Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

WOLFRAM SCHNEIDER

Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls

Diese Publikation wird gefördert von der Senatsverwaltung für Wissenschaft, Forschung und Kultur des Landes Berlin und von dem Ministerium für Wissenschaft, Forschung und Kultur des Landes Brandenburg, im Rahmen des Projektes Kooperativer Bibliotheksverbund Berlin Brandenburg (KOBV).

Preprint SC 99-21 (Juli 1999)

Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls

eingereicht von

Wolfram Schneider

als Diplomarbeit am Fachbereich Informatik
der Technischen Universität Berlin

betreut von

Diplom-Informatiker Josef Willenborg

Gutachter:

Prof. Dr. Erhard Konrad und Prof. Dr. Martin Grötschel

Berlin, Juli 1999

Wolfram Schneider
Matr.-Nr. 136513
Garibaldistr. 50
D-13158 Berlin
E-Mail: schneider@zib.de
http://wolfram.schneider.org

Konrad-Zuse-Zentrum für
Informationstechnik Berlin
Josef Willenborg
Takustraße 7
D-14195 Berlin-Dahlem
E-Mail: willenborg@zib.de
http://www.zib.de

Technische Universität Berlin
Fachbereich Informatik
Wissensbasierte Systeme
Prof. Dr. Erhard Konrad
Franklinstr. 28/29
D-10587 Berlin
http://www.cs.tu-berlin.de

Technische Universität Berlin
Fachbereich Mathematik
Prof. Dr. Martin Grötschel
Sekretariat MA 4-1
Straße des 17. Juni 136
D-10623 Berlin-Charlottenburg
http://www.math.tu-berlin.de

Überblick

Diese Diplomarbeit beschreibt ein verteiltes Bibliotheks-Informationssystem für bibliographische Datenbanken im Internet. Der Name des Systems ist *ZACK*.

Der Benutzer kann mit *ZACK* in einer oder mehreren bibliographischen Datenbanken nach einem Dokument suchen und die Treffer in die eigene lokale Datenbank übernehmen. Mit der Übernahme der Datensätze aus einer fremden Datenbank wird die Erfassung neuer Dokumente wesentlich erleichtert, da die Eigenkatalogisierung auf ein Minimum beschränkt werden kann. Es wird doppelte Arbeit vermieden, und die Datensätze haben eine gleichbleibend hohe Qualität.

Bei der verteilten Suche mit *ZACK* wird parallel in mehreren Datenbanken gesucht. Dubletten werden als solche erkannt. Dem Benutzer wird eine übersichtliche Kurztrefferliste ohne doppelte Einträge angeboten. Er kann dann selbst entscheiden, aus welcher Datenbank er die Datensätze übernimmt. Die verteilte Suche hat in der Praxis eine deutlich bessere Trefferquote gebracht als die Suche in nur einer Datenbank. Dabei bleibt die Antwortzeit in einem für den Benutzer akzeptablen Rahmen. Die Kurztrefferliste wird durch die Dublettenkontrolle kürzer und übersichtlicher.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

Seit den siebziger Jahren werden Kataloge in Bibliotheken elektronisch erfaßt. Die Entwicklung der Computertechnik in den letzten Jahren hat die Voraussetzungen dafür geschaffen, daß immer mehr Informationen in Datenbanken verwaltet werden. Mit dem Erfolg des World-Wide-Web beginnen viele Anbieter, ihre Datenbanken externen Nutzern zu öffnen. Die Benutzer können dann direkt über das Internet in der Datenbank recherchieren und Daten austauschen.

Ziel dieser Diplomarbeit ist die Entwicklung eines Bibliotheks-Informationssystems, das Bibliothekare bei der Recherche und Erfassung von Dokumenten unterstützt. Der Name des Systems ist *ZACK*[1]

Der Benutzer von *ZACK* kann in einer oder mehreren bibliographischen Datenbanken nach einem Dokument suchen und das geeignete Dokument in die lokale Datenbank übernehmen. Mit der Übernahme der Datensätze aus einer fremden Datenbank wird die Erfassung neuer Dokumente wesentlich erleichtert, da die Eigenkatalogisierung auf ein Minimum beschränkt werden kann. Es wird doppelte Arbeit vermieden, und die Datensätze haben eine gleichbleibend hohe Qualität.

Bei der verteilten Suche, die *ZACK* ermöglicht, wird parallel in mehreren Datenbanken gesucht. Dubletten werden als solche erkannt. Dem Benutzer wird eine übersichtliche Kurztrefferliste ohne doppelte Einträge angeboten. Er kann dann selbst entscheiden, aus welcher Datenbank er die Datensätze übernimmt. Die verteilte Suche hat in der Praxis eine deutlich bessere Trefferquote gebracht als die Suche nur in einer Datenbank. Dabei bleibt die Antwortzeit in einem für die Benutzer akzeptablen Rahmen. Die Kurztrefferliste wird durch die Dublettenkontrolle deutlich kürzer und übersichtlicher.

Es existieren bereits einige Systeme, die eine verteilte Suche *ohne Dublettenkontrolle* anbieten (siehe Abschnitt 2.3 Existierende Systeme zur verteilten Suche, Seite 8). Das in dieser Diplomarbeit entwickelte System *ZACK* bietet erstmals die Möglichkeit, gleichzeitig in mehreren Datenbanken unterschiedlicher Bibliotheken zu suchen und online eine Dublettenkontrolle durchzuführen.

ZACK kann von jedem Computer aus benutzt werden, gleich welche Hardware, welches Betriebssystem oder welche graphische Oberfläche benutzt wird (Windows 3.11, X11, VT100 Terminals). Erforderlich ist nur ein Web-Browser - eine Software, die inzwischen auf fast jedem Rechner verfügbar ist. Für jeden Nutzer kann anhand seiner Absenderadresse (IP-Adresse) eingestellt werden, auf welche Datenbanken er Zugriff hat und ob er die Datensätze in seine eigene Datenbank übernehmen darf. *ZACK* ist zweisprachig. Der Benutzer kann zwischen einer deutschen oder englischen Oberfläche wählen.

[1] *ZACK* ist ein Eigenname und keine Abkürzung

Überblick über die Kapitel

In Kapitel 2 **Verteilte Suche** werden grundsätzliche Vorüberlegungen zu Informationssystemen angestellt. Es wird dargelegt, warum die Suche in mehreren Datenbanken bessere Ergebnisse bringt als die Suche in nur einer Datenbank.

In Kapitel 3 **Modellierung von ZACK** wird anhand des Client-Server-Modells beschrieben, wie die Suche in einer Bibliotheksdatenbank über das Internet erfolgt. Dabei wird kurz erläutert, wie die Kommunikation zwischen Client und Server abläuft und welche unterschiedlichen Protokolle in *ZACK* verwendet werden. Es wird das Z39.50 Protokoll vorgestellt.

In Kapitel 4 **Implementierung von ZACK** wird das eigene System vorgestellt. Es wird beschrieben, in welchen Schritten *ZACK* entworfen und implementiert wurde. Es wird untersucht, welche vorhandene Software für die Entwicklung dieses Systems in Betracht kam und welche in *ZACK* verwendet wird. Mit dem ersten System kann man nur in einer Datenbank suchen, während man mit dem zweiten System parallel in mehreren Datenbanken suchen kann.

Im zweiten System von *ZACK* werden bei der Dublettenkontrolle Datensätze unterschiedlicher Herkunft miteinander verglichen. Bevor dies geschehen kann, müssen Zeichensatz, Fehleingaben (z.B. doppelte Leerzeichen), unterschiedliche Erfassungspraktiken erkannt und bearbeitet werden. In Kapitel 5 **Normierung** werden Datensätze unter diesen Gesichtspunkten analysiert.

In Kapitel 6 **Dublettenkontrolle** wird beschrieben, wie die Dublettenerkennung in *ZACK* durchgeführt wird. Es werden die verwendeten Algorithmen, der benötigte Rechenaufwand in *ZACK* und die Ergebnisse der Dublettenkontrolle erläutert.

Als Ergebnis der Dublettenkontrolle wird dem Benutzer eine Kurztrefferliste ohne doppelte Einträge angeboten. In Kapitel 7 **Ausgaben von Dubletten** werden die hierbei zur Auswahl stehenden Verfahren beschrieben, bewertet und erläutert, welches Verfahren in *ZACK* verwendet wird.

Das Kapitel 8 **Praktische Ergebnisse einer verteilten Suche** beschreibt, wie erfolgreich die verteilte Suche in der Praxis tatsächlich ist. Dazu werden die Anfragen von Bibliothekaren aus Brandenburg ausgewertet, die mit *ZACK* gesucht haben - einmal die Anfragen eines Tages und einmal über den Zeitraum von mehreren Monaten.

In Kapitel 9 **Probleme im laufenden Betrieb** werden kleinere und größere Probleme beschrieben, die in der praktischen Nutzung der mit *ZACK* angesprochenen Z39.50-Server aufgetreten sind. Im Detail werden die Konfigurationsprobleme mit den Z39.50-Servern unterschiedlicher Bibliothekssysteme, unterschiedlicher Hersteller und unterschiedlicher Bibliotheken und Bibliotheksverbünde aufgeführt.

In Kapitel 10 **Ausblick** wird ein Fazit der Arbeit gezogen und Vorschläge für eine weitere Nutzung des Systems gemacht.

Im Anhang A **Analyse der MAB2-Datensätze der Deutschen Bibliothek** werden 2,5 Millionen Datensätze der Deutschen Bibliothek statistisch ausgewertet. Ziel ist es festzustellen, welche Felder in den Datensätzen wirklich genutzt werden, wie das Verhältnis zwischen Büchern aus Verlagen und sonstiger Literatur ist und welche Beziehungen (Hierarchien, Verweise) zwischen den Datensätzen existieren. Diese Informationen sind für die Dublettenkontrolle und die Ausgabe der Datensätze erforderlich.

Im Anhang B **Z39.50-Server** befindet sich eine Liste der genutzen Z39.50 Server; im Anhang C **Software** eine kurze Beschreibung der für *ZACK* geschriebenen Software; im Anhang D **Zugriffsstatistik** eine Auswertung der Nutzung von *ZACK* im praktischen Betrieb. Zum Anhang gehören außerdem ein **Abkürzungsverzeichnis** und ein **Literaturverzeichnis**.

Kapitel 2

Verteilte Suche

Bis vor wenigen Jahren waren Bibliothekssysteme in sich geschlossenene Systeme. Ein Benutzer hatte über seine Benutzeroberfläche lediglich Zugriff auf die lokale Datenbank seines Systems. Um auf Datenbestände eines entfernten Systems zugreifen zu können, mußte er sich in das entfernte System einwählen (z.B. über eine telnet Verbindung) und dann über die Benutzeroberfläche des entfernten Systems die gewünschten Recherchen durchführen. Der Benutzer mußte somit mit den Eigenheiten mehrerer Systeme vertraut sein. Für die Übertragung von Daten aus dem entfernten ins eigene System waren weitere nicht standardisierte Verfahren erforderlich.

Die Idealvorstellung, einem Benutzer über seine eigene, ihm vertraute Benutzeroberfläche, unter Verwendung der ihm bekannten Funktionen den Zugriff auf alle für ihn relevanten Datenbestände zu ermöglichen, war die Basis für die Definition des Protokolls Z39.50. Über Z39.50 werden Funktionen, die dem Benutzer bisher nur für den Zugriff auf die lokalen Datenbestände zur Verfügung standen, auch auf entfernte Datenbanken anwendbar. Aus Sicht des Benutzers besteht prinzipiell kein Unterschied zwischen entfernten und lokalen Datenbanken (siehe auch Kapitel 3.4 Modellierung, Seite 11).

Für Bibliotheksmitarbeiter als Nutzer eines Bibliothekssystems vereinfacht sich durch diese Systemöffnung die Katalogisierung neuer Bibliotheksobjekte, da Katalogisate nun aus entfernten Systemen ins eigene System übernommen werden können (nach [Her96]).

Mit Z39.50 kann man nicht nur in einer entfernten Datenbank, sondern auch in mehreren entfernten Datenbanken gleichzeitig recherchieren (verteilte Suche). Für den Benutzer erhöht sich damit die Wahrscheinlichkeit, die gewünschte Information zu finden.

Systeme zur verteilten Suche verfügen über keine eigene Datenbank, sondern nutzen die Datenbanken anderer Anbieter über das Internet. Die verteilte Suche kann deshalb nicht mehr Funktionalität bei der Recherche bieten als von den einzelnen Datenbanken selbst angeboten wird ([KVK99]).

Für die verteilte Suche werden im folgenden grundsätzliche Vorüberlegungen angestellt. In Abschnitt 2.1 **Kriterien zur Bewertung von Informationssystemen** wird am Modell Precision und Recall beschrieben, nach welchen Kriterien die Qualität eines Informationssystems bewertet werden kann. Dabei wird insbesondere auf die verteilte Suche eingegangen hinsichtlich der Fragestellung, welche Ergebnisse von der verteilten Suche erwartet werden dürfen.

Der Abschnitt 2.2 **Bibliotheken und Bibliotheksverbünde in Deutschland** beschreibt die Geschichte der Bibliotheksverbünde und Bibliothekssysteme in Deutschland. Das in Rahmen dieser Diplomarbeit entwickelte System *ZACK* verfügt als verteiltes System über keine eigene Datenbank und nutzt die Datenbanken der deutschen Bibliotheksverbünde und einiger Universitätsbibliotheken.

In Abschnitt 2.3 **Existierende Systeme zur verteilten Suche** werden andere laufende Systeme, die eine verteilte Suche in deutschsprachigen Bibliothekssystemen anbieten, kurz vorgestellt.

2.1 Kriterien zur Bewertung von Informationssystemen

Informationssysteme werden an ihrem Nutzen gemessen. Wie hoch ist die Qualität der erreichten Lösung? Liefert das System die gewünschten Leistungen? Wie hoch sind die Kosten (Zeitaufwand für den Benutzer, benötigte Rechenleistung)?

Um die Qualität eines Informationssystems zu beurteilen, legt man im Information Retrieval (IR) [1] das Konzept der *Relevanz* zugrunde. Die Relevanz beschreibt dabei die Beziehung zwischen der Anfrage und einem einzelnen Treffer in der Antwortmenge (siehe [Fuh97]).

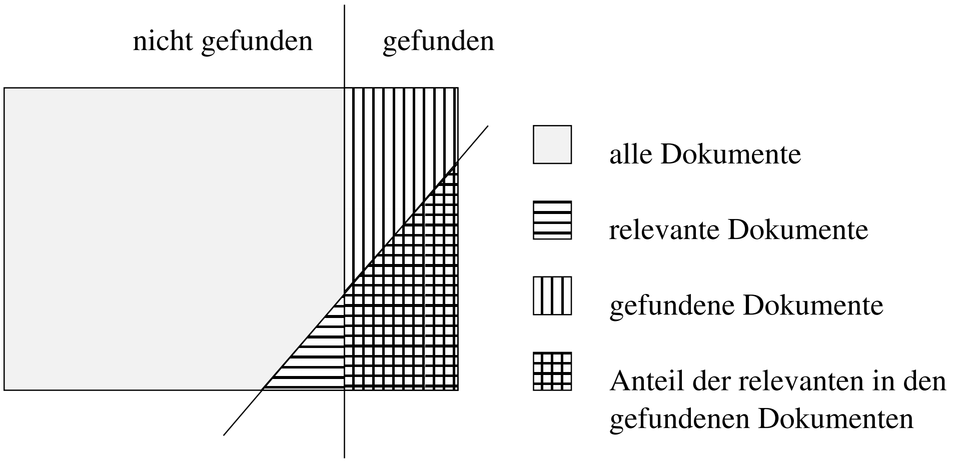


Abbildung 2.1: Maße für Boolesches Retrieval

Das Konzept der Relevanz wird in der Abbildung 2.1 verdeutlicht. Der Benutzer stellt eine Anfrage. Das System sucht in der gesamten Datenbank (Menge *aller Dokumente*) nach den für die Anfrage *relevanten Dokumenten*. Als Ergebnis gibt das System eine Menge an *gefundenen Dokumenten* (Treffermenge) aus. Für den Benutzer ergeben sich die folgenden Fragen:

1. Hat das System alle relevanten Dokumente gefunden? Falls nicht, wie hoch ist der Anteil der nicht gefundenen relevanten Dokumente (Fläche mit waagerechten Linien)?

2. Sind die vom System *gefundenen Dokumente* auch wirklich relevant zur Anfrage? Falls nicht, wie hoch ist der Anteil der *relevanten* (karierte Fläche) und der *nicht relevanten* Dokumente (Fläche mit senkrechten Linien) in der Treffermenge?

Der Benutzer sieht sich die *gefundenen Dokumente* nacheinander an und bewertet sie. Die Bewertung geschieht nach einem einfachen Schema: entweder sind die Dokumente relevant zur gestellten Anfrage, oder sie sind es nicht.

$$Precision: \quad p = \frac{\text{Anzahl der relevanten Dokumente} \cap \text{Anzahl der gefundenen Dokumente}}{\text{Anzahl der gefundenen Dokumente}}$$

[1] Will man den Gegenstand des Information Retrieval mit wenigen Worten beschreiben, so ist die Formulierung "inhaltliche Suche in Texten" wohl am treffendsten. Tatsächlich wird damit aber nur ein wesentlicher - wenn auch der wichtigste - Bereich des Information Retrieval umschrieben, den man auch häufig als Textretrieval oder Dokumentenretrieval bezeichnet. Das klassische Anwendungsgebiet des Textretrieval sind Literaturdatenbanken, in denen Kurzfassungen von Veröffentlichungen gespeichert werden, und die den Anwendern die Suche nach für sie relevanter Literatur in einem bestimmten Fachgebiet ermöglichen sollen (Definition nach [Fuh97]).

Die Precision gibt den Anteil der relevanten an den gefundenen Dokumenten wieder.

$$\textit{Recall}: \quad r = \frac{\text{Anzahl relevante Dokumente} \cap \text{Anzahl gefundene Dokumente}}{\text{Anzahl der relevanten Dokumente}}$$

Recall bezeichnet den Anteil der relevanten in den gefundenen Dokumenten.

Die Größe der Precision ist für jeden Benutzer eines Informationssystems direkt ersichtlich. Er sieht sich die gefundenen Dokumente an und bestimmt das Verhältnis der relevanten zu den gefundenen Dokumenten. Die Größe des Recall ist dagegen für einen Benutzer nicht erkennbar. Der Grund hierfür liegt in der Schwierigkeit, die Menge der *relevanten Dokumente* präzise zu bestimmen. Dies ist mit vertretbarem Aufwand nicht möglich (siehe [Fuh97]).

Der Wunsch, alle Suchanfragen bedienen zu können (hoher Recall) und dabei wenig irrelevante Dokumente zu liefern (hohe Precision), läßt sich kaum erfüllen. Will man alle relevanten Dokumente finden, so muß man in Kauf nehmen, daß auch nicht relevante Dokumente gefunden werden. Auf der anderen Seite - will man nur relevante Dokumente geliefert haben (hohe Precision) - verringert sich die Anzahl der gefundenen Dokumente, und man ignoriert eventuell andere relevante Dokumente. In der Praxis wird man versuchen, einen Mittelweg zwischen Precision und Recall zu wählen.

Es bietet sich an, nicht nur in einer Datenbank zu suchen. Falls das gewünschte Buch nicht in der einen Bibliothek vorhanden ist, so findet man es vielleicht in der nächsten oder übernächsten Bibliothek. Der Benutzer trifft eine Vorauswahl von Datenbanken, von denen er annimmt, daß sie für seine Anfrage geeignet sind. Beispielsweise erwartet man ein deutschsprachiges Buch eher in einer deutschen Bibliothek und nicht einer amerikanischen Bibliothek zu finden. Ein mathematisches Buch wird man eher in einer mathematischen Spezialbibliothek als in einer Allgemeinbibliothek erwarten.

Bei der Suche in zwei Datenbanken addieren sich die Anzahl aller vorhandenen Dokumente. Ein Teil der Dokumente ist in beiden Bibliotheken vorhanden (siehe Abbildung 2.2).

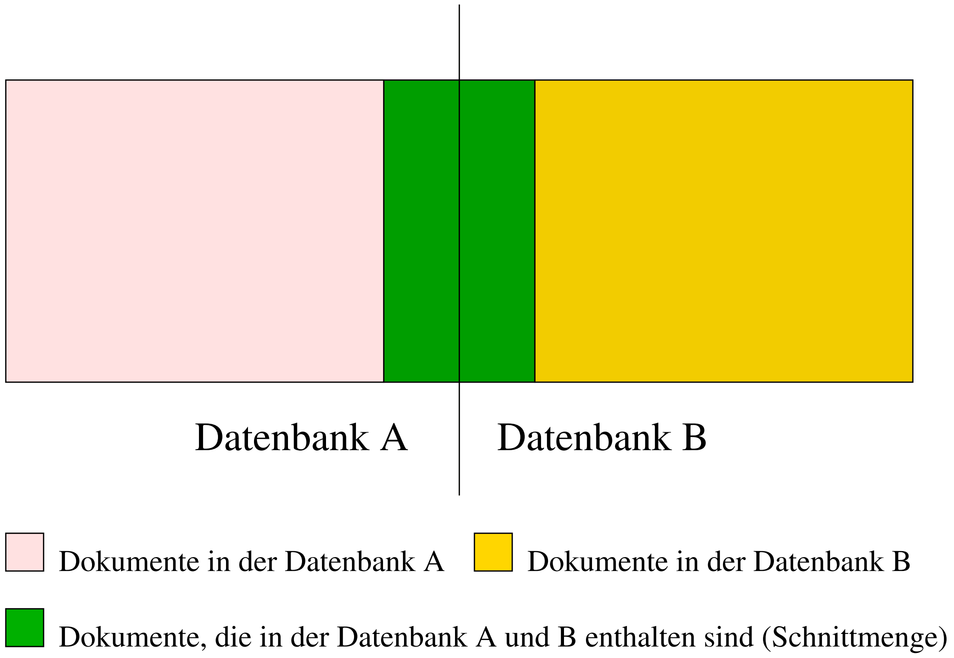


Abbildung 2.2: Bestand der Datenbanken A und B

Ebenfalls addieren sich die Anzahl der relevanten Dokumente in beiden Datenbanken und die Anzahl der gefundenen Dokumente bei der verteilten Suche (siehe Abbildung 2.3).

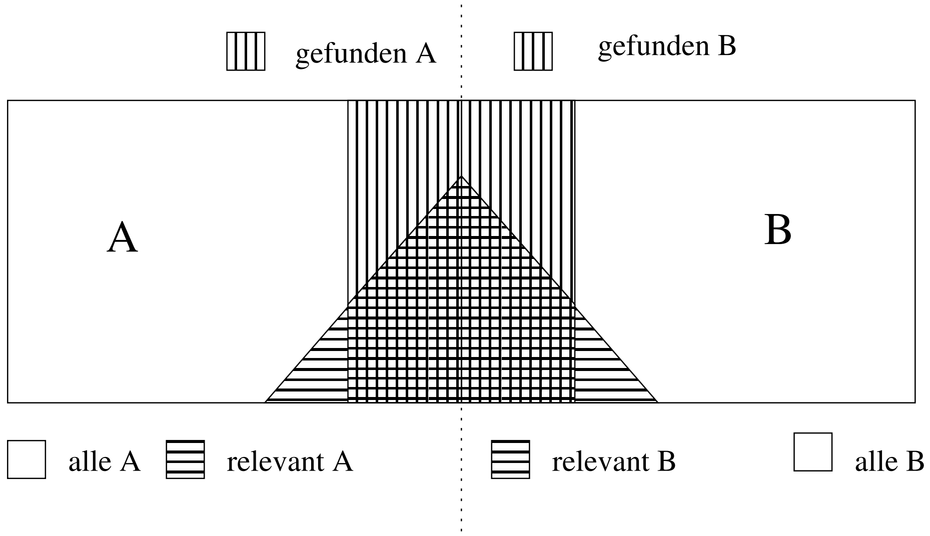


Abbildung 2.3: Relevanz bei verteilter Suche

Die Bewertung der gefundenen Dokumente ist bei der verteilten Suche aufwendiger geworden. Es gibt mehr gefundene Dokumente. Der Benutzer muß nicht nur zwischen relevanten und nicht relevanten Dokumenten unterscheiden, sondern auch die in beiden Datenbanken gefundenen Dokumente (Dubletten) herausfiltern (siehe Abbildung 2.4).

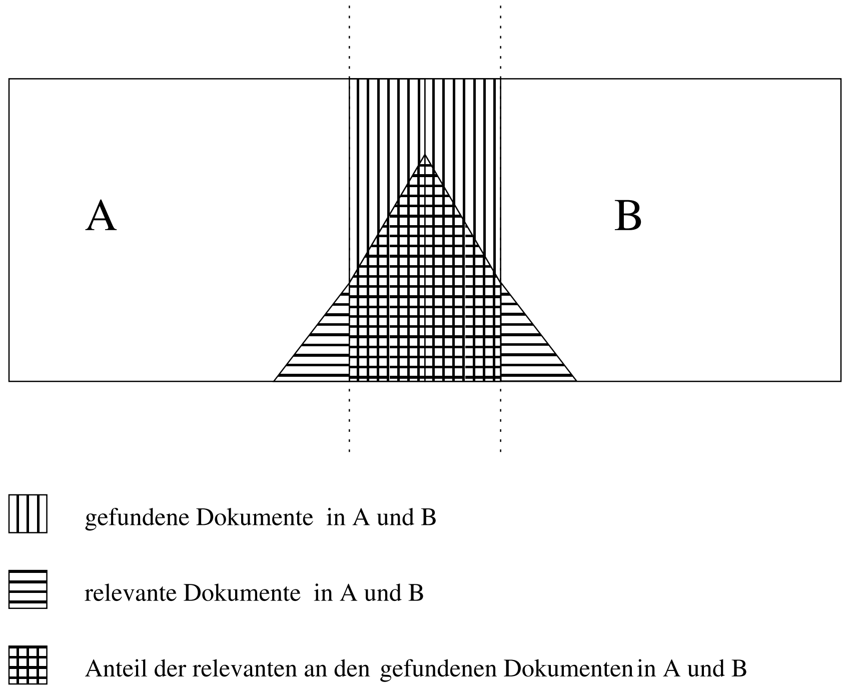


Abbildung 2.4: Relevanz bei verteilter Suche mit Dublettenkontrolle

Das in dieser Diplomarbeit entwickelte System *ZACK* ist ein WWW-Z39.50-Gateway und übernimmt die Dublettenkontrolle für den Benutzer automatisch. *ZACK* erkennt anhand vorgegebener Kriterien, ob es sich um gleiche oder ähnliche Dokumente handelt und bietet dem Benutzer eine übersichtliche Kurztrefferliste der gefundenen Dokumente. Die Anzahl der gefundenen Dokumente wird deutlich reduziert (siehe Kapitel 6 Dublettenkontrolle, Seite 60 und

Kapitel 7 Ausgabe von Dubletten, Seite 78).

Der Benutzer muß in *ZACK* bei der verteilten Suche nur noch zwischen den zur Anfrage *relevanten Dokumenten* und *nicht relevanten* Dokumenten unterscheiden. Bei der verteilten Suche in mehreren Datenbanken addiert sich die Anzahl der relevanten Dokumente. Für den Benutzer erhöht sich damit die Wahrscheinlichkeit, daß die gewünschte Information auch tatsächlich gefunden wird (Nutzen). Dabei sind die Kosten für den Benutzer (Zeitaufwand) bei der verteilten Suche mit Dublettenkontrolle nur minimal höher als bei der Suche in nur einer Datenbank.

Das System *ZACK* wird im Kapitel 3 **Modellierung** (Seite 10) und Kapitel 4 **Implementierung** (Seite 16) detailliert beschrieben.

2.2 Bibliotheken und Bibliotheksverbünde in Deutschland

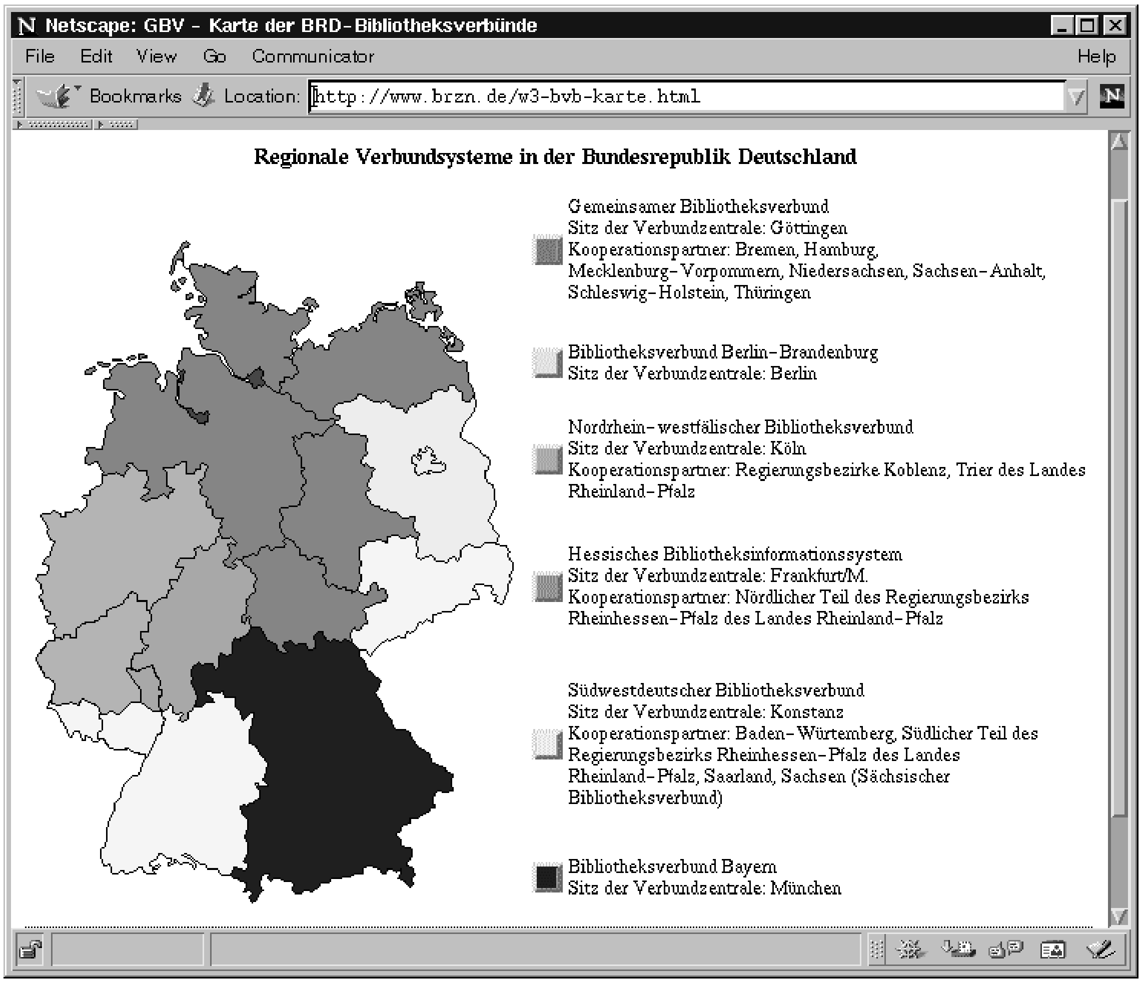


Abbildung 2.5: Karte der deutschen Bibliotheksverbünde [2]

In den siebziger Jahren bauten die Bibliotheken in Deutschland regionale Verbundsysteme auf. Durch eine arbeitsteilige Katalogisierung wollte man die internen Arbeitsabläufe optimieren. Computersysteme waren damals sehr teuer und aufwendig zu verwalten - zu teuer und zu aufwendig für eine einzelne Bibliothek. In den 80er Jahren wurden die Arbeitsabläufe weiter optimiert. Es wurden als Ergänzung zu den bestehenden Bibliotheksverbünden lokale Bibliothekssysteme angeschafft. Die Hard- und Software hatte sich weiterentwickelt, und es wurden Alternativen zu Großrechnern verfügbar. Die lokalen Bibliothekssysteme bieten Dienste für den

[2] Das Copyright für die Karte der deutschen Bibliotheksverbünde liegt beim Deutsches Bibliotheksinstitut (DBI) bzw. beim Gemeinsamen Bibliotheksverbund (GBV), [IVB96], [Ver99]).

Nutzer an (z.B. OPACs). Mit einigen Systemen kann man in entfernten Systemen recherchieren und Daten übertragen. In den 90er Jahren ist ein umfassender Zugriff auf lokale Ressourcen und externe Informationsquellen mit einer einheitlichen Oberfläche möglich. Computer sind wesentlich leistungsfähiger und billiger geworden. Hardware, Software und die Netzinfrastruktur sind standardisiert und an jedem Arbeitsplatz verfügbar ([Dug98], siehe auch [Kru94], [Abl97], [BIB99])

Der Aufbau regionaler Verbundsysteme wurde von den Bundesländern gefördert und finanziert. Später schlossen sich mehrere regionale Verbundsysteme über die Landesgrenzen hinweg zusammen und gründeten überregionale Bibliotheksverbünde. In Deutschland gibt es 6 überregionale Bibliotheksverbünde (siehe Abbildung 2.5).

Die Datenbank eines Bibliotheksverbundes enthält je nach Größe des Bibliotheksverbundes ca. 5-10 Millionen Titelsätze. Der Bestand einer mittelgroßen Universitätsbibliothek liegt bei mehreren hunderttausend Titelsätzen.

Die meisten Bibliotheksverbünde bieten inzwischen einen Zugang zu ihrem System über das Z39.50-Protokoll an. Auch die Deutsche Bibliothek und einige (Universitäts-)Bibliotheken ermöglichen einen Zugang zu ihren Systemen über Z39.50. Addiert man den Bestand der Bibliothekssysteme in Deutschland mit einem Z39.50-Server, so kann man mit *ZACK* in über 40 Millionen Datensätzen recherchieren.

Eine Liste der Bibliotheksverbünde und Bibliotheken mit Adressen, Ansprechpartnern, Größe des Bestandes sowie technischen Daten der Z39.50-Server ist im Anhang B Z39.50-Server (Seite 115) zu finden.

2.3 Existierende Systeme zur verteilten Suche

Bislang existieren im deutschsprachigen Raum nur Systeme, die eine verteilte Suche *ohne Dublettenkontrolle* anbieten. Nachfolgend werden drei bekannte Systeme kurz vorgestellt. Alle drei Systeme bieten eine verteilte Suche in deutschsprachigen Bibliothekssystemen über das World-Wide-Web an.

Karlsruher Virtueller Katalog (KVK)

Der Karlsruher Virtuelle Katalog (KVK) bietet eine WWW-Oberfläche für Bibliothekskataloge an. Die eingegebenen Suchanfragen werden an mehrere WWW-Bibliothekskataloge gleichzeitig weitergereicht und die jeweiligen Treffer gesammelt und angezeigt (siehe [KVK99]).

Eine Dublettenkontrolle findet nicht statt. Optional kann die Trefferliste *zusätzlich* sortiert werden. Gesucht werden kann nur in Datenbanken, die einen Zugang über das Web anbieten. Z39.50-Server können nicht direkt angesprochen werden. Den KVK gibt es seit 1996. Der KVK ist ein sehr erfolgreiches System und wird von vielen Benutzern täglich genutzt (mehr als 10.000 Suchanfragen pro Werktag). Gesucht werden kann in praktisch allen deutschsprachigen Bibliotheksverbünden (einschließlich Österreich und Schweiz) und in Buchhandelsverzeichnissen (Amazon, Verzeichnis lieferbarer Bücher). Der KVK wurde an der Universitätsbibliothek Karlsruhe in Zusammenarbeit mit der Fakultät für Informatik im Rahmen einer Studienarbeit entwickelt.

Die Deutsche Bibliothek

Die Deutsche Bibliothek bietet eine verteilte Suche in Z39.50-Servern über ihr Z39.50 Gateway ([DDB99a]) an. Gesucht werden kann in den Datenbanken der Deutschen Bibliothek, des Gemeinsamen Bibliotheksverbundes, des Bayrischen Bibliotheksverbundes und des

Südwestdeutschen Bibliotheksverbundes. Eine Dublettenkontrolle findet nicht statt. Die Ausgabe der Ergebnisse erfolgt nach Datenbank, zuerst die ersten 10 Treffer aus der Datenbank X, dann die ersten 10 Treffer aus der Datenbank Y usw. Für das Z39.50 Gateway der DDB ist ein moderner Web-Client erforderlich, der Frames (Rahmen) und JavaScript unterstützt ([Net99]). Das WWW-Z39.50-Gateway der Deutschen Bibliothek entstand im Rahmen des DBV-OSI Projektes (siehe Abschnitt 4.2 Vergleich und Auswahl vorhandener Software, Seite 17).

Bibliotheksverbund Bayern

Der Bibliotheksverbund Bayern bietet eine verteilte Suche ohne Dublettenkontrolle in den Datenbanken des Bayrischen Bibliotheksverbundes, der Deutschen Bibliothek, des Gemeinsamen Bibliotheksverbundes, des Südwestdeutschen Bibliotheksverbundes und der Universitätsbibliothek Augsburg an (siehe [BVB99]). Bei kleinen Treffermengen werden die Datensätze sortiert nach Titel und Autor ausgegeben. Bei vielen Treffern wird auf die Sortierung verzichtet, und die Ausgabe erfolgt nach den Datenbanken (siehe auch im Abschnitt zuvor bei der Deutschen Bibliothek).

Das WWW-Z39.50-Gateway des Bibliotheksverbundes Bayern entstand im Rahmen des DBV-OSI Projektes (siehe Abschnitt 4.2 Vergleich und Auswahl vorhandener Software, Seite 17). Es wurde von der Firma Harbinger GmbH in Karlsruhe entwickelt.

Kapitel 3

Modellierung von ZACK

ZACK ist ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50-Protokolls. Der Benutzer kann mit seinem Web-Client über das mit *ZACK* entwickelte WWW-Z39.50-Gateway in Bibliotheksdatenbanken recherchieren. In diesem Kapitel wird anhand des Client-Server-Modells beschrieben, wie die Suche in einer Bibliotheksdatenbank über das Internet erfolgt. Dabei wird kurz erklärt, wie die Kommunikation zwischen Client und Server abläuft und welche unterschiedlichen Protokolle in *ZACK* verwendet werden.

3.1 Client-Server-Modell

Zum Suchen in den bibliographischen Datenbanken wird das Client-Server-Modell verwendet. Der Benutzer (Client) stellt eine Anfrage an die Datenbank (Server), und der Server liefert die Ergebnisse zurück.

Client und Server kommunizieren über eine gemeinsame Sprache. Diese Sprache ist ein vordefiniertes Protokoll, mit dessen Hilfe sich Client und Server verständigen und Daten austauschen ([Tan95]).

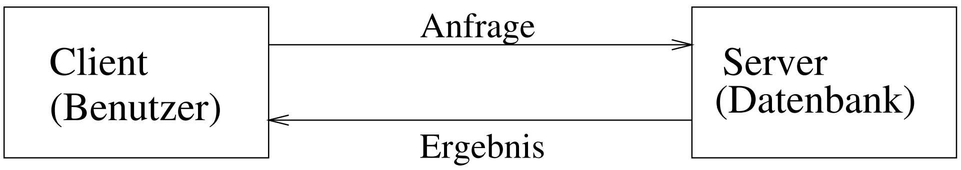


Abbildung 3.1: Client-Server-Kommunikation

Das Client-Server-Modell ermöglicht es, daß Client und Server auf unterschiedlichen Rechnern, verschiedener Hardware und verschiedenen Betriebssystemen laufen. Client und Server können räumlich voneinander getrennt sein.

3.2 Protokolle

Eines der bekanntesten Kommunikationsprotokolle ist das Hypertext-Transfer-Protokoll (HTTP). Es wird im World-Wide-Web (WWW) verwendet. Ein anderes Protokoll, das in *ZACK* verwendet wird, ist das Z39.50 Protokoll. Beide Protokolle werden im einzelnen vorgestellt und die Gemeinsamkeiten sowie die Unterschiede erklärt.

3.3 Das World Wide Web (WWW)

Web-Browser sind einfach zu bedienende Benutzeroberflächen für Web-Dokumente im Internet. Zum Datentransfer zwischen Web-Browsern (Web-Clients) und Web-Servern wird das Hypertext-Transfer-Protokoll (HTTP) verwendet, zur Darstellung der Hypertexte im Web-Browser das Dokumentenformat HTML (Hyper Text Markup Language).

HTTP ist ein Kommunikationsprotokoll für multimediale Informationssysteme. Es ist ein allgemeines und verbindungsloses Protokoll. Es wird im World-Wide-Web seit 1990 genutzt ([HTT99b], [HTT96]). Das HTTP-Protokoll ist zustandslos. Für jede Anfrage an den Web-Server wird eine neue Verbindung aufgebaut, das Ergebnis zurückgeliefert und danach die Verbindung abgebaut.

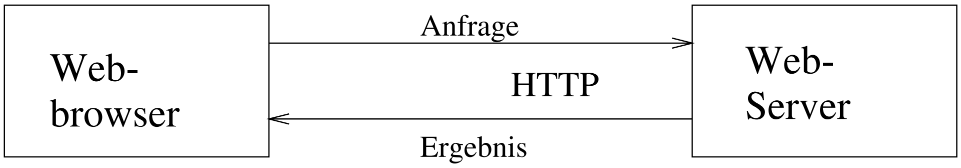


Abbildung 3.2: Client-Server-Kommunikation im WWW

HTML ist die *lingua franca* zum Publizieren von Hypertexten im World-Wide-Web. HTML ist ein Dokumentenformat. Es enthält sowohl logische Strukturen (Absätze, Listen) als auch Layout-Anweisungen (Farbe, Schriftgröße). HTML ist ein internationaler Standard, der vom World Wide Web Consortium entwickelt und gepflegt wird ([HTT99a]).

Es gibt viele Anbieter von Web-Clients und Web-Servern. Die Anwender haben die Wahl zwischen kommerziellen Anbietern sowie kostenlosen Produkten von Firmen und Forschungseinrichtungen. Web-Clients sind auf fast allen Rechnern verfügbar.

Web-Browser gehören zu den erfolgreichsten Anwendungen im Internet. Für die Softwareentwicklung stellt das Web eine enorme Arbeitserleichterung dar. Eine einmal manuell geschriebene oder mit einem Programm dynamisch erzeugte HTML-Seite kann von Benutzern weltweit gelesen werden. Der Softwareentwickler kann sich ganz auf die Entwicklung der Serverprogramme konzentrieren. Client, Server und die Protokolle wurden bereits von anderen Firmen entwickelt, gepflegt und an die Benutzer verteilt ([Apa99], [Net99]).

3.4 Z39.50

Z39.50 ist die Nummer einer ANSI-Norm. Es ist ein Protokoll zur Kommunikation zwischen bibliothekarischen Datenbanksystemen (Server) und Zugriffsprogrammen (Clients). Z39.50 erlaubt die Suche in heterogenen Datenbanken aus der gewohnten lokalen Programmumgebung. Die Verwendung des Z39.50 Protokolls führt zu einer Unabhängigkeit von der Datenbank, der lokalen Abfragesyntax, dem eingesetzten Betriebssystem und der Hardware. Man kann sich das Z39.50-Protokoll als ein Art Datenbank-Esperanto vorstellen, das jedem Client ermöglicht, mit jeder Datenbank einen Dialog zu führen (aus [TUB98b]).

Das Protokoll wurde Anfang 1996 der ISO zur Verabschiedung als internationaler Standard vorgelegt (ISO 23950). Die Wurzeln des Protokolls reichen zurück bis ins Jahr 1984. Die Versionen 1 und 2 des Standards wurden in den Jahren 1988 und 1992 als Z39.50-1988 bzw. Z39.50-1992 verabschiedet. Alle drei Protokollversionen definieren Z39.50 als OSI-Anwendungsprotokoll, d.h. als Protokoll der Ebene 7 des OSI-Modells. Ungeachtet dieser Tatsache wird Z39.50 in nahezu allen Realisierungen als Internet-Protokoll angewandt (aus [Her96], siehe auch [LOC99b]).

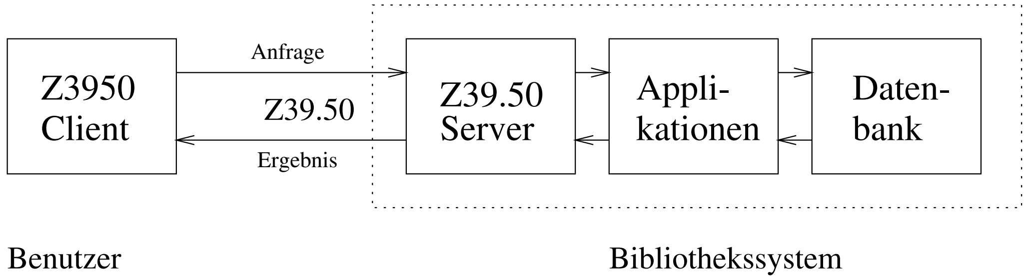


Abbildung 3.3: Client-Server-Kommunikation mit Z39.50

Das Z39.50 Protokoll ist verbindungsorientiert. Der Client baut eine stehende Verbindung zum Server auf und authentifiziert sich mit seinem Namen und Paßwort. Der Nutzer stellt seine Anfrage an den Z39.50-Server. Der Server leitet die Anfrage an das lokale Bibliothekssystem weiter und gibt die Anzahl der gefundenen Treffer zurück. Der Nutzer kann sich nun eine bestimmte Anzahl von Datensätzen - z.B. die ersten 10 - vom Server holen. Oder er stellt eine neue Suchanfrage.

Die Verbindung bleibt solange bestehen, bis der Nutzer sie beendet oder der Server sie wegen zu langer Inaktivität abbricht. Sämtliche Voreinstellungen des Benutzers wie die Länge der Datensätze, Zeichensatz, Nummer des zuletzt gelesenen Datensatzes, Name der Datenbank, Paßwort usw. bleiben während einer Sitzung erhalten.

Das Z39.50 Protokoll ist umfangreich ([Got96]). Die meisten kommerziellen Anbieter beschränken sich deshalb bei der Entwicklung ihrer Software auf einen Teil (Core) des Protokolls ([LOC99b], [BAT99]). Es gibt nur wenige Anbieter von Z39.50-Clients und Z39.50-Servern (siehe Kapitel 4.2, Seite 17). Fast alle Produkte sind kostenpflichtig. Die Z39.50-Clients sind nur für wenige Betriebssysteme (Windows95, Windows NT) verfügbar.

Weitere Literatur zu Z39.50 ist in [PR97], [Lyn97], [CLI96], [Z3995], [LOC99b], [FAQ99], [NIS97], [Zso99], [Zpr99], [Zte99b], [FW97], [KR95], [ZRE96], [BAT99], [DCZ98], [Den96], [Intil], [Gal98], [BOP99], [TUB98b], [Boo99a], [Boo99b], [MARil], [MAB99], [UNI98], [BIB95] und [MAR99] zu finden.

3.5 Das WWW und bibliographische Datenbanken

Wie verbindet man jetzt das WWW mit Z39.50? Web-Clients sind überall verbreitet und einfach zu bedienen, Z39.50-Clients sind es nicht. Z39.50-Server bieten einen universellen Zugang zu bibliothekarischen Datenbanksystemen an, den man nur mit einem Z39.50-Client nutzen kann. Die Benutzer aber wollen mit ihrem Web-Browser auf die Datenbanken zugreifen, ohne vorher neue Software zu installieren oder neue Befehle zu erlernen.

Web-Clients sind nur bedingt zur Suche in Datenbanken geeignet, da die Verbindung zwischen Client und Server zustandslos ist. Für jede Anfrage an den Web-Server wird eine neue Verbindung aufgebaut, das Ergebnis zurückgeliefert und danach die Verbindung abgebaut. Der Web-Server hat keine Information darüber, welche Anfragen der Web-Client (Benutzer) zuvor bereits gestellt hat. Dies ist in diesem Fall ein Nachteil, da sich Suchanfragen häufig auf vorherige Anfragen beziehen, zum Beispiel mit dem Befehl "*gib die nächsten 10 Treffer aus*".

Dieses Dilemma wird mit einem WWW-Z39.50-Gateway gelöst. Das WWW-Z39.50-Gateway setzt die zustandslose Kommunikation zwischen Web-Client und Web-Server in eine verbindungsorientierte Kommunikation zur Datenbank um (Abbildungen 3.4, 3.5 und 3.6).

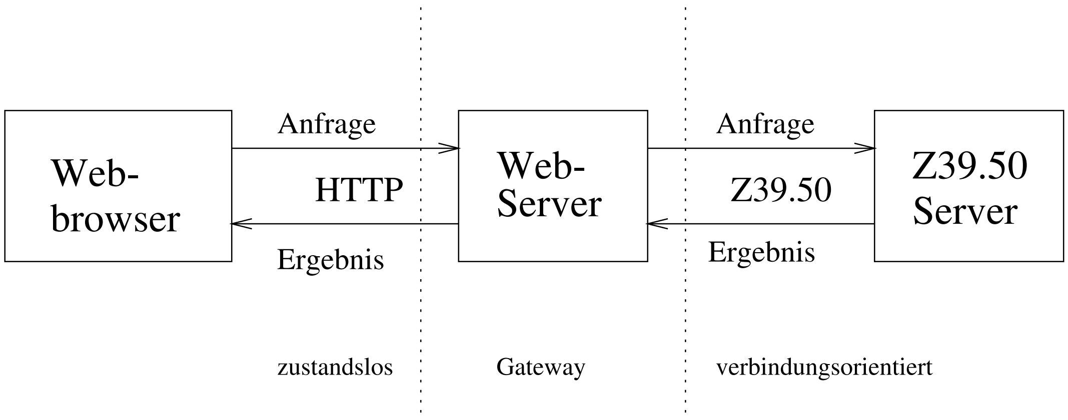


Abbildung 3.4: Client-Server-Kommunikation im WWW-Z39.50-Gateway

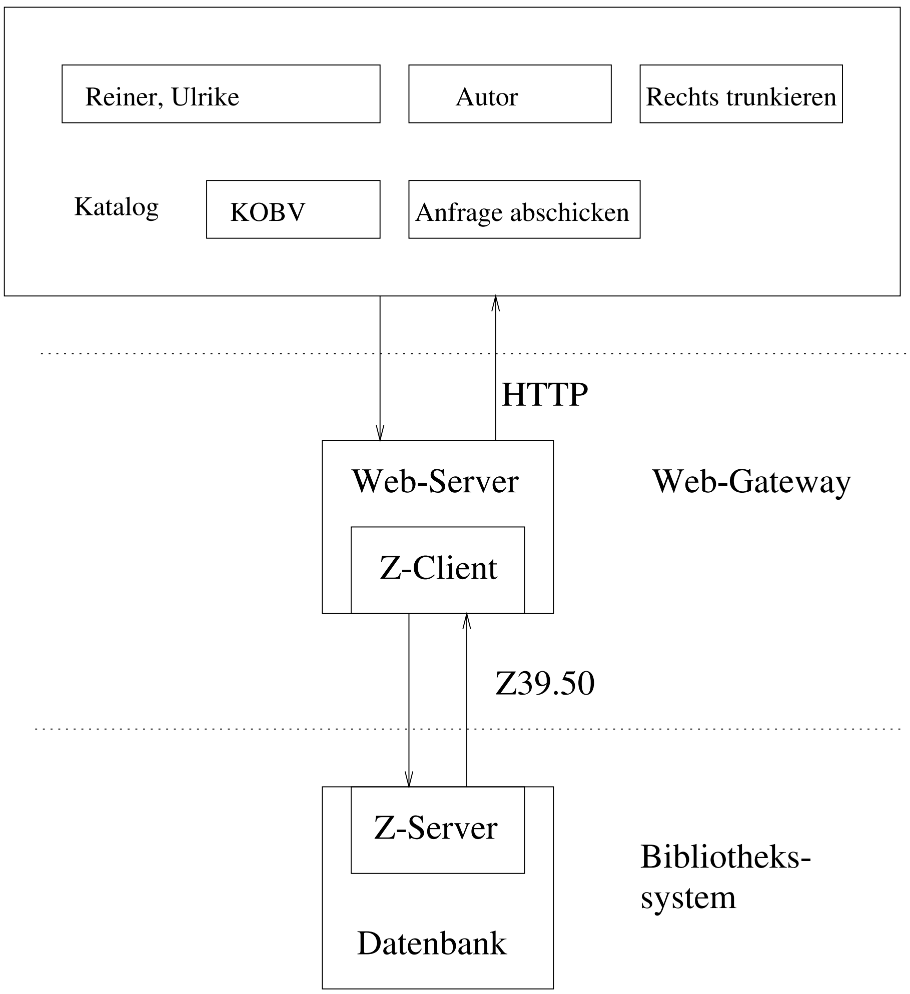


Abbildung 3.5: Oberfläche WWW-Z39.50-Gateway

Der Web-Client kommuniziert mit dem Web-Server. Die Datenbank selbst kann der Web-Client nicht direkt ansprechen - dies übernimmt das WWW-Z39.50-Gateway (siehe Abbildung 3.6).

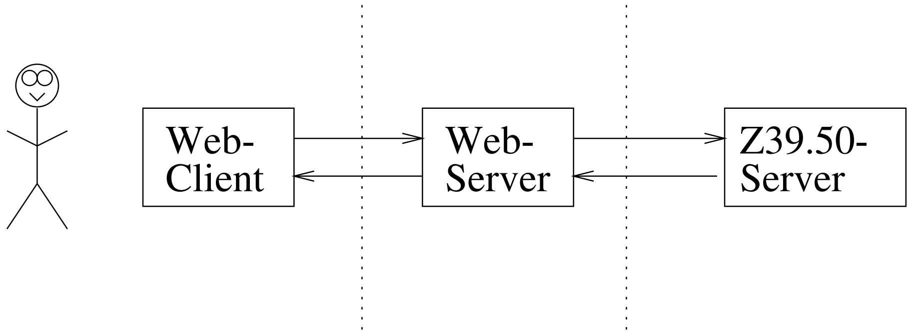


Abbildung 3.6: WWW-Z39.50-Gateway

Die Kommunikation zwischen Web-Client und dem WWW-Z39.50 Gateway gestaltet sich wie folgt: Der Benutzer startet seinen Web-Client (Browser) und gibt die Adresse des Web-Servers ein. Der Web-Server schickt eine Suchmaske (eine HTML-Seite) an den Browser zurück. Der Benutzer gibt in der Suchmaske die Werte für die gewünschten Suche ein, zum Beispiel *Autor=salton*, und schickt die Anfrage an den Web-Server. Der Web-Server bearbeitet die Anfrage des Clients. Er wandelt sie in eine Z39.50-Anfrage um und schickt sie an den Z39.50-Server. In der bibliographischen Datenbank wird jetzt nach den gewünschten Datensätzen gesucht, in diesem Fall nach Büchern des Autors *"Salton"*. Als Ergebnis werden die Datensätze dem Web-Server zurückgeliefert. Der Web-Server wandelt die Ergebnisse der Datenbank in das HTML-Format um und schickt sie an den Web-Client, der die Ergebnisse dem Benutzer präsentiert.

3.6 Modell verteilte Suche

Bei der verteilten Suche wird parallel in mehreren Datenbanken gesucht (Abbildungen 3.7, 3.8 und 3.7). Der Benutzer wählt die gewünschten Datenbanken aus und schickt seine Anfrage an das WWW-Z39.50-Gateway. Das Gateway bearbeitet die Anfrage, wandelt sie in eine Z39.50-Anfrage um und schickt sie *gleichzeitig* an die betreffenden Z39.50-Server. Die in den Datenbanken gefundenen Datensätze werden an den Web-Server zurückgeliefert. Das Gateway startet eine Dublettenkontrolle und erkennt gleiche Datensätze. Dem Benutzer wird eine übersichtliche Kurztrefferliste ohne doppelte Einträge angeboten.

Die verteilte Suche hat in der Praxis eine deutlich bessere Trefferquote gebracht als die Suche in nur einer Datenbank. Dabei bleibt die Antwortzeit in einem für die Benutzer akzeptablen Rahmen.

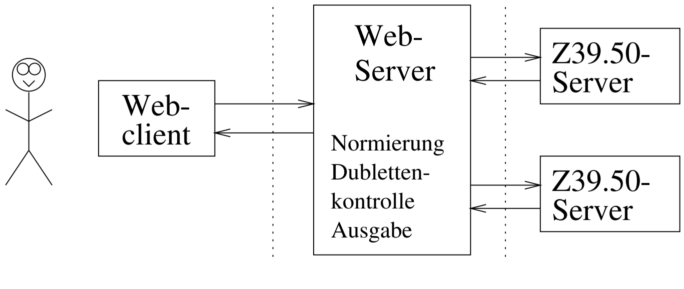


Abbildung 3.7: Modell verteilte Suche im WWW-Z39.50-Gateway

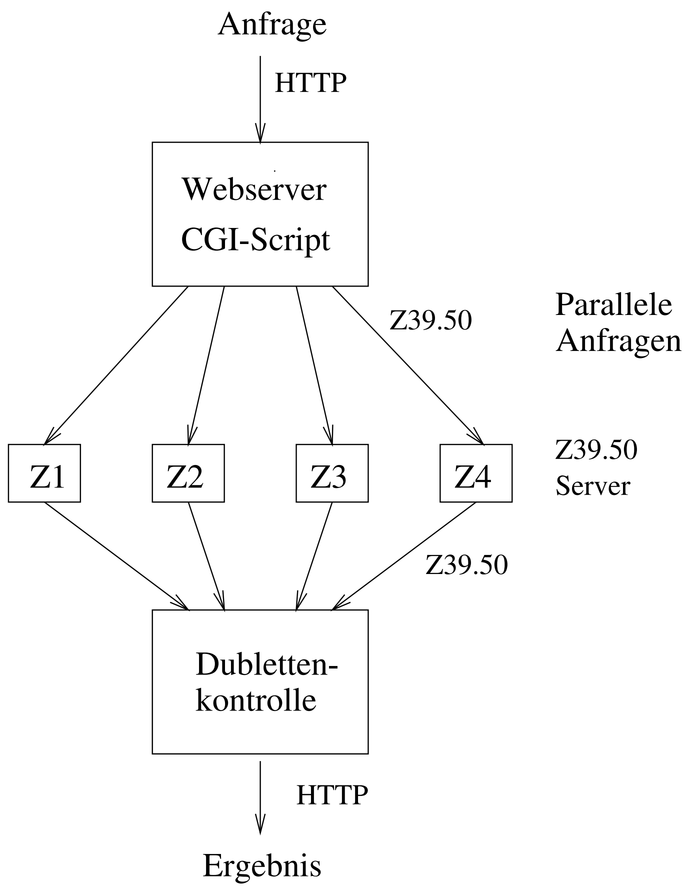


Abbildung 3.8: Datenfluß verteilte Suche im WWW-Z39.50-Gateway

3.7 Zusammenfassung

Das mit *ZACK* entwickelte WWW-Z39.50-Gateway hat in der Praxis das gewünschte Ergebnis gebracht: die Nutzer können mit ihrem gewohnten Web-Browser auf die Datenbanken zugreifen. Sie brauchen dafür keine neue (kostenpflichtige) Software zu installieren und müssen nicht den Umgang mit neuer Software erlernen und üben.

Kapitel 4

Implementierung von ZACK

In diesem Kapitel wird das im Rahmen dieser Diplomarbeit entwickelte System *ZACK* vorgestellt. Es wird beschrieben, in welchen Schritten *ZACK* entworfen und implementiert wurde. Es wird dargelegt, welche vorhandene Software für die Entwicklung dieses Systems in Betracht kam und und welche in *ZACK* verwendet wird. *ZACK* wurde in zwei Schritten entwickelt: mit dem ersten System kann man nur in einer Datenbank suchen, und mit dem zweiten System kann man in mehreren Datenbanken parallel mit Dublettenkontrolle suchen.

In Abschnitt 4.1 **Benutzeroberfläche** wird beschrieben, welche Funktionalität *ZACK* bietet, wie die Benutzerführung und wie die Suchmasken gestaltet wurden.

In Abschnitt 4.2 **Verwandte Software** wird dargelegt, welche Standardsoftware für *ZACK* in Betracht kommt und welche verwendet wird.

In Abschnitt 4.3 **Performance** wird untersucht, wie schnell die Z39.50-Server in Deutschland sind und wieviele Datensätze pro Sekunde von einer Datenbank übernommen werden können.

In Abschnitt 4.4 **Erstes System** wird der Recherche-Client für Z39.50-Datenbanken beschrieben. Mit dem Recherche-Client kann man in einem Bibliothekssystem recherchieren und Daten übernehmen.

In Abschnitt 4.5 **Zweites System: Parallele Suche in mehreren Z39.50-Datenbanken** wird beschrieben, wie man parallel in mehreren Z39.50-Datenbanken recherchieren kann und wie dem Benutzer die Ergebnisse der Dublettenkontrolle präsentiert werden.

Für die Entwickung von *ZACK* waren theoretische Vorüberlegungen und umfangreiche Tests erforderlich. Diese werden im Anschluß an dieses Kapitel in dem Kapitel 5 **Normierung** (Seite 40), Kapitel 6 **Dublettenkontrolle** (Seite 60) und Kapitel 7 **Ausgabe von Dubletten** (Seite 78), Kapitel 8 **Praktische Ergebnisse** einer verteilten Suche (Seite 84) sowie Anhang A **Analyse der MAB2-Datensätze der Deutschen Bibliothek** (Seite 102) beschrieben.

4.1 Benutzeroberfläche von ZACK

Die ersten Fragen, die man sich beim Design von Oberflächen stellt, lauten: Wer sind die Nutzer? Was sind ihre Aufgaben? Welches Vorwissen haben diese Nutzer? Zum Beispiel wird sich eine Suchmaske für Patienten inhaltlich, in Stil und Sprache und in der Detailliertheit von einer für Ärzte stark unterscheiden (siehe auch [Shn97b], [Shn97a]).

Die Zielgruppe von *ZACK* sind Bibliothekare und erfahrene Benutzer. Diese Nutzergruppe hat Erfahrung bei der Recherche in Datenbanken und mit Web-OPACs. Für sie steht die Funktionalität des Systems im Vordergrund. Sie wird das System regelmäßig nutzen und bald wissen, wie das System zu bedienen ist.

Bei der Gestaltung der Oberfläche von *ZACK* werden diese Anforderungen berücksichtigt. Die Nutzer sollen mit wenigen Handgriffen ihr Ziel erreichen und bestmögliche Ergebnisse bei der Recherche erzielen.

Mit der Titelsuche kann man nach Werken in einer Bibliothek suchen. Hier sind bis zu drei Boolesche Verknüpfungen möglich (*UND*, *ODER*, *NICHT*) (siehe Abbildung 4.7, Seite 29). Man kann sich bis zu 1.000 Treffer anzeigen lassen. In der Treffermenge kann navigiert werden (nächste 10 Treffer, vorherige 10 Treffer; siehe Abbildung 4.4, Seite 25).

Mit der Registersuche kann man im Bestand einer Bibliothek blättern - vergleichbar mit der Suche in einem Zettelkatalog (siehe Abbildung 4.8, Seite 30).

ZACK unterstützt unterschiedliche Formate (MAB2, USMARC, UNIMARC) und Zeichensätze (ANSEL, ISO8859-1, ISO5426). Es können mehrere Datensätz zugleich in die lokale Datenbank übernommen werden (siehe Abbildung 4.5, Seite 26).

ZACK kann von jedem Computer aus benutzt werden, gleich welche Hardware, welches Betriebssystem oder graphische Oberfläche benutzt wird (Windows 3.11, X11, VT100 Terminals). Es wurde auf die Verwendung von JavaScript, Java und Frames bewußt verzichtet, da diese Weiterentwickungen des WWW nicht auf allen Rechnern verfügbar sind.

Einige Datenbanken sind kostenpflichtig bzw. nur für bestimmte Nutzer freigegeben. Für *ZACK* wird deshalb eine einfache Zugriffskontrolle entwickelt. Für jeden Nutzer kann anhand seiner Absenderadresse (IP-Adresse) eingestellt werden, auf welche Datenbanken er Zugriff hat und ob er die Datensätze im Kategorienformat in seine eigene Datenbank übernehmen darf.

ZACK ist zweisprachig. Der Benutzer kann zwischen einer deutschen oder englischen Oberfläche wählen (siehe im Anhang Abbildung C.2, Seite 131 und Abbildung C.5, Seite 133).

4.2 Verwandte Software

Z39.50-Software

Für *ZACK* wird eine Z39.50-Software benötigt, die im Batch-Modus [1] arbeiten kann. Ein Z39.50-Client mit graphischer Oberfläche ist nicht geeignet, da die Ergebnisse der Recherche vor der Ausgabe weiterverabeitet werden müssen (Normierung, Dublettenkontrolle). In die engere Wahl fielen das YAZ-Toolkit der Firma Index Data ([Ind99]) aus Dänemark und der DBV-OSI Client aus dem DBV-OSI Projekt ([DBV99]).

Das YAZ-Toolkit wird von der Firma Index Data kostenlos zur Verfügung gestellt. Jeder darf das Toolkit benutzen, den Quellcode verändern und den eigenen Bedürfnissen anpassen. Eine Lizenzvereinbarung oder Lizenzgebühren sind nicht erforderlich. YAZ wird von Index Data gepflegt und weiterentwickelt. Bei Fragen und Problemen kann man sich per E-Mail an die Entwickler wenden und erhält kostenlos Hilfe und Tips. YAZ ist ein kleines und übersichtliches Softwarepaket. Es ist portabel und läuft auf verschiedenen Betriebssystemen (Unix, Windows). Viele kommerzielle Z39.50-Clients ([Zna97], [Boo99a]) benutzen das YAZ-Toolkit von Index Data.

Das DBV-OSI Projekt (Deutscher Bibliothekenverbund - Open Systems Interconnection) ist ein vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie und der Deutschen Forschungsgemeinschaft (DFG) gefördertes Projekt. Das Projekt DBV-OSI begann im Jahr 1993 und wurde 1997 beendet ([DBV99]). Im Rahmen dieses Projektes wurde ein Z39.50-Server und ein Z39.50-Client entwickelt. Die DBV-OSI Software ist über FTP frei verfügbar. Die Software ist umfangreicher und komplexer als das YAZ Toolkit von Index Data.

ZACK verwendet das YAZ-Toolkit, da es die gewünschte Funktionalität bietet und weiterhin von den Entwicklern unterstützt wird.

[1] Im Batch-Modus werden Anweisungen nacheinander automatisch ausgeführt, ohne Interaktion mit dem Benutzer.

Eine Übersicht über Z39.50-Client mit graphischer Oberfläche (GUI) ist in [CLI96] zu finden. Die bekanntesten (kommerziellen) Z39.50-Client sind der ZNavigator [Zna97] und BookWhere 2000 [Boo99a].

Web-Server

Für das WWW-Z39.50-Gateway wird ein Web-Server benötigt. Das Apache-Projekt entwickelt einen robusten, voll funktionsfähigen Web-Server, der kostenlos und im Quellcode verfügbar ist ([Apa99]). Der Apache-Web-Server hat sich in der Praxis bewährt und gehört zu den am weitesten verbreiteten Web-Servern im Internet. *ZACK* verwendet deshalb den Apache-Server.

Die Programmiersprache Perl

ZACK ist in der Computersprache Perl5 geschrieben ([WCS96], [Per99]). Mit Perl5 kann man schnell einen Protoypen entwickeln und testen. Für Perl5 gibt es eine umfangreiche Softwarebibliothek, insbesondere zur Programmierung von CGI-Scripten.

Die für *ZACK* geschriebene Software wird im Anhang C **Kurzbeschreibung der Software** auf den Seiten 124 vorgestellt.

4.3 Performance

Einer der wichtigsten Punkte bei der verteilten Suche ist die Antwortzeit. Wie lange muß der Benutzer auf seine Antwort warten? Woran liegt es, daß es solange dauert? Kann das System *ZACK* so optimiert werden, daß die Benutzer nicht unnötig lange warten müssen? Wieviele Benutzer können *ZACK* gleichzeitig benutzen?

ZACK wird von den Nutzern nur angenommen werden, wenn die Antwortzeiten in einem akzeptablen Rahmen bleiben. *ZACK* ist ein interaktives System, und die Ergebnisse einer verteilten Suche sollten innerhalb von 10 Sekunden vorliegen.

In den folgenden Überlegungen wird unterschieden zwischen Benutzer- und Serversicht. Erstere spiegelt dabei die Sicht des Benutzers wieder, während letztere die Sicht des Systembetreibers beschreibt.

Performance aus Benutzersicht

Der Zeitaufwand für den Benutzer bei der Recherche setzt sich aus mehreren Teilschritten zusammen, die nacheinander bearbeitet werden:

1. Der Benutzer schickt die Anfrage über das Internet an den Web-Server ab.
2. Der Web-Server nimmt Anfrage an, wandelt sie in eine Z39.50-Anfrage um und schickt sie an den Z39.50-Server. Der Z39.50-Server sucht in der Datenbank und liefert die Ergebnisse an den Web-Server zurück. Der Web-Server wandelt die Datensätze in eine benutzerfreundliche HTML-Darstellung um.
3. Der Web-Server schickt die HTML-Seite an den Web-Client über das Internet.
4. Der Web-Client baut die HTML-Seite auf.

(Siehe auch das Kapitel 3 **Modellierung** von *ZACK*, Seite 10).

Jeder dieser Teile addiert sich zur gesamten Antwortzeit. Mögliche Engpässe aus Benutzersicht sind:

- Ein langsamer Netzanschluß (28.8 Kbit/s Modem).
- Ein überlastetes Internet.
- Ein langsamer lokaler Rechner (alter PC).
- Es werden zu viele Daten zurückgeliefert, und der Nutzer muß warten, bis die vom Web-Server gelieferte HTML-Seite vollständig da ist.

ZACK liefert nicht mehr Daten zurück als zur Darstellung der Ergebnisse unbedingt notwendig sind. *ZACK* kann problemlos mit einem 28.8Kbit/s Modem (entspricht ca. 3,3KByte/s) genutzt werden. Die HTML-Seiten sind so gestaltet, daß die HTML-Tags und die Verweise (Links) auf andere Seiten möglichst wenig Speicherplatz verbrauchen. Die HTML-Seiten können mit einem moderen Web-Client inkrementell gelesen werden, d.h. sobald ein Teil der HTML-Seite (z.B. das erste Drittel) beim Web-Client angekommen ist, kann die Oberfläche aufgebaut werden. Der Rest der HTML-Seite wird nach und nach aufgebaut. Der Benutzer muß also nicht warten, bis alle Daten da sind, sondern er erhält schon nach ein bis zwei Sekunden die ersten Ergebnisse. Eine HTML-Seite mit Kurztrefferliste ist ca. 7-14 KByte lang, je nach Anzahl der Treffer.

Performance aus Serversicht

Die Serversicht spiegelt die Sicht des Systembetreibers wider. Sie ist unabhängig von speziellen Benutzerstandpunkten (siehe [Fuh97]). Den Systembetreiber interessiert vor allem, ob die gewählte Computertechnik (Hard- und Software, Netzanbindung) der erwarteten Last durch die Nutzer standhält.

Unter dem Gesichtspunkt der Performance stellt sich der Systembetreiber die folgenden Fragen:

- Wieviele Anfragen pro Sekunde können die angesprochenen Datenbanken verarbeiten?
- Wieviele Anfragen über das Web-Gateway kann das eigene System verarbeiten?
- Wieviele gleichzeitige Z39.50-Verbindungen können auf dem eigenen Server laufen?
- Wieviele Benutzer können gleichzeitig mit dem eigenen System arbeiten, ohne daß sich die Antwortzeiten spürbar ändern?
- Wieviel Bandbreite braucht das System mindestens bei voller Belastung, damit der Netzanschluß des Servers nicht der Engpaß wird?
- Sind die Hardwarekomponenten des Servers gut aufeinander abgestimmt? Sind genug CPUs, RAM und schnelle Festplatten vorhanden? Ist bei voller Belastung genug RAM verfügbar, so daß sich die laufenden Prozesse nicht gegenseitig den Hauptspeicher wegnehmen?

Bei *ZACK* treten keine Engpässe auf dem Server auf. Der Rechner se2 [2] hat zwei schnelle CPUs und ausreichend Hauptspeicher. Das Konrad-Zuse-Zentrum ist mit 155MBit/s an das Internet angeschlossen, so daß der Netzanschluß des Servers gesichert ist.

ZACK ist kein öffentlicher Service. *ZACK* wird von Bibliothekaren der Europa-Universität Viadrina Frankfurt (Oder) und der Brandenburgischen Technischen Universität Cottbus genutzt. Die Anzahl der Zugriffe ist deshalb überschaubar (siehe Anhang D Zugriffsstatistik WWW-Z39.50-Gateway, Seite 136). Es gibt nur wenige gleichzeitige Verbindungen - die Belastung des eigenen Web-Servers ist gering.

[2]Eine UltraSPARC-II mit 336 MHz, siehe Abkürzungsverzeichnis

Performance Z39.50-Server in Deutschland

Ein möglicher Engpaß bei der Recherche ist die Geschwindigkeit, mit der die Z39.50-Server die Datensätze liefern. In dem nachfolgenden Test wird untersucht, wieviele Datensätze pro Sekunde aus einer Bibliotheksdatenbank übernommen werden können. Ziel ist es festzustellen, wie schnell die Z39.50-Server die Datensätze liefern und wieviele Daten man innerhalb einer bestimmten Zeitspanne von ca. 10 Sekunden maximal erhält. Die Benutzer sind nicht bereit, zu lange auf die Ergebnisse zu warten. Es muß daher darauf geachtet werden, daß die Ergebnisse innerhalb einer vorgegebenen Zeitspanne eintreffen und dann die Dublettenkontrolle gestartet werden kann.

In diesem Test werden die Z39.50-Server der Deutschen Bibliothek (DDB), des Bibliotheksverbundes Bayern (BVB), des Kooperativen Bibliotheksverbundes Berlin-Brandenburg (KOBV) und der Technischen Universität Braunschweig (TUBS) untersucht. Der GBV war bei diesem Test nicht ansprechbar. Es wird nach dem Titel "*Rostock*" gesucht, ohne Trunkierung (d.h. "*Rostocker*" wird nicht gefunden). Danach wird eine vorgegebene Anzahl von Datensätzen im Kurzformat (Brief) geholt.

Die Datensätze sind je nach Datenbank durchschnittlich 350-600 Bytes groß. Die gemessene Zeit umfaßt den Verbindungsaufbau (init), die Suche (search) und die Anzeige der Ergebnisse (present).

Der KOBV und die TUBS sind unbelastete Testserver. BVB und DDB sind Server im Produktionsbetrieb. Die Tests wurden an einem Donnerstag um 15 Uhr durchgeführt. Zum Testen wurde der Z39.50-Client aus dem YAZ-Toolkit verwendet (siehe Abschnitt 4.2, [YAZ99]).

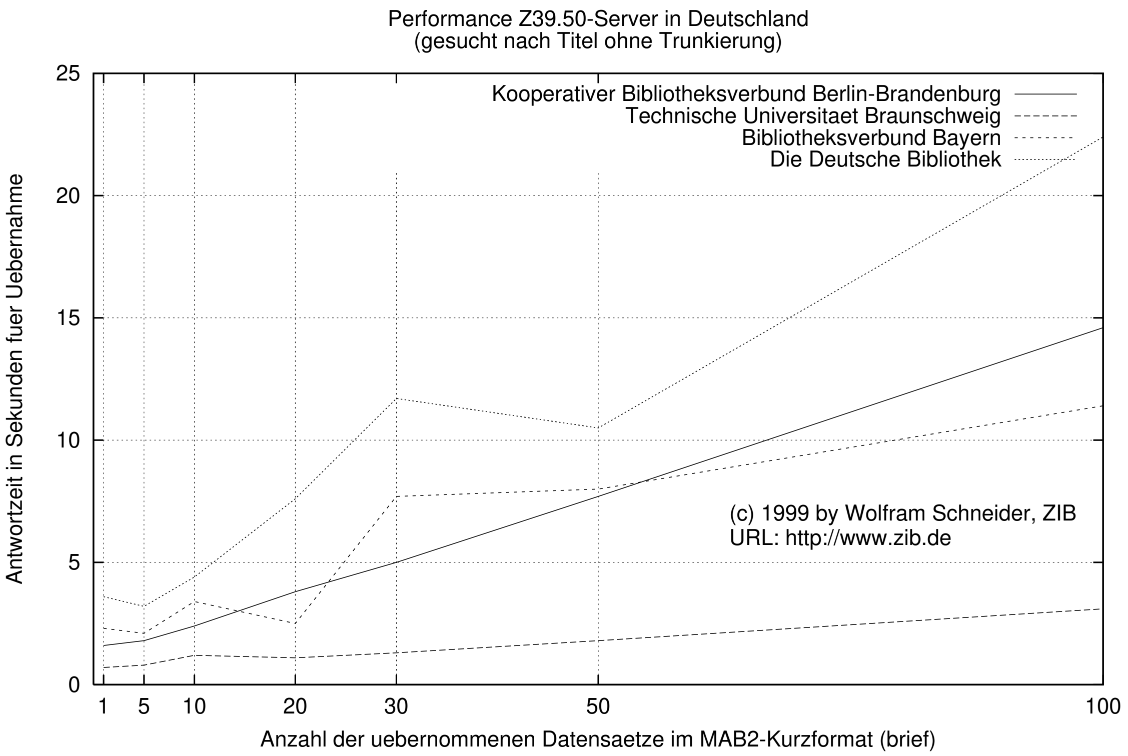


Abbildung 4.1: Performance Z39.50-Server in Deutschland, Datenübernahme

Anzahl der Datensätze	KOBV in sec	TUBS in sec	BVB in sec	DDB in sec
1	1,6	0,7	2,3	3,6
5	1,8	0,8	2,1	3,2
10	2,4	1,2	3,4	4,4
20	3,8	1,1	2,5	7,6
30	5,0	1,3	7,7	11,7
50	7,7	1,8	8,0	10,5
100	14,6	3,1	11,4	22,4

Tabelle 4.1: Performance Z39.50 Server in Deutschland, Datenübernahme

Je nach Datenbank werden 5-10 Datensätze pro Sekunde zurückgeliefert. 50 Datensätze erhält man innerhalb von 10 Sekunden. Die Messergebnisse pro Datenbank schwanken stark - was eben 10 Sekunden gedauert hat, kann beim nächsten Test mit den gleichen Eingabewerten 20 Sekunden oder länger dauern.

Das System allegro der Technischen Universität Braunschweig ist am schnellsten. Es liefert 100 Datensätze in 3 Sekunden. Es ist nicht klar, warum die Antwortzeiten innerhalb einer Datenbank so stark schwanken. Durch eine bessere Konfiguration und Hardwareausstattung auf Seiten der Anbieter lassen sich die Antwortzeiten der Z39.50-Server sicherlich noch verbessern.

Sucht man in 4 Datenbanken parallel, so kann man innerhalb von 10 Sekunden 200 Datensätze holen und danach auf Dubletten untersuchen.

Der Test ist nicht repräsentativ. Er wurde nur an einem Tag mit einer Anfrage durchgeführt. Für einen umfangreichen Test muß man die Anfragen an mehreren Tagen, zu unterschiedlichen Tageszeiten und mit unterschiedlichen Anfragen (Titel und Autor) durchführen.

Die Hersteller von Bibliothekssoftware führen eigene Performancemessungen ihrer Systeme durch. Weitere Literatur dazu ist in [HOR97] und [CBuJKW95] zu finden.

Performance von ZACK

In diesem Abschnitt wird kurz beschrieben, wie schnell *ZACK* bei der Recherche ist. Es wird untersucht, wie lange der Aufruf der CGI-Scripte (Suchmaske) dauert und wielange der Benutzer durchschnittlich bei der Suche in einer Datenbank und bei der verteilten Suche auf die Ergebnisse wartet.

Starten der CGI-Scripte: Der Start der CGI-Scripte dauert auf dem Rechner se2 [3] ca. 0,3 Sekunden (siehe auch Anhang C.2, Seite 125).

Suche in einer Datenbank: Die Suche nach einem Autor über das WWW-Z39.50-Gateway in der Technischen Universität Braunschweig - dem schnellsten bekannten Z39.50-Server - dauert ca. eine Sekunde bei wenigen Treffern (<3) und bis zu 3 Sekunden bei 10 und mehr Treffern. Bei der Deutschen Bibliothek wartet man bei wenigen Treffern 3 Sekunden und bei 10 Treffern ca. 4-5 Sekunden (zur Suche in einer Datenbank, siehe Kapitel 4.4, Seite 23).

Verteilte Suche: Es wird die ISBN-Nummer *3-928861-23-9* parallel in den Datenbanken BVB, bac, DDB und GBV gesucht. Insgesamt gibt es 3 Treffer. Es dauert ca. 4 Sekunden, bis die Datensätze von allen Datenbanken geliefert werden. Die Dublettenkontrolle benötigt weniger als 1 Sekunde. Insgesamt wartet der Benutzer ca. 5 Sekunden (siehe Abbildung 4.13 verteilte Suche nach ISBN-Nummer, Seite 35).

[3] Eine UltraSPARC-II mit 336 MHz, siehe Abkürzungsverzeichnis.

Zweites Beispiel: Es wird nach dem Autor "*Dalitz, Wolfgang*" in 6 Datenbanken gesucht: TUBS (3 Treffer), BVB (10 Treffer), bac (0 Treffer), DDB (7 Treffer), FH Potsdam (4 Treffer) GBV (11 Treffer). Insgesamt gibt es 35 Treffer. Es dauert ca. 7 Sekunden, bis die Datensätze von allen Datenbanken geliefert werden. Die Dublettenkontrolle dauert ca. 0,5 Sekunden. Insgesamt wartet der Benutzer ca. 8 Sekunden auf die Kurztrefferliste (siehe auch das Kapitel 4.14 Verteilte Suche nach Autor und Titel, Seite 37).

Zusammenfassung

Kurze Antwortzeiten sind möglich. Der Flaschenhals sind die Z39.50-Server, die die Daten nicht schnell genug liefern können, und der Zeitaufwand für den Verbindungsaufbau. Die Dublettenkontrolle ist für kleine und mittlere Datenmengen mit weniger als 100 Datensätze sehr schnell (siehe Kapitel 6.5 Effizienz der Dublettenkontrolle, Seite 73).

4.4 Erstes ZACK-System: Recherche-Client für Z39.50-Datenbanken

In diesem Abschnitt wird der Recherche-Client für Z39.50-Datenbanken vorgestellt. Mit dem Recherche-Client kann man in einem Bibliothekssystem recherchieren und Daten übernehmen (Siehe auch Kapitel Modellierung, Seite 10).

4.4.1 Einstiegsseite ZACK erstes System

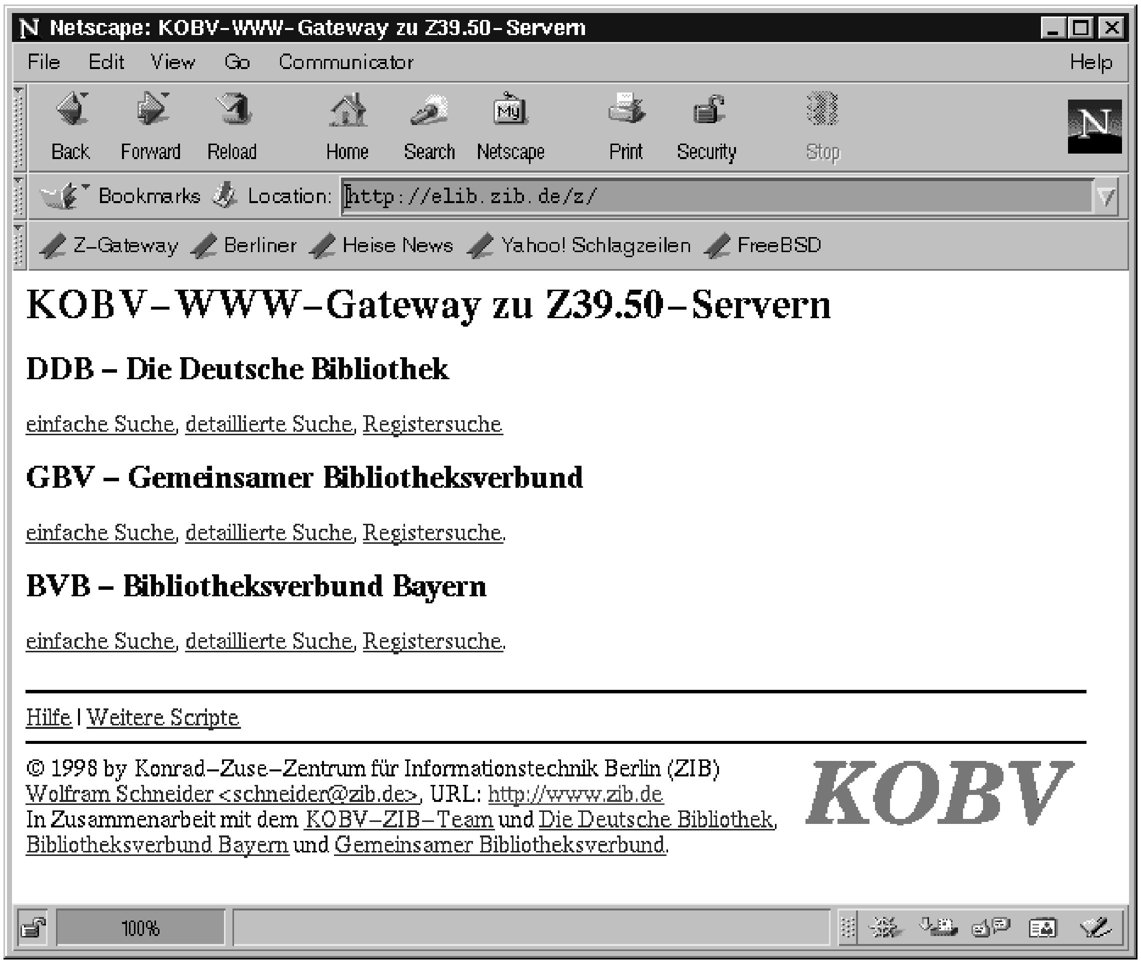


Abbildung 4.2: Einstiegsseite WWW-Z39.50-Gateway, erstes *ZACK*-System

Die Abbildung zeigt die Einstiegsseite des ersten *ZACK*-Systems. Zur Auswahl stehen die Datenbanken der Deutschen Bibliothek, des Gemeinsamen Bibliotheksverbundes der Länder Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen, Sachsen-Anhalt, Schleswig-Holstein und Thüringen, sowie der des Bibliotheksverbundes Bayern. Bei der *einfachen Suche* kann nur nach einem Attribut (z.B. Autor) gesucht werden. Bei der *detaillierten Suche* können mehrere Attribute miteinander verknüpft werden (z.B. Autor *und* Titel). Mit der Registersuche kann man im Bestand einer Bibliothek blättern - vergleichbar mit der Suche in einem Zettelkatalog.

4.4.2 Einfache Suche nach Autor

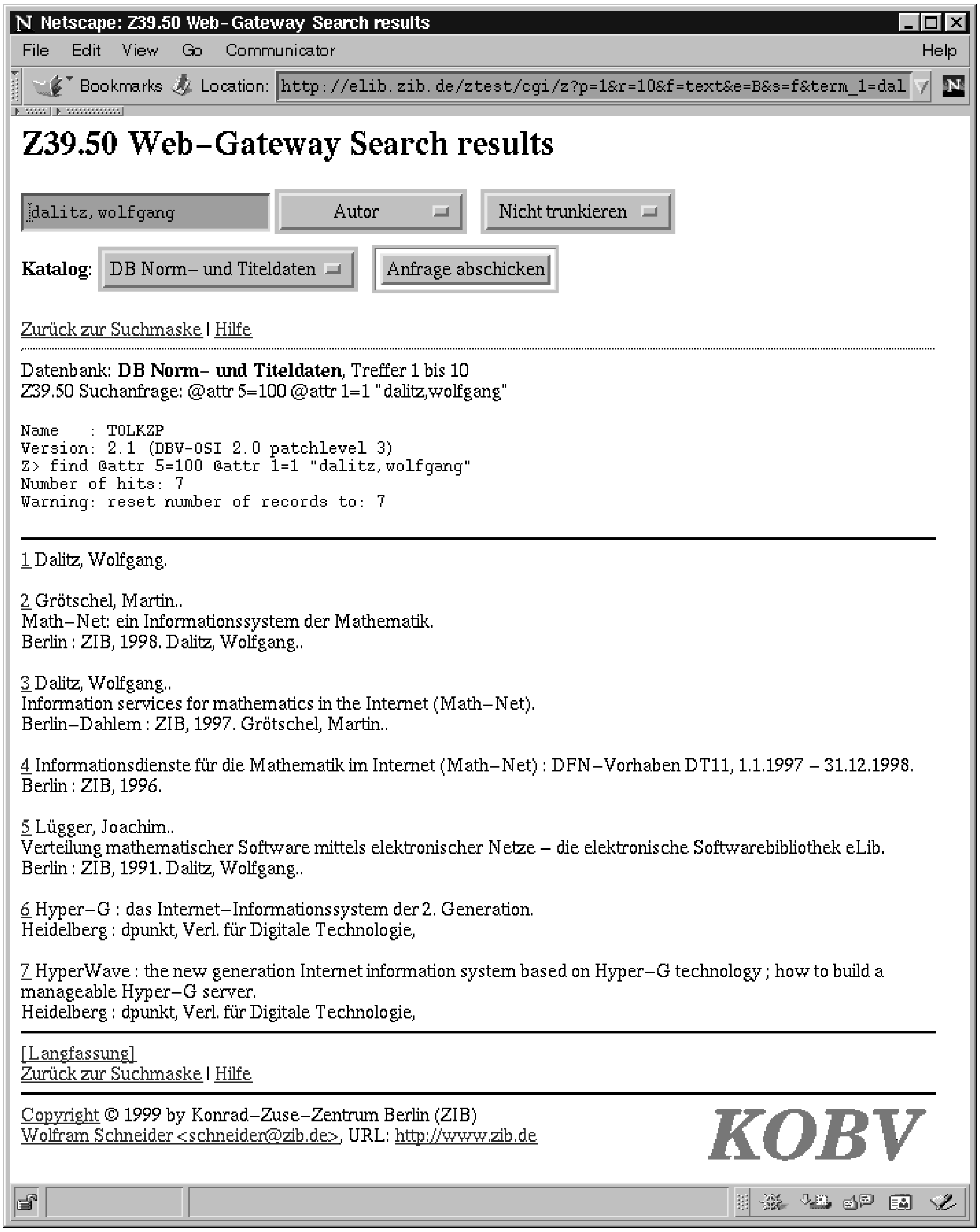


Abbildung 4.3: Suche nach Autor *Dalitz, Wolfgang*

In diesem Beispiel wird nach dem Autor "*Dalitz, Wolfgang*" in der Deutschen Bibliothek (Katalog DB Norm- und Titeldaten) gesucht. Gesucht wird ohne Trunkierung, d.h. nur nach Dokumenten, die den Autor "*Dalitz, Wolfgang*" exakt enthalten. Es werden maximal die ersten zehn Treffer - in diesem Fall sieben - in einer Kurztrefferliste ausgegeben. Jeder Datensatz hat eine Nummer in der Trefferliste. Klickt man auf diese Nummer, erhält man den Datensatz im MAB2- oder USMARC-Format (Kategorienformat). Mit der Verweisung *Langfassung* kann man sich die Treffer in einer erweiterten Kurztrefferliste ansehen - siehe auch die Abbildung 4.4

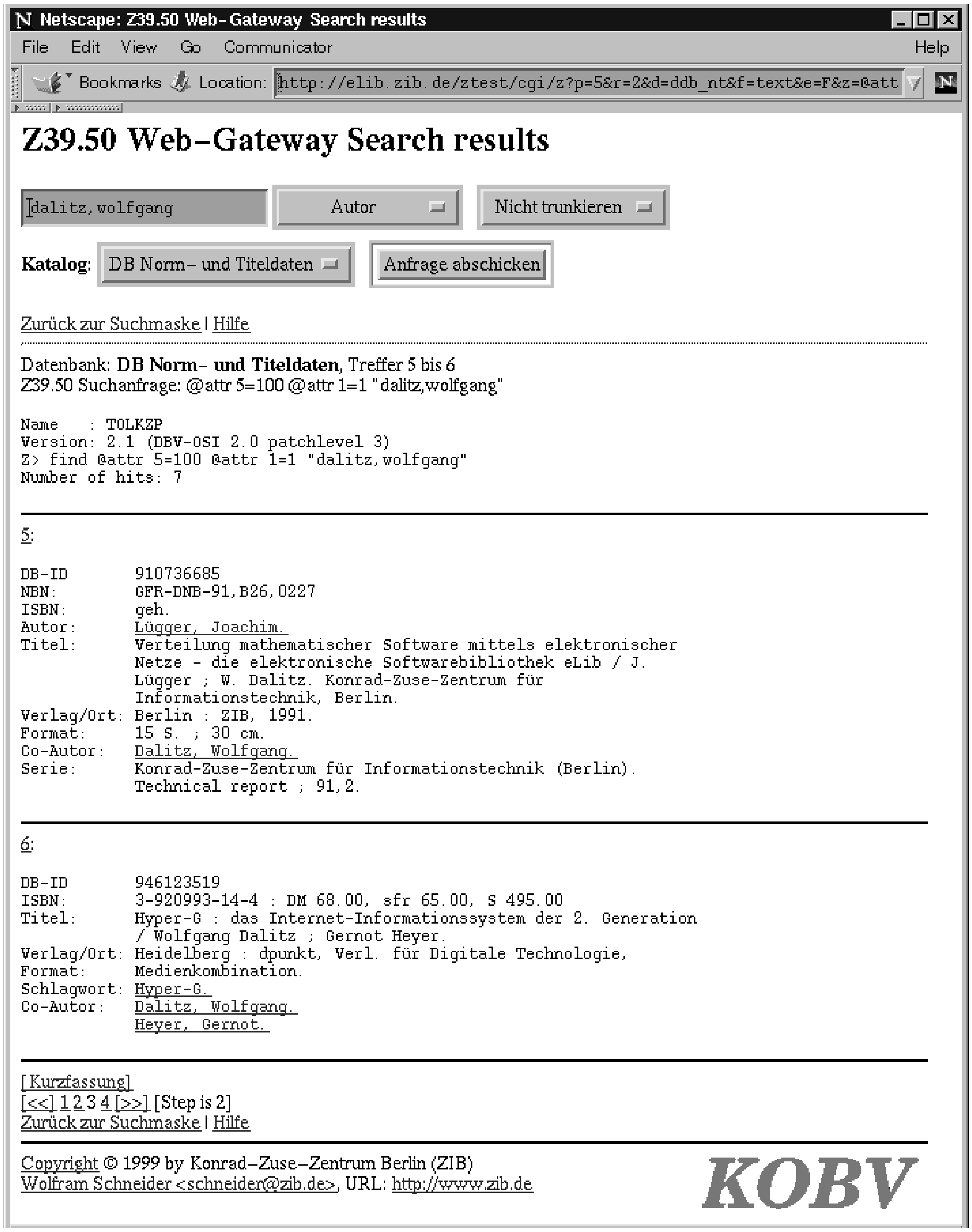


Abbildung 4.4: 5. und 6. Treffer in Textdarstellung anzeigen

Gesucht wurde nach dem Autor "*Dalitz, Wolfgang*" in der Deutschen Bibliothek. Ausgegeben werden jetzt der 5. und 6. Treffer in einer ausführlichen Textdarstellung. Dazu gehören auch die Identifikationsnummer (DB-ID) des Datensatzes, die ISBN-Nummer und die Schlagwörter. Mit dem Link *Kurzfassung* kann man sich die Treffer wieder in der Kurztrefferliste ansehen. In der Treffermenge kann mit den Links $\ll$ (Rückwärts, vorherige 2 Treffer, hier Treffer Nummer 3 und 4) und $\gg$ (Vorwärts, nächste 2 Treffer, hier Treffer Nummer 7) navigiert werden.

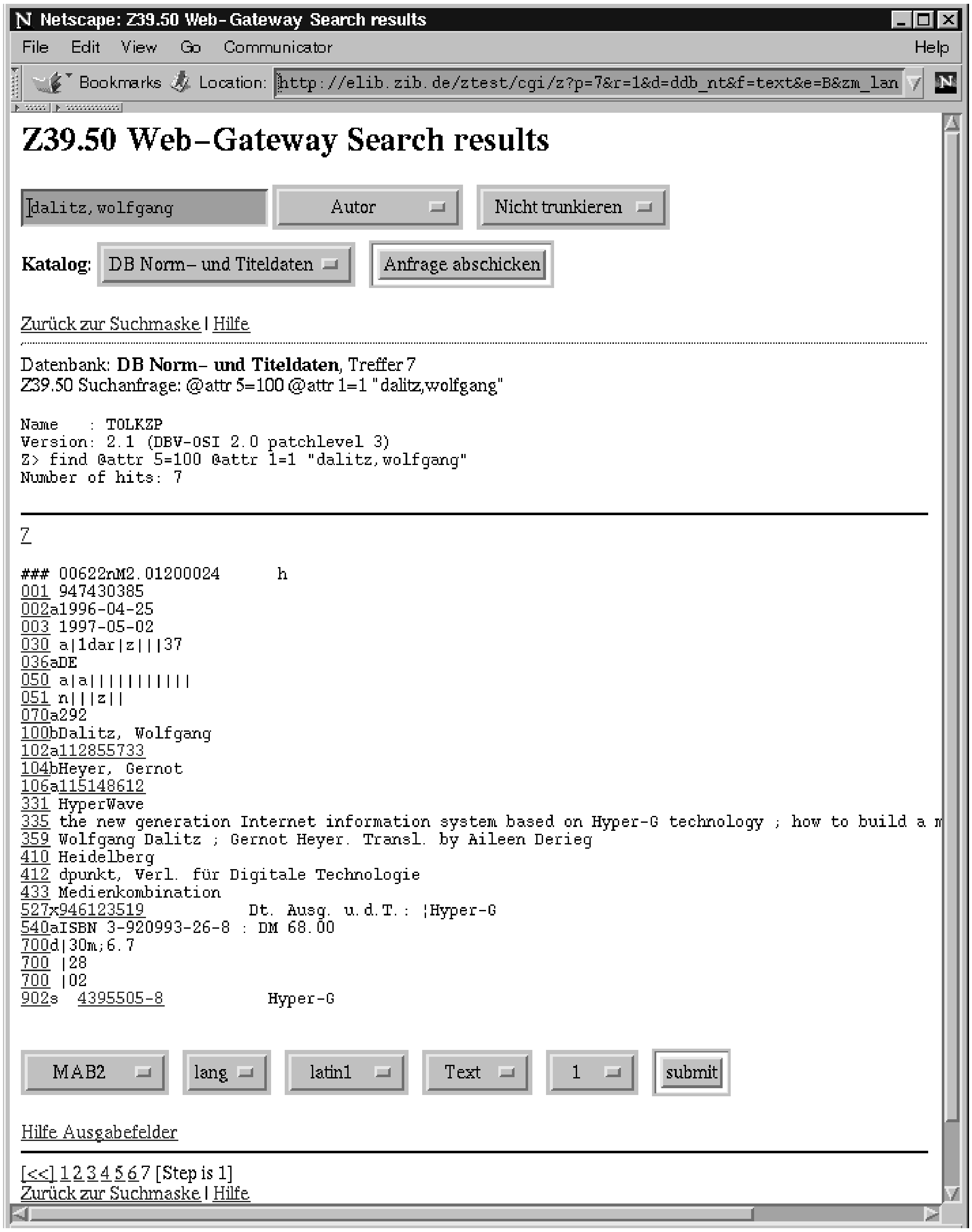


Abbildung 4.5: Treffer Nr. 7 in MAB anzeigen lassen

Gesucht wurde nach dem Autor "*Dalitz, Wolfgang*" in der Deutschen Bibliothek. Ausgegeben wird jetzt der 7. Treffer im Kategorienformat MAB2. Von den MAB2-Feldnummern führen Links auf die Online-Dokumentation von MAB2 (siehe im Anhang Abbildung C.7, Seite 135).

Referenzen auf verknüpfte Datensätze sind gesetzt. Folgt man im Feld 102 dem Link 112855733, so wird eine neue Suchanfrage nach dem Personennamen mit der Identifikationsnummer 112855733 gestellt. Als Ergebnis erhält man den Eintrag aus der Personennamendatei (PND) zu "*Dalitz, Wolfgang*". Der Link im Feld 106 verweist auf die ID des zweiten Verfassers "*Heyer, Gernot*" in der Personennamendatei; der Link im Feld 527 ist ein Verweis auf parallele

Ausgaben - hier die originale deutsche Ausgabe des Buches; und der Link im Feld 902 ist ein Verweis auf die ID der ersten Schlagwortkette "*Hyper-G*" in der Schlagwortnormdatei (SWD).

Nach dem Datensatz folgt eine Menüleiste. Mit dem Menü kann man die Optionen für die Übernahme in die lokale Datenbank festlegen. In diesem Beispiel: das Format MAB2, Volldarstellung, Zeichensatz latin1, Übernahme als Text, Anzahl der zu importierenden Datensätze gleich eins.

Personennamensatz *Dalitz, Wolfgang*

```
### 00182nM2.01200024      p
001 112855733
002a19910610
003 19970502
020a112855733¦1100
030 |a1d|||||||||
065 b|||
068cf
068bx
068ev
070 1100
070a292
800 Dalitz, Wolfgang
830 Dalitz, W.
```

Schlagwortsatz *Hyper-G*

```
### 00302nM2.01200024      s
001 4395505-8
002a19951211
003 19951213
030 |a1dzznz|||||
040 30m¦6.7
067 s100000|
070 1250
070a292
070b1250
800sHyper-G
808aVorlage
845s|Internet / Informationssystem / Hypermedia
845s|Informationssystem / Hypermedia / Internet
845s|Hypermedia / Informationssystem / Internet
```

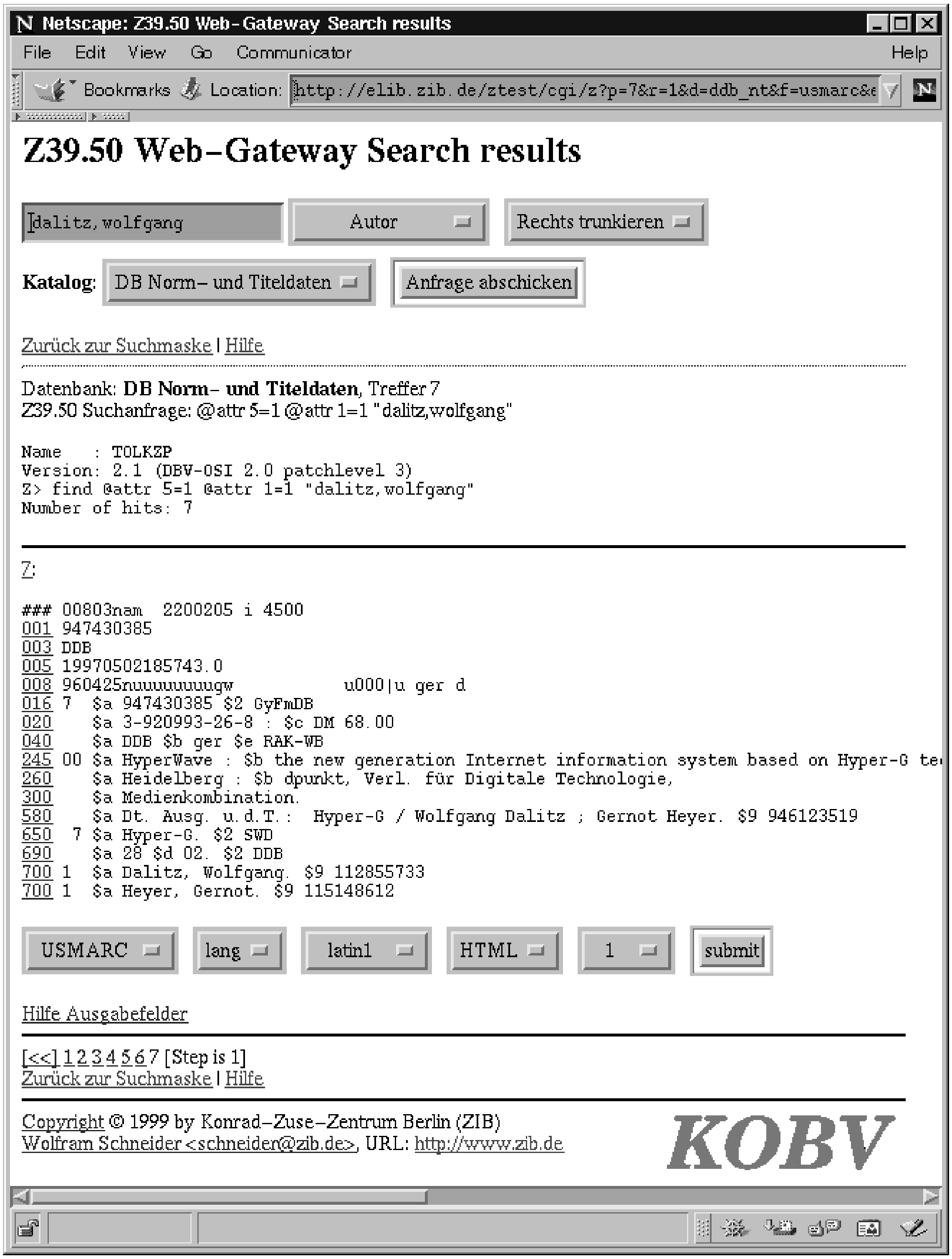


Abbildung 4.6: 7. Treffer in USMARC anzeigen lassen

Gesucht wurde nach dem Autor “*Dalitz, Wolfgang*” in der Deutschen Bibliothek. Ausgegeben wird jetzt der 7. Treffer im Kategorienformat USMARC. Von den USMARC-Feldnummern führen Links auf die Online-Dokumentation von USMARC (siehe Anhang Abbildung C.6, Seite 134).

Im unteren Drittel des Bildschirms kann man Optionen zur Übernahme in die lokale Datenbank einstellen. Hier sind eingestellt das Format *USMARC*, Volldarstellung *lang*, Zeichensatz *latin1* (ISO 8859-1), Darstellung als *HTML*-Seite, *einen* Datensatz ausgeben.

4.4.3 Detaillierte Suche

Im Unterschied zur einfachen Suche kann man bei der detaillierten Suche zwei oder drei Suchbegriffe eingegeben und mit Booleschen Operatoren verknüpfen. Dazu stehen die Operatoren "*UND*", "*ODER*", "*UND NICHT*" zur Verfügung.

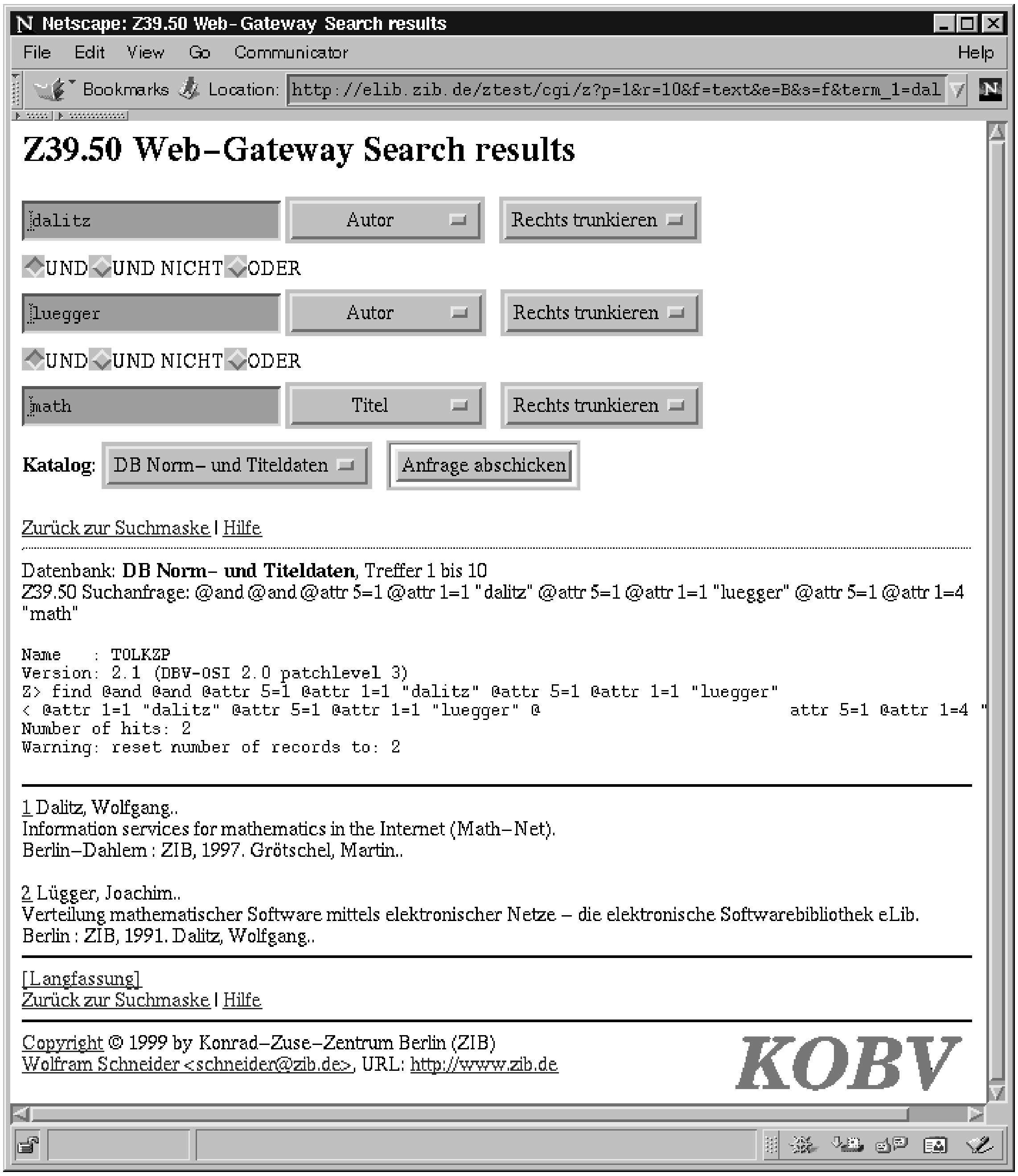


Abbildung 4.7: Suche nach Autor *Dalitz* und Autor *Luegger* und Titel *Math*

In diesem Beispiel wird nach Dokumenten der Autoren "*Dalitz*" und "*Luegger*" gesucht, die im Titel das Stichwort "*Math*" enthalten. Die Vornamen der Autoren werden mit der Option "*Rechts trunkieren*" bei der Suche ignoriert. Beim Titel wird ebenfalls rechts trunkiert, d.h. es wird nach Wörtern im Titel gesucht, die mit "*math*" anfangen, z.B. "Mathematik" oder "mathematisch". Es werden zwei Treffer gefunden und als Kurztrefferliste ausgegeben. Wie in der einfachen Suche ist es möglich, zur Langfassung und zum Kategorienformat zu verzweigen.

4.4.4 Registersuche

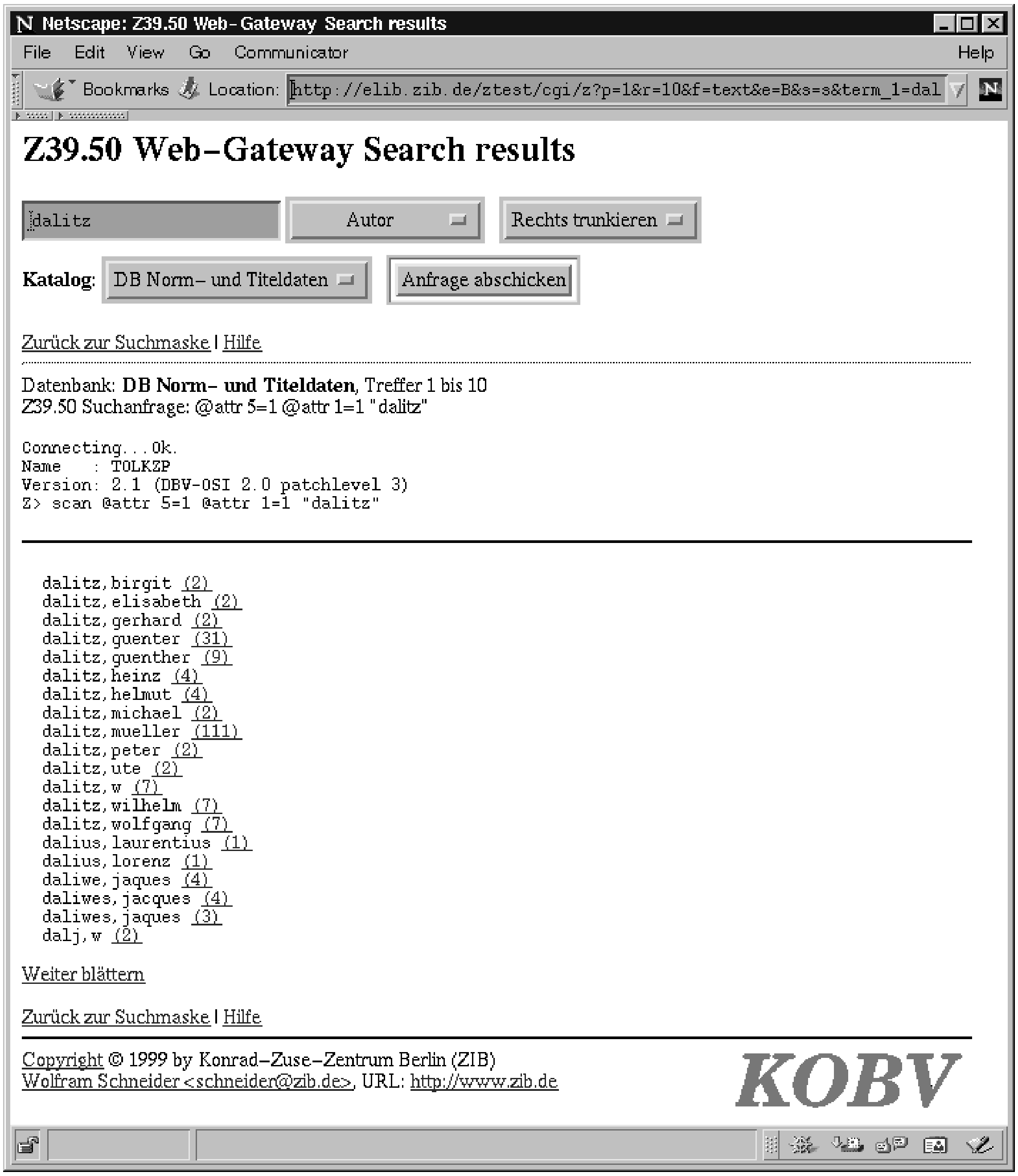


Abbildung 4.8: Registersuche nach Autor *Dalitz*

Mit der Registersuche kann man im Bestand einer Bibliothek blättern - vergleichbar mit der Suche in einem Zettelkatalog. Zu jedem Attribut (Autor, Titel etc.) ist angegeben, wieviele Datensätze zu dem betreffenden Autor oder Titel existieren.

In diesem Beispiel wird nach dem Autor "*Dalitz*" im Register der Deutschen Bibliothek gesucht. Als Ergebnis erhält man eine sortierte Liste der ersten 20 Einträge (hier Autoren), die mit dem Wort "*Dalitz*" anfangen. Hinter den Autoren steht in Klammern die Anzahl der Datensätze, in denen der Autor "*Dalitz*" enthalten ist. In diesem Beispiel gibt es zwei Dokumente von der Autorin "*Dalitz, Birgit*". Folgt man dieser Verweisung, wird eine neue Anfrage nach der Autorin "*Dalitz, Birgit*" gestellt - diesmal allerdings werden die Datensätze ausgegeben (Titelsuche) (siehe Abbildung 4.9).

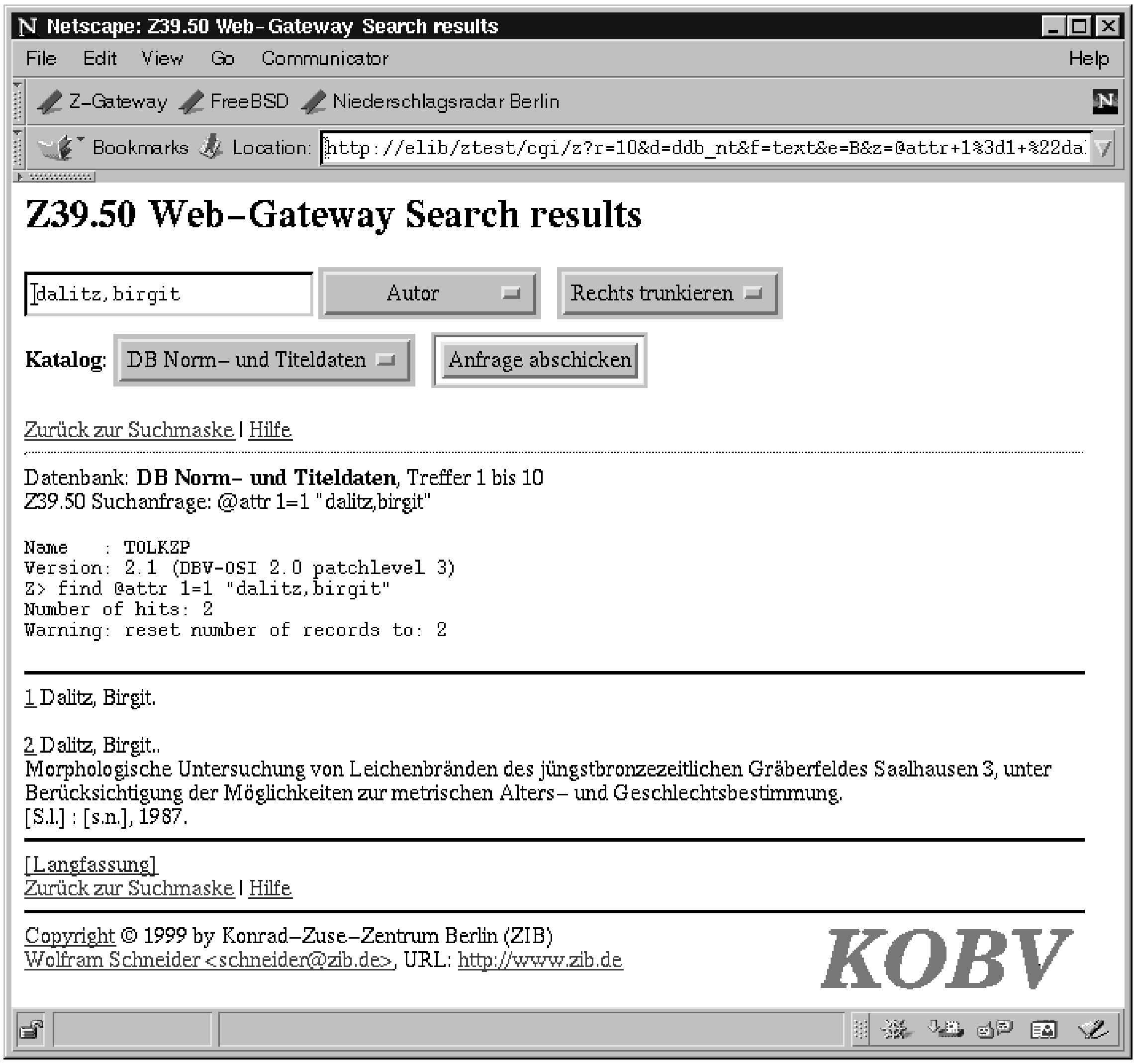


Abbildung 4.9: Ergebnis der Titelsuche nach vorheriger Registersuche

Nach der Registersuche (siehe vorherige Abbildung 4.8) wird eine neue Anfrage nach der Autorin "*Dalitz, Birgit*" in der Deutschen Bibliothek gestellt. Die beiden gefundenen Datensätze werden als Kurztrefferliste ausgegeben.

Bei der Datenbank der Deutschen Bibliothek gibt es eine Besonderheit: Norm- und Titeldaten stehen in einer Datenbank. In diesem Beispiel ist der erste Treffer ein Personennamensatz zu *Dalitz, Birgit* aus der Personennamendatei (PND). Der zweite Treffer ist der publizierte Titel der Autorin (siehe auch Kapitel 9 Probleme im laufenden Betrieb, Seite 97).

4.4.5 Suche mit Schlagwörtern - ein Thesaurus

Ein Thesaurus ist eine geordnete Zusammenstellung von Begriffen mit ihren (natürlichsprachlichen) Bezeichnungen ([Fuh97]). Die wesentlichen Merkmale eines Thesaurus sind:

Terminologische Kontrolle: Die terminologische Kontrolle dient der Erfassung von mehrdeutigkeiten und Unschärfen der natürlichen Sprache (z.B. Frisör ⇔ Friseur, Rundfunk ⇔ Hörfunkt, Bank ⇔ Bank).

Beziehungsgefüge: Die Darstellung von Beziehungen zwischen Begriffen. Dazu gehören z.B. hierarchische Relationen mit Ober- und Unterbegriffen (Obstbaum ⇔ Steinobstbaum) und verwandte Begriffe.

Ein Drittel der Datensätze in der Deutschen Bibliothek sind mit Schlagwörtern bzw. Schlagwortketten versehen. Dies ist eine riesige Wissensbasis von mehreren hunderttausend sachlich erschlossenen Publikationen. Mit der Integration der Schlagwortnormdatei in ihre Datenbank bietet die Deutsche Bibliothek einen einfachen Thesaurus an. Dies wird anhand der Abbildungen 4.10 und 4.11 deutlich.

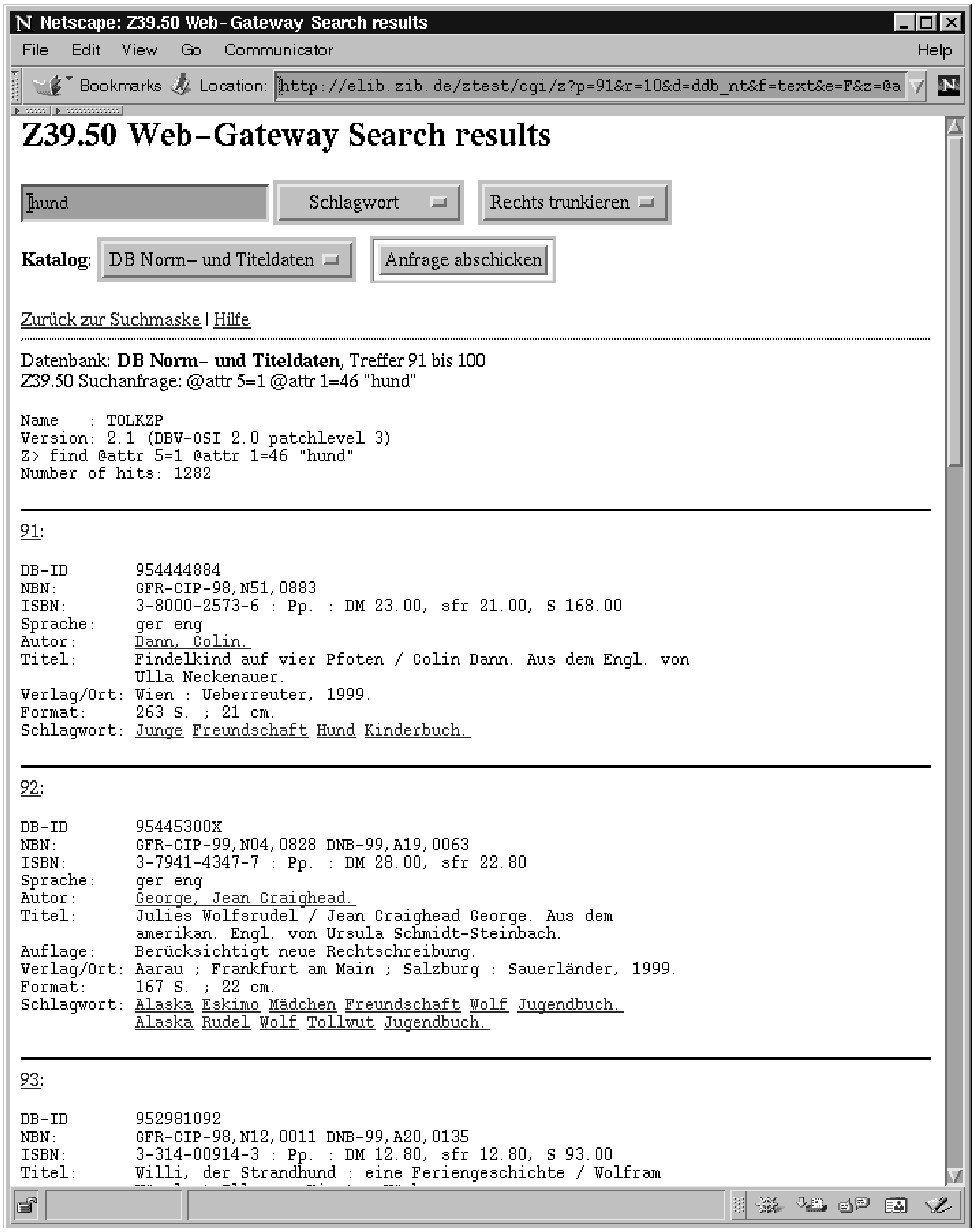
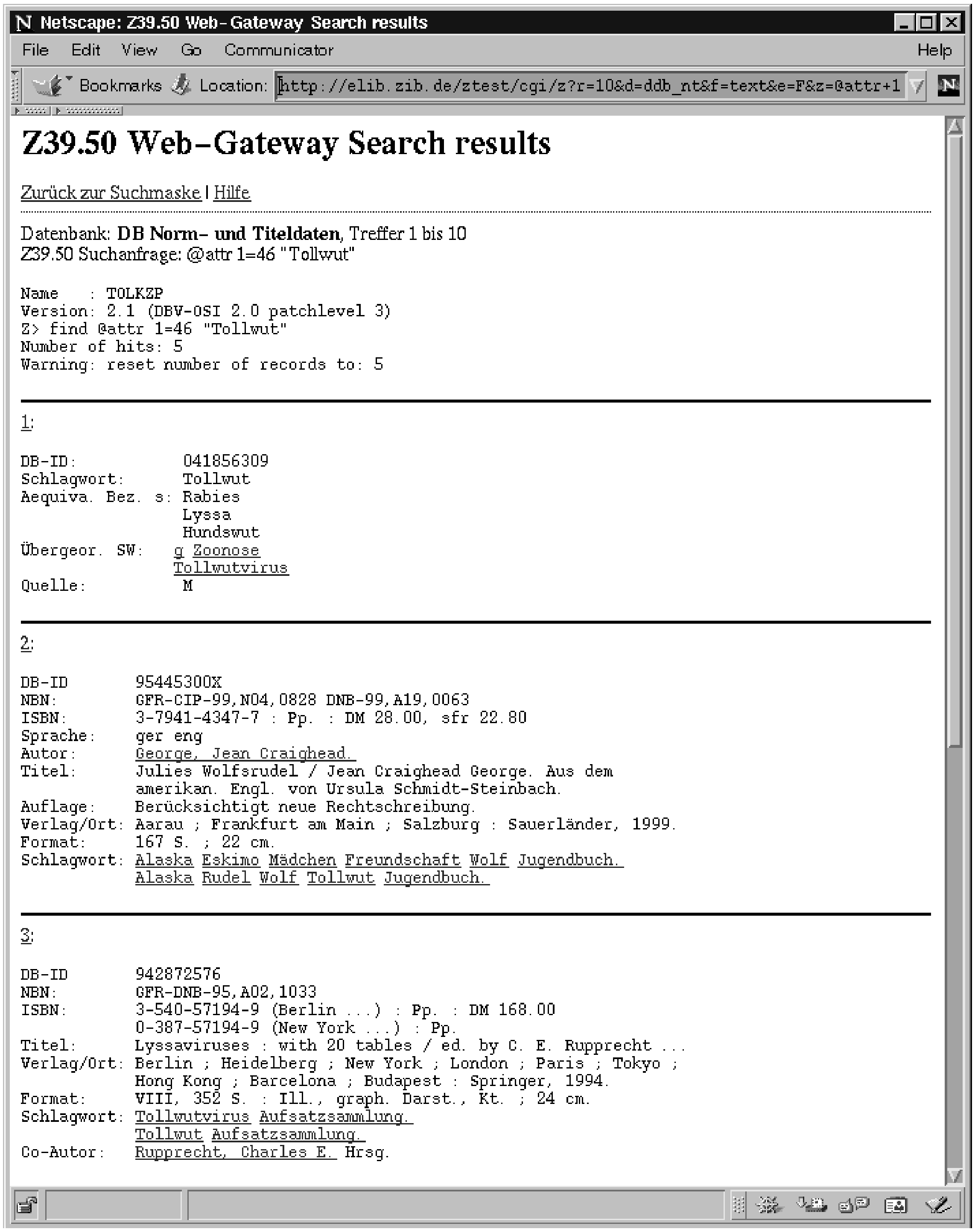


Abbildung 4.10: Suche nach Schlagwort *Hund*

In dem Beispiel aus Abbildung 4.10 wird nach dem Schlagwort "*Hund*" in der Deutschen Bibliothek gesucht. Gesucht wird mit Trunkierung, d.h. nach allen Schlagwörtern, die mit *Hund* anfangen, z.B. *Hund*, *Hundeerziehung*, *Hundehaltung* etc. Es werden 1282 Treffer gefunden. Ausgegeben wird jetzt der 91. und 92. Treffer. Referenzen zum Autor oder zu anderen Schlagwörtern sind soweit als möglich gesetzt. Folgt man dem Schlagwort *Tollwut* in Treffer 92, wird eine Anfrage nach dem Schlagwort *Tollwut* gestellt. Siehe die folgende Abbildung 4.11.


Netscape: Z39.50 Web-Gateway Search results

File Edit View Go Communicator Help

Bookmarks Location: http://elib.zib.de/ztest/cgi/z?r=10&d=ddb_nt&f=text&e=F&z=@attr+1

Z39.50 Web-Gateway Search results

Zurück zur Suchmaske | Hilfe

Datenbank: **DB Norm- und Titeldaten**, Treffer 1 bis 10
Z39.50 Suchanfrage: @attr 1=46 "Tollwut"

```
Name   : TOLKZP
Version: 2.1 (DBV-OSI 2.0 patchlevel 3)
Z> find @attr 1=46 "Tollwut"
Number of hits: 5
Warning: reset number of records to: 5
```

1:

```
DB-ID:           041856309
Schlagwort:      Tollwut
Aequiva. Bez. s: Rabies
                 Lyssa
                 Hundswut
Übergeor. SW:   g Zoonose
                Tollwutvirus
Quelle:          M
```

2:

```
DB-ID       95445300X
NBN:        GFR-CIP-99,N04,0828 DNB-99,A19,0063
ISBN:       3-7941-4347-7 : Pp. : DM 28.00, sfr 22.80
Sprache:    ger eng
Autor:      George, Jean Craighead.
Titel:      Julies Wolfsrudel / Jean Craighead George. Aus dem
            amerikan. Engl. von Ursula Schmidt-Steinbach.
Auflage:    Berücksichtigt neue Rechtschreibung.
Verlag/Ort: Aarau ; Frankfurt am Main ; Salzburg : Sauerländer, 1999.
Format:     167 S. ; 22 cm.
Schlagwort: Alaska Eskimo Mädchen Freundschaft Wolf Jugendbuch.
            Alaska Rudel Wolf Tollwut Jugendbuch.
```

3:

```
DB-ID       942872576
NBN:        GFR-DNB-95,A02,1033
ISBN:       3-540-57194-9 (Berlin ...) : Pp. : DM 168.00
            0-387-57194-9 (New York ...) : Pp.
Titel:      Lyssaviruses : with 20 tables / ed. by C. E. Rupprecht ...
Verlag/Ort: Berlin ; Heidelberg ; New York ; London ; Paris ; Tokyo ;
            Hong Kong ; Barcelona ; Budapest : Springer, 1994.
Format:     VIII, 352 S. : Ill., graph. Darst., Kt. ; 24 cm.
Schlagwort: Tollwutvirus Aufsatzsammlung.
            Tollwut Aufsatzsammlung.
Co-Autor:   Rupprecht, Charles E. Hrsg.
```

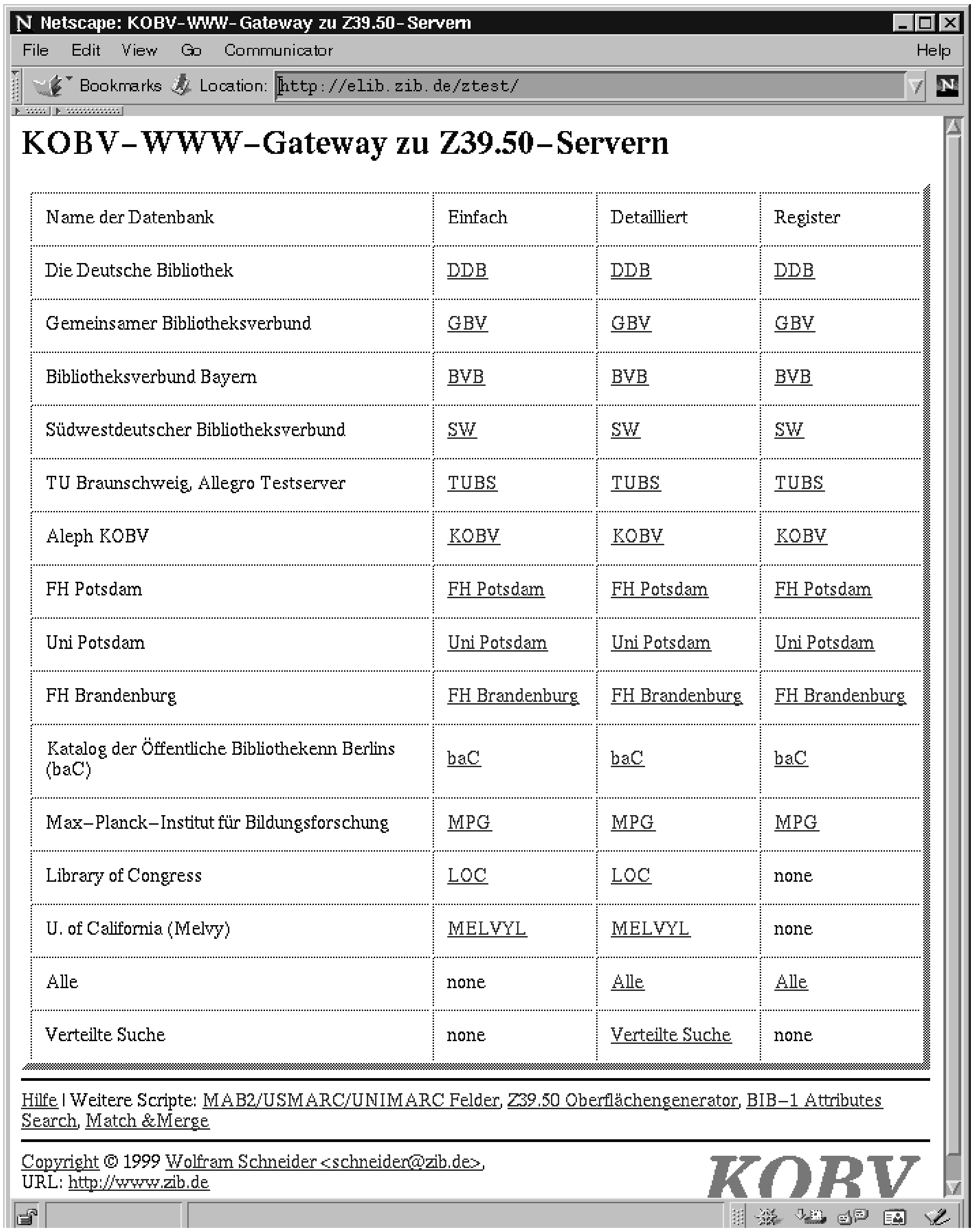
Abbildung 4.11: Suche nach Schlagwort *Tollwut*

Es wurde nach dem Schlagwort *Tollwut* in der Deutschen Bibliothek gesucht (siehe vorherige Abbildung 4.10). Es werden 5 Treffer gefunden und in einer ausführlichen Textdarstellung ausgegeben. Der erste Treffer ist ein Schlagwortsatz aus der Schlagwortnormdatei (SWD) für das Schlagwort *Tollwut*. Es enthält die Synonyme *Rabies*, *Lyssa* und *Hundswut*. Das übergeordnete Schlagwort ist *Zoonose*.

4.5 Zweites ZACK-System: Parallele Suche in mehreren Z39.50-Datenbanken

Das zweite *ZACK*-System ist eine Weiterentwicklung des ersten *ZACK*-Systems. Als besonderes neues Feature wird die verteilte Suche angeboten (siehe auch das Kapitel 2 Verteilte Suche, Seite 3). Der Benutzer kann nun in mehreren Datenbanken gleichzeitig recherchieren. Die Treffer werden auf Dubletten überprüft. Dem Benutzer wird eine übersichtliche Kurztrefferliste ohne doppelte Einträge angeboten.

4.5.1 Einstiegsseite ZACK zweites System



KOBV-WWW-Gateway zu Z39.50-Servern

Name der Datenbank	Einfach	Detailliert	Register
Die Deutsche Bibliothek	DDB	DDB	DDB
Gemeinsamer Bibliotheksverbund	GBV	GBV	GBV
Bibliotheksverbund Bayern	BVB	BVB	BVB
Südwestdeutscher Bibliotheksverbund	SW	SW	SW
TU Braunschweig, Allegro Testserver	TUBS	TUBS	TUBS
Aleph KOBV	KOBV	KOBV	KOBV
FH Potsdam	FH Potsdam	FH Potsdam	FH Potsdam
Uni Potsdam	Uni Potsdam	Uni Potsdam	Uni Potsdam
FH Brandenburg	FH Brandenburg	FH Brandenburg	FH Brandenburg
Katalog der Öffentliche Bibliothekenn Berlins (baC)	baC	baC	baC
Max-Planck-Institut für Bildungsforschung	MPG	MPG	MPG
Library of Congress	LOC	LOC	none
U. of California (Melvy)	MELVYL	MELVYL	none
Alle	none	Alle	Alle
Verteilte Suche	none	Verteilte Suche	none

Hilfe | Weitere Scripte: MAB2/USMARC/UNIMARC Felder, Z39.50 Oberflächengenerator, BIB-1 Attributes Search, Match &Merge

Copyright © 1999 Wolfram Schneider <schneider@zib.de>,
URL: http://www.zib.de

KOBV

Abbildung 4.12: Einstiegsseite WWW-Z39.50-Gateway, zweites *ZACK*-System

Die Abbildung 4.12 zeigt die Einstiegsseite des zweiten *ZACK* Systems. Zur Auswahl stehen alle bekannten deutschen Z39.50-Server und zwei amerikanische Anbieter. Im Vergleich zum ersten System stehen deutlich mehr Z39.50-Server in Deutschland zur Auswahl. Die Suchmöglichkeiten *einfach*, *detailliert*, *Register* entsprechen dem des ersten *ZACK*-Systems.

4.5.2 Verteilte Suche nach ISBN-Nummer

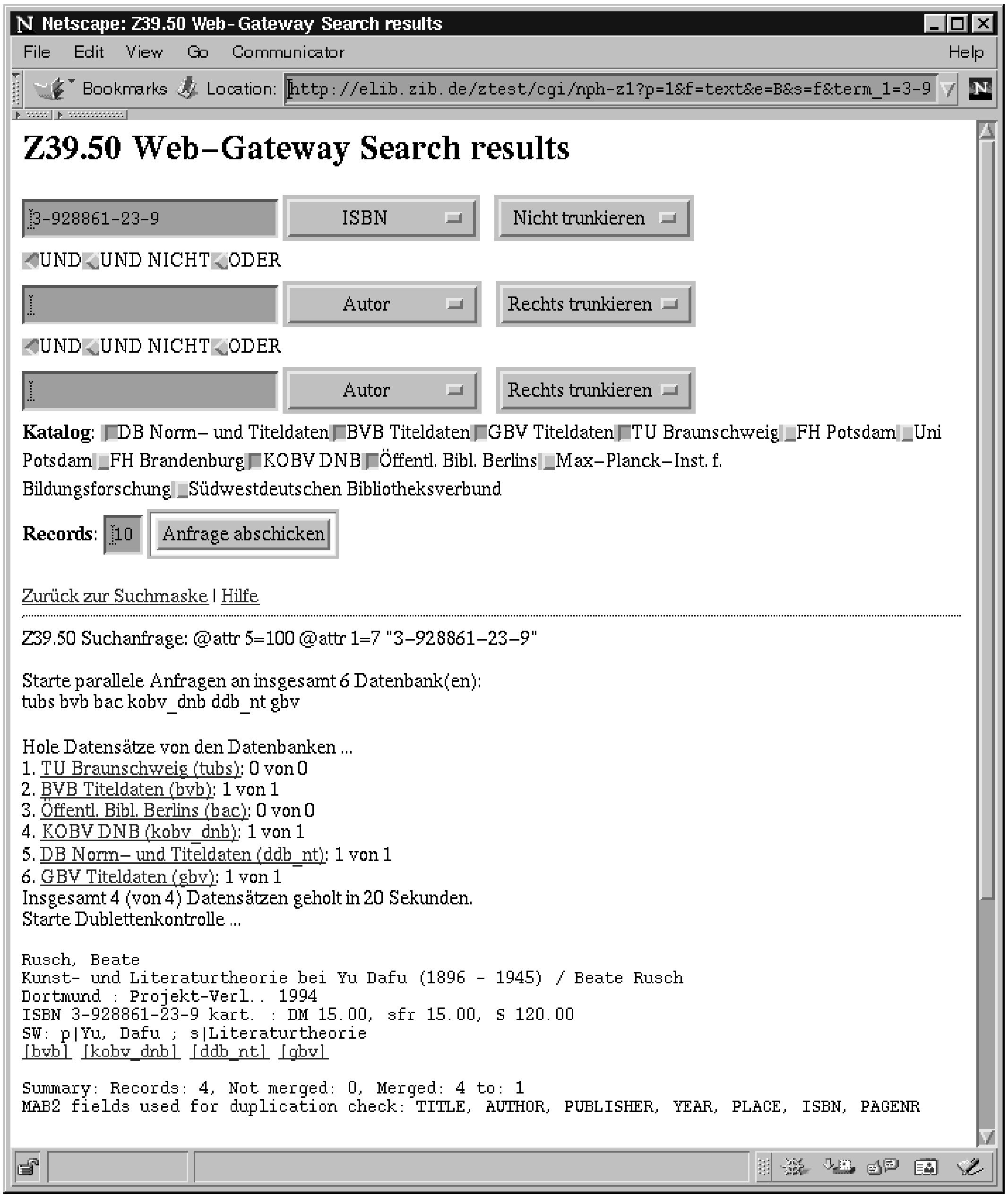


Abbildung 4.13: Verteilte Suche nach ISBN-Nummer

In diesem Beispiel wird nach der ISBN-Nummer "*3-928861-23-9*" gleichzeitig in mehreren Datenbanken gesucht. Die Anfragen werden an die Datenbanken der Technischen Universität Braunschweig, des Bibliotheksverbundes Bayern, der Öffentlichen Bibliotheken Berlins, des KOBV-Testservers, der Deutschen Bibliothek und des Gemeinsamen Bibliotheksverbundes gestellt. Das gewünschte Buch ist beim Bibliotheksverbund Bayern, im KOBV-Testserver, der Deutschen Bibliothek und dem Gemeinsamen Bibliotheksverbund jeweils einmal vorhanden. Die Datensätze werden von den Datenbanken geholt und auf Gleichheit bzw. Ähnlichkeit überprüft.

Die Dublettenkontrolle ergibt, daß alle Datensätze gleich sind. Ein Datensatz wird in der Kurztrefferliste ausgeben, mit dem Zusatz, in welchen Datenbanken er vorhanden ist. Folgt man diesem Link, erhält man den Datensatz von der betreffenden Bibliothek. Für die Dublettenkontrolle wurden die Attribute *Titel*, *Autor*, *Verlag*, *Jahr*, *Verlagsort*, *ISBN-Nummer* und *Seitennummer* verwendet. Zur besseren Übersicht werden die Datensätze vom BVB, DDB, GBV und KOBV (in dieser Reihenfolge) hier leicht gekürzt im MAB2-Kategorienformat ausgegeben. Hier nicht dargestellt werden interne oder automatisch erzeugte Felder (z.B. Feldnummern 002-099) sowie Schlagwörter.

```
### 00499nM2.01000024      h
001 00097905259
100 Rusch, Beate
331 Kunst- und Literaturtheorie bei Yu Dafu (1896 - 1945)
359 Beate Rusch
410 Dortmund
412 Projekt-Verl.
425a1994
433 79 S.
451 Edition Cathay ; 2
540aISBN 3-928861-23-9

### 00626nM2.01200024      h
001 941905101
100 Rusch, Beate
331 Kunst- und Literaturtheorie bei Yu Dafu (1896 - 1945)
359 Beate Rusch
410 Dortmund
412 Projekt-Verl.
425a1994
433 79 S.
435 21 cm
451 Edition Cathay ; Bd. 2
540aISBN 3-928861-23-9 kart. : DM 15.00, sfr 15.00, S 120.00

### 00528nM2.01000024      h
001 16.390660.2
100 Rusch, Beate
331aKunst- und Literaturtheorie bei Yu Dafu (1896 - 1945)
359 Beate Rusch
410 Dortmund
412 Projekt-Verl.
425a1994
433 79 S
435 21 cm
451 Edition Cathay ; Bd. 2
511 Erscheinungsjahr in Vorlageform:1994
540aISBN 3-928861-23-9 (kart.) : DM 15.00, sfr 15.00, S 120.00

### 00653nM2.01200024      h
001 941905101
100 Rusch, Beate
331 Kunst- und Literaturtheorie bei Yu Dafu (1896 - 1945)
359 Beate Rusch
410 Dortmund
412 Projekt-Verl.
425a1994
433 79 S.
435 21 cm
451 Edition Cathay ; Bd. 2
540aISBN 3-928861-23-9 kart. : DM 15.00, sfr 15.00, S 120.00
```


4.5.3 Verteilte Suche nach Autor und Titel

Abbildung 4.14: Verteilte Suche nach dem Buch *Hyper-G*, erster Teil

In diesem Beispiel wird nach dem Buch "*Hyper*" der Autoren "*Dalitz, Wolfgang*" und "*Heyer, Gernot*" gesucht. Der Titel wird mit der Option "Rechts trunkieren" gesucht - damit ist sichergestellt, daß auch "*hyperg*" und "*Hyper-G*" gefunden werden. Die Anfragen werden an die Datenbanken der Technischen Universität Braunschweig, des Bibliotheksverbundes Bayern, des KOBV-Testservers, der Deutschen Bibliothek, der Fachhochschule Potsdam und des Gemeinsamen Bibliotheksverbundes gleichzeitig gestellt.

Das gewünschte Buch ist beim Bibliotheksverbund Bayern 3 mal, beim KOBV-Testserver 2 mal, bei der Deutschen Bibliothek 2 mal, der Fachhochschule Potsdam 2 mal und dem Gemeinsamen Bibliotheksverbund 3 mal vorhanden. Die insgesamt 13 Datensätze werden von den Datenbanken geholt und auf Gleichheit bzw. Ähnlichkeit überprüft. Die Dublettenkontrolle ergibt, daß drei unterschiedliche Werke in den Datenbanken vorhanden sind. Diese werden in einer Kurztrefferliste ausgegeben (siehe die Abbildung 4.15).

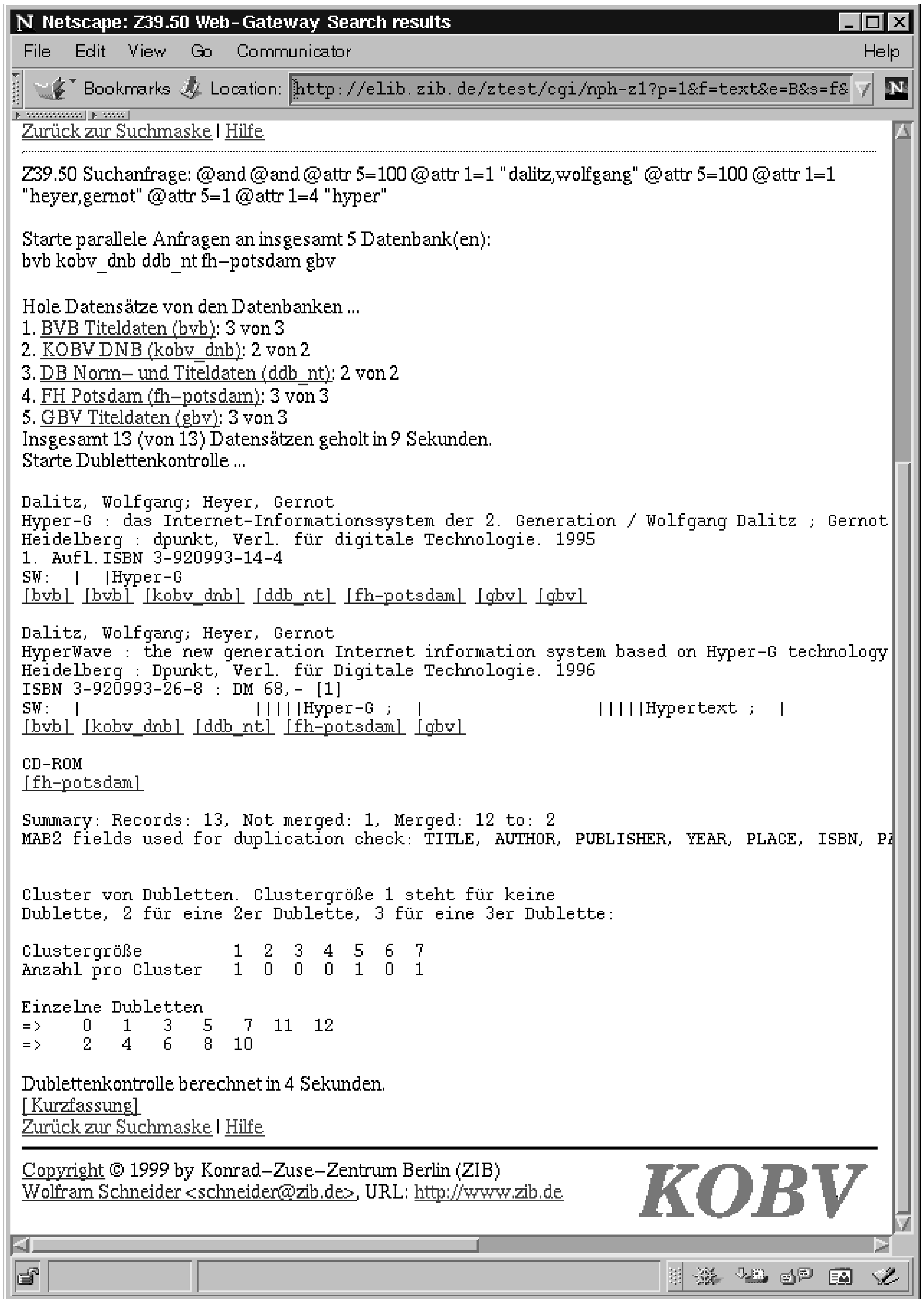


Abbildung 4.15: Verteilte Suche nach dem Buch *Hyper-G*, zweiter Teil

[Fortsetzung der vorherigen Abbildung 4.14] Die drei Titel sind einmal die originale deutsche Ausgabe, die englische Übersetzung und die CD-ROM. Die deutsche Ausgabe gibt es 7 mal: 2 mal beim BVB, einmal beim KOBV, einmal bei der DDB, einmal bei der FH Potsdam und 2 mal beim GBV. Die englische Ausgabe gibt es 5 mal: jeweils einmal beim BVB, KOBV, DDB, FH Potsdam und GBV. Die CD-ROM gibt es als Datensatz bei der FH-Potsdam.

Für die Dublettenkontrolle wurden die Attribute *Titel*, *Autor*, *Verlag*, *Jahr*, *Verlagsort*, *ISBN* und *Seitennummer* verwendet. Es wurde ein Dokument gefunden, das einmal vorkommt; ein Dokument, das 5 mal vorkommt (5er Dublette) und ein Dokument, das 7 mal (7er Dublette) vorkommt. Die Dublettenkontrolle hat 4 Sekunden auf dem Rechner elib [4] gedauert.

[4] Eine SPARCstation-20, 50 Mhz CPU, siehe im Glossary unter elib

Kapitel 5

Normierung

Bei der Dublettenkontrolle werden Datensätze unterschiedlicher Herkunft miteinander verglichen. Bevor dies geschehen kann, müssen Zeichensatz, Fehleingaben (z.B. doppelte Leerzeichen), unterschiedliche Erfassungspraktiken erkannt und bearbeitet werden. In diesem Kapitel werden unterschiedliche Datensätze unter diesen Gesichtspunkten analysiert.

Ziel der Normierung ist es, die unterschiedlichen Schreibweisen eines Wortes oder einer Wortgruppe zu vereinheitlichen. Es werden Unterschiede in den Zeichenketten ausgeglichen, die keine oder nahezu keine inhaltliche Bedeutung haben ([RM94], [Rus99a], [LSW99], [DHHS91], [SH91]).

Sonderzeichen werden entfernt oder durch andere Zeichen ersetzt, Großbuchstaben in Kleinbuchstaben umgewandelt, überflüssige Leerzeichen gelöscht. Die Anzahl der verwendeten Zeichen wird von 256 Zeichen auf 26 Buchstaben und 10 Ziffern reduziert. Die Normierung ist um so effizienter, je weniger unterschiedliche Schreibweisen generiert werden.

Eine perfekte Normierung existiert jedoch nicht. Die Normierung kann unbeabsichtigt unterschiedliche Sachverhalte zusammenfassen. Beispielsweise kann "*Frankfurt am Main*" und "*Frankfurt an der Oder*" zu dem Wort "*frankfurt*" zusammengefaßt werden. Diese Fehler sind in der Anwendung vernachlässigbar, da zur Dublettenprüfung jeweils mehrere Attribute (Titel, Autor, Verlag, Jahr etc.) herangezogen werden. Eine zufällige Übereinstimmung eines Attributes [1] wird durch den Vergleich der anderen Attribute korrigiert.

Zuerst werden die wichtigsten **Normierungsfunktionen** in Abschnitt 5.1 aufgelistet und anhand von Beispielen kurz erläutert. In Abschnitt 5.2 wird erklärt, **wie und in welcher Reihenfolge** die Normierungsfunktionen im System *ZACK* auf die Attribute angewandt werden.

In Abschnitt 5.3 wird die **Normierung des Attributes Autor mit Daten unterschiedlicher Herkunft getestet**. Es wird manuell geprüft, ob die Normierung des Attributes *Autor* korrekte oder falsche Ergebnisse ergibt.

In Abschnitt 5.4 werden die **unterschiedlichen Schreibweisen von** *Frankfurt am Main* und *Frankfurt an der Oder* in der Deutschen Bibliothek untersucht.

In Abschnitt 5.5 wird die **Normierung von** *ZACK* **auf 2,5 Millionen DNB-Datensätze** der Deutschen Bibliothek angewendet und statistisch analysiert. In Abschnitt 5.6 wird die **Normierung von** *ZACK* **auf Datensätze unterschiedlicher Herkunft** (DDB, BVB, GBV) angewandt und statistisch analysiert.

Die Untersuchungen in diesem Kapitel beziehen sich nur auf das Format MAB2 und Datensätze von deutschen Bibliotheken. Andere Formate wie USMARC oder UNIMARC werden nicht betrachtet.

[1] Dies kann z.B. auch der Fall sein, wenn zwei Autoren denselben Vor- und Nachnamen haben.

5.1 Normierungsfunktionen

Die Normierung eines Attributes kann in mehrere Teilschritte zerlegt werden. Dadurch ist es wesentlich einfacher, die Normierung eines Attributes zu implementieren und im laufenden Betrieb zu konfigurieren. Jeder Teilschritt beinhaltet eine Normierungsfunktion. Einige Normierungsfunktionen können auf alle Attribute angewandt werden (z.B. doppelte Leerzeichen löschen), andere Funktionen sind speziell für bestimmte Attribute entwickelt worden (z.B. ISBN Vereinheitlichung).

Nachfolgend werden die wichtigsten und bekanntesten Normierungsfunktionen aufgelistet und anhand von Beispielen kurz erläutert. Die Beispiele wurden den Datensätzen der Deutschen Bibliothek entnommen. Wie und in welcher Reihenfolge die Normierungsfunktionen in der Praxis angewandt werden, wird in Abschnitt 5.2 erläutert.

Zur besseren Lesbarkeit werden die Zeichenfolgen in Anführungszeichen gesetzt. Damit bleiben auch die Leerzeichen am Anfang oder Ende sichtbar, die durch die Anwendung einer *einzelnen* Normierungsfunktion entstehen. In der Praxis werden die überflüssigen Leerzeichen durch eine weitere Normierungsfunktion entfernt.

5.1.1 Allgemeine Normierungsfunktionen

Eckige Klammern

Alle Zeichen zwischen eckigen Klammern ("[", "]") und die eckigen Klammern selbst werden gelöscht. Zwischen eckigen Klammern stehen häufig Anmerkungen und Ergänzungen der Katalogisierer, die nicht in der Vorlage enthalten sind. Da diese Anmerkungen Interpretationsspielräume zulassen, sind sie für die Dublettenkontrolle in verteilten Systemen nur bedingt geeignet.

vorher	nachher
"2. ed., [Nachdr.]"	"2. ed., "
"[Briefmarkenfreunde Perl]"	""
"Cambridge [u.a.]"	"Cambridge "

Tabelle 5.1: Normierungsfunktion: Eckige Klammern

Nicht-Sortierzeichen

Alle Zeichen zwischen den Nicht-Sortierzeichen ("¬") und die Nicht-Sortierzeichen selbst werden gelöscht. Zwischen dem Nicht-Sortierzeichen stehen Wörter, die bei der Sortierung im Register nicht beachtet werden sollen. Dazu gehörten meistens die Artikel "*der*", "*die*", "*das*" und "*the*". Der Artikel eines Titels hat eine geringe inhaltliche Bedeutung. Alternativ: nur die Nicht-Sortierzeichen werden gelöscht (siehe Normierungsfunktion Sonderzeichen löschen).

vorher	nachher
"¬Die¬ Spatzen im Birnbaum"	" Spatzen im Birnbaum"
"Wildsmith, Brian ¬[Ill.]¬"	"Wildsmith, Brian "
"¬M.¬ Groß"	" Groß"
"¬Der¬ Steppenwolf"	"Steppenwolf"
"Der Steppenwolf"	"Der Steppenwolf"
"Steppenwolf"	"Steppenwolf"

Tabelle 5.2: Normierungsfunktion: Nicht-Sortierzeichen

Anhand des realen Beispieles *Steppenwolf* aus der DDB ist schwer zu entscheiden, welche der beiden Varianten besser ist. Es kommt in der Praxis doch recht häufig vor, daß die Nicht-Sortierzeichen bei der Erfassung vergessen werden.

Groß-und Kleinschreibung

Alle Zeichen werden klein geschrieben. Alternativ: alle Kleinbuchstaben werden in Großbuchstaben umgewandelt. Die Groß- und Kleinschreibung hat inhaltlich praktisch keine Bedeutung. Sie wird bei fast allen Bibliotheksdatenbanken ignoriert.

vorher	nachher
"Schatten im Paradies"	"schatten im paradies"
"Wall, Larry"	"wall, larry"
"HyperWave"	"hyperwave"

Tabelle 5.3: Normierungsfunktion: Groß-und Kleinschreibung

Umlaute konvertieren

Umlaute (8 Bit) werden durch Buchstaben (7 Bit, ASCII) ersetzt ("*ä*" ⇒ "*ae*"). Dadurch wird die Anzahl der Zeichen weiter reduziert und unterschiedliche Schreibweisen ein und desselben Autors erkannt.

vorher	nachher
"Lügger, Joachim"	"Luegger, Joachim"
"Große Brandenburger Ausgabe"	"Grosse Brandenburger Ausgabe"
"Großraum-Städteatlas Saarland"	"Grossraum-Staedteatlas Saarland"

Tabelle 5.4: Normierungsfunktion: Umlaute konvertieren

Sonderzeichen löschen

Alle Sonderzeichen (Nicht-Buchstaben oder Nicht-Ziffern) werden gelöscht bzw. durch ein Leerzeichen ersetzt. Die Anzahl der Zeichen wird drastisch reduziert.

vorher	nachher
"1. Aufl., 1. - 230. Tsd."	"1 Aufl 1 230 Tsd "
"Schwarzkopf & Schwarzkopf"	"Schwarzkopf Schwarzkopf"
"Hukkanen, Marja-Leena"	"Hukkanen Marja Leena"
"1. Aufl., [dt. Ausg. der 2. [Orig.]-Aufl., neue Aufl., neue Übers.]"	"1 Aufl dt Ausg der 2 Orig Aufl neue Aufl neue Übers "

Tabelle 5.5: Normierungsfunktion: Sonderzeichen löschen

Bestimmte Leerzeichen löschen

Leerzeichen am Anfang, Leerzeichen am Ende und doppelte Leerzeichen in der Zeichenfolge werden gelöscht. Diese Leerzeichen haben keinerlei inhaltliche Bedeutung.

vorher	nachher
"1 Bl. "	"1 Bl."
" Spatzen im Birnbaum"	"Spatzen im Birnbaum"

Tabelle 5.6: Normierungsfunktion: Bestimmte Leerzeichen löschen

Alle Leerzeichen löschen

Sämtliche Leerzeichen in der Zeichenfolge werden gelöscht. Die Leerzeichen trennen Wörter und erleichtern den Lesefluß. Zur Erkennung von unterschiedlichen Schreibweisen sind sie eher hinderlich.

vorher	nachher
“informations system”	“informationssystem”
“Dalitz, Wolfgang”	“Dalitz,Wolfgang”
“dalitz w”	“dalitzw”
“Akazienblüthen aus der Schweiz”	“AkazienblüthenausderSchweiz”
“Frankfurt am Main”	“FrankfurtamMain”

Tabelle 5.7: Normierungsfunktion: Alle Leerzeichen löschen

Abkürzungen löschen

Die Abkürzung “u.a.” (und andere) wird gelöscht. Diese Abkürzung ist nicht wichtig für die Dublettenkontrolle. Sie wird von Bibliothekaren benutzt, wenn sie nicht alle Verlagsorte aufnehmen.

vorher	nachher
“Berlin u.a.”	“Berlin ”
“Stuttgart [u.a.]”	“Stuttgart []”

Tabelle 5.8: Normierungsfunktion: Abkürzung “u.a.” löschen

Abkürzungen ausschreiben

Häufig auftretende Abkürzungen wie “*Aufl.*”, “*Verl.*” etc. werden ausgeschrieben, und der Punkt am Ende der Abkürzung wird gelöscht. Diese Funktion ist sehr aufwendig und benötigt viel Rechenzeit. Für jede Sprache (deutsch, englisch, französisch etc.) müssen die gängigen Abkürzungen festgelegt werden.

vorher	nachher
“Greifenverl.”	“Greifenverlag”
“Ungekürzte Ausg.”	“Ungekürzte Ausgabe”
“Aufl.”	“Auflage”
“Freie Univ., Fachbereich Mathematik”	“Freie Universität, Fachbereich Mathematik”
“Aufbau-Verl.”	“Aufbau-Verlag”
“Dt. Ärzte-Verl.”	“Deutscher Ärzte-Verlag”

Tabelle 5.9: Normierungsfunktion: Abkürzungen ausschreiben

Trunkieren nach Länge

Es wird nur der Anfang der Zeichenfolge verwendet. Der Rest wird abgeschnitten. Dadurch können auf einfache Art viele Schreibvarianten auf eine Form reduziert werden.

Nach 5 Zeichen abschneiden:

vorher	nachher
“Frankfurt am Main”	“Frank”
“Görlitz”	“Görli”
“Bonn”	“Bonn”

Tabelle 5.10: Normierungsfunktion: Trunkieren nach Länge, 5 Zeichen

Trunkieren nach definierten Trennzeichen

Tritt ein bestimmtes Zeichen (Schrägstrich, Semikolon, Komma) auf, wird dieses Zeichen und alle nachfolgenden ignoriert. Das Semikolon trennt z.B. die Verlagsorte voneinander.

vorher	nachher
“Berlin ; Weimar”	“Berlin ”
“Cambridge ; Köln [u.a.]”	“Cambridge”

Tabelle 5.11: Normierungsfunktion: Trunkieren nach definierten Trennzeichen

Zahlen suchen

Es wird nach einer (arabischen) Zahl in der Zeichenfolge gesucht und die erste gefundene verwendet. Alternativ: sind mehrere Zahlen in der Zeichenfolge vorhanden, so wird die größte genommen. Diese Funktion wird für die Attribute *Jahr*, *Seitenzahl* und *Auflage* benötigt, in denen nur Zahlen ausgewertet werden.

vorher	nachher
“1. Aufl.”	“1”
“1. Aufl., 1. - 230. Tsd.”	“1”
“21.-30. Tsd.”	“21”
“1990”	“1990”
“1911 - 1916”	“1911”

Tabelle 5.12: Normierungsfunktion: Zahlen suchen, erste Zahl

vorher	nachher
“1. Aufl.”	“1”
“1. Aufl., 1. - 230. Tsd.”	“230”
“21.-30. Tsd.”	“30”
“1990”	“1990”
“1911 - 1916”	“1916”

Tabelle 5.13: Normierungsfunktion: Zahlen suchen, größte Zahl

Die größte Zahl zu finden, ist komplexer und rechenintensiver als die erste Zahl zu finden.

5.1.2 Spezialfälle

ISBN

In der ISBN-Nummer sind die Zahlen 0 bis 9, der Buchstabe “X” und der Bindestrich als gültige Werte definiert. Alle anderen Zeichen sind nicht erlaubt (siehe auch [Ott94]).

Kleinere Tippfehler bei der Eingabe werden korrigiert:

- Ein kleines “x” wird durch ein großes “X” ersetzt.
- Die Buchstaben “O” und “o” werden durch eine Null (“0”) ersetzt.
- Die Bindestriche werden entfernt. Sie enthalten keine für die Dublettenkontrolle notwendige Information - im Gegenteil, häufig sind die Bindestriche bei der Erfassung falsch eingegeben worden.

Zum Schluß wird eine Zahl mit 10 Ziffern (einschließlich des Buchstaben "X") zurückgegeben. Falls keine Zahl mit 10 Stellen gefunden wird, ist der Wert des Attributes ISBN leer.

vorher	nachher
"ISBN 3-89602-119-2 Pp. : DM 38.00"	"3896021192"
"ISBN 3-928861-23-9"	"3928861239"
"zkart. (Pr. nicht mitget.)"	""

Tabelle 5.14: Normierungsfunktion: ISBN

Jahr

Das Jahr ist vierstellig und beginnt mit der Ziffer "1" oder "2" für Jahreszahlen ab dem Jahr 1000 (1999, 1834). Ältere Bücher sind praktisch nicht vorhanden (siehe auch Normierungsfunktion *Zahlen suchen*).

vorher	nachher
"1985"	"1985"
"[1993]"	"1993"
"c 1993"	"1993"
"(1995)"	"1995"
"[circa 1980]"	"1980"
"([1990]) - "	"1990"

Tabelle 5.15: Normierungsfunktion: Jahr

Autor

Vom Vornamen wird nur der erste Buchstabe übernommen. Die Namen der Autoren sind nach dem Schema "*Nachname, Vorname*" definiert. *Nachname* und *Vorname* sind durch ein Komma getrennt. Die Vornamen der Autoren werden öfters von Verlagen abgekürzt. Durch die Normierung wird dann erkannt, daß "*luegger, j*" und "*luegger, joachim*" denselben Autor bezeichnen.

vorher	nachher
"Lohrum, Rita"	"Lohrum, R"
"Reich-Ranicki, Marcel"	"Reich-Ranicki, M"
"Todt, Wilfried Wolfgang"	"Todt, W"

Tabelle 5.16: Normierungsfunktion: Autor

5.2 Attributspezifische Normierung in ZACK

Da nicht alle der oben genannten Normierungsfunktionen für alle Attribute sinnvoll sind, erfolgt die Normierung der Zeichenfolgen attributspezifisch - für jedes Attribut wird festgelegt, welche Normierungsfunktionen in welcher Reihenfolge im System *ZACK* angewendet wird (siehe auch [Rus99a]).

Die Reihenfolge der verwendeten Normierungsfunktionen ist wichtig. Zum Beispiel darf man die Sonderzeichen erst löschen, nachdem man die Umlaute konvertiert hat.

Autor

Verwendete Normierungsfunktionen:

1. Groß-und Kleinschreibung
2. Eckige Klammern
3. Umlaute konvertieren
4. Vom Vornamen wird nur der erste Buchstabe genommen
5. Sonderzeichen löschen
6. Überflüssige Leerzeichen löschen

vorher	nachher
“Dalitz, Wolfgang”	“dalitz w”
“Mozart, Wolfgang Amadeus”	“mozart w”
“Mendelssohn Bartholdy, Felix”	“mendelssohn bartholdy f”
“Kästner, Erich”	“kaestner e”
“Goethe, Johann Wolfgang ¬von¬”	“goethe j”
“Georgius ¡de Hungaria¿”	“georgius de hungaria”
“DJ BoBo ¬[voc]¬”	“dj bobo”
“Pythagoras”	“pythagoras”
“Loca ¬La¬”	“loca la”

Tabelle 5.17: *ZACK*: Normierung Attribut Autor

Titel

Verwendete Normierungsfunktionen:

1. Groß-und Kleinschreibung
2. Eckige Klammern
3. Umlaute konvertieren
4. Sonderzeichen löschen
5. Überflüssige Leerzeichen löschen

vorher	nachher
“Statistische Berichte”	“statistische berichte”
“Ägypten”	“aegypten”
“Stille Nacht, heilige Nacht”	“stille nacht heilige nacht”
“Verwaltungsbericht ...”	“verwaltungsbericht”
“¬Das¬ Neue Testament”	“das neue testament”
“MS-DOS-Handbuch”	“ms dos handbuch”
“Lyrik & Prosa”	“lyrik prosa”
“¬Die¬ dollsten Dinger für ¬10¬ [zehn] Finger”	“die dollsten dinger fuer 10 finger”
“Elvis Presley - Stereo [19]57”	“elvis presley stereo 57”

Tabelle 5.18: *ZACK*: Normierung Attribut Titel

Seitenzahl

Normierungsfunktion erste Zahl.

vorher	nachher
"48 S."	"48"
"2 Mikrofiches"	"2"
"Medienkombination"	""
"[12] S."	"12"
"12 St. in Umschlag"	"12"
"1 Videokassette [VHS] (60 Min.)"	"1"

Tabelle 5.19: *ZACK*: Normierung Attribut Seitenzahl

Verlagsort

Verwendete Normierungsfunktionen:

1. Groß-und Kleinschreibung
2. Eckige Klammern
3. Umlaute konvertieren
4. Trunkieren nach Zeichen - alles ab dem ersten Komma, Semikolon oder Schrägstrich ignorieren
5. Sonderzeichen löschen
6. Trunkieren nach Länge - nur die ersten 5 Zeichen
7. Überflüssige Leerzeichen löschen

vorher	nachher
"Ötztal"	"oetzt"
"Reinbek bei Hamburg"	"reinb"
"Hamburg ; München"	"hambu"
"München; Hamburg"	"muenc"
"Halle (Saale)"	"halle"
"Bern ; Stuttgart"	"bern"
"Freiburg [Breisgau]"	"freib"
"Karl-Marx-Stadt"	"karl"
"Bonn ; Albany ¬[u.a.]¬	"bonn"

Tabelle 5.20: *ZACK*: Normierung Attribut Verlagsort

Alternativ: am besten wäre es wahrscheinlich, einzelne Wörter zu extrahieren und mit logischen *ODER* zu verbinden. Zum Beispiel: *Berlin* oder *Muenchen*. Dies ist aber wesentlich aufwendiger.

Verlag

Verwendete Normierungsfunktionen:

1. Groß-und Kleinschreibung
2. Eckige Klammern
3. Trunkieren nach Zeichen - alles ab dem ersten Komma, Semikolon oder Schrägstrich ignorieren
4. Umlaute konvertieren
5. Sonderzeichen löschen
6. Abkürzungen ausschreiben
7. Trunkieren nach Länge - nur die ersten 5 Zeichen
8. Alle Leerzeichen löschen

vorher	nachher
“Springer”	“sprin”
“EMI-Electrola”	“emiel”
“¬de¬ Gruyter”	“degru”
“Müller”	“muell”
“mvg-Verl.”	“mvgve”
“Verl. Die Blaue Eule”	“verla”
“Verlag Die blaue Eule“	“verla”
“Volk u. Wissen”	“volku”
“I & Ear”	“iear”
“Siemens-Aktienges., ¬[Abt. Verl.]¬”	“sieme”

Tabelle 5.21: *ZACK*: Normierung Attribut Verlag

Jahr

Normierungsfunktion Jahr.

vorher	nachher
“1990”	“1990”
“[1993]”	“1993”
“c 1993”	“1993”
“(1995) -”	“1995”
“[1990 ?”	“1990”
“[circa 1980]”	“1980”
“([1986 ?])”	“1986”

Tabelle 5.22: *ZACK*: Normierung Attribut Jahr

Auflage

Normierungsfunktion erste Zahl oder Text. Text entspricht:

1. Groß-und Kleinschreibung
2. Umlaute konvertieren
3. Sonderzeichen löschen
4. Überflüssige Leerzeichen löschen

vorher	nachher
"Orig.-Ausg."	"orig ausg"
"2., überarb. Aufl."	"2"
"Orig.-Ausg., 1. Aufl."	"1"
"[3. Aufl.]"	"3"
"[Spielpartitur]"	"spielpartitur"

Tabelle 5.23: *ZACK*: Normierung Attribut Auflage

ISBN

Nur Normierungsfunktion ISBN.

vorher	nachher
"ISBN 3-928861-23-9"	"3928861239"
"ISBN 3-87096-149-x"	"387096149X"
"ISBN 3-89602-119-2 Pp. : DM 38.00"	"3896021192"
"ISBN 3-540-56740-2 = 0-387-56740-2"	"3540567402"
"ISBN 3-468-20270-9 flexibler	"3468202709"
"ISBN 0-387-90148-5 (New York ...) Pp."	"0387901485"

Tabelle 5.24: *ZACK*: Normierung Attribut ISBN

Die ISBN-Nummer läßt sich sehr gut normieren und ist auch deshalb eines der wichtigsten Attribute bei Dublettenkontrolle.

5.3 Test der Normierung des Attributes Autor mit Daten aus unterschiedlichen Datenbanken

In diesem Abschnitt wird untersucht, ob die in *ZACK* verwandte Normierung des Attributes *Autor* korrekte oder falsche Ergebnisse ergibt. Dazu wird die Anfrage *titel=perl* an die Datenbanken der Deutschen Bibliothek (DDB), des Gemeinsamen Bibliotheksverbundes (GBV), des Bibliotheksverbundes Bayern (BVB) und der Technischen Universität Braunschweig (TUBS) gestellt. Es werden insgesamt 345 Treffer gefunden und alle Datensätze im Format MAB2 geholt.

Von 345 Datensätzen enthalten 267 das Autor-Feld 100, die restlichen 78 Datensätze haben kein Feld 100. Nach den ersten Normierungschritten (Kleinschreibung, eckige Klammern löschen, überflüssige Leerzeichen entfernen) bleiben 105 unterschiedliche Autoren übrig.

Im Feld 100 wird jetzt eine zusätzliche Normierung vorgenommen. Vom Vornamen wird nur noch der erste Buchstabe genommen, aus "*gruner, klaus*" wird "*gruner, k*". Die Anzahl der unterschiedlichen Autoren reduzierte sich dadurch um 6 Autoren, von 105 auf 99.

In 5 der 6 Fälle wurde die unterschiedliche Schreibweise der Autorennamen richtig erkannt. In nur einem Fall wurden zwei verschiedene Personen zusammengefaßt. Die Beurteilung, ob zwei Namen denselben Autor bezeichnen, wurde anhand der MAB-Datensätze getroffen. Dazu wurden andere Attribute herangezogen, wie z.B. Titel, ISBN-Nummer, Verlag und Jahr.

Die gefundenen Autoren im Einzelnen:

Nur Anfangsbuchstabe des Vornamens	Ursprüngliche Schreibweisen	Anmerkung
"perl,m"	"perl, martin lewis"	unterschiedliche Personen
	"perl, matthias"	
"quigley,e"	"quigley, e."	dieselbe Person, unterschiedliche Schreibweise
	"quigley, ellen"	
	"quigley, ellie"	
"till,d"	"till, dave"	dieselbe Person unterschiedliche Schreibweise
	"till, david"	
"wall,l"	"wall, l."	dieselbe Person unterschiedliche Schreibweise
	"wall, larry"	
"Âsliwa,m"	"Âsliwa, micha/l"	dieselbe Person unterschiedliche Schreibweise
	"Âsliwa, michaø"	

Tabelle 5.25: Manuelle Normierung des Attributes Autor bei verteilter Suche

5.4 Test der Normierung der Zeichenfolge Frankfurt

In diesem Abschnitt werden die unterschiedlichen Schreibweisen von *Frankfurt am Main* und *Frankfurt an der Oder* untersucht. Dazu wurden in den 2,5 Millionen DNB-Datensätzen der Deutschen Bibliothek im MAB-Feld 410 nach dem Wort *Frankfurt* gesucht.

Frankfurt an der Oder

Straßennamen wie z.B. "Frankfurt/Oder, [Beerenweg 14]" wurden ignoriert und als "Frankfurt/Oder" gewertet. Leerzeichen am Anfang oder Ende wurden ignoriert.

```
Frankfurt (Oder)
Frankfurt an der Oder
Frankfurt, Oder
Frankfurt/O.
Frankfurt/Oder
Frankfurt/Oder
[Frankfurt (Oder)]
[Frankfurt Oder]
[Frankfurt/O]
[Frankfurt/Oder]
```

Frankfurt am Main

Straßennamen wie z.B. "Frankfurt a.M., Wiener Str. 61" werden in dieser Auswertung ignoriert und als "Frankfurt a.M." gewertet. Stadtteile wie z.B. "Frankfurt/Rödelheim" oder "Frankfurt/Sulzbach" sind ebenfalls nicht aufgeführt. Fehlende Klammern werden nicht gewertet: "Frankfurt (Main])" ⇒ "Frankfurt (Main)". Leerzeichen am Anfang oder Ende der Zeichenfolge werden ignoriert.


```
Frankfurt
Frankfurt (M)
Frankfurt (M.)
Frankfurt (Main)
Frankfurt ([Main)]
Frankfurt ([Main])
Frankfurt (a.M.)
Frankfurt (am Main)
Frankfurt (main)
Frankfurt / M.
Frankfurt / Main
Frankfurt /M.
Frankfurt /Main
Frankfurt M
Frankfurt M.
Frankfurt Main
Frankfurt [(am Main)]
Frankfurt [/Main]
Frankfurt [a.M.]
Frankfurt [am Main]
Frankfurt a Main
Frankfurt a. M.
Frankfurt a. Main
Frankfurt a./M.
Frankfurt a.M
Frankfurt a.M.
Frankfurt am M.
Frankfurt am Main
Frankfurt am Mainn
Frankfurt am Man
Frankfurt am main
Frankfurt na Majn
Frankfurt na Majne
Frankfurt nad Menem
Frankfurt on Main
Frankfurt on the Main
Frankfurt(M)
Frankfurt(Main)
Frankfurt/ M
Frankfurt/M
Frankfurt/M.
Frankfurt/M[ain]
Frankfurt/Main
Frankfurt/Main.
Frankfurt/Meno
Frankfurt/a.M.
Frankfurt/m
Frankfurt/m.
Frankfurt[/M.]
Frankfurt[/Main]
Frankfurtam Main
```

Anmerkung: in einigen Fällen wurde “Frankfurt am Main” bzw. nur der Zusatz “am Main” übersetzt.

```
Frankfurt na Majn
Frankfurt na Majne
Frankfurt nad Menem
Frankfurt on Main
Frankfurt on the Main
```

Die Anzahl der unterschiedlichen Schreibweisen kann deutlich durch die folgenden Schritte reduziert werden:

1. Ersetzen der Zeichen Klammer auf, Klammer zu, Punkt, Schrägstrich durch Leerzeichen, danach Ersetzen von mehreren aufeinanderfolgenden Leerzeichen durch ein einzelnes Leerzeichen. Löschen von Leerzeichen am Anfang und am Ende.

   Beispiel 1) “Frankfurt / Main.” wird zu “Frankfurt Main ” und dann zu “Frankfurt Main”

   Beispiel 2) “Frankfurt/Main” wird zu “Frankfurt Main”

2. Kleinschreibung. “Frankfurt M” wird zu “frankfurt m”

```
frankfurt
frankfurt a m
frankfurt a main
frankfurt am m
frankfurt am main
frankfurt am mainn
frankfurt am man
frankfurt m
frankfurt m ain
frankfurt main
frankfurt meno
frankfurt na majn
frankfurt na majne
frankfurt nad menem
frankfurt on main
frankfurt on the main
frankfurtam main
```

Es verbleiben 17 unterschiedliche Schreibweisen. Ein weiterer Normierungschritt ist das Löschen sämtlicher Leerzeichen. Beispiel:

vorher	nachher
“frankfurt a m”	“frankfurtam”
“frankfurt m”	“frankfurtm”
“frankfurt main”	“frankfurtmain”

Tabelle 5.26: Normierung von *Frankfurt*, ohne Leerzeichen

Dadurch reduziert sich die Anzahl der unterschiedliche Schreibweisen auf 15.

```
frankfurt          frankfurtmain
frankfurtam        frankfurtmeno
frankfurtamain     frankfurtnadmenem
frankfurtamm       frankfurtnamajn
frankfurtammain    frankfurtnamajne
frankfurtammainn   frankfurtonmain
frankfurtamman     frankfurtonthemain
frankfurtm
```

Beschränkt man den Vergleich auf die ersten 5 Buchstaben, so bleibt nur noch eine Schreibweise übrig:

vorher	nachher
"frankfurt"	"frank"
"frankfurtammain"	"frank"

Tabelle 5.27: Normierung von *Frankfurt*, nur die ersten 5 Buchstaben

Anmerkung: für diesen Test wurde nach Datensätzen gesucht, die das Wort "Frankfurt" (Groß-und Kleinschreibung wird ignoriert) im Feld Verlagsort haben. Tippfehler, Abkürzungen oder Übersetzungen wurden deshalb nicht erfaßt. Beispiele aus den Daten der Deutschen Bibliothek sind:

Übersetzungen

```
Frankfort on the Main
Frankfort/M.
Frankfort-s. Main
Frankfort a.d. Main
```

Tippfehler

Um die Tippfehler und Übersetzungen zu finden, wurde nach dem Wort "frank" gesucht und aus der Ergebnismenge die Treffer "frankfurt", "franken" und "frankreich" herausgefiltert.

```
Frankurt am Main
Frankfzrt (Main)
Frankurt am Main
Frankkfurt/M.
Frankfut am Main
Frankfur a.M.
```

Abkürzung

In seltenen Fällen wird Frankfurt am Main auch mit "Ffm" abgekürzt. Beispiel:

```
F[rank]f[urt/]M[ain]
Ffm [Frankfurt, Main]
```

Zusammenfassung

Es gibt unzählige Schreibweisen von "*Frankfurt*". Mit der Reduzierung der Zeichenlänge auf 5 Zeichen kann man die unterschiedlichen Schreibweisen sicher erkennen.

5.5 Test der attributspezifischen Normierung in ZACK mit Datensätzen der Deutschen Bibliothek

Zunächst wird untersucht, wie sich die Normierung auf die Datensätze aus einer Datenbank auswirkt. Dazu wurden 2,5 Mio Datensätze der Deutschen Bibliothek analysiert. Es wurde erwartet, daß es nur wenige unterschiedliche Schreibweisen gibt. In der DDB arbeiten die Katalogisierer nach festen internen Regeln, es wird dasselbe Bibliothekssystem verwendet. Es wird davon ausgegangen, daß die Deutsche Bibliothek in regelmäßigen Abständen ihren Datenbestand pflegt und Fehler oder abweichende Schreibweisen behebt.

Im einzelnen wurden die Datenlieferungen der Deutschen Bibliothek aus den Jahren 1986 bis 1997 sowie die Nachlieferungen 1998 und 1999 verwendet (siehe im Anhang A, Seite 102). Insgesamt wurden 2.535.095 Datensätze analysiert. In diesem Test wurde kein Unterschied zwischen den verschiedenen Satztypen gemacht.

Feld	Normierungsstufe	Anzahl	Werte	Durchschnitt
100 (Autor)	0	1.648.452	697.409	2,36
	1	1.648.452	619.739	2,66
	2	1.648.452	428.209	3,85
	3	1.648.452	427.325	3,86
331 (Titel)	0	2.191.768	1.531.777	1,43
	1	2.190.296	1.507.645	1,45
	2	2.190.296	1.507.645	1,45
	3	2.190.296	1.495.318	1,46
403 (Auflage)	0	612.401	66.340	9,23
	1	442.399	1.401	315,77
	2	442.399	1.401	315,77
	3	442.399	1.401	315,77
410 (Verlagsort)	0	1.771.550	40.513	3,73
	1	1.770.365	32.887	53,66
	2	1.770.265	8.692	203,66
	3	1.770.265	8.692	203,66
412 (Verlag)	0	1.771.550	161.518	10,97
	1	1.770.010	149.508	11,84
	2	1.769.920	41.515	42,63
	3	1.769.920	41.515	42,63
425 (Jahr)	0	2.107.365	1.462	1.441,43
	1	2.107.349	221	9535,52
	2	2.107.349	221	9535,52
	3	2.107.349	221	9535,52
433 (Seitenzahl)	0	1.961.648	130.849	14,99
	1	1.936.649	2.903	667,12
	2	1.936.649	2.903	667,12
	3	1.936.649	2.903	667,12
540 (ISBN)	0	1.413.387	937.000	1,51
	1	974.097	846.084	1,15
	2	974.097	846.029	1,15
	3	974.097	846.029	1,15

Tabelle 5.28: *ZACK*: Normierung der DNB Datensätze

Für die Normierung wurden die MAB-Felder 100 (Autor), 331 (Titel), 403 (Auflage), 410 (Verlagsort), 412 (Verlag), 425 (Jahr), 433 (Seitenzahl), 540 (ISBN) untersucht.

Legende Normierungsstatistik

Nachfolgend wird die Tabelle zur Normierungsstatistik der Datensätze der Deutschen Bibliothek erläutert.

Normierungsstufe: angewandte Normierungsverfahren für die Felder.

Stufe 0: Keine Normierung, die Felder werden unverändert gelassen.

Stufe 1: Normierung des Zeichensatzes, Entfernung von Kommentaren und überflüssigen Leerzeichen. Es findet praktisch kein Informationsverlust statt. Im einzelnen: Großbuchstaben werden zu Kleinbuchstaben umgesetzt, alles zwischen Klammern und Nicht-Sortierzeichen gelöscht, Jahr ist 4-stellig, ISBN besteht aus 10 Ziffern oder dem Buchstaben X und Bindestrichen, Seitennummern und Auflage sind Zahlen; Umlaute nach ASCII umgewandelt, Sonderzeichen entfernt, doppelte Leerzeichen und Leerzeichen am Zeilenanfang und -ende entfernt.

Stufe 2: Normierung von Zeichenfolgen. Es findet ein minimaler Informationsverlust statt. Im einzelnen: Nur den ersten Buchstaben des Vornamens im Feld Autor speichern: “wall, larry” ⇒ “wall l” Nur den ersten Verlagsort speichern: “berlin ; new york ; hongkong” ⇒ “berlin”. Nur die ersten 5 Zeichen vom Verlag speichern. Nur die ersten 5 Zeichen vom Verlagsort speichern. Bindestriche in ISBN entfernen, kleines “x” durch großes “X” ersetzen, den Buchstaben “o” durch die Ziffer Null “0” ersetzen. (Anmerkung: erst danach werden Umlaute, Sonderzeichen und Leerzeichen wie unter 1 angegeben umgewandelt bzw. entfernt.)

Stufe 3: Normierung von Zeichenfolgen. Es findet ein Informationsverlust statt: Im einzelnen: Sämtliche Leerzeichen im Titel und Autor löschen. Nur die ersten 50 Zeichen vom Titel abspeichern.

Die Normierung geschieht stufenweise. Stufe 2 schließt alle Normierungen von Stufe 1 mit ein; Stufe 3 schließt alle Normierungen von Stufe 2 und Stufe 1 mit ein.

Anzahl: Anzahl der Datensätze, in denen das betreffende Feld existiert und nicht leer ist. Dieser Wert kann sich von Normierungsstufe zu Normierungsstufe verringern, da unter Umständen ein Wert als ungültig erkannt wird und das Feld als leer gewertet wurde.

Beispiel: im Feld 540 (ISBN) steht häufig die Einbandart und/oder der Preis. Diese Werte können nicht für einen ISBN-Vergleich herangezogen werden und wurden deshalb nicht gewertet. Beispiel:

“`in Mappe :  DM 29.00`”
“`kart.  (nicht im Sortimentsbuchh.)`”

Werte: Anzahl der unterschiedlichen Werte im betreffenden Feld. Beispiel: Im Feld 425, Normierungsstufe 0, gibt es 1.462 verschiedene Jahreszahlen und in der dritten Normierungsstufe 221 verschiedene Jahreszahlen. Die Anzahl der unterschiedlichen Jahreszahlen hat sich durch die Normierung deutlich verringert.

Durchschnitt: Quotient aus Anzahl der Datensätze und den unterschiedlichen Werten. Beispiel: Im Feld 100 (Autor), Normierungsstufe 0 kommen im Durchschnitt auf einen Autor 2,36 Datensätze. In der dritten Normierungsstufe steigt der Durchschnitt auf 3,85 Datensätze pro Autor.

Analyse und Ergebnisse

Wie erwartet, hielten sich die unterschiedlichen Schreibweisen in Grenzen. Zum Beispiel tritt der Fall *kleines "x" in ISBN-Nummer* nur 55 mal auf. Die Anzahl unterschiedlicher Titel hat sich durch die Normierung nur minimal verändert. Die Anzahl unterschiedlicher Autoren reduzierte sich in der ersten Stufe der Normierung um 10%. Die Verkürzung des Vornamens bringt deutliche Vorteile (Reduzierung der Anzahl unterschiedlicher Autoren um 39%) - jedoch mit dem Risiko, unterschiedliche Autoren zusammenzufassen.

Eine Normierung von Verlagsort und Verlag bringt deutliche Vorteile. Die Normierung von Auflage, Seitenzahl und Jahr ist zwingend erforderlich, da es viele unterschiedliche Schreibweisen gibt. Die ISBN-Nummer muß normiert werden, da in vielen Fällen auch der Preis im Feld 540 (ISBN) steht.

5.6 Test der attributspezifischen Normierung mit Datensätzen unterschiedlicher Herkunft

In diesem Abschnitt werden die Normierungsfunktionen in *ZACK* auf Datensätze unterschiedlicher Herkunft (DDB, BVB, GBV, TUBS) angewandt und statistisch analysiert. Dazu werden 7 unterschiedliche Anfragen an die Datenbanken gestellt und die Datensätze im Format MAB2 geholt.

Anfrage	Anzahl der Datensätze
titel=akazie	60
titel=birnbaum	342
titel=perl	345
titel=karstadt	61
titel=pankow	212
autor=dalitz,wolfgang	28
autor=rusch,beate	9

Tabelle 5.29: Anfragen an mehrere Datenbanken

Es wird erwartet, daß die Anzahl der unterschiedlichen Schreibweisen höher ist als in nur einer Datenbank. Jede dieser Bibliotheken bzw. Bibliotheksverbünde hat eigene Regelinterpretationen hinsichtlich der Erfassung. Die Bibliotheken verwenden Bibliothekssysteme unterschiedlicher Hersteller.

Für die Normierung wurden die MAB-Felder 100 (Autor), 331 (Titel), 410 (Verlagsort), 412 (Verlag), 425 (Jahr) und 540 (ISBN) untersucht.

Autor

Normierungs- funktion	birnbaum	karstadt	pankow	akazie	perl	rusch, beate	dalitz, wolfgang
Anzahl der Datensätze	342	61	212	60	345	9	28
Anzahl Feld Autor	283	33	106	50	267	6	20
Anzahl Werte	98 100%	22 100%	66 100%	32 100%	107 100%	2 100%	3 100%
Klein- schreibung	98 100%	22 100%	66 100%	32 100%	106 99,1%	2 100%	3 100%
ohne Klammern	97 99,0%	22 100%	66 100%	32 100%	106 99,1%	2 100%	3 100%
Vorname gekürzt	87 88,8%	22 100%	61 92,4%	30 93,8%	99 92,5%	3 100%	3 100%
Sonder- zeichen	86 87,8%	22 100%	61 92,4%	30 93,8%	99 92,5%	3 100%	3 100%

Tabelle 5.30: *ZACK*: Normierung nach verteilter Suche: Attribut Autor

Beispiel: Bei der verteilten Suche nach dem Titel *birnbaum* werden 342 Treffer gefunden. 283 davon besitzen das Attribut Autor. Es gibt 98 unterschiedliche Schreibweise für die Autoren (dies ist die Basis für die weitere Prozentrechnung). Nach der Kleinschreibung sind es immer noch 98 unterschiedliche Autoren. Nach Entfernung der Klammern verbleiben nur noch 97 (99,0%) Autoren. Nimmt man vom Autor nur noch den Nachnamen und den ersten Buchstaben des Vornamens, verringert sich die Anzahl der unterschiedlichen Autoren auf 87 (88,8%). Nach Entfernen der Sonderzeichen verbleiben insgesamt 86 (87,8%) unterschiedliche Autoren übrig.

Verlag

Normierungs- funktion	birnbaum	karstadt	pankow	akazie	perl	rusch, beate	dalitz, wolfgang
Anzahl der Datensätze	342	61	212	60	345	9	28
Anzahl Feld	255	40	136	54	279	7	22
Anzahl Werte	131 100%	22 100%	72 100%	31 100%	96 100%	4 100%	6 100%
Klein- schreibung	131 100%	22 100%	72 100%	31 100%	89 92,7%	4 100%	5 83,3%
ohne Klammern	129 98,5%	20 90,9%	71 98,6%	31 100%	87 90,6%	4 100%	3 50%
Abge- schnitten	128 97,7%	19 86,4%	69 95,8%	31 100%	83 86,5%	4 100%	3 50%
Sonder- zeichen	124 94,7%	18 81,8%	69 95,8%	31 100%	81 84,4%	4 100%	3 50%
nur 8 Zeichen	111 84,7%	15 68,2%	63 87,5%	28 90,3%	76 79,2%	3 75%	3 50%
mit Tipp- fehlern	105 80,2%	13 59,1%	60 83,3%	26 83,9%	72 75,0%	3 75%	3 50%
nur 5 Zeichen	107 81,7%	13 59,1%	58 80,6%	28 90,3%	73 76,0%	3 75%	3 50%

Tabelle 5.31: *ZACK*: Normierung nach verteilter Suche: Attribut Verlag

Jahr

Normierungs- funktion	birnbaum	karstadt	pankow	akazie	perl	rusch, beate	dalitz, wolfgang
Anzahl der Datensätze	342	61	212	60	345	9	28
Anzahl Feld	323	53	196	58	320	7	24
Anzahl Werte	81 100%	27 100%	54 100%	29 100%	53 100%	2 100%	6 100%
nur vier Ziffer	79 97,5%	24 88,9%	49 90,7%	29 100%	49 92,5%	2 100%	6 100%

Tabelle 5.32: *ZACK*: Normierung nach verteilter Suche: Attribut Jahr

Titel

Normierungs- funktion	birnbaum	karstadt	pankow	akazie	perl	rusch, beate	dalitz, wolfgang
Anzahl der Datensätze	342	61	212	60	345	9	28
Anzahl Feld Titel	336	60	209	59	334	7	25
Anzahl Werte	163 100%	46 100%	126 100%	40 100%	169 100%	3 100%	12 100%
Klein- schreibung	163 100%	46 100%	126 100%	40 100%	158 93,5%	3 100%	12 100%
ohne Klammern	136 100%	46 100%	125 99,2%	40 100%	157 92,9%	3 100%	12 100%
Sonder- zeichen	149 91,5%	43 93,5%	121 96,%	38 95,5%	153 90,5%	3 100%	12 100%
nur 50 Zeichen	149 91,5%	43 93,5%	119 94,5%	38 95,5%	152 89,9%	3 100%	12 100%
mit Tipp- fehlern	146 89,6%	43 93,5%	117 92,9%	36 90%	149 88,2%	3 100%	12 100%
nur 40 Zeichen	146 89,6%	43 93,5%	119 94,5%	38 95%	152 89,9%	3 100%	12 100%
mit Tipp- fehlern	144 88,3%	43 93,5%	116 92,1%	36 90%	150 88,8%	3 100%	11 91,7%
nur 25 Zeichen	141 86,5%	43 93,5%	116 92,1%	36 90%	150 88,8%	3 100%	11 91,7%
mit Tipp- fehlern	141 86,5%	43 93,5%	112 88,9%	35 87,5%	148 87,6%	3 100%	11 91,7%

Tabelle 5.33: *ZACK*: Normierung nach verteilter Suche: Attribut Titel

Verlagsort

Normierungs-funktion	birnbaum	karstadt	pankow	akazie	perl	rusch, beate	dalitz, wolfgang
Anzahl der Datensätze	342	61	212	60	345	9	28
Anzahl Feld	286	44	174	57	289	7	23
Anzahl Werte	105 100%	17 100%	41 100%	29 100%	105 100%	2 100%	4 100%
Klein-schreibung	105 100%	17 100%	41 100%	29 100%	104 99,0%	2 100%	4 100%
ohne Klammern	93 88,6%	14 82,4%	32 78,0%	28 96,6%	92 87,6%	2 100%	4 100%
Abge-schnitten	80 76,2%	14 82,4%	25 61,0%	25 86,2%	64 61,0%	2 100%	4 100%
Sonder-zeichen	79 75,2%	14 82,4%	25 61,0%	25 86,2%	63 60,0%	2 100%	4 100%
nur 8 Zeichen	74 70,5%	14 82,4%	24 58,5%	24 82,8%	63 60,0%	2 100%	4 100%
mit Tipp-fehlern	69 65,7%	13 75,5%	21 51,2%	23 79,3%	62 59,0%	2 100%	2 50%
nur 5 Zeichen	73 69,5%	13 76,5%	21 51,2%	24 82,8%	62 59,0%	2 100%	2 50%

Tabelle 5.34: *ZACK*: Normierung nach verteilter Suche: Attribut Verlagsort

ISBN

Normierungs-funktion	birnbaum	karstadt	pankow	akazie	perl	rusch, beate	dalitz, wolfgang
Anzahl der Datensätze	342	61	212	60	345	9	28
Anzahl Feld	172	12	89	39	249	6	12
Anzahl Werte	124 100%	9 100%	59 100%	30 100%	189 100%	5 100%	5 100%
nur Ziffern	62 50%	2 22,2%	25 42,4%	13 43,3%	113 59,8%	2 40%	2 40%
x groß schreiben	61 49,2%	2 22,2%	24 40,7%	12 40,0%	107 56,6%	2 40%	2 40%
ohne Binde-strich	61 49,2%	2 22,2%	24 40,7%	12 40,0%	94 49,7%	2 40%	2 40%

Tabelle 5.35: *ZACK*: Normierung nach verteilter Suche: Attribut ISBN

Legende Normierung nach verteilter Suche

Anzahl der Datensätze: Gibt die Anzahl der Treffer an, die in allen Datenbanken gefunden wurde.

Anzahl Feld: Gibt die Anzahl der Datensätze an, in denen das betreffende Feld vorhanden ist.

Anzahl Werte: Gibt an, wieviele unterschiedliche Werte es für das betreffende Attribut gibt - z.B. wieviele unterschiedliche Autoren in allen Datensätzen gefunden wurden.

Kleinschreibung: Großbuchstaben werden in Kleinbuchstaben umgewandelt, siehe Normierungsfunktion *Groß-und Kleinschreibung* (Kapitel 5.1.1, Seite 41).

Ohne Klammer: Alle Zeichen zwischen runden und eckigen Klammern, und die Klammern selbst werden gelöscht. Siehe Normierungsfunktion *Eckige Klammern* (Kapitel 5.1.1).

Vorname gekürzt: Siehe Normierungsfunktion *Autor* (Kapitel 5.1.1).

Sonderzeichen: Siehe Normierungsfunktion *Sonderzeichen löschen* (Kapitel 5.1.1).

nur X Zeichen: Siehe Normierungsfunktion *Trunkieren nach Länge* (Kapitel 5.1.1).

nur 4 Ziffern: Siehe Normierungsfunktion *Jahr* (Kapitel 5.1.1).

Analyse und Ergebnisse

Beim Attribut *Autor* reduzierten sich die die unterschiedlichen Schreibweisen um rund 10%, beim *Titel* um 10-13%, beim Verlag um rund 20%, beim Verlagsort um rund 30%, beim Jahr um durchschnittlich 7% und bei der ISBN-Nummer um durchschnittlich 60%.

Die Verkürzung des Vornamens bringt große Vorteile - aber mit dem Risiko, verschiedene Autoren zusammenzufassen. Eine Normierung von Verlagsort und Verlag bringt deutliche Vorteile. Die Normierung von ISBN, Auflage, Seitenzahl und Jahr ist zwingend erforderlich.

Kapitel 6

Dublettenkontrolle

In diesem Kapitel wird beschrieben, wie die Dublettenerkennung in *ZACK* durchgeführt wird. Es werden die verwendeten Algorithmen, der benötigte Rechenaufwand in *ZACK* und die Ergebnisse der Dublettenkontrolle erläutert.

Im ersten Abschnitt 6.1 wird erklärt, **was Dubletten** sind und wie sie entstehen. In Abschnitt 6.2 wird eine **manuelle Dublettenkontrolle** durchgeführt. Es wird untersucht, inwieweit man anhand der Datensätze entscheiden kann, ob es sich um dieselben Werke handelt und welche Probleme dabei auftauchen. In Abschnitt 6.3 wird beschrieben, **wie die Dublettenkontrolle in ZACK durchgeführt** wird und welche Algorithmen dabei Verwendung finden. In Abschnitt 6.4 wird ein Tool vorgestellt, mit dessen Hilfe man eine **interaktive Dublettenkontrolle** durchführen kann. Der Benutzer kann die Dublettenkontrolle in *ZACK* online testen, optimieren und bewerten. In Abschnitt 6.5 wird untersucht, wie **effizient die in ZACK verwendeten Algorithmen** bei der Dublettenkontrolle in der Praxis sind. Im letzten Abschnitt 6.6 werden diejenigen Fälle untersucht, bei denen die maschinelle Dublettenkontrolle **falsche Ergebnisse** liefert.

6.1 Was ist eine Dublette?

Bibliothekare erfassen jede Ausgabe einer Publikation (z.B. die zweite Auflage) gesondert. Bei der Aufnahme wird streng darauf geachtet, daß jede Ausgabe nur einmal aufgenommen wird und keine Mehrfachspeicherung (Redundanz) entsteht. Tritt die Mehrfachspeicherung trotzdem auf, spricht man von einer Dublette.

Bei der verteilten Suche sind Dubletten unvermeidlich - das gesuchte Buch (respektive die gesuchte Ausgabe) ist in mehreren Bibliotheken vorhanden und wird bei der Suche auch mehrfach gefunden. Für den Benutzer sind Dubletten störend, da er bei der Suche eine größere Anzahl von Treffern erhält und dann viel Zeit für die Prüfung aufwenden muß, um herauszufinden, welche Datensätze gleich sind (siehe Kapitel 7 Ausgabe von Dubletten, Seite 78).

Die Dublettenkontrolle in *ZACK* orientiert sich am Werk und nicht an unterschiedlichen Ausgaben. Allerdings wird unterschieden nach Sprache und zum Teil nach der physikalischen Ausgabe (z.B. Buch + CDROM). Zum Beispiel ist *Deutschland ein Wintermärchen* von Heinrich Heine ein eindeutig zu identifizierendes Werk; es erschien in verschiedenen Auflagen. Für die meisten Nutzer einer Bibliothek ist der Inhalt einer Publikation wichtiger als die Auflage. Der Verlag, Erscheinungsjahr und Seitenzahl sind eher zweitrangig.

Für weitere Informationen zu Dubletten in Bibliotheksdatenbanken wird auf die Literatur in [Hyl96], [BFM96], [uSU95], [DHHS91], [SH91], [Pay96], [Coy92] [Rid92], [ORO93], [BH91], [KW95], [Ton92], [Wil79], [Hic79], [Goy87], [Mac79], [LSW99] und [Kub99a] verwiesen.

6.2 Manuelle Dublettenkontrolle

Die Dublettenkontrolle mit einem Computer kann nicht besser sein als die Dublettenkontrolle durch einen Menschen. In diesem Abschnitt wird untersucht, inwieweit man anhand der Datensätze entscheiden kann, ob es sich um dieselben Werke handelt und welche Probleme dabei auftauchen.

Als Test werden Bücher zur Computersprache *Perl* gesucht. Es wird die Anfrage *titel=perl* an die Datenbanken der Deutschen Bibliothek (DDB), des Gemeinsamen Bibliotheksverbundes (GBV), des Bibliotheksverbundes Bayern (BVB) und der Technischen Universität Braunschweig (TUBS) gestellt. Insgesamt werden 345 Datensätze gefunden und im Format MAB2 geholt.

Datenbank	Anzahl der Datensätze
BVB	116
DDB	104
GBV	114
TUBS	11

Anzahl der Datensätze nach Bibliothek

Die gefundenen Datensätze wurden nach Autor und Titel sortiert und anschließend auf Papier ausgedruckt. Erwartet werden Bücher über die Computersprache *Perl*. Gefunden werden Werke aus verschiedenen Bereichen:

- Computersprache Perl, CGI-Programmierung
- Gemeinde Perl: Vereine, Landkarten, amtliche Bekanntmachungen, Werbung/Touristikinformationen der Gemeinde Perl, Stadtpläne, geologische Karten, Radwanderkarten von Perl und Umgebung; Kirchgesangsverein Perl
- Personen mit dem Namen Perl: Übersetzer, Sponsoren, Herausgeber
- biologische Literatur über Perl-Zellulose
- chemische Literatur über dentalen Basiskunststoffe

Der größte Teil der Werke (>50%) beinhaltet die Computersprache Perl.

Aufgetretene Probleme

Bei der manuellen Dublettenkontrolle mit dem Beispiel *titel=perl* sind die folgenden Probleme und Fragestellungen aufgetreten:

Karten: Es gibt mehrere Landkarten der Gemeinde Perl aus dem Saarland (Wetter-, Radwander-, Stadt-, Waldkarten). Soll man diese zusammenfassen und als ein Werk betrachten?

ISBN und Auflagen: Mehrere Auflagen des gleichen Werkes haben dieselbe ISBN-Nummer. Es gibt allerdings auch Fälle, wo neue Auflagen neue ISBN-Nummern haben. Was ist hier eine Dublette und was nicht?

Seitenzahl: Die Seitenzahlen gleicher Werke sind leicht verschieden (Abweichung ein bis fünf Seitenzahlen). Sind das Fehler bei der Aufnahme der Werke oder hat sich bei einer späteren Auflage die Zeitenzahl z.B. durch Korrekturen geändert?

Jahr: Das Erscheinungsjahr ist manchmal verschieden angegeben (Abweichung plus/minus ein Jahr). Sind das Tippfehler oder unterschiedliche Auflagen?

Autor: In Autorenamen treten Tippfehler auf. Beispiel: "Haug, Gunter" ⇒ "Hug Gunter"; "Till, Dave" ⇒ "Till, David".

Multimedia-Dokumente: Die DDB gibt das Jahr bei Multimedia-Dokumenten (Buch + CDROM) nicht aus. Statt dessen steht im Datensatz "Medienkombination". Man kann deshalb schwer entscheiden, um welche Auflage es sich handelt.

Fehlende Auflage: Viele Datensätze haben keine Angabe zur Auflage

Unvollständige Datensätze: Bei der Durchsicht der Datensätze stellte sich heraus, daß die Datensätze nicht so homogen wie erwartet waren. Der GBV liefert im MAB2-Vollformat keine ID des Datensatzes (MAB2 Feld 001). In dem dann für diesen Test verwendeten MAB2-Kurzformat fehlte häufig der Autor (Feld 100) des Werkes. Dadurch wurden die GBV-Datensätze anders sortiert und standen in der gedruckten Fassung 100 Seiten entfernt von den ähnlichen Datensätzen von DDB, BVB und TUBS.

Bewertung der Felder

Nicht alle Felder eignen sich für die Dublettenkontrolle.

Gut geeignet sind (nach Normierung) ISBN (Feld 540), Autor (Feld 100), Titel (Feld 331), Jahr (Feld 425)

Bedingt geeignet sind Verlag (Feld 412), Verlagsort (Feld 410), Seitenanzahl (Feld 433) und die Zusätze zum Hauptsachtitel (Feld 335). Diese Felder sind nicht in allen Datensätzen vorhanden. Oft gibt es unterschiedliche Schreibweisen, insbesondere bei Verlag und Verlagsort.

Zusammenfassung

Ursprünglich war geplant, die Datensätze linear (vom ersten bis zum letzten) per Hand durchzulesen und die Dubletten aufzuschreiben - eine Art Memory-Spiel für Bibliothekare ;-). Dieses Verfahren stellte sich als zu aufwendig heraus. Das menschliche Kurzzeitgedächtnis kann nicht so viele Informationen zwischenspeichern. Bei mehr als 50 Datensätzen ist die manuelle Dublettenkontrolle nur noch bedingt möglich. Die Augen ermüden sehr schnell, und man verliert leicht den Überblick.

Die Wahl der Testdaten erwies sich als sehr gut. Unter den Datensätzen befinden sich sowohl Haupt- als auch Untersätze ("h", "u", siehe auch im Anhang Abbildung A.1, Seite 103). Es gab neben den erwarteten Büchern über die Computersprache Perl kirchliche Literatur, übersetzt von Carl Johann Perl; diverse andere Bücher über Menschen mit dem Nachnamen Perl; Bücher, amtliche Bekanntmachungen und Werbung/Touristikinformationen der Gemeinde Perl; Stadtpläne, geologische Karten, Radwanderkarten von Perl und Umgebung; biologische Literatur über Perl-Zellulose.

Die Erfahrungen mit den unterschiedlichen Schreibweisen wurde für die Normierung in Kapitel 5 verwendet (siehe auch Kapitel 9 Probleme).

6.3 Maschinelle Dublettenkontrolle in ZACK

Um Dubletten für den Benutzer zusammenfassen [1] zu können, muß der Computer die Datensätze miteinander vergleichen und entscheiden, ob sie gleiche oder ähnliche Werke beschreiben.

Der Vergleich findet über ausgewählte Attribute (Autor, Titel etc.) statt. Diese werden zueinander ins Verhältnis gesetzt. Im einfachsten Fall sind alle Attribute gleich. Komplizierter wird es, wenn es ein Attribut in nur einem Datensatz gibt, oder wenn sich die Attribute nur minimal unterscheiden.

Bevor die Attribute eines Datensatzes verglichen werden, müssen sie normiert werden. In Kapitel 5 Normierung wird beschrieben, wie Zeichensatz, Fehleingaben (z.B. doppelte Leerzeichen) und unterschiedliche Erfassungspraktiken erkannt und bearbeitet werden.

6.3.1 Vergleich von Attributen

Beim Vergleich eines Attributes zweier Datensätze gibt es vier Fälle:

1. Das Attribut X existiert in beiden Datensätzen.
2. Das Attribut X existiert im ersten Datensatz, aber nicht im zweiten.
3. Das Attribut X existiert nicht im ersten Datensatz, aber im zweiten.
4. Das Attribut X existiert in keinem der beiden Datensätze

Für die Dublettenkontrolle ist vor allem der erste Fall *das Attribut existiert in beiden Datensätzen* interessant. Die Fälle 2 und 3 - das Attribut existiert in nur einem der beiden Datensätze sind von geringer Bedeutung (siehe Positive Gewichtung II, Tabelle 6.1, Seite 64). Der letzte Fall ist für die Dublettenkontrolle uninteressant - was nicht vorhanden ist, kann auch nicht verglichen werden.

ZACK arbeitet mit *Gleichheit* und *Ähnlichkeit*. Gleichheit und Ähnlichkeit sind in *ZACK* wie folgt definiert:

Gleichheit: Beide Attribute sind gleich.

Ähnlichkeit: Kleinere Unterschiede zwischen den Attributen werden nicht beachtet. Dazu gehören bei Zahlen kleine Abweichungen nach oben oder unten (+5 Seiten, -5 Seiten). Bei Zeichenfolgen werden ein oder zwei Tippfehler ignoriert (siehe 6.3.4 Trigramme).

In *ZACK* wird die Gleichheit zweier Datensätze nach den Regeln eines Expertensystems festgestellt. Es werden mehrere Attribute auf Gleichheit bzw. Ähnlichkeit überprüft und eine entsprechende Gewichtung vergeben. Alle Gewichtungen der einzelnen Regeln werden am Schluß zu einer Gesamtbewertung verrechnet. Anhand der Gesamtbewertung wird entschieden, ob es sich um eine Dublette handelt (nach [RM94], siehe auch [Pup88] und [BFM96]).

Bei den Gewichtungen handelt es sich um "Symptom-Diagnose-Wahrscheinlichkeiten". Diese geben an, wie wahrscheinlich bei einem bestimmten Symptom (z.B. Attribut ISBN gleich) die Diagnose "*die Sätze sind dublett*" sind. Bei den Wahrscheinlichkeiten handelt es sich um empirische Wahrscheinlichkeiten, die von Experten geschätzt werden. Es sind keine statistischen Wahrscheinlichkeiten.

[1] Auf die Ausgabe von Dubletten wird im Kapitel 7 detailliert eingegangen.

Attribut	positive Gewichtung	positive Gewichtung II	negative Gewichtung
Titel	70	0	30
Autor	40	10	30
Jahr	20	0	20
Verlag	20	5	10
Verlagsort	20	5	10
Seitennummer	30	5	20
Auflage	10	5	5
ISBN	80	0	10

Tabelle 6.1: Dublettenkontrolle in *ZACK*: Gewichtungen der Attribute beim Vergleich

Legende Gewichtungen der Attribute

Positive Gewichtung (Pro1): Wird vergeben, wenn die Attribute in beiden Datensätzen übereinstimmen.

Positive Gewichtung II (Pro2): Wird vergeben, wenn das betreffende Attribut in einem Datensatz existiert und im anderen Datensatz nicht. Alternativ könnte man diese Gewichtung vergeben, wenn zwei Attribute ähnlich, aber nicht gleich sind.

Negative Gewichtung (Con): Wird vergeben, wenn die Attribute nicht gleich oder ähnlich sind.

Zur besseren Lesbarkeit werden die Gewichtungen hier in Prozent angegeben. Eine positive Gewichtung von 70 steht also für eine Wahrscheinlichkeit von 70% bzw. 0,7, daß die Datensätze dublett sind.

Ziel in *ZACK* ist die Erkennung von gleichen Werken, nicht nur gleichen Ausgaben. Deshalb werden die negative Gewichtungen generell niedriger geschätzt als positive Gewichtungen.

Das Werk wird vor allem durch den Autor und den Titel bestimmt. Deshalb wird eine hohe positive und negative Gewichtung für Autor und Titel vergeben. Die ISBN-Nummer erhielt ebenfalls eine hohe positive Gewichtung. Wenn zwei Bücher dieselbe ISBN-Nummer haben, dann handelt es sich mit sehr hoher Wahrscheinlichkeit um das gleiche Werk. Verlag, Verlagsort, Auflage und ISBN-Nummer bezeichnen die physikalische Ausgabe. Ein Buch kann in verschiedenen Verlagen erscheinen, z.B als Hardcover und als Paperback. Die negativen Gewichtungen für Verlag, Verlagsort und ISBN-Nummer werden deshalb niedrig geschätzt, um trotzdem inhaltlich gleiche Publikationen als solche zu erkennen.

6.3.2 Berechnung der Gesamtgewichtung

Bei *ZACK* besteht die Gesamtbewertung aus zwei Werten: der positiven Gesamtbewertung (Argumente für eine Dublette) und der negativen Gesamtbewertung (Argumente, die gegen eine Dublette sprechen). Die positive Gesamtbewertung faßt alle positiven Gewichtungen zusammen und die negative Gesamtbewertung alle negativen Gewichtungen.

Für die Berechnung der Gesamtbewertung gibt es zwei Alternativen:

1. Alle Bewertungen werden addiert.

2. Alle Bewertungen werden zu einer Gesamtevidenz berechnet.

In *ZACK* wird die Gesamtevidenz verwendet. Die Gesamtevidenz ist immer ein Wert zwischen 0 und 1. Bei kleinen Evidenzen verhält sich die Gesamtevidenz wie die Addition, je größer die Gesamtevidenz wird, desto weniger erhöhen weitere Evidenzen ihren Wert (siehe auch [Pup88], Seite 52f).

Die positive bzw. negative Gesamtevidenz wird nach der Formel berechnet:

$$G_1 = E_1$$

$$G_n = G_{n-1} + (1 - G_{n-1}) * E_n$$

Einzelevidenzen: E_1 bis E_n (mit $0 \leq E_i \leq 1$)
Gesamtevidenz: G (mit $0 \leq G \leq 1$)

Es gilt:

$$G = 1 - (1 - E_1) * (1 - E_2) * \ldots (1 - E_{n-1}) * (1 - E_n)$$

Beweis (vollständige Induktion):
$n = 1$ | ok
$n \Rightarrow n + 1$
$1 - (1 - E_1) * ... * (1 - E_{n+1})$
$= 1 - (1 - E_1) * ... * (1 - E_n) * (1 - E_{n+1})$
$= 1 - (1 - E_1) * ... * (1 - E_n) + (1 - E_1) * ... * (1 - E_n) * E_{n+1}$ | ausmultiplizieren
$= \underbrace{1 - (1 - E_1) * ... * (1 - E_n)}_{G_n} + [1 \underbrace{-1 + (1 - E_1) * ... * (1 - E_n)}_{-G_n}] * E_{n+1}$ | Null addieren

$= G_n + [1 - G_n] * E_{n+1}$
$= G_{n+1}$

Beispiel für den Vergleich zweier Datensätze mit Gewichtung:

	Beispiel 1	Beispiel 2
Titel	¬Die¬ Akazie	¬Die¬ Akazie
Autor	Simon, Claude	Simon, Claude
Jahr	1998	1993
Verlag	Suhrkamp	Suhrkamp
Verlagsort	Frankfurt am Main	Frankfurt am Main
Seitennummer	354 S.	354 S.
Auflage	1. Aufl.	1. Aufl.
ISBN	ISBN 3-518-22302-X Pp. : DM 29.80	ISBN 3-518-38732-4

Tabelle 6.2: Beispiel Dublettenkontrolle in *ZACK*: Attribute vor Normierung

Attribut	Beispiel 1	Beispiel 2	Pro1	Pro2	Con	positive Gesamt-evidenz	negative Gesamt-evidenz
						0	0
Titel	die akazie	die akazie	70	-	-	70	0
Autor	simonc	simonc	40	-	-	82	0
Verlag	suhrk	suhrk	20	-	-	85,6	0
Jahr	1998	1993	-	-	20	85,6	20
Verlagsort	frank	frank	20	-	-	88,48	20
ISBN	351822302X	3518387324	0	-	10	88,48	28
Seitennummer	354	354	30	-	-	91,936	28
Auflage	1	1	10	-	-	92,7424	28

Tabelle 6.3: Beispiel Dublettenkontrolle in *ZACK*: mit Normierung und Berechnung der positiven und negative Gesamtevidenzen

positive Gesamtevidenz:
$G = 1 - (1 - 0,7) * (1 - 0.4) * (1 - 0,2) * (1 - 0,2) * (1 - 0.3) * (1 - 0,1)$
$G = 1 - 0,3 * 0,6 * 0,8 * 0,7 * 0,9 * 0,8 = 0,927474$

negative Gesamtevidenz:
$G = 1 - (1 - 0,2) * (1 - 0,1)$
$G = 1 - 0,8 * 0,9 = 0,28$

Jede zusätzliche Evidenz füllt den restlichen Bereich zwischen der bisherigen Gesamtevidenz und dem maximalen Wert 1 anteilmäßig auf. Wenn die bisherige Gesamtevidenz beispielsweise bei 0,7 liegt, so füllt die hinzuzurechnende Evidenz 0,4 den zwischen 0,7 und 1,0 liegenden Bereich 0,3 zu 0,4 (40%) = 0,12 auf. Die neue Gesamtevidenz beträgt somit $0,7 + 0,12 = 0,82$.

Datensätze werden als Dubletten erkannt, wenn die positive Gesamtevidenz über dem positiven Schwellwert von 0,75 und gleichzeitig die negative Gesamtevidenz unter dem negativen Schwellwert von 0,4 liegt. Andere Fälle (positiver Schwellwert von 0,75 nicht erreicht oder negativer Schwellwert von 0,4 überschritten) gelten als nicht dublett. Die Schwellwerte für die positive und negative Gesamtevidenz werden empirisch bestimmt (siehe auch Abschnitt 6.4 Interaktive Dublettenkontrolle).

Die grundlegende Schwäche dieses Verfahrens liegt darin, daß die Evidenzen so behandelt werden, als ob sie statistisch voneinander unabhängig wären. In Wirklichkeit jedoch impliziert ein gleicher Titel mit hoher Wahrscheinlichkeit auch eine gleiche ISBN-Nummer.

Die Reihenfolge der verglichenen Attribute ändert nichts am Gesamtergebnis der Gesamtevidenz. Es ist also unerheblich, ob man in der Reihenfolge Titel, Autor und ISBN vergleicht oder in der umgekehrten Reihenfolge ISBN, Autor und Titel (siehe auch [RM94] und [Pup88]).

6.3.3 Ähnliche Zahlen

Die Bestimmung der Ähnlichkeit von Zahlen ist einfach. Die Zahlen werden mit einer gewissen zulässigen Abweichung miteinander verglichen.

Jahr: +/- ein Jahr

Seitenzahl: +/- 5 Seiten

Tippfehler und unterschiedliche Zählweisen bei der Seitennumerierung werden erkannt und ausgeglichen (siehe auch Kapitel Normierung 5, Seite 5).

6.3.4 Ähnliche Zeichenfolgen

Um Tippfehler als solche zu erkennen, werden in *ZACK* Trigramme verwendet. Trigramme sind Zeichenfolgen der Länge 3. Die Zeichenfolgen werden am Anfang und am Ende mit dem Zeichen "_" aufgefüllt, um bessere Ergebnisse bei kurzen Wörtern zu erhalten. Bei der Ähnlichkeitssuche über Trigramme werden zunächst die Wörter auf die Menge der enthaltenen Trigramme abgebildet, also z.B. für das Wort "*martha*" und "*marta*":

"martha" $\Rightarrow$ {"_ma", "mar", "art", "rth", "tha", "ha_"}
"marta" $\Rightarrow$ {"_ma", "mar", "art", "rta", "ta_"}

Wort	_ma	mar	art	rth	tha	ha_	rta	ta_
martha	1	1	1	1	1	1	0	0
marta	1	1	1	0	0	0	1	1

Abbildung 6.1: Trigramme für *martha* und *marta*

Danach vergleicht man die zu beiden Wörtern gehörenden Trigramme jeweils miteinander und bildet die Differenz [2]. In diesem Beispiel gibt es insgesamt 8 verschiedene Trigramme. 3 Trigramme sind in beiden Wörtern vorhanden, 5 Trigramme gibt es in nur jeweils einem Wort. Das Löschen des Buchstabens "h" hat bewirkt, daß im neuen Wort *marta* 3 Trigramme fehlen und 2 hinzugekommen sind.

In *ZACK* werden bei kurzen Zeichenfolgen (Länge $<$ 8 Zeichen) ein Tippfehler und bei längeren Zeichenfolgen (Länge >= 8 Zeichen) zwei Tippfehler toleriert.

Trigramme sind eine effiziente und wirkungsvolle Methode zur Ähnlichkeitssuche auf Zeichenfolgen. Für weitere Informationen zu Trigrammen wird auf die Literatur in [ZPZ81], [Goy84] und [Hyl96]) verwiesen.

6.4 Interaktive Dublettenkontrolle

Um die Dublettenkontrolle besser konfigurieren und bewerten zu können, wurde das CGI-Script `match` geschrieben (siehe auch Kapitel Software, Seite 124).

Die folgenden Abbildungen 6.2 (Seite 68), 6.3 (Seite 69) und 6.4 (Seite 70) zeigen das Script `match` (siehe auch Kapitel C Kurzbeschreibung der Software). Das Script `match` führt eine Dublettenkontrolle mit positiver und negativer Gewichtung durch. Der Benutzer kann die jeweiligen Gewichtungen einstellen, den positiven oder negativen Schwellwert, die Art der Normierungen, ob kleine Fehler ignoriert werden (Seitenzahl, Tippfehler im Titel, Jahreszahl +/- 1 Jahr etc.). Das CGI-Script `match` bietet 3 Beispiele zum Testen der Dublettenkontrolle in *ZACK* an:

1. 60 Datensätze aus der Deutschen Bibliothek und dem Bibliotheksverbund Bayern, die bei der Suche nach dem *titel=Akazie* gefunden wurden.

2. 26 Datensätze aus der Deutschen Bibliothek, der Technischen Universität Braunschweig, dem Gemeinsamen Bibliotheksverbund und dem Bibliotheksverbund Bayern, die bei der Suche nach dem *autor=Dalitz, Wolfgang* gefunden wurden.

[2]Bei den Trigrammen handelt es sich um Vektoren in einem 36^3-dimensionalen Raum (26 Buchstaben + 10 Ziffern). Es wird der euklidische Abstand der beiden Vektoren bestimmt. Der Abstand der Vektoren $\vec{A}$ und $\vec{B}$ ist: $\|\vec{D}\| = \sqrt{\sum_{i=aaa}^{999}(a_i - b_i)^2}$

3. Zwei Datensätze aus dem Beispiel Akazie. Der Benutzer kann jedes Feld oder auch den ganzen Datensatz ändern und sich die Auswirkungen auf die Dublettenkontrolle ansehen. Zum Beispiel könnte man gezielt Tippfehler einfügen oder Felder löschen. Wenn einem diese beiden Beispieldatensätze nicht gefallen, kann man die Datensätze auch löschen und durch andere Datensätze ersetzen.

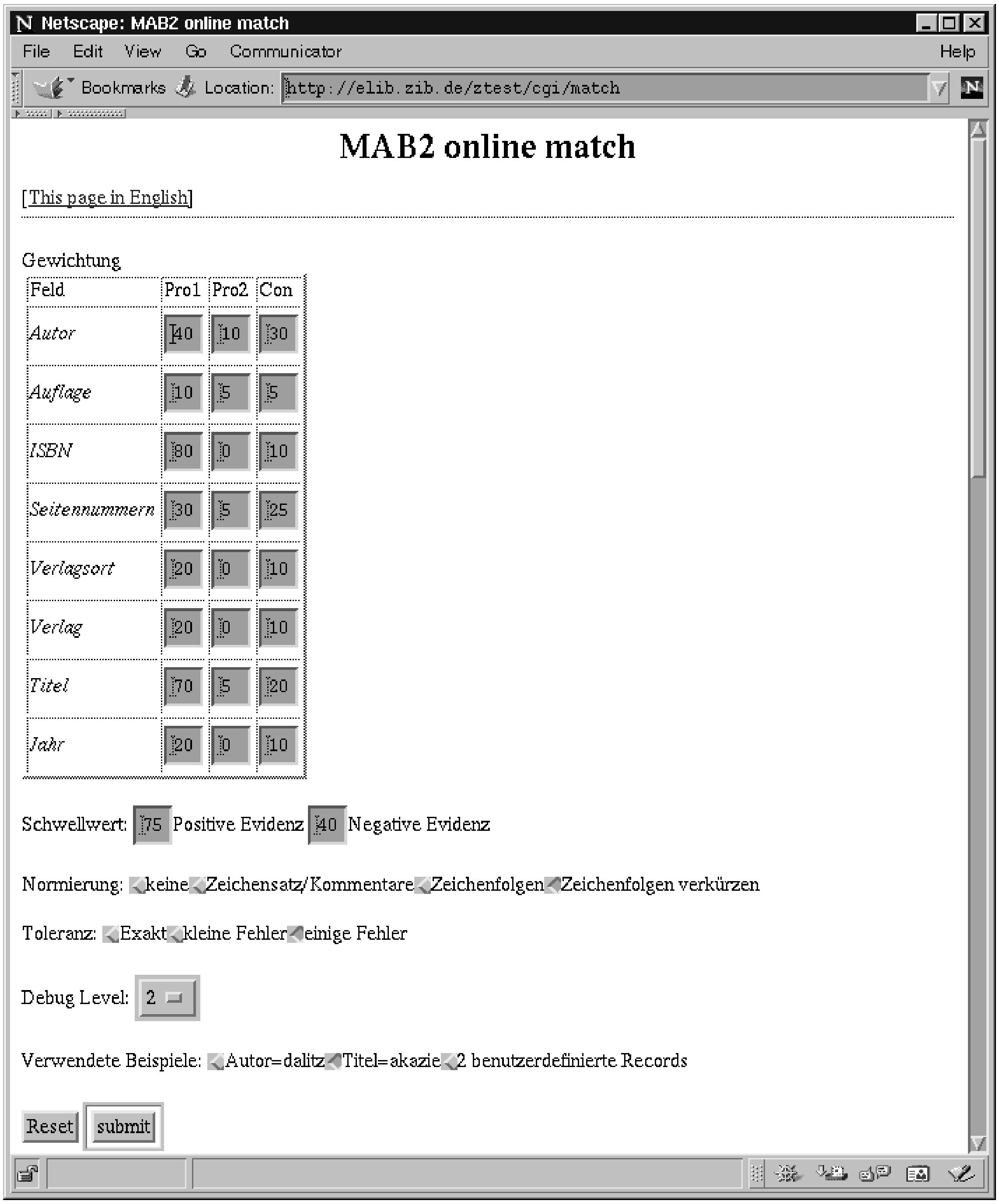


Abbildung 6.2: CGI-Script Interaktive Dublettenkontrolle, erster Teil

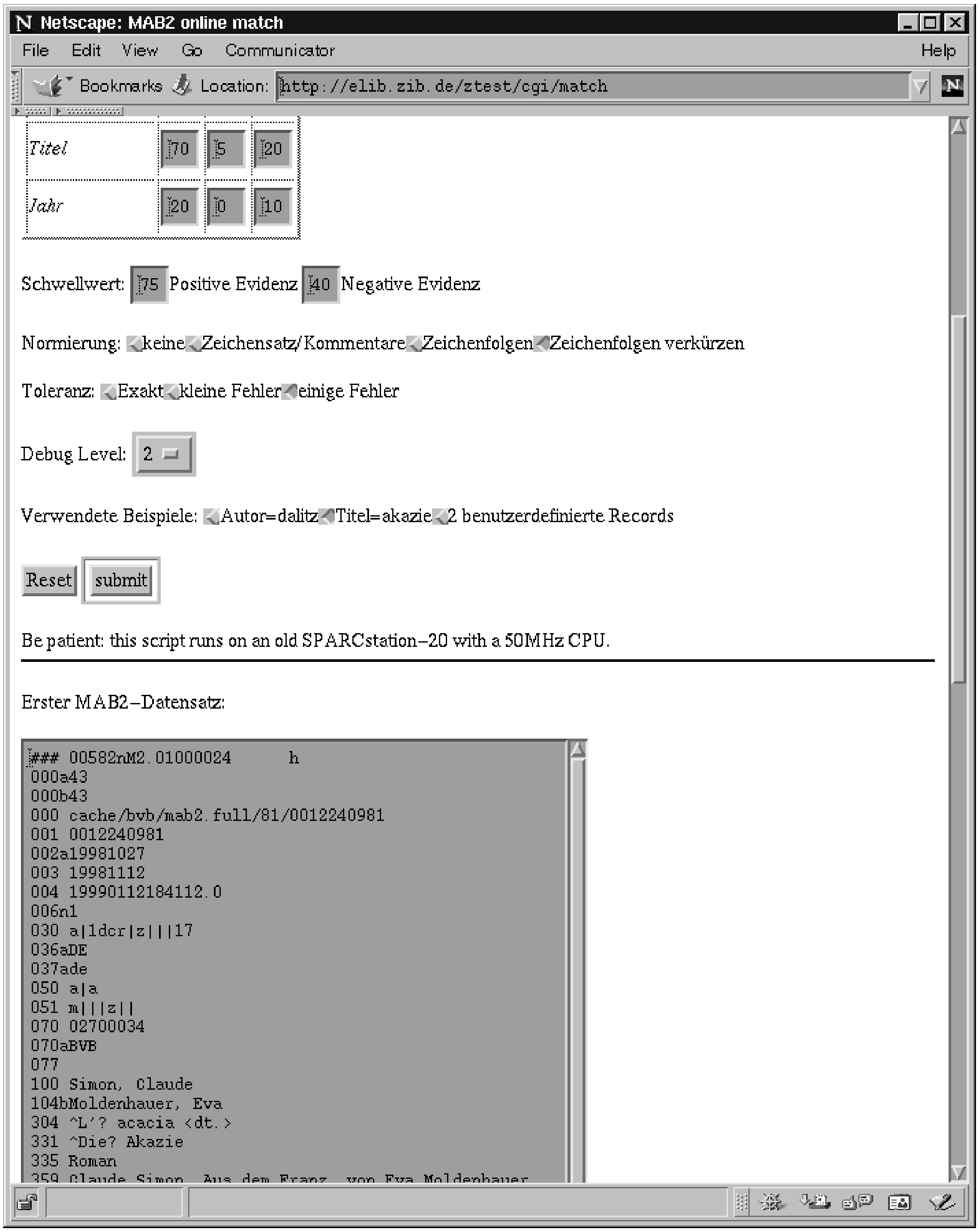


Abbildung 6.3: CGI-Script Interaktive Dublettenkontrolle, zweiter Teil

Abbildung 6.4: CGI-Script Interaktive Dublettenkontrolle, dritter Teil

Legende CGI-Script match

Toleranz: Gibt die Kriterien für den Vergleich der Attribute an.

- **exakt:** Die Attribute müssen genau übereinstimmen.
- **kleine Fehler:** Das Jahr darf um den Wert +/- 1 abweichen; die Seitennummer darf um dem Wert +/- 5 abweichen; Titel, Autor, Verlag, Verlagsort dürfen maximal 2 Tippfehler beinhalten.

einige Fehler: Bei unterschiedlicher Länge von Titel, Autor, Verlag und Verlagsort wird nur bis zum Ende der kürzeren Zeichenfolge verglichen.

Beispiel:

a) "*Verteilung mathematischer Software mittels elektronischer Netze*"
b) "*Verteilung mathematischer Software*"

Es wird beim Vergleich von a) mit b) nur bis zum Wort Software verglichen.

Gewichtung: *Pro1* ist die positive Evidenz, die bei Gleichheit oder Ähnlichkeit eines Attributes vergeben wird. *Pro2* ist eine zweite positive Evidenz, die vergeben wird, wenn das Attribut in einem Datensatz existiert und im anderen nicht. *Con* ist die negative Evidenz, die bei Ungleichheit der Attribute vergeben wird.

Für jedes Attribut gibt es einen Pro1, Pro2 und Con Wert im Wertebereich von 0 bis 100. Zur besseren Lesbarkeit werden die Gewichtungen hier in Prozent angegeben. Eine positive Gewichtung von 70 steht also für eine Wahrscheinlichkeit von 70% bzw. 0,7, daß die Datensätze dublett sind.

1. positive Gewichtung (Pro1): beide Attribute stimmen überein

2. positive Gewichtung (Pro2): das Attribut ist in einem Datensatz belegt, aber nicht im zweiten Datensatz.

3. negative Gewichtung (Con): beide stimmen nicht übereinstimmen.

Ein Wert von 0 führt zu keiner Änderung der Gesamtevidenz - das Attribut wird bei der Dublettenkontrolle ignoriert.

Ein Wert von 100 sorgt dafür, daß die Gesamtevidenz den Maximalwert 100 erreicht. D.h. bei Gleichheit eines *einzelnen* Attributes - z.B. die ISBN- Nummer - wird der maximale Wert erreicht. Zusätzliche weitere gleiche Attribute (Autor, Titel etc.) erhöhen dann die Gesamtevidenz nicht mehr.

Schwellwert für Gesamtevidenz: Anhand dieser Schwellwerte wird entschieden, ob es sich um Dubletten handelt oder nicht.

- Positive Gesamtevidenz: 75, für Attribute, die übereinstimmen
- Negative Gesamtevidenz: 40, für Attribute, die nicht übereinstimmen.

Die Datensätze werden als Dubletten erkannt, wenn die positive Gesamtevidenz über dem Schwellwert von 75 und gleichzeitig die negative Gesamtevidenz unter dem Schwellwert von 40 liegt. Andere Fälle (positiver Schwellwert von 75 nicht erreicht oder negativer Schwellwert von 40 überschritten) gelten als nicht dublett.

Debug: 0 Nur die Kurztrefferliste wird ausgegeben.

1 Ausgabe der Gesamtevidenz Ergebnisse für jeden Vergleich (G-SORT)

2 Ausgabe der MAB2-Datensätze im Kategorienformat, gefolgt von einer Liste der Datensätze, die ein Attribut gemeinsam haben (Recordlist).

Beispiel:

```
### 00331nM2.01000024      h
001 0011134474
002a19970107
003 19970107
004 19990112184111.0
006n1
```

```
030 u|1dcr|z|||37
037ade
050 a|a
051 m||||||
070 00700037
070aBVB
077 $a37$c1
331 Kurze Belehrung über den Anbau des unächten Akazienbaums
335 für fränkische Forstbediente und Landwirthe
410 Nürnberg
412 Felßecker
425a1797
433 20 S.

TITLE -> kurze belehrung ueber den anbau des unaechten akazienbaums :$VAR1 = {
          6 => 1,
          7 => 1
        };

AUTHOR ->  :$VAR1 = {};

PUBLISHER -> felss :$VAR1 = {
          6 => 1
        };

YEAR -> 1797 :$VAR1 = {
          6 => 1,
          7 => 1
        };

PLACE -> nuern :$VAR1 = {
          6 => 1,
          7 => 1
        };

ISBN ->  :$VAR1 = {};

PAGENR -> 20 :$VAR1 = {
          6 => 1
        };

Recordlist: 6 7
G-SORT: 6 7 Pro1: 110, pro2: 5, con: 10, E-Pro: 0.818 E-Con: 0.100 matched
```

Die Datensätze 6 und 7 haben denselben (normierten) Titel "*kurze belehrung ueber den anbau des unaechten akazienbaums*"; Jahr, Verlagsort sind ebenfalls gleich. Der Datensatz 6 hat kein Attribut *Autor*. Den (normierten) Verlag "*felss*" gibt es in nur diesem Datensatz. Die Datensätze 6 und 7 werden als minimal ähnlich erkannt. Die Gesamtgewichtung ergibt: positive Gewichtung (Pro1) in der Summe 110 (Titel + Jahr + Verlagsort), positive Gewichtung II (Pro2) in der Summe 5 (Seitennummer), negative Gewichtung in der Summe 10 (Verlag). Die positive Gesamtevidenz beträgt 0,818, die negative Gesamtevidenz 0,1. Die Datensätze werden als Dublette erkannt - die positive Gesamtevidenz liegt über dem Schwellwert 0,75 und die negative Gesamtevidenz unter dem Schwellwert 0,4.

6.5 Effizienz der Dublettenkontrolle

Für die Dublettenkontrolle müssen die gefundenen Datensätze miteinander verglichen werden. Im einfachsten (und ungünstigsten Fall) wird jeder Datensatz mit jedem verglichen. Dafür sind

$$(n-1)+(n-2)+(n-3)+..+1=\frac{n^2-n}{2}$$

Vergleiche nötig. Die Anzahl der Vergleiche wächst quadratisch mit der Anzahl der Datensätze (O^2). Für den Vergleich von 10 Datensätze sind 45 Vergleiche notwendig, bei 30 Datensätze sind es bereits 435 Vergleiche.

	1	2	3	4	5	6	7	8
1								
2	X							
3	X	X						
4	X	X	X					
5	X	X	X	X				
6	X	X	X	X	X			
7	X	X	X	X	X	X		
8	X	X	X	X	X	X	X	

Abbildung 6.5: Aufwand Algorithmus *vergleiche jeden Datensatz mit jedem*

In der Abbildung 6.5 werden 8 Datensätze jeweils miteinander verglichen. Es werden insgesamt 28 Vergleiche durchgeführt. Der Buchstabe "x" steht in diesem Beispiel für einen Vergleich, leere Felder für stehen für *kein Vergleich*.

Mit einem temporären Index kann man die Anzahl der notwendigen Vergleiche deutlich reduzieren. Man trifft eine Vorauswahl von Datensätzen, die eine gewisse minimale Ähnlichkeit miteinander haben (Cluster). Nur diese werden dann miteinander verglichen und nicht mehr jeder Datensatz mit jedem.

In *ZACK* wird die Vorauswahl von ähnlichen Datensätzen nach einer einfachen Bedingung getroffen: Es müssen zwei der bei der Dublettenkontrolle verwendeten Attribute (z.B. Titel, Autor, Verlag, Verlagsort, Jahr, Seitenzahl) *genau* übereinstimmen.

Die Vorauswahl von Datensätzen (Clusterbildung) wird an den folgenden fiktiven Beispielen in den Abbildungen 6.6 und 6.7 verdeutlicht. In beiden Beispielen werden 8 Datensätze auf Dubletten überprüft.

	1	2	3	4	5	6	7	8
1								
2	X							
3	X	X						
4			X					
5			X	X				
6			X	X	X			
7						X		
8						X	X	

Abbildung 6.6: Aufwand optimierter Algorithmus mit Index, Beispiel 1

In der Abbildung 6.6 wird eine Vorauswahl von minimal ähnlichen Datensätzen getroffen. Es werden die Cluster mit den Datensätzen {1, 2, 3}, {3, 4, 5, 6} und {6, 7, 8} gebildet. Insgesamt sind jetzt nur noch 12 Vergleiche notwendig.

	1	2	3	4	5	6	7	8
1								
2	X							
3	X	X						
4								
5		X						
6		X						
7								
8						X		

Abbildung 6.7: Aufwand optimierter Algorithmus mit Index, Beispiel 2

Zweites Beispiel: in der Abbildung 6.7 wird eine Vorauswahl von minimal ähnlichen Datensätzen getroffen. Es werden die Cluster mit den Datensätzen {1, 2, 3}, {2, 5, 6} und {6, 8} gebildet. Insgesamt sind jetzt nur noch 6 Vergleiche notwendig.

Die Effizienz der Clusterbildung hängt von den verwendeten Algorithmen und der Zusammensetzung der Datensätze ab. Der Algorithmus für die Clusterbildung und den temporären Index in *ZACK* wird in der Tabelle 6.4 untersucht (siehe auch Abbildung 6.8). Dazu werden 5 Anfragen nach einem Titel und 2 nach einem Autor an die Datenbanken der Deutschen Bibliothek (DDB), des Gemeinsamen Bibliotheksverbundes (GBV), des Bibliotheksverbundes Bayern (BVB) und der Technischen Universität Braunschweig (TUBS) gestellt (siehe auch Kapitel Normierung, Seite 49). Ziel ist es festzustellen, wie effizient der Algorithmus in der Praxis ist.

Anfrage	Anzahl Treffer	Vergleiche $\frac{(n^2-n)}{2}$	Vergleiche *ZACK* Index	in Prozent	Vergleiche pro Treffer	Zeit in Sekunden
titel=akazie	60	1.770	172	9,7 %	2,9	0,5
titel=birnbaum	342	58.311	6.704	11,5 %	19,6	5,6
titel=karstadt	61	1.830	216	11,8 %	3,5	0,4
titel=pankow	212	22.366	820	3,7 %	3,9	1,4
titel=perl	345	59.340	5.154	8,7 %	14,9	5,3
autor=dalitz,wolfgang	28	378	254	67,2 %	9,1	0,6
autor=rusch,beate	9	36	24	66,7 %	2,7	0,2

Tabelle 6.4: Dublettenkontrolle in *ZACK*: Aufwand und Rechenzeit mit Index

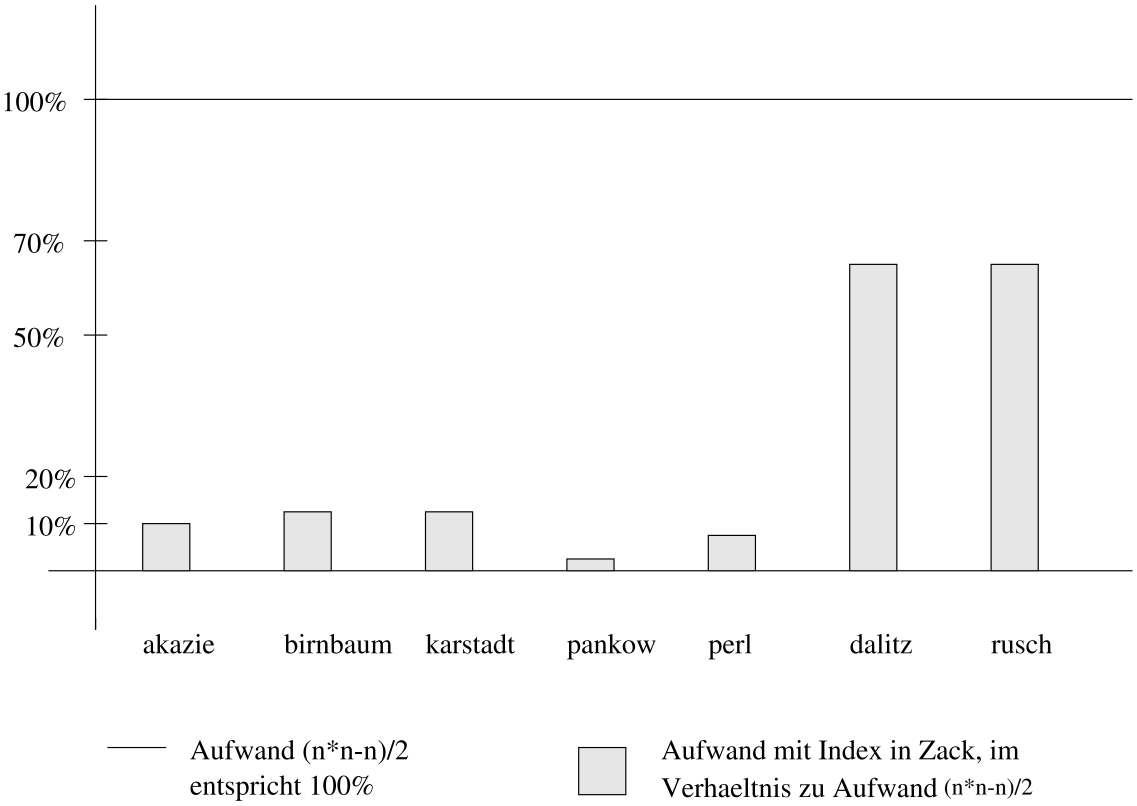


Abbildung 6.8: Aufwand mit Index im Verhältnis zu *vergleiche jeden Datensatz mit jedem*

Legende Aufwand und Rechenzeit mit Index

Anfrage: Die an die Datenbanken gestellte Anfrage.

Anzahl Treffer: Die Summe der Treffer von allen Datenbanken insgesamt.

Vergleiche: Jeder Datensatz wird mit jedem verglichen. Der Aufwand wächst quadratisch mit der Anzahl der Datensätze: $(n^2 - n)/2$. Dies ist der ungünstigste Fall (worst case).

Vergleiche ZACK Index: Diese Spalte gibt die Anzahl der notwendigen Vergleiche mit einem Index im System *ZACK* an.

In Prozent: Gibt an, wie effektiv die Nutzung eines Indexes im Unterschied zum Verfahren *jeden Datensatz mit jedem zu vergleichen* ist.

Vergleiche pro Treffer: Gibt an, wieviele Vergleiche durchschnittlich pro Datensatz mit einem Index notwendig sind.

Zeit in Sekunden: Gibt die Rechenzeit [3] in *ZACK* für die Dublettenkontrolle mit Index an.

Es hat sich gezeigt, daß man mit einem Index die Anzahl der notwendigen Vergleiche deutlich reduzieren kann. Bei der Suche im Attribut *Titel* reduziert sich der Aufwand für die Vergleiche um rund 90%, bei der Suche mit dem Attribut *Autor* immerhin noch um 1/3.

Bei der Autorensuche haben die Treffer fast alle denselben Autor. Deshalb wird die Vorauswahl von ähnlichen Datensätzen schwieriger und komplexer. Im ungünstigsten Fall sind die

[3] Die Rechenzeit wurde auf dem Rechner se2 gemessen, eine UltraSPARC-II mit 336 MHz, siehe Abkürzungsverzeichnis.

Cluster so groß wie die Anzahl der Datensätze, und der Aufwand mit einem Index ist genauso groß wie bei einem Vergleich "*jeder Datensatz mit jedem*" (O^2). Bei großen Datenmengen (>100) gibt es mehr potentielle Dubletten und die Größe der Cluster wächst. Die Anzahl der notwendigen Vergleiche wird um so geringer, je besser die Cluster der minimal ähnlichen Datensätze gebildet werden. Dabei ist entscheidend, daß die Cluster möglichst klein bleiben (Anzahl < 5), da innerhalb eines Clusters alle Datensätze miteinander verglichen werden.

Für den Aufbau des temporären Indexes wird Rechenzeit benötigt. Der Aufwand dafür wächst linear mit der Anzahl der Datensätze. Mit einem Index kann man nur gleiche Attribute vergleichen. Sobald Tippfehler auftauchen, ist ein Vergleich nicht mehr möglich.

6.6 Probleme in der Praxis

Die Dublettenkontrolle liefert in seltenen Fällen falsche Ergebnisse. Es werden Datensätze als dublett erkannt, die nicht das gleiche Werk bezeichnen.

Beispiel: Ein Autor veröffentlicht innerhalb eines Jahres mehrere Publikationen in demselben Verlag. Die Titel sind verschieden, aber Autor, Verlag, Verlagsort und Jahr sind gleich.

Zur besseren Übersicht werden die Datensätze von der DDB hier leicht gekürzt im MAB2-Kategorienformat ausgegeben. Nicht dargestellt werden interne oder automatisch erzeugte Felder (z.B. Feldnummern 002-099) sowie Schlagwörter.

```
### 00700nM2.01200024      h
001 952637898
100 Alevras, Dimitris
104aGrötschel, Martin
108aWessäly, Roland
331 Cost efficient network synthesis from leased lines
359 Dimitris Alevras ; Martin Grötschel ; Roland Wessäly. \
      Konrad-Zuse-Zentrum für Informationstechnik Berlin, ZIB
410 Berlin
412 ZIB
425a1997
433 19 S.
435 30 cm
451 Preprint / Konrad-Zuse-Zentrum für Informationstechnik Berlin ; SC 97,22
454bKonrad-Zuse-Zentrum für Informationstechnik <Berlin>: Preprint
455 SC 97,22

### 00706nM2.01200024      h
001 951590138
100 Alevras, Dimitris
104aGrötschel, Martin
108aWessäly, Roland
331 Capacity and survivability models for telecommunication networks
359 Dimitris Alevras ; Martin Grötschel ; Roland Wessäly. \
      Konrad-Zuse-Zentrum für Informationstechnik Berlin, ZIB
410 Berlin
412 ZIB
425a1997
433 14 S.
435 30 cm
451 Preprint / Konrad-Zuse-Zentrum für Informationstechnik Berlin ; SC 97,24
454bKonrad-Zuse-Zentrum für Informationstechnik <Berlin>: Preprint
455 SC 97,24
```

In diesem Beispiel ist der Titel verschieden, aber die Autoren (Dimitris Alevras, Roland Wessäly und Martin Grötschel), Verlagsort (Berlin), Verlag (ZIB), Jahr (1997) und die Seitenzahl (19 bzw. 14 Seiten, Toleranz +/- 5 Seiten) sind gleich.

Bei Neuauflagen von Klassikern (z.B. Goethe, Faust) tritt ein weiteres Problem auf. Titel und Autor sind identisch, nur die Herausgeber, Verlag, Jahr, Seitenzahl und ISBN sind verschieden. Titel und Autor haben in *ZACK* eine hohe Gewichtung, da sie die wichtigsten Attribute zur Bestimmung eines Werkes sind. Soll man diese Neuauflagen jetzt als Dublette zusammenfassen? Falls nicht, muß man gegebenenfalls anhand des Attributes Herausgeber und ISBN-Nummer entscheiden, ob die Auflagen verschieden sind.

Die ISBN-Nummer sollte bei jeder Ausgabe vom Verlag neu vergeben ([Ott94]) werden. Bücher mit ungleichen ISBN-Nummern können deshalb durchaus das gleiche Werk beschreiben. Deshalb wurde in *ZACK* eine geringe negative Gewichtung vergeben, falls die ISBN-Nummern nicht übereinstimmen.

Die Dublettenkontrolle in *ZACK* verwendet z.Z. nicht die *zweite Person* (Feld 104) und auch nicht die *Zusätze zum Hauptsachtitel* (Feld 335).

Es hat sich als Fehler herausgestellt, daß *ZACK* die *Zusätze zum Hauptsachtitel* ignoriert. Dieses Feld ist für die Dublettenkontrolle wichtig. Die zweite Person sollte ebenfalls in die Dublettenerkennung einbezogen werden, um Neuauflagen von Klassikern besser zu erkennen.

Kapitel 7

Ausgabe von Dubletten

Bei der Dublettenkontolle werden gleiche bzw. ähnliche Datensätze gefunden. Diese werden in einer geeigneten Art und Weise dem Benutzer präsentiert.

Die Datensätze werden in Kurzdarstellung - einer kurzen, kompakten, leicht verständlichen Form - ausgegeben. In der Kurzdarstellung werden nur wenige Attribute ausgegeben, hier z.B. Autor, Titel, Verlag, Verlagsort und Jahr. Alle anderen Attribute werden ignoriert (z.B. ISBN, Auflage, Seitenzahl, 2. Autor, 3. Autor, Schlagwörter etc.).

Zur Ausgabe der Dubletten gibt es drei Alternativen:

1. Man gibt alle dubletten Datensätze und die zugehörigen Bibliotheken hintereinander aus.

2. Man wählt einen Datensatz aus den Dubletten und gibt diesen aus. Zusätzlich wird angegeben, in welchen Bibliotheken der Titel vorhanden ist.

3. Man fügt die als dublett erkannten Datensätze zu einem neuen Datensatz zusammen und gibt diesen Datensatz aus. Zusätzlich wird angegeben, in welchen Bibliotheken der Titel vorhanden ist.

Grundsätzlich unterscheiden sich dublette Datensätze in der Ausgabe nur minimal voneinander, da sie zuvor anhand einiger wichtiger Attribute [1] als gleich oder sehr ähnlich erkannt worden sind. Wenn die vollständigen Datensätze untereinander sehr ähnlich sind, wird sich auch in der Kurzdarstellung dieser Datensätze kaum etwas ändern. Es geht bei der Ausgabe von Dubletten vor allem darum, die Dubletten dem Benutzer in einer übersichtlichen und verständlichen Form zu präsentieren.

Die zur Verfügung stehenden Alternativen werden nun im einzelnen vorgestellt und bewertet.

7.1 Alle dubletten Datensätze werden ausgegeben

Der Benutzer sieht alle Datensätze. Da sich die Datensätze kaum unterscheiden, ist die Präsentation redundant.

Erstes Beispiel: Das Buch *HyperWave* von Wolfgang Dalitz und Gernot Heyer, erschienen 1996 im dpunkt Verlag, werden im GBV, BVB und in der DDB nachgewiesen. In der Kurzdarstellung unterscheiden sich die Datensätze nicht. Der Benutzer sieht dreimal den gleichen Titel.

[1] Die in der Dublettenkontrolle untersuchten Attribute z.B. Autor, Titel, ISBN, Jahr etc.

```
Dalitz, Wolfgang; Heyer, Gernot: HyperWave. dpunkt, Verl.
für digitale Technologie, Heidelberg, 1996.
Vorhanden in der Bibliothek: GBV

Dalitz, Wolfgang; Heyer, Gernot: HyperWave. dpunkt, Verl.
für digitale Technologie, Heidelberg, 1996.
Vorhanden in der Bibliothek: BVB

Dalitz, Wolfgang; Heyer, Gernot: HyperWave. dpunkt, Verl.
für digitale Technologie, Heidelberg, 1996.
Vorhanden in der Bibliothek: DDB
```

Zweites Beispiel: Das Buch *Die Akazie* von Claude Simon, übersetzt aus dem Französischen von Eva Moldenhauer, erschienen bei Suhrkamp. Das Buch wurde von Suhrkamp in mehreren Auflagen veröffentlicht. Erkennbar ist dies für den Benutzer in der Kurzdarstellung nur anhand der unterschiedlichen Erscheinungsjahre. Einzelheiten zu diesem Buch werden in Abschnitt 7.3 Zusammenführen zu einem Datensatz (Seite 81) ausführlich erläutert.

```
Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1998.
Vorhanden in der Bibliothek: DDB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1998.
Vorhanden in der Bibliothek: BVB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1993.
Vorhanden in der Bibliothek: BVB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1993.
Vorhanden in der Bibliothek: DDB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1991.
Vorhanden in der Bibliothek: BVB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1991.
Vorhanden in der Bibliothek: BVB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer Suhrkamp. Frankfurt am Main, 1991.
Vorhanden in der Bibliothek: BVB

Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1991.
Vorhanden in der Bibliothek: DDB
```

In der Kurzdarstellung werden nur einige Daten ausgegeben, und zwar Autor, Titel, Verlag, Verlagsort und Jahr. Unterscheiden sich die Datensätze gegebenenfalls in der Auflage oder ISBN-Nummer, so wird diese Information dem Benutzer vorenthalten, und die Ausgabe der Datensätze scheint gleich zu sein. Ist allerdings das Jahr unterschiedlich, sieht der Benutzer diese Information sofort und weiß auch, in welcher Bibliothek sich das Werk in der jüngsten Ausgabe befindet.

Dieses Verfahren ist einfach und wird deshalb von vielen Meta-Suchmaschinen verwendet (siehe [KVK99], [Met99b], [DDB99a], [BVB99]). Es kommt auch ohne vorherige Dublettenprüfung aus - eine Sortierung der Datensätze nach Autor, Titel und Jahr liefert die gleichen Ergebnisse.

Das Präsentieren aller dubletten Datensätze bläht die Ausgabe jedoch unnötig auf. Der Benutzer muß länger auf die Ergebnisse warten, da mehr Daten vom Server übermittelt werden. Bei vielen Dubletten wird die Ausgabe unübersichtlich. Praktisch wird dem Benutzer die Dublettenkontrolle aufgebürdet, er muß gegebenenfalls eine lange Liste von Treffern durchlesen, die sich auf den ersten Blick zu gleichen scheinen.

Im Rahmen dieser Diplomarbeit wurde anhand mehrerer Beispiel eine manuelle Dublettenkontrolle durchgeführt (siehe Kapitel 6.2, Seite 61). Es hat sich dabei gezeigt, daß der Benutzer sehr leicht kleine Unterschiede in den Datensätzen übersieht und schnell ermüdet.

7.2 Auswahl eines Datensatzes

Von den dubletten Datensätzen wird ein Datensatz ausgewählt und in der Kurzdarstellung ausgegeben. Zusätzlich wird angegeben, in welcher Datenbank bzw. Bibliothek die Datensätze gefunden wurde.

Beispiel:

```
Dalitz, Wolfgang; Heyer, Gernot: HyperWave. dpunkt, Verl.
für digitale Technologie, Heidelberg, 1996.
Vorhanden in den Bibliotheken: GBV, BVB, DDB
```

Dieses Verfahren ist einfach zu realisieren. Die Auswahl des Datensatzes für die Ausgabe kann zufällig oder nach einem bestimmtem Kriterium erfolgen. Mögliche Kriterien sind:

- Nach Datenbank

  Gibt es den Datensatz in der Datenbank A und B, dann wird immer der Datensatz aus der Datenbank A gewählt. Die Präferenz für eine Datenbank kann z.B. aufgrund der geographischen Entfernung der Bibliotheken getroffen werden oder weil einer Datenbank ein höherer Nutzwert [2] als der anderen zugesprochen wird.

- Nach Jahr

  Unterscheiden sich zwei Datensätze im Erscheinungsjahr, so wird der Datensatz mit der höheren Jahreszahl ausgewählt. Hat der Benutzer die Wahl zwischen mehreren Ausgaben, will er in der Regel die aktuelle Ausgabe lesen - in der Hoffnung, daß die späteren Ausgaben weniger Fehler enthalten oder bei wissenschaftlichen Dokumentationen dem aktuellen Wissensstand angepaßt wurden.

[2] Die Einschätzung, ob eine Datenbank "gut" ist oder nicht, ist subjektiv. Für Bibliothekare in Deutschland mögen dies die Datensätze der Deutschen Bibliothek sein. Aus Mathematikersicht könnte die Entscheidung anders ausfallen - der Mathematiker wird vielleicht die mathematische Spezialbibliothek der Allgemeinbibliothek vorziehen.

- Nach Größe und Anzahl der Attribute

  Enthält ein Datensatz mehr Information - mehr Attribute (z.B. Schlagwörter) oder es steht mehr in den Attributen - so wird er gegenüber dem Datensatz mit weniger Information bevorzugt.

- Nach dem Zufallsprinzip

  Es werden keine Regeln vorgegeben, nach denen die Auswahl erfolgt. Es wird zufällig irgendein Datensatz aus den Dubletten gewählt.

7.3 Zusammenführen zu einem Datensatz

Man fügt die dubletten Datensätze zu einem neuen Datensatz zusammen und gibt den neuen Datensatz aus.

Beim Zusammenfügen wird jedes Feld aus dem Datensatz A mit dem betreffenden Feld aus dem Datensatz B verglichen und umgekehrt. Sind die beiden Werte gleich, wird der Wert eines der beiden Datensätze in den neuen Datensatz übernommen. Ist ein Feld nur im Datensatz A vorhanden, aber nicht im Datensatz B, dann wird der Wert aus dem Datensatz A genommen. Gleiches gilt für den umgekehrten Fall - gibt es ein Feld nur im Datensatz B, aber nicht in A, dann wird der Wert von B übernommen. Unterscheiden sich die Werte eines Feldes zwischen den Datensätzen, werden *beide* Werte in den neuen Datensatz übernommen. [3].

Beispiel:

```
Simon, Claude: Die Akazie : Roman. Aus dem Franz. von
Eva Moldenhauer. Suhrkamp. Frankfurt am Main, 1991, 1993, 1998.
Vorhanden in den Bibliotheken: 3 x DDB, 5 x BVB
```

Dieses Werk erschien 1991 bei Suhrkamp in 3 Auflagen (1., 2., 3. Auflage) mit identischer ISBN-Nummer als gebundenes Buch. 1993 brachte es Suhrkamp noch einmal als Taschenbuch heraus mit einer neuen ISBN-Nummer. Es erschien 1998 wiederum in der 1. Auflage als gebundenes Buch, mit neuer ISBN-Nummer. Die Seitenzahl ist bei jeder Ausgabe gleich, ebenso der Autor, die Übersetzerin, der Verlag und der Verlagsort.

Datenbank	Jahr	Auflage	ISBN	Seitenzahl	Format
DDB	1991	1	3-518-40349-4	354	gebunden
BVB	1991	1	3-518-40349-4	354	gebunden
BVB	1991	2	3-518-40349-4	354	gebunden
BVB	1991	3	3-518-40349-4	354	gebunden
DDB	1993	1	3-518-38732-4	354	Taschenbuch
BVB	1993	1	3-518-38732-4	354	Taschenbuch
DDB	1998	1	3-518-22302-X	354	gebunden
BVB	1998	1	3-518-22302-X	354	gebunden

Tabelle 7.1: Inkonsistente Vergabe der ISBN-Nummern

An diesem Beispiel zeigt sich recht deutlich die inkonsistente Vergabe von Auflagen und ISBN-Nummern. Es gibt zweimal eine 1. Auflage für das gebundene Buch, zuerst 1991 und

[3] Im MAB2-Format dürfen die meisten Felder mehrfach auftreten - aus Datenbanksicht ist dieses Verfahren also unproblematisch

dann nochmal 1998. Warum der Verlag 1998 wieder eine erste Auflage vergibt, ist unklar. Er hätte genauso gut eine 4. Auflage vergeben können. Zum Glück unterscheiden sich die beiden 1. Auflagen in der ISBN-Nummer. Nach dem ISBN-Standard sollen die Verlage für jede Auflage eines Titels eine ISBN-Nummer vergeben ([Ott94]). In diesem Fall hätte für die Auflagen aus dem Jahr 1991 (1., 2., 3.,) jeweils eine neue ISBN-Nummer vergeben werden müssen. Es ist nicht ersichtlich, warum sich die Verlage nicht immer an den ISBN-Standard halten. Offensichtlich wird die ISBN-Nummer häufig nicht geändert, wenn die Auflagen zeitlich kurz aufeinander folgen (ein paar Monate oder innerhalb eines Jahres). [4]

Die Dublettenkontrolle wird durch die inkonsistente Vergabe der ISBN-Nummern deutlich erschwert. Die ISBN-Nummer ist nicht als eindeutige Identifikationsnummer für eine physikalische Ausgabe geeignet.

Dieses Verfahren ist komplizierter als die beiden vorherigen und aufwendiger in der Implementierung. Es wird in Melvil ([Pay96], [Coy91]) eingesetzt.

7.4 Verwendete Variante

ZACK benutzt die 2. Alternative. Es wird von den dubletten Datensätzen ein Datensatz ausgewählt und in der Kurzdarstellung ausgegeben (Siehe Modul MAB2merge.pm in Anhang C, Seite 124).

Die Auswahl erfolgt nach Datenbank und Erscheinungsjahr. Zuerst wird nach einer Rangliste der Datenbanken verglichen und danach bei Gleichheit nach Jahr. So erhält man immer die neueste Ausgabe aus der Datenbank, der die höchste Priorität zugesprochen wurde. Beispiel: ist der Datensatz in der Deutschen Bibliothek vorhanden und in einer anderen Datenbank, dann wird der Datensatz von der Deutschen Bibliothek gewählt. Werden mehrere dublette Datensätze bei der Deutschen Bibliothek gefunden, wird der Datensatz mit der höheren Jahreszahl (neueste Ausgabe) ausgegeben.

Datenbank	Priorität
DDB	100
BVB	60
KOBV	40
sonstige	0

Tabelle 7.2: Priorität der Datenbanken bei der Ausgabe

Die Deutsche Bibliothek erhielt die höchste Priorität, da sie für die meisten Bibliothekare in Deutschland die Referenz ist.

7.5 Praktische Ergebnisse

Die Kurztrefferliste wird durch die Dublettenkontrolle kürzer und übersichtlicher. In der Praxis hat sich gezeigt, daß die Kurztrefferliste nach der Dublettenkontrolle um ein Drittel kürzer oder nur noch halb so lang war wie vor der Dublettenkontrolle.

In der Tabelle 7.3 werden die Ergebnisse einer parallelen Suche mit Dublettenkontrolle ausgewertet. Die Anfragen wurden im Februar 1999 an die Datenbanken der Deutschen Bibliothek, des Gemeinsamen Bibliotheksverbundes, des Bibliotheksverbundes Bayern und der Technischen

[4] Aus Sicht eines Buchkäufers kann dies sogar vorteilhaft sein. Er bestellt ein Buch mit der ISBN-Nummer des betreffenden Buches. Statt der Antwort: "*dieses Buch ist vergriffen und wird nicht mehr ausgeliefert*" oder "*diese ISBN gibt es nicht mehr*" erhält er das Buch in der letzten lieferbaren Auflage.

Universität Braunschweig gestellt. Weitere Details zu diesen Beispielen sind in Kapitel 5 Normierung (Seite 40) beschrieben.

Anfrage	Anzahl der Datensätze	Anzahl der Werke	in Prozent
titel=akazie	60	40	66,6 %
titel=birnbaum	342	165	48,2 %
titel=perl	345	169	49,0 %
titel=karstadt	61	44	72,1 %
titel=pankow	212	129	60,9 %
autor=dalitz,wolfgang	28	13	46,4 %
autor=rusch,beate	9	4	44,4 %

Tabelle 7.3: Länge der Kurztrefferliste nach Dublettenkontrolle

Legende

Anzahl der Datensätze: Anzahl der Treffer, die in allen Datenbanken gefunden wurde.

Anzahl der Werke: Anzahl der Treffer nach Dublettenkontrolle.

In Prozent: Auf wieviel Prozent die Kurztrefferliste durch die Dublettenkontrolle geschrumpft ist.

7.6 Zusammenfassung

Jedes Verfahren hat Vor- und Nachteile. Einfach zu realisieren sind die Verfahren *alle dubletten Datensätze ausgeben* (Alternative 1) und *Auswahl eines Datensatzes* (Alternative 2). Das *Zusammenführen mehrerer Datensätze zu einem neuen* (Alternative 3) ist aufwendiger. Die Präsentation sämtlicher Dubletten bläht die Ausgabe unnötig auf und ist daher kaum für eine benutzerfreundliche Darstellung geeignet. Bei den beiden anderen Alternativen (Auswahl bzw. Zusammenführen) wird die Kurztrefferliste in der Praxis um ein Drittel bis zur Hälfte kürzer.

Der Benutzer hat immer die Möglichkeit, auf jeden einzelnen der dubletten Datensätze zuzugreifen. Der Computer sagt nur, *"dieses Werk gibt es zweimal in der Bibliothek X und einmal in der Bibliothek Y"*, letztlich muß immer der Benutzer die Entscheidung treffen, aus welcher Datenbank er den Datensatz übernehmen will.

Kapitel 8

Praktische Ergebnisse einer verteilten Suche

In diesem Kapitel wird untersucht, wie erfolgreich die verteilte Suche in der Praxis wirklich ist. Dazu werden die Anfragen von Bibliothekaren ausgewertet, die mit *ZACK* gesucht haben - einmal für die Anfragen eines Tages und einmal für den Zeitraum von mehreren Monaten.

Das System *ZACK* wird seit mehreren Monaten von Bibliothekaren der Europa-Universität Viadrina Frankfurt (Oder) und der Brandenburgischen Technischen Universität Cottbus für die Erfassung von Büchern regelmäßig genutzt (siehe auch im Anhang D, Seite 136). Die Bibliothekare wählen dabei, in welcher Datenbank sie suchen - eine parallele Suche in mehreren Datenbanken gleichzeitig war zu dem Zeitpunkt der Analyse noch nicht möglich.

In den Log-Dateien des Webservers werden alle Zugriffe auf das WWW-Z39.50-Gateway protokolliert. Dazu gehört auch die Information, wonach die Nutzer gesucht haben. Die Suchanfragen der Brandenburger Bibliothekare beziehen sich zu 3/4 auf das Attribut ISBN-Nummer. Für diese Analyse werden deshalb nur die Anfragen mit ISBN-Nummer ausgewertet. Die ISBN-Nummer eignet sich sehr gut für einen einfachen Test. Man sucht nach der ISBN-Nummer und erhält als Ergebnis, daß der Titel vorhanden ist oder nicht vorhanden ist. Die Ergebnismenge ist klein (kein Treffer, 1 Treffer, 2 Treffer, eventuell auch 3 Treffer).

Um herauszufinden, welche ISBN-Nummern die Benutzer gefunden haben, werden die Anfragen nochmal an die Datenbanken der Deutschen Bibliothek (DDB), des Bibliotheksverbundes Bayern (BVB) und des Gemeinsamen Bibliotheksverbundes (GBV) gestellt und die Ergebnisse statistisch ausgewertet.

8.1 Auswertung der Suchanfragen eines Tages

Auswertung der ISBN-Suchanfragen an das WWW-Z39.50-Gateway vom Montag, den 7. Dezember 1998, von 8 Uhr bis 15 Uhr 45.

Es wurden 55 verschiedene ISBN-Nummern gesucht. Davon sind durchschnittlich 80-90% in jeder Datenbank vorhanden.

In allen Datensätzen gefundene ISBN-Nummern

```
0-387-98254-X  3-211-83192-4  3-410-32843-2  3-412-13996-3
3-455-10362-6  3-468-28291-5  3-499-13316-4  3-518-06521-1
3-518-06525-4  3-518-11985-0  3-528-03875-6  3-528-05662-2
3-528-06660-1  3-540-56898-0  3-540-58355-6  3-540-60311-5
3-540-61757-4  3-540-94098-7  3-540-96678-1  3-545-33149-0
3-593-35351-2  3-7253-0451-3  3-7253-0455-6  3-7253-0488-2
3-7643-3817-2  3-7643-5124-1  3-7643-5333-3  3-7759-0210-4
```

3-8058-3129-3	3-8114-1895-5	3-8226-1886-1	3-8272-5088-9
3-8272-5400-0	3-85447-556-X	3-87988-067-0	3-87988-079-4
3-88474-059-8	3-88474-060-1	3-89360-928-8	3-927282-20-0

Interessant sind besonders die Fälle, bei denen die ISBN-Nummer nicht gefunden wurde:

Datenbanken	Anzahl der nicht gefundenen Datensätze
DDB	6
GBV	5
BVB	10

Tabelle 8.1: Anzahl der nicht gefundenen ISBN-Nummern in einer Datenbank

40 ISBN-Nummern (73%) sind in allen 3 Datenbanken vorhanden. Im ungünstigsten Fall, wenn jedesmal die nicht geeignete Datenbank zur Suche gewählt worden wäre, werden 15 ISBN-Nummern (27%) nicht gefunden.

In der folgenden Tabelle werden die ISBN-Nummern aufgeschlüsselt, die nicht in jeder Datenbank gefunden wurden. Das “x” steht für gefunden in der betreffenden Datenbank, leere Felder stehen für null Treffer. Beispiel: Die ISBN-Nummer 3-406-36911-1 gibt es bei der DDB und beim GBV, nicht aber beim BVB.

ISBN-Nummer	DDB	GBV	BVB
3-351-02058-9		x	
3-406-36911-1	x	x	
3-427-64951-2		x	x
3-468-91112-2	x		x
3-486-23992-9		x	x
3-518-06523-8	x	x	
3-519-12275-8	x	x	
3-534-13348-X		x	
3-540-64767-8	x		x
3-581-69064-0			
3-598-23737-5	x	x	
3-86135-620-1	x		x
3-89451-024-2		x	
3-925795-27-8	x		
3-929285-07-X	x	x	

Tabelle 8.2: ISBN-Nummern, die in einer oder mehreren Datenbanken nicht vorhanden sind

Aus dieser Tabelle kann man ablesen, ob eine Kombination von zwei oder drei Datenbanken bessere Ergebnisse bringt:

Kombination von zwei Datenbanken	Anzahl der nicht gefundenen Datensätze
DDB und GBV	1
DDB und BVB	4
GBV und BVB	2

Tabelle 8.3: Anzahl der nicht gefundenen ISBN-Nummern in zwei Datenbanken

Kombination von drei Datenbanken	Anzahl der nicht gefundenen Datensätze
DDB und GBV und bVB	1

Tabelle 8.4: Anzahl der nicht gefundenen ISBN-Nummern in drei Datenbanken

Eine verteilte Suche bringt deutlich bessere Ergebnisse. Die Anzahl der nicht gefundenen Dokumente sinkt von 10-20% auf 2%. Der Test ist nicht repräsentativ - allerdings praxisnah. Es werden Anfragen an die Datenbanken geschickt, so wie sie auch von den Bibliothekaren bei der täglichen Arbeit gestellt werden. Bei diesem Test sind neuere Werke überrepräsentiert. Es wurde überwiegend nach Büchern gesucht, die kürzlich von der Bibliothek erworben wurden. Einige dieser Bücher sind deshalb noch nicht in allen großen Bibliotheksverbünden erfaßt.

8.2 Auswertung der Suchanfragen über mehrere Monate

Auswertung der ISBN-Suchanfragen an das WWW-Z39.50-Gateway vom 10. Mai bis 8. Dezember 1998.

Es wurden 3824 verschiedene ISBN-Nummern gesucht. Davon sind durchschnittlich 87-92% in jeder Datenbank vorhanden. 3135 ISBN-Nummern (81,98%) sind in allen 3 Datenbanken vorhanden. Im ungünstigsten Fall, wenn jedesmal die nicht geeignete Datenbank zur Suche gewählt worden wäre, werden 689 ISBN-Nummern (18,02%) nicht gefunden.

Datenbanken	Anzahl der gefundenen Datensätze	in Prozent
DDB	3550	92,83%
GBV	3344	87,45%
BVB	3543	92,65%

Tabelle 8.5: Massentest ISBN-Nummer, gefunden in einer Datenbank

Kombination von zwei Datenbanken	Anzahl der gefundenen Datensätzen	in Prozent
DDB und GBV	3674	96,08%
DDB und BVB	3642	95,24%
GBV und BVB	3712	97,07%

Tabelle 8.6: Massentest ISBN-Nummer, gefunden in zwei Datenbanken

Kombination von drei Datenbanken	Anzahl der gefundenen Datensätzen	in Prozent
DDB und BVB und GBV	3726	97,44%

Tabelle 8.7: Massentest ISBN-Nummer, gefunden in drei Datenbanken

3726 ISBN-Nummern sind in mindestens einer Datenbank vorhanden, d.h. wenn jedesmal die passende Datenbank gewählt wird, liegt die Trefferquote bei 97,44%.

Bei der Suche in einer Datenbank wurden 13-8% der Dokumente nicht gefunden. Die Anzahl der nicht gefundenen Dokumente sinkt auf 3-5% bei der Suche in zwei Datenbanken. Bei der Suche in drei Datenbanken sinkt die Anzahl der nicht gefundenen Dokumente auf 2,5%.

Dieser Test konnte in kurzer Zeit durchgeführt werden. Es wurde nur geprüft, ob die ISBN-Nummern in den Datenbanken vorhanden sind und nicht die Datensätze selbst geholt. Letzteres hätte die Bibliothekssysteme zu stark belastet.

Datenbank	ISBN-Anfragen pro Sekunde
DDB	7
BVB	20
GBV	3

Tabelle 8.8: Geschwindigkeit der Datenbanken bei der Suche nach ISBN-Nummern

8.3 Zusammenfassung

Für den Anwender bringt die Suche in mehreren Datenbanken gleichzeitig deutlich bessere Ergebnisse. Die Anzahl der nicht gefundenen Dokumente sinken von rund 10% auf rund 2,5% - die Fehlerquote verringert sich bei der Suche auf ein Viertel.

Bei diesem Test sind neuere Werke überrepräsentiert. Es wurde nach Büchern gesucht, die von den Brandenburger Biliothekaren in Cottbus und Frankfurt (Oder) erworben wurden. Einige dieser Bücher sind erst kürzlich erschienen und deshalb noch nicht in allen großen Bibliotheksverbünden erfaßt.

Diese Auswertung bezog sich nur auf die Suche nach ISBN-Nummern. Es wäre wünschenswert, einen ähnlichen Test auch mit den Attributen *Autor* und *Titel* durchzuführen. Der Aufwand dafür ist allerdings wesentlich höher als beim Attribut *ISBN*.

Kapitel 9

Probleme im laufenden Betrieb

In diesem Kapitel werden kleinere und größere Probleme beschrieben, die in der praktischen Nutzung der Z39.50-Server aufgetreten sind. Der Wartungsaufwand für die Nutzung der Z39.50-Server ist wesentlich höher als zunächst angenommen wurde: die Konfigurationsprobleme mit den Z39.50-Servern unterschiedlicher Bibliothekssysteme, unterschiedlicher Hersteller und unterschiedlicher Bibliotheken und Verbünde werden im Detail aufgeführt.

Auf die Probleme wird jetzt in den folgenden Abschnitten 9.1 **Unterschiedliche Erfassungspraktiken**, 9.2 **Austauschformat MAB2** und 9.3 **Z39.50-Server** im einzelnen eingegangen.

9.1 Unterschiedliche Erfassungspraktiken

Für das Katalogisieren gibt es Regelwerke. Das in Deutschland am weitesten verbreitete ist RAK (Regeln für die alphabetische Katalogisierung [HP96], [RAK99]). Dieses Regelwerk ist sehr umfangreich und detailliert. Trotzdem läßt es noch genügend Interpretationsspielraum für die Aufnahme neuer Werke, die eine Dublettenkontrolle deutlich erschweren oder zum Teil unmöglich machen.

Exemplarisch wird dies anhand der Aufnahme einiger Preprints des Konrad-Zuse-Zentrums für Informationstechnik Berlin aus den Jahren 1992 und 1993 verdeutlicht. Die Preprints sind Online auf dem Web-Server verfügbar [1]. Die vier Preprints haben dieselben Autoren und einen ähnlichen Titel. Sie befassen sich mit dem gleichen Thema - Algorithmen zur Optimierung von Routen. Die Preprints sind aber eigenständige Werke und erschienen anschließend in 3 verschiedenen mathematischen Zeitschriften.

Nachfolgend wird die Aufnahme der Preprints vom Bibliotheksverbund Bayern (BVB) mit der der Deutschen Bibliothek verglichen. Es wird untersucht, wie Autor und Titel erfaßt wurden und welche Beziehungen es zwischen den Datensätzen gibt. Außerdem wird geprüft, ob man die Preprints unter ihrem bekannten Titel bzw. ihren Autoren in den Datenbanken findet.

Aus Übersichtsgründen wurden der Gemeinsame Bibliotheksverbund (GBV) und der KOBV nicht in diese Untersuchung einbezogen. Der KOBV hat die (Fremd-)Daten der Deutschen Bibliothek übernommen und liefert deshalb die gleichen Ergebnisse wie der Z39.50-Server der Deutschen Bibliothek. Der GBV ist bei der Katalogisierung ähnlich wie der BVB vorgegangen.

Konrad-Zuse-Zentrum

Erfassung der Preprints in der Bibliothek des Konrad-Zuse-Zentrum für Informationstechnik Berlin:

[1] http://www.zib.de/bib/pub/pw/index.de.html

1992

- SC 92-08 Martin Grötschel, Alexander Martin, Robert Weismantel: Packing Steiner Trees: Polyhedral Investigations.
- SC 92-09 Martin Grötschel, Alexander Martin, Robert Weismantel: Packing Steiners Trees: A Cutting Plane Algorithm and Computational Results.

1993

- SC 93-01 Martin Grötschel, Alexander Martin, Robert Weismantel: Packing Steiner Trees: Further Facets.
- SC 93-02 Martin Grötschel, Alexander Martin, Robert Weismantel: Packing Steiner Trees: Separation Algorithms.

Bibliotheksverbund Bayern

Der Bibliotheksverbund Bayern (BVB) hat die vier Preprints als Hauptsätze (h) aufgenommen. Der Titel selbst wurde in jedem Datensatz unterschiedlichen Feldern zugeordnet. Der erste Teil "*Packing Steiner Trees*" wurde als Hauptsachtitel (Feld 331) und der zweite Teil nach dem Doppelpunkt als Zusatz zum Hauptsachtitel (Feld 335) aufgefaßt.

Die Datensätze kann man ohne Schwierigkeiten über die Autoren oder den Titel finden. Der Titel muß dabei vollständig angegeben werden ("*Packing Steiner trees polyhedral investigations*") oder mit Rechtstrunkierung ("*Packing Steiner trees*").

Zur besseren Übersicht werden die Datensätze vom BVB hier leicht gekürzt im MAB2-Kategorienformat ausgegeben. Nicht dargestellt werden interne oder automatisch erzeugte Felder (z.B. Feldnummern 002-099) sowie Schlagwörter.

```
### 00587nM2.01000024      h
001 00066446503
100 Grötschel, Martin
104aMartin, Alexander
108aWeismantel, Robert
331 Packing Steiner trees
335 polyhedral investigations
410 Berlin
412 Konrad-Zuse-Zentrum für Informationstechnik Berlin
425a1992
433 29 S. : graph. Darst.
451 Konrad-Zuse-Zentrum für Informationstechnik <Berlin>: Preprint SC ; 1992,8
568 93,B15,0375

### 00663nM2.01000024      h
001 00072235217
100 Grötschel, Martin
104aMartin, Alexander
108aWeismantel, Robert
331 Packing Steiner trees
335 a cutting plane algorithm and computational results
410 Berlin
412 ZIB
425a1992
433 33 S. : graph. Darst.
451 Konrad-Zuse-Zentrum für Informationstechnik <Berlin>: Preprint SC ; 1992,9
568 93,B19,0226
```

```
### 00562nM2.01000024      h
001 0009999500x
100 Grötschel, Martin
104aMartin, Alexander
108aWeismantel, Robert
331 Packing Steiner trees
335 further facets
410 Berlin
412 Konrad-Zuse-Zentrum für Informationstechnik
425a1994
433 20 S. : graph. Darst.
451 Konrad-Zuse-Zentrum für Informationstechnik <Berlin>: Preprint SC ; 1993,1

### 00591nM2.01000024      h
001 00091108306
100 Grötschel, Martin
104aMartin, Alexander
108aWeismantel, Robert
331 Packing Steiner trees
335 separation algorithms
410 Berlin
412 Konrad-Zuse-Zentrum für Informationstechnik
425a1993
433 31 S. : graph. Darst.
451 Konrad-Zuse-Zentrum für Informationstechnik <Berlin>: Preprint SC ; 1993,2
568 94,B11,0198
```

Die Deutsche Bibliothek

Die Deutsche Bibliothek hat die vier Preprints zu einem mehrbändigen begrenzten Werk zusammengefaßt. Es gibt einen Hauptsatz und vier untergeordnete Untersätze. Der Hauptsatz enthält die Autoren, den Titel (bzw. einen Teil davon), den Verlag und Verlagsort. Die Untersätze enthalten nur den Titel (bzw. den restlichen Teil des Titels), Jahr, Seitenzahl, die Nummer des Preprints und eine Referenz auf den Hauptsatz (Feld 010) (zu den Satztypen siehe auch im Anhang A.1 Verteilung der Satztypen in der DNB, Seite 103).

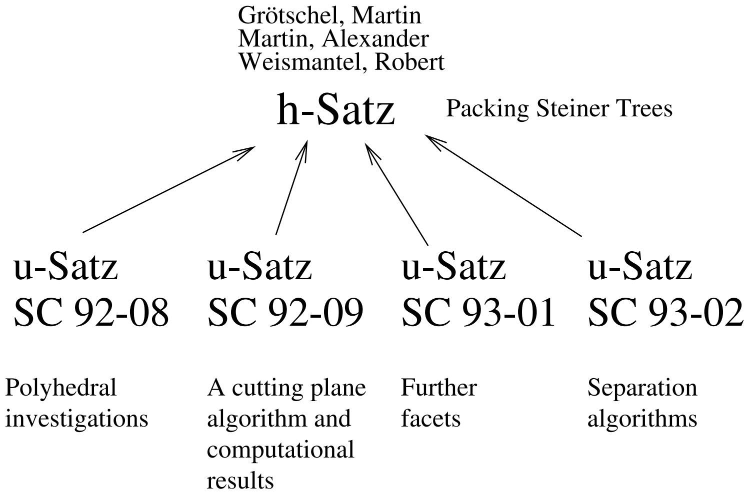


Abbildung 9.1: Packing Steiner Trees, h- und u-Sätze der Deutschen Bibliothek

Es gibt nur einen Verweis von den Untersätzen auf den Hauptsatz im MAB2-Format. Der umgekehrte Weg geht nicht, d.h. wenn man einen Hauptsatz gefunden hat, weiß man nicht, wieviele und welche Untersätze es zu diesem Datensatz gibt.

Es ist technisch möglich, diese Referenzen nachträglich in der Datenbank zu erzeugen und über das Bibliothekssystem dem Benutzer zugänglich zu machen. Vor der Ausgabe der Datensätze prüft das System dann, ob es zu diesem h-Satz in der internen Datenstruktur auch u-Sätze gibt. Die Netz- und Systembelastung ist in diesem Fall minimal, da alle Daten lokal vorliegen und indexiert sind.

Über Z39.50 ist dies praktisch nicht möglich. Will man wissen, welcher u-Satz zu welchem h-Satz (und umgekehrt) gehört, muß man für *jeden einzelnen* Datensatz eine neue Anfrage stellen. Die Kommunikation zwischen Z39.50-Server und Z-Client würde deutlich länger dauern und zu unzumutbaren Wartezeiten für den Benutzer führen.

Zur besseren Übersicht werden die Datensätze von der DDB hier leicht gekürzt im MAB2-Kategorienformat ausgegeben. Nicht dargestellt werden interne oder automatisch erzeugte Felder (z.B. Feldnummern 002-099) sowie Schlagwörter.

```
### 00416nM2.01200024      h
001 942769899
100 Grötschel, Martin
104aMartin, Alexander
108aWeismantel, Robert
331 Packing Steiner trees
410 Berlin
412 ZIB

### 00429nM2.01200024      u
001 930533054
010 942769899
331 Polyhedral investigations
425a1992
433 29 S.
451 Preprint / Konrad-Zuse-Zentrum für Informationstechnik Berlin ; 92,8
574 95,B03,0422

### 00474nM2.01200024      u
001 930671333
010 942769899
331 A @cutting plane algorithm and computational results
425a1992
433 33 S.
451 Preprint / Konrad-Zuse-Zentrum für Informationstechnik Berlin ; SC 92,9
574 95,B03,0422

### 00416nM2.01200024      u
001 942769295
010 942769899
331 Further facets
425a1994
433 20 S.
451 Preprint / Konrad-Zuse-Zentrum für Informationstechnik Berlin ; SC 93,1
453r551857102
574 95,B03,0442

### 00439nM2.01200024      u
001 940084902
010 942769899
331 Separation algorithms
425a1993
433 31 S.
451 Preprint / Konrad-Zuse-Zentrum für Informationstechnik Berlin ; SC 93,2
574 95,B03,0422
```

Wie beim Bibliotheksverbund Bayern wurde der Titel in jedem Datensatz unterschiedlichen Feldern zugeordnet. Der Anfang "*Packing Steiner Trees*" wurde als Hauptsachtitel (Feld 331) in den Hauptsatz (h) aufgenommen und alles, was nach dem Doppelpunkt steht, als Hauptsachtitel (Feld 331) im Untersatz (u) aufgenommen.

Die u-Sätze konnten nur mit sehr viel Aufwand gefunden werden. Da in den u-Sätzen die Autoren fehlen, kann man sie auch nicht über den Autorindex finden. Die Suche mit dem Titel gestaltet sich schwierig, da a) der erste Teil des Titels in den u-Sätzen fehlt und b) die Deutsche Bibliothek keine Phrasensuche im Titel erlaubt. Beispiel: das Werk "*Packing Steiner Trees: Polyhedral Investigations*" findet man nur, wenn man nach *titel="Polyhedral" UND titel="Investigations"* sucht. Mit der Suche nach *autor=groetschel* oder *titel="Polyhedral Investigations"* hat man keinen Erfolg.

9.2 Austauschformat MAB2

9.2.1 Interpretation des Standards

Das Maschinelle Austauschformat für Bibliotheken (MAB) entstand zum Austausch von Daten. Der MAB-Standard ([MAB99], [DDB99b]) wurde von der Deutschen Bibliothek in Zusammenarbeit mit den Bibliotheksverbünden und den wichtigsten Einrichtungen des deutschen Bibliothekswesens entwickelt und 1995 verabschiedet. MAB2 wird laufend weiterentwickelt. Bisher sind zwei Ergänzungslieferungen erschienen. Die zweite Auflage erscheint zugleich mit der dritten Ergänzungslieferung im April 1999. Detaillierte Informationen zu bibliothekarischen Datenformaten finden sich in [Eve99] und [Eve94].

Die Anwendung des MAB-Formats wird in der MAB-Dokumentation verbindlich geregelt. Trotzdem bleibt jeder Bibliothek und jedem Verbund ein gewisser Spielraum bei der Auslegung des Standards. Ein MAB2-Datensatz von der Deutschen Bibliothek kann sich durchaus von einem MAB2-Datensatz des Gemeinsamen Bibliotheksverbundes unterscheiden - trotz gleicher Regeln bei der Katalogisierung (RAK) und beim Austauschformat MAB2. Die Deutsche Bibliothek hat ihre Interpretation des MAB2-Standardes dokumentiert und als DDB-MAB2 Standard ([DNB96]) veröffentlicht. Von den anderen Verbünden liegt keine Dokumentation vor, wie sie MAB2 in ihren Systemen verwenden und austauschen.

Die Deutsche Bibliothek aktualisiert ihre Z39.50 Serversoftware regelmäßig mehrmals im Jahr. Praktisch nach jeder Änderung der Software ändert sich auch das Format der gelieferten Datensätze. Manchmal fehlen einige Felder - entweder im Kurzformat oder im Vollformat. Oder es wird der Inhalt der Felder geändert - zum Beispiel durch die Einführung des Füllzeichen "|" (Pipesymbol).

Die Deutsche Bibliothek informiert ihre Partner regelmäßig (und rechtzeitig) über ihre Wartungstermine. Allerdings werden nicht Änderungen an den Datensätzen gemeldet. Diese Änderungen wurden im Rahmen der Arbeit an *ZACK* eher zufällig entdeckt.

9.2.2 Systemspezifische Umsetzung des Standards

Das MAB2-Format ist ein Format zum Austausch von Datensätzen. Intern werden in den Bibliothekssystemen eigene Formate (z.B PICA-MARC) verwendet. Beim Export der Datensätze wird festgelegt, wie das interne Format auf die MAB2-Felder abgebildet wird. Jeder Hersteller und jede Bibliothek hat eigene Vorstellungen davon, wie die Umsetzung erfolgt.

Als Beispiel: Bei Büchern mit mehreren ISBN-Nummern benutzen die meisten Bibliothekssysteme mehrere Felder, für jede ISBN-Nummer eines. Das System allegro (Technische Universität Braunschweig) schreibt dagegen alle ISBN-Nummern in ein Feld, da bei allegro Felder nicht wiederholt werden können.

- Technische Universität Braunschweig

  ```
  540aISBN 3-540-56740-2 = 0-387-56740-2
  ```

- Die Deutsche Bibliothek

  ```
  540aISBN 3-540-56740-2 (Berlin ...)  Pp.  :  DM 148.00
  540aISBN 0-387-56740-2 (New York ...)  Pp.
  ```

- Bibliotheksverbund Bayern

  ```
  540aISBN 3-540-56740-2
  540aISBN 0-387-56740-2
  ```

9.2.3 MAB2-Kurzformat

Für die Darstellung einer großen Anzahl von Treffern ist es nicht nötig, die vollständigen Datensätze an den Benutzer zu schicken. Deshalb wird im Z39.50 Protokoll ein Kurzformat (brief) eingeführt. Das Kurzformat enthält nur die wichtigsten Felder (Autor, Titel, ISBN, Jahr), die für eine Kurzübersicht notwendig sind. Dagegen wird im Vollformat (full) der komplette Datensatz übermittelt, mit Schlagwörtern, Identifikationsnummern, Verweisen auf andere Datensätze etc. Im Z39.50 Protokoll ist allerdings nicht definiert, wie das Kurzformat im einzelnen aufgebaut ist.

Auch im MAB2-Standard ist nicht definiert, welche Felder zum Kurzformat gehören. Meistens werden nur die Felder 001 bis einschließlich Feld 540 (ISBN) ausgegeben. Die restlichen Felder (z.B. Schlagwörter etc.) werden abgeschnitten. Ein Datensatz im Kurzformat unterscheidet sich also vom Vollformat nur durch einige fehlende Felder [2] am Ende des Datensatzes. Seit Anfang März hat die Deutsche Bibliothek ihr Kurzformat geändert. Das Feld 100 (Name der 1. Person) wird nicht mehr ausgegeben, wenn es sich um einen Herausgeber (Indikator b) handelt. Damit ist das MAB2-Kurzformat für die hier implementierte Dublettenkontrolle nicht mehr geeignet.

9.3 Z39.50 Server

9.3.1 Allgemeines

Austauschformate

Nicht alle Server unterstützen alle Austauschformate. Zum Beispiel liefert der Bayerische Bibliotheksverbund kein USMARC, das System SISIS (FH Potsdam, FH Brandenburg) kein SUTRS, das System allegro (TU Braunschweig) kein UNIMARC und ALEPH (KOBV) je nach Datenbank nur MAB2 oder nur USMARC (siehe [MAB99], [MARil], [UNI98]).

Target-Profile

Ein Target-Profil ist eine detaillierte technische Beschreibung des Z39.50-Servers. Es wird beschrieben, welche Anfragen, Zeichensätze, Formate, Datenbanken, Attribute etc. vom Server (Target) unterstützt werden.

Von vielen Servern sind die Profile [3] nicht bekannt. Es ist nicht dokumentiert, wie z.B. das Attribut 1 (Personal Name) intern auf die Datenbank abgebildet wird. Sind das nur Werke von

[2] MAB2 Feldnummern 541 bis 999

[3] Rühmliche Ausnahmen: Die Deutsche Bibliothek und der Bibliotheksverbund Bayern haben die Target-Profile auf ihrem Web-Server veröffentlicht

einem Autor oder auch Werke über einen Autor [4] ? Oder sind auch die Herausgeber mit dem Attribut 1 erfaßt? Sind im Attribut Publisher (1018) nur die Verlagsnamen (z.B. Suhrkamp) oder auch die Verlagsorte (z.B. Frankfurt am Main) suchbar?

Im Z39.50-Standard ist nicht definiert, wie die jeweiligen BIB1-Attribute auf die Datenbank abgebildet werden ([BIB95], [Z3995]). Es gibt Richtlinien, in denen eine Umsetzung von BIB1 auf USMARC vorgeschlagen wird (siehe [BIB95]). Für das MAB2-Format gibt es keine entsprechenden Empfehlungen.

BIB1 Attribute

Viele Server unterstützen nur eine bestimmte Anzahl von BIB1-Attributen. Es ist häufig nicht bekannt, welche unterstützt werden und ob einige davon Synonyme für andere Attribute sind (siehe auch [BIB98], [BIB95]).

Index

Für die Suche in der Datenbank wird ein Index angelegt. Für unterschiedliche Attribute werden unterschiedliche Indexe angelegt, z.B. ein Index für Autor, ein Index für Titel und ein Index für ISBN-Nummern. Die Bibliothek oder der Verbund legt fest, welche MAB2-Felder für den Aufbau eines Indexes benutzt werden. Zum Beispiel die Felder 331, 335 und 310 für das Titelregister. Außerdem wird festgelegt, ob in dem Index Wörter oder Wortgruppen (Phrasen) verwendet werden.

Als Beispiel: Im Titelindex der Deutschen Bibliothek kann nur nach Wörtern gesucht werden. Die Suche nach einem exakten Titel (Wortgruppe) ist nicht möglich. Will man nach einem exakten Titel suchen, so muß man alle Wörter des Titels durch eine Boolesche "*UND*"-Verknüpfung verbinden und erhält eventuell einen größeren Informationsballast.

"*Der Botanische Garten*" $\Rightarrow$ "*Botanischer*" UND "*Garten*"

Der Bayerische Bibliotheksverbund (BVB) und der Gemeinsame Bibliotheksverbund (GBV) dagegen unterstützen die Suche nach Wortgruppen (Phrasen). Dieses Beispiel veranschaulicht, daß eine verteilte Suche mit "*Der Botanische Garten*" nicht ohne weiteres möglich ist. Dieselbe Anfrage kann bei jeder Datenbank zu unterschiedlichen Ergebnissen führen, je nachdem, wie der Datenbankindex aufgebaut wurde und wie die Standardeinstellungen [5] des Z39.50-Servers sind.

Zweites Beispiel: Mit dem System ALEPH 500 (KOBV) konnte man zunächst nicht nach dem vollständigen Namen eines Autors suchen. Statt dessen mußte durch eine UND-Verknüpfung nach Nachname und Vorname des Autors getrennt gesucht werden:

"*autor=Dalitz, Wolfgang*" $\Rightarrow$ "*autor=Dalitz UND autor=Wolfgang*" [6]

Drittes Beispiel: Der Index für die Autoren ist üblicherweise nach dem Schema "*Nachname, Vorname*" aufgebaut. Um nach dem Autor "*Wolfgang Dalitz*" zu suchen, muß der Benutzer "*Dalitz, Wolfgang*" eingeben. Der Autorname kann auch trunkiert werden. Zum Beispiel findet

[4] Im BIB1-Standard ist ein Attribut für Autor (Bücher von einem Autor) definiert. Praktisch wird es aber nicht genutzt, sondern nur das Attribut *Personal Name*. Die meisten Benutzer erwarten wahrscheinlich, daß bei der Suche nach "Thomas Mann" im Menü *Autor* auch die Bücher über ihn, Nachdrucke, Kritiken, Übersetzungen etc. gefunden werden. Ob die Erwartungen der Benutzer wirklich so sind, sei dahingestellt - über dieses Thema gibt es durchaus verschiedene Auffassungen. Die deutsche Bibliothek liefert bei der Suche nach "Thomas Mann" mit dem Attribut *Personal Name* auch die Bücher über ihn.

[5] Zum Beispiel Suchen mit Trunkierung oder ohne Trunkierung

[6] Groß- und Kleinschreibung ist egal

man mit "*Dalitz, W*" die Autoren "*Wolfgang Dalitz*" und "*Wilhelm Dalitz*" in der deutschen Bibliothek (DDB).

In der Tabelle 9.1 werden die Ergebnisse einer verteilten Suche nach dem Autor "*Wolfgang Dalitz*" in verschiedenen Datenbanken gegenübergestellt. Dabei wird untersucht, welche Schemata von den Datenbanken unterstützt werden und wieviele Treffer jeweils zurückgeliefert werden.

Anfrage	rechts trunkiert	TUBS	BVB	KOBV	DDB	FH-Potsdam	GBV	SW
"dalitz, wolfgang" (Nachname, Vorname)	nein	3	10	9	7	4	9	2
"dalitz,wolfgang" (ohne Leerzeichen nach dem Komma)	ja	3	10	9	7	4	8 [7] von 9	0 [8]
"dalitz wolfgang" (ohne Komma)	nein	0 [9]	10	9	0	4	0	0
"dalitz, w" (Vorname abgekürzt)	ja	4	16	0 [10]	14	4	19	? [11]
"dalitz, w" (Vorname abgekürzt)	nein	0	0	0	7	0	1	0
"wolfgang, dalitz" (Vorname, Nachname)	nein	0	0	0	0	4 [12]	0	?

Tabelle 9.1: Verteilte Suche mit Attribut Autor

Das Schema *Nachname, Vorname* wird von allen Datenbanken korrekt unterstützt. Einige Systeme - z.B. SISIS - unterstützen auch das Schema "*Vorname, Nachname*". TUBS, DDB, GBV und SW liefern null Treffer, wenn das Komma zwischen Nachname und Vorname fehlt. BVB, KOBV und FH-Potsdam liefern auch ohne Komma dieselben Ergebnisse wie mit Komma.

Die Anzahl der Indexe ist durch die verwendete Software und Hardware begrenzt. Der Aufbau eines Indexes kostet viel Zeit und Speicherplatz. Die Anzahl der Indexe ist immer ein Kompromiß zwischen den Wünschen der Anwender (möglichst viele) und der realen EDV.

Zuviele Treffer

Bei einer Maximalanzahl von Treffern (z.B. mehr als 10.000) brechen viele Server die Suche ab. Die Deutsche Bibliothek liefert bei der Suche nach dem Autor "Schneider" null Treffer. Bei der Suche nach "Schneider,A" werden immerhin schon 702 Treffer gefunden. Der GBV liefert bei der Suche nach dem Autor "Schneider" 9999 Treffer, der BVB 17518 Treffer. Das System des KOBV hat keine klare Trefferobergrenze, bricht aber nach ca. 10 Sekunden mit der Suche ab. Dies entspricht in der Praxis ca. 50.000 Treffern.

[7] Der GBV hat 9 Datensätze gefunden. Aber nur die ersten 8 konnten gelesen werden. Der 9. Datensatz ist im MAB2-Vollformat nicht verfügbar und der Z39.50-Server liefert die Fehlermeldung "[238] Record not available in requested syntax".

[8] Der SW erwartet offensichtlich ein Leerzeichen nach dem Komma

[9] TUBS, DDB, GBV und SW liefern null Treffer, wenn das Komma zwischen Nachname und Vorname fehlt.

[10] Der KOBV kann keine Rechtstrunkierung von Vornamen. Die Autoren sind in einem Wortindex abgelegt und die Suche nach "*autor=Dalitz, W*" wird in "*autor=Dalitz UND autor=W*" zerlegt. Die Suche nach dem Autor "W" liefert zuviele Treffer und die Datenbank liefert einen Fehler bzw. keine Treffer zurück.

[11] Der Server des SW war nicht ansprechbar.

[12] Die FH Potsdam findet die Datensätze auch, wenn man *Vorname* und *Nachname* vertauscht.

UND-Verknüpfungen sind bei zu vielen Treffern leider auch nicht möglich. Die Suche nach einem Autor "*Schneider*", der ein Buch mit dem Titel "*Katze*" geschrieben hat, liefert bei der DDB ebenfalls null Treffer.

Datenbank	Maximale Treffer	Ausgabe Trefferzahl	Antwortzeit in Sekunden
DDB	9999	0	5
BVB	ca. 400.000	exakt oder Fehlermeldung [13]	1-2
GBV	9999	9999 oder Fehlermeldung	10-15
KOBV	ca. 20.000 bis 50.000	exakt oder Fehlermeldung	4-15
TUBS	ca. 5.000 bis 10.000	exakt oder Fehlermeldung	5-30

Tabelle 9.2: Antwortverhalten bei großen Treffermengen

Legende Antwortverhalten

Maximale Treffer: Empirisch ermittelte maximale Treffer des Bibliothekssystem bei der Suche mit dem Attribut *Autor*.

Ausgabe Trefferzahl: Antwort des Bibliothekssystems bei vielen Treffern.

Antwortzeit in Sekunden: Gibt die Zeit an, bis das Bibliothekssystem bei vielen Treffern die Ergebnisse zurückliefert bzw. mit einer Fehlermeldung abbricht.

9.3.2 Spezialfälle

Verwendeter Zeichensatz beim Datenaustausch

Einige Server liefern Datensätze in einem Zeichensatz zurück, der nicht im Standard definiert ist - z.B. das System *allegro* verwendete anfangs ANSEL ([ANS99a]) für MAB2-Datensätze. Dieser Fehler ist bei allegro inzwischen behoben.

ALEPH 500 verwendet den im Internet gebräuchlichen ISO8859-1 (latin1) Zeichensatz für die westeuropäischen Sprachen. Im MAB2-Standard ist ISO8859-1 nicht als gültiger Zeichensatz erwähnt. Dieser Fehler ist harmlos, da die meisten Web-Gateways intern die Zeichensätze nach ISO8859-1 umwandeln und alle Zeichen ignorieren, die nicht in ISO8859-1 enthalten sind (z.B. polnisches Ł, Haček).

Unerwartetes Austauschformat

Das System SISIS liefert keine Datensätze im Austauschformat USMARC. Stattdessen wird das UNIMARC-Austauschformat verwendet. Üblicherweise geben die Z39.50-Server eine Fehlermeldung an den Client aus, wenn ein Format nicht unterstützt wird.

Beispiel: Der Bayerische Bibliotheksverbund (BVB) unterstützt kein USMARC und gibt die folgende Fehlermeldung aus.

[13] Entweder liefert der Server die genaue Anzahl der gefundenen Treffer - z.B. 5820 Treffer - oder aber er bricht mit einer Fehlermeldung ab, z.B. "Resources exhausted - no results available" oder " [109] Database unavailable – Time-out, Value: 30 sec"


```
Z> format usmarc
Z> show
Sent presentRequest (2+1).
Received presentResponse.
Diagnostic message(s) from database:
    [239] Record syntax not supported -- '1,2,840,10003,5,10'
```

Der Z39.50 Standard läßt beide Varianten zu - die Anfrage entweder mit einer Fehlermeldung abzubrechen oder ein *ähnliches* Format zurückzuliefern. Die meisten Systemanbieter in Deutschland haben sich für die Fehlermeldung entschieden.

Software noch in der Entwicklung

Der Z39.50-Server von ALEPH 500 ist nicht in der Lage, MAB2-Daten korrekt zu übermitteln. Die Angabe der Größe der Datensätze war falsch. Er ist derzeit für eine Nutzung im produktiven Betrieb noch nicht [14] geeignet. Die Firma ExL arbeitet an einer Verbesserung.

Begrenzte Anzahl der Indexe

Allegro unterstützt nur 11 Indexe. Einige davon (2 oder 3?) werden bereits intern vom System belegt. Mit Autor, Titel, ISBN, Verlag, Jahr und Schlagwörtern sind schon 6 Indexe vergeben. Wünsche nach weiteren Indexen sind kaum zu realisieren.

Norm- und Titeldaten

Die meisten bibliograpischen Datenbanken liefern nur die Titeldaten (Informationen über ein Buch) zurück. Die Deutsche Bibliothek (DDB) liefert in ihrem System ILTIS allerdings auch die Normdaten (Informationen über einen Autor oder Institution) zurück. Beispiel:

Bei der Suche nach dem Autor "*Dalitz*" liefert die DDB 184 Treffer. Zuerst werden die Personendaten (PND) ausgegeben, danach die einzelnen Titel.

```
Dalitz, Helmut
Dalitz, Helmut a1947-
Dalitz, Michael
Dalitz, Wolfgang
Dalitz, Gerhard
Dalitz, Birgit
Dalitz, Elisabeth
[...]
Dalitz, Wolfgang
Hyper-G: das Internet-Informationssystem der 2. Generation
Heidelberg: dpunkt, Verl. für digitale Technologie, 1995.
```

Der Bayerische Bibliotheksverbund findet bei der selben Anfrage 77 Treffer und liefert nur die Titeldaten:

```
Dalitz, Wolfgang
Hyper-G: das Internet-Informationssystem der 2. Generation
Heidelberg: dpunkt, Verl. für digitale Technologie, 1995.
[...]
```

[14] Stand März 1999

Der Benutzer kann mit den Personendaten nicht viel anfangen. Er will bei der Suche nach einem Buch, geschrieben vom Autor "*Dalitz*", nicht jedesmal wissen, wieviele Autoren es mit dem Namen "*Dalitz*" gibt.

Aus diesem Grund sollten die Normdaten und die Titeldaten nicht zusammen in einer Datenbank stehen, sondern in zwei getrennten Datenbanken verfügbar sein (zum Thema Normdaten siehe auch [Kub97]).

Instabiler Server des Gemeinsamen Bibliotheksverbundes (GBV)

Der Z39.50-Server des GBV stürzt aus unerklärlichen Gründen regelmäßig ab. Weiterhin können manche Datensätze nicht im gewünschten Datenformat geliefert werden.

Beispiel: Gesucht wird im GBV nach dem Autor "luegger,j" mit Rechtstrunkierung (d.h. "luegger,joachim" sollte auch gefunden werden). Es wurden 18 Datensätze bei der Suche gefunden. Die Ausgabe erfolgt im MAB2-Kurzformat (brief). Es wird der 16. Treffer im MAB2-Kurzformat ausgegeben:

```
Z> find @attr 5=1 @attr 1=1 luegger,j
Search was a success.
Number of hits: 18, setno 6
Z> elements b
Z> format mab
Z> show 16+1
Sent presentRequest (16+1).
Received presentResponse.
Records: 1
databaseName: GVK
Record type: MAB ResultSetPosition 16
### 00269nM2.01000024      h
001 01.845217.5
003 19980625
030 e5zz0017
050 aa
051 m
070aGBV
100 Lügger, Joachim
104aDalitz, Wolfgang
331aVerteilung mathematischer Software mittels elektronischer Netze - \
        die elektronische Softwarebibliothek eLib
410 Berlin
412 ZIB
425a1991
Z>
```

Jetzt wird der 16. Datensatz nochmal angefordert, diesmal als vollständiger (full) MAB2-Datensatz. Die Anfrage scheitert. Die Ursache ist unklar. Die Datensätze 1-15 und 17 und 18 können im MAB-Vollformat gelesen werden.

```
Z> elements F
Z> show 16+1
Sent presentRequest (16+1).
Received presentResponse.
Records: 1
databaseName: GVK
Diagnostic message(s) from database:
    [238] Record not available in requested syntax
```

Ein weiteres Problem ist, daß der GBV manchmal den ersten Buchstaben eines Feldes nicht ausgibt. Aus dem "*Konrad-Zuse-Zentrum*" wird so ein "*onrad-Zuse-Zentrum*". Die Dublettenkontrolle wird durch die fehlenden Buchstaben wesentlich komplizierter, wenn nicht gar unmöglich.

Weitere Informationen zur Evaluation von Z39.50-Servern sind in [Rus99b], [Got96] und [BAT99] zu finden.

9.4 Zusammenfassung

Das Z39.50-Protokoll gibt die Syntax für den Dialog zwischen Client und Server vor. Zwischen den Datenbanken gibt es in der Praxis semantische Differenzen, die bei der verteilten Suche berücksichtigt werden müssen.

Der Wartungsaufwand für die Nutzung der Z39.50-Server ist wesentlich höher als zunächst angenommen. Es muß regelmäßig geprüft werden, ob die Z39.50-Server korrekte Daten zurückliefern. Diese Prüfung kann teilweise automatisch mit Scripten durchgeführt werden. Die manuelle Kontrolle kann sie nicht ersetzen. Die Dokumentation der meisten Z39.50-Server ist mangelhaft. Die Zuverlässigkeit der Z39.50-Server läßt zu wünschen übrig - an manchen Tagen beispielsweise ist eine stabile Verbindung zum Gemeinsamen Bibliotheksverbund nicht möglich.

Für die produktive Nutzung von Z39.50-Servern verbleiben immer die Unsicherheitsfaktoren:

- Bibliotheken benutzen unterschiedliche Bibliothekssysteme von unterschiedlichen Herstellern. Jeder dieser Hersteller hat seine eigene Interpretation von MAB2 und den Attributen für die Suche.

- Die Anbieter von Software bringen regelmäßig neue Versionen ihrer Produkte heraus. Bei jeder neuen Version kann sich das zurückgelieferte MAB2-Format oder die Attribute für die Suche ändern.

- Jede Bibliothek oder jeder Bibliotheksverbund hat ihre eigene Interpretation, wie MAB2 im Kurzformat aussieht. Bei vielen Systemen kann der Administrator einstellen, was der Benutzer zurückgeliefert bekommt. D.h. selbst wenn zwei Bibliotheken dieselbe Software verwenden, kann sich das zurückgelieferte MAB2 Format unterscheiden.

- Für die Suche in der Datenbank wird ein Index angelegt. Für unterschiedliche Attribute werden unterschiedliche Indexe angelegt, z.B. ein Index für Autor, ein Index für Titel und ein Index für ISBN-Nummern. Außerdem wird festgelegt, ob in dem Index Wörter oder Wortgruppen verwendet werden. Es ist selten dokumentiert, wie der Index aufgebaut wurde. Die Indexe der Bibliotheken sind nicht unbedingt miteinander vergleichbar. Zum Beispiel fügt die Technische Universität Braunschweig die Schlagwörter dem Titelindex hinzu, andere Anbieter tun dies nicht. Der Benutzer stellt die gleiche Anfrage, und jede Datenbank antwortet leicht verschieden (gefunden in Titel, gefunden in Titel und Schlagwörtern).

Trotz all dieser Probleme lief *ZACK* über mehrere Monate stabil. Einige der aufgeführten Probleme - insbesondere die technischen - sind in Zusammenarbeit mit den Anbietern der Z39.50-Server zu lösen, andere nicht. Man muß damit leben, daß trotz detaillierter und umfassend definierter Standards (MAB2, RAK, Z39.50) viele Dinge im alltäglichen Betrieb nicht so funktionieren wie erwartet.

Kapitel 10

Ausblick

ZACK hat die in das System gesetzten Erwartungen erfüllt. Der Benutzer kann in einer oder mehreren bibliographischen Datenbanken nach einem Dokument suchen und die Treffer in die eigene lokale Datenbank übernehmen. Die verteilte Suche hat in der Praxis eine deutlich bessere Trefferquote gebracht als die Suche in nur einer Datenbank. Dabei bleibt die Antwortzeit in einem für die Benutzer akzeptablen Rahmen. Je nach Anzahl der Datenbanken (3-6) und der gefundenen Datensätze (10-150) erhält der Benutzer innerhalb von 5 bis 8 Sekunden das Ergebnis. Die Kurztrefferliste wird durch die Dublettenkontrolle bis zur Hälfte kürzer und dadurch übersichtlicher.

Mit *ZACK* wurden viele Erfahrungen für die praktische Nutzung von Z39.50, die Normierung und die Dublettenkontrolle gesammelt. In Zukunft (1999/2000) wird eine Teilfunktionalität von *ZACK* durch den Kooperativen Bibliotheksverbund Berlin-Brandenburg (KOBV) angeboten.

Das System *ZACK* wird seit mehreren Monaten von Brandenburger Bibliothekaren für die Erfassung von Büchern produktiv genutzt. Für sie konnte damit die Zeit bis zur Einführung der neuen Verbund-Software im KOBV überbrückt werden.

Es gibt noch viele Ideen, wie man *ZACK* verbessern kann. Dazu gehören:

Weitere Formate: *ZACK* unterstützt bei der Dublettenkontrolle nur das deutsche Austauschformat MAB. In der englischsprachigen Welt ist das Format USMARC vorherrschend. Es wäre wünschenswert, wenn auch USMARC bei der Dublettenkontrolle verwendet werden könnte. Langfristig ist das Ziel, unterschiedliche Formate (USMARC, MAB2, Dublin Core etc.) gleichzeitig bei der Dublettenkontrolle verwenden zu können. Beispielsweise wird man die Datensätze von der Library of Congress im Format USMARC holen, von der Deutschen Bibliothek im Format MAB2 und dann die Dublettenkontrolle starten.

Zeichensatz: *ZACK* verwendet intern den für die westeuropäischen Sprachen gebräuchlichen Zeichensatz ISO8859-1 (latin1). Bücher in osteuropäischen Sprachen (polnisch, tschechisch, russisch) werden deshalb bei der Normierung, Dublettenkontrolle und Ausgabe benachteiligt, weil ihre Zeichen nicht korrekt dargestellt werden.

Verteilte Registersuche: Die verteilte Registersuche (Scan) konnte aus Zeitmangel nicht mehr implementiert werden. Wünschenswert wäre es, die verteilte Registersuche in der nächsten Version von *ZACK* anzubieten.

Clusterbildung: Bei der verteilten Suche mit dem Attribut Autor verbraucht die Dublettenkontrolle relativ viel Rechenzeit. Die Algorithmen müssen für diesen Spezialfall besser optimiert werden.

Kurztrefferliste: Das MAB2-Format ist sehr umfangreich, in der Deutschen Bibliothek werden bis zu 244 verschiedene Felder genutzt. Für die Kurztrefferliste und die textuelle Darstellung der Datensätze werden nur wenige Felder genutzt (Autor, Titel, Jahr). In einigen Fällen sind diese Felder nicht belegt, und die Information steht in anderen Feldern. Eine Ausgabe sieht dann unvollständig aus. Wünschenswert wäre, die Kurztrefferliste für diese Fälle besser zu parametrisieren.

Testen der Z39.50-Server: Ein umfassender manueller Test eines Z39.50-Servers dauert mehrere Tage ([Rus99b]), ein einfacher Test einen halben Tag. Dies ist sehr arbeitsaufwendig, insbesondere, wenn man viele Server zu prüfen hat. Es sollte ein Programm entwickelt werden, das die Tests automatisch oder halbautomatisch durchführt. Weiterhin könnte man mit diesem Programm dann auch die Z39.50-Server regelmäßig überwachen und Änderungen (Konfiguration, neue Software) automatisch feststellen.

Anhang A

Analyse der MAB2-Datensätze der Deutschen Bibliothek

In diesem Anhang werden 2,5 Millionen Datensätze der Deutschen Bibliothek statistisch ausgewertet. Ziel ist es festzustellen, welche Felder in den Datensätzen wirklich genutzt werden, wie das Verhältnis zwischen Büchern aus Verlagen und sonstiger Literatur ist, welche Beziehungen (Hierarchien, Verweise) zwischen den Datensätzen existieren.

Die Analyse war erforderlich, da es keine statistischen Informationen zu den MAB2-Datensätzen der Deutschen Bibliothek gab.

Die Verteilung und Nutzung der MAB2-Felder ist wichtig für:

- Normierung: welche Felder gibt es überhaupt, welche muß man genauer überprüfen.
- Vergleich von Datensätzen (Dublettenkontrolle).
- Verknüpfung von Datensätzen: wie ist der hierarchische Aufbau bzw. die Verlinkung mit anderen Datensätzen, z.B. den Normdaten.
- Beurteilung der Qualität der Datensätze. Zum Beispiel haben Datensätze mit Schlagwörtern eine höhere Qualität als Datensätze ohne Schlagwörter.
- Aufbau von Indexen zur Suche in der Datenbank: zum Beispiel welche Felder man für den Index *Autor* nutzt sowie die Abschätzung der Größe des Indexes.

Im Rahmen des KOBV-Projektes ([KOB99]) wurden von der Deutschen Bibliothek (DDB) 2,5 Millionen Datensätze der Deutschen Nationalbibliographie (DNB) erworben. Für die Statistik werden Datensätze der Deutschen Bibliothek aus den Jahren 1986 bis 1998 sowie die Nachlieferungen bis Februar 1999 verwendet. Die Datensätze wurden von der DDB im Format MAB2 auf Magnetbändern geliefert. Die Analyse der Daten erfolgt in zwei Schritten. Zuerst werden die Daten aus dem MAB2-Bandformat in das MAB2-Diskettenformat umgewandelt. Danach werden die Datensätze mit dem Perl-Script mab2stat (siehe Anhang C Kurzbeschreibung der Software, Seite 128) analysiert. Die Berechnung der Statistik für alle Datensätze dauerte auf dem Rechner se2 [1] ca. 2 Stunden.

A.1 Aufschlüsselung nach Satztyp

Im MAB-Format wird jedem Datensatz ein Satztyp zugeordnet. Der Satztyp kennzeichnet den bibliographischen Sachverhalt und/oder die Funktion und Rangordnung des Satzes. Im

[1] Eine UltraSPARC-II mit 336 MHz, siehe Abkürzungsverzeichnis

DDB-MAB2-Format unterscheidet man zwischen Hauptsatz (h) und Untersatz (u oder y) (aus MAB2-Datendienst, [DNB96]).

Die Untersätze (u oder y) und die zugehörigen Hauptsätze (h) bilden eine hierarchische Struktur (siehe Abbildung A.1). Auf der obersten Stufe (h-Satz) stehen die Informationen, die allen Bänden gemeinsam sind. Auf der unteren Stufe (u-Satz) stehen die spezifischen Angaben zu einem Band. Erst die Informationen aus beiden Hierarchiestufen ergeben die vollständige Information für den Band.

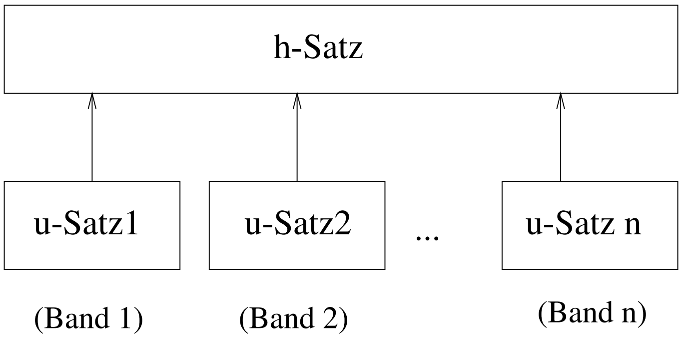


Abbildung A.1: Mehrbändige begrenzte Werke mit Bandaufführung, h- und u-Sätze

Ein Hauptsatz kann einen oder mehrere Unterdatensätze besitzen. Die untergeordneten Sätze (u-Satz) sind mit dem direkt übergeordneten Datensatz verknüpft. Im Feld 010 der u-Sätze steht die ID des h-Satzes (Feld 001 im h-Satz). Eine Verknüpfung in der Gegenrichtung - vom h-Satz zu den u-Sätzen - gibt es nicht (siehe auch [Kub99b]).

Bei der Dublettenkontrolle kann man nur Datensätze vom gleichen Typ miteinander vergleichen. Für die Ausgabe von u-Sätzen benötigt man auch den zugehörigen übergeordneten h-Satz, da z.B. der Gesamttitel nur im übergeordneten h-Satz steht.

Verteilung der Satztypen in der DNB

Diese Tabelle gibt die Verteilung der Satztypen h (Hauptsatz) und Untersatz (u, y) in allen untersuchten Datensätzen an.

Typ	Anzahl	Prozent
h	2.115.316	83,44%
u	411.759	16,24%
y	8.020	0,32%
Summe	2535095	100%

Tabelle A.1: DNB: Verteilung der Satztypen

Die 411.759 u-Sätze verweisen auf 178.864 verschiedene h-Sätze (bzw. y-Sätze). Dies entspricht ca. 2,3 u-Sätze pro h-Satz. 1/4 aller Datensätze haben eine hierarchische Struktur (u-Sätze plus zugehöriger h-Satz). Die restlichen 3/4 sind einbändige Werke (h-Sätze) und ohne Hierarchiestufe.

A.2 Untersuchte Reihen

Beschreibung der Reihen

Die Deutsche Nationalbibliographie umfaßt im einzelnen die folgenden Reihen:

- Reihe A: Monographien und Periodika des Verlagsbuchhandels
  Erscheinungsweise: wöchentlich
- Reihe B: Monographien und Periodika außerhalb des Verlagsbuchhandels
  Erscheinungsweise: wöchentlich
- Reihe C: Karten
  Erscheinungsweise: wöchentlich
- Reihe G: Fremdsprachige Germanica und Übersetzungen deutschsprachiger Werke
  Erscheinungsweise: vierteljährlich
- Reihe H: Hochschulschriften
  Erscheinungsweise: monatlich
- Reihe M: Musikalien
  Erscheinungsweise: monatlich
- Reihe N: Vorankündigungen Monographien und Periodika (CIP)
  Erscheinungsweise: wöchentlich
- Reihe T: Musiktonträger
  Erscheinungsweise: monatlich

(aus MAB2-Datendienst [DNB96])

Verteilung nach Reihe

Die Datensätze werden auf Magnetbändern geliefert und gliedern sich in Reihen. Die Reihen C (Karten) und G (Fremdsprachige Germanica und Übersetzungen deutschsprachiger Werke) wurden vom KOBV-Projekt nicht erworben und fehlen deshalb in dieser Statistik.

Reihe	Anzahl	in Prozent
A = Bücher, die in Verlagen erschienen	1.261.457	49,76%
B = Bücher, die nicht in Verlagen erschienen	619.883	24,45%
H = Hochschulschriften	257.239	10,15%
M = Musikalien	95.576	3,77%
N = CIP Titel	8.467	0,33%
T = Tonträger	292.473	11,54%
Summe	2.535.095	100%

Tabelle A.2: DNB: Verteilung nach Reihe

Knapp die Hälfte der Datensätze beschreiben Bücher, die in Verlagen erschienen sind. Ein Viertel der Datensätze sind Bücher, die nicht in Verlagen erschienen sind. Das Verhältnis *Bücher in Verlagen erschienen* zu *Bücher nicht in Verlagen erschienen* beträgt zwei zu eins. 3/4 aller Datensätze beschreiben Bücher. Jeder 10. Datensatz ist eine Hochschulschrift.

DNB Bandlieferungen der Jahre 1986-1998

Für die statistische Auswertung wurden die Datensätze aus den Jahren 1986 bis 1998 verwendet.

- Reihe A = Bücher, in Verlagen erschienen:

  A8651, A8751, A8851, A8951, A9051, A9151, A9251, A9351, A9451, A9551, A9651, A9751, a9851 [2]

- Reihe B = Bücher, nicht in Verlagen erschienen:

  B8651, B8751, B8851, B8951, B9051, B9151, B9251, B9351, B9451, B9551, B9651, B9751, b9851,

- Reihe H = Hochschulschriften:

  H8612, H8712, H8812, H8912, H9012, H9112, H9212, H9312, H9412, H9512, H9612, H9712, h9812

- Reihe M = Musikalien:

  M8612, M8712, M8812, M8912, M9012, M9112, M9212, M9312, M9412, M9512, M9612, M9712, m9812

- Reihe T = Tonträger:

  t8612, t8712, t8812, t8912, t9012, t9112, t9212, t9312, t9412, t9512, t9612, t9712, t9812

Die Dateinamen haben zusätzlich die Endung `mab2band`. D.h. die Datensätze für die Reihe A, Jahrgang 86, Wochenlieferung 1-51 befinden sich in der Datei `A8651mab2band`.

Es gibt insgesamt 2.493.334 Datensätze in den Bandlieferungen, dies entspricht 98,35% aller Datensätze. Die Datensätze liegen im Zeichensatz ISO 5426 vor. Sie sind zusammen 1355MB groß (MAB2 Diskettenformat).

Nachlieferungen 1999

Für die statistische Auswertung wurden außerdem die folgenden Nachlieferungen aus dem Jahr 1999 verwendet.

- Reihe A = Bücher, in Verlagen erschienen:

  A9901, A9902, A9903, A9904, A9905, A9906, A9907, A9908, A9909

- Reihe B = Bücher, nicht in Verlagen erschienen:

  B9901, B9902, B9903, B9904, B9905, B9906, B9907, B9908, B9909

- Reihe H = Hochschulschriften:

  H9901, H9902

- Reihe M = Musikalien: M9901, M9902

- Reihe N = CIP (Cataloguing in publication) Titel

  N9901, N9902, N9903, N9904, N9905, N9906, N9907 N9908, N9909

- Reihe T = Tonträger:

  T9901, T9902

[2] Die Groß- und Kleinschreibung der Dateinamen hat keinerlei Bedeutung für den Inhalt der Lieferung. Sie ist eine Konvention der Deutschen Bibliothek - Großschreibung sowie der Name der Reihe. In einigen wenigen Fällen wurde von dieser Konvention abgewichen. a9851 müßte eigentlich A9851 heißen.

Die Dateinamen haben zusätzlich die Endung `ti2.dat`. D.h. die Datensätze für die Reihe A, Jahrgang 99, Wochenlieferung 06 befinden sich in der Datei `A9906ti2.dat`.

Die Nachlieferungen enthalten insgesamt 41.761 Datensätze, das enspricht 1,65% aller Datensätze. Die Datensätze liegen im Zeichensatz ISO 5426 vor. Sie sind zusammen 24,4MB groß (MAB2 Diskettenformat).

A.3 Verteilung der MAB2-Felder

Die Tabelle A.3 auf den Seiten 107 bis 112 gibt einen Überblick über die von der Deutschen Bibliothek genutzten MAB2-Felder. Die Statistik der MAB2-Feldnummern wurde für alle Satztypen (h, u, y) zusammen durchgeführt und nochmal einzeln für die Satztypen h und u. Aus Platzgründen wurde die Tabelle nach jeweils 40 Einträgen umgebrochen. Insgesamt hat die Tabelle 244 Einträge (entspricht 7 Teil-Tabellen).

Es wurden 2.535.095 bibliographische MAB2-Datensätze ausgewertet. Diese enthielten 647.909.320 Datenfelder, im Durchschnitt sind das 25,5 Felder je Satz.

Die Anzahl der y-Datensätze ist mit 8.020 (0,32%) verschwindend gering. Die dreistufige hierarchische Struktur mit den y-Sätzen kommt in der DDB sehr selten vor, zukünftig soll sie ganz entfallen. Auf die Auswertung der y-Datensätze wird deshalb hier verzichtet.

Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
1.	001	2.535.095	100,00%	2.115.316	100,00%	411.759	100,00%
2.	004	2.535.095	100,00%	2.115.316	100,00%	411.759	100,00%
3.	030	2.535.095	100,00%	2.115.316	100,00%	411.759	100,00%
4.	050	2.535.095	100,00%	2.115.316	100,00%	411.759	100,00%
5.	070	2.535.095	100,00%	2.115.316	100,00%	411.759	100,00%
6.	331	2.191.768	86,46%	2.030.204	95,98%	157.009	38,13%
7.	425	2.107.365	83,13%	1.740.419	82,28%	366.946	89,12%
8.	574	2.075.669	81,88%	1.663.563	78,64%	411.759	100,00%
9.	544	2.042.552	80,57%	1.641.608	77,61%	400.944	97,37%
10.	700	2.028.496	80,02%	1.981.928	93,69%	39.033	9,48%
11.	036	2.024.807	79,87%	2.024.807	95,72%	0	0,00%
12.	433	1.961.648	77,38%	1.660.199	78,48%	301.354	73,19%
13.	051	1.927.157	76,02%	1.876.781	88,72%	42.368	10,29%
14.	435	1.852.422	73,07%	1.781.405	84,21%	70.974	17,24%
15.	037	1.779.019	70,18%	1.402.673	66,31%	376.346	91,40%
16.	410	1.771.550	69,88%	1.770.682	83,71%	868	0,21%
17.	412	1.771.550	69,88%	1.770.682	83,71%	868	0,21%
18.	100	1.648.452	65,03%	1.600.219	75,65%	47.862	11,62%
19.	102	1.648.451	65,03%	1.600.220	75,65%	47.860	11,62%
20.	359	1.540.298	60,76%	1.493.060	70,58%	46.517	11,30%
21.	540	1.413.387	55,75%	1.165.001	55,07%	248.291	60,30%
22.	335	994.296	39,22%	963.602	45,55%	30.474	7,40%
23.	434	946.459	37,33%	849.594	40,16%	96.865	23,52%
24.	501	847.725	33,44%	754.307	35,66%	93.060	22,60%
25.	451	768.801	30,33%	717.606	33,92%	51.178	12,43%
26.	902	760.590	30,00%	755.633	35,72%	305	0,07%
27.	403	612.401	24,16%	502.813	23,77%	109.573	26,61%
28.	400	566.406	22,34%	459.329	21,71%	107.062	26,00%
29.	454	554.632	21,88%	508.361	24,03%	46.270	11,24%
30.	453	554.608	21,88%	508.339	24,03%	46.268	11,24%
31.	456	551.734	21,76%	505.464	23,90%	46.269	11,24%
32.	455	549.137	21,66%	502.867	23,77%	46.269	11,24%
33.	010	419.779	16,56%	0	0,00%	411.759	100,00%
34.	090	419.779	16,56%	0	0,00%	411.759	100,00%
35.	568	404.167	15,94%	354.022	16,74%	50.145	12,18%
36.	089	386.578	15,25%	0	0,00%	382.023	92,78%
37.	903	384.635	15,17%	383.479	18,13%	157	0,04%
38.	029	372.520	14,69%	337.719	15,97%	34.454	8,37%
39.	200	365.788	14,43%	350.542	16,57%	15.223	3,70%
40.	202	365.625	14,42%	350.379	16,56%	15.223	3,70%

Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
41.	052	356.374	14,06%	238.535	11,28%	117.839	28,62%
42.	519	345.854	13,64%	345.758	16,35%	96	0,02%
43.	104	314.558	12,41%	303.162	14,33%	11.238	2,73%
44.	106	314.558	12,41%	303.162	14,33%	11.238	2,73%
45.	551	279.807	11,04%	243.881	11,53%	35.924	8,72%
46.	076	255.735	10,09%	219.431	10,37%	36.304	8,82%
47.	304	232.529	9,17%	232.529	10,99%	0	0,00%
48.	437	185.801	7,33%	156.404	7,39%	29.397	7,14%
49.	370	184.260	7,27%	167.874	7,94%	16.344	3,97%
50.	907	157.455	6,21%	156.664	7,41%	60	0,01%
51.	517	148.544	5,86%	125.984	5,96%	22.560	5,48%
52.	599	90.509	3,57%	90.509	4,28%	0	0,00%
53.	908	88.930	3,51%	88.632	4,19%	31	0,01%
54.	108	87.973	3,47%	81.320	3,84%	6.641	1,61%
55.	110	87.973	3,47%	81.320	3,84%	6.641	1,61%
56.	542	87.232	3,44%	87.052	4,12%	159	0,04%
57.	415	83.127	3,28%	82.942	3,92%	185	0,04%
58.	417	83.127	3,28%	82.942	3,92%	185	0,04%
59.	805	79.095	3,12%	78.516	3,71%	490	0,12%
60.	361	64.081	2,53%	63.064	2,98%	1.017	0,25%
61.	531	59.006	2,33%	58.981	2,79%	21	0,01%
62.	204	56.368	2,22%	47.907	2,26%	8.457	2,05%
63.	206	56.353	2,22%	47.892	2,26%	8.457	2,05%
64.	310	41.297	1,63%	41.297	1,95%	0	0,00%
65.	333	38.609	1,52%	38.609	1,83%	0	0,00%
66.	038	37.712	1,49%	32.465	1,53%	5.247	1,27%
67.	057	37.526	1,48%	36.378	1,72%	1.148	0,28%
68.	533	33.928	1,34%	33.900	1,60%	14	0,00%
69.	341	33.057	1,30%	32.348	1,53%	680	0,17%
70.	360	32.078	1,27%	32.078	1,52%	0	0,00%
71.	912	31.787	1,25%	31.615	1,49%	16	0,00%
72.	112	31.528	1,24%	27.179	1,28%	4.349	1,06%
73.	114	31.527	1,24%	27.178	1,28%	4.349	1,06%
74.	461	29.504	1,16%	28.591	1,35%	913	0,22%
75.	556	22.514	0,89%	20.787	0,98%	1.727	0,42%
76.	510	20.891	0,82%	15.963	0,75%	4.928	1,20%
77.	116	19.308	0,76%	16.689	0,79%	2.619	0,64%
78.	118	19.308	0,76%	16.689	0,79%	2.619	0,64%
79.	913	18.198	0,72%	18.130	0,86%	8	0,00%
80.	208	15.850	0,63%	9.880	0,47%	5.970	1,45%

Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
81.	210	15.846	0,63%	9.876	0,47%	5.970	1,45%
82.	710	14.542	0,57%	14.063	0,66%	31	0,01%
83.	811	14.297	0,56%	14.202	0,67%	89	0,02%
84.	300	12.837	0,51%	12.837	0,61%	0	0,00%
85.	120	12.098	0,48%	10.562	0,50%	1.536	0,37%
86.	122	12.098	0,48%	10.562	0,50%	1.536	0,37%
87.	464	11.285	0,45%	10.676	0,50%	609	0,15%
88.	466	11.285	0,45%	10.676	0,50%	609	0,15%
89.	463	11.283	0,45%	10.674	0,50%	609	0,15%
90.	465	11.282	0,45%	10.673	0,50%	609	0,15%
91.	800	10.120	0,40%	9.948	0,47%	172	0,04%
92.	802	10.090	0,40%	10.014	0,47%	76	0,02%
93.	532	9.713	0,38%	9.679	0,46%	1	0,00%
94.	917	9.510	0,38%	9.412	0,44%	5	0,00%
95.	527	9.072	0,36%	9.070	0,43%	2	0,00%
96.	418	8.597	0,34%	8.597	0,41%	0	0,00%
97.	505	8.452	0,33%	8.420	0,40%	32	0,01%
98.	124	8.194	0,32%	7.200	0,34%	994	0,24%
99.	126	8.194	0,32%	7.200	0,34%	994	0,24%
100.	536	7.673	0,30%	6.812	0,32%	861	0,21%
101.	518	7.560	0,30%	7.141	0,34%	419	0,10%
102.	212	7.558	0,30%	3.979	0,19%	3.579	0,87%
103.	214	7.558	0,30%	3.979	0,19%	3.579	0,87%
104.	541	6.822	0,27%	5.752	0,27%	1.070	0,26%
105.	128	5.721	0,23%	5.115	0,24%	606	0,15%
106.	130	5.721	0,23%	5.115	0,24%	606	0,15%
107.	918	5.482	0,22%	5.432	0,26%	1	0,00%
108.	530	4.664	0,18%	4.662	0,22%	2	0,00%
109.	132	4.230	0,17%	3.832	0,18%	398	0,10%
110.	134	4.230	0,17%	3.832	0,18%	398	0,10%
111.	817	3.809	0,15%	3.781	0,18%	28	0,01%
112.	334	3.273	0,13%	3.273	0,15%	0	0,00%
113.	216	3.226	0,13%	1.702	0,08%	1.524	0,37%
114.	218	3.226	0,13%	1.702	0,08%	1.524	0,37%
115.	529	3.128	0,12%	3.126	0,15%	2	0,00%
116.	136	3.028	0,12%	2.754	0,13%	274	0,07%
117.	138	3.028	0,12%	2.754	0,13%	274	0,07%
118.	808	2.811	0,11%	2.796	0,13%	15	0,00%
119.	512	2.462	0,10%	2.334	0,11%	128	0,03%
120.	806	2.346	0,09%	2.305	0,11%	41	0,01%

Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
121.	653	2.289	0,09%	937	0,04%	1.351	0,33%
122.	140	2.107	0,08%	1.920	0,09%	187	0,05%
123.	142	2.107	0,08%	1.920	0,09%	187	0,05%
124.	652	2.069	0,08%	628	0,03%	1.441	0,35%
125.	922	1.973	0,08%	1.952	0,09%	1	0,00%
126.	504	1.613	0,06%	1.608	0,08%	0	0,00%
127.	823	1.563	0,06%	1.555	0,07%	8	0,00%
128.	637	1.474	0,06%	1.440	0,07%	34	0,01%
129.	144	1.402	0,06%	1.272	0,06%	130	0,03%
130.	146	1.402	0,06%	1.272	0,06%	130	0,03%
131.	644	1.396	0,06%	1.359	0,06%	37	0,01%
132.	220	1.372	0,05%	776	0,04%	596	0,14%
133.	222	1.372	0,05%	776	0,04%	596	0,14%
134.	654	1.323	0,05%	465	0,02%	858	0,21%
135.	619	1.213	0,05%	1.176	0,06%	37	0,01%
136.	923	1.019	0,04%	1.017	0,05%	1	0,00%
137.	345	987	0,04%	984	0,05%	3	0,00%
138.	814	986	0,04%	975	0,05%	11	0,00%
139.	148	932	0,04%	843	0,04%	89	0,02%
140.	150	932	0,04%	843	0,04%	89	0,02%
141.	342	822	0,03%	822	0,04%	0	0,00%
142.	471	810	0,03%	789	0,04%	21	0,01%
143.	829	794	0,03%	792	0,04%	2	0,00%
144.	224	734	0,03%	415	0,02%	319	0,08%
145.	226	734	0,03%	415	0,02%	319	0,08%
146.	503	707	0,03%	645	0,03%	62	0,02%
147.	152	642	0,03%	562	0,03%	80	0,02%
148.	154	642	0,03%	562	0,03%	80	0,02%
149.	804	569	0,02%	569	0,03%	0	0,00%
150.	812	541	0,02%	531	0,03%	10	0,00%
151.	820	465	0,02%	465	0,02%	0	0,00%
152.	156	435	0,02%	366	0,02%	69	0,02%
153.	158	435	0,02%	366	0,02%	69	0,02%
154.	228	432	0,02%	235	0,01%	197	0,05%
155.	230	432	0,02%	235	0,01%	197	0,05%
156.	511	403	0,02%	366	0,02%	37	0,01%
157.	502	398	0,02%	398	0,02%	0	0,00%
158.	343	371	0,01%	370	0,02%	1	0,00%
159.	160	296	0,01%	241	0,01%	55	0,01%
160.	162	296	0,01%	241	0,01%	55	0,01%

Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
161.	927	291	0,01%	291	0,01%	0	0,00%
162.	810	282	0,01%	282	0,01%	0	0,00%
163.	507	267	0,01%	262	0,01%	5	0,00%
164.	473	258	0,01%	244	0,01%	14	0,00%
165.	474	258	0,01%	244	0,01%	14	0,00%
166.	475	258	0,01%	244	0,01%	14	0,00%
167.	476	258	0,01%	244	0,01%	14	0,00%
168.	516	255	0,01%	231	0,01%	24	0,01%
169.	611	244	0,01%	214	0,01%	30	0,01%
170.	613	244	0,01%	214	0,01%	30	0,01%
171.	634	242	0,01%	209	0,01%	33	0,01%
172.	232	236	0,01%	127	0,01%	109	0,03%
173.	234	236	0,01%	127	0,01%	109	0,03%
174.	621	226	0,01%	196	0,01%	30	0,01%
175.	623	221	0,01%	191	0,01%	30	0,01%
176.	624	221	0,01%	191	0,01%	30	0,01%
177.	625	221	0,01%	191	0,01%	30	0,01%
178.	626	221	0,01%	191	0,01%	30	0,01%
179.	164	210	0,01%	158	0,01%	52	0,01%
180.	166	210	0,01%	158	0,01%	52	0,01%
181.	826	197	0,01%	197	0,01%	0	0,00%
182.	168	152	0,01%	104	0,00%	48	0,01%
183.	170	152	0,01%	104	0,00%	48	0,01%
184.	818	141	0,01%	137	0,01%	4	0,00%
185.	570	137	0,01%	22	0,00%	115	0,03%
186.	523	130	0,01%	130	0,01%	0	0,00%
187.	928	123	0,00%	123	0,01%	0	0,00%
188.	236	122	0,00%	57	0,00%	65	0,02%
189.	238	122	0,00%	57	0,00%	65	0,02%
190.	172	120	0,00%	76	0,00%	44	0,01%
191.	174	120	0,00%	76	0,00%	44	0,01%
192.	176	106	0,00%	66	0,00%	40	0,01%
193.	178	106	0,00%	66	0,00%	40	0,01%
194.	180	88	0,00%	52	0,00%	36	0,01%
195.	182	88	0,00%	52	0,00%	36	0,01%
196.	816	70	0,00%	70	0,00%	0	0,00%
197.	184	66	0,00%	38	0,00%	28	0,01%
198.	186	66	0,00%	38	0,00%	28	0,01%
199.	240	62	0,00%	29	0,00%	33	0,01%
200.	242	62	0,00%	29	0,00%	33	0,01%

Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
201.	655	61	0,00%	61	0,00%	0	0,00%
202.	188	56	0,00%	28	0,00%	28	0,01%
203.	190	56	0,00%	28	0,00%	28	0,01%
204.	932	54	0,00%	54	0,00%	0	0,00%
205.	346	49	0,00%	49	0,00%	0	0,00%
206.	192	46	0,00%	22	0,00%	24	0,01%
207.	194	46	0,00%	22	0,00%	24	0,01%
208.	937	46	0,00%	46	0,00%	0	0,00%
209.	822	43	0,00%	43	0,00%	0	0,00%
210.	824	41	0,00%	40	0,00%	1	0,00%
211.	196	39	0,00%	19	0,00%	20	0,00%
212.	198	39	0,00%	19	0,00%	20	0,00%
213.	244	31	0,00%	10	0,00%	21	0,01%
214.	246	31	0,00%	10	0,00%	21	0,01%
215.	828	22	0,00%	22	0,00%	0	0,00%
216.	248	20	0,00%	7	0,00%	13	0,00%
217.	250	20	0,00%	7	0,00%	13	0,00%
218.	933	20	0,00%	20	0,00%	0	0,00%
219.	938	15	0,00%	15	0,00%	0	0,00%
220.	942	12	0,00%	12	0,00%	0	0,00%
221.	252	9	0,00%	3	0,00%	6	0,00%
222.	254	9	0,00%	3	0,00%	6	0,00%
223.	347	8	0,00%	8	0,00%	0	0,00%
224.	947	8	0,00%	8	0,00%	0	0,00%
225.	407	7	0,00%	6	0,00%	1	0,00%
226.	943	7	0,00%	7	0,00%	0	0,00%
227.	481	6	0,00%	6	0,00%	0	0,00%
228.	509	6	0,00%	6	0,00%	0	0,00%
229.	256	5	0,00%	1	0,00%	4	0,00%
230.	258	5	0,00%	1	0,00%	4	0,00%
231.	260	3	0,00%	1	0,00%	2	0,00%
232.	262	3	0,00%	1	0,00%	2	0,00%
233.	264	3	0,00%	1	0,00%	2	0,00%
234.	266	3	0,00%	1	0,00%	2	0,00%
235.	268	3	0,00%	1	0,00%	2	0,00%
236.	270	3	0,00%	1	0,00%	2	0,00%
237.	349	3	0,00%	3	0,00%	0	0,00%
238.	659	2	0,00%	0	0,00%	2	0,00%
239.	272	1	0,00%	1	0,00%	0	0,00%
240.	274	1	0,00%	1	0,00%	0	0,00%
Nr.	Feld	Alle Typen	in %	h Sätze	in %	u Sätze	in %
241.	276	1	0,00%	1	0,00%	0	0,00%
242.	278	1	0,00%	1	0,00%	0	0,00%
243.	280	1	0,00%	1	0,00%	0	0,00%
244.	282	1	0,00%	1	0,00%	0	0,00%

Tabelle A.3: DNB: Statistik der MAB2-Feldnummern, h- und u-Sätze

Erläuterungen zur Tabelle der Feldstatistik

Nr. (erste Spalte): Fortlaufende Nummer in der Tabelle. Aus Platzgründen wurde die Tabelle nach jeweils 40 Einträgen umgebrochen. Insgesamt hat die Tabelle 244 Einträge (entspricht 7 Teil-Tabellen).

Feld (zweite Spalte): Feldnummer der MAB2-Datensätze. Die Feldnummer besteht aus 3 Ziffern. In der Tabelle stehen die Feldnummern zuerst, die am häufigsten in allen Datensätzen vorkommen.

Alle Typen (dritte Spalte): Wieviele Datensätze (alle Satztypen: h, u und y) das betreffende Feld enthalten. Es wird nicht berücksichtigt, ob das Feld Daten enthält oder leer ist. Enthält ein Datensatz ein Feld mehrfach - z.B. mehrere ISBN-Nummern - so wird das Feld nur einmal berücksichtigt. Die prozentuale Angabe bezieht sich auf die Gesamtzahl der Datensätze im Verhältnis zur Häufigkeit des betreffenden Feldes. 100,00 Prozent heißt, daß das Feld in jedem Datensatz existiert. Da jedes Feld nur einmal pro Datensatz berücksichtigt wird, kann die prozentuale Angabe den Wert 100 nicht übersteigen.

Beispiel: In Zeile 21 steht das Feld 540 (ISBN). Von den insgesamt 2.535.095 Datensätzen (h, u, und y) besitzen 1.413.387 Datensätze dieses Feld, das entspricht 55,75%.

h-Sätze (vierte Spalte): Wieviele Datensätze vom Typ h (Hauptsatz) das betreffende Feld enthalten sowie das dazugehörige prozentuale Verhältnis.

Beispiel: In Zeile 21 steht das Feld 540 (ISBN). Von den insgesamt 2.115.316 h-Datensätzen besitzen 1.165.001 Datensätze dieses Feld, das entspricht 55,07%.

u-Sätze (fünfte Spalte): Wieviele Datensätze vom Typ u (Untersatz) das betreffende Feld enthalten sowie das dazugehörige prozentuale Verhältnis.

Beispiel: In Zeile 21 steht das Feld 540 (ISBN). Von den insgesamt 411.759 u-Datensätzen besitzen 248.291 Datensätze dieses Feld, das entspricht 60,30%.

Einige der für die Dublettenkontrolle genutzten Attribute (Titel, Autor, Verlag, Jahr, Verlagsort, ISBN-Nummer und Seitennummer) sind in vielen Datensätzen nicht vorhanden. Die Dublettenkontrolle muß diese Fälle berücksichtigen. Gegebenenfalls müssen andere MAB2-Felder zur Bestimmung der Attribute herangezogen werden.

Autor (Feld 100): Dieses Feld ist in insgesamt 65,0% aller Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist das Autorfeld zu 75,7% belegt. D.h. 1/3 der Datensätze enthalten kein Autorfeld.

Titel (Feld 331): Dieses Feld ist in insgesamt 86,5% aller Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist das Titelfeld zu 96,0% belegt.

Verlagsort (Feld 410): Dieses Feld ist in insgesamt 83,7% aller Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist der Verlagsort zu 69,9% belegt.

Verlag (Feld 412): Dieses Feld ist in insgesamt 83,7% aller Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist das Feld Verlag zu 69,9% belegt. D.h. 1/6 bis 1/3 der Datensätze enthalten damit keine Verlagsangabe.

ISBN-Nummern (Feld 540): Dieses Feld ist in insgesamt 55,7% aller Datensätze vorhanden. Etwas mehr als die Hälfte der Datensätze besitzen eine ISBN-Nummer.

Seitennummer (Feld 433): Das Feld Seitennummer ist in insgesamt 77,4% aller Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist das Feld Seitennummer zu 78,5% belegt. D.h. 1/4 der Datensätze enthalten damit keine Seitennummer.

Auflage (Feld 403): Das Feld Auflage ist in insgesamt 24,2% aller Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist das Feld Auflage zu 23,8% enthalten. Damit besitzen nur 1/4 der Datensätze das Feld Auflage.

Schlagwörter (Feld 902): Das Feld für das erste Schlagwort ist in 30,5% der Datensätze vorhanden. Betrachtet man nur die h-Sätze, ist das Feld zu 35,7% belegt. Damit sind nur 1/3 der Datensätze mit Schlagwörtern versehen.

Die häufigsten Datenfelder

Es werden in der DNB überraschend viele Feldnummern genutzt. Ingesamt gibt es 244 unterschiedliche Feldnummern. Die 75 häufigsten Feldnummern treten in 99% der Datensätze auf. Alle anderen 169 Felder kommen in weniger als 1% der Datensätze vor. Es wäre aber ein voreiliger Schluß, diese seien unwichtig und könnten abgeschafft werden.

Was diese Statistik nicht kann

- Aussagen über einzelne Datensätze treffen.
- Aussagen über Beziehungen zwischen Feldern treffen, z.B. wieviele Werke mit ISBN (Feld 540) haben kein Feld für die Auflage (403). Allerdings können schon gewisse Annahmen getroffen werden: z.B. gibt es 1.648.452 mal das Feld 100 (Autor) und 1.648.451 mal das Feld 102 (Identifikationsnummer des Autors). D.h. praktisch gibt es zu jedem Autor auch eine zugehörige Identifikationsnummer.
- Aussagen über einzelne Reihen treffen.
- Aussagen darüber treffen, ob sich über die Jahre etwas an der Katalogisierungpraxis geändert hat.

Weitere Informationen zu MAB und zur statistischen Analyse von Datensätzen finden sich in [DNB96], [MAB99], [Eve94] und in [Reu99].

Anhang B

Z39.50-Server

Für *ZACK* werden viele Z39.50-Server genutzt. Im folgenden werden die Adressen der Bibliotheken, die Größe des Bestandes, die Ansprechpartner für die Z39.50-Server sowie die technischen Daten der Server aufgeführt. Die meisten Z39.50-Server sind nicht öffentlich zugänglich. Um eine Benutzerkennung zu bekommen, wende man sich bitte an die genannten Ansprechpartner.

B.1 Bibliotheksverbünde in Deutschland

Eine Karte der deutschen Bibliotheksverbünde ist in Kapitel 2.2 Bibliotheken und Bibliotheksverbünde in Deutschland (Seite 7) zu finden.

Bibliotheksverbund Bayern (BVB)

Generaldirektion der Bayerischen Staatlichen Bibliotheken und dem Bibliotheksverbund Bayern. Die Datenbank umfaßt ca. 9,7 Millionen Titelsätze. Stand November 1997 ([BVB97] [1]). Es wird ein BIS-System eingesetzt. Der Z39.50-Server wurde von der Firma Harbinger GmbH (ehemals INOVIS) entwickelt. Der Server befindet sich im produktiven Betrieb.

Adresse:
Generaldirektion der Bayerischen Staatlichen Bibliotheken
D-80328 München
URL: http://www.bib-bvb.de
URL: http://www-opac.bib-bvb.de

Ansprechpartner:
Roland Jäkle
E-Mail: jaekle@bvbnt1.bib-bvb.de
Tel.: 089/28638268

Target Profile:
http://www-opac.bib-bvb.de/subbvb/ordbvb/targetprofile.htm

Hostname	Port	User and Password	Database
bvbx3.bib-bvb.de	31310	auf Anfrage	BVBSR011

[1] Quelle: `http://bvbx1.bib-bvb.de/subbvb/ordbvb/info/bestand.htm`

Gemeinsamer Bibliotheksverbund (GBV)

Gemeinsamer Bibliotheksverbund der Länder Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen, Sachsen-Anhalt, Schleswig-Holstein und Thüringen.

Die Datenbank umfaßt 13 Millionen Datensätze zu Büchern, Zeitschriften, Dissertationen, Mikroformen und elektronischen Dokumenten, davon 6,8 Millionen Titelsätze mit 12,6 Millionen Besitznachweisen aus dem Verbundbereich inklusive Fremddaten. Stand ca. März 1999 ([GBV99] [2]).

Der Bestand dürfte leicht höher liegen, da nach den Informationen auf dem Server des Deutschen Bibliotheksinstituts ([DBI97] [3]) bereits am 31.12.1997 dreizehn Millionen Titelsätze im GBV Verbundkatalog (inklusive Fremddaten) vorlagen. Es wird ein PICA-System eingesetzt ([PIC99]). Der Server befindet sich im produktiven Betrieb.

Adresse:
Verbundzentrale des GBV (GBV/VZ)
Bibliotheksrechenzentrum Niedersachsen (BRZN)
Platz der Göttinger Sieben 1
D-37073 Göttingen
Tel.: 0551/39-5207
Fax.: 0551/39-2408
URL: http://www.brzn.de

Ansprechpartner:
Michael Rathai
Tel.: 0551/39-5269
E-Mail: rathai@gbv.de

Hostname	Port	User and Password	Database
z3950.brzn.de	210	auf Anfrage	GVK

Südwestdeutscher Bibliotheksverbund (SWB)

Südwestdeutscher Bibliotheksverbund (einschließlich Sachsen, Saarland und der südliche Teil von Rheinland-Pfalz), Bibliotheksservice-Zentrum Baden-Württemberg. Die Datenbank umfaßt ca. 7,1 Millionen Titelsätze. Stand Februar 1999 ([SWB99] [4]). Es wird derzeit ein BIS-System eingesetzt. Der Server befindet sich im produktiven Betrieb.

Adresse:
Bibliotheksservice-Zentrum
Baden-Württemberg
Universität Konstanz
D-78457 Konstanz
URL: http://www.swbv.uni-konstanz.de

Ansprechpartner:
Andreas Schnell
E-Mail: schnell@hegne.bsz-bw.de
Tel.: 07531/88-4179

[2] Quelle: `http://www.gbv.de/help/du/nmn_obn.shtml`
[3] Quelle: `http://www.dbi-berlin.de/dbi_koo/vsekr/verbund/vs-1997.htm`
[4] Quelle: `http://www.swbv.uni-konstanz.de/statistik/daten/zugang99.shtml`

Hostname	Port	User and Password	Database
hegau.bsz-bw.de	2080	auf Anfrage	BIBL

Hostname	Port	User and Password	Database
sunsv02.bsz-bw.de	2080	auf Anfrage	BIBL

Hostname	Port	User and Password	Database
sunsv02.bsz-bw.de	3100	auf Anfrage	BIBL

Hochschulbibliothekszentrum NRW (HBZ)

Nordrhein-westfälischer Bibliotheksverbund (Land Nordrhein-Westfalen und nördliche Teile von Rheinland-Pfalz). Die Datenbank des HBZ enthält 8,6 Millionen Titelsätze. Stand 31.12.1998 ([HBZ99] [5]). Eingesetzt wird ein BIS-System.

Adresse:
Hochschulbibliothekszentrum des Landes
Nordrhein-Westfalen
Postfach 41 04 80
D-50864 Köln
Tel: 0221/40075-0
Fax: 0221/40075-180
URL: http://www.hbz-nrw.de
E-Mail: Info@hbz-nrw.de

Kein Z39.50 Server bekannt.

Hessisches Bibliotheks-Informationssystem (HEBIS)

HEBIS ist der elektronische Dienstleistungsverbund aller großen wissenschaftlichen Bibliotheken in Hessen und Teilen von Rheinland-Pfalz/Rheinhessen. Die Datenbank umfaßt ca. 2,4 Millionen Monographien-Titel, ca. 2 Millionen Titel-Fremddaten aus der DNB und 700.000 Fremdtitel aus der Zeitschriftendatenbank (ZDB). Stand 11. März 1999 ([HEB99] [6]). Es wird ein PICA-System eingesetzt.

Adresse:
Hessischer Zentralkatalog
Bockenheimer Landstraße 134-138
D-60325 Frankfurt am Main
Fax: 069/212-3 94 04
URL: http://www.hebis.de/

Kein Z39.50 Server bekannt.

[5] Quelle: `http://www.hbz-nrw.de/hbz/online.html`
[6] Quelle: `http://www.hebis.de/hebis/statistik.html`

B.2 Projekt Kooperativer Bibliotheksverbund Berlin-Brandenburg (KOBV)

Konrad-Zuse-Zentrum / Zentrale des KOBV-Projektes

Die Datenbank umfaßt ca. 2,5 Millionen Datensätze [7] der Deutschen Bibliothek von 1986 bis 1998 (Fremddaten). Als Bibliothekssystem wird ALEPH 500 eingesetzt. Der Z39.50-Server wurde von der Firma Index Data im Auftrag von ExLibris, die das ALEPH-System entwickelt und vertreibt, implementiert ([Ind99]). Der Z39.50-Server befindet sich im Testbetrieb.

Adresse:
Kooperativer Bibliotheksverbund Berlin Brandenburg (KOBV)
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)
Takustr. 7
D-14195 Berlin-Dahlem
Fax: 030/84185-269
Tel: 030/84185-209
URL: http://www.kobv.de

Ansprechpartner:
Josef Willenborg
E-Mail: willenborg@zib.de
Tel.: 030/84185-319

Hostname	Port	User and Password	Database
se.kobv.de	9909	auf Anfrage	dnb01

Universität Potsdam (uni-potsdam)

Die Datenbank der Universitätsbibliothek Potsdam umfaßt ca. 300.000 Titel [8]. Es wird ein allegro-System eingesetzt.

Adresse:
Universität Potsdam
Bereichsbibliothek Babelsberg
August-Bebel-Str. 89, Haus 1
14482 Potsdam
URL: http://www.ub.uni-potsdam.de

Ansprechpartner für technische Fragen:
Raffaele Torsello
E-Mail: torsello@info.ub.uni-potsdam.de
Tel.: 0331/977-3277

Ansprechpartner für organisatorische Fragen:
Dr. Andreas Degkwitz
E-Mail: degkwitz@info.ub.uni-potsdam.de
Tel.: 0331/977-1249

Hostname	Port	User and Password	Database
141.89.36.196	2020	auf Anfrage	AVDEMO

[7] Import der Bandlieferungen der Deutschen Bibliothek 1986-1997. Stand März 1999
[8] Quelle: `Mündliche Angabe von Stefan Lohrum, KOBV, März 1999`

Fachhochschule Potsdam (fh-potsdam)

Die Datenbank der Bibliothek der FH Potsdam umfaßt ca. 50.000 Titel. [9]. Eingesetzt wird ein SISIS-System.

Adresse:
Fachhochschule Potsdam
SG Organisation / Datenverarbeitung
Friedrich-Ebert-Straße 4
D-14467 Potsdam
Postfach 600608
Tel.: 0331/580-2053
Fax.: 0331/580-2999
URL: http://www.fh-potsdam.de

Ansprechpartner:
Andreas Schwenk
E-Mail: schwenk@fh-potsdam.de
Tel.: 0331/580-2053

Hostname	Port	User and Password	Database
193.175.237.103	3950	auf Anfrage	-

Fachhochschule Brandenburg (fh-brandenburg)

Die Datenbank der FH Brandenburg umfaßt ca. 50.000 Titel [10]. Eingesetzt wird ein SISIS-System.

Adresse:
Fachhochschule Brandenburg
Rechenzentrum
Sabine Neumann
Magdeburger Str. 50
D-14770 Brandenburg
Fax: 03381/355-199
URL: http://www.fh-brandenburg.de

Ansprechpartner:
Sabine Neumann
E-Mail: neumann@fh-brandenburg.de
Tel.: 03381/355-175

Hostname	Port	User and Password	Database
195.37.0.30	3950	auf Anfrage	-

[9] Quelle: `Mündliche Angabe von Stefan Lohrum, KOBV, März 1999`
[10] Quelle: `Mündliche Angabe von Stefan Lohrum, KOBV, März 1999`

B.3 Sonstige Z39.50-Server in Deutschland

Die Deutsche Bibliothek (DDB)

Als nationalbibliographisches Informationszentrum hat die Deutsche Bibliothek [11] die Aufgabe, alle eingesandten Pflicht- und Belegexemplare zu verzeichnen. Die Datenbank umfaßt ca. 7,4 Millionen Datensätze. Stand September 1998 ([DDB98a] [12]). Es wird ein PICA-System eingesetzt. Der Server befindet sich im produktiven Betrieb.

Adresse:
Deutsche Bibliothek Frankfurt am Main
Adickesallee 1
D-60322 Frankfurt am Main
URL: http://www.ddb.de
Tel: 069/1525-0
Fax: 069/1525-1010

Ansprechpartner für fachliche und organisatorische Fragen:
Claudia Werner
E-Mail: werner@dbf.ddb.de

Ansprechpartner für technische Fragen und Probleme:
Martina Wiegand
E-Mail: wiegand@dbf.ddb.de

Target Profile:
http://www.ddb.de/partner/ddb_profile.htm

Norm- und Titeldaten

Hostname	Port	User and Password	Database
z3950.dbf.ddb.de	210	auf Anfrage	ILTIS

Ausschließlich Normdaten

Hostname	Port	User and Password	Database
z3950.dbf.ddb.de	210	auf Anfrage	ILTIS

Ausschließlich Titeldaten

Hostname	Port	User and Password	Database
z3950.dbf.ddb.de	210	auf Anfrage	ILTIS

Technische Universität Braunschweig (TUBS)

Lokalbestand der Universitätsbibliothek der Technischen Universität Braunschweig. Der allegro-Katalog umfaßt ca. 600.000 Titel. Der Inhalt dieser Datenbank ist derselbe wie der in der PICA-Datenbank, die die Benutzer im Lesesaal für die Recherche und Ausleihe verwenden. Stand 27.10.98 ([TUB98a] [13]). Eingesetzt wird ein allegro-System.

[11] "*Die Deutsche Bibliothek*" ist ein Eigenname. Dem üblichen Sprachgebrauch folgend wird in dieser Arbeit der Artikel klein geschrieben

[12] Quelle: `http://www.ddb.de/gabriel/en/countries/germany_gateway_dbf.htm`

[13] Quelle: `http://www.biblio.tu-bs.de/allegro/z3950/z39_dbs.htm`

Adresse:
Universitätsbibliothek
Postfach 3329
D-38023 Braunschweig
Tel.: 0531/391-5026, -5011
Fax.: 0531/391-5836
URL: http://www.biblio.tu-bs.de

Ansprechpartner:
Bernhard Eversberg
E-Mail: B.Eversberg@tu-bs.de
Tel.: 0531/391-5026

Hostname	Port	User and Password	Database
ubsun01.biblio.etc.tu-bs.de	2020	opac/opac	opac

Berliner allegroCatalog (baC)

Katalog der Öffentlichen Bibliotheken Berlins, die ein allegro-System einsetzen. Die Datenbank umfaßt 1.481.408 Datensätze. Stand 27.10.98. [14]

Adresse:
Stadtbibliothek Wilmersdorf
Brandenburgische Str. 2
D-10713 Berlin
Tel.: 030/8641-3948
Fax.: 030/8641-3455
E-Mail: stadtbibl@ba-wilm.verwalt.berlin.de
URL: http://www.berlin.de

Ansprechpartner für technische Fragen:
Bernhard Eversberg
E-Mail: B.Eversberg@tu-bs.de
Tel.: 0531/391-5026

Hostname	Port	User and Password	Database
ubsun01.biblio.etc.tu-bs.de	2020	opac/opac	bac

Max-Planck-Institut für Bildungsforschung (MPG)

Eine Spezialbibliothek, die Werke aus den Wissenschaftsbereichen Soziologie, Erziehungswissenschaften und Entwicklungspsychologie (insbesondere Bildungsforschung, Lebensverlaufsforschung, Gerontologie ...) sammelt. Die Datenbank umfaßt ca. 100.000 Titel. Stand März 1999 ([MPI99] [15]). Es wird ein allegro-System eingesetzt.

Adresse:
Bibliothek und wissenschaftliche Dokumentation
Max-Planck-Institut für Bildungsforschung
Lentzeallee 94
D-14195 Berlin

[14] Quelle: `http://www.biblio.tu-bs.de/allegro/z3950/z39_dbs.htm`
[15] Quelle: `http://www.mpib-berlin.mpg.de/DOK/ewas.htm`

Tel.: 030/82406227
Fax.: 030/8249939
URL: http://www.mpib-berlin.mpg.de/DOK/ehome.htm

Ansprechpartner:
Roland Bertelmann
E-Mail: roland@mpib-berlin.mpg.de
Tel.: 030/82406-227

Hostname	Port	User and Password	Database
lib.mpib-berlin.mpg.de	2020	opac/opac	opac

B.4 Z39.50 Server weltweit

Zum Testen der verwendeten Z39.50-Software und des eigenen Systems wurden auch internationale Z39.50-Server genutzt.

BIBSYS

Ein Bibliothekssystem für norwegische Universitäten, Forschungseinrichtungen und die norwegische Nationalbibliothek.

Hostname	Port	User and Password	Database
z3950.bibsys.no	2100	nicht erforderlich	BIBSYS

Bell Labs

Lucent Technologies Library Network.

Hostname	Port	User and Password	Database
z3950.bell-labs.com	210	nicht erforderlich	books, gils, netlib, factbook

Hostname	Port	User and Password	Database
z3950.bell-labs.com	210	any/	acc1

Hostname	Port	User and Password	Database
z3950.bell-labs.com	210	any/any	acc2

Library of Congress (LOC)

Target Profile:
`http://lcweb.loc.gov/z3950/lcserver.html` ([LOC99c]).

Offizieller Server

Hostname	Port	User and Password	Database
ibm2.loc.gov	2210	nicht erforderlich	ocat

Test-Server

Hostname	Port	User and Password	Database
ibm2.loc.gov	210	nicht erforderlich	ocat

University of California Catalog

Hostname	Port	User and Password	Database
melvyl.ucop.edu	210	nicht erforderlich	catalog

Center for Research Libraries (CRL)

Hostname	Port	User and Password	Database
crlcatalog.uchicago.edu	210	nicht erforderlich	innopac

University of Wisconsin-Madison

Hostname	Port	User and Password	Database
z3950.adp.wisc.edu	210	nicht erforderlich	madison

B.5 Weitere Testserver

Die Library of Congress ([LOC99a]) hat auf ihren Web-Seiten eine Liste ([Zte99b] [16]) mit Z39.50-Servern veröffentlicht, die für Tests genutzt werden dürfen. Der Zugang zu diesen Datenbanken ist kostenlos, es wird kein Paßwort verlangt und der Anbieter hat nichts dagegen, daß man neue und gegebenenfalls noch fehlerhafte Clients verwendet.

[16] `http://lcweb.loc.gov/z3950/agency/register/testport.html`

Anhang C

Kurzbeschreibung der Software ZACK

ZACK ist in der Computersprache Perl5 geschrieben ([WCS96], [Per99]). Mit Perl5 kann man schnell einen Prototypen entwickeln und testen. Für Perl5 gibt es eine umfangreiche Softwarebibliothek, insbesondere zur Programmierung von CGI-Scripten. Die Software für *ZACK* gliedert sich in drei Teile:

1. **Perl-Module:** Stellen allgemeine Funktionen zur Verfügung, welche von verschiedenen Programmen oder CGI-Scripten genutzt werden.

2. **CGI-Scripte:** Programme, die vom Web-Server gestartet werden.

3. **Scripte:** Programme, die auf der Kommandozeile gestartet werden.

C.1 MAB2-Perl-Module

Für das MAB2-Format gibt es noch keine Softwarebibliothek. Die benötigten Funktionen wurden deshalb für *ZACK* selbst entwickelt.

C.1.1 Ein- und Ausgabe

MAB2.pm ist ein Modul zum Einlesen von MAB2 Datensätzen. Es stellt die Funktionen `ReadMAB2`, `ReadRawMAB2` und `ReadFormattedMAB2` bereit.

Die Funktion `ReadRawMAB2` erwartet einen Filedescriptor als Argument und gibt einen MAB2-Datensatz als Liste zurück. Die Eingabe ist ein unformatierter Datenstrom, so wie er vom Z39.50-Server geschickt oder auf Magnetband (Tape) geliefert wird.

Die Funktion `ReadFormattedMAB2` erwartet einen Filedescriptor als Argument und gibt einen MAB2-Datensatz als Liste zurück. Die Eingabe ist ein formatierter Datenstrom, so wie er im MAB2-Diskettenformat geliefert wird.

Die Funktion `ReadMAB2` erwartet ein MAB2-Liste als Argument und gibt die MAB2-Datenstruktur zurück. Die Liste besteht aus den Zeilen eines einzelnen MAB2 Datensatzes im Diskettenformat. Die MAB2-Datenstruktur enthält Metainformationen über den Datensatz (Länge, Typ, Version etc.) und ein assoziatives Array der MAB2-Felder.

MAB2out.pm stellt Funktionen zur Ausgabe und Formatierung von MAB2-Objekten bereit.

Die Funktion `WriteTextMAB2` erwartet ein MAB2-Datenobjekt als Argument. Die Ausgabe ist ein für den Benutzer verständlicher, gut lesbarer formatierter Text in Kurzform.

Die Funktion `WriteFormattedMAB2` erwartet ein MAB2-Datenobjekt als Argument. Die Ausgabe ist ein MAB2-Datensatz im Diskettenformat als Liste. Diese Funktion ist das Gegenstück zur Funktion `ReadMAB2`.

C.1.2 Normierung und Dublettenkontrolle

MAB2norm.pm ist ein Modul zur Normierung von MAB2-Datensätzen. Die Funktion `normalisierung` normiert die Felder in MAB2-Datensätzen. Die Variable `$normierung` gibt an, in welcher Form die Felder normiert werden. Es wird eingesetzt im zweiten System von *ZACK* (siehe Kapitel 5 Normierung, Seite 40).

MAB2merge.pm stellt Funktionen zum Mergen (Mischen) von MAB2-Datensätzen bereit. Die Funktion `MergeCandidate` ermittelt aus einer Liste von MAB2-Datensätzen den "*besten*" Datensatz.

Der beste Datensatz wird anhand eines Vergleiches mehrerer Attribute - z.B. Datenbank, Größe des Datensatzes, Erscheinungsjahr - ermittelt.

Die Variable `$algorithm` gibt an, welcher Algorithmus für den Vergleich herangezogen wird. Unterstützt werden zur Zeit:

`YEAR`	die neueste Ausgabe (höchstes Erscheinungsjahr)
`DATABASE`	nach Datenbank, bestimmte Datenbanken haben eine höhere Priorität als andere
`SIZE`	der Datensatz mit den meisten Zeichen wird ausgewählt
`DBY`	nach Datenbank und Jahr. Zuerst wird nach einer Rangliste der Datenbanken verglichen und danach bei Gleichheit nach Jahr. So erhält man immer die neueste Ausgabe aus der Datenbank mit der höchsten Priorität.

Es wird eingesetzt im zweiten System von *ZACK* (siehe Kapitel 7 Ausgabe von Dubletten, Seite 78).

C.2 CGI-Scripte

Das Common Gateway Interface (CGI) ist eine Schnittstelle zwischen dem Web-Server und externen Programmen. Der Web-Server ruft das gewünschte Programm auf, und das Programm bearbeitet die Anfrage des Benutzers.

Die Programme können in einer beliebigen Computersprache geschrieben sein (Perl, TCL, C). Ein Fehler im Programm bringt den Web-Server nicht zum Absturz. Der Web-Server läuft weiter und kann weitere Anfragen beantworten.

Die Programme sind unabhängig vom Web-Server und lassen sich deshalb leicht verändern oder nach Fehlern durchsuchen.

C.2.1 Suche

z ist ein CGI-Script zur Suche in einer Z39.50 Datenbank. Unterstützt werden Titelsuche, Registersuche und Boolesche Verknüpfungen. Siehe in Kapitel 4 Implementierung die Abbildungen 4.3, Seite 24 und 4.7, Seite 29. Es wird eingesetzt in *ZACK*, im ersten wie im zweiten System.

z1 ist ein CGI-Script zur parallelen Suche in mehreren Z39.50 Datenbanken. Siehe Script `z` und in Kapitel 4 Implementierung die Abbildung 4.13 auf der Seite 35 und die Abbildung 4.14, Seite 37. Es wird eingesetzt im 2. System von *ZACK*.

zmenu ist ein Suchmasken-Generator. Der Benutzer kann sich individuell seine Suchmaske zusammenstellen. Er wählt aus, in welchen Attributen standardmäßig gesucht werden soll (Titel, Autor, ISBN etc.), wieviele Boolesche Verknüpfungen es geben soll (ein, zwei oder drei), in welcher Datenbank gesucht wird, die Sprache der Maske (Deutsch, Englisch), die Suchart Find oder Scan, ob die Anfrage trunkiert sein soll. Siehe Abbildung C.1 Seite 130 und C.2 Seite 130. Es wird eingesetzt im ersten System von *ZACK* (siehe Kapitel 4 Implementierung, Seite 16).

zmenu2 ist ein Suchmasken-Generator für die parallele Suche. Das CGI-Script `zmenu2` unterscheidet sich in nur einem Punkt vom CGI-Script `zmenu`: der Benutzer kann mehrere Datenbanken für die parallele Suche auswählen. Es wird eingesetzt im zweiten System von *ZACK* (siehe Kapitel 4.5 Implementierung, Seite 34).

C.2.2 Dokumentation

Diese CGI-Scripte werden im ersten System von *ZACK* eingesetzt (siehe Kapitel 4 Implementierung, Seite 16).

field stellt eine Suchmaske bereit zur Suche in der MAB2, USMARC und UNIMARC Dokumentation. `field` leitet die Anfragen an die entsprechenden CGI-Scripte `d`, `l`, `m` und `u` weiter.

d wird dazu verwendet, möglichst kurze URL auf die MAB2-Dokumentation ([MAB99], [DDB99b]) zu definieren. Statt "`../doc/mab/titelmab300.html#300`" schreibt man "`d?t300`".

Die Länge der URLs wird von ca. 40-60 Zeichen pro Link auf 6 Zeichen verkürzt. Dies ist wichtig, wenn man bei der Ausgabe im Kategorienformat von jeder Feldnummer Links auf die Dokumentation legen will.

Ein CGI-Script ist flexibler als ein direkter Link. So wird vorher überprüft, ob die gewünschten MAB2-Felder gültig sind und auf welchen Teil der MAB2-Dokumentation (Titel, Schlagwort, Personendaten) sich die Anfrage bezieht. Zum Beispiel sind die Felder kleiner als 100 für alle Datentypen (Titel, Schlagwort etc.) gleich. Siehe auch die CGI-Scripte `field`, `l`, `m`, `u`.

l wird dazu verwendet, möglichst kurze URLs auf die Kurzfassung der USMARC-Dokumentation ([MARil]) zu definieren. Statt
"`../doc/usmarc/usmarc-bib200.html#245`"
schreibt man "`l?b245`". Siehe auch die CGI-Scripte `m`, `d` und `u`.

m wird dazu verwendet, möglichst kurze URLs auf die Langfassung der USMARC-Dokumentation zu setzen. Statt
"`../doc/usmarc-concise-bibliographic/ecbdtils.html#mrcb245`"
schreibt man "`m?b245`". Siehe auch das CGI-Script `l`.

u wird dazu verwendet, möglichst kurze URLs auf die Kurzfassung der UNIMARC-Dokumentation ([UNI98]) zu definieren. Statt
"`../doc/unimarc-concise-bibliographic/concise-000.html#020`"
schreibt man "`u?b020`". Siehe auch die CGI-Scripte `d`, `m` und `l`

attr ist ein CGI-Script zur Suche nach BIB1-Attributen in der Online BIB1-Dokumentation ([BIB98]).

C.2.3 Dublettenkontrolle

match Dieses CGI-Script führt eine Dublettenkontrolle mit positiver und negativer Gewichtung durch. Dabei lassen sich die Gewichtungen einzeln einstellen, der positive und negative Schwellwert bestimmen sowie die Art der Normierung, ob kleine Fehler ignoriert werden (Seitenzahl, Tippfehler im Titel, Jahreszahl +/- 1 Jahr etc.). Siehe auch in Kapitel 6.4 Interaktive Dublettenkontrolle die Abbildungen 6.2, 6.3 und 6.4 (Seiten 68, 69 und 70). Es wird eingesetzt im zweiten System von *ZACK*.

C.2.4 FastCGI

FastCGI ist eine Erweiterung des CGI-Interfaces. Im Unterschied zu CGI-Scripten werden FastCGI-Scripte nicht sofort beendet. Ein FastCGI-Script kann deshalb mehrere Anfragen beantworten und eine stehende Verbindung (Session) zu einem Z39.50-Server aufbauen. Bei der Suche in nur einer Datenbank ist die Verwaltung der Session unproblematisch, bei der Suche in mehreren Datenbanken deutlich aufwendiger. Deshalb wird eine stehende Verbindung zum Z39.50-Server nur im ersten System von *ZACK* verwendet. Weitere Informationen zu FastCGI finden sich auf der Fast CGI Homepage ([FCG99]).

ddb.fcg ist ein FastCGI-Script zur Suche in der Deutschen Bibliothek. Es wird eingesetzt in *ZACK* im ersten System.

C.3 Scripte und Programme

C.3.1 Normierung und Dublettenkontrolle

mab2normierung ist ein Testscript zur Normierung der Attribute Titel, Autor, Verlag, Verlagsort, Jahr und ISBN. Die Ausgabe ist eine Statistik der Normierungsfunktionen. Für jede Normierungsfunktion wird angegeben, ob sich die Anzahl der unterschiedlichen Schreibweisen reduziert hat. Es wurde für die Normierung in Kapitel 5 (Seite 40) genutzt.

mab2match sucht nach Dubletten und gibt die zueinander passenden Datensätze aus. Zuerst werden die Felder der Datensätze normiert. Anschließend wird mit einer gewissen Toleranz entschieden, ob zwei oder mehrere Datensätze gleich sind. Die Datensätze werden in einer Kurztrefferliste ausgegeben. Auf Wunsch kann man sich auch im Detail anzeigen lassen, warum zwei Datensätze als dublett erkannt worden sind. Optional können die Normierungsfunktionen, die Gewichtung und die Toleranz beim Vergleich (z.B. Seitenzahl +/- 5 Seiten) angegeben werden.

Es wird eingesetzt im zweiten System von *ZACK* (siehe Kapitel 4.5, Seite 34). Die CGI-Scripte `z1` und `match` benutzen `mab2match` zur Ermittlung der Dubletten.

mab2premerge ist ein Testscript für das Perl-Module `MAB2merge.pm`. Es liest Dubletten ein, ermittelt den "*besten*" Datensatz und gibt ihn aus. Es wurde in Kapitel 7 Ausgabe von Dubletten, Seite 78, zum Testen der Algorithmen verwendet.

mab2splitnorm liest MAB2-Datensätze im Diskettenformat ein, normiert die Felder Autor, Titel, Verlagsort, Verlag, Jahr, ISBN, Seitennummer und Auflage und gibt die normierten Werte aus. Die Ausgabe erfolgt für jedes Feld in eine separate Datei.

Der besseren Übersicht wegen und um Platz zu sparen, wird die Ausgabe sortiert und komprimiert. Für jeden Wert wird angegeben, wie oft er vorkommt. Zum Schluß werden die Werte nach der Häufigkeit ihres Auftretens sortiert, die am häufigsten auftretenden zuerst.

Es wurde für die Normierung in Kapitel 5 (Seite 40) genutzt.

C.3.2 Einlesen und Analyse

mab2extract liest MAB2-Datensätze im Diskettenformat ein und gibt nur die MAB2-Datensätze von dem gewünschten Satztyp aus. Gültige MAB2-Satztypen sind h, k, l, n, p, s, u und y.

mab2split liest MAB2-Datensätze im Diskettenformat ein und gibt die MAB2-Datensätze entsprechend ihrem Satztyp aus. Gültige MAB2-Satztypen sind h, k, l, n, p, s, u und y. Die Ausgabe erfolgt in den Dateien `file`.*satztyp*.

mab2stat liest MAB2-Datensätze im Diskettenformat ein und gibt eine Statistik über die eingelesenen Datensätze aus. Die Statistik enthält Angaben über die Verteilung der MAB2-Satztypen (Titel-, Personen-, Schlagwortdatensatz) und die MAB2-Feldnummern (001, 100, 902 etc.). Es wurde für die Statistik im Anhang A (Seite 102) und die Normierung in Kapitel 5 (Seite 40) genutzt.

tape2disk liest MAB2-Datensätze im Magnetbandformat ein und wandelt die Datensätze in das Diskettenformat um. Es wurde für die Statistik im Anhang A (Seite 102) genutzt.

mab2sort mab2sort ist ein (Test-)Script zum Sortieren von MAB2-Datensätzen. Die MAB2-Datensätze werden nach Autor, Titel, ISBN-Nummer und Jahr alphabetisch sortiert. Dieses Script eignet sich gut zum Ausdrucken von Datensätzen. Es wurde für manuelle Dublettenkontrolle in Kapitel 6.2, Seite 61, genutzt.

C.3.3 Sonstige

recode ist ein Programm zum Umwandeln von Zeichensätzen. Es wurde von Françqis Pinard geschrieben und ist im Quellcode frei verfügbar. Das Programm wurde um die bibliographischen Zeichensätze ISO 5426 und ANSEL erweitert ([ANS99a], [ISO99], [Kos99], [I1899], [REC98]). Es wird in *ZACK* für die Normierung (Kapitel 5, Seite 40) und die Ausgabe der Datensätze (Kapitel 7, Seite 78) genutzt.

zc ist ein Front-End zum YAZ-Kommandozeilen-Client ([YAZ99]). Es erwartet als Argument den Namen einer Datenbank (z.B. "ddb") und startet dann den YAZ-Client mit den richtigen Optionen (Rechnernamen, Portnummern, Datenbanknamen und Paßwörter). Der YAZ-Client stellt die Verbindung zum Z39.50-Server her, und der Benutzer kann dann interaktiv Anfragen stellen. Die Optionen für die Z39.50-Server sind im Script selbst gespeichert. Unterstützt werden alle bekannten Z39.50-Server (siehe Anhang B Z39.50-Server, Seite 115). Es wird in *ZACK* als Z39.50-Client für die Kommunikation mit den Z39.50-Servern genutzt.

C.4 Entwicklungswerkzeuge

CVS Das Concurrent Versions System (CVS) ist ein Tool zur Verwaltung und Versionskontrolle von Software. Es wurde sowohl für die Software als auch für die Dokumentation von *ZACK* genutzt. Es wird die Version CVS 1.10 verwendet.

LaTeX Die Dokumentation wurde mit TeX, Version 3.14159 (C version 6.1) gesetzt.

Perl5 *ZACK* ist in der Computersprache Perl5 geschrieben. Es wird die Version Perl 5.004_04 verwendet.

SunOS *ZACK* wurde auf dem Betriebssystem SunOS 5.5.1 entwickelt. *ZACK* ist portabel und kann problemlos auf einem PC mit dem Betriebssystem FreeBSD 3.2 installiert werden.

C.5 Screenshots der CGI-Scripte

Das CGI-Script `zmenu` (Abbildung C.1, Seite 130) erzeugt interaktiv eine Suchmaske für das WWW-Z39.50-Gateway. Der Benutzer kann angeben, welche Datenbanken und Attribute für die Suchmaske verwendet werden sollen. Außerdem kann die Sprache (Englisch oder Deutsch), die Suchart (Registersuche oder Titelsuche) und die Anzahl der Booleschen Verknüpfungen eingestellt werden.

`zmenu` wird verwendet, um den Benutzern eine individuell auf ihre Bedürfnisse zugeschnittene Suchmaske anzubieten.

Das CGI-Script wird in der Regel nicht vom Benutzer interaktiv aufgerufen. Der Administrator stellt für den Benutzer eine Suchmaske zusammen und speichert die URL des CGI-Scripts mit allen Voreinstellungen als Link ab. Der Benutzer klickt auf den Link und erhält seine persönliche Suchmaske.

Der Administrator kann ohne großen Verwaltungsaufwand Änderungen an der Suchmaske vornehmen. Er muß nur noch ein CGI-Script mit den entsprechenden Links pflegen. Jede Änderung am CGI-Script wirkt sich sofort auf alle Suchmasken aus. Zum Beispiel muß eine Aktualisierung des Copyrights nur noch an einer Stelle vorgenommen werden und nicht in Dutzenden statisch abgelegten HTML-Seiten.

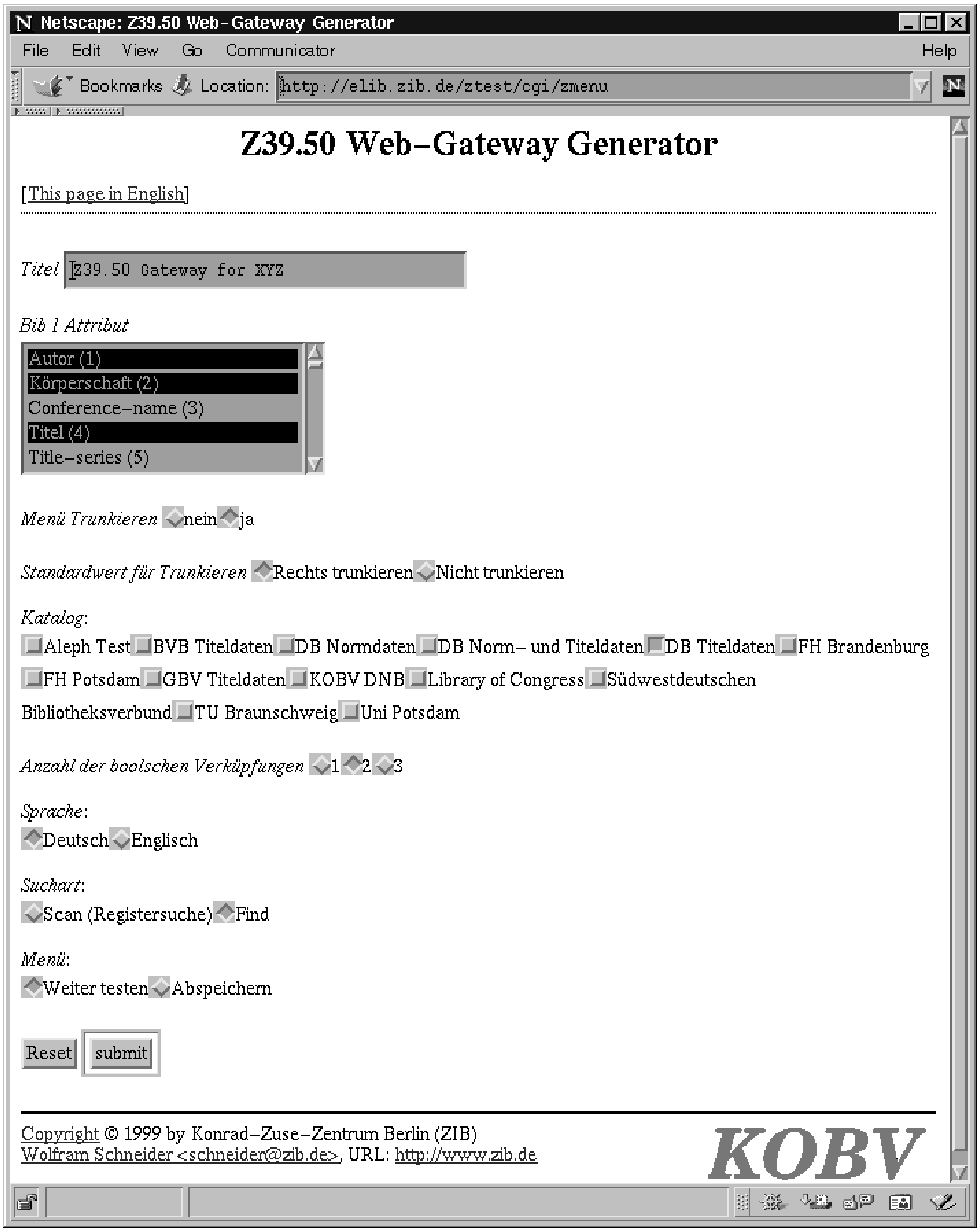


Abbildung C.1: Menügenerator, deutsch

`zmenu` ist zweisprachig, deutsch und englisch. Folgt man dem Link *This page in English*, so wird die Sprache gewechselt, und die Ausgabe erfolgt in englischer Sprache (siehe Abbildung C.2, Seite 131).

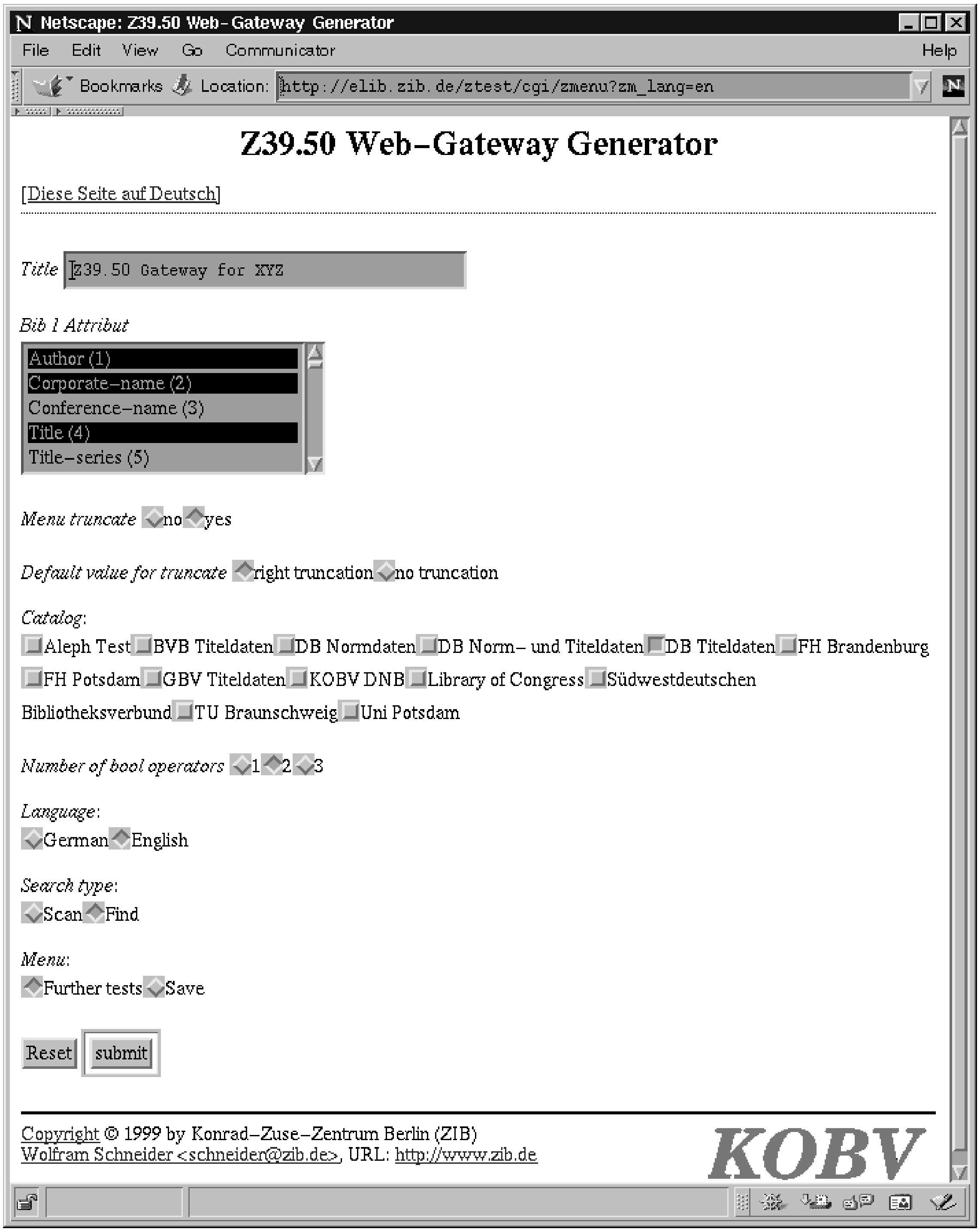


Abbildung C.2: Menügenerator, englisch

Der Menügenerator für die Suchmaske, diesmal in englisch. Siehe Abbildung C.1, Seite 130.

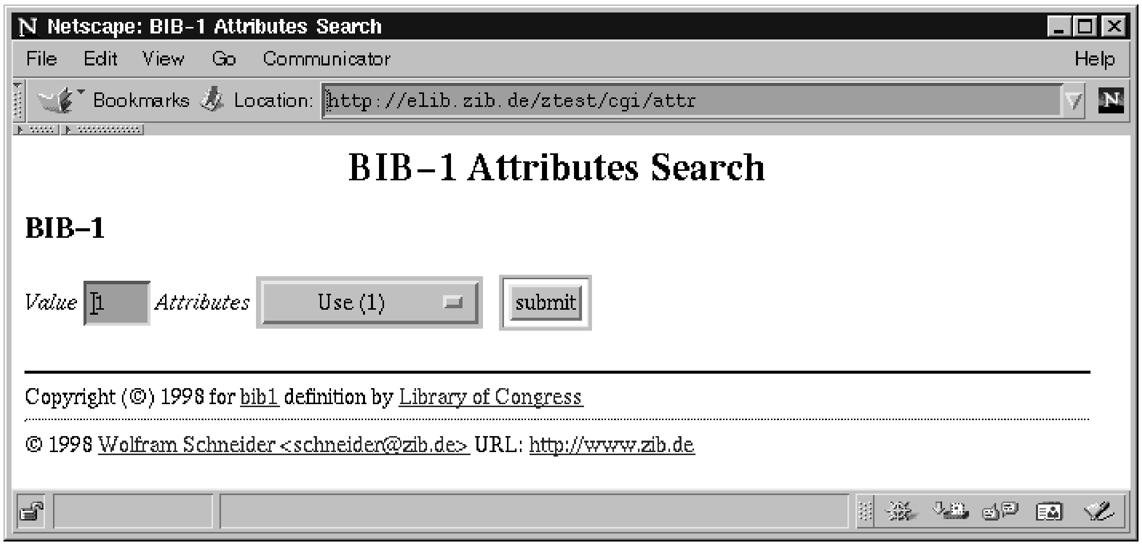


Abbildung C.3: Suche nach BIB-1 Attributen in der Dokumentation

Ein CGI-Script zur Suche nach Attributen in der BIB-1 Dokumentation. Beispielsweise kann man nach dem Attribut 1 mit *Truncation* suchen und erhält als Ergebnis die Dokumentation zur Rechtstrunkierung.

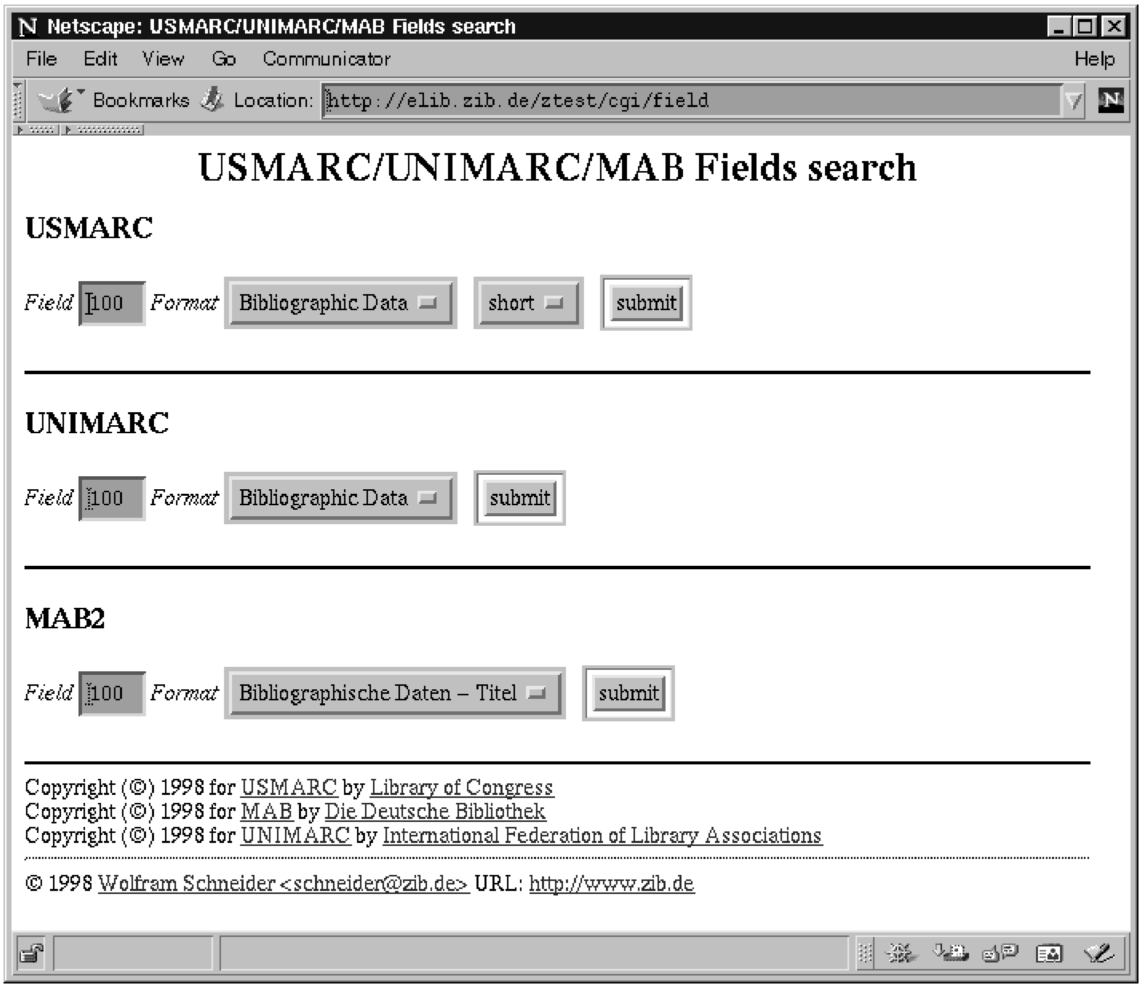


Abbildung C.4: USMARC, UNIMARC, MAB2 Feldsuche

Ein CGI-Script zur Suche nach Feldern in der MAB2, USMARC und UNIMARC Dokumentation. Beispielsweise kann man nach dem Feld 100 in MAB2-Titeldatensätzen suchen und erhält als Ergebnis die Dokumentation zum Feld 100 (Name der 1. Person in Ansetzungsform). Siehe auch Abbildung C.7, Seite 135.

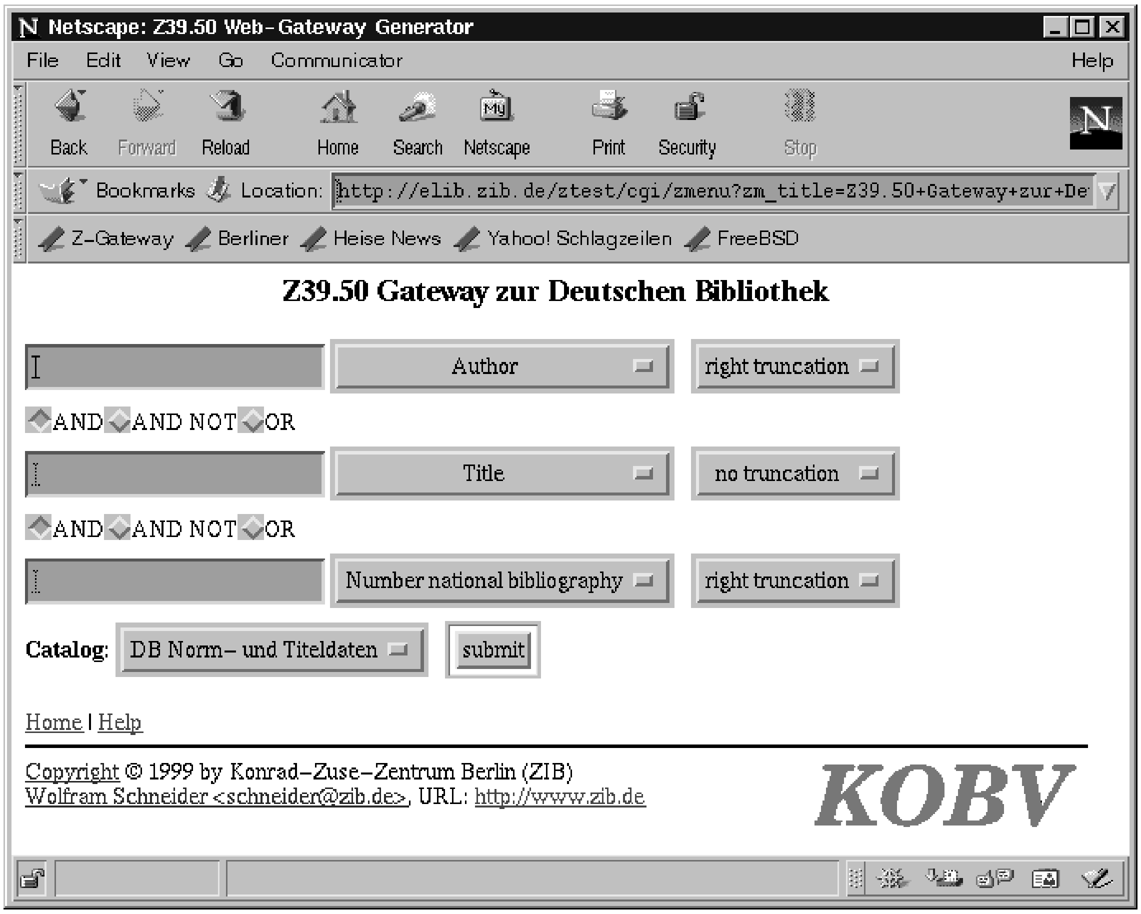
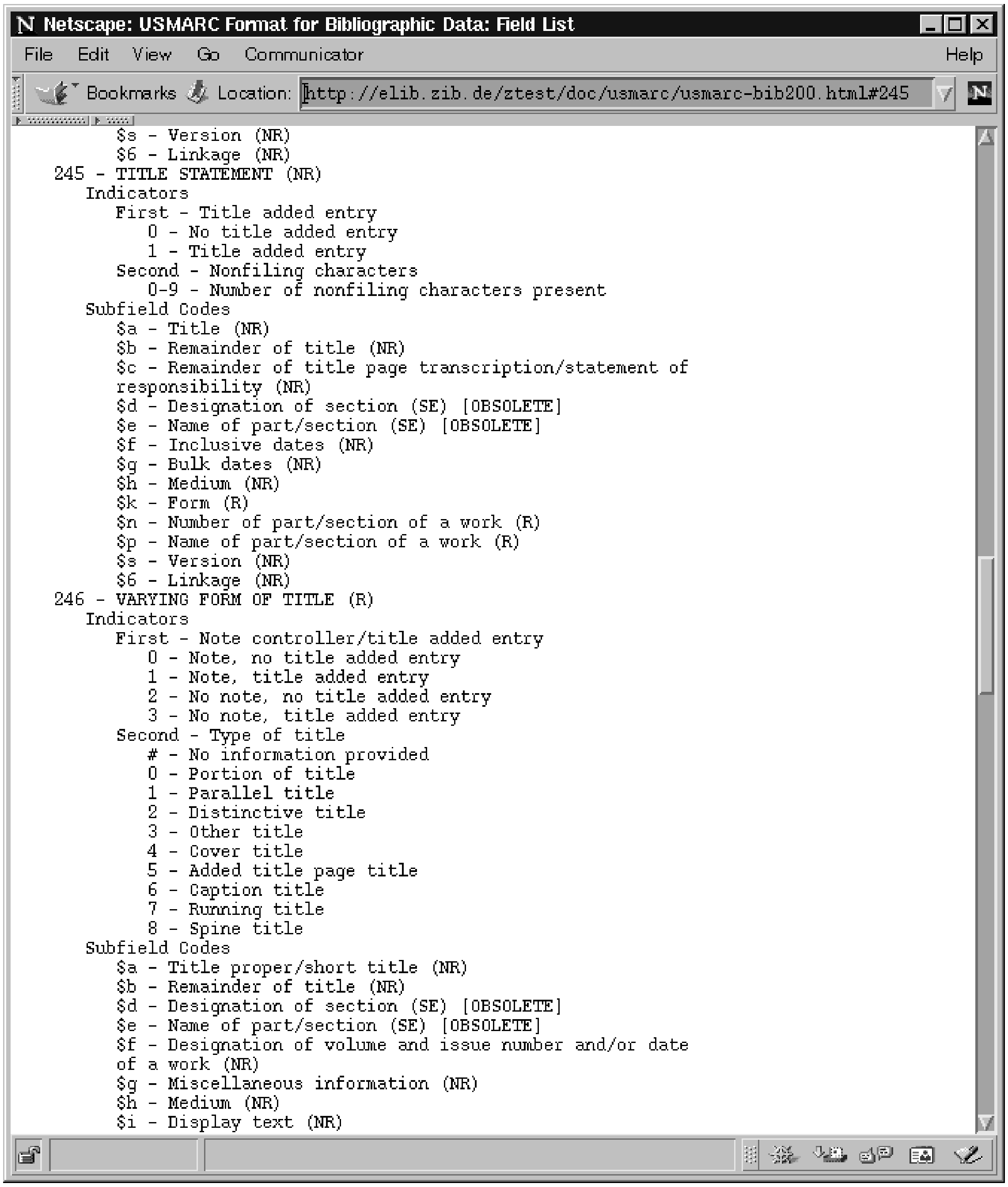


Abbildung C.5: Suchmaske in englisch

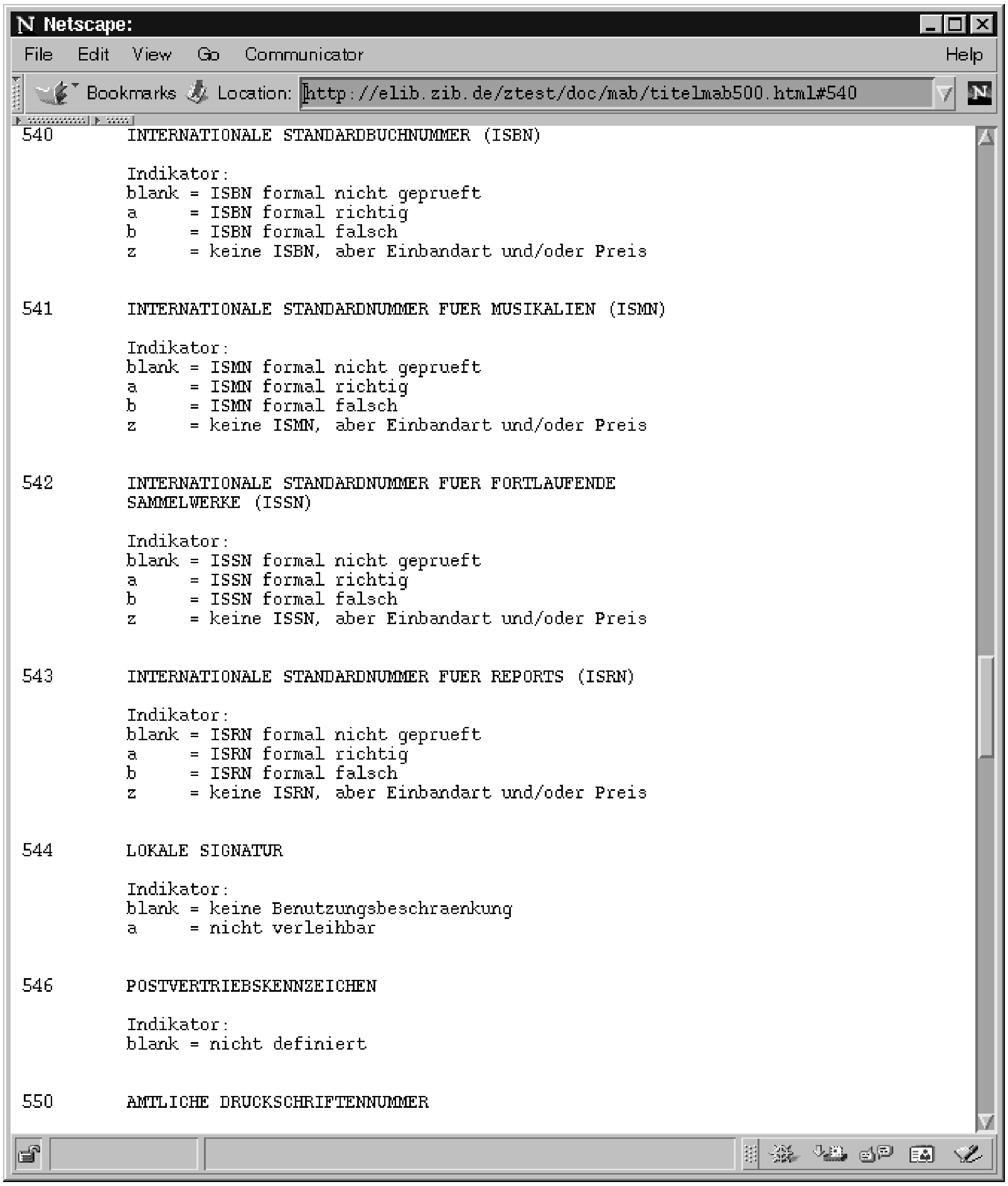
Suchmaske für die Datenbank der Deutschen Bibliothek, diesmal in englisch.




```
         $s - Version (NR)
         $6 - Linkage (NR)
   245 - TITLE STATEMENT (NR)
      Indicators
         First - Title added entry
            0 - No title added entry
            1 - Title added entry
         Second - Nonfiling characters
            0-9 - Number of nonfiling characters present
      Subfield Codes
         $a - Title (NR)
         $b - Remainder of title (NR)
         $c - Remainder of title page transcription/statement of
         responsibility (NR)
         $d - Designation of section (SE) [OBSOLETE]
         $e - Name of part/section (SE) [OBSOLETE]
         $f - Inclusive dates (NR)
         $g - Bulk dates (NR)
         $h - Medium (NR)
         $k - Form (R)
         $n - Number of part/section of a work (R)
         $p - Name of part/section of a work (R)
         $s - Version (NR)
         $6 - Linkage (NR)
   246 - VARYING FORM OF TITLE (R)
      Indicators
         First - Note controller/title added entry
            0 - Note, no title added entry
            1 - Note, title added entry
            2 - No note, no title added entry
            3 - No note, title added entry
         Second - Type of title
            # - No information provided
            0 - Portion of title
            1 - Parallel title
            2 - Distinctive title
            3 - Other title
            4 - Cover title
            5 - Added title page title
            6 - Caption title
            7 - Running title
            8 - Spine title
      Subfield Codes
         $a - Title proper/short title (NR)
         $b - Remainder of title (NR)
         $d - Designation of section (SE) [OBSOLETE]
         $e - Name of part/section (SE) [OBSOLETE]
         $f - Designation of volume and issue number and/or date
         of a work (NR)
         $g - Miscellaneous information (NR)
         $h - Medium (NR)
         $i - Display text (NR)
```

Abbildung C.6: Beschreibung zu Feld 245 (Titel) im Format USMARC

Gesucht wird die Beschreibung zum Feld 245 im Format USMARC. Das Ergebnis ist ein Verweis auf die betreffende Stelle in der Kurz-Dokumentation von USMARC.




```
540       INTERNATIONALE STANDARDBUCHNUMMER (ISBN)

          Indikator:
          blank = ISBN formal nicht geprueft
          a     = ISBN formal richtig
          b     = ISBN formal falsch
          z     = keine ISBN, aber Einbandart und/oder Preis


541       INTERNATIONALE STANDARDNUMMER FUER MUSIKALIEN (ISMN)

          Indikator:
          blank = ISMN formal nicht geprueft
          a     = ISMN formal richtig
          b     = ISMN formal falsch
          z     = keine ISMN, aber Einbandart und/oder Preis


542       INTERNATIONALE STANDARDNUMMER FUER FORTLAUFENDE
          SAMMELWERKE (ISSN)

          Indikator:
          blank = ISSN formal nicht geprueft
          a     = ISSN formal richtig
          b     = ISSN formal falsch
          z     = keine ISSN, aber Einbandart und/oder Preis


543       INTERNATIONALE STANDARDNUMMER FUER REPORTS (ISRN)

          Indikator:
          blank = ISRN formal nicht geprueft
          a     = ISRN formal richtig
          b     = ISRN formal falsch
          z     = keine ISRN, aber Einbandart und/oder Preis


544       LOKALE SIGNATUR

          Indikator:
          blank = keine Benutzungsbeschraenkung
          a     = nicht verleihbar


546       POSTVERTRIEBSKENNZEICHEN

          Indikator:
          blank = nicht definiert


550       AMTLICHE DRUCKSCHRIFTENNUMMER
```

Abbildung C.7: Beschreibung zu Feld 540 (ISBN) im Format MAB2

Gesucht wird die Beschreibung zum Feld 540 im Format MAB2. Das Ergebnis ist ein Verweis auf die betreffende Stelle in der Kurz-Dokumentation von MAB2.

Anhang D

Zugriffsstatistik des WWW-Z39.50-Gateways ZACK

ZACK wird seit mehreren Monaten von Bibliothekaren der Europa-Universität Viadrina Frankfurt (Oder) und der Brandenburgischen Technischen Universität Cottbus genutzt.

In den Log-Dateien des Web-Servers werden alle Zugriffe auf das WWW-Z39.50-Gateway protokolliert. Dazu gehört auch die Information, wonach die Nutzer gesucht haben und wieviele Treffer zur Anfrage gefunden werden. Für den Anbieter einer Datenbank sind neben der Anzahl der Anfragen auch die Anzahl der gelesenen Datensätze interessant. Einige Datenbanken sind kostenpflichtig. Die Anbieter berechnen eine Pauschale oder für jede Anfrage bzw. jeden einzelnen Datensatz Gebühren.

D.1 Zugriffsstatistik von Januar bis April 1999

Die meisten Zugriffe (>90%) erfolgen auf die Datenbank *ILTIS* der Deutschen Bibliothek. In den folgenden Tabellen werden die Anzahl der benutzten Datensätze ausgewertet, aufgeschlüsselt nach Benutzer und Format der Datensätze (Kurz- bzw. Vollformat).

Januar 1999

Benutzer	Kurz-format	Voll-format	Daten-übernahme	Insgesamt
.zib.de	280	17	31	328
.euv-frankfurt-o.de	2.993	1.817	2.644	7.454
.ub.tu-cottbus.de	2.038	597	305	2.940
Insgesamt	5.311	2.431	2.980	10.722

Tabelle D.1: *ZACK*: Zugriffsstatistik DDB/ILTIS, Januar 1999

Februar 1999

Benutzer	Kurz-format	Voll-format	Daten-übernahme	Insgesamt
.zib.de	156	0	15	171
.euv-frankfurt-o.de	2.469	1.064	1.947	5.480
.ub.tu-cottbus.de	2.654	692	294	3.640
Insgesamt	5.279	1.756	2.256	9.291

Tabelle D.2: *ZACK*: Zugriffsstatistik DDB/ILTIS, Februar 1999

März 1999

Benutzer	Kurz-format	Voll-format	Daten-übernahme	Insgesamt
.zib.de	312	33	429	774
.euv-frankfurt-o.de	3.700	1.076	2.110	6.886
.ub.tu-cottbus.de	2.457	707	233	3.397
Insgesamt	6.469	2.431	2.772	11.057

Tabelle D.3: *ZACK*: Zugriffsstatistik DDB/ILTIS, März 1999

April 1999

Benutzer	Kurz-format	Voll-format	Daten-übernahme	Insgesamt
.zib.de	68	7	13	88
.euv-frankfurt-o.de	2.090	1.541	2.295	5.926
.ub.tu-cottbus.de	3.004	742	342	4.088
Insgesamt	5.162	2.290	2650	10.102

Tabelle D.4: *ZACK*: Zugriffsstatistik DDB/ILTIS, April 1999

Legende Zugriffsstatistik

Benutzer:	Alle Benutzer aus der betreffenden Domain
Kurzformat:	Anzeige von Datensätzen im Kurztitelformat (Brief)
Vollformat:	Anzeige von Datensätzen im Vollformat (Full)
Datenübernahme:	Anzeige von Datensätzen im Vollformat (MAB, USMARC, UNIMARC) und Bereitstellung zur Datenübernahme

Auflösung der Domains

.euv-frankfurt-o.de	Europa-Universität Viadrina Frankfurt (Oder), Universitätsbibliothek
.zib.de	Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)
.ub.tu-cottbus.de	Brandenburgische Technische Universität Cottbus, Universitätsbibliothek

D.2 Zusammenfassung

ZACK wird regelmäßig von Bibliothekaren der Europa-Universität Viadrina Frankfurt (Oder) und der Brandenburgischen Technischen Universität Cottbus genutzt. Monatlich werden durchschnittlich 10.000 Datensätze aus der Datenbank ILTIS der Deutschen Bibliothek gelesen - dies entspricht ca. 500 Datensätzen an einem normalen Werktag. Ein Viertel der Datensätze wird in das lokale Bibliothekssystem übernommen.

Anhang E

Abkürzungsverzeichnis

E.1 Abkürzungen

ALEPH ALEPH 500 (Automated Library Expandable Program) ist ein Softwareprodukt, das für die Verwaltung von Bibliotheken und Rechenzentren von Ex Libris entworfen und entwickelt wurde.

allegro Allegro ist ein Datenbanksystem für kleinere und mittlere Bibliotheken. Es wird seit 1980 an der Universitätsbibliothek Braunschweig entwickelt.

ANSEL siehe ANSI/NISO Z39.47-1993

ANSI/NISO Z39.47-1993 Amerikanischer Zeichensatz zum Austausch von bibliographischen Informationen (siehe USMARC).

ANSI American National Standards Institute

Apache The Apache Project. Das Projekt entwickelt einen robusten, voll funktionsfähigen Web-Server, der kostenlos und im Quellcode verfügbar ist.

APDU Application Protocol Data Unit

ASN.1 Abstract Syntax Notation One (ISO 8824)

baC Berliner allegroCatalog, Öffentliche Bibliotheken Berlins

BIB1 bib-1 Attribute Set, siehe auch Z39.50

BIS Bibliotheks-Informationssystem der Firma DABIS

brief auf deutsch: Kurzformat (siehe auch unter "*full*")

BVB Bibliotheksverbund Bayern

CIP Cataloguing in publication; Titelaufnahme in der Veröffentlichung

CGI Common Gateway Interface

cottbus Brandenburgische Technische Universität Cottbus

DB siehe DDB

DBI Deutsches Bibliotheksinstitut, Berlin

DBV-OSI Deutscher Bibliothekenverbund - Open Systems Interconnection

DC Dublin Core

DDB Die Deutsche Bibliothek, Frankfurt am Main

DDB-MAB2 Anwendung des MAB2-Formates durch die Deutsche Bibliothek

DNB Deutsche Nationalbibliographie

elib Der Rechner elib.zib.de, eine SPARCstation-20 mit zwei 50 Mhz CPUs und 160MB Hauptspeicher. Baujahr ca. 1995.

EXL Ex Libris (Deutschland) GmbH, Hamburg

ffo Europa-Universität Viadrina Frankfurt (Oder)

fh-brandenburg Fachhochschule Brandenburg

fh-potsdam Fachhochschule Potsdam

FTP File Transfer Protocol

full auf deutsch: Vollformat. Bei der Übernahme von Datensätzen werden die Datensätze in voller Länge (ungekürzt) über das Z39.50 Protokoll geliefert (siehe auch *brief*).

GBV Gemeinsamer Bibliotheksverbund der Länder Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen, Sachsen-Anhalt, Schleswig-Holstein und Thüringen

GKD Gemeinsame Körperschafts-Datei

HBZ Hochschulbibliothekszentrum NRW

HEBIS Hessisches Bibliotheks-Informationssystem

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

ID Identifikationsnummer

IP Internet Protocol

IR Information Retrieval: inhaltliche Suche in Texten

ISO International Organization for Standardization

ISO 23950 Siehe Z39.50

ISO 5426:1983 Internationaler Standard zum Austausch von bibliograpischen Informationen

ISO 8859-1 Internationaler Zeichensatz für westeuropäische Sprachen

KOBV Kooperativer Bibliotheksverbund Berlin-Brandenburg

KVK Karlsruher Virtueller Katalog

LAN Local Area Network

latin1 siehe ISO 8859-1

LOC Library of Congress

MAB2 Reorganisation des MAB-Formates; veröffentlicht 1995 von der Deutschen Bibliothek

MAB Maschinelles Austauschformat für Bibliotheken

MARC Machine-Readable Cataloging (format). Ein (amerikanischer) Standard zum Austausch von bibliographischen Informationen in maschinenlesbarer Form.

MELVYL Projekt der California Digital Library, University of California

MPG Max-Planck-Institut für Bildungsforschung

NISO National Information Standards Organization, USA

OCLC Online Computer Library Center

OPAC Online Public Access Catalog, Online-Benutzerkatalog

Pica-MARC Internformat des Bibliothekssystems Pica

RAK Regeln für die alphabetische Katalogisierung

RFC Request for Comment, Standards im Internet

PND Personennamendatei

Perl Perl ist eine Computersprache.

RPN Reverse Polish Notation, umgekehrte polnische Notation

Screenshots Ausdruck vom Bildschirm

se2 Der Rechner se2.kobv.de ist eine SUN Enterprise mit zwei UltraSPARC-II 336 Mhz CPUs und einem Gigabyte Hauptspeicher, Baujahr 1998.

SISIS Sisis Informationssysteme GmbH, Oberhaching

SUTRS Simple Unstructured Text Record Syntax

SWB Südwestdeutscher Bibliotheksverbund

SWD Schlagwortnormdatei

SW siehe SWB

TCP Transmission Control Protocol, siehe auch IP

TUBS Technische Universität Braunschweig

uni-potsdam Universität Potsdam

URI Uniform Resource Identifier

URL Uniform Resource Locator

USMARC siehe MARC

WAN Wide area network

WWW World Wide Web

Z siehe Z39.50

Z39.50-1995 Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (Version 3)

Z39.47-1993 siehe ANSI/NISO Z39.47-1993

ZACK Name des eigenen Systems. *ZACK* ist ein Eigenname und keine Abkürzung

ZDB Zeitschriftendatenbank

ZIB Konrad-Zuse-Zentrum für Informationstechnik Berlin

E.2 MAB2-Feldbezeichnungen

Feld-nummer	Kurzbezeichnung
001	IDENTIFIKATIONSNUMMER DES DATENSATZES
010	IDENTIFIKATIONSNUMMER DES DIREKT UEBERGEORDNETEN DATENSATZES
100	NAME DER 1. PERSON IN ANSETZUNGSFORM
102	IDENTIFIKATIONSNUMMER DES PERSONENNAMENSATZES DER 1. PERSON
104	NAME DER 2. PERSON IN ANSETZUNGSFORM
106	IDENTIFIKATIONSNUMMER DES PERSONENNAMENSATZES DER 2. PERSON
108	NAME DER 3. PERSON IN ANSETZUNGSFORM
200	NAME DER 1. KOERPERSCHAFT IN ANSETZUNGSFORM
310	HAUPTSACHTITEL IN ANSETZUNGSFORM
331	HAUPTSACHTITEL IN VORLAGEFORM ODER MISCHFORM
335	ZUSAETZE ZUM HAUPTSACHTITEL
359	VERFASSERANGABE
403	AUSGABEBEZEICHNUNG IN VORLAGEFORM
410	ORT(E) DES 1. VERLEGERS, DRUCKERS USW.
412	NAME DES 1. VERLEGERS, DRUCKERS USW.
425	ERSCHEINUNGSJAHR(E)
433	UMFANGSANGABE
434	ILLUSTRATIONSANGABE / TECHNISCHE ANGABEN ZU TONTRAEGERN
435	FORMATANGABE
455	BANDANGABE
527	HINWEISE AUF PARALLELE AUSGABEN
540	INTERNATIONALE STANDARDBUCHNUMMER (ISBN)
902	KETTENGLIED DER 1. SCHLAGWORTKETTE

Tabelle E.1: Kurzbeschreibung der MAB2-Feldnummern

Eine vollständige Liste der MAB2-Feldnummern ist bei der Deutschen Bibliothek erhältlich ([MAB99], [DDB99b], [DNB96]).

Anhang F

Literaturverzeichnis

Das Literaturverzeichnis enthält gedruckte und elektronische Literatur. Zur gedruckten Literatur gehören veröffentlichte Bücher, Artikel aus wissenschaftlichen Zeitschriften sowie technische Berichte von Forschungseinrichtungen. Die elektronische Literatur enthält Hinweise auf im Internet veröffentlichte Informationen - Handbücher, Standards, Projekte und Homepages.

Die Homepages der deutscher Bibliotheken und Bibliotheksverbünde sind im Anhang B Z39.50 Server aufgeführt.

[Abl97] *Ablösesystem: Informationen über die Neuentwicklung einer Bibliotheksverbundsoftware.* Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (HBZ), Mai 1997. <URL:http://www.hbz-nrw.de/hbz/nsys/abloese.html>.

[Ach99] *The Collection of Computer Science Bibliographies*, April 1999. <URL:http://liinwww.ira.uka.de/bibliography/>.

[ANS99a] *ANSI/NISO Z39.47-1993.* National Information Standards Organization, April 1999. <URL:http://www.niso.org/stantech.html#z3947>.

[Apa99] *Apache Server Project*, April 1999. <URL:http://www.apache.org>.

[BAT99] Inc. Blue Angel Technologies. *An Evaluation of Z39.50 within the SILO Project.* State Library of Iowa, April 1999. <URL:http://www.silo.lib.ia.us/bluang.html>.

[Ber96] Michael Berger. *The user meets the MELVYL(R) system: an analysis of user transactions.* Technischer Bericht 7, Devision of Library Automation, University of California, May 1996. <URL:ftp://ftp.dla.ucop.edu/pub/techreport/user_meets_MELVYL.txt>.

[BFM96] Hella Braune, Hildegard Franck, und Rainer Müller. *Verbundkatalog maschinenlesbarer Katalogdaten deutscher Bibliotheken: Projektbericht 1989 - 1995.* Deutsches Bibliotheksinstitut, Berlin, 1996. ISBN 3-87068-949-8.

[BH91] John A. Bunge und John C. Handley. *Sampling to estimate the number of duplicates in a database.* Computational Statistic & Data Analysis, (11):S. 65–74, 1991.

[BIB95] *Attribute set BIB-1 (Z39.50-1995): semantics.* Library of Congress, September 1995. <URL:ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt>.

[BIB98] *Bib-1 Attribute Set.* Library of Congress, Dezember 1998. <URL:http://lcweb.loc.gov/z3950/agency/defns/bib1.html>.

[BIB99] *HBZ Deutsche Bibliotheken Online: Eine Zusammenstellung aller deutschen Bibliotheken, die Dienste im Internet anbieten*, 22. April 1999. <URL:http://www.hbz-nrw.de/hbz/germlst/>.

[Boo99a] *BookWhere Home Page - Internet Search and Retrieval Software*, April 1999. <URL:http://www.bookwhere.com>.

[Boo99b] *BookWhere Searchable Databases: Z3950*, April 1999.
<URL:http://www.bookwhere.com/library.htm>.

[BOP99] *BOPAC 2 Trial System*, 26. Januar 1999.
<URL:http://www.bopac2.comp.brad.ac.uk/~bopac2/htdocs/evaluate.shtml>.

[BVB97] *Beschreibung des Datenbestandes im Verbundkatalog des Bibliotheks-Verbundes Bayern.* Bibliotheksverbund Bayern, November 1997.
<URL:http://bvbx1.bib-bvb.de/subbvb/ordbvb/info/bestand.htm>.

[BVB99] *Bibliotheksverbund Bayern OPAC/SUBITO.* Bibliotheksverbund Bayern, April 1999.
<URL:http://www-opac.bib-bvb.de>.

[CBuJKW95] Prof. A.B. Cremers, O. Balownew, T. Bode und J. Kalinski, und J. Wolff. *HBZ-Ablösesystem: Auswertung eines Performanztests.* Hochschulrechenzentrum des Landes NRW (HBZ), 1995.
<URL:http://www.hbz-nrw.de/hbz/nsys/abloese.html>.

[CLI96] *Z39.50 Client Survey.* DSTC Pty Ltd., The University of Queensland, 4. September 1996.
<URL:http://www.dstc.edu.au/RDU/reports/zreviews/z3950-client-survey.html>.

[Coy91] Karen Coyle. *Record format for the MELVYL(R) catalog.* Technischer Bericht 3, Devision of Library Automation, University of California, Juni 1991.
<URL:ftp://ftp.dla.ucop.edu/pub/techreport/recformat.txt>.

[Coy92] Karen Coyle. *Rules for merging MELVYL(R) records.* Technischer Bericht 6, Devision of Library Automation, University of California, Juni 1992.
<URL:ftp://ftp.dla.ucop.edu/pub/techreport/mergingrec.txt>.

[DBI97] *Statistik der Verbundsysteme: Informationen zu den regionalen und überregionalen Verbundsystemen in Deutschland.* Deutsches Bibliotheksinstitut, 31. Dezember 1997.
<URL:http://www.dbi-berlin.de/dbi_koo/vsekr/verbund/vs-1997.htm>.

[DBV99] *DBV-OSI II: Ein virtuelles Informationsnetz für Datenbankrecherche, Dokumentbestellung und Dokumentlieferung*, April 1999.
<URL:http://www.ddb.de/partner/dbv-osi_ii.htm>.

[DC:97a] *The 5th Dublin Core Metadata Workshop: Helsinki, Finland, October 6-8, 1997*, 24. Oktober 1997.
<URL:http://linnea.helsinki.fi/meta/projects.html>.

[DC97b] *Dublin Core Metadata Element Set: Reference Description.* Dublin Core Metadata Initiative, Oktober 1997.
<URL:http://purl.oclc.org/dc/about/element_set.htm>.

[DC98] *Dublin Core Metadata for Resource Discovery.* Request for Comments: Network Working Group, September 1998.
<URL:ftp://ftp.cs.tu-berlin.de/pub/doc/rfc/rfc2413.gz>.

[DCZ98] *Dublin Core and Z39.50*, 2. Februar 1998.
<URL:http://www.oclc.org/~levan/docs/dublincoreandz3950.html>.

[DDB98a] *OPAC der Deutschen Bibliothek Frankfurt am Main über Z39.50-Gateway.* Die Deutsche Bibliothek, September 1998.
<URL:http://www.ddb.de/gabriel/en/countries/germany_gateway_dbf.htm>.

[DDB99a] *Die Deutsche Bibliothek - Z39.50 Gateway*, April 1999.
<URL:http://z3950gw.dbf.ddb.de/>.

[DDB99b] *MAB2 : Maschinelles Austauschformat für Bibliotheken.* Die Deutsche Bibliothek, 2. Auflage, 1999. ISBN 3-933641-00-4. Loseblatt - Ausgabe. (Auf dem Stand der Ergänzungslieferung 3).

[Den96] Ray Denenberg. *Z39.50 Recent Developments and Future Prospects*. Library of Congress, Oktober 1996.
<URL:http://lcweb.loc.gov/z3950/agency/papers/kbr.html>.

[DHHS91] Thomas Dierig, Silke Horny, Karin Höpfner, und Karin Söllner. *Untersuchungen zur Einführung eines "Allgemeingültigen Bibliographischen Codes (ABC)" beim Südwestdeutschen Bibliotheksverbund (SWB-Verbund)*. ABI-Technik 11, (3):S. 173–190, 1991.

[DNB96] *MAB2-Datendienst Deutsche Nationalbibliographie*. Die Deutsche Bibliothek, 2. Auflage, 1996. Loseblatt - Ausgabe. (3. Ergänzungslieferung Januar 1999).

[Dug98] Berndt Dugall. *Entwicklung der Bibliotheks-EDV - Von zentralen zu dezentralen Verbundstrukturen*. Kooperativer Bibliotheksverbund Berlin-Brandenburg, 1. September 1998.
<URL:http://www.kobv.de/events/97oct/dugall/>.

[DUP99] *Z39.50 Duplicate Detection Service* . Library of Congress, April 1999.
<URL:http://lcweb.loc.gov/z3950/agency/amend/dedup.html>.

[Eve94] Bernhard Eversberg. *Was sind und was sollen bibliothekarische Datenformate*. Univ.-Bibliothek der TU, Braunschweig, 1994. ISBN 3-927115-21-5.
<URL:http://www.allegro-c.de/allegro/formate/formate.htm>.

[Eve99] Bernhard Eversberg. *Spickzettel für das Katalogisieren*, April 1999.
<URL:http://www.allegro-c.de/allegro/formate/spick.htm>.

[FAQ99] *Questions and Answers about Z39.50*. Sirsi Corporation, April 1999.
<URL:http://www.sirsi.com/Products/z3950nl.html>.

[FCG99] *The Official FastCGI Home Page*, April 1999.
<URL:http://www.fastcgi.com/>.

[Fuh97] Norbert Fuhr. *Skriptum Information Retrieval*. Lehrstuhl VI des Fachbereichs Informatik an der Universität Dortmund, 7. Mai 1997.
<URL:http://ls6-www.informatik.uni-dortmund.de/ir/teaching/courses/ir/>.

[FW97] Sonya Finnigan und Nigel Ward. *Z39.50 Made Simple*. Distributed Systems Technology Centre, University of Queensland, August 1997.
<URL:http://www.dstc.edu.au/DDU/projects/ZINC/zsimple.htm>.

[Gal98] *GALILEO Z39.50 Error Codes*, 14. September 1998.
<URL:http://www.peachnet.edu/galileo/errors.html>.

[GBV99] *GBV Verbundkatalog mit Fremddaten Informationen*. Gemeinsamer Bibliotheksverbund der Länder Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen, Sachsen-Anhalt, Schleswig-Holstein und Thüringen, April 1999.
<URL:http://www.gbv.de/help/du/nmn_obn.shtml>.

[Got96] E. Gottswinter. *Praktischer Einsatz einer SR–Testumgebung*. (2), 1996.
<URL:http://www.ubka.uni-karlsruhe.de/dfg/bericht/bericht.html>.

[Goy84] Pankay Goyal. *An Investigation of Different String Coding Methods*. Journal of the American Society for Information Science, 35, (4):S. 248–252, März 1984.

[Goy87] Pankay Goyal. *Duplicate record identification in bibliographic databases*. Inform. Systems Vol. 12, (3):S. 239–242, Februar 1987.

[HBZ99] *Das Retrievalsystem HBZR*. Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (HBZ), 25. März 1999.
<URL:http://www.hbz-nrw.de/hbz/online.html>.

[HEB99] *HEBIS - Statistische Übersicht*. Hessisches Bibliotheks-Informationssystem, 11. März 1999.
<URL:http://www.hebis.de/hebis/statistik.html>.

[Her96] Bernd Hergeth. *Z39.50 in Bibliotheken und im World-Wide-Web*. Erste INETBIB-Tagung in der Universitätsbibliothek Dortmund, 11-13. März 1996.
<URL:`http://www.ub.uni-dortmund.de/Inetbib/v_herget.htm`>.

[HF96] Sebastian Hammer und John Favaro. *Z39.50 and the World Wide Web*. D-Lib Magazine, März 1996.
<URL:`http://www.dlib.org/dlib/march96/briefings/03indexdata.html`>.

[Hic79] Thomas B. Hickey. *Automatic Detection of Duplicate Monographic Records*. Journal of Library Automation Vol 12/2, Seiten 125–142, Juni 1979.

[HOR97] *HORIZON Benchmark: Benchmark der HORIZON-Applikation auf einem SUN E4000-Cluster*. SUN Benchmark Center, Menlo Park, CA, Juni 1997.
<URL:`http://www.swbv.uni-konstanz.de/lokalsys/horizon/benchmark.html`>.

[HP96] Klaus Haller und Hans Popst. *Katalogisierung nach den RAK-WB. Eine Einführung in die Regeln für die alphabetische Katalogisierung in wissenschaftlichen Bibliotheken*. Saur, München [u.a.], 5., überarbeitete Auflage, 1996. ISBN 3-598-11305-6.

[HTT96] *Hypertext Transfer Protocol – HTTP/1.0*. Request for Comments: Network Working Group, Mai 1996.
<URL:`ftp://ftp.cs.tu-berlin.de/pub/doc/rfc/rfc1945.gz`>.

[HTT99a] *HTML Home Page*. W3C - The World Wide Web Consortium, Juni 1999.
<URL:`http://www.w3.org/MarkUp/`>.

[HTT99b] *HTTP - Hypertext Transfer Protocol Overview*. W3C - The World Wide Web Consortium, Juni 1999.
<URL:`http://www.w3.org/Protocols/`>.

[Hyl96] Jeremy A. Hylton. *Identifying and Merging Related Bibliographic Records*. Diplomarbeit, Massachusetts Institute of Technology, 1996.

[I1899] *Zeichensätze*, April 1999.
<URL:`ftp://dkuug.dk/i18n/charmaps/`>.

[III99] *Innovative Interfaces Inc.*, April 1999.
<URL:`http://www.iii.com/`>.

[Ind99] *Index Data, Homepage*, April 1999.
<URL:`http://www.indexdata.dk/`>.

[Intil] *A pointer page about Z39.50 Resources*, 1999 April.
<URL:`http://ds.internic.net/z3950/z3950.html`>.

[ISO99] *ISO5426: ISO/TC46/SC4 Standards and Scope Statements*. Information Systems Organization, April 1999.
<URL:`http://lcweb.loc.gov/loc/standards/isotc46/sc4standards.html`>.

[IVB96] *Informationen zu den regionalen und überregionalen Verbundsystemen in Deutschland*. Deutsches Bibliotheksinstitut, Berlin, 5., überarbeitete und aktualisierte Auflage, April 1996. ISBN 3-87068-496-8.

[KL97] Monika Kuberek und Stefan Lohrum:. *Kooperativer Bibliotheksverbund Berlin-Brandenburg - Schnittstelle Lokalsysteme-Suchmaschine, Spezifikation der Anforderungen*. Technischer Bericht TR 97-11, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), September 1997.
<URL:`http://www.kobv.de/docs/`>.

[KOB99] *Kooperativer Bibliotheksverbund Berlin-Brandenburg*, April 1999.
<URL:`http://www.kobv.de`>.

[Kos99] *Zeichenkodierung*, April 1999.
<URL:`http://www.kostis.net/charsets/`>.

[KR95] John A. Kunze und R. P. C. Rodgers. *Z39.50 in a Nutshell: An Introduction to Z39.50*. Lister Hill National Center for Biomedical Communications, National Library of Medicine, Juli 1995.
<URL:http://www.informatik.tu-darmstadt.de/VS/Infos/Protocol/Z39.50/z%39.50-nutshell.html>.

[Kru94] Tanja Krutky. *Veränderungen der Katalogisierungstätigkeit durch regionale Verbundsysteme*. Bibliothek - Forschung und Praxis, (1):S. 9–19, 1994.

[Kub97] Monika Kuberek. *Kooperativer Bibliotheksverbund Berlin-Brandenburg - Normdaten im KOBV*. Technischer Bericht TR 97-12, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Oktober 1997.
<URL:http://www.kobv.de/docs/>.

[Kub99a] Monika Kuberek. *Match- und Merge-Verfahren in der KOBV-Suchmaschine - Bibliothekarische Vorüberlegungen*. Technischer Bericht SC 99-16, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Juni 1999.

[Kub99b] Monika Kuberek. *Umgang mit hierarchischen Strukuren (MAB2) in der KOBV-Suchmaschine*. Technischer Bericht SC 99-15, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Juni 1999.

[KVK99] *Karlsruher Virtueller Katalog (externer Zugang)*, April 1999.
<URL:http://www.ubka.uni-karlsruhe.de/kvk.html>.

[KW95] John W. Kirriemuir und Peter Willett. *Identification op duplicate and near-duplicate full-text records in database search-outputs using hierarchic cluster analysis*. Program, Vol 29, (3):S. 241–256, Juli 1995.

[LOC97] *Dublin Core/MARC/GILS Crosswalk*. Library of Congress: Network Development and MARC Standards Office, April 1997.
<URL:http://lcweb.loc.gov/marc/dccross.html>.

[LOC99a] *Library of Congress Home Page*, April 1999.
<URL:http://www.loc.gov/>.

[LOC99b] *Z39.50 maintenance agency home page*. Library of Congress, April 1999.
<URL:http://lcweb.loc.gov/z3950/agency/>.

[LOC99c] *LC Z39.50 Server Configuration Guidelines*. Library of Congress, 26. März 1999.
<URL:http://lcweb.loc.gov/z3950/lcserver.html>.

[LSW99] Stefan Lohrum, Wolfram Schneider, und Josef Willenborg. *De-duplication in KOBV*. Technischer Bericht SC 99-05, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Februar 1999.
<URL:http://www.zib.de/bib/pub/pw/index.de.html>.

[Lyn97] Clifford A. Lynch. *The Z39.50 Information Retrieval Standard*. D-Lib Magazine, April 1997.
<URL:http://www.dlib.org/dlib/april97/04lynch.html>.

[MAB99] *Maschinelles Austauschformat für Bibliotheken, Homepage*, April 1999.
<URL:http://www.ddb.de/partner/mab.htm>.

[Mac79] Keith D. MacLaury. *Automatic Merging of Monographic Data Bases - Use of Fixed-Length Keys Derived from Title Strings*. Journal of Library Automation Vol 12/2, Seiten S. 143–155, Juni 1979.

[MAR99] *MARC to UCS/Unicode Character Mapping*. Library of Congress: Network Development & MARC Standards Office, Februar 1999.
<URL:http://lcweb.loc.gov/marc/marc2ucs.html>.

[MARil] *MARC Standards*. Library of Congress Network Development and MARC Standards Office, 1999 April.
<URL:http://lcweb.loc.gov/marc/>.

[Met99a] *Deutsche Meta-Suchmaschine*, April 1999.
<URL:http://meta.rrzn.uni-hannover.de/>.

[Met99b] *MetaCrawler*, April 1999.
<URL:http://www.metacrawler.com>.

[MPI99] *MPIB Biblio Einführung*. Max-Planck-Institut für Bildungsforschung, März 1999.
<URL:http://www.mpib-berlin.mpg.de/DOK/ewas.htm>.

[Net99] *Netscape Netcenter - Download &; Upgrade Page for browsers, servers, shareware*, Juni 1999.
<URL:http://www.netscape.com/>.

[NIS97] *NIST Z39.50 Implementation papers*, Mai 1997.
<URL:http://lcweb.loc.gov/z3950/agency/papers/nist.html>.

[ORO93] Edward T. O'Neill, Sally A. Rogers, und W. Michael Oskins. *Characteristics of Duplicate Records in OCLC's Online Union Catalog*. Library Resources & Technical Services 37, (1):S. 59–71, 1993.

[Ott94] Tania Otto. *Die Entstehung und Verbreitung der ISBN und ihre Verwendung in deutschen wissenschaftlichen Bibliotheken*, März 1994. Diplomarbeit.

[Pay96] Margarete Payer. *MELVYL als Beispiel einer zentralen Datenbank für ein Hochschulnetz*. Technischer Bericht, 17. Juni 1996.
<URL:http://www.payer.de/einzel/melvyl.htm>.

[Per99] *Perl.com Homepage*, April 1999.
<URL:http://www.perl.com>.

[PIC99] *Pica - Bibliotheekautomatisering en Online Informatiediensten*, April 1999.
<URL:http://www.pica.nl/ne/>.

[PR97] Sandra D. Payette und Oya Y. Rieger. *Z39.50: The User's Perspective*. D-Lib Magazine, April 1997.
<URL:http://www.dlib.org/dlib/april97/cornell/04payette.html>.

[Pup88] Frank Puppe. *Einführung in Expertensysteme*. Springer, Berlin, 1988. ISBN 3-540-19481-9.

[RAK99] *Regeln für die alphabetische Katalogisierung: RAK-WB*. Deutsches Bibliotheksinstitut, Berlin, 3., überarbeitete Auflage, 1999. Hrsg. von der Kommission des Deutschen Bibliotheksinstituts für Erschließung und Katalogmanagement. Loseblatt - Ausgabe.

[REC98] *GNU recode*. Free Software Foundation, August 1998.
<URL:http://www.cn.gnu.org/software/recode/recode.html>.

[Reu99] *Reuse: Vorschläge zur verbesserten Umsetzung Mehrteiliger MARC-Strukturen*. Niedersächsische Staats- und Universitätsbibliothek, April 1999.
<URL:http://www.oclc.org/oclc/cataloging/reuse_project/final_reuse_german.htm>.

[Rid92] M. J. Ridley. *An expert system for quality control and duplicate detection in bibliographic databases*. Program, Vol 26, (1):S. 1–18, Januar 1992.

[RM94] Markus Reichart und Michael W. Mönnich. *Dublettenkontrolle in bibliographischen Datenbanken*. Bibliothek - Forschung und Praxis, (2):S. 193–216, 1994.

[Rus99a] Beate Rusch. *Normierungen von Zeichenfolgen als erster Schritt des Precise Match*. Technischer Bericht SC 99-13, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 1999.
<URL:http://www.zib.de/bib/pub/pw/index.de.html>.

[Rus99b] Beate Rusch. *Test des KOBV-Z39.50 Servers DNB01 (März 1999)*. Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 6. April 1999. Interner Bericht des KOBV.

[SH91] Karin Söllner und Karin Höpfner. *Studie zur zur Einführung eines "Allgemeingültigen Bibliographischen Codes (ABC)" beim Südwestdeutschen Bibliotheksverbund (SWB-Verbund)*. Technischer Bericht 3, Universität Konstanz: Südwestdeutscher Bibliotheksverbund, 31.01.1991.

[Shn97a] Ben Shneiderman. *Designing Information-Abundant Websites: Issues and Recommendations*. Technischer Bericht CS-TR-3634, Department of Computer Science & Institute for Systems Research, University of Maryland, 26. Februar 1997.
<URL:http://www.cs.umd.edu/projects/hcil/members/bshneiderman/ijhcs/main.html>.

[Shn97b] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publishing Company, Reading, MA, 3. Auflage, 1997. ISBN 0201694972.

[SWB99] *SWB-Statistik: Zugänge in die Datenbank 1999*. Bibliotheksservice-Zentrum Baden-Württemberg (BSZ), 6. April 1999.
<URL:http://www.swbv.uni-konstanz.de/statistik/daten/zugang99.shtml>.

[Tan95] Andrew S. Tanenbaum. *Distributed operating systems*. Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1995. ISBN 0-13-219908-4.

[Ton92] Stephen R. Toney. *Cleanup and Deduplication of an International Bibliographic Database*. Information Technology and Libraries, Seiten 19–28, März 1992.

[TUB98a] *Z3950-allegro-Datenbanken: Lokalbestand der Universitätsbibliothek*. Universitätsbibliothek Braunschweig, 31. Dezember 1998.
<URL:http://www.biblio.tu-bs.de/allegro/z3950/z39_dbs.htm>.

[TUB98b] *Z3950-allgemeine Information*. Universitätsbibliothek Braunschweig, 31. Dezember 1998.
<URL:http://www.biblio.tu-bs.de/allegro/z3950/basic.htm#Was_ist_das>.

[UNI98] *Universal Bibliographic Control and International MARC Core Programme*. IFLA: International Federation of Library Associations and Institutions, Mai 1998.
<URL:http://www.ifla.org/VI/3/p1996-1/concise.htm>.

[uSU95] Jiří Kende und Steffen Uhlig. *Dublettenermittlung bei der Zusammenführung von Bibliotheken: (Nicht nur) ein statistisches Verfahren*. Bibliothek - Forschung und Praxis, (3):S. 412–419, 1995.

[Ver99] *Regionale Verbundsysteme in der Bundesrepublik Deutschland*, Juni 1999.
<URL:http://www.brzn.de/w3-bvb-karte.html>.

[WCS96] Larry Wall, Tom Christiansen, und Randal L. Schwartz. *Programming Perl*. O'Reilly, Bonn [u.a.], 2. Auflage, 1996. ISBN 1-56592-149-6.

[Wil79] Martha E. Williams. *Automatic Merging of Monographic Data Bases*. Journal of Library Automation Vol 12/2, Seiten S. 156–168, Mai 1979.

[Wil97] Josef Willenborg. *Die Suchmaschine des KOBV - Spezifikationen der Anforderungen*. Technischer Bericht TR 97-13, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), August 1997.
<URL:http://www.kobv.de/docs/>.

[YAZ99] *The YAZ Toolkit*, März 1999.
<URL:http://www.indexdata.dk/yaz.html>.

[Z3995] *Z39.50-1995 Maintenance Agency Text*. Library of Congress, Juli 1995.
<URL:ftp://ftp.loc.gov/pub/z3950/official/>.

[ZET99] *Finsiel zeta suite home page*, April 1999.
<URL:http://zeta.tlcpi.finsiel.it/>.

[Zna97] *The ZNavigator*, 4. Dezember 1997.
<URL:http://www.sbu.ac.uk/litc/caselib/znavig.htm>.

[Zpr99] *Z39.50 Projects*. Library of Congress, April 1999. <URL:http://lcweb.loc.gov/z3950/agency/projects/projects.html>.

[ZPZ81] E. M. Zamora, J. J. Pollack, und Antonio Zamora. *The use of trigram analysis for spelling error detection*. Information Processing & Managment Vol 17, (6):S. 305–316, 1981.

[ZRE96] *Z39.50 Client and Web Gateway Surveys*. DSTC, The University of Queensland, Australia, September 1996. <URL:http://www.dstc.edu.au/RDU/reports/zreviews/index.html>.

[Zso99] *Z39.50 Software*. Library of Congress, April 1999. <URL:http://lcweb.loc.gov/z3950/agency/projects/software.html>.

[Zte99b] *Z39.50 Hosts Available for Testing*. Library of Congress, April 1999. <URL:http://lcweb.loc.gov/z3950/agency/register/testport.html>.

Danksagung

Diese Diplomarbeit entstand in Zusammenarbeit mit verschiedenen Personen und Institutionen. Ich danke an dieser Stelle ganz herzlich:

Prof. Dr. Erhard Konrad für die Leitung der Diplomarbeit. Außerdem seiner Mitarbeiterin Ulrike Reiner, bei der ich viel über Informationssysteme und Bibliotheken gelernt habe.

Prof. Dr. Martin Grötschel für die Leitung der Diplomarbeit. Außerdem dem Projektleiter des KOBV Joachim Lügger am Konrad-Zuse-Zentrum, der dem Projekt KOBV den Weg geebnet hat.

dem KOBV-Team am Konrad-Zuse-Zentrum. Insbesondere Beate Rusch und Josef Willenborg dafür, daß sie soviel Zeit für die Betreuung und kritische Begleitung der Diplomarbeit aufgewendet haben.

der Deutschen Bibliothek für den Zugang zu ihrem Z39.50 Server. Insbesondere Martina Wiegand, die geduldig meine Fragen zum Z39.50 Server und dem MAB2-Format der deutschen Bibliothek beantwortet hat.

den Verbünden Bibliotheksverbund Bayern, Gemeinsamer Bibliotheksverbund und dem Südwestdeutschen Bibliotheksverbund für den Zugang zu ihren Z39.50 Servern.

der Universitätsbibliothek Braunschweig für den Zugang zu ihrem allegro-Z39.50 Server. Dieser Server war der erste Z39.50-Server einer deutschen Universitätsbibliothek und nicht eines Bibliotheksverbundes.

den Brandenburger Bibliothekaren von der Universitätsbibliothek der Europa-Universität Viadrina Frankfurt (Oder) und der Brandenburgischen Technischen Universität Cottbus für die produktive Nutzung von *ZACK*. Insbesondere Herrn Günter Todt für seine Kritik, Anregungen und aufmerksamen Beobachtungen.

der Firma Index Data dafür, daß sie ihre Z39.50-Software kostenlos und ohne Restriktionen der Allgemeinheit zur Verfügung gestellt hat.