwörter zumeist sehr häufig verwendete sind. Mit einer Stichwortsuche sind diese bibliographischen Daten nur sehr schwierig in einer Suchanfrage zu isolieren, denn in solchen Fällen qualifizieren sich sehr viele Treffer als Ergebnis.
  Mit der Realisierung von Phrasenindizes kann diesem Problem in sehr günstiger Weise begegnet werden. Bei der Entscheidung für ein Indexsystem sind die Möglichkeiten der Einrichtung eines entsprechend funktionierenden jedenfalls aufmerksam zu verfolgen
- sowohl das Index-System als auch die zugehörige Retrieval-Software müssen den `UTF8`-Zeichensatz möglichst vollständig unterstützen und evtl. sinnvolle Ersetzungsmechanismen von `UTF8` in den (extended) `Latin1`-Zeichensatz bieten, um außerhalb der lateinischen Grundzeichen ein Retrieval auch dann zu ermöglichen, wenn das entsprechende Zeichen über die Eingaben nicht zugänglich ist[24]
- um die Indizierung möglichst zeitnah am aktuellen Stand der Daten des Bibliothekssystems halten zu können, ist die Möglichkeit einer Indizierung von Teilbereichen oder gar einzelner Datensätze von Vorteil. Relativiert wird diese Notwendigkeit, wenn die Indizierung sehr großer Datenmengen durch entsprechende Leistungsfähigkeit auch in sehr kurzen Zeiträumen erfolgen kann[25]
- vor allem aus Kostengründen (zumeist aber auch aus Gründen größerer Flexibilität und Transparenz der eingesetzen Methodik) werden Open-Source-Produkte bevorzugt
- die Unabhängigkeit von bestimmten Betriebssystemen ist für den langfristigen, dauerhaften Einsatz von großem Vorteil und ist in diesem Fall besonders wünschenswert. Eine kontinuierliche Weiterentwicklung der Produkte ist dafür Voraussetzung und bedarf genauer Beobachtung bzw. der Kenntnis des Entwicklungsfortgangs.

Aufgrund dieser Vorgaben konnten zwei Produkte ausgewählt werden, wobei zur konkreten Realisierung die Entscheidung für das zweite (`Swish-e`) gefällt wurde:

1. Die Open-Source-Library `Lucene` erfüllt nicht nur sämtliche zur Auswahl herangezogenen Kriterien, sondern bringt zudem viele weitere, sinnvolle Funktionen mit.[26] Dazu gehören
   • solche zur „Normalisierung" von Begriffen (=die Rückführung bestimmter Wortformen auf die Grundform)
   • Funktionen für überaus performant arbeitende Bereichssuchen
   • ein Query-Parser zur Optimierung von kombinierten Suchanfragen mit großer Teilmengen-Trefferanzahl
   • eine von den Programmierern so bezeichnete „Fuzzy-Suche", bei der morphologisch ähnliche Wörter mitgesucht werden
   • sowie eine „Wildcard-Funktion" neben der üblichen Wort-Trunkierung.

   `Lucene` ist für die effektive Indizierung großer Datenmengen geeignet.
   Die auf der entsprechenden Web-Site publizierten Benchmark-Ergebnisse[27] zeigen, dass auch mehrere

[23] das heißt, solange ausschließlich bibliographische Titeldaten indiziert werden, kann auf diese Mechanismen zumeist ohne Verlust von wesentlichen Sortiermöglichkeiten verzichtet werden.
Sollen jedoch neben den Titeldaten zudem Referenzen auf Volltexte im gleichen Index abgebildet werden, bieten die üblichen Gewichtungsfaktoren mitunter wichtige erweiterte Funktionen

[24] das meint vor allem die Möglichkeiten der Rückführung typographisch kombinierter Zeichen auf die entsprechenden lateinischen Grundbuchstaben
wie z.B. `š` → `s`.
Im Übrigen sind bei der Indexierung jene Ersetzungs-Mechanismen von (Sonder-)Zeichen zu bedenken, die in Jele (2001, S.38-40) auszugsweise angeführt sind

[25] eine Entscheidung für eine regelmäßige Gesamt-Neuindizierung würde quasi zu einem „`Offline-PAC`" führen, der z.B. wöchentlich aktualisiert würde. Bibliothekssysteme, bei denen zwischen einem „Publikums-" und einem „Bearbeiter-Katalog" unterschieden wird, sind hinreichend bekannt, etabliert und (ein wenig abhängig von eher modischen Erscheinungen) verbreitet

[26] ein kurzer aktueller Abriss der Funktionen von `Lucene` findet sich in Naber (2005)

[27] vgl.
`http://lucene.apache.org/java/docs/benchmarks.html`

Millionen Texte/Datensätze damit indizierbar und mit ansprechender Performanz retrievebar sind (auch sehr komplexe Queries liefern Antworten typischerweise im Sekundenbereich).
Laut den dort publizierten Ergebnissen dauert ein Index-Lauf ca. `40s` für `1000 Dokumente` mit der Größe von jeweils ca. `1024 Byte`, bei denen zudem die Zugehörigkeit von Indexeinträgen zu `10 Kategorien` berücksichtigt werden musste. Das ergibt für 1 Million Datensätze eine Gesamt-Indexierungsdauer von ca. `11 Stunden` auf durchschnittlicher PC-Hardware.
Bei der Berücksichtigung dieser Angaben muss angemerkt werden, dass diese Kennzahlen durchaus Relevanz für die Indizierung bibliographischer Daten haben, denn die Kenngröße von `1024 Byte` entspricht in etwa der Größe eines durchschnittlichen bibliographischen Datensatzes.

Das Einsatzgebiet von `Lucene` sehen die Programmierer nicht auf die Indizierung typischer, unstrukturierter Text-Dokumente beschränkt. Das Zutreffen dieser Einschränkung verhindert ja zumeist, dass eine x-beliebige „Volltext-Suchmaschine“ zur Indizierung und zum Retrieval bibliographischer Daten verwendet werden kann. Beim praktischen Einsatz von `Lucene` stehen alle wesentlichen Funktionen zur Indizierung von strukturierten Daten zur Verfügung. Dies zeigt sich auch daran, dass das Kategorisierungsschema, nach dem zu indizierende Daten strukturiert sein können, offen definiert ist. Dies kommt den Ansprüchen der Indizierung bibliographischer Daten entgegen, da die allermeisten Kategorisierungsschemata ebenso offen definiert sind.

`Lucene` ist in der jeweils aktuellsten Version in der Programmiersprache `Java` realisiert, liegt jedoch als Portierung in einigen gängigen Programmiersprachen vor. Zudem steht für einige Programmiersprachen eine Schnittstelle (`API =Application Programmable Interface`) zur Verfügung, um `Lucene` in eigene Anwendungen integrieren zu können.

2. Während `Lucene` sich dem/r Benutzer/in gegenüber als Programm-Bibliothek (Library) versteht, deren Schnittstellen in jeder Hinsicht offen, transparent und portabel sind und die sich aus diesem Grund gut in jede Anwendung einbetten lässt, so ist `Swish-e` ein Programm-Paket, das viele Anwendungsfälle durch fertige Programme bereits abdeckt.[28]

   In den Kernfunktionalitäten sind beide Produkte vergleichbar:
   • `Swish-e` ist in der Lage eine große Anzahl an Dokumenten (Texte, Datensätze) zu indizieren. Leider sind keine vergleichbaren Leistungstests zugänglich. Die Mehrzahl der Anwender indiziert damit jedoch einige Hunderttausend Dokumente auf handelsüblicher PC-Hardware.
   Eigene Tests zeigen, dass das Indexsystem durchaus auch jenseits einer Million Dokumente noch sehr performant funktioniert (wobei wiederum angenommen werden kann, dass die zu indizierenden Inhalte der Dokumente typischerweise in der Größenordnung von `1024 Byte` vorkommen).[29]
   Die im konkreten Fall zu indizierende Anzahl an bibliographischen Datensätzen liegt mit ca. `370.000` im Bereich der typischen Anwendungsfälle. Die Vervierfachung dieser Anzahl brachte für das Retrieval unter Laborbedingungen (=ohne „natürliche Stresssituationen“ des Web-Servers durch hunderte gleichzeitige Benutzer/innen-Anfragen) keine signifikanten Abweichungen im Antwortverhalten.
   Die Antwortzeiten für die in *Abschnitt 2* angeführten Beispiele liegen im Bereich von `2-4s` – und sind damit um den Faktor `10` kürzer als jene des Vergleichs-OPAC-Systems
   • „Filter“ zum direkten Indizieren der gängigen Dokumentformate sind im Programmpaket bereits vorgesehen und einsatzbereit. Das bedeutet, dass neben den bibliographischen Datensätzen auch Volltextquellen in den dafür typischen Formaten `PDF` und neuerdings vermehrt `XML/XSL` ohne Umwege (über die sonst meist notwendige externe Konvertierung) in den gleichen Index aufgenommen werden können.

   Alle weiteren Funktionalitäten sind – wie oben angeführt – jenen von `Lucene` sehr ähnlich und werden aus diesem Grund auch nicht weiter besprochen. Eine vollständige Liste findet sich unter den so bezeichneten „Key features“ auf der entsprechenden Web-Site von `Swish-e`.[30]

Besonders der Umstand, dass für die gängigen Anwendungsfälle[31] bereits vorgefertigte und leicht zu adaptierende Programmelemente im Programmpaket von `Swish-e` enthalten sind, führte zur Entscheidung für dieses Tool.

[28] `Swish-e` = `Simple Web Indexing System for Humans` – `Enhanced`; wobei die Namensgebung nicht dazu verleiten darf anzunehmen, dass ausschließlich „Web-Inhalte“ (also `HTML`-Texte) indizierbar wären

[29] die beiden als Beispiele in diesem Text angeführten Datensätze (vgl. *Abb. 8* und *Abb. 10*) sind zum direkten Vergleich mit diesen Angaben `682` und `643 Byte` lang

[30] vgl. `http://swish-e.org/docs/readme.html`

[31] dazu zählten für uns besonders
• eine Beispiel-Web-Suchmaske, anhand derer die Integration mittels `CGI` (=`Common Gateway Interface` des Web-Servers) in eine bestehende Web-Server-Umgebung gezeigt wird
• die mitgelieferten Tools zur Inspektion und Kontrolle des Indexsystems
• sowie die referenzierten Demo-Installationen, die eine Vielfalt an Gestaltungs- und Einsatzmöglichkeiten aufzeigten

4 Zusammenfassung

Es konnte gezeigt werden, dass spezifische Nachteile, die OPAC-Systeme vor allem im Bereich der Performanz des Datenretrievals aufweisen, durch die externe Indizierung bibliographischer Daten behoben werden können.

Die durchgeführten Messungen zur Leistungsfähigkeit am Beispiel eines Open-Source-Produkts (`Swish-e`) sowie die Berücksichtigung der publizierten Werte zu den Benchmarks eines Vergleichsprodukts (`Lucene`) weisen darauf hin, dass auf handelsüblicher PC-Hardware sehr ansprechende Ergebnisse erzielbar sind (während komplexere Bibliothekssysteme üblicherweise sehr aufwändige und teure Hardware voraussetzen).
Unter künstlichen Stressbedingungen (hergestellt durch eine typische „Last-Simulation“) zeigt sich, dass die Performanz des eingesetzten Web-Servers deutlicher messbar ist, als jene des Indexsystems. Aus diesem Grund darf nicht vergessen werden, beide Komponenten auf die vorhandene Situation entsprechend abzustimmen.

Erwähnt muss werden, dass bei diesem Ansatz neben merklich spürbaren Verbesserungen der Systemperformanz auch wesentliche Einsparmöglichkeiten im Bereich notwendiger OPAC-Lizenzen erkennbar sind.

Zu bedenken bleibt, dass hier überwiegend ein Datenretrieval zugrunde gelegt wird, das (idealisiert) allein aus der Sicht eines/r „Suchmaschinen-gewohnten“ Benutzers/in definiert wird. Unter Berücksichtigung grundlegender, bibliothekarischer Bedürfnisse und Notwendigkeiten ist der Ansatz wohl um die Hinweise in Eversberg (2003) zu erweitern.

5 Literaturverzeichnis

5.1 Gedruckte Quellen

Eversberg, Bernhard (2003): Zur Theorie der Bibliothekskataloge und Suchmaschinen. In: Die Bibliothek zwischen Autor und Leser. 92. Deutscher Bibliothekartag in Augsburg 2002. (=ZfBB-Sonderheft 84), S.113-126
online aktualisiert:
`http://www.allegro-c.de/formate/tks.htm`
sowie in englischer Sprache:
`http://www.allegro-c.de/formate/tlcse.htm`

Jele, Harald (2001): Informationstechnologien in Bibliotheken. Oldenbourg, München

Kostädt, Peter (2003): FAST DATA Search für den schnellen Zugriff auf die HBZ-Verbunddaten. (=Vortrag anlässlich der Verbundkonferenz des HBZ. Köln, 18.11.2003)
online:
`http://www.hbz-nrw.de/kunden/verbundkonferenz/fast_kostaedt.pdf`

Naber, Daniel (2005): Herr der Suche. Eigene Anwendungen mit Volltextsuche erweitern. In: c't. Magazin für Computertechnik, Heft 7, S.196-199

5.2 Online-Quellen

`http://www.contentmanager.de/magazin/news_h5612-print_fast_und_universitaet_bielefeld_foerdern.html` : Fast Search & Transfer Mitteilung: FAST und Universität Bielefeld fördern Einsatz von Enterprise-Suchtechnologie. Beitrag mit eigenem Erscheinungsdatum vom 20.08.2003

`http://dublincore.org/` : Definition der Kategorien nach Dublin-Core

`http://selfhtml.org/` : Stefan Münz u.a. : SELFHTML (siehe die Beschreibung zum standardkonformen Einsatz der Kategorien nach Dublin-Core innerhalb der Meta-Tags von HTML)

`http://swish-e.org/` : Open-Source-Software zur Indizierung von Volltexten

`http://lucene.apache.org/` : Open-Source-Software zur Indizierung von Volltexten

`http://www.at-web.de/newsletter/archiv/news107.htm`: @-web.de. Online-Magazin für Nachrichten aus dem Web. Newsletter 107 vom 11.07.2003

`http://www.biblio.tu-bs.de/allegro/news/acn972.htm` : Eversberg, Bernhard (1997) : Schillers Räuber gefaßt! (=allegro news Nr.46, Ausgabe 97/2 vom 16. Mai 1997)

`http://www.galaxquery.org/` : Open-Source-Implementierung eines XQuery-Systems

`http://www.w3.org/XML/Query/` : Aktuelle Standardbeschreibung von XQuery sowie eine Produktübersicht (Open- und Closed-Source) zur Indizierung von XML-Daten

Dr. Harald Jele ist Leiter der Abteilung EDV-Administration und -Entwicklung der Universitätsbibliothek Klagenfurt

Adresse:
Universität Klagenfurt
Universitätsstraße 65-67
9020 Klagenfurt, Österreich
Fax: 0043-463-2700-9599
E-Mail:harald.jele@uni-klu.ac.at