

Kölner Arbeitspapiere zur Bibliotheks- und Informationswissenschaft

Band 41

Zugang zu verteilten bibliografischen
Datenbeständen

Konzepte und Realisierungen für
heterogene Umgebungen

Marie-Luise Fritz

Juni 2004

Fachhochschule Köln
Fakultät für Informations- und Kommunikationswissenschaften
Institut für Informationswissenschaft

Fritz, Marie-Luise:
Zugang zu verteilten bibliografischen Datenbeständen:
Konzepte und Realisierungen für heterogene Umgebungen.
Köln : Fachhochschule Köln, Fakultät für
Informations- und Kommunikationswissenschaften,
Institut für Informationswissenschaft, 2004. -
(Kölner Arbeitspapiere zur Bibliotheks- und Informationswissenschaft; 41)

ISSN (elektronische Version) 1434-1115

Die Kölner Arbeitspapiere zur Bibliotheks- und Informationswissenschaft berichten über aktuelle Forschungsergebnisse des Instituts Informationswissenschaft der Fachhochschule Köln. Veröffentlicht werden sowohl Arbeiten der Dozentinnen und Dozenten als auch herausragende Arbeiten der Studierenden. Die Kontrolle der wissenschaftlichen Qualität der Veröffentlichungen liegt bei der Schriftleitung.
Jeder Band erscheint in elektronischer Version (über unsere Homepage: <http://www.fbi.fh-koeln.de/institut/papers/arbeitspapiere.php>).

Fachhochschule Köln Fakultät für Informations- und Kommunikationswissenschaften
Institut für Informationswissenschaft
Claudiusstr.1 D-50678 Köln
Tel.: 0221/8275-3376, Fax: 0221/3318583
E-Mail: schriftenreihe@fbi.fh-koeln.de

Schriftleitung: Prof. Dr. Haike Meinhardt, Sabine Drescher, Nina Falkenstein
© FH Köln 2004

ABSTRACT

Das steigende Angebot an Datenbanken im Bereich bibliografischer Informationen führt dazu, dass es für Benutzer sehr mühsam und mit einem enormen Zeitaufwand verbunden ist, in allen einzelnen Systemen zu recherchieren, da diese unterschiedliche Benutzeroberflächen und Recherchemöglichkeiten anbieten. Verteilte bibliografische Informationssysteme ermöglichen einen einheitlichen Einstiegspunkt zu heterogenen Systemen. Sie bündeln verschiedene Angebote. In diesem Text wird ein Überblick über diese verteilten Informationssysteme gegeben. Es wird dargestellt, wie eine Suche in einem solchen System abläuft, welche weiteren Funktionen angeboten werden können. Darüber hinaus wird betrachtet, wie verteilte Informationssysteme aufgebaut und vernetzt sind. Die Darstellungen werden verdeutlicht durch praktische Beispiele aus dem Bibliotheks- und Informationswesen. Insbesondere wird auf den KVK, ZACK, die KOBV-Suchmaschine, die Digitale Bibliothek NRW, das DIMDI, vascoda und Renardus eingegangen.

INHALT

EINFÜHRUNG.....	4
1 ORGANISATION UND ABFRAGE BIBLIOGRAFISCHER DATENBESTÄNDE	7
1.1 Definition bibliografischer Datenbestände	7
1.2 Quellen bibliografischer Datenbestände	8
1.3 Verwaltung der Datenbestände in Datenbanken.....	9
1.3.1 Architektur	10
1.3.2 Indizes	11
1.4 Suche in Datenbanken.....	12
1.4.1 Oberfläche für die Suche.....	12
1.4.2 Retrievalfunktionen.....	13
1.4.3 Kurzanzeige.....	14
1.4.4 Vollanzeige	14
2 FUNKTIONEN VERTEILTER INFORMATIONSSYSTEME	15
2.1 Bereitstellen einer gemeinsamen Recherche- oberfläche	16
2.2 Suchen.....	17
2.3 Lokalisieren.....	17
2.4 Anfordern, Liefern	18
2.5 Benutzer erkennen, Zugriff kontrollieren	19
2.6 Zusätzliche Funktionen	19
3 ARCHITEKTUR VERTEILTER INFORMATIONSSYSTEME.....	20
3.1 Modell für die Architektur: MIA Logical Architecture	20
3.2 Datenhaltung	22
3.2.1 Verteilte Suche	23
3.2.2 Zentrale Suche durch Harvesting.....	24
3.2.3 Vergleich der verteilten Suche und der zentralen Suche durch Harvesting	26
4 SUCHE IN VERTEILTEN INFORMATIONSSYSTEMEN.....	30
4.1 Auswahl von Datenquellen	30
4.2 Sucheinstiege	32
4.3 Retrievalfunktionen.....	33
4.4 Feldinhalte.....	34
4.5 Ausgabe der Kurztrefferliste.....	35
4.5.1 Anzeige und Anordnung der Treffer.....	35
4.5.2 Umgang mit Dubletten.....	36

5	TECHNISCHE REALISIERUNG VERTEILTER INFORMATIONSSYSTEME	38
5.1	Z39.50	39
5.2	HTTP.....	42
5.3	Web-Services	44
5.3.1	Standards: SOAP und WSDL	45
5.3.2	ZING/SRW	46
5.3.3	Protokoll für Harvesting - OAI-PMH	46
5.4	Einsatz von Protokollen in verteilten Informationssystemen.....	48
5.4.1	Einheitlicher Zugriff.....	48
5.4.2	Gateways	49
5.4.3	Individuelle Anpassung.....	49
6	VERSCHIEDENE REALISIERUNGEN IN DEUTSCHLAND	51
6.1	KVK.....	51
6.2	ZACK.....	53
6.3	KOBV-Suchmaschine.....	55
6.4	Digitale Bibliothek NRW.....	57
6.5	DIMDI.....	59
6.6	vascoda.....	61
6.7	Renardus.....	63
7	SCHLUSS	65
8	LITERATURVERZEICHNIS	66

EINFÜHRUNG

Ein Merkmal unserer heutigen Zeit ist die Fülle von elektronisch vorliegenden Informationen, hier als Datenbestände bezeichnet¹. Je unübersichtlicher und heterogener das Informationsangebot ist, desto mühsamer ist es für einen Benutzer, bei einer Recherche alle benötigten Informationen zu finden. Er muss zum einen wissen, wo er Informationen finden kann, zum anderen muss er bei einer Recherche mit verschiedenen Datenquellen, das heißt mit verschiedenen Informationsspeichern, umgehen können. Letzteres kann sehr anspruchsvoll sein, da einzelne Datenquellen oft unterschiedliche Speicherstrukturen, Oberflächen und Zugangssysteme besitzen. Eine Lösung für dieses Problem ist die Schaffung eines einheitlichen Zugangs zu verschiedenen heterogenen Datenquellen. Systeme, die einen solchen Zugang bieten, sind das Thema dieser Arbeit.

Voraussetzung für einen einheitlichen Zugang zu verschiedenen Datenquellen ist die Vereinigung der einzelnen Datenquellen. Dies kann entweder physisch stattfinden, indem alle Datenbestände in einem einzigen zentralen System zusammengefasst werden, oder virtuell. In dieser Arbeit geht es um virtuelle Einheiten, bei denen die einzelnen Datenquellen dem Benutzer zwar als eine Einheit dargestellt werden, bei einer Recherche jedoch auf die einzelnen unabhängigen Datenquellen, auch Provider genannt, zugegriffen wird. Für diese Vereinheitlichung wird ein System benötigt, das als Bindeglied zwischen den Benutzer und die unterschiedlichen Provider geschaltet ist. Dies wird als Broker-System, Broker, Middleware² oder auch als "trading place"³ bezeichnet. Durch die Verknüpfung entsteht ein neues System, und zwar ein verteiltes Informationssystem⁴.

Im Rahmen dieser Arbeit werden nur verteilte Informationssysteme betrachtet, die mit Hilfe von Brokern verknüpft sind und die einen speziellen Inhalt, und zwar bibliografische Daten, anbieten (eine Definition für bibliografische Daten folgt in Kapitel 1). Deshalb bezieht sich der Begriff „verteilte Informationssysteme“ in dem ganzen vorliegenden Text ausschließlich auf die soeben beschriebenen Systeme.

¹ Vgl. Schmidt, Joachim W. und Gerald Schröder: [Artikel] Daten. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S. 160

² Vgl. zum Beispiel Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment. Online: <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/dner-arch.html> [Stand: 18.05.2001, Abrufdatum: 07.10.2003, Datei: mia_dner.htm]

³ Dempsey, Lorcan; Rosemary Russell und Robin Murray: A utopian place of criticism: brokering access to network information. In: Journal of Documentation 55 (1999) 1, S. 52

⁴ Vgl. Ness, Andreas J: Eine Systemarchitektur für die Gestaltung und das Management verteilter Informationssysteme. Band 143 aus der Reihe IPA-IAO Forschung und Praxis. Berlin: Springer 1990, S. 23

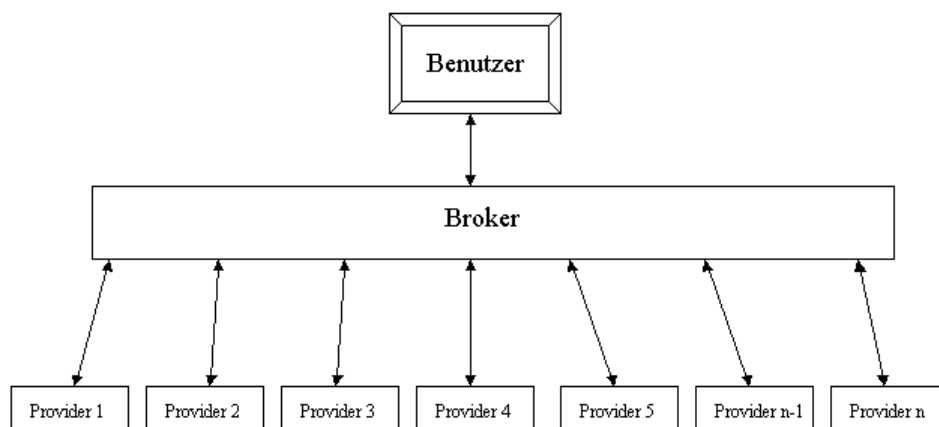


Abb. 1: Darstellung der drei Komponenten Benutzer, Broker und Provider. Wegen der Übersichtlichkeit ist nur ein Benutzer dargestellt; es können jedoch auch mehrere Benutzer auf einen Broker zugreifen.

Ziel dieser Arbeit ist es, einen Überblick über den Aufbau und die Funktionsweise von verteilten bibliografischen Informationssystemen zu geben. Dies geschieht, indem zuerst verschiedene Elemente solcher Systeme theoretisch dargestellt und Möglichkeiten für die Realisierung erläutert werden (Kapitel 1-5) und anschließend einige existierende verteilte Informationssysteme vorgestellt werden (Kapitel 6). Der Schwerpunkt liegt dabei auf der Suche verteilter Datenbestände und nicht auf der Anzeige und Beschaffung von diesen.

In Kapitel 1 wird der Blick auf die Provider gerichtet. Es werden Inhalte und Formen bibliografischer Datenbanken betrachtet sowie die Verwaltung von Datenbeständen und der Zugriff auf diese. In allen folgenden Kapiteln geht es um verteilte Informationssysteme. In Kapitel 2 werden die verschiedenen Funktionen, die ein solches System erfüllen kann, vorgestellt, in Kapitel 3 dessen Architektur. Im Rahmen der Architektur werden unterschiedliche Möglichkeiten der Datenhaltung aufgeführt und verglichen. In Kapitel 4 wird eine bereits in Kapitel 2 erwähnte Funktion eines verteilten Informationssystems, und zwar die Suche, ausführlicher betrachtet, bevor in Kapitel 5 eingesetzte Protokolle und dazugehörige Schnittstellen vorgestellt werden.

Grundlage für die Aufführung der verschiedenen Möglichkeiten in Kapitel 2 bis 5 ist zum einen die aus dem Literaturverzeichnis zu entnehmende Fachliteratur, zum anderen beruhen die Aussagen auf der Analyse und dem Vergleich existierender Systeme. Der Schwerpunkt liegt auf Systemen, die im deutschen Bibliothekswesen eine wichtige Rolle spielen. Sieben von ihnen, der KVK (Karlsruher Virtueller Katalog), ZACK, die KOBV-Suchmaschine (Suchmaschine des Kooperativen Bibliotheksverbunds Berlin-Brandenburg), die Digitale Bibliothek NRW (Digitale Bibliothek Nordrhein-Westfalen), das DIMDI (Deutsches Institut für Medizinische Dokumentation und Information), vascoda und Renardus, werden in Kapitel 6 vorgestellt. Sie führen die Bandbreite der verschiedenen Möglichkeiten bei

dem Aufbau und der Funktionsweise verteilter Informationssysteme vor Augen.

Spezifische Informationen über diese Systeme konnten nicht ausschließlich aus öffentlich zugänglichen Dokumenten gewonnen werden. Deshalb stammen manche Informationen aus internen Berichten oder wurden aus einem Schriftverkehr mit Fachleuten sowie durch persönlich geführte Telefonate mit diesen gewonnen. Insbesondere für die Vorstellung des noch relativ neuen Systems *vascoda* war die Autorin auf zahlreiche persönliche Angaben angewiesen. Neben den in Kapitel 6 vorgestellten Systemen basiert die Darstellung in Kapitel 1 bis 5 zum Teil auch auf Informationen über die Virtuelle Fachbibliothek Sozialwissenschaften und EULER sowie auf dem Service *iPort* von OCLC PICA.

1 ORGANISATION UND ABFRAGE BIBLIOGRAFISCHER DATENBESTÄNDE

In diesem Kapitel wird der Blick auf einzelne bibliografische Datenbestände und Datenquellen gelenkt und noch nicht auf die Verknüpfung von diesen. Im ersten Abschnitt wird definiert, was bibliografische Datenbestände sind. Anschließend werden verschiedene Datenquellen, in denen sich die bibliografischen Datenbestände befinden, aufgeführt. In den weiteren Abschnitten geht es um den Aufbau und die Funktionsweise von Datenquellen und um die Suche nach darin abgelegten Beständen.

1.1 Definition bibliografischer Datenbestände

Bibliografische Daten sind Informationen über Dokumente⁵. Unter Dokumenten sind hier Publikationen wie zum Beispiel Bücher, Zeitschriften, weitere Medien wie CD-ROMs, Videos oder Websites zu verstehen. Außerdem Diplomarbeiten, Dissertationen, Normen, Reports, Patentschriften, Konferenzberichte und vieles mehr, was größtenteils zur Grauen Literatur gehört. Bibliografische Daten als Informationen über Dokumente sind zum Beispiel Angaben über Titel und Autor eines Dokuments. Solche Angaben sind Daten über andere Daten, in unserem Fall über Dokumente, weshalb sie auch als Metadaten bezeichnet werden können. Sie dienen dazu, Dokumente eindeutig zu identifizieren und verweisen damit auf diese⁶.

Das Gale Directory of Databases⁷ grenzt bibliografische Daten von anderen ab, indem es sie der Gruppe der wortorientierten ("word-oriented") Daten zuordnet im Gegensatz zu den zahlorientierten ("number-oriented"), bildorientierten ("image-oriented") und lautorientierten ("sound-oriented"). Innerhalb der wortorientierten Daten unterscheidet es unter anderem zwischen bibliografischen (bibliographic) Daten, Patent-/Warenzeichen-Daten (patent/trademark), Volltexten (full-text) und Adressdaten (directory).

Diese Einteilung erscheint auf den ersten Blick klar und logisch. Allerdings lassen sich Volltexte und bibliografische Daten heute schwer trennen, da bibliografische Daten oft zusammen mit einem digital vorliegenden Dokument in Suchsystemen abgelegt sind. In einem solchen Fall können Daten recherchierbare Metadaten und gleichzeitig Teile eines Dokumentes selbst sein. Bei einer HTML-Seite zum Beispiel strukturieren sogenannte Tags das

⁵ Vgl. Begriffe im Kontext des Information Retrieval: Web-Lehrsammlung HU Berlin. Online: <http://www.ib.hu-berlin.de/~is/web-lehrsammlung/Begriffe/Retrieval.htm> [Stand: 02.05.1999, Abrufdatum: 04.09.2003, Datei: ir_begriffe.htm]

⁶ Vgl. Humphrey, Susanne M. und Biagio John Melloni: Databases: a primer for retrieving information by computer. New Jersey: Prentice-Hall 1986, S. 2

⁷ Gale Directory of Databases: Volume 1: Online Databases: 2002 Part 1. Hrsg.: Erin Nagel. Detroit: Gale Group 2001, S. XXI

Dokument und machen dadurch Teile eines Dokuments zu Metadaten. Die typische Trennung von bibliografischen Daten und Dokumenten, das heißt von bibliografischen Daten und Volltexten, ist damit nicht mehr klar beizubehalten. In der vorliegenden Arbeit sind unter dem Begriff "bibliografische Datenbestände" alle Bestände eingeschlossen, die Dokumente nachweisen, unabhängig davon, ob das Dokument selbst enthalten ist oder ob nur darauf verwiesen wird.

1.2 Quellen bibliografischer Datenbestände

Bibliografische Datenbestände sind in verschiedenen Arten von Speicher- und Zugangssystemen abgelegt, die hier, wie bereits definiert, als Datenquellen bezeichnet werden. In dieser Arbeit werden nur elektronische Datenquellen berücksichtigt, weshalb auch nur diese hier aufgezählt werden.

Die im Bibliothekswesen am häufigsten benutzten Quellen sind Kataloge. Kataloge weisen den Bestand von Bibliotheken nach und sind öffentlich zugänglich, weshalb sie auch als OPAC (Online Public Access Catalogue) bezeichnet werden. Verbundkataloge bzw. -datenbanken enthalten Daten über die Bestände mehrerer Bibliotheken. Darüber hinaus gibt es Spezialkataloge, die bestimmte Medien oder Inhalte nachweisen, wie zum Beispiel die Zeitschriftendatenbank (ZDB), die die Zeitschriftenbestände vieler deutscher Bibliotheken enthält⁸.

Neben Katalogen gibt es viele weitere Quellen bibliografischer Datenbestände. In manchen Datenquellen sind ausschließlich spezielle Medienformen, Publikationen eines bestimmten Landes oder nur an bestimmte Organisationen oder Institutionen gebundene Dokumente nachgewiesen. Ein Beispiel für eine Medienform sind Zeitschriften. Entweder können die einzelnen Zeitschriftenartikel als elektronische Volltexte in Datenbanken abgelegt sein oder nur die Inhaltsverzeichnisse in sogenannten "table of contents". Der Buchhandel als organisatorische Einheit weist seine Bestände in Buchhandelsverzeichnissen nach. Neben diesen formalen Kriterien findet jedoch häufiger eine Auswahl nach Fachgebieten statt⁹. Diese Datenquellen werden als bibliografische Fachdatenbanken bezeichnet. Es gibt auch Datenquellen, die Dokumente enthalten, die sowohl nach fachlichen Gesichtspunkten ausgewählt sind als auch nach formalen.

Noch nicht erwähnt sind die Dokumente, die im World Wide Web (WWW), verkürzt auch als Web bezeichnet, angeboten werden. Das WWW wird hier, wie im allgemeinen Sprachgebrauch üblich, als Synonym zu "Internet" gebraucht, obwohl es genau genommen nur eine von vielen Anwendungen des Internet ist¹⁰. Das WWW als Ganzes könnte als eine Datenquel-

⁸ Vgl. „Homepage der Zeitschriftendatenbank (ZDB)“. Online: <http://www.zdb-opac.de> [Abrufdatum: 18.10.2003, Datei: zdb.htm]

⁹ Vgl. Begriffe im Kontext des Information Retrieval

¹⁰ Fasbender, Andreas und Otto Spaniol: [Artikel] WWW. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S. 808

le bezeichnet werden. Dies wäre aus zwei Gründen jedoch sehr unspezifisch. Zum einen, weil über das WWW neben einzelnen Webseiten auch auf alle bereits oben erwähnten Quellen zugegriffen werden kann, wenn die technischen Voraussetzungen dafür erfüllt sind. Zum anderen gibt es kein Zugangssystem für das gesamte WWW, auch wenn universelle Suchmaschinen versuchen, möglichst viel nachzuweisen. Deshalb wird hier nicht das WWW als Ganzes als Datenquelle bezeichnet, sondern stattdessen Systeme im WWW, die Zugang zu einer Auswahl von Dokumenten aus dem WWW schaffen. Dies leisten Portale und Suchmaschinen¹¹.

An Internetportalen und -suchmaschinen wird deutlich, dass zu Quellen bibliografischer Datenbestände nicht nur Systeme gehören, die Datenbestände enthalten, sondern auch Broker-Systeme, die lediglich für den Zugang zu bibliografischen Daten sorgen. Auch hier gibt es Mischformen: Quellen können sowohl eigene Daten anbieten als auch Zugang zu anderen schaffen.

Weitere Datenquellen sind Dokumentlieferdienste. Sie werden weniger für den Nachweis von Dokumenten als vielmehr für die Bestellung von diesen genutzt.

1.3 Verwaltung der Datenbestände in Datenbanken

Nach der Aufzählung der verschiedenen Datenquellen wird nun der Blick darauf gelenkt, wie bibliografische Datenbestände abgelegt und verwaltet werden. Dies geschieht in der Regel in Datenbanken. Zuerst wird die Architektur von Datenbanken vorgestellt. Anschließend Indizes, die eine der wichtigen Grundvoraussetzungen dafür sind, dass abgelegte Datenbestände an Hand von inhaltlichen Kriterien schnell wiedergefunden werden.

¹¹ Vgl. Rösch, Hermann: Portale in Internet, Betrieb und Wissenschaft: Marktplatz und Instrument des Kommunikations- und Wissensmanagements. In: BIT Online (2001) 3. Online: <http://www.b-i-t-online.de/archiv/2001-03/fach1.htm> [Stand: 2001, Abrufdatum: 17.10.2003, Datei: portal.htm]

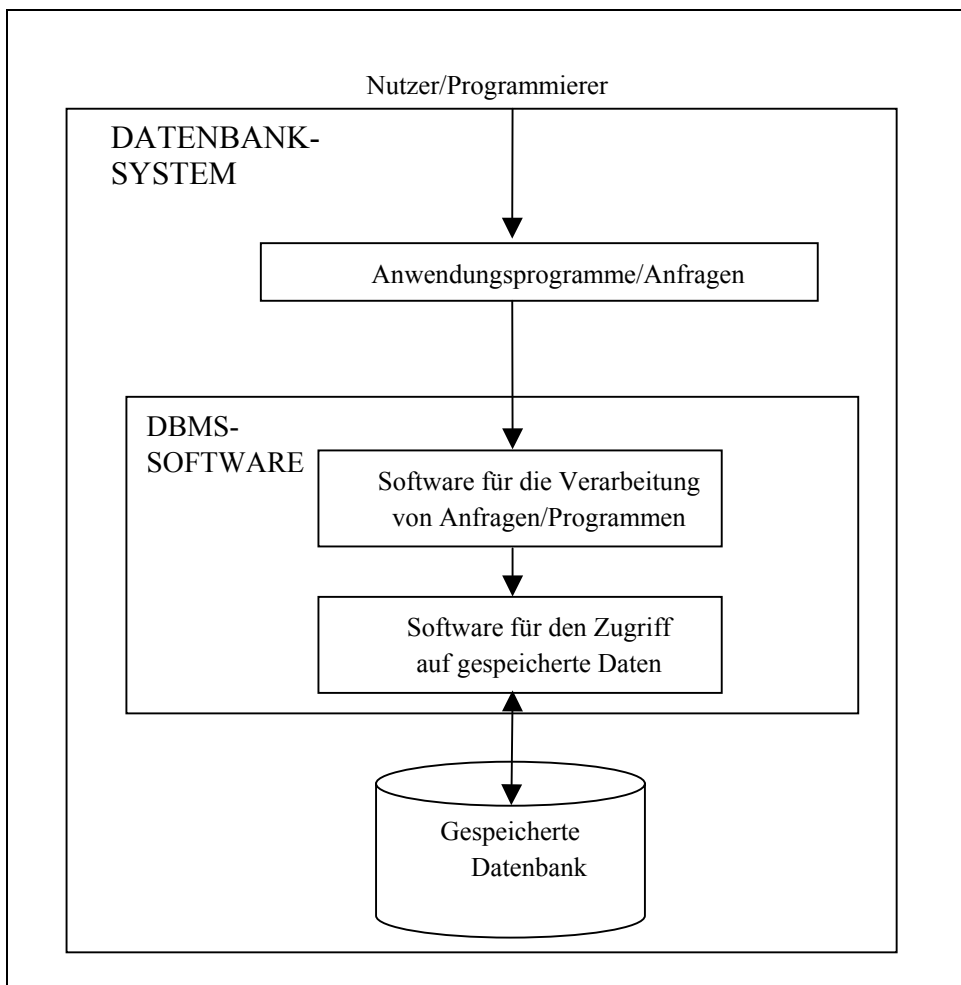


Abb.2: Vereinfachte Darstellung eines Datenbanksystems

Quelle: Elmasri, Ramez und Shamkant B. Navathe: Grundlagen von Datenbanksystemen. 3. Aufl. München: Pearson Studium 2002, S. 26. Die Abbildung wurde leicht verändert.

1.3.1 Architektur

Der Begriff "Datenbank" beinhaltet im engeren Sinn lediglich den gespeicherten Datenbestand, im weiteren Sinn gilt er als Synonym für "Datenbanksystem". In diesem technisch ausgerichteten Abschnitt wird der Begriff im engeren Sinn verwendet, in allen anderen Abschnitten wird auf Grund seiner Gebräuchlichkeit¹² der weite Begriff benutzt. Ein Datenbanksystem besteht aus einer oder mehrerer Datenbanken und aus Programmen (s. Abb. 2)¹³. Wie aus Abbildung 2 deutlich wird, sind Daten und Programme strikt getrennt.

¹² Vgl. Gaus, Wilhelm: Dokumentations- und Ordnungslehre: Theorie und Praxis des Information Retrieval. 3. Aufl. Berlin: Springer 2000, S. 35

¹³ Vgl. Elmasri, Ramez und Shamkant B. Navathe: Grundlagen von Datenbanksystemen, 3. Aufl. München: Pearson Studium 2002, S. 26.

Es gibt verschiedene Arten von Programmen: Anwendungsprogramme und Datenbankmanagementsysteme (DBMS). Ein Datenbankmanagementsystem enthält Software für die Verarbeitung von Anfragen und für den Zugriff auf gespeicherte Daten. Anwendungsprogramme stellen die Schnittstelle zu Benutzern her. Ein Datenbanksystem kann von verschiedenen Nutzern angesprochen werden, wie zum Beispiel von Administratoren oder Endnutzern. In dieser Arbeit geht es um Endnutzer. Bei Endnutzern kann es sich sowohl um Menschen als auch um Maschinen, das heißt um Computerprogramme, handeln.

In Datenbanksystemen sind die Datenbestände als Datensätze strukturiert in der Datenbank abgelegt. Das bedeutet, dass ein Datensatz in einzelne, verschieden gekennzeichnete Bestandteile, sogenannte Datenfelder, zerlegt wird. Welche Felder angelegt werden, ist die Entscheidung des Datenbankherstellers. Für die spätere Suche ist diese Entscheidung grundlegend. Bei der Beschreibung eines Dokuments können zum Beispiel der Titel und der Untertitel einem einzigen Datenfeld zugeordnet werden oder zwei verschiedenen. Dadurch werden Weichen gestellt, wie der Datenbanknutzer bei einer Recherche suchen kann: einzeln nach einem Zusatz zum Sachtitel oder nur in Verbindung mit dem Titel.

1.3.2 Indizes

Eine Art von Datenbanksystemen sind Information Retrieval Systeme (IRS)¹⁴. Diese sind speziell darauf ausgerichtet, Dokumente oder deren Metadaten zu verwalten und zugänglich zu machen¹⁵. In den in diesem Kapitel noch folgenden Ausführungen werden Mechanismen dieser Systeme vorgestellt.

Um Datenbestände, insbesondere sehr große, schnell und effektiv nach bestimmten Zeichenketten durchsuchen zu können, findet in der Regel keine sequentielle Suche in den einzelnen Datensätzen, der sogenannten „primären Dateiorganisation“¹⁶, statt, sondern es werden „sekundäre Zugriffspfade“¹⁷ erstellt. Dies geschieht mit Hilfe von „Indizes“¹⁸, im Bereich des Information Retrieval auch als invertierte Listen bezeichnet. Ein Index ist eine geordnete Liste von Wörtern, die Verweise auf Datensätze enthalten, in denen das entsprechende Wort vorkommt. Neben Verweisen können Indizes weitere für das Information Retrieval wichtige Informationen enthalten, wie zum Beispiel Gewichtungsfaktoren für die einzelnen Verweise. Bei einer Suche werden die Wörter oder Zeichen, die vom Benutzer eingegeben werden, mit denen im Index verglichen. Sobald Zeichenketten als gleich er-

¹⁴ Vgl. Begriffe im Kontext des Information Retrieval

¹⁵ Vgl. Salton, Gerard und Michael McGill: Information Retrieval: Grundlegendes für Informationswissenschaftler. Hamburg: McGraw Hill 1987, S. 8

¹⁶ Elmasri, Ramez und Shamkant B. Navathe: Grundlagen von Datenbanksystemen, S. 141

¹⁷ Ebd., S. 185

¹⁸ Elmasri verwendet für den Plural auch "Indexe" (Elmasri: Grundlagen von Datenbanksystemen, S. 185ff.)

kannt werden, kann im Index festgestellt werden, welche Datensätze die gesuchte Zeichenkette enthalten.

Bei der Generierung von Indizes muss entschieden werden, welche Datenfelder ausgewertet werden. Die von Suchmaschinen im WWW oft angewendete Praxis, jedes Wort eines Dokumentes im Index zu verzeichnen und damit suchbar zu machen (Volltextindexierung), wird in vielen bibliografischen Datenbanken nur eingeschränkt angewendet. Meist wird ein Index auf einzelne Datenfelder beschränkt, so dass es möglich ist, nach Inhalten ausgewählter Felder zu suchen. Neben feldspezifischen Indizes gibt es in Retrievalsystemen meist noch einen Basic Index für die sogenannte Freie Suche, in dem mehrere Felder, vorrangig die inhaltsrelevanten wie Titel und Abstract, ausgewertet sind¹⁹.

Ein Datenfeld kann bei der Indexierung wort- oder phraseninvertiert werden²⁰. Bei der Wortinvertierung wird jedes Wort einzeln als Suchbegriff in den Index aufgenommen, bei der Phraseninvertierung das ganze Datenfeld als eine Einheit.

1.4 Suche in Datenbanken

Damit ein Benutzer auf eine Datenbank zugreifen kann, muss es eine Schnittstelle zwischen beiden Komponenten geben. Es wurde bereits gesagt, dass es sich bei einem Benutzer um einen Menschen, aber auch um eine Maschine handeln kann. Schnittstellen für Maschinen werden in Kapitel 5 behandelt. In diesem Kapitel geht es um Schnittstellen für menschliche Benutzer. Heute sind dies meist grafische Oberflächen. Bei einer Recherche kommt ein Benutzer mit verschiedenen Oberflächen in Berührung. Es gibt meist eine Oberfläche für die Suche, eine für die Kurzanzeige und eine für die Vollanzeige. Diese drei Oberflächen werden nun mit ihren Funktionen vorgestellt. Dazwischen werden Retrievalfunktionen in einem eigenen Abschnitt behandelt.

1.4.1 Oberfläche für die Suche

Eine Suchoberfläche dient dazu, Suchanfragen eines Benutzers entgegenzunehmen. Es werden hier zwei Arten der Suche unterschieden: die vom Informationsbedürfnis eines Benutzers ausgehende Suche und die vom Datenbestand ausgehende. In dieser Arbeit ist bei der Verwendung des Begriffs „Suche“ in der Regel an die Art der Suche gedacht, die vom Informationsbedürfnis eines Benutzers ausgeht. Bei dieser wird einem Benutzer eine Suchmaske zur Verfügung gestellt, in die er Anfragen eingeben kann. Grafische Benutzeroberflächen stellen dafür Eingabefelder und Auswahlmenüs

¹⁹ Vgl. Das Dateikonzept in Retrievalsystemen. Web-Lehrsammlung HU-Berlin. Online: <http://www.ib.huberlin.de/~is/Grundlagen/Dateikonzept.htm> [Stand: 01.05.2002, Abrufdatum: 04.09.2003, Datei: ir_dateien.htm]

²⁰ Vgl. Das Prinzip der Invertierung. Web-Lehrsammlung HU-Berlin. Online: <http://www.ib.huberlin.de/~is/Grundlagen/Invertierung.htm> [Stand: 01.05.2002, Abrufdatum: 04.09.2003, Datei: ir_invertierung.htm]

zur Verfügung. Es gibt in manchen Systemen auch die Möglichkeit, Suchanfragen mit Hilfe einer Kommandosprache einzugeben. Eine solche Sprache stellt umfangreiche Suchkommandos und Retrievalfunktionen (s. 1.4.2) zur Verfügung und ist somit ein sehr mächtiges Suchinstrument. Eingesetzte Anfragesprachen sind zum Beispiel Messenger, DIALOG oder grips-Open²¹.

Wenn eine Suchmaske mit Eingabefeldern arbeitet, stellt sich die Frage, wie viele Felder sie als Sucheinstiege anbietet. Der allgemeinste Einstieg ist ein einziges feldübergreifendes Suchfeld. Bei der feldspezifischen Suche werden einzelne Datenfelder getrennt suchbar gemacht. Meist bietet ein System verschiedene Suchmodi, zum Beispiel als Standard- und Experten-suche bezeichnet, an, so dass sowohl eine sehr differenzierte Suche möglich ist als auch eine allgemeinere. Die Suchmodi unterscheiden sich nicht nur in zur Verfügung stehenden Sucheinstiegen, sondern auch in dem Angebot an Retrievalfunktionen (s. 1.4.2).

Die vom Datenbestand ausgehende Suche bietet einem Benutzer an, in den vorhandenen Datenbeständen zu „stöbern“, was als „Browsing“ bezeichnet wird. Die Datenbestände sind mit Hilfe eines Klassifikationssystems angeordnet²². So kann sich ein Benutzer zwischen systematisch geordneten Dokumenten oder Dokumentnachweisen hin- und herbewegen.

1.4.2 Retrievalfunktionen

In einem Information Retrieval System werden viele Funktionen angeboten, die eine Recherche erleichtern und Suchergebnisse verbessern können²³. Maßstab für die Bewertung von Rechercheergebnissen sind die beiden Werte Recall und Precision²⁴. Beide Begriffe teilen einen Bestand je nach Suchanfrage in relevante und irrelevante Treffer ein. Recall gibt an, wie viele von allen relevanten Treffern, die verfügbar sind, gefunden wurden, Precision geht nur von der Trefferliste aus und misst, wie viele relevante Treffer darunter sind. Retrievalfunktionen dienen dazu, entweder Recall oder Precision zu verbessern. Einige wichtige werden hier nun aufgezählt.

Wichtige Verknüpfungsmechanismen sind die drei Booleschen Operatoren UND, ODER, NICHT. Suchwörter, die durch UND verbunden sind, müssen alle in einem Datensatz enthalten sein, bei ODER reicht es, wenn eines der Suchwörter vorhanden ist. NICHT schließt Wörter aus. Angaben über die Position bestimmter Wörter zueinander sind durch Abstandsoperatoren möglich.

Durch Trunkierung oder Maskierung wird die Treffermenge erhöht indem Suchwörter nicht exakt eingegeben werden müssen, sondern

²¹ Messenger wird von dem Host STN eingesetzt, DIALOG von dem gleichnamigen Host und grips von dem DIMDI (s. Kapitel 6.5)

²² Vgl. Stock, Wolfgang G: Checkliste für Retrievalsysteme: Qualitätskriterien von Suchmaschinen. In: Password (2000) 5, S. 23

²³ Vgl. Kowalski, Gerald: Information retrieval systems: theory and implementation. Boston, Kluwer Academic Publishers, 1997, S. 25ff.

²⁴ Vgl. ebd., S. 4ff.

unbekannte Zeichen durch einen Platzhalter ersetzt werden können. Bei Unsicherheit über Suchwörter ist auch das Aufblättern von Indizes eine Hilfe oder der Blick in einen Thesaurus. Aus Indizes und Listen können Suchbegriffe übernommen werden.

Weitere Retrievalfunktionen sind relevance ranking (s. 1.4.3) und relevance feedback. Letzteres kann jedoch erst nach einer ersten durchgeführten Suche eingesetzt werden.

1.4.3 Kurzanzeige

Nach einer Suche wird dem Benutzer in der Regel mit der Kurzanzeige das Rechercheergebnis mitgeteilt. Für jeden gefundenen Datenbestand erscheint ein stellvertretender kurzer Eintrag, in dem für die Identifikation eines Datenbestandes wesentliche Merkmale aufgeführt sind und der mit dem vollständigen Datensatz verknüpft ist.

Die Kurzanzeige kann verschiedene Funktionen anbieten. Im einfachsten Fall stellt sie lediglich eine Ergebnisliste dar, die vom Benutzer durchgeblättert werden kann.

Darüber hinaus können die Einträge nach bestimmten Kriterien sortiert sein. Oft kann der Benutzer selber Kriterien für die Sortierung angeben, wie zum Beispiel das Erscheinungsjahr eines Dokuments. Beim relevance ranking werden Treffer nach der Relevanz angeordnet. Eine weitere Funktion der Trefferliste kann zum Beispiel die Gruppierung von Treffern sein oder das Angebot, in der Trefferliste zu suchen²⁵.

1.4.4 Vollanzeige

Von der Kurzanzeige aus gelangt der Benutzer in der Regel auf die Vollanzeige. Dort wird ein Datenbestand vollständig angezeigt. Es ist in manchen Systemen, wie zum Beispiel im KVK²⁶, möglich, verschiedene Datenformate für die Vollanzeige zu wählen. Ein Datenformat ist eine Codierung für einen Datensatz. Das bedeutet, dass bestimmten Datenfeldern bestimmte Erkennungscode zugeteilt sind. Für die elektronische Übertragung von Datensätzen gibt es Standarddatenformate. Im Bereich des Bibliothekswesens sind dies zum Beispiel das Maschinelle Austauschformat für Bibliotheken (MAB) und MARC (Machine-Readable Cataloging)²⁷. Ein allgemeinerer internationaler Codierungsstandard für das WWW ist Dublin Core (DC). Dieser besitzt ursprünglich 15 Datenelemente zur Beschreibung von Dokumenten. Ein relativ neues Format für das Übersenden bibliografischer Daten in einer URL ist die OpenURL oder der Standard ZING/SRU. Beide werden in Kapitel 5 näher beschrieben.

²⁵ Vgl. Heinisch, Christian: Zur Usability von anbieterunabhängigen Metasuchoberflächen für die Fachinformationsrecherche im Invisible Web: Aufgabenstellung und Realisierung. In ABI-Technik 21 (2001) 4, S. 316f.

²⁶ S. Kapitel 6.1

²⁷ Vgl. Eversberg, Bernhard: Was sind und was sollen bibliothekarische Datenformate? Online: <http://www.zdb-opac.de> [Stand: 25.1.1999, Abrufdatum: 18.10.2003, Datei: format.htm]

2 FUNKTIONEN VERTEILTER INFORMATIONSSYSTEME

Nachdem im vorangegangenen Kapitel einzelne Datenquellen betrachtet wurden, wenden wir uns nun verteilten Informationssystemen zu. In diesem Kapitel werden die Aufgaben dieser Systeme behandelt, und zwar aus der Sicht des Benutzers. So stehen Funktionen, die mit Suchen und Finden zusammenhängen, im Vordergrund, im Gegensatz zu Funktionen, die für Systemadministratoren wichtig sind, wie zum Beispiel die Ausgabe von Benutzerstatistiken. Ebenfalls nicht eingegangen wird auf technische Funktionen, wie zum Beispiel die Realisierung des gleichzeitigen Zugriffs mehrerer Benutzer auf einen Broker.

Da es mit dem "MIA Functional Model"²⁸ (MODELS Information Architecture Functional Model) ein Modell gibt, das die Funktionen von verteilten Informationssystemen aus Benutzersicht aufführt, wurde dieses als Grundlage für die Ausführungen in diesem Kapitel genutzt. Das MIA Functional Model wurde im Jahre 1999 im Rahmen einer Studie von einer Projektgruppe mit dem Namen MODELS (Moving to Distributed Environments for Library Services) entwickelt²⁹. Das Projekt ging von UKOLN aus, einer bibliothekarischen Einrichtung, die im Bereich von digitaler Information Dienstleistungen erbringt und Forschung betreibt³⁰.

Als vier grundlegende Funktionen nennt das Modell "discover", "locate", "request" und "deliver", die mit „suchen“, „lokalisieren“, „anfordern“ und „liefern“ übersetzt werden. Alle vier werden als Funktionen in diesem Kapitel behandelt. Darüber hinaus werden hier die Funktionen „Zugriff kontrollieren“ ("authorize"), „Benutzer erkennen“ ("authenticate"), „Bereitstellen einer gemeinsamen Rechercheoberfläche“ sowie „zusätzliche Funktionen“ vorgestellt. Die beiden erstgenannten werden im MIA Functional Model erwähnt, wenn sie auch nicht zu den vier Grundfunktionen gehören³¹. Die Bereitstellung einer gemeinsamen Rechercheoberfläche ist vermutlich deshalb nicht in dem Modell mit aufgeführt, weil sie als Grundvoraussetzung betrachtet wurde und weniger als eine Funktion.

Nicht jedes System bietet alle Funktionen an³², nur die Suche und das Bereitstellen der gemeinsamen Rechercheoberfläche sind unerlässlich. In

²⁸ Gardner, Tracy; Paul Miller und Rosemary Russell: MIA Functional Model, version 0.3
Online: <http://www.ukoln.ac.uk/dlis/models/requirements/func/> [Stand: 13.09.1999, Abrufdatum: 07.10.2003, Datei: mia_functional.htm]

²⁹ Ebd.

³⁰ Vgl. Keane, Shirley: UKOLN: mission statement. Online: <http://www.ukoln.ac.uk/ukoln/mission/>
[Stand: 24.4.2003, Abrufdatum: 17.10.2003, Datei: ukoln_1.htm]

³¹ In neueren Anwendungen des MIA stehen beide Funktionen auf einer Stufe mit den vier Grundfunktionen.
(Vgl. Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment)

³² Vgl. Dempsey, Lorcan; Rosemary Russell und Robin Murray: A utopian place of criticism: brokering access to network information, S. 44f.

der Praxis lassen sich die genannten Funktionen teilweise schwer trennen, da sie zum Teil ineinander übergehen können.

In Anlehnung an den Sprachgebrauch in dem Modell wird der weiter gefasste Begriff „Ressource“ anstatt „Dokument“ verwendet.

2.1 Bereitstellen einer gemeinsamen Rechercheoberfläche

Eine einheitliche, gemeinsame Rechercheoberfläche ist eine wesentliche Voraussetzung dafür, dass ein Benutzer auf mehrere verschiedene Datenbestände von einem Einstiegspunkt aus zugreifen kann. In den meisten Fällen bieten verteilte Informationssysteme eine grafische Benutzeroberfläche an. Für die Suche, die Kurzanzeige und die Vollanzeige gibt es unterschiedliche Oberflächen.

Ein verteiltes Informationssystem kann sowohl für die Suche als auch für die Kurz- und Vollanzeige eine gemeinsame Oberfläche anbieten. Es ist jedoch auch möglich, für die Kurz- oder Vollanzeige Originaloberflächen der einzelnen Datenquellen mit einzubeziehen.

Für die Vollanzeige von Datenbeständen verlinken viele verteilte Informationssysteme auf die Originaloberflächen der jeweiligen Provider. Der Vorteil bei einem Wechsel auf die Originaloberflächen liegt darin, dass alle Dienste des entsprechenden Providers genutzt werden können. Nachteilig ist, dass Dienste, die ein verteiltes Informationssystem anbietet, durch ein Verlassen nicht weiter genutzt werden können. Außerdem wird eine Recherche durch die Systemwechsel unübersichtlich und uneinheitlich. Ein weiterer Nachteil, der jedoch in erster Linie für kommerzielle Systeme gilt, ist der Verlust der Kundenbindung.

In einem verteilten Informationssystem muss es eine gemeinsame Kurzanzeige geben. Es ist jedoch möglich, dass Originaloberflächen auch in der Kurzanzeige eine Rolle spielen. Der KVK bindet, je nach gewählter Form der Anzeige, Suchergebnisse von Providern in dem Originallayout der Datenquelle ein³³. In diesem Fall befindet sich der Benutzer bei der Kurzanzeige in dem Angebot des Brokers, das providerspezifische Layout wird jedoch sichtbar.

Nicht jedes verteilte Informationssystem weist die Ergebnisse einer Suche in einer gemeinsamen Kurzanzeige nach. Die KOBV-Suchmaschine³⁴ zum Beispiel liefert als gemeinsame Kurzanzeige lediglich eine Liste, in der die Anzahl der Treffer pro Datenquelle aufgeführt sind. Um die einzelnen Treffer identifizieren zu können, kann pro Datenquelle eine separate Kurztrefferliste aufgerufen werden.

³³ S. Kapitel 6.1

³⁴ S. Kapitel 6.3

2.2 Suchen

Die Funktion „suchen“ bzw. „discover“ ist die Kernfunktion eines Broker-Systems. Sie erfüllt die Aufgabe, vom Benutzer gesuchte Datenbestände zu finden und auf diese durch eine Kurzanzeige hinzuweisen³⁵. Die Vollanzeige gehört schon zu der Funktion „liefern“ (2.4). Da für das tiefere Verständnis der Suchabläufe die in Kapitel 3 vorgestellte Architektur eines Broker-Systems bekannt sein muss, behandelt erst Kapitel 4 die Funktion „discover“. Als Schwerpunkt dieser Arbeit wird die Suche damit ausführlicher behandelt als alle anderen Funktionen.

2.3 Lokalisieren

Wenn mehrere inhaltlich gleiche oder fast gleiche Ressourcen vorliegen, die nicht unmittelbar angezeigt werden können, zum Beispiel weil sie kostenpflichtig sind oder weil sie nicht elektronisch vorliegen, ist es notwendig, eine Ressource bevor sie angefordert wird („request“) mit Hilfe des Benutzers eindeutig zu identifizieren. Dieser Vorgang wird als „Lokalisieren“ bzw. als „locate“ bezeichnet.

Der Begriff „locate“ ist geprägt durch die Welt der Bibliotheken, in denen Ressourcen zum Großteil gedruckt vorliegen und nur ein elektronischer Nachweis über diese in Datenbanken abgelegt ist. Da ein Buch zum Beispiel in verschiedenen Bibliotheken vorliegen kann, ist es bei einer Bestellung wichtig, genau den Standort zu bestimmen, aus dem bestellt werden soll. Liegt eine Ressource digital in einer Datenbank vor, spielt das Kriterium des Standortes bzw. des Speicherplatzes in der Regel keine Rolle, da es keinen Unterschied macht, ob eine digitale Ressource von einem Server aus Japan oder aus Deutschland angefordert wird. Aus diesem Grund wird der Begriff „locate“ zum Teil durch den allgemeineren Begriff „detail“ ersetzt³⁶. Dieser umfasst nicht nur den Unterschied des verschiedenen Standortes, sondern auch Kriterien wie verschiedene Ausgaben, Sprachen, Formate und Bedingungen, die durch einen Provider an eine Ressource geknüpft sind.

Oft sind der Standort und weitere an eine Ressource geknüpfte Bedingungen schon in der Kurzanzeige sichtbar, was dafür spricht, die Lokalisierung teilweise auch der Funktion „discover“ zuzurechnen³⁷. Bei besonderen Bestellbedingungen ist „locate“ jedoch vielmehr als Anfang der Anforderungsphase zu sehen und kann somit auch zum Prozess der Bestellung gerechnet werden³⁸. Hier wird die Funktion entsprechend dem MIA Functional Model bewusst als eine eigenständige vorgestellt.

³⁵ Vgl. Powell, Andy und Liz Lyon: JISC Information Environment Architecture: Functional Model
Online: <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/functional-model/> [Stand: 09.04.2003,
Abrufdatum: 08.10.2003, Datei: jisc_1.htm]

³⁶ Vgl. ebd. sowie Gardner, Tracy; Paul Miller und Rosemary Russell: MIA Functional Model

³⁷ Vgl. Powell, Andy und Liz Lyon: JISC Information Environment Architecture: Functional Model

³⁸ Vgl. ebd.

Es wurden Tools, wie zum Beispiel SFX, LinkFinderPlus oder Link-Source entwickelt, die die Funktion „locate“ anbieten. In aller Kürze wird hier die Funktionsweise von SFX, das von der Firma ExLibris entwickelt wurde, exemplarisch vorgestellt³⁹. SFX ist ein Mechanismus, der folgendermaßen funktioniert: von einer in der Kurz- oder Vollanzeige dargestellten Ressource aus kann ein Link (SFX-Button) betätigt werden, der auslöst, dass ein zusätzliches Fenster geöffnet wird. Dieses Fenster bietet verschiedene Datenquellen an, die voraussichtlich Zugang zu der Ressource bieten, von der aus der SFX-Button betätigt wurde. Das Fenster ist kontext-sensitiv, also je nach Standort oder Benutzer verschieden. Informationen darüber, auf welche Datenquellen welcher Server oder Benutzer zugreifen darf, liegen in einer von ExLibris gepflegten Datenbank vor, der KnowledgeBase⁴⁰. Die Daten, die durch die Betätigung des SFX-Buttons an die KnowledgeBase weitergegeben werden, sind in einem bestimmten Format, der OpenURL, codiert und beinhalten eine eindeutige Identifikation des Datensatzes und des Servers, von dem die Anfrage kommt. In diesem System spielen die Funktionen „authenticate“ und „authorize“, die in Abschnitt 3.5 vorgestellt werden, ebenfalls eine Rolle.

Ein nicht standardisiertes Verfahren für die Lokalisierung findet zum Beispiel in der Digitalen Bibliothek NRW mit Hilfe einer Verfügbarkeitsfunktion statt, die in Kapitel 6.4 vorgestellt wird.

2.4 Anfordern, Liefern

Die beiden letzten Grundfunktionen der MIA Functional Architecture „anfordern“ („request“) und „liefern“ („deliver“) werden hier zusammengefasst, da sie sich schwer getrennt betrachten lassen. „Request“ besagt, dass eine Ressource vom Benutzer angefordert wird, „deliver“, dass diese vollständig geliefert wird. Der Begriff „deliver“ ist wiederum auf die Printwelt ausgerichtet, in neueren Anwendungen⁴¹ ist er durch den Begriff „access“ erweitert worden und auch im MIA Functional Model selbst ist von „access“ die Rede.

Unter „Lieferung“ kann zum einen verstanden werden, dass nach einer Kurzanzeige die Vollanzeige erscheint, unabhängig davon, aus was die Vollanzeige besteht, ob es sich um Metadaten oder um eine vollständige Ressource handelt. In dem MIA Functional Model ist unter „deliver“ jedoch vielmehr die Lieferung einer bestimmten Ressource zu verstehen, sei es in digitaler oder gedruckter Form. In diesem Fall kann ein verteiltes Informati-

³⁹ Die Angaben in diesem Abschnitt beruhen auf der Erkundung des kostenlosen Testzugangs zu SFX von Anfang September 2003 („Testzugang zu SFX“. Online: <http://www.sfxit.com/trysfx/index.html> [Stand: 6.2.2003, Abrufdatum: 17.10.2003, Datei: sfx_1.htm]) sowie auf: SFX context sensitive linking: kontextsensitives Linking mit einem unabhängigen Linkserver. Hrsg.: ExLibris. Ex Libris 2003, in der Anlage enthalten

⁴⁰ Vgl. SFX context sensitive linking: kontextsensitives Linking mit einem unabhängigen Linkserver

⁴¹ zum Beispiel in der JISC-Architecture (Powell, Andy und Liz Lyon: JISC Information Environment Architecture: Functional Model) oder in der DNER-Technical Architecture (Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment)

onssystem verschiedene Mechanismen anbieten, die dem Benutzer zum Beispiel durch Fernleihe oder elektronische Zeitschriftenlieferdienste Zugang zu einer Ressource verschaffen. Eine Lieferung findet dann entweder innerhalb oder außerhalb eines Broker-Systems statt.

2.5 Benutzer erkennen, Zugriff kontrollieren

Wenn ein Benutzer identifiziert wird ("authenticate"), ist es für ein System möglich, auf ein Benutzerprofil zurückzugreifen, sofern eines vorliegt. Dieses kann entweder vom Benutzer oder von Administratoren erstellt werden oder Programme können es auf Grund von Beobachtungen generieren. Es kann auch eine Mischung aus beiden Komponenten vorliegen. Mit Hilfe des Benutzerprofils ist es möglich, an Personen angepasste Arbeitsumgebungen ("landscapes") zu schaffen. Eine solche Umgebung zeichnet sich zum einen durch formale Einstellungen aus, wie zum Beispiel durch das Layout, zum anderen durch inhaltliche Eingrenzungen wie die Aufführung nur der Datenquellen, auf die der Benutzer rechtlich Zugriff hat ("authorize"). Eine Identifizierung kann auf Ebene der Maschinen durch Erkennen der IP-Adresse des Benutzers geschehen oder durch das Einloggen des Benutzers selbst⁴².

Die Verwendung von IP-Adressen ist hilfreich, wenn es sich um Computer einer Institution, zum Beispiel einer Bibliothek, handelt, die nur Personen benutzen können, die das Recht haben, auf Informationen zuzugreifen, die darüber angeboten werden. Eine Erkennung auf der Ebene einzelner Personen durch Einloggen ist notwendig, wenn ein Informationssystem über das WWW zugänglich ist.

2.6 Zusätzliche Funktionen

Unter zusätzlichen Funktionen werden hier Dienste zusammengefasst, die verteilte Informationssysteme zusätzlich anbieten können. Insbesondere im Bereich von Internet-Portalen spielen diese zusätzlichen Dienste eine wichtige Rolle. Es gehört dazu zum Beispiel die regelmäßige Zusendung von Newslettern. Darüber hinaus werden zum Teil Dienste angeboten, die die Kundenbindung erhöhen sollen. Dafür müssen Angebote zur Verfügung gestellt werden, die ein Kunde häufig nutzt, wie zum Beispiel Adressbücher oder Kalender⁴³.

⁴² Vgl. Powell, Andy und Liz Lyon: JISC Information Environment Architecture: Functional Model

⁴³ Vgl. Rösch, Hermann: Portale in Internet, Betrieb und Wissenschaft: Marktplatz und Instrument des Kommunikations- und Wissensmanagements

3 ARCHITEKTUR VERTEILTER INFORMATIONSSYSTEME

Auch für die Architektur verteilter Informationssysteme gibt es ein Modell (MIA Logical Architecture). In diesem werden die Abläufe, die in einem verteilten Informationssystem stattfinden, verschiedenen Schichten zugeordnet und somit in eine Struktur gebracht. Nach dieser Übersicht werden die verschiedenen Möglichkeiten der Datenhaltung in verteilten Informationssystemen vorgestellt, wobei auf zwei Möglichkeiten, und zwar auf die verteilte und die durch Harvesting zum Teil zentralisierte Datenhaltung, näher eingegangen wird.

3.1 Modell für die Architektur: MIA Logical Architecture

Neben dem bereits in Kapitel 2 erwähnten MIA Functional Model entstand ebenfalls im Rahmen der MODELS-Initiative (Moving to Distributed Environments for Library Services) von UKOLN die MIA Logical Architecture⁴⁴. Es handelt sich um ein Modell, das die Abläufe bei einer Recherche in verteilten Informationssystemen fünf Schichten zuordnet.

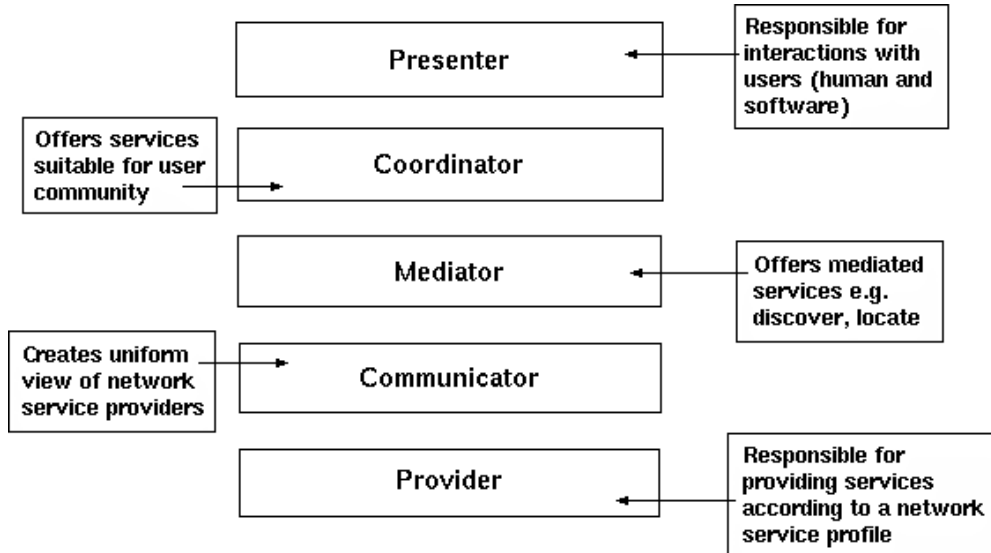


Abb. 3: Darstellung der MIA Logical Architecture

Quelle: Gardner, Tracy; Paul Miller und Rosemary Russell: The MIA Logical Architecture, version 0.3. Online: <http://www.ukoln.ac.uk/dlis/models/requirements/arch/> [Stand: 13.09.1999, Abrufdatum: 08.10.2003, Datei: mia_logical.htm]

⁴⁴ Vgl. Gardner, Tracy; Paul Miller und Rosemary Russell: The MIA Logical Architecture, version 0.3. Online: <http://www.ukoln.ac.uk/dlis/models/requirements/arch/> [Stand: 13.09.1999, Abrufdatum: 08.10.2003, Datei: mia_logical.htm]

Die einzelnen Schichten heißen der Reihenfolge nach Presenter, Coordinator, Mediator, Communicator und Provider. "Presenter" ist die Schicht, die dem Benutzer am nächsten ist und die Schnittstelle zu diesem herstellt, die Provider-Schicht ist vom Benutzer am weitesten entfernte.

Presenter:

In dieser Schicht findet die direkte Kommunikation mit dem Benutzer statt. Der Benutzer kann ein Mensch sein oder ein System. Wenn es sich um einen Menschen handelt, muss eine Benutzeroberfläche bereitgestellt werden, zum Beispiel eine grafische. Somit ist diese Schicht für die in Kapitel 2 zu Beginn vorgestellte Funktion „Bereitstellen einer gemeinsamen Rechercheoberfläche“ verantwortlich. Handelt es sich bei dem Benutzer um ein System, müssen für die Verständigung entsprechende Schnittstellen zur Verfügung stehen. Es werden sowohl Eingaben des Benutzers angenommen und an die nächste Schicht weitergeleitet als auch Antworten ausgegeben.

Coordinator:

Der zweiten Schicht zwischen dem Presenter und dem Mediator ist die Aufgabe zugeordnet, eine benutzerspezifische Umgebung zu gestalten. Dazu enthält sie Angaben über eine laufende Sitzung und Benutzerprofile. Sie leitet neben der Anfrage des Benutzers zusätzliche für die Suche wichtige benutzerspezifische oder sitzungsabhängige Angaben an den Mediator weiter und bearbeitet das Suchergebnis dementsprechend. Diese Schicht ist damit an der Erfüllung der Funktionen „Benutzer erkennen“ und „Zugriff kontrollieren“ beteiligt.

Mediator:

Die dritte Schicht ist dafür verantwortlich, mit Anfragen an Provider umzugehen. Von ihr aus wird die Kommunikation mit diesen gestartet. Der Mediator entscheidet, welche Provider angesprochen werden und sammelt die Ergebnisse ein. Er ist das Bindeglied für die verschiedenen einzelnen Provider. Dieser Schicht werden die vier Grundfunktionen „suchen“, „lokalisieren“, „anfordern“ und „liefern“ zugeordnet.

Communicator:

Diese Schicht ist für die Übersetzung und Übertragung von Anfragen und Antworten zwischen dem Mediator und den Providern verantwortlich. Dafür sind detaillierte Kenntnisse über den Umgang mit den einzelnen Providern notwendig. Im Communicator werden verschiedene Protokolle eingesetzt.

Provider:

Dieser untersten Schicht werden die einzelnen Systeme zugeordnet, die Zugang zu Informationen bieten. Über die Architektur dieser einzelnen Systeme sagt die MIA Logical Architecture nichts aus, da es in der Verantwortung jedes einzelnen Datenbankanbieters steht, wie er seine Daten zugänglich macht. Die Provider müssen Schnittstellen anbieten, über die auf ihr Angebot zugegriffen werden kann.

Die MIA Logical Architecture macht keine Aussagen darüber, wie die einzelnen Aufgaben technisch realisiert werden, sondern stellt nur einen Rahmen dar. In der Veröffentlichung des Modells heißt es dazu: "MIA aims to be a generic architecture supporting multiple architectural choices"⁴⁵. Die MIA Logical Architecture ist Ausgangspunkt für die Entwicklung der Architektur vieler Systeme ist⁴⁶. Je nachdem, ob eine Architektur auf dem Modell beruht, lassen sich bestehende Systeme sehr gut oder weniger gut in den fünf MIA-Schichten darstellen⁴⁷.

Bildet man die drei in der Einleitung vorgestellten Komponenten Benutzer, Broker-System und Provider auf das Modell der MIA Logical Architecture ab, so können die drei mittleren Schichten, der Coordinator, Mediator und Communicator, als Broker bezeichnet werden, der Presenter gehört zur Ebene des Benutzers und die Provider-Schicht entspricht den Providern⁴⁸.

3.2 Datenhaltung

Wenn bei einer Recherche die Bestände mehrerer Provider berücksichtigt werden sollen, gibt es die bereits in der Einleitung erwähnten beiden Möglichkeiten, die verteilten Datenbestände entweder physisch zusammenzuführen, das heißt, sie in eine zentrale Datenbank hineinzukopieren, oder virtuell mit Hilfe eines Broker-Systems.

Ein typisches Beispiel für die zentrale Datenhaltung sind viele Bibliotheksverbundkataloge in Deutschland⁴⁹. Sie fassen Datenbestände verschiedener Kataloge zusammen, so dass bei einer Suche in der zentralen Datenbank die einzelnen Kataloge nicht eingebunden sind. Eine dezentrale Datenhaltung liegt zum Beispiel bei ZACK oder vascoda vor, die in Kapitel 6 vorgestellt werden. Dort wird bei einer Recherche direkt auf die einzelnen

⁴⁵ Gardner, Tracy; Miller, Paul und Rosemary Russell: The MIA Logical Architecture

⁴⁶ zum Beispiel für Agora (Realizing the Hybrid Library. Stephen Pinfield; Jonathan Eaton; Catherine Edwards [u.a.]. In: D-Lib Magazine 4 (1998) 10. Online: <http://www.dlib.org/dlib/october98/10pinfield.html> [Stand: Oktober 1998, Abrufdatum: 08.10.2003, Datei: agora.htm]

⁴⁷ Vgl. Renardus: project deliverable D1.1: Evaluation report of existing broker models in related projects. Michael Day; Anders Ardö, Matthew J. Dovey [u.a.]. Online: http://www.renardus.org/about_us/deliverables/d1_1/D1_1_final.pdf [Stand: 27.04.2000, Abrufdatum: 08.10.2003, Datei: renardus_d1_1.pdf]

⁴⁸ Vgl. Ianelle, Renato: Technical Review of RDNC Subject Gateway Services. Online: <http://www.rdn.ac.uk/publications/studies/technical-review/report.pdf> [Stand: 10.11.1999, Abrufdatum: 13.10.2003, Datei: rdnc.pdf], S. 16

⁴⁹ Nicht dazu gehört zum Beispiel der KOBV (s. Kapitel 6.3).

heterogenen Provider zugegriffen. Neben den für eine Architektur typischen Anwendungen gibt es Misch- und Sonderformen. Eine Sonderform stellen Hosts dar. Diese greifen in der Regel auf heterogene Datenbanken zu, die sie jedoch zentral auf ihren Servern vorliegen haben und deren Inhalte angepasst worden sind. Da es in dieser Arbeit um Systeme geht, die in eine Recherche die unabhängigen, selbstständigen Provider mit einbeziehen, wird die Sonderform der Hosts an dieser Stelle nicht weiter betrachtet. Auf Grund ihrer geschichtlichen Bedeutung und ihrer Wichtigkeit im Informations- und Dokumentationswesen wird jedoch in Kapitel 6 mit der Vorstellung des DIMDI noch weiter auf sie eingegangen.

Eine Mischform zwischen zentraler und dezentraler Datenhaltung ist das Anlegen datenbankübergreifender, zentraler Indizes für die Suche, wobei der Zugriff auf die vollständigen Datensätze weiterhin dezentral erfolgt. Da bei dieser Mischform die einzelnen Provider bei einer Recherche miteinbezogen werden, wird dieses Verfahren neben der rein dezentralen Datenhaltung näher betrachtet. Es wird zunächst die Suche in dezentralen Umgebungen, die sogenannte verteilte Suche, vorgestellt, anschließend die zentrale Suche. Im letzten Abschnitt werden beide Verfahren gegenübergestellt und verglichen.

3.2.1 Verteilte Suche

Die verteilte Suche, auch als cross search bezeichnet, ist eine parallele Suche über mehrere Datenbanken⁵⁰. Das bedeutet, dass ein System, das die verteilte Suche durchführt, auf die Indizes der jeweiligen einzelnen Provider zugreift. Es handelt sich also um ein dezentrales Modell⁵¹. Damit eine solche Suche erfolgreich sein kann, müssen auf der inhaltlichen und technischen Ebene viele Anpassungen getroffen werden, die in den folgenden beiden Kapiteln näher betrachtet werden.

Wichtig ist für die verteilte Suche, dass ein Mechanismus zur Verfügung steht, der die Suche in den einzelnen Datenquellen beendet. Dies geschieht in der Regel mit Hilfe eines sogenannten "Timeouts". Das bedeutet, dass eine bestimmte Zeit, zum Beispiel 20 Sekunden, eingestellt ist, nach der ein Broker-System eine Suche in einer Datenquelle abbricht, falls sie noch nicht von alleine abgeschlossen wurde. Für das Suchergebnis kann ein Timeout jedoch negative Auswirkungen haben, da die Trefferliste unvollständig ist. Um den Benutzer durch ein Ergebnis nicht zu irritieren, ist es notwendig, ihm deutlich zu machen, wenn Datenquellen bei der Suche nicht berücksichtigt werden konnten.

⁵⁰ Vgl. Schneider, Wolfram: Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls: Diplomarbeit am Fachbereich Informatik der Technischen Universität Berlin. Berlin 1999, S. 3

⁵¹ Vgl. Jost, Michael: EULER-TAKEUP: Report on Sustainable Service and Business Model. Online: <http://www.emis.de/projects/EULER/takeup/d2.html> [Stand: 28.11.2002, Abrufdatum: 8.10.2003, Datei: euler_1.htm]

Trefferlisten werden bei der verteilten Suche oft kontinuierlich aufgebaut, je nachdem, in welcher Reihenfolge die Antworten der Provider bei dem Broker ankommen.

Für die verteilte Suche gibt es ein Standardprotokoll mit dem Namen Z39.50. Neben diesem Standard gibt es weitere Realisierungsmöglichkeiten, auf die in Kapitel 5 eingegangen wird.

3.2.2 Zentrale Suche durch Harvesting

Bei der zentralen Suche, auch als zentrales Modell⁵² bezeichnet, liegt ein gemeinsamer Index vor, in dem gesucht wird. Genaugenommen liegt nicht nur ein einziger Index vor, sondern es werden für verschiedene Sucheinstiege feldspezifische und feldübergreifende, wort- und phraseninvertierte Indizes erstellt. Es ist hier bei dem zentralen Modell jedoch meist nur von „dem“ oder „einem zentralen Index“ die Rede, um die Funktionsweise zu verdeutlichen. Diese Bezeichnung umfasst alle einzelnen Indizes.

Für den Aufbau eines Index werden Metadaten von Datenbeständen der verschiedenen Provider zusammengeführt. Da Provider, die lediglich Metadaten enthalten, fast komplett in einen zentralen Index übernommen werden müssen, grenzt das Harvesting, lediglich auf die Datenhaltung bezogen, in diesen Fällen nah an eine zentrale Datenhaltung. Die KOBV-Suchmaschine ist ein gutes Beispiel dafür. In ihrem Index befinden sich jedoch nicht die kompletten Datensätze der Provider, sondern sie übernimmt nur suchrelevante Daten sowie Daten, die für die Dublettenkontrolle⁵³ wichtig sind und für die Anzeige eines Treffers in der Kurzanzeige⁵⁴. Bei elektronischen Volltexten, für die das Verfahren des Harvesting primär entwickelt wurde⁵⁵, ist dies anders.

Für die Gewinnung der Metadaten für die Indexierung gibt es drei verschiedene Möglichkeiten: Gathering, Harvesting oder Alerting⁵⁶. Gathering bedeutet, dass Indexdaten ohne Beteiligung des Systems, das die Daten enthält, gewonnen werden. Dies ist der Mechanismus, der oft für den Aufbau von Indizes im Internet angewendet wird. Beim Harvesting findet eine beidseitige Kooperation statt: Provider bieten strukturierte Metadaten an, die von einem Broker geholt werden. Werden Metadaten von einzelnen Providern an das System geschickt, das den Index generiert, so spricht man von Alerting. Bei der Generierung eines gemeinsamen Index aus vielen Daten-

⁵² Vgl. ebd.

⁵³ S. Kapitel 4.5.2

⁵⁴ Vgl. Der Kooperative Bibliotheksverbund Berlin Brandenburg. Martin Grötschel; Monika Kuberek; Stefan Lohrum [u. a.]. Online: <ftp://ftp.zib.de/pub/zib-publications/reports/SC-99-52.pdf> [Stand: 12/1999, Abrufdatum: 16.10.2003, Datei: kobv_1.pdf], S. 21

⁵⁵ Vgl. Lagoze, Carl und Herbert Van de Sompel: The Open Archives Initiative: building a low-barrier interoperability framework. Online <http://www.openarchives.org/documents/oai.pdf> [Abrufdatum: 8.10.2003, Datei: oai_1.pdf], S. 1

⁵⁶ Vgl. für diesen Abschnitt: Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment

banken spielt das Harvesting die herausragende Rolle, weshalb hier nur auf diese Methode eingegangen wird.

Metadaten müssen von den Providern in einem bestimmten vereinbarten Format zur Verfügung gestellt werden. Nachdem diese von den einzelnen Datenbanken durch Harvesting gewonnen wurden, können sie anschließend normalisiert, auf Dubletten kontrolliert, eventuell um weitere Angaben ergänzt und durch Indizes für die Suche zugänglich gemacht werden⁵⁷. Voraussetzung für dieses Modell ist zum einen, dass sich die Provider und das Broker-System auf ein einheitliches Protokoll und Format für die Übertragung der Metadaten verständigen. Zum anderen müssen die Metadaten eine eindeutige Identifikation des Datenbestandes, auf den sie verweisen, besitzen, damit ein Broker-System wenn nötig auf die entsprechenden Datenbestände zugreifen kann. Es gibt verschiedene Verfahren für die Vergabe von Identifikationen. Von besonderer Bedeutung sind die OAI-identifier, die im OAI-PMH-Protokoll beschrieben sind⁵⁸. Das OAI-PMH ist ein Standardprotokoll, das für das Harvesting entwickelt wurde. Es wird in Kapitel 5 vorgestellt.

In einem Broker-System muss eingestellt sein, wie oft Metadaten gewonnen werden und welche. Es können alle Metadaten eines Providers auf einmal angefordert werden ("bulk-harvesting"⁵⁹) oder durch selektives Harvesting nur einzelne. Vorteile beim Anfordern aller Metadaten sind, dass Anfragen nicht kompliziert sind und dass für die Übertragung nur eine Verbindung aufgebaut werden muss, so dass weder der Server noch der Provider durch viele Verbindungen belastet sind⁶⁰. Probleme kann es geben, wenn keine „Flow Control“⁶¹ eingesetzt wird, ein Mechanismus, der es ermöglicht, große Datenmengen zu übertragen. Ein weiteres Problem sind Übertragungsfehler, die beim bulk-Harvesting einen sehr viel größeren Schaden anrichten als beim selektiven Harvesting⁶².

Beim selektiven Harvesting werden verschiedene Kriterien bestimmt, nach denen einzelne Datensätze für das Harvesting herausgesucht werden. Ein Auswahlkriterium kann zum Beispiel ein „Datestamp“⁶³, also eine Zeitstempel, sein, über den die Aktualität eines Datensatzes ermittelt werden kann. Neben formalen Auswahlkriterien kann es auch inhaltliche geben. Voraussetzung dafür ist, wie auch bei dem Datestamp, dass entsprechende

⁵⁷ Vgl. Lynch, Clifford A.: Metadata Harvesting and the Open Archives Initiative. In: ARL Bimonthly Report 217 (2001). Online: <http://www.arl.org/newsltr/217/mhp.html> [Stand: August 2001, Abrufdatum: 8.10.2003, Datei: oai_2.htm]

⁵⁸ Vgl. The Open Archives Initiative Protocol for Metadata Harvesting, version 1.1. Online: <http://www.openarchives.org/OAI/1.1/openarchivesprotocol.htm> [Stand: 20.6.2001, Abrufdatum: 8.10.2003, Datei: oaipmh_1.htm]

⁵⁹ Vgl. Arc - An OAI Service Provider for Digital Library Federation. Xiaoming Liu; Kurt Maly; Mohammad Zu-bair [u.a.]. In: D-Lib Magazine 7 (2001) 4. Online: <http://www.dlib.org/dlib/april01/liu/04liu.html> [Stand: 4/2001, Abrufdatum: 8.10.2003, Datei: arc.htm]

⁶⁰ Vgl. ebd.

⁶¹ Vgl. The Open Archives Initiative Protocol for Metadata Harvesting

⁶² Vgl. Arc - An OAI Service Provider for Digital Library Federation

⁶³ Vgl. The Open Archives Initiative Protocol for Metadata Harvesting

einheitliche Angaben zu dem Auswahlkriterium in den Metadaten verfügbar sind. Sollen also die Metadaten zu Datensätzen, die das Fachgebiet „Bibliothekswesen“ umfassen, durch Harvesting gewonnen werden, so müssen alle betroffenen Datensätze eine einheitliche Angabe enthalten, die sie als Datensätze aus diesem Fachgebiet kennzeichnet. Ein sehr gezieltes Harvesting ermöglichen Benachrichtigungsmechanismen einzelner Provider, die dem Broker in regelmäßigen Abständen mitteilen, wenn Metadaten neu geholt oder gelöscht werden müssen.

Um für Broker den Aufwand zu verringern, kann hierarchisches Harvesting eingesetzt werden. Das bedeutet, dass beim Harvesting nicht alle Broker auf alle einzelnen Datenquellen zugreifen, sondern dass es zwischengeschaltete Systeme, sogenannte „Aggregators“⁶⁴ gibt, die Metadaten aus einigen Datenquellen enthalten und diese für andere Broker zur Verfügung stellen. In der Regel sind in einem Aggregator inhaltlich ähnliche Daten vereinigt.

3.2.3 Vergleich der verteilten Suche und der zentralen Suche durch Harvesting

Die verteilte Suche und die zentrale Suche durch Harvesting haben jeweils ihre Vor- und Nachteile. Beide Modelle werden nun in einigen wesentlichen Punkten einander gegenübergestellt und verglichen. Als Hilfe für den Leser sind in Tabelle 1 die nun folgenden Ausführungen verkürzt dargestellt.

Bei einer verteilten Suche können nur die Retrievalfunktionen angeboten werden, die die einzelnen Datenquellen zur Verfügung stellen. Da Provider jeweils verschiedene Retrievalfunktionen anbieten, wird ein gemeinsames Angebot bei der verteilten Suche oft nach dem Provider gerichtet, der die wenigsten Funktionen unterstützt. Darauf wird im folgenden Kapitel genauer eingegangen. Die Retrievalfunktionen beim Harvesting sind von den einzelnen Datenbanken unabhängig. Etwas Ähnliches gilt für den zentralen Index: dieser ist von den Indizes der einzelnen Provider unabhängig, so dass ein Broker nach Belieben verschiedene Indizes erstellen kann. Außerdem kann er, wie bereits oben gesagt, die Daten für den Index normieren und um zusätzliche Angaben anreichern. Dieser Punkt, der Feldinhalte betrifft, wird ebenfalls im folgenden Kapitel näher behandelt. Nachteilig an einem gemeinsamen Index ist, dass für einen Broker mit der Pflege und Erstellung eines gemeinsamen Index ein hoher Aufwand verbunden ist⁶⁵.

Bei der verteilten Suche wird auf die Indizes der jeweiligen Provider zugegriffen. Zwar ist ein speziell für einen Provider generierter Index sehr gut an ein System angepasst, aber dieser kann wegen der Vereinheitlichung in der verteilten Suche oft nicht genutzt werden. Der Broker ist in diesem Fall auf standardisiert vorliegende Metadaten in den Indizes angewiesen. Es

⁶⁴ Vgl. Powell, Andy und Liz Lyon: The JISC Information Environment and Web services. In: Ariadne 31 (2002). Online: <http://www.ariadne.ac.uk/issue31/information-environments/> [Stand: 04/2002, Abrufdatum: 15.10.2003, Datei: jisc_2.htm]

⁶⁵ Vgl. Jost, Michael: EULER-TAKEUP: Report on Sustainable Service and Business Model

wird deutlich, dass der Broker bei einem zentralen Index einen viel größeren Einfluss auf den Suchstandard hat als bei der verteilten Suche.

Kriterien	Verteilte Suche	Zentrale Suche durch Harvesting
Retrieval-funktionen	<ul style="list-style-type: none"> ▪ providerabhängig ▪ meist pro Provider verschieden 	<ul style="list-style-type: none"> ▪ vom Broker abhängig ▪ einheitlich
Index	<ul style="list-style-type: none"> ▪ providerabhängig ▪ keine Kontrolle durch Broker ▪ speziell an Datenbank angepasst ▪ kein Pflegeaufwand für Broker 	<ul style="list-style-type: none"> ▪ einheitlich durch Broker ▪ Kontrolle durch Broker ▪ weniger speziell angepasst ▪ Pflegeaufwand
Datenbasis (Feldinhalte)	<ul style="list-style-type: none"> ▪ providerabhängig ▪ z. T. einheitlich durch allg. Standards 	<ul style="list-style-type: none"> ▪ Normalisierung und Datenanreicherung möglich ▪ einheitliche Felder durch Harvesting
Verfügbarkeit	<ul style="list-style-type: none"> ▪ providerabhängig 	<ul style="list-style-type: none"> ▪ Kurzanzeige nur vom Broker abhängig (Nadelöhr) und vom erfolgreichen Harvesting ▪ Vollanzeige providerabhängig
Belastung Provider	<ul style="list-style-type: none"> ▪ viele unnötige Anfragen ▪ Überlastung bei Zugriff von Broker zu Broker 	<ul style="list-style-type: none"> ▪ gezielter Zugriff auf Datensätze ▪ Harvesting teilweise Belastung
Browsing	<ul style="list-style-type: none"> ▪ kompliziert zu implementieren 	<ul style="list-style-type: none"> ▪ leicht zu implementieren
Technische Realisierung	<ul style="list-style-type: none"> ▪ technisch aufwändiger 	<ul style="list-style-type: none"> ▪ technisch weniger aufwändig
Anzahl eingebundener Provider	<ul style="list-style-type: none"> ▪ abhängig von Netzverbindung, Anzahl und Größe der Daten 	<ul style="list-style-type: none"> ▪ abhängig von Kapazität des Brokers
Schnelligkeit der Suche	<ul style="list-style-type: none"> ▪ providerabhängig ▪ abhängig von Netzverbindung, Anzahl und Größe der Daten 	<ul style="list-style-type: none"> ▪ abhängig vom Broker
Aktualität	<ul style="list-style-type: none"> ▪ real time 	<ul style="list-style-type: none"> ▪ abhängig vom Harvesting
Copyright	<ul style="list-style-type: none"> ▪ wird nicht verletzt 	<ul style="list-style-type: none"> ▪ kann verletzt werden

Tabelle 1: Gegenüberstellung der verteilten Suche und der zentralen Suche durch Harvesting

Bei der zentralen Suche können die Bestände von Datenquellen durchsucht werden, auch wenn die Datenquellen zur Zeit der Suche nicht erreichbar sind⁶⁶. So findet in der Regel eine vollständigere Suche statt. Allerdings führt die Zentralisierung auch dazu, dass durch den Index ein einziges Nadelöhr für die Suche geschaffen wird. Ist der Index nicht erreichbar, kann keine Suche durchgeführt werden. Dazu kommt, dass ein Broker zwar nicht während einer Suche auf den Zugang zu Providern angewiesen ist, dafür jedoch beim Harvesting und bei der Vollanzeige.

Als Vorteil des zentralen Index gilt, dass die einzelnen Provider weniger belastet sind, da nicht dauernd unnötige Anfragen an sie gerichtet werden. Dies gilt insbesondere dann, wenn ein Broker nicht auf eine einzelne Datenbank zugreift, sondern wiederum auf einen Broker. Ein Beispiel dafür ist die KOBV-Suchmaschine, die als eine Datenquelle im KVK angeboten wird. Sie ist selbst ein verteiltes Informationssystem und als eine Einheit in ein weiteres verteiltes Informationssystem, den KVK, eingebunden. Würden alle Anfragen des KVK an die KOBV-Suchmaschine über eine verteilte Suche beantwortet, so könnten insbesondere kleine Provider über ihre Kapazität hinaus belastet werden⁶⁷. Allerdings kann auch das Harvesting von Metadaten zu einer erheblichen Belastung von Providern führen. Es gibt jedoch Möglichkeiten, für das Harvesting sowohl die Verfügbarkeit als auch die Belastbarkeit von Providern zu erhöhen. In einem Artikel aus dem D-Lib Magazine vom November 2002⁶⁸ wird dafür ein "OAI-PMH cache service" vorgestellt.

In einem zentralen Index kann neben den bereits erwähnten Retrieval-funktionen auch das Browsing leichter implementiert werden als bei der verteilten Suche⁶⁹. Dazu kommt, dass es technisch aufwändiger ist, eine verteilte Suche zu realisieren als lediglich auf einen einzigen Index zuzugreifen⁷⁰.

Noch ein Vorteil des zentralen Index ist der, dass gegenüber der verteilten Suche mehr Datenquellen in eine Suche mit einbezogen werden können⁷¹. Die verteilte Suche kommt bei zu großen Beständen an ihre Grenzen. Dabei ist auch wichtig, ob es sich bei den Providern wiederum um verteilte Informationssysteme handelt oder ob es bibliografische Datenbanken sind, die die Bestände direkt enthalten.

Genaue Angaben über die maximale Anzahl von Datenbanken, die bei einer verteilten Suche parallel durchsucht werden können, gibt es nicht. In einem Artikel aus dem Jahr 2000 heißt es dazu: "There is mixed opinion on

⁶⁶ Vgl. Lynch, Clifford A: Metadata Harvesting and the Open Archives Initiative

⁶⁷ S. Kapitel 6.3

⁶⁸ Vgl. A scalable architecture for harvest-based Digital Libraries: The ODU/Southampton Experiments. Xiaoming Liu; Tim Brody; Stevan Harnad [u.a.]. In: D-Lib Magazine 8 (2002) 11. Online: <http://www.dlib.org/dlib/november02/liu/11liu.html> [Stand: 11/2002, Abrufdatum: 8.10.2003, Datei: harvesting_1.htm]

⁶⁹ Vgl. Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment

⁷⁰ S. Kapitel 4

⁷¹ Ebd.

how many databases can be effectively parallel searched - the experts' opinions vary between 10 and 100⁷²". Abhängig ist die maximale Anzahl von der Belastbarkeit und Schnelligkeit der eingesetzten Netzwerkverbindung und von der Anzahl und Größe der Datensätze, die pro Datenquelle übertragen werden⁷³. Diese beiden Werte spielen auch bei der Schnelligkeit der Suche eine Rolle. Eine verteilte Suche ist deshalb in der Regel langsamer. Dazu kommt, dass sie auch noch von den Antwortzeiten der einzelnen Server abhängig ist⁷⁴.

Ein Nachteil des zentralen Index ist die nicht immer vorhandene Aktualität der Metadaten. Die Aktualität hängt davon ab, wie oft ein Harvesting stattfindet und stattfinden kann. Wenn ein Provider bei einem Harvestingversuch des Brokers nicht verfügbar ist oder wenn Metadaten falsch übermittelt werden, leidet darunter die Aktualität.

Außerdem stellen sich bei der zentralen Metadatenhaltung Fragen bezüglich des Copyrights, da für einen zentralen Index Daten kopiert werden müssen. Bei dem verteilten Informationssystem Renardus⁷⁵ war das Copyright der Ausschlag dafür, dass kein zentraler Index angelegt werden konnte⁷⁶.

In der Praxis sind häufig Mischformen anzutreffen, bei denen ein Teil der Daten zentral vorgehalten wird, ein Teil dezentral. Ein Beispiel dafür ist der in Kapitel 6.3 vorgestellte KOBV. Mit dieser Variante kann einigen Nachteilen der jeweiligen Architektur aus dem Weg gegangen werden oder sie können zumindest abgemildert werden.

⁷² Dovey, Matthew J.: So you want to build a union catalogue? In: Ariadne 23 (2000) März. Online: <http://www.ariadne.ac.uk/issue23/dovey/intro.html> [Stand: März 2000, Abrufdatum: 8.10.2003, Datei: katalog.htm]

⁷³ Vgl. Der Kooperative Bibliotheksverbund Berlin Brandenburg, S. 11

⁷⁴ Vgl. Lynch, Clifford A: Metadata Harvesting and the Open Archives Initiative S. Kapitel 6.7

⁷⁶ Vgl. Renardus: project deliverable D1.4: Carpenter, Leona: Architectural Model for the Renardus System. Online: http://www.renardus.org/about_us/deliverables/d1_4/D1_4_final.pdf [Stand: 11.12.2000, Abrufdatum: 8.10.2003, Datei: renardus_d1_4.pdf], S. 8

4 SUCHE IN VERTEILTEN INFORMATIONSSYSTEMEN

In diesem Kapitel geht es um die Suche in verteilten Informationssystemen. Gemäß der Definition des MIA Functional Model gehört zur Suche lediglich die Durchführung der Suche und die Ausgabe der Kurztref ferliste und nicht die Vollanzeige. In Anlehnung an diese Definition beschränkt sich auch dieses Kapitel nur auf den Suchprozess bis zur Ausgabe der Kurztref ferliste. Die Reihenfolge der Abschnitte orientiert sich an dem Ablauf einer Suche. So wird zuerst die Auswahl von Datenquellen betrachtet, anschließend Sucheinstiege, Retrievalfunktionen und Feldinhalte. Zuletzt wird auf die Ausgabe von Ergebnissen in der Kurzanzeige eingegangen. Für die einzelnen Abläufe werden verschiedene Realisierungsmöglichkeiten genannt. Aussagen in diesem Kapitel beruhen zum großen Teil auf dem Vergleich der in der Einleitung genannten verteilten Informationssysteme.

In den einzelnen Unterkapiteln wird jeweils getrennt auf die verteilte Suche und die Suche in zentralen Indizes eingegangen, da sich die in diesem Kapitel genannten Suchabläufe bei den beiden Verfahren meist unterscheiden.

4.1 Auswahl von Datenquellen

Bei der Suche in einem durch Harvesting erzeugten zentralen Index stellt sich nicht die Frage, in welchen Datenquellen bei welcher Suchanfrage recherchiert werden soll, da automatisch alle Datenquellen berücksichtigt werden. Bei der verteilten Suche ist dies anders. Zwar gibt es dort auch die Möglichkeit, die Suchanfrage eines Benutzers automatisch an alle beteiligten Datenquellen weiterzuleiten, in der Regel wird einem Benutzer jedoch angeboten, für eine Suche einzelne Datenquellen auszuwählen.

Damit ein Benutzer eine sinnvolle Auswahl treffen kann, muss er darüber informiert werden, was sich in den einzelnen Datenquellen befindet. Dies kann entweder durch erläuternde Texte und Anmerkungen geschehen oder durch Cluster. Cluster⁷⁷ sind nach inhaltlichen oder formalen Gesichtspunkten gebildete Kategorien, denen Datenquellen zugeordnet sind. Zum Beispiel können Datenquellen eines bestimmten Fachgebietes in einem Cluster zusammengefasst sein. Aus der Zuordnung einzelner Datenquellen zu einem Cluster kann ein Benutzer Informationen über die in den Datenquellen vorliegenden Bestände entnehmen. Cluster dienen jedoch nicht in erster Linie

⁷⁷ Penke, K.; D. Scharna und J. Willenborg: Bibliotheksverzeichnisse zur Auswahl von Bibliotheksbeständen in verteilten Informationssystemen: Untersuchung und Einsatz im Kooperativen Bibliotheksverbund Berlin-Brandenburg. Online: http://www.kobv.de/deutsch/content/wir_ueber_uns/docs/biblioverz.pdf [Abrufdatum: 8.10.2003, Datei: kobv_2.pdf], S. 13f.

der Information, sondern sie sind dafür geschaffen, für die Auswahl von Datenquellen verwendet zu werden. Ihre Quellen können als Ganzes in eine Suche miteinbezogen werden und erleichtern und beschleunigen somit die Auswahl der Datenquelle. Meist kann ein Benutzer wählen, ob er ein ganzes Cluster aufrufen will oder nur einzelne Datenquellen daraus.

Je nach Suchanfrage benötigt ein Benutzer verschiedene Cluster. Teilweise sind für seine Anfragen fachliche Cluster geeignet, teilweise sind andere Anordnungen nötig, zum Beispiel die Anordnung nach Medienformen. Aus diesem Grund ist es für einen Benutzer hilfreich, wenn Datenquellen nach verschiedenen Kriterien angeordnet sind oder angeordnet werden können. Christian Heinisch, ein Informationsspezialist, der an der Entwicklung von verteilten Informationssystemen beteiligt war,⁷⁸ fordert diese Flexibilität:

„Eine Hinführung zu den ‚richtigen‘ oder ‚optimalen‘ Quellen erfordert generell ein flexibles, invertierbares Menüsystem, das sich den verschiedenen Sichtweisen und Herangehensweisen des Nutzers anpasst, und die Zusammenstellung und Darstellung der angebotenen Quellen zur jeweiligen Fragestellung optimiert“⁷⁹.

Bei einer verteilten Suche können nicht nur Benutzer Datenquellen für eine Suche auswählen, sondern auch Broker. Die Auswahl durch einen Broker wird aus technischen Gründen eingesetzt, um einzelne Datenquellen nicht unnötig mit Suchanfragen zu belasten und um durch Verringerung des Datenflusses eine Suche zu beschleunigen⁸⁰.

Vor einer Suche ermittelt ein Broker, welche Datenquellen für eine Suche passende Treffer liefern könnten und schickt daraufhin nur an die ausgewählten Datenquellen die Suchanfrage. Dies wird als „query routing“⁸¹ bezeichnet. Der Suche nach Dokumenten ist damit eine Suche nach Datenquellen vorgeschaltet. Die Grundlage für die Auswahl geeigneter Datenquellen können entweder Metadatenbanken, die Informationen über das Profil einzelner Datenquellen enthalten, sein oder sogenannte "centroids". Centroids sind indexähnliche Listen, die Begriffe enthalten, die in einer Datenbank suchbar sind⁸². Sie werden mit Hilfe spezieller Protokolle, meist mit dem Common Indexing Protocol (CIP)⁸³, automatisch erstellt.

⁷⁸ Vgl. „Homepage der Heinisch Informationsbewältigung e.Kfm“.

Online: <http://www.Informationsbewaeltigung.de> [Stand: 2003, Abrufdatum: 8.10.2003, Datei: heinisch.htm]

⁷⁹ Heinisch, Christian: Zur Usability von anbieterunabhängigen Metasuchoberflächen für die Fachinformationsrecherche im Invisible Web, S. 314. Formatierungen wurden im Zitat nicht übernommen.

⁸⁰ Vgl. Renardus: project deliverable D2.1 Technical standards: Whois++ Data access protocol. Online: http://www.renardus.org/about_us/deliverables/d2_1/WHOIS++.html [Stand: 6/2000, Abrufdatum: 20.10.2003, Datei: renardus_d2_1b.htm]

⁸¹ Vgl. John Kirriemuir, Dan; Susan Welsh und Martin Jon Knight: Cross-searching subject gateways: the query routing and forward knowledge approach. In: D-Lib Magazine 4 (1998) 1. Online: <http://www.dlib.org/dlib/january98/01kirriemuir.html> [Stand: 01/1998, Abrufdatum: 8.10.2003, Datei: routing.htm]

⁸² Vgl. ebd.

⁸³ Vgl. Allen J. und M. Mealling: RFC 2651: The Architecture of the Common Indexing Protocol (CIP). Online: <http://www.faqs.org/rfcs/rfc2651.html> [Stand: August 1999, Abrufdatum: 8.10.2003, Datei: cip.htm]

4.2 Sucheinstiege

Wie in Kapitel 1.4.1 beschrieben, können bei einer Suche entweder Suchanfragen in Suchmasken eingegeben werden oder es kann Browsing in den vorhandenen Datenbeständen stattfinden. In verteilten Informationssystemen können beide Arten der Suche realisiert werden.

In Suchmasken werden feldspezifische oder feldübergreifende Suchfelder angeboten. Die Voraussetzung für eine feldspezifische Suche in verteilten Informationssystemen ist, dass Provider gleiche oder ähnliche Datenfelder anbieten. Da es in dieser Arbeit lediglich um Provider geht, die bibliografische Daten anbieten, kann von einer gewissen Einheitlichkeit ausgegangen werden. Vorteilhaft ist es, wenn sich alle Provider, die in ein verteiltes Informationssystem eingeschlossen sind, an gemeinsame Standards bezüglich der angebotenen Datenfelder halten. Es kann sich dabei um Standards handeln, die speziell zwischen den in einem Informationssystem zusammengefassten Providern getroffen werden, oder um allgemeine Standards.

Im Bereich der elektronischen Publikationen wird im Allgemeinen auf den Standard „Dublin Core“ zurückgegriffen, im deutschen Bibliothekswesen werden stattdessen meist MAB und teilweise MARC eingesetzt. Das Problem bei diesen Standardformaten ist, dass zwar Felder definiert sind, um eine standardisierte Übertragung zu gewährleisten, es jedoch keineswegs sicher ist, dass bestimmte Felder auch inhaltlich gefüllt sind.

Beim Harvesting ist ein Standard unerlässlich, da die von den Providern gewonnenen Daten vom Broker verarbeitet werden müssen. Metadaten werden dafür meist in Dublin Core codiert übertragen. Da dies ein sehr unspezifisches Format ist, können einzelne Provider zusätzlich eine Erweiterung von Dublin Core anbieten⁸⁴. So können auf ein spezielles Fachgebiet begrenzte Broker durch Harvesting ausführlichere Metadaten aufnehmen und verwenden. Je feiner die Metadaten strukturiert sind, desto speziellere Sucheinstiege können einem Benutzer angeboten werden.

Auch bei der verteilten Suche halten Provider sich an Standards. Der KOBV-Suchmaschine und Renardus liegen zum Beispiel speziell für das jeweilige System entwickelte Standards für Datenfelder und deren Inhalte zu Grunde, die übrigens auf allgemeinen Standards aufbauen⁸⁵. Zum Teil finden bei einer verteilten Suche jedoch keine Absprachen statt, so dass ein Broker auf das zurückgreifen muss, was er vorfindet. Ist dies der Fall, gibt es verschiedene Möglichkeiten für das Anbieten von Sucheinstiegen. Es kann ein größter gemeinsamer Nenner an gemeinsamen Suchfeldern aus allen beteiligten Datenquellen als Sucheinstieg angeboten werden. Nur so ist gewährleistet, dass mit einer Suchanfrage die Bestände aller Datenquellen berücksichtigt werden.

⁸⁴ Vgl. Lagoze, Carl und Herbert Van de Sompel: The Open Archives Initiative, S. 4

⁸⁵ S. Kapitel 6.3 und 6.7

Nachteilig ist, dass der größte gemeinsame Nenner in der Regel aus sehr wenigen Feldern besteht und deshalb oft nur eine unspezifische Suche möglich ist. In diesem Verfahren legen die Datenquellen, die wenig spezifische Suchfelder anbieten, also die „schwächsten“ Teilnehmer, die Sucheinstige für ein verteiltes Informationssystem fest. Dieses Problem kann durch "dynamische Masken"⁸⁶, die an die jeweilige Suchsituation angepasst werden, verringert werden. Das bedeutet, dass nur aus den jeweils für eine Suche ausgewählten Datenbanken der größte gemeinsame Nenner gebildet wird. Daneben können für Suchfelder Prioritäten vergeben werden, die dazu führen, dass Suchfelder auch dann angeboten werden, wenn sie nicht zu der Menge des größten gemeinsamen Nenners gehören, sondern nur in einzelnen zu durchsuchenden Datenbanken auftauchen⁸⁷.

Das Problem bei dem Anbieten von Sucheinstiegen, die nur in einem Teil der angeschlossenen Datenquellen suchbar sind, ist, dass bei einer Suche nicht alle ausgewählten Datenbanken berücksichtigt werden. Der Benutzer muss auf diese Einschränkung hingewiesen werden. Meist geschieht dies in dem Hilfetext, so dass wohl wenige Benutzer darauf aufmerksam werden.

Soll ein Benutzer in systematisch angeordneten Datenbeständen browsen können anstatt über eine Suchmaske zu suchen, müssen alle Datenbestände einheitlich klassifiziert sein. In Kapitel 4.4 wird näher darauf eingegangen.

4.3 Retrievalfunktionen

Die Suche in einem zentralen Index ist unabhängig von den Retrievalfunktionen der einzelnen Datenquellen. Es hängt lediglich von dem Broker ab, welche Retrievalfunktionen angeboten werden.

Im Gegensatz dazu ist eine verteilte Suche stark von dem Angebot einzelner Provider abhängig. Meist wird ein größter gemeinsamer Nenner an Retrievalfunktionen angeboten, so dass alle Datenquellen in eine Suche mit einbezogen werden können. Dies hat zur Folge, dass, wie schon bei den Sucheinstiegen, der Provider, der am wenigsten anbietet, das Angebot in einem verteilten Informationssystem bestimmt. Um diesem entgegenzuwirken, kann ein "Konzept[s] der Alternativstrategien"⁸⁸ verfolgt werden. Dieses ermöglicht, dass über einen größten gemeinsamen Nenner hinaus Retrievalfunktionen angeboten werden können, ohne dass dadurch Datenquellen, die wenig Retrievalfunktionen anbieten, von vorneherein von einer Suche ausgeschlossen werden. Dafür werden die einzelnen Retrievalfunktionen in eine hierarchische Reihenfolge gebracht. Wenn nun bei einer Suche eine Retrievalfunktion eingesetzt wird, die von einer Datenquelle nicht unterstützt wird, so wird in dieser Datenquelle stattdessen die in der Hierarchie nächstniedrigere Funktion verwendet. Ein Beispiel hierfür könnte die Suche

⁸⁶ Vgl. Heinisch, Christian: Zur Usability von anbieterunabhängigen Metasuchoberflächen für die Fachinformationsrecherche im Invisible Web, S. 314

⁸⁷ Vgl. ebd., S. 315

⁸⁸ Ebd., S. 315

mit dem Booleschen Operator UND sein. Wenn ein Provider diese Retrievalfunktion nicht unterstützt, sondern nur das Boolesche ODER, so würde bei dem Einsatz des Konzepts der Alternativstrategien die Suche in dem entsprechenden Provider nicht mit einer 0-Treffer-Liste enden, sondern das UND würde in ein ODER umgewandelt.

4.4 Feldinhalte

Wie bereits dargestellt, werden bei der feldspezifischen Suche in verteilten Informationssystemen Felder angeboten, die in den einzelnen Datenquellen gleich oder ähnlich sind. Es stellt sich nun die Frage, wann Felder als gleich bezeichnet werden können. In erster Linie natürlich, wenn sie gleiche Elemente, wie zum Beispiel Autorennamen oder Titel, nachweisen. Für das Retrieval ist darüber hinaus wichtig, wie Elemente nachgewiesen sind. Ist ein Autor zum Beispiel in einer Datenquelle phraseninvertiert und über die Zeichenkette „Nachname, Vorname“ suchbar, in einer anderen über die Zeichenkette „Vorname Nachname“, so wird der Autor mit einer Suchanfrage nicht in beiden Datenquellen gefunden. Neben den Unterschieden in der Ansetzung und Form gibt es zwischen verschiedenen Datenquellen Unterschiede in der Sprache und insbesondere in der Inhaltserschließung.

Um zu einer Einheitlichkeit von Feldinhalten zu gelangen, ist der einfachste Weg, dass Absprachen zwischen einzelnen Providern eines verteilten Informationssystems getroffen werden. Wie bereits erwähnt, geschieht dies bei Renardus und bei der KOBV-Suchmaschine. Im Bereich des Bibliothekswesens kann darüber hinaus auf allgemein verwendete Standards für Feldinhalte zurückgegriffen werden, wie zum Beispiel auf die Schlagwortnormdatei (SWD) für die inhaltliche Erschließung.

Wenn Feldinhalte verschiedener Datenquellen jedoch unterschiedlich vorliegen, können Anpassungen vorgenommen werden. Der Vorteil beim Harvesting ist, dass alle Daten für eine Suche bei dem Broker aufliegen und er diese nach Belieben verändern oder ergänzen kann. Aus diesem Grund ist ein Browsing in systematischen Begriffsordnungen durch das Harvesting gut zu realisieren.

Bei der verteilten Suche ist eine Anpassung wiederum schwieriger. Bei der Vereinheitlichung verschiedener Sacherschließungssysteme zum Beispiel muss eine "semantische Transformation"⁸⁹ stattfinden. Dabei werden entweder die Datensätze in den verschiedenen Datenbanken um Suchterme eines ausgewählten Erschließungssystems ergänzt ("Retrokonversion" (S. 338)) oder die Anfragen werden in die Dokumentationssprachen der jeweiligen Datenbanken übersetzt („Anfragemanipulation“ (S. 339)). Grundlage

⁸⁹ Müller, Matthias N.O.: Integration unterschiedlich erschlossener Datenbestände am Beispiel der Virtuellen Fachbibliothek Sozialwissenschaften. In: Competence in content: 25. Online-Tagung der DGI: Frankfurt am Main, 3. bis 5. Juni 2003: Proceedings. Frankfurt am Main, Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis e.V. 2003, S. 337. Alle Seitenangaben im Abschnitt beziehen sich auf diese Ausgabe.

für beide Realisierungen der Transformation sind entweder individuell erstellte Übersetzungstabellen, sogenannte Cross-Konkordanzen, oder automatisierte Verfahren⁹⁰ (S. 343-344).

Ein besonderes Problem stellt sich für die Suche im Bereich der Inhaltsererschließung, wenn keine Anpassung möglich ist, da sich in einzelnen Datenquellen die Erschließungstiefe und -verlässlichkeit stark unterscheiden. Einen Lösungsansatz dafür bietet das sogenannte Schalenmodell, das in der Virtuellen Fachbibliothek Sozialwissenschaften eingesetzt wird⁹¹. Datenquellen werden verschiedenen Schalen zugeordnet. In der innersten Schale befinden sich "qualitativ hochwertige, umfangreich erschlossene und möglichst alle zentralen Dokumente eines Fachgebiets"⁹². Bei einer Recherche kann der Benutzer sich für die Suche nur in inneren Schalen entscheiden und damit eine hochwertige Inhaltsererschließung nutzen oder für eine breitere Recherche in allen Schalen und muss dafür auf eine durchgehend gute Erschließung verzichten.

4.5 Ausgabe der Kurztrefferliste

In verteilten Informationssystemen ist der Nachweis der Ergebnisse einer Suche meist in einer gemeinsamen Kurztrefferliste zusammengefasst. Es wird zuerst betrachtet, wie die einzelnen Ergebnisse angezeigt werden und wie sie in der Kurztrefferliste angeordnet werden. Anschließend geht es um den Umgang mit Dubletten.

4.5.1 Anzeige und Anordnung der Treffer

Von einem Broker mit einem zentralen Index wird die Anzeige und Anordnung der Treffer nicht anders durchgeführt als in einer einzigen Datenbank. Im Gegensatz dazu ist bei der verteilten Suche eine Anpassung der Suchergebnisse aus den einzelnen Datenquellen nötig. Die Treffer können einheitlich angezeigt werden oder jeweils in der Art, wie es in ihrer Ursprungsdatenbank üblich ist. Wenn sie einheitlich angezeigt werden sollen, müssen die Ergebnisse von den Providern in einem Format geliefert werden, das der Broker auswerten kann. Hierfür werden meist die bereits erwähnten Datenformate MAB, MARC oder Dublin Core eingesetzt. Diese Auswertung findet bei einem zentralen Index schon beim Harvesting statt und ist somit aus dem Suchprozess ausgegliedert.

Die einzelnen Ergebnisse können auf Trefferlisten entweder nach ihrem Fundort getrennt angeordnet sein oder gemischt werden und dann nach anderen Kriterien, wie zum Beispiel der Relevanz (durch relevance ranking), sortiert werden. Ein Sortieren der einzelnen Treffer nach datenbankübergrei-

⁹⁰ Unter automatisierte Verfahren zählen quantitativ-statistische Verfahren. Darüber hinaus nennt Müller noch die Transformation mit Hilfe neuronaler Netze.

⁹¹ Vgl. Kluck, Michael; J. Krause und M.N.O. Müller: Virtuelle Fachbibliothek Sozialwissenschaften: Projektskizze. Bonn: IZ Sozialwissenschaften 2000, S. 14ff.

⁹² Müller, Matthias N.O.: Integration unterschiedlich erschlossener Datenbestände am Beispiel der Virtuellen Fachbibliothek Sozialwissenschaften, S. 337

fenden Kriterien ist bei der verteilten Suche zum einen aufwändig, zum anderen nur mit einer zeitlichen Verzögerung möglich, da die Trefferliste in diesem Fall nicht kontinuierlich aufgebaut werden kann.

4.5.2 Umgang mit Dubletten

Dubletten sind der Nachweis für gleiche Dokumente. Bei der Abfrage mehrerer bibliografischer Datenbanken ist es nichts Außergewöhnliches, dass Datenquellen gleiche Dokumente nachweisen. Es gibt zwei verschiedene Definitionen für gleiche Dokumente: eine basiert auf dem Werk, die andere auf dem Dokument⁹³. Die auf dem Werk basierende legt den Inhalt einer Veröffentlichung zu Grunde, das "geistige[...] Produkt eines Verfassers"⁹⁴, und lässt formale Unterschiede außen vor.

Dies soll an einem Beispiel verdeutlicht werden: Angenommen, es liegt Goethes Faust in zwei verschiedenen Ausgaben vor: einmal als Reclam-Heft, einmal als Dtv-Taschenbuch. Bei der Dublettenerkennung, die auf Werken basiert, werden beide Bücher als Dublette erkannt, nicht jedoch bei der Dublettenerkennung, die auf Dokumenten basiert, da bei dieser sowohl inhaltliche als auch formale Kriterien berücksichtigt werden. Verteilte Informationssysteme legen unterschiedliche Definitionen für Dubletten zu Grunde. In ZACK, das in Kapitel 6.2 näher vorgestellt wird, kann der Benutzer die Definition sogar selbst in der Suchmaske wählen. Die KOBV-Projektgruppe bezeichnet die Erkennung auf Grundlage von Werken als benutzerfreundlicher, da die Trefferliste kürzer sei und ein Benutzer auf einen Blick sehe, ob er fündig geworden sei oder nicht, ohne viele ähnliche Dokumentangaben vergleichen zu müssen⁹⁵.

Es findet nicht immer eine Dublettenerkennung statt. Bei einer Erkennung werden bestimmte, möglichst vorher normierte Datenfelder verglichen. Der Vergleich kann fehlertolerant sein, so dass gewisse Abweichungen auch bei Dubletten zugelassen sind. Um eine zuverlässige Erkennung zu haben, sollten mehrere Felder verglichen werden. Hilfreich ist die Zuordnung verschiedener Gewichtungsfaktoren für die einzelnen Felder, so dass es für die Dublettenerkennung bei manchen Feldern wichtiger ist, dass sie identisch sind, als bei anderen⁹⁶. Bei der verteilten Suche findet eine Dublettenerkennung bei der Suche statt, in zentralen Indizes geschieht dies schon bei der Aufnahme von Metadaten in den Index nach dem Harvesting.

Bei der verteilten Suche ist nach einer Dublettenerkennung zu klären, wie Dubletten in der Kurzanzeige aufgeführt werden sollen. Es ist möglich, einen Datensatz aus den Dubletten auszuwählen, der stellvertretend für die anderen für die Kurzanzeige ausgewählt wird. Es kann einfach der erste

⁹³ Vgl. Kuberek, Monika: Dublettenbehandlung (Match- und Merge-Verfahren) in der KOBV-Suchmaschine: Grundlagen. Online: http://www.kobv.de/deutsch/content/wir_ueber_uns/kobv_projekt_1997-2000/docs/sc-99-16.pdf [Stand: 1999, Abrufdatum: 8.10.2003, Datei: kobv_3.pdf], S. 4

⁹⁴ Ebd., S. 7

⁹⁵ Vgl. ebd., S. 8

⁹⁶ Ein solcher Schlüssel wird bei ZACK (s. Kapitel 6.2) verwendet

Treffer angezeigt werden⁹⁷ oder es können für die Auswahl andere Kriterien zu Grunde liegen, wie zum Beispiel die Länge eines Datensatzes, seine Aktualität oder seine Datenquelle. Es kann bei der verteilten Suche auch sinnvoll sein, den Benutzer in die Entscheidung, welche Dubletten angezeigt werden, miteinzubeziehen⁹⁸. Als weitere Möglichkeit kann für die Anzeige ein Nachweis aus den Daten der einzelnen Dubletten neu generiert werden. Dafür müssen Angaben darüber vorliegen, wie dieser generiert werden soll. Von dem Nachweis in der Kurzanzeige aus müssen Verknüpfungen zu den einzelnen doppelten Datensätzen vorliegen.

In einem zentralen Index kann die Frage, welche Datensätze bei Dubletten abgespeichert werden, ähnlich beantwortet werden wie die, welche Datensätze bei der verteilten Suche angezeigt werden. Entweder werden Dubletten zu einem neuen Datensatz zusammengefasst oder einzelne doppelte Datensätze beseitigt⁹⁹. Die Auswahl, welcher Datensatz bei der Beseitigung von Dubletten behalten wird, kann auf Grund der gleichen Kriterien getroffen werden wie bei der verteilten Suche.

Die KOBV-Suchmaschine löst das Problem der Dubletten für ihren zentralen Index zum Beispiel folgendermaßen: es wird auf Grund bestimmter Kriterien bei Dubletten ein Datensatz für die Indexierung ausgewählt; dieser wird allerdings im Bereich der inhaltlichen Erschließung um Schlagwörter von anderen doppelten Datensätzen angereichert¹⁰⁰.

⁹⁷ Vgl. Renardus: project deliverable D1.1: Evaluation report of existing broker models in related projects, S. 81

⁹⁸ Vgl. John Kirriemuir, Dan; Susan Welsh und Martin Jon Knight: Cross-searching subject gateways: the query routing and forward knowledge approach

⁹⁹ Vgl. Lynch, Clifford A: Metadata Harvesting and the Open Archives Initiative

¹⁰⁰ Vgl. Der Kooperative Bibliotheksverbund Berlin Brandenburg, S. 21

5 TECHNISCHE REALISIERUNG VERTEILTER INFORMATIONSSYSTEME

In verteilten Informationssystemen greifen sowohl Menschen auf Systeme zu als auch Maschinen bzw. Programme. Ist der Benutzer ein Mensch, so kann er, wie in Kapitel 1.4.1 beschrieben, auf einen Broker entweder mit Hilfe einer menügesteuerten grafischen Benutzeroberfläche zugreifen oder direkt mit einer Kommandosprache. Maschinen greifen nicht auf eine Benutzeroberfläche zu, sondern kommunizieren auf einer tieferen Ebene über andere Schnittstellen, zum Beispiel mit der Software für die Verarbeitung von Programmen (s. Abb. 2).

Es wird von einem Client-Server basierten Datenaustausch ausgegangen. Clients schicken Anfragen an Server und diese reagieren darauf durch das Senden von Ergebnissen¹⁰¹. Damit ein Datenaustausch zwischen verschiedenen Systemen möglich ist, müssen Server und Clients sich an Protokolle, das heißt an "feste Regeln für den Austausch"¹⁰², halten. Um mit Protokollen umgehen zu können, müssen Systeme Schnittstellen dafür zur Verfügung stellen. Das OSI-Referenzmodell ordnet die Protokolle, die bei einer Nachrichtenübertragung zum Einsatz kommen, 7 Schichten zu¹⁰³.

Hier werden Protokolle der obersten Schicht, das heißt der Schicht, mit der der Benutzer direkt kommuniziert, behandelt. Diese Schicht ist für die Anwendungen verantwortlich. Die folgenden Protokolle umfassen nicht nur Aufgaben der obersten OSI-Schicht, sondern teilweise der obersten drei, wenn sie nicht auf dem OSI-Standard, sondern auf dem OSI-ähnlichen Standard TCP/IP beruhen. Dieser besteht nur aus vier Schichten, so dass dessen Anwendungsschicht die Aufgaben der drei obersten OSI-Schichten (Application, Presentation und Session) vereint¹⁰⁴.

In diesem Kapitel werden zuerst die beiden Standardprotokolle Z39.50 und HTTP vorgestellt. Anschließend werden mit Web-Services auf HTTP beruhende Verfahren für verteilte Informationssysteme dargestellt. Die beiden Web-Services ZING/SRW und OAI-PMH werden einzeln vorgestellt. Zum Schluss folgt ein allgemeiner Überblick über Verfahren für die Verknüpfung verteilter Datenquellen. Nicht eingegangen wird auf spezielle Dienste, die über den einfachen Zugang hinausgehen, wie zum Beispiel die Benutzererkennung oder die Auswahl geeigneter Datenbanken. Es werden nicht alle existierenden Zugangsmechanismen und -protokolle vorgestellt,

¹⁰¹ Vgl. Eberhart, Andreas und Stefan Fischer: Java-Bausteine für E-Commerce-Anwendungen: verteilte Anwendungen mit Servlets, CORBA und XML. München: Hanser 2000, S. 15f.

¹⁰² Jakobs, Kai und Otto Spaniol: [Artikel] Protokoll. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S. 559

¹⁰³ Vgl. Quernheim, Ulrich und Otto Spaniol: [Artikel] OSI-Referenzmodell. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy; Otto Spaniol. Berlin: Springer 1999, S. 515f.

¹⁰⁴ Vgl. Netzwerke: Grundlagen. Hrsg.: RRZN/ Universität Hannover. 3. Aufl. Nackenhei: Herdt 2000, S. 81

sondern es wurde eine Auswahl getroffen an Hand der Systeme, die dieser Arbeit zu Grunde liegen. Im letzten Abschnitt wird gezeigt, wie diese Protokolle und Verfahren in verteilten Informationssystemen eingesetzt werden und wie sie gemeinsam genutzt werden können.

5.1 Z39.50

Das Protokoll Z39.50¹⁰⁵ definiert einen Standard für die Kommunikation zwischen einem Server und einem Client bei der Suche und Lieferung bibliografischer Daten. Es wird auch als Such- und Retrievalprotokoll bezeichnet. Die erste Version wurde 1984 von der NISO (National Information Standards Organization), einem Komitee des "American National Standards Institute" (ANSI), verabschiedet¹⁰⁶. Dieses Komitee, das für Standards im Bereich des Bibliotheks- und Informationswesens verantwortlich ist, hat innerhalb des ANSI die Bezeichnung Z39¹⁰⁷. Daraus leitet sich der Name des Protokolls ab. Die Zahl 50 in dem Namen Z39.50 sagt aus, dass es sich um den 50. Standard handelt, den die NISO verabschiedet hat.

Ein sehr ähnliches Protokoll auf internationaler Ebene, "Search and Retrieve" (SR) genannt, wurde in die zweite Version 1992 übernommen, die dritte und neueste Version stammt von 1995. Gepflegt wird das Protokoll von einem eigenen Gremium, der Z39.50 Maintenance Agency, das zur Library of Congress gehört¹⁰⁸.

Konzept:

Wie bereits erwähnt beruht Z39.50 auf dem Client-Server-Modell. Zwischen dem Client und dem Server fließen nur standardisierte Daten. Aus diesem Grund setzt die Nutzung von Z39.50 voraus, dass im Client eine Software vorhanden ist, das sogenannte "origin", das Anfragen in eine Standardform umwandelt und standardisierte Rechercheergebnisse annehmen kann¹⁰⁹. Das Gleiche gilt für den Server, dessen Umwandlungssoftware als "target" bezeichnet wird: eine standardisierte Anfrage wird vom target entgegengenommen, auf die jeweilige Datenbank angepasst und Ergebnisse werden standardisiert ausgegeben¹¹⁰. Origins und targets müssen demnach individuell auf die jeweiligen Systeme angepasst werden. Z39.50 eignet sich gut für den Einsatz in verteilten Informationssystemen, da eine Anfrage

¹⁰⁵ Vgl. Z39.50 –1995: Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. Hrsg.: Z39.50 Maintenance Agency. Online: <ftp://ftp.loc.gov/pub/z3950/official/part1.pdf> [Stand: Juli 1995, Abrufdatum: 8.10.2003, Datei: z3950_1.pdf]

¹⁰⁶ Lynch, Clifford A.: The Z39.50 Information Retrieval Standard: Part I: A strategic view of its past, present and future. In: D-Lib Magazine (1997) 4. Online: <http://www.dlib.org/dlib/april97/04lynch.html> [Stand: 4/1997, Abrufdatum: 9.10.2003, Datei: z3950_3.htm]

¹⁰⁷ Vgl. ebd.

¹⁰⁸ Vgl. „Homepage der Z39.50 Standard National Agency“. Online: <http://www.loc.gov/z3950/agency/> [Stand: 14.10.2003, Abrufdatum: 21.10.2003, Datei: maintenance.htm]

¹⁰⁹ Vgl. Up and running: Implementing Z39.50: proceedings of a symposium sponsored by the State Library of Iowa November 26, 1996, Ames, Iowa. Hrsg.: Sara L. Randall. Bethesda: NISO Press 1997, S. 3

¹¹⁰ Vgl. ebd.

durch Z39.50 automatisch in eine einheitliche Form gebracht wird, so dass sie von allen Systemen verstanden wird, die ein Z39.50-target implementiert haben. Auf die Benutzeroberfläche und die Struktur der Datenbanken hat Z39.50 keinen Einfluss.

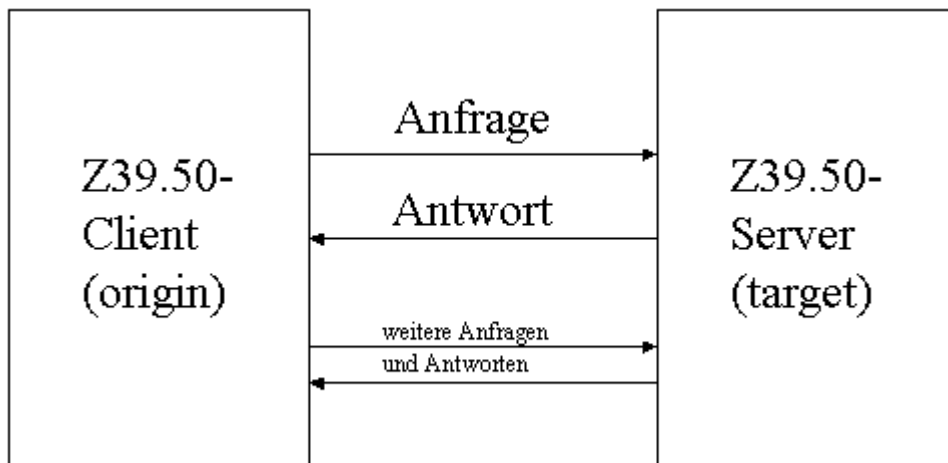


Abb. 4: Kommunikation über Z39.50 zwischen Servern und Clients

Anfragetypen, Ausgabe:

Z39.50 umfasst verschiedene Anfragetypen, die vom Client an den Server gesendet werden. Hier werden einige genannt.

Zu den Kernanfragetypen gehört der Startbefehl "init", mit dem der Client eine Sitzung eröffnet und "close", durch den er sie wieder schließt. Zwischen diesen beiden Befehlen können mehrere Anfragen und Antworten zwischen einem Server und einem Client ausgetauscht werden. Schließt ein Client eine Sitzung nicht, trennt der Server nach zu langer Inaktivität die Verbindung¹¹¹. Da Z39.50 Sitzungen unterhalten kann, gehört es zu den als zustandswahrend bezeichneten Protokollen.

Ein weiterer Anfragetyp ist "search". Dadurch wird eine Suche ausgelöst, worauf als Antwort die Anzahl der Treffer geliefert wird. Um die einzelnen Treffer zu erhalten, schickt der Client die Anfrage "present". Die Anfrage "sort" liefert eine sortierte Trefferliste, "explain" gibt Daten über die zu durchsuchende Datenbank heraus.

Browsing in Indizes ist durch die Anfrage "scan" möglich. Durch diese kann auch eine verteilte Registersuche implementiert werden, indem pro Datenbank eine bestimmte Anzahl von Registereinträgen angefordert wird, die für die Ausgabe gemischt und zusammengefasst werden. In dem verteil-

¹¹¹ Vgl. Dierolf, Uwe und Michael W. Mönich: KVK - Karlsruher Virtueller Katalog: zwei Jahre virtuell. In: B.I.T. Online (1998) 3, S. 160

ten Informationssystem ZACK zum Beispiel ist eine verteilte Registersuche mit Hilfe dieser Funktion realisiert¹¹².

Codierung von Nachrichten:

Die Anfragetypen von Z39.50 sorgen noch nicht dafür, dass übertragene Daten so codiert sind, dass sie sowohl von dem Server als auch von dem Client ausgewertet werden können. Das Protokoll definiert über die Anfragetypen hinaus für das Senden von Suchanfragen an den Server Abfragesprachen (query languages). Diese sind in Listen, so genannten attribute-sets, definiert¹¹³. Sie legen zum Beispiel einzelne Feldnamen fest und Retrievalfunktionen, die verwendet werden können.

Es gibt verschiedene attribute-sets. Im Bibliothekswesen wird zum Beispiel „Bib-1“ eingesetzt, das als Feldnamen typische bibliografische Attribute wie „author“ und „title“ enthält. Origin und target müssen beide das attribute-set kennen und unterstützen, damit eine Verständigung möglich ist. Für die Anfragen ist ein Format festgelegt, sondern auch für die Antworten. Dafür werden bestimmte Datenformate unterstützt, wie zum Beispiel MARC. Am Anfang einer Kommunikation verständigen der Server und der Client sich über die unterstützten attribute-sets und Datenformate.

Bedeutung:

Z39.50 spielt insbesondere in der Bibliothekswelt bei dem Zugang zu elektronischen bibliografischen Daten eine große Rolle¹¹⁴. Viele Kataloge sind über eine Z39.50-Schnittstelle zugänglich. Verteilte Suchen wurden noch bis vor kurzem in der Regel über Z39.50 abgewickelt¹¹⁵. Heute gibt es jedoch mit Web-Services, die in Abschnitt 5.3 vorgestellt werden, auch andere bedeutende Mechanismen für die verteilte Suche.

Probleme bereitet bei Z39.50 die Anpassung von Targets und Origins aufeinander. Nicht alle Targets und Origins unterstützen alle existierenden Codierungen für die Ein- und Ausgabe sowie weitere Einstellungen¹¹⁶. Um für eine bessere Verständigung zu sorgen, wurden meist für einzelne fachliche Anwendergruppen verschiedene Profile gebildet, die ein bestimmtes Set

¹¹² S. Kapitel 6.2

¹¹³ Vgl. Up and running: Implementing Z39.50, S. 4

¹¹⁴ Vgl. für diesen Abschnitt: Renardus: project deliverable D2.1 Technical Standards: Z39.50 and profiles. Online: http://www.renardus.org/about_us/deliverables/d2_1/Z3950.html [Stand: 20.6.2000, Abrufdatum: 8.10.2003, Datei: renardus_d2_1a.htm]

¹¹⁵ Vgl. Veen, Theo van: Harvesting metadata and the use of application profiles: ELAG 2003: Cross language applications and the web, workshop 1. Online: http://www.elag2003.ch/ws/ws_1.pdf [Stand: 4/2003, Abrufdatum: 9.10.2003, Datei: harvesting_2.pdf], S. 2

¹¹⁶ Vgl. Miller, Paul: Z39.50 for all. In: Ariadne 21 (1999). Online: <http://www.ariadne.ac.uk/issue21/z3950/intro.html> [Stand: 9/1999, Abrufdatum: 9.10.2003, Datei: z3950_2.htm]

an zu unterstützenden Codes festlegen. Mit dem Bath Profile¹¹⁷ gibt es auch ein spartenübergreifendes Profil, das auf Grundfunktionen beschränkt ist.

5.2 HTTP

HTTP¹¹⁸ ist die Abkürzung für "Hypertext Transfer Protocol". Es handelt sich dabei um ein Netzwerkprotokoll, das die Datenübertragung im WWW regelt. Es wurde Ende der 80er-Jahre von Tim Berners-Lee, einem englischen Physiker, der heute Direktor des W3C ist, als Standard für das World Wide Web (WWW) entwickelt¹¹⁹. Die neueste Version HTTP/1.1 wurde 1999 veröffentlicht und gilt derzeit als Standard.

Konzept:

Eine Kommunikation über HTTP läuft folgendermaßen ab: ein HTTP-Client eröffnet eine Sitzung, indem er eine Anfrage an einen Server sendet (request), dieser antwortet auf die Anfrage (response) und schließt damit die Sitzung¹²⁰. Der Client wertet danach die Antwort aus. Da HTTP nicht in der Lage ist, eine über mehrere Anfragen andauernde Sitzung zwischen einem Client und einem Server aufrecht zu erhalten, ist es im Gegensatz zu Z39.50 zustandslos.

Anfragetypen, Ausgabe, Codierung von Nachrichten:

Für HTTP requests werden hier zwei Anfragetypen, "methods" genannt, vorgestellt, und zwar GET und POST. Beide werden vom Client verwendet, um Daten vom Server anzufordern. Der Client muss spezifische Angaben über die angeforderten Daten weitergeben. Dies geschieht bei dem Anfragetyp GET ausschließlich mit Hilfe einer URI (Uniform Resource Identifier). Eine URI kann lediglich aus einer Zieladresse, der URL (Uniform Resource Locator), bestehen oder noch zusätzliche Angaben enthalten. Bei einer Suche in Datenbanken über das WWW können zum Beispiel Suchbegriffe an die URL angehängt werden. Eine URI könnte in einem solchen Fall folgendermaßen aussehen: `http://www.adresse.de/test.cgi?query=dc.title=cat&maximumRecords=10&recordSchema=DC&sortKeys=title,dc&startRecord=1`¹²¹

¹¹⁷ Vgl. The Bath Profile: an international Z39.50 specification for library applications and resource discovery release 1.1. Hrsg.: Carrol Lunau; Paul Miller und William E. Moen. Online: <http://www.nlc-bnc.ca/bath/bp-current.htm> [Stand: 2/2001, Abrufdatum: 13.10.2003, Datei: bath.htm]

¹¹⁸ Vgl. Hypertext Transfer Protocol - HTTP/1.1. Fielding, R.; J. Gettys; J. Mogul [u.a.]. Online: <http://www.ietf.org/rfc/rfc2616.txt> [Stand: 6/1999, Abrufdatum: 8.10.2003, Datei: http_1.txt]

¹¹⁹ Vgl. Internet: eine Einführung in die Nutzung der Internet-Dienste. Hrsg.: von RRZN/ Universität Hannover. 8. Aufl. Hannover: RRZN 2000, S. 42

¹²⁰ Vgl. Professional Java Serverprogramming. Danny Ayers; Hans Bergstein; Michael Bogovich [u.a.]. Birmingham: Wrox Press 1999, S. 790

¹²¹ leicht verändert aus: SRW-Search/Retrieve Web Service: Z39.50 International: next generation: SRU: URL access mechanism. Hrsg.: Z39.50 Maintenance Agency. Online: <http://www.loc.gov/z3950/agency/zing/srw/sru.html> [Stand: 25.11.2002, Abrufdatum: 9.10.2003, Datei: sru_1.htm]

Die angehängten Begriffe sind nach einem bestimmten Standard codiert oder nicht. Standards für die Codierung von bibliografischen Daten in einer URI sind die OpenURL¹²² und ZING/SRU (URL Access Mechanism)¹²³. Die obige Beispiel-URI ist in ZING/SRU codiert. ZING/SRU wurde von ZING (Z39.50-International: Next Generation) entwickelt, einer Arbeitsgruppe der Z39.50 Maintenance Agency.

Wenn bei Anfragen der Befehl POST verwendet wird, kann neben der URI separat eine größere Datenmenge an Eingaben übertragen werden. Die Eingabe kann wiederum individuell codiert sein oder in einem standardisierten Format. SOAP, das in Kapitel 5.3.1 vorgestellt wird, stellt einen solchen Standard dar.

Bei Antworten über HTTP überliefert ein Server, wenn er keine Fehlermeldung herausgibt, die angeforderten Daten, meist Dokumente oder andere Ressourcen, an einen Client. Da in der Regel viele verschiedene Clients auf Server zugreifen, müssen die Ergebnisse der Server in einer einheitlichen Standardsprache codiert bzw. ausgezeichnet vorliegen. Die Standardauszeichnungssprache im WWW ist HTML (Hypertext Markup Language). Berners-Lee führte sie im Zusammenhang mit der Entwicklung von HTTP ein¹²⁴.

Um HTML zu definieren, wurde eine Meta-Auszeichnungssprache verwendet (SGML). Heute wird als Auszeichnungssprache weniger SGML eingesetzt, sondern eine Untermenge davon, und zwar XML (Extensible Markup Language). Mit Hilfe von XML kann eine Fülle von neuen Auszeichnungssprachen generiert werden. Die Generierung findet heute meist mit Hilfe eines „XML-Schemas“¹²⁵ statt. Darin ist festgelegt, wie bestimmte Elemente eines Dokuments codiert werden sollen. Um Nachrichten auswerten zu können, müssen Empfänger entsprechende Schnittstellen besitzen. Die Standard-Clients im WWW, die HTML verstehen, sind Browser.

Abfrage von Datenbanken über HTTP

Wenn Datenbanken über HTTP abgefragt werden sollen, stehen dem zwei Eigenschaften von HTTP im Wege: die Zustandslosigkeit und die Unfähigkeit, dynamische Seiten zu erzeugen¹²⁶.

Die Zustandslosigkeit ist ein Problem, da bei jeder Anfrage über HTTP eine neue Verbindung zwischen einem Server und einem Client aufgebaut wird. Das bedeutet, dass nicht auf einen vorherigen Datenaustausch zurück-

¹²² OpenURL Standard: a draft American national standard developed by the National Information Standards Organization. Online: http://library.caltech.edu/openurl/Documents/OpenURL_Draft_Standard_Edited_03.doc [Stand: 2003, Abrufdatum: 14.10.2003, Datei: openurl_1.doc]

¹²³ SRW-Search/Retrieve Web Service: Z39.50 International: next generation: SRU: URL access mechanism.

¹²⁴ Vgl. Internet: eine Einführung in die Nutzung der Internet-Dienste, S. 42

¹²⁵ Vgl. XML Schema. W3C Architecture Domain. Hrsg.: W3C. Online: <http://www.w3.org/XML/Schema> [Stand: 4/2000, Abrufdatum: 7.10.2003, Datei: xml.htm]

¹²⁶ Vgl. Architektur von Web-Informationssystemen. Gerti Kappel; Werner Retschitzegger; Birgit Pröll [u.a.]. In: Web & Datenbanken: Konzepte, Architekturen, Anwendungen. Hrsg.: Erhard Rahm und Gottfried Vossen. Heidelberg: dpunkt-Verl. 2003, S. 102. Alle Angaben im Text beziehen sich auf diese Ausgabe.

gegriffen werden kann, was bei einer Suche in Datenbanken jedoch oft notwendig ist. Deshalb gibt es Mechanismen, wie zum Beispiel Cookies, mit Hilfe derer die Zustandslosigkeit aufgehoben werden kann (S. 110). Cookies sind Textdateien, die im Auftrag eines Web-Servers vorübergehend oder langfristig auf dem Speicher eines lokalen Rechners abgelegt werden und in denen Informationen über den Datenaustausch zwischen einem Server und Client gespeichert werden.

Das zweite Problem ist nicht so leicht zu lösen. HTTP ist darauf ausgelegt, statische Seiten zu überliefern. Bei einer Recherche in Datenbanken kann jedoch vor einer Suche keine statische Seite aufgerufen werden, da zum Beispiel die Kurzanzeige einer Suche nicht statisch vorliegt, sondern erst durch die Suche in der Datenbank dynamisch generiert wird. Damit dynamische Seiten erstellt werden können, müssen Programme in Webseiten integriert werden (S. 102). Diese Programme können entweder auf dem Client oder auf dem Server angesiedelt sein. Zu den serverseitigen gehören unter anderem CGI (Common Gateway Interface), ASP (Active Server Pages), PHP und Java-Servlets, zu den clientseitigen Java-Applets und verschiedene Plug-Ins (S. 103).

Bei den clientseitigen Techniken findet der Zugriff auf Datenbanken direkt vom Client aus statt. Dies führt dazu, dass die Verbindung zwischen Server und Client nicht über HTTP aufgebaut wird, sondern über andere Protokolle (S. 106-109). Für eine Verständigung muss sowohl auf dem Client als auch auf dem Server die notwendige Software installiert sein. Bei den serverseitigen Techniken wird ein auf dem Server liegendes Programm aufgerufen, das eine Schnittstelle zwischen einem WWW-Server und Datenbanken darstellt. Es verarbeitet Eingaben des Clients und stellt für ihn statische Ausgaben, meist in Form von HTML-Seiten, bereit.

Bedeutung:

Da immer mehr Menschen Zugang zum WWW haben¹²⁷, gewinnt dieses Netz zunehmend an Bedeutung und somit auch das Protokoll HTTP. Aus den Problemen, die sich bei der Einbindung von Datenbanken ins WWW stellen, wird deutlich, dass HTTP im Gegensatz zu Z39.50 kein Protokoll ist, das ursprünglich auf die Suche in Datenbanken ausgerichtet war. Trotzdem spielt HTTP in verteilten Informationssystemen heute eine wichtige Rolle. Dies wird in den folgenden Abschnitten deutlich.

5.3 Web-Services

Web-Services sind Dienste, die über das WWW angeboten werden. Die Datenübertragung erfolgt demnach HTTP-basiert. Sie sind darauf ausgerichtet, von Maschinen genutzt zu werden und nicht mit Hilfe eines Browsers

¹²⁷ Vgl. Eimeren, Birgit von; Heiz Gerhard und Beate Frees: Internetverbreitung in Deutschland: unerwartet hoher Zuwachs. In *Media Perspektiven* (2003) 5, S. 339

direkt von Menschen¹²⁸. Das bedeutet, dass Standards für den Zugriff auf die Dienste verfügbar sein müssen. Die meisten Web-Services beruhen neben HTTP auf den folgenden vier Standards: SOAP, WSDL, UDDI und WS-Inspection¹²⁹. Auf die beiden zuletzt genannten Standards wird nicht weiter eingegangen, da sie nicht für den Aufbau von verteilten Systemen verantwortlich sind, sondern für das Auffinden von diesen im WWW. Die anderen beiden Standards SOAP und WSDL werden im folgenden Abschnitt betrachtet. Anschließend werden mit ZING/SRW und OAI-PMH zwei konkrete Web-Services vorgestellt.

5.3.1 Standards: SOAP und WSDL

SOAP ist die Abkürzung für Simple Object Access Protocol und ist ein Protokoll für den Nachrichtenaustausch¹³⁰. Es enthält selbst keine Anfragetypen, sondern setzt auf Datenübertragungsprotokollen, in der Regel auf HTTP, auf. Bei der Nutzung von HTTP wird eine Anfrage mit dem Anfragetyp POST gemacht. SOAP ermöglicht eine XML-basierte standardisierte Übertragung von Anfragen und Antworten. Die Struktur für die übertragenen Nachrichten ist in einem XML-Schema, dem sogenannten SOAP-Envelope, festgelegt. Der Envelope besteht aus einem "Header", der allerdings nicht zwingend vorkommen muss, und einem "Body". Im Body wird eine Nachricht übermittelt, der Header kann zusätzliche, die Nachricht betreffende Informationen übertragen, wie zum Beispiel Benutzerdaten für den Zugriff. Das Protokoll definiert neben der Struktur der übertragenen Nachrichten in sogenannten "SOAP encoding rules" die Art der übertragenen Daten und vieles mehr.

WSDL bedeutet Web Service Description Language. Es ist eine Sprache für die Beschreibung von Diensten¹³¹, mit Hilfe derer einzelne Codierungen für den Austausch zwischen einem Server und einem Client genau festgelegt sind. Dazu gehört zum Beispiel die Angabe, wie ein Autor oder ein Titel gekennzeichnet werden und welche Werte diese enthalten dürfen, beispielsweise Buchstaben oder Zahlen. Darüber hinaus enthält die WSDL Angaben über die Art der Verbindung zwischen Servern und Clients, über den Zugang zu diesen und weitere Festlegungen. Wenn ein Server und ein Client Nachrichten austauschen, müssen sie sich auf die gleiche WSDL beziehen, damit ein Austausch möglich ist. Verschiedene Web-Services haben in der Regel ihre eigene, angepasste WSDL, die im SOAP-Body aufgerufen wird.

Bedeutung:

¹²⁸ Vgl. Web Services. Markus Keidl; Alfons Kemper; Stefan Seltzsaam [u.a.]. In: Web & Datenbanken: Konzepte, Architekturen, Anwendungen. Hrsg.: Erhard Rahm und Gottfried Vossen. 1. Heidelberg: dpunkt-Verl 2003, S. 294

¹²⁹ Vgl. ebd., S. 295

¹³⁰ Vgl. Simple Object Access Protocol (SOAP) 1.1: W3C Note. Online: <http://www.w3.org/TR/SOAP/> [Stand: 8.5.2000, Abrufdatum: 9.10.2003, Datei: soap_1.htm]

Wie bereits gesagt, sind SOAP und WSDL grundlegende Standards für Web-Services. Sie gewinnen zunehmend an Bedeutung. Das neue verteilte Informationssystem vascoda¹³² ist zum Beispiel ein typischer Web-Service, der auf SOAP und WSDL beruht. Auch die Digitale Bibliothek NRW will ihre Technik mehr an Web-Services anpassen¹³³.

5.3.2 ZING/SRW

Die bereits im Zusammenhang mit ZING/SRU erwähnte Arbeitsgruppe ZING entwickelte mit ZING/SRW einen Web-Service, der auf Funktionen und Konzepten von Z39.50 aufbaut¹³⁴. Vereinfacht gesagt, ist dies Z39.50 für das WWW. ZING/SRW baut auf der Semantik von Z39.50 auf und legt einen Standard für eine Suche fest. Es besitzt einen Anfragetyp (Search-Retrieve), mit dem Dokumente gesucht werden können¹³⁵. Die Anfragen werden über HTTP entweder mit dem Befehl POST als SOAP-Message geschickt oder mit GET in der URI. Bei letzterem wird ZING/SRU als ein Teil von ZING/SRW eingesetzt¹³⁶. Antworten werden in XML ausgegeben. Damit Anfragen und Antworten einheitlich übertragen werden, gibt es eine WSDL für ZING¹³⁷.

Da ZING/SRW erst Ende 2002 verabschiedet wurde, ist es noch nicht weit verbreitet¹³⁸. Von den Systemen, die dieser Arbeit zu Grunde liegen, verwendet iPort ZING/SRU¹³⁹.

5.3.3 Protokoll für Harvesting - OAI-PMH

OAI-PMH ist die Abkürzung für Open Archives Initiative - Protocol for Metadata Harvesting¹⁴⁰. Es wurde im Jahre 2001 von der Open Archives Initiative, einem Zusammenschluss von Informationsanbietern, als ein Standard für das Harvesting herausgebracht mit dem Ziel, die Verbreitung von elektronischen Dokumenten zu vereinfachen¹⁴¹. Das Protokoll ist Client-

¹³¹ Vgl. Web Services, S. 295

¹³² S. Kapitel 6.6

¹³³ S. Kapitel 6.4

¹³⁴ SRW-Search/Retrieve Web Service: Z39.50 International: next generation: background. Hrsg.: Z39.50 Maintenance Agency. Online: <http://www.loc.gov/z3950/agency/zing/srw/> [Stand: 25.11.2002, Abrufdatum: 9.10.2003, Datei: srw_1.htm]

¹³⁵ Vgl. Dovey, Matthew J.: JISC Information Environment Service Level Descriptions. Online: <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/service-description/study/jiscie-sld.pdf> [Stand: 9.4.2003, Abrufdatum: 9.10.2003, Datei: protokolle.pdf], S. 9

¹³⁶ In vielen Publikationen werden ZING/SRW und ZING/SRU als zwei verschiedene Dienste genannt; auf den Seiten des ZING erscheint es jedoch als Unterdienst, weshalb diese Darstellung hier verwendet wird.

¹³⁷ Vgl. ZING SRU/SRW WSDL Specification, version 1.0. Online: <http://www.loc.gov/z3950/agency/zing/srw/srw-bindings.wsdl> [Stand: 2.12.2002, Abrufdatum: 21.10.2003, Datei: srw_wsdl.xml]

¹³⁸ SRW-Search/Retrieve Web Service: Z39.50 International: next generation: implementors. Hrsg.: Z39.50 Maintenance Agency. Online: <http://www.loc.gov/z3950/agency/zing/srw/implementors.html> [Stand: 5.6.2003, Abrufdatum: 14.10.2003, Datei: srw_2.htm]

¹³⁹ Vgl. Diedrichs, Reiner: iPort als Portal für Bibliotheken im GBV. Online: <http://www.gbv.de/du/pdf/mupo.pdf> [Stand: 27.05.2003, Abrufdatum: 07.10.2003, Datei: gbv.pdf], S. 4

¹⁴⁰ Vgl. The Open Archives Initiative Protocol for Metadata Harvesting, version 1.1

¹⁴¹ Vgl. Lagoze, Carl und Herbert Van de Sompel: The Open Archives Initiative

Server basiert. Es ist für Broker-Systeme erstellt worden, damit diese mit Hilfe eines Standards zentrale Indizes aufbauen können. So kommt der Endnutzer mit diesem Protokoll nicht in Berührung. Allerdings gibt es inzwischen auch Ansätze, die für die Abfrage von Datenquellen über OAI eine Benutzeroberfläche bereitstellen¹⁴².

Anfragetypen, Ausgabe:

Als Web-Service setzt das OAI-PMH auf HTTP auf. Anfragen werden entweder mit GET oder POST übertragen. Das OAI-PMH wird für die Nachrichtenübermittlung genutzt. Es stellt sechs Anfragetypen zur Verfügung, von denen hier vier vorgestellt werden.

Mit "GetRecord" wird der Metadatensatz zu einem bestimmten Dokument angefordert. Dafür ist es wichtig, dass Datenbestände eindeutig identifiziert sind, was mit Hilfe der "unique identifiers" geschieht. "ListRecords" führt ein selektives Harvesting durch. Mit "Identify" werden Informationen über eine Datenbank angefordert, von der Metadaten gewonnen werden, und "ListMetadataFormats" ermöglicht es einem Client, Informationen über die unterstützten Metadatenformate zu erhalten. Es können verschiedene Datenformate unterstützt werden, meist wird jedoch Dublin Core verwendet. Was mit Metadaten innerhalb des Broker-Systems nach dem Harvesting geschieht, liegt außerhalb des OAI-PMH.

Es gibt eine Anzahl von weiteren Abläufen, die für das Harvesting wichtig sind, jedoch nicht im OAI-PMH festgelegt sind. Dazu gehört zum Beispiel die Zugangskontrolle¹⁴³.

Codierung von Nachrichten:

Nachrichten, die bei OAI-PMH-Anfragen übertragen werden, können zum Beispiel unique identifiers sein oder Angaben für das selektive Harvesting. In welcher Form diese Angaben zu machen sind, ist im Protokoll festgelegt.

Antworten werden XML-codiert ausgegeben. Ein Schema dafür ist im Protokoll festgelegt. Für die Verständigung zwischen Clients und Servern muss eine spezielle OAI-PMH-Schnittstelle implementiert sein. Ein Server muss dafür konfiguriert sein, OAI-PMH-Anfragen entgegenzunehmen und an die entsprechende Software weiterzugeben, die die Anfragen auswerten kann¹⁴⁴.

¹⁴² Vgl. Sompel, Herbert van de; Jeffrey A. Young und Thomas B. Hickey: Using the OAI-PMH...Differently. In: D-Lib Magazine 9 (2003) 7/8. Online: <http://www.dlib.org/dlib/july03/young07young.html> [Stand: 8/2003, Abrufdatum: 20.10.2003, Datei: oaipmh_2.htm]

¹⁴³ Vgl. Lynch, Clifford A: Metadata Harvesting and the Open Archives Initiative.

¹⁴⁴ Vgl. The Open Archives Initiative Protocol for Metadata Harvesting, version 1.1

Bedeutung:

Das OAI-PMH hat sich, obwohl es noch nicht so lange besteht, schon relativ breit durchgesetzt¹⁴⁵. Es wird nicht nur für das Harvesting von bibliografischen Daten verwendet, sondern zum Beispiel auch dafür, Informationen über Datenquellen an Broker weiterzugeben¹⁴⁶.

5.4 Einsatz von Protokollen in verteilten Informationssystemen

Nach der Vorstellung der einzelnen Protokolle und Verfahren geht es nun darum, wie diese Protokolle einzeln oder in Verbindung mit anderen in verteilten Informationssystemen eingesetzt werden.

Für die Verbindung zwischen einem menschlichen Benutzer und dem Broker werden meist HTTP und HTML eingesetzt, da für einen Benutzer damit ein Zugang mit Hilfe eines Web-Browsers möglich ist. Wenn Maschinen als Benutzer auf Broker zugreifen, können dabei die gleichen Protokolle zum Einsatz kommen, die auch zwischen Brokern und Providern verwendet werden. Der Einsatz dieser Protokolle zwischen Brokern und Providern wird nun genauer betrachtet. Der Fall, dass alle Provider eine einheitliche Schnittstelle für den Zugriff auf ihre Datenbestände zur Verfügung stellen, wird getrennt dargestellt von dem Fall, wo eine individuelle Anpassung notwendig ist. Der Einsatz von Gateways wird ebenfalls gesondert betrachtet.

5.4.1 Einheitlicher Zugriff

Bei einem einheitlichen Zugriff müssen ein Server und ein Client die gleiche Schnittstelle besitzen, um Daten auswerten zu können. Für verteilte Informationssysteme bedeutet dies, dass alle Provider die gleiche Schnittstelle zur Verfügung stellen müssen. Die in diesem Kapitel bereits vorgestellten Protokolle und Mechanismen werden zum großen Teil für diesen einheitlichen Zugriff eingesetzt, da sie Standards darstellen. Sie werden nun nochmals zusammenfassend aufgezählt.

Das im Bibliothekswesen noch am weitesten verbreitete Protokoll ist Z39.50. Der Einsatz dieses Protokolls bedeutet für die Provider, dass sie alle ein Z39.50-Target besitzen müssen. Daneben gibt es die HTTP-basierten Verfahren. An erster Stelle seien Web-Services genannt. Es gibt Web-Services, die allgemeine Standards sind wie das OAI-PMH und ZING/SRW mit ZING/SRU. Daneben kann für verteilte Informationssysteme jedoch mit

¹⁴⁵ Vgl. Müller, Uwe und Klotz-Berendes, Bruno: Tutorium: die Open Archive Initiative. Online: http://eldorado.uni-dortmund.de:8080/bib/2003/workshop_oai.html;internal&action=buildframes.action [Stand: 8.5.2003, Abrufdatum: 9.10.2003, Datei: oaipmh_3.pdf]

¹⁴⁶ Vgl. zum Beispiel Renardus

Hilfe von SOAP und WSDL ein eigener Web-Service definiert werden. Dies ist zum Beispiel der Fall bei vascoda¹⁴⁷.

Neben den erwähnten Protokollen und Mechanismen gibt es noch viele weitere Möglichkeiten des einheitlichen Zugriffs. Andy Powell, der Leiter des Teams "Distributed Systems and Services" von UKOLN¹⁴⁸, veröffentlichte 1999 ein Schaubild¹⁴⁹, in dem weitere Protokolle aufgeführt sind, die allerdings nicht alle für die Suche in Datenbanken eingesetzt werden, sondern auch für andere Funktionen.

5.4.2 Gateways

Es wurde deutlich, dass zwischen den einzelnen Providern und dem Broker in der Regel andere Protokolle zum Einsatz kommen als zwischen dem Broker und Benutzern. Der Broker hat deshalb die Aufgabe, zwischen verschiedenen Protokollen der Benutzer und Provider zu vermitteln. Damit kommt ihm die Aufgabe eines Gateways zu. Er besitzt zwei Schnittstellen und ist in der Lage, Nachrichten von einer Schnittstelle an die andere weiterzugeben und umzuwandeln. Im Bibliothekswesen oft eingesetzt wird zum Beispiel ein WWW-Z39.50-Gateway. Der Aufbau eines solchen Gateways ist durch das Projekt „ZACK“, das in Kapitel 6.2 vorgestellt wird, realisiert worden.

5.4.3 Individuelle Anpassung

Eine individuelle Anpassung von Schnittstellen zwischen einem Broker und Providern ist entweder notwendig, wenn es sich um eine HTTP/HTML-Schnittstelle handelt, die Datenbanken über verschiedene serverseitige oder clientseitige Verfahren an das WWW anbindet, oder wenn von einem Broker aus nicht nur Provider mit einer bestimmten Standardschnittstelle aufgerufen werden, sondern Provider mit verschiedenen Schnittstellen.

Im erstgenannten Fall tritt das Problem auf, dass bei der Einbindung von Datenbanken über HTTP/HTML zum Teil individuelle Codierungen verwendet werden. Ein Benutzer weiß durch die Benutzeroberfläche oder die Kommandosprache, wie er Daten einzugeben hat. Zum Beispiel ist ein Eingabefeld, in das Autorennamen für die Suche eingegeben werden können, in einer grafischen Benutzeroberfläche entsprechend beschriftet. Problematisch wird es, wenn Maschinen mit individuellen Codierungen von WWW-Datenbanken umgehen müssen, da sie ihre Daten nicht ohne weiteres auf ein individuelles System abbilden können. Dafür müssen Anpassungen an die verschiedenen Codierungen gemacht werden. Dies geschieht zum Beispiel beim KVK mit Hilfe von „Strukturbeschreibungsdateien“¹⁵⁰, die Über-

¹⁴⁷ S. Kapitel 6.6

¹⁴⁸ Vgl. Keane, Shirley: UKOLN staff. Online: <http://www.ukoln.ac.uk/ukoln/staff/> [Stand: 9.10.2003, Abrufdatum: 18.10.2003, Datei: ukoln_2.htm]

¹⁴⁹ Powell, Andy: DNER portal architecture, 1999. Online: <http://www.rdn.ac.uk/publications/mia/> [Stand: 20.6.2002, Abrufdatum: 14.10.2003, Datei: diagramm.htm]

¹⁵⁰ Dierolf, Uwe und Michael Mönnich: Karlsruher Virtueller Katalog (KVK): Neue Dienstleistung im World Wide Web. In: Bibliotheksdienst 30 (1996) 8/9, S. 1397

setzungstabellen enthalten. Der KVK hält für jedes Zielsystem eine solche Datei vor¹⁵¹.

Wenn von einem Broker-System nicht nur Provider angesprochen werden, die alle die gleiche Schnittstelle besitzen, so muss für das Ansprechen jedes Providers ein eigenes Programm geschrieben werden, das zum einen in der Lage ist, die Anfrageschnittstelle eines Zielsystems anzusprechen, zum anderen, verschiedene Codierungen, wie zum Beispiel Autorenfelder, aufeinander abzubilden. Die Digitale Bibliothek NRW arbeitet auf diese Weise und benutzt für das Ansprechen kleine Programme, sogenannte PERL-Skripte¹⁵². Der Vorteil der individuellen Anpassung ist der, dass einzelne Datenbanken auf keine Standards und Vorgaben Rücksicht nehmen müssen¹⁵³ und dass viele verschiedene Schnittstellen, zum Beispiel auch direkt SQL-Schnittstellen (structured query language), eingebunden werden können. Der Nachteil dieses Verfahrens ist, dass dies sehr aufwändig sein kann¹⁵⁴.

¹⁵¹ S. Kapitel 6.1

¹⁵² S. Kapitel 6.4

¹⁵³ Vgl. Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment

¹⁵⁴ Vgl. Pieper, Dirk und Friedrich Summann: Die Entwicklung des Zugangssystems der Digitalen Bibliothek NRW. In: NfD 50 (1999), S. 400

6 VERSCHIEDENE REALISIERUNGEN IN DEUTSCHLAND

In diesem Kapitel werden einige für das deutsche Bibliotheks- und Informationswesen bedeutende verteilte Informationssysteme vorgestellt. Im Einzelnen sind dies der KVK (Karlsruher Virtueller Katalog), ZACK, die KOBV-Suchmaschine (Suchmaschine des Kooperativen Bibliotheksverbunds Berlin-Brandenburg), die Digitale Bibliothek Nordrhein-Westfalen, das DIMDI (Deutsches Institut für Medizinische Dokumentation und Information), vascoda und Renardus. Es wurden, bis auf Renardus, nur deutsche Systeme ausgewählt. Da Renardus als europäisches Angebot zu einem Teil auch deutsch ist, wird es wegen seiner interessanten Funktionalitäten ebenfalls vorgestellt. Die einzelnen Systeme unterscheiden sich in ihrer Architektur, in der Verwirklichung verschiedener Dienste, in den verwendeten Protokollen, in unterstützten Schnittstellen und in ihren Einsatzbereichen. Damit bietet dieses Kapitel einen praktischen Überblick über sehr unterschiedliche Realisierungsmöglichkeiten.

Bei der Vorstellung wird auf die in Kapitel 2 bis 5 behandelten Punkte eingegangen. Der Schwerpunkt bei den aufgeführten Funktionen von verteilten Informationssystemen liegt, der Arbeit entsprechend, auf der Suche. Die Rechercheoberfläche wird als Voraussetzung für die Suche ebenfalls ausführlicher behandelt. Auf weitere Funktionen wird nicht vollständig eingegangen, sondern es wird nur auf einzelne interessante Angebote eingegangen.

6.1 KVK¹⁵⁵

Der KVK (Karlsruher Virtueller Katalog) ist eine Metasuchmaschine über verschiedene Bibliothekskataloge, Verbunddatenbanken, die Zeitschriften-datenbank und Buchhandelsverzeichnisse. Er wurde von der Universitätsbibliothek Karlsruhe und der dortigen Fakultät für Informatik entwickelt und im Juli 1996 in Betrieb genommen¹⁵⁶. Das Angebot des KVK steht kostenlos zur Verfügung. In einem Entwicklungsbericht des KVK von Ende 2002 heißt es, „dass der KVK nach wie vor eines der wichtigsten bibliographischen Rechercheinstrumente in Deutschland [sei]“¹⁵⁷. Nutzungsstatisti-

¹⁵⁵ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.1 auf der Erkundung des Informationssystems durch die Autorin von Anfang September 2003. Zugang: „Homepage des KVK“. Online: <http://kvk.uni-karlsruhe.de> [Abrufdatum: 16.10.2003, Datei: kvk_2.htm]

¹⁵⁶ Dierolf, Uwe und Michael W. Mönlich: Mehrwertdienste durch Virtuelle Kataloge: 5 Jahre Karlsruher Virtueller Katalog. In: B.I.T. Online 4 (2001) 3, S. 261

¹⁵⁷ Vgl. Dierolf, Uwe: KVK goes Salt Lake City. In: B.I.T. Online 5 (2002) 4, S. 336

ken unterstützen diese Aussage¹⁵⁸. Aus diesen Statistiken wird auch deutlich, dass der KVK vor allem von Bibliotheken genutzt wird.

Architektur

Der KVK führt eine verteilte Suche durch¹⁵⁹. Eine nicht abgeschlossene Suche wird durch einen Timeout beendet, dessen Dauer manuell vom Benutzer eingestellt werden kann. Die Trefferliste wird kontinuierlich aufgebaut.

Funktionen¹⁶⁰

Rechercheoberfläche: Der KVK bietet eine Oberfläche für die Suche an und eine für die Kurzanzeige. Von der Kurzanzeige gelangt der Benutzer für die Vollanzeige direkt auf die Originaloberfläche der entsprechenden Datenquellen. Die Rechercheoberfläche ist einheitlich. Für die Kurzanzeige können zwei Ausgabeformen gewählt werden (neu und klassisch). In der klassischen Variante werden die Suchergebnisse zum Teil in ein einheitliches Format gebracht, teilweise werden sie in dem Format der Kurzanzeige der jeweiligen Datenbank angezeigt. Die Kurzanzeige für die neue Version ist einheitlich.

Suche:

- Auswahl von Datenquellen: Der Benutzer gibt in der Suchmaske an, welche Datenbanken durchsucht werden sollen.
- Sucheinstiege, Feldinhalte, Retrievalfunktionen: Für die Suchfelder wurde pauschal ein größter gemeinsamer Nenner aller beteiligter Datenquellen gebildet. „Pauschal“ bedeutet hier, dass es kein reiner größter gemeinsamer Nenner ist, da auch Suchfelder angeboten werden, die in manchen Datenquellen nicht oder nur eingeschränkt verwendbar sind. Dies wird nicht auf den ersten Blick sichtbar, sondern der Benutzer muss dazu den Hilfe- und Informationstext aufrufen. Absprachen über Felder und deren Inhalte zwischen den einzelnen Providern gibt es nicht, sondern der KVK nutzt das, was er vorfindet¹⁶¹. Allerdings sind die einzelnen Bibliothekskataloge durch die Verwendung gemeinsamer Standards wie beispielsweise RAK (Regeln für die alphabetische Katalogisierung), die RSWK (Regeln für die Schlagwortkatalogisierung) und die SWD (Schlagwortnormdatei) bis zu einem gewissen Grad aufeinander abge-

¹⁵⁸ Vgl. Dierolf, Uwe und Michael W. Mönnich: Mehrwertdienste durch Virtuelle Kataloge: 5 Jahre Karlsruher Virtueller Katalog, S. 262

¹⁵⁹ Vgl. Dierolf, Uwe: Karlsruher Virtueller Katalog mit über 30 Millionen Büchern und Zeitschriften: bundesweit erstmalige Dienstleistung der UB Karlsruhe.
Online: <http://www.uni-karlsruhe.de/~presse/Veroeffentlichungen/Unikath/Unikath96/ub.html>
[Stand: 27.11.1996, Abrufdatum: 13.10.2003, Datei: kvk_1.htm]

¹⁶⁰ Vgl. für diesen Abschnitt auch: „Suchhilfe KVK“.
Online: http://www.ubka.uni-karlsruhe.de/hylib/kvk_help.html [Abrufdatum: 16.10.2003, Datei: kvk_3.htm]

¹⁶¹ Vgl. Dierolf, Uwe und Michael W. Mönnich: KVK - Karlsruher Virtueller Katalog: zwei Jahre virtuell, S. 159f

stimmt. An Retrievalfunktionen steht ein größter gemeinsamer Nenner zur Verfügung (Trunkierung und der Boolesche Operator UND)¹⁶².

- Ausgabe der Kurztrefeferliste: Treffer werden in der Kurzanzeige nach Fundort sortiert unabhängig voneinander hintereinander aufgelistet, meist in einem einheitlichen Format. Es findet keine Dublettenerkennung statt.

Weitere Funktionen: Für die Vollanzeige kann vom KVK aus, je nach Angebot des jeweiligen Katalogs, das Format der Vollanzeige gewählt werden (als Alternative MAB). Eine Sonderform des KVK, die bisher nur für die Universitätsbibliotheken Karlsruhe und Mannheim besteht, ermöglicht es, Fernleihbestellformulare vom KVK aus abzuschicken¹⁶³. Welche weiteren Funktionen nach einer Suche zur Verfügung stehen, hängt von dem jeweiligen Angebot der Datenquelle ab, auf deren Oberfläche der Benutzer durch die Vollanzeige geleitet wird.

Technische Realisierung¹⁶⁴

Vom Benutzer wird der KVK mit Hilfe eines WWW-Browsers über das Internet genutzt. Die Datenquellen, auf die über den KVK zugegriffen wird, müssen über das WWW, das heißt HTTP- und HTML-basiert, abgefragt werden können. Der KVK greift mit Hilfe eines CGI-Skriptes auf die WWW-Schnittstellen der Datenquellen zu. Die Anpassung einer Suchanfrage an eine Datenquelle und der Ergebnisse an den KVK geschieht mit Hilfe eines CGI-Skriptes.

Ein Problem, das sich beim KVK stellt, ist der Umgang mit sitzungsorientierten WWW-Katalogen¹⁶⁵. Diese setzen, wie in Kapitel 5.2 beschrieben, voraus, dass zum Beispiel Cookies auf dem Client installiert werden können. Nachdem für den KVK zuerst eine Realisierungsvariante mit der Einbeziehung von Z39.50-Schnittstellen gewählt wurde¹⁶⁶, ist der KVK nun in der Lage, Sitzungen zu verwalten¹⁶⁷.

6.2 ZACK¹⁶⁸

ZACK ist ein System zur verteilten Suche, das im Rahmen einer Diplomarbeit im Jahre 1999 in Zusammenarbeit mit der Technischen Universität und dem Konrad-Zuse-Zentrum für Informationstechnik Berlin entwickelt wur-

¹⁶² Vgl. ebd., S. 160

¹⁶³ Vgl. Dierolf, Uwe und Michael W. Mönnich: Mehrwertdienste durch Virtuelle Kataloge: 5 Jahre Karlsruher Virtueller Katalog, S. 264f.

¹⁶⁴ Vgl. für diesen Abschnitt: Dierolf, Uwe und Michael Mönnich: Karlsruher Virtueller Katalog (KVK): neue Dienstleistung im World Wide Web. S. 1396ff.

¹⁶⁵ Vgl. Dierolf, Uwe und Michael W. Mönnich: KVK - Karlsruher Virtueller Katalog: zwei Jahre virtuell, S. 160ff.

¹⁶⁶ Vgl. ebd.

¹⁶⁷ Vgl. Dierolf, Uwe: KVK goes Salt Lake City, S. 336

¹⁶⁸ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.2 auf der Erkundung des Informationssystem durch die Autorin von Anfang September 2003. Zugang: „Homepage von ZACK: deutsche Bibliotheken“. Online: <http://opus.tu-bs.de/zack/cgi/zack?ds-de> [Abrufdatum: 16.10.2003, Datei: zack_1.htm]

de¹⁶⁹. Es bietet eine Oberfläche für die Suche in deutschen Bibliothekskatalogen und Verbunddatenbanken und eine für die Suche in internationalen Bibliotheken. Vorgestellt wird hier die Version mit dem deutschen Angebot. Der Zugang ist kostenfrei. Das System wird heute von einigen Bibliotheken als Datenlieferant für die retrospektive Katalogisierung genutzt¹⁷⁰.

Architektur

ZACK führt eine verteilte Suche durch¹⁷¹. Eine Suche wird durch einen vor-eingestellten Timeout beendet. Die Trefferliste wird kontinuierlich aufgebaut, die einzelnen Treffer erscheinen jedoch auf einmal, da sie vor der Ausgabe bearbeitet werden (s.u.).

Funktionen¹⁷²

Rechercheoberfläche: Eine einheitliche Oberfläche von ZACK gibt es sowohl für die Recherche als auch für die Kurz- und Vollanzeige.

Suche:

- Auswahl von Datenquellen: Der Benutzer gibt in der Suchmaske an, welche Datenbanken durchsucht werden sollen.
- Sucheinstiege, Feldinhalte, Retrievalfunktionen: Für die Suchfelder und deren Inhalte gilt das gleiche wie beim KVK, allerdings ist es für einen Benutzer noch schwerer als beim KVK zu erkennen, dass in einzelnen Datenbanken manche Felder nicht suchbar sind. Darüber hinaus ist eine verteilte Registersuche möglich. An Retrievalfunktionen (alle drei Booleschen Operatoren, Rechtstrunkierung) steht ein größter gemeinsamer Nenner zur Verfügung.
- Ausgabe der Kurztrefferliste: Die Treffer in der Kurzanzeige werden in einem einheitlichen Format hintereinander aufgelistet, wobei Dubletten erkannt werden. Bei diesen wird in der Ausgabe nach bestimmten Kriterien ein bestehender Datensatz ausgewählt und angezeigt¹⁷³. An die Treffer werden die jeweiligen Fundorte angehängt, die einen Link zur Anzeige des Datensatzes im MAB2- oder MARC-Format darstellen. Ob Werke oder Ausgaben als Dubletten gelten, kann in der Suchmaske manuell eingestellt werden.

¹⁶⁹ Vgl. Schneider, Wolfram: Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls

¹⁷⁰ Zum Beispiel nutzen es die Stadtbücherei Altena und das Goethe-Institut in München. (Vgl. Gusik, Antonius <inetbib@ub.uni-dortmund.de> „Re(2): KVK und Digibib“ 10.9.2003.

Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22248.html> [Abrufdatum: 13.10.2003, Datei: inet_4.htm] sowie Allers, Heinrich <inetbib@ub.uni-dortmund.de> „Fremddaten für mehrbändige Werke (war: KVK und Digibib)“ 10.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22356.html> [Abrufdatum: 13.10.2003, Datei: inet_5.htm])

¹⁷¹ Vgl. Schneider, Wolfram: Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls, S. 3

¹⁷² Vgl. für diesen Abschnitt auch: „Suchhilfe ZACK“. Online: <http://opus.tu-bs.de/zack/help.html> [Abrufdatum: 16.10.2003, Datei: zack_2.htm]

¹⁷³ Vgl. Schneider, Wolfram: Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls, S. 63ff.

Technische Realisierung¹⁷⁴

Der Benutzer greift auf die ZACK-Oberfläche mit Hilfe eines WWW-Browsers zu. Die einzelnen Provider müssen eine Z39.50-Schnittstelle besitzen. Die Besonderheit von ZACK liegt darin, dass zwischen Benutzer und Provider ein WWW-Z39.50-Gateway gestellt ist, das es ermöglicht, dass Benutzer auch ohne einen eigenen Z39.50-Client auf Z39.50-Datenbanken zugreifen können. Das von ZACK entwickelte WWW-Z39.50-Gateway ist Bestandteil einiger komplexer Systeme: es wird zum Beispiel in der KOBV-Suchmaschine und in der Digitalen Bibliothek NRW eingesetzt¹⁷⁵.

6.3 KOBV-Suchmaschine¹⁷⁶

Der KOBV (Kooperativer Bibliotheksverbund Berlin Brandenburg) ist der Bibliotheksverbund von Berlin und Brandenburg¹⁷⁷. Wie alle Verbände erfüllt er unter anderem die Aufgabe, die Bestände der einzelnen Bibliotheken aus dem Verbund nachzuweisen. Dies wird im KOBV mit Hilfe der KOBV-Suchmaschine geleistet. Sie bietet einen einheitlichen Zugang zu den Bibliothekskatalogen des Verbundes und ermöglicht es auch, in anderen bedeutenden deutschen Bibliothekskatalogen zu recherchieren. Offiziell gestartet wurde die KOBV-Suchmaschine im Jahre 1999¹⁷⁸. Sie beruht auf der Portalsoftware MetaLib der Firma ExLibris. Die Nutzung ist kostenlos.

Zurzeit wird als Erweiterung der Suchmaschine ein KOBV-Informationsportal entwickelt¹⁷⁹. Darin sind nicht nur Bibliothekskataloge eingeschlossen, sondern auch verschiedene andere elektronische Datenbestände, wie zum Beispiel elektronische Zeitschriften.

Architektur:

Die Architektur der KOBV-Suchmaschine ist eine Mischform. Sie besitzt einen zentralen Index, in dem die Bestände eines Teils der am KOBV beteiligten Bibliotheken nachgewiesen sind, und greift auf die übrigen Bibliothekskataloge mit einer verteilten Suche zu¹⁸⁰. Der zentrale Index ist inhaltlich einer Verbunddatenbank sehr ähnlich. Allerdings ist die Erstellung der gemeinsamen Datenbank organisatorisch völlig verschieden. Der Aufbau und die Pflege des Index läuft nicht über das Standardprotokoll OAI-PMH, sondern es wird dafür eine proprietäre Lösung eingesetzt (S. 24). Die ver-

¹⁷⁴ Vgl. für diesen Abschnitt ebd., S. 12ff.

¹⁷⁵ S. Kapitel 6.3 und 6.4

¹⁷⁶ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.3 auf der Erkundung des Informationssystems durch die Autorin von Anfang September 2003. Zugang: „Homepage der KOBV-Suchmaschine“. Online: <http://search.kobv.de> [Abrufdatum: 16.10.2003, Datei: kobv_6.htm]

¹⁷⁷ Vgl. „Homepage des KOBV“. Online: <http://www.kobv.de> [Abrufdatum: 16.10.2003, Datei: kobv_5.htm]

¹⁷⁸ Vgl. Der Kooperative Bibliotheksverbund Berlin Brandenburg, S. 3

¹⁷⁹ Entwicklungsprojekt „KOBV Informationsportal“: Teilprojekt der 1. Stufe: Monika Kuberek; Monika Lill; Stefan Litsche [u.a.]. Online: <http://www.zib.de/kuberek/publications/ZR-01-41.pdf> [Stand: 10/2001, Abrufdatum: 13.10.2003, Datei: kobv_4.pdf]

¹⁸⁰ Vgl. Der Kooperative Bibliotheksverbund Berlin Brandenburg, S. 12ff. Alle Seitenangaben im Abschnitt beziehen sich auf diese Ausgabe.

teilte Suche wird durch einen voreingestellten Timeout beendet. Die KOBV-Suchmaschine wird über den KVK angeboten, allerdings nicht das gesamte Angebot, sondern nur der zentrale Index. Der KOBV-Index wird im KVK als eine Datenquelle behandelt.

Funktionen¹⁸¹

Rechercheoberfläche: Die KOBV-Suchmaschine stellt für die Recherche, die Kurz- und die Vollanzeige eine einheitliche Oberfläche zur Verfügung. Allerdings ist zusätzlich von der Kurz- und von der Vollanzeige aus ein Link zu der Vollanzeige der jeweiligen originalen Datenquelle gelegt, so dass die spezifischen Funktionen und Dienste der beteiligten Bibliotheken genutzt werden können. Die Kurzanzeige besteht aus zwei Teilen: zuerst wird die Anzahl der Treffer pro Datenquelle aufgeführt. Erst, wenn eine bestimmte Datenquelle ausgewählt wird, werden in einer Kurzanzeige die Treffer dieser Quelle sichtbar.

Suche:

- Auswahl von Datenquellen: Der Benutzer wählt die zu durchsuchenden Bibliothekskataloge selbst aus.
- Sucheinstiege, Feldinhalte, Retrievalfunktionen: Es gibt einen gemeinsamen KOBV-Standard¹⁸². Dieser legt bestimmte Felder fest, die verfügbar sein müssen und deren Inhalt normiert eingegeben werden soll. Allerdings gilt der Standard nur für Neueingaben. Es wurde entschieden, Altdaten so zu lassen, wie sie sind¹⁸³. An Retrievalfunktionen stehen die drei Booleschen Operatoren AND, OR, NOT zur Verfügung sowie Trunkierung.
- Ausgabe der Kurztrefnerliste: In der KOBV-Suchmaschine findet bisher noch keine Dublettenerkennung statt. Allerdings ist in den gemeinsamen Index eine Dublettenerkennung eingebaut¹⁸⁴.

Weitere Funktionen: Die KOBV-Suchmaschine bietet einem Benutzer die Möglichkeit, sich bei dem System anzumelden, so dass ein persönliches Benutzerprofil erstellt werden kann. Dieses Angebot steht kostenlos zur Verfügung. Zurzeit wird an einem Modul gearbeitet, das eine Fernleihe über den KOBV anbietet¹⁸⁵.

¹⁸¹ Vgl. für diesen Abschnitt auch: „Suchhilfe KOBV-Suchmaschine“. Online: zugänglich über <http://search.kobv.de> [Stand: 3.2.2003, Abrufdatum: 16.10.2003, Datei (nur Inhalt der Seite): kobv_9.doc]

¹⁸² Kuberek, Monika: Gemeinsamer KOBV-Standard. Online: <http://www.zib.de/kuberek/publications/SC-99-14.pdf> [Stand: 1.6.1999, Abrufdatum: 13.10.2003, Datei: kobv_7.pdf], S. 4.

¹⁸³ Vgl. ebd., S. 5

¹⁸⁴ Vgl. Der Kooperative Bibliotheksverbund Berlin Brandenburg, S. 21

¹⁸⁵ Vgl. Kuberek, Monika; Stefan Lohrum und Raluca Rascanu: Endnutzerbasierte Online-Fernleihe im KOBV: Kurzbeschreibung. In: ZIB-Report 02-55 (2002). Online: <http://www.zib.de/kuberek/publications/ZR-02-55.pdf> [Stand: 12/2002, Abrufdatum: 16.10.2003, Datei: kobv_8.pdf]

Technische Realisierung¹⁸⁶

Der Benutzer kann auf die KOBV-Suchmaschine entweder über das WWW mit Hilfe eines Web-Browser zugreifen oder von einer Z39.50-Schnittstelle ausgehend. Der Zugriff von einer Z39.50-Schnittstelle wurde für Bibliotheken eingerichtet und dafür ein spezielles Verfahren, die „sequentiell verteilte Suche“ (S. 19) entwickelt. Anfangs wurden Provider nur über ihre Z39.50-Schnittstelle eingebunden (S. 13ff). Für die Verbindung zwischen der Z39.50-Schnittstelle der Provider und einer HTTP-basierten WWW-Schnittstelle von Benutzern wird das Gateway ZACK eingesetzt. Heute wird auf Provider nicht mehr ausschließlich über Z39.50 zugegriffen, sondern auch HTTP-basiert mit Hilfe von Web-Services. So wird auf den GBV und die Staatsbibliothek zu Berlin über eine Art ZING/SRU zugegriffen: Anfragen werden als URL codiert und Ergebnisse in XML-Strukturen geliefert¹⁸⁷.

6.4 Digitale Bibliothek NRW¹⁸⁸

Die Digitale Bibliothek NRW (Nordrhein-Westfalen), abgekürzt auch als DigiBib bezeichnet, bietet Zugang zu Bibliothekskatalogen, Zeitschriftendatenbanken, Dokumentlieferdiensten und anderen digitalen Dokumenten. Ihr Angebot ist interdisziplinär. Sie wurde vom Hochschulbibliothekszenrum Nordrhein-Westfalen in Zusammenarbeit mit der UB Bielefeld und den Firmen IHS und axion entwickelt und im Juni 1999 gestartet¹⁸⁹. Sie beruht auf der Portalsoftware IPS (Information Portal Suite) der Firma IHS. Die Nutzung ist kostenfrei, allerdings bietet die DigiBib auch Zugang zu kostenpflichtigen Datenquellen.

Architektur

Die DigiBib besitzt keinen zentralen Index, sondern es findet eine verteilte Suche statt¹⁹⁰. Eine nicht abgeschlossene Suche wird durch einen Timeout beendet. Die Trefferliste wird kontinuierlich aufgebaut.

¹⁸⁶ Vgl. für diesen Abschnitt: Der Kooperative Bibliotheksverbund Berlin Brandenburg, S. 12ff. Alle Seitenangaben im Abschnitt beziehen sich auf diese Ausgabe.

¹⁸⁷ Diese Information stammt aus Telefonat 1 [Datum: 16.10.2001, Gesprächspartner: Stefan Lohrum, Berlin, Bereichsleiter Informationstechnik am Zuse Institut Berlin, Mitarbeit am KOBV]

¹⁸⁸ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.4 auf der Erkundung des Informationssystems durch die Autorin von Anfang September 2003. Zugang: „Homepage der Digitalen Bibliothek NRW, Standort FHB Köln“. Online: <http://www.digibib-nrw.de> [Abrufdatum: 15.10.2003, Datei (screenshot): digibib_2.bmp]

¹⁸⁹ Ollig, Hans: Einführung in DigiBib und Fernleihe. Online: http://www.hbz-nrw.de/produkte_dienstl/DigiBib/Vortraege/22012003/FH_DigiBib_220103.pdf [Abrufdatum: 18.10.2003, Datei: digibib_5.pdf], S. 6ff.

¹⁹⁰ Vgl. Töteberg, Ingrid: Die Digitale Bibliothek - das universale Portal des Wissens. Online: http://www.hbz-nrw.de/produkte_dienstl/DigiBib/ [Stand: 6.12.2002, Abrufdatum: 16.10.2003, Datei: digibib_4.htm]

Funktionen¹⁹¹

Rechercheoberfläche: Die Digitale Bibliothek NRW bietet sowohl für die Recherche als auch für die Kurz- und Vollanzeige eine einheitliche Oberfläche an.

Suche:

- Auswahl von Datenquellen: In der Suchmaske muss ein Benutzer eingeben, welche Datenbanken durchsucht werden sollen.
- Sucheinstiege, Feldinhalte, Retrievalfunktionen: Für die Suchfelder und Feldinhalte gilt im Wesentlichen das Gleiche wie beim KVK, weshalb dies hier nicht nochmals ausgeführt wird. Unterschiede zum KVK liegen darin, dass die einzelnen Datenquellen inhaltlich insgesamt heterogener sind, so dass bei einem einheitlichen Zugriff mehr Inkonsistenz in Kauf genommen werden muss. Trotz der größeren Heterogenität der angebotenen Quellen stehen mehr Retrievalfunktionen als beim KVK zur Verfügung (Trunkierung sowie die drei Booleschen Operatoren UND, ODER, NICHT).
- Ausgabe der Kurztrefeferliste: Die Treffer werden in der Kurzanzeige nach Fundort sortiert in einem einheitlichen Format unabhängig voneinander hintereinander aufgelistet. Es findet demnach keine Dublettenerkennung statt.

Weitere Funktionen: Die Lokalisierung wird mit Hilfe einer Verfügbarkeitsfunktion standortabhängig und somit an einen Benutzer angepasst, durchgeführt¹⁹². Die Anpassung an einen Standort oder an einen Benutzer ist möglich, da ein Benutzer sich einloggen muss, um die Digitale Bibliothek zu nutzen. Meist erfolgt das Einloggen über den Namen einer Bibliothek. Neben dem Einloggen kann eine Benutzererkennung auch über die ID des Benutzerrechners erfolgen¹⁹³. Nach dem Einloggen stehen einem Benutzer in der Suchoberfläche nur die Datenquellen zur Verfügung, zu denen er Zugang hat. Für die Lieferung ist teilweise auch eine Fernleihe über die DigiBib möglich.

Technische Realisierung¹⁹⁴

Vom Benutzer wird die Digitale Bibliothek NRW über einen WWW-Browser genutzt. Provider können sowohl Z39.50-Schnittstellen anbieten als auch über HTTP zugängliche. Z39.50-Datenbanken werden bisher mit Hilfe der Software des WWW-Z39.50-Gateways von ZACK mit einbezo-

¹⁹¹ Vgl. für diesen Abschnitt auch: „Suchhilfe Digitale Bibliothek“. Online: zugänglich über <http://www.digibib-nrw.de> [Abrufdatum: 15.10.2003, Datei (nur Inhalt der Seite): digibib_3.doc]

¹⁹² Vgl. Die Digitale Bibliothek NRW: Technisches Konzept. Margarete Groos; Josef Hardt; Artur Nold [u.a.]. Online: <http://www.ub.uni-bielefeld.de/digibib-nrw/techkon.htm> [Stand: 28.4.1998, Abrufdatum: 13.10.2003, Datei: digibib_1.htm]

¹⁹³ Vgl. Pieper, Dirk und Friedrich Summann: Die Entwicklung des Zugangssystems der Digitalen Bibliothek NRW, S. 399

¹⁹⁴ Sofern nicht anders angegeben, vgl. für diesen Abschnitt: ebd. S. 399ff.

gen¹⁹⁵. Bei HTTP-basierten Schnittstellen werden für die verschiedenen Zielkataloge Suchanfragen und -ergebnisse durch PERL-Skripte umformuliert, die auf die jeweilige Datenbank abgestimmt sind. Die Daten werden in HTML oder XML übertragen¹⁹⁶. Demnächst wird das technische Konzept der DigiBib verändert. Statt ZACK soll ein internes Z39.50-Gateway implementiert werden; daneben soll die Übertragung stärker auf Web-Services ausgerichtet werden¹⁹⁷.

6.5 DIMDI¹⁹⁸

Das DIMDI (Deutsches Institut für Medizinische Dokumentation und Information) hat unter anderem die Funktion eines Hosts, indem es Zugang zu Informationen aus dem Gesamtbereich Medizin anbietet¹⁹⁹. Es wurde 1969, für die Geschichte des Computers also sehr früh, als eine Behörde des damaligen Bundesministeriums für Gesundheit gegründet²⁰⁰. Datenquellen sind über 70 verschiedene Datenbanken, die vorwiegend bibliografische Daten enthalten²⁰¹. Das DIMDI stellt einen Teil der angebotenen Datenbanken kostenlos zur Verfügung, für den Zugang zu den übrigen muss ein Benutzer bezahlen²⁰². Die Datenbanken des DIMDI sind neben dem Angebot der ZBMed (Deutsche Zentralbibliothek für Medizin) in MedPilot eingebunden. MedPilot ist die Virtuelle Fachbibliothek Medizin²⁰³ und wurde im Jahre 2000 gegründet.

Architektur

Das DIMDI greift, im Gegensatz zu allen bisher vorgestellten Systemen, in der Regel nicht auf die originalen Datenquellen zu. Stattdessen hat es die Datenbanken, die abgefragt werden können, zentral bei sich vorliegen²⁰⁴. Die Inhalte der Datenbanken werden von den Datenbankherstellern an das

¹⁹⁵ Diese Angabe stammt aus E-Mail 1 [Datum: 8.9.2003, Verfasser: Peter Mayr, Köln, Mitarbeiter in der Technik der Digitalen Bibliothek im HBZ NRW]

¹⁹⁶ Ebd.

¹⁹⁷ Ebd. Der Zeitpunkt der Umstellung war im September 2003 noch nicht klar. Zu der Zeit wurde noch mit einer Testversion des neuen Systems gearbeitet.

¹⁹⁸ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.5 auf der Erkundung des Informationssystems durch die Autorin von Anfang September 2003. Zugang: „Homepage für die Suche im Angebot des DIMDI“. Online: <http://www.dimdi.de/de/db/recherche.htm> [Abrufdatum: 15.10.2003, Datei: dimdi_6.htm]

¹⁹⁹ „Homepage des DIMDI“. Online: <http://www.dimdi.de> [Abrufdatum: 15.10.2003, Datei: dimdi_3.htm]

²⁰⁰ Vgl. DIMDI Kurzporträt. Hrsg.: DIMDI. Online: <http://www.dimdi.de/de/dimdi/dd/index.htm> [Abrufdatum: 15.10.2003, Datei: dimdi_4.htm]

²⁰¹ Vgl. DIMDI Datenbanken: Benutzerhandbuch. Köln: DIMDI 2002. Online: <http://www.dimdi.de/de/db/dbinfo/dbhandbuchg.pdf> [Abrufdatum: 15.10.2003, Datei: dimdi_5.pdf], S. 6

²⁰² Vgl. „Homepage des DIMDI“

²⁰³ Vgl. „Homepage von MedPilot“. Online: <http://www.medpilot.de> [Abrufdatum: 15.10.2003, Datei (screenshot): medpilot.bmp]

²⁰⁴ Vgl. Medizininformation: umfassend und unabhängig: Deutsches Institut für Medizinische Dokumentation und Information: Jahresbericht 2002. Hrsg.: DIMDI. Online: <http://www.dimdi.de/de/dimdi/public/jahr/jahr2002.pdf> [Stand: 2003, Abrufdatum: 13.10.2003, Datei: dimdi_1.pdf], S. 30

DIMDI geschickt²⁰⁵. Der Host spielt sie in sein speziell entwickeltes Datenbank- und Retrievalsystem ein, wo die einzelnen Daten für eine Verbesserung und Erleichterung der Suche „harmonisiert“²⁰⁶, das heißt möglichst stark vereinheitlicht, werden. Bei der gleichzeitigen Abfrage mehrerer Datenbanken wird innerhalb der Bestände des DIMDI eine verteilte Suche durchgeführt²⁰⁷. Eine Ausnahme stellt die Datenbank des Bibliotheksverbands Bayern (BVB) dar, die nicht beim DIMDI aufliegt, sondern auf die direkt zugegriffen wird²⁰⁸.

Funktionen²⁰⁹

Rechercheoberfläche: Alle Vorgänge, von der Suche bis zur Vollanzeige, laufen über einheitliche Oberflächen des DIMDI. Bei der Kurzanzeige erscheinen nicht direkt die einzelnen Treffer, sondern dem Benutzer wird zuerst die Anzahl der Treffer mitgeteilt.

Suche:

- Auswahl von Datenquellen: Die zu durchsuchenden Datenbanken werden vom Benutzer ausgewählt. Für die Auswahl kann auf fachliche Cluster, sogenannte “Superbases“²¹⁰, zurückgegriffen werden. Als Hilfe für die Auswahl wird für manche Bereiche grips-Index angeboten. Dies ist eine Übersichtssuche, durch die angezeigt wird, wie viele Treffer es für eine Suchanfrage pro Datenbank gibt.
- Sucheinstiege, Feldinhalte, Retrievalfunktionen: Es kann sowohl menügesteuert über eine grafische Benutzeroberfläche gesucht werden (grips-WebSearch) als auch direkt mit einer Kommandosprache (grips-Open). Grips-WebSearch bietet verschiedene Suchmodi an. In dem Expertenmodus steht ein Feld zur Verfügung, in das Befehle in der Kommandosprache eingegeben werden müssen. Je nachdem, welcher Modus gewählt wird, stehen verschiedene Suchfelder und Retrievalfunktionen zur Verfügung. In der Suche über Menüs wird ein größter gemeinsamer Nenner an Suchfeldern angeboten, wohingegen ein Benutzer mit der Kommandosprache freier ist. Zwischen den Datenlieferanten gibt es keine inhaltlichen Absprachen über Feldinhalte; stattdessen erfüllt das DIMDI diese Aufgabe, indem es, wie bereits geschrieben, Daten bei dem Einspielen in grips anpasst.

²⁰⁵ Diese Information stammt aus E-Mail 3 [Datum: 30.9.2003, Verfasser: Susanne Breuer, Köln, Mitarbeiterin des DIMDI, Verantwortliche für die Öffentlichkeitsarbeit]

²⁰⁶ Medizininformation: umfassend und unabhängig, S. 31

²⁰⁷ Diese Information stammt aus E-Mail 4 [Datum: 2.10.2003, Verfasser: Olaf Ritter, Köln, Mitarbeiter des DIMDI]

²⁰⁸ Vgl. Memokarte des Bibliotheksverbands Bayern. Hrsg.: DIMDI.
Online: <http://www.dimdi.de/de/db/dbinfo/dbmemo/1a00ger.pdf> [Stand: 10/2001, Abrufdatum: 13.10.2003, Datei: dimdi_2.pdf]

²⁰⁹ Vgl. für diesen Abschnitt auch: „Suchhilfe DIMDI“. Online: zugänglich über <http://www.dimdi.de/de/db/recherche.htm> [Abrufdatum: 15.10.2003, Datei: dimdi_7.htm]

²¹⁰ DIMDI Datenbanken: Benutzerhandbuch, S. 6

-
- Ausgabe der Kurztrefeferliste: Die Treffer erscheinen in der Kurztrefeferliste nach Datenbanken sortiert. Eine Dublettenerkennung findet bei der Standardsuche meist automatisch statt, in den anderen Suchmodi muss sie extra aufgerufen werden.

Technische Realisierung

Ein Benutzer greift auf das DIMDI meist mit Hilfe eines WWW-Browsers oder über Telnet zu. Der Zugriff auf Datenbanken über HTTP wird mit Java-Servlets ermöglicht²¹¹. Da bei dem DIMDI fast alle Datenbanken zentral vorliegen, muss es keine Absprachen über Schnittstellen mit anderen Datenanbietern treffen. Die verteilte Suche findet innerhalb des Systems des DIMDI statt. Die Daten aus den Datenquellen werden dem Host meist per FTP geliefert; sie liegen in unterschiedlichen Formaten vor, zunehmend in XML²¹².

Als Sonderform wird auf den BVB über eine Z39.50-Schnittstelle zugegriffen²¹³.

6.6 vascoda²¹⁴

Vascoda ist ein interdisziplinäres Internetportal, über das Virtuelle Fachbibliotheken Deutschlands und Informationsverbände, wie zum Beispiel MedPilot, zugänglich sind²¹⁵. Das bedeutet, dass vascoda ausschließlich auf verteilte Informationssysteme zugreift. Es wird als "Grundbaustein für die 'Digitale Bibliothek Deutschland'"²¹⁶ bezeichnet. Das Portal wurde im August 2003 mit dem Beginn der IFLA in Berlin eröffnet. Die Nutzung ist kostenfrei. Allerdings werden über die Oberfläche von vascoda kostenpflichtige Datenbestände nachgewiesen. Will ein Benutzer Zugang zu diesen haben, muss er sich bei dem jeweiligen System einloggen.

Architektur

Vascoda greift bei der Suche auf keinen eigenen Index zu, sondern es findet eine verteilte Suche statt²¹⁷. Eine nicht abgeschlossene Suche wird durch einen Timeout beendet. Die Trefferliste wird dem Benutzer erst angezeigt, wenn sie vollständig ist.

²¹¹ Vgl. Medizininformation: umfassend und unabhängig, S. 25

²¹² Diese Information stammt aus E-Mail 3

²¹³ Vgl. Memokarte des Bibliotheksverbands Bayern

²¹⁴ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.6 auf der Erkundung des Informationssystems durch die Autorin von Anfang September 2003. Zugang: „Homepage von vascoda“. Online: <http://www.vascoda.de> [Abrufdatum: 15.10.2003, Datei (screenshot): vascoda_1.bmp]

²¹⁵ Vgl. Vascoda: über uns. Online: zugänglich über <http://www.vascoda.de> [Abrufdatum: 15.10.2003, Datei (screenshot): vascoda_2.bmp]

²¹⁶ Vgl. ebd.

²¹⁷ Helmes, Leni: Vascoda: technische Umsetzung.

Online: <http://www.dl-forum.de/Foren/Vascoda/volltextdokumente/helmes.pdf> [Stand: 6.3.2003, Abrufdatum: 20.10.2003, Datei: vascoda_4.pdf]

Funktionen²¹⁸

Rechercheoberfläche: Vascoda bietet eine gemeinsame Oberfläche für die Suche und eine für die Kurzanzeige an. Für die Vollanzeige wird der Benutzer direkt auf die Originaloberfläche des entsprechenden Informationssystems geleitet.

Suche:

- Auswahl von Datenquellen: Der Benutzer kann keine einzelnen Datenquellen für die Suche auswählen, sondern es stehen ihm stattdessen vier verschiedene fachliche Cluster zur Verfügung. Die einzelnen Informationssysteme sind diesen Clustern zugeordnet.
- Sucheinstiege, Feldinhalte, Retrievalfunktionen: Für Sucheingaben gibt es nur ein einziges allgemeines Suchfeld. Im Hintergrund werden verschiedene Felder durchsucht. Für das zweite Release, das in nächster Zeit herauskommen soll²¹⁹, ist eine erweiterte Suche geplant, in der die feldspezifische Suche möglich sein soll²²⁰. Absprachen über Felder und deren Inhalte zwischen den einzelnen Informationssystemen gibt es zum Teil durch eine gemeinsame WSDL (s.u.). An Retrievalfunktionen steht ein größter gemeinsamer Nenner zur Verfügung (Boolesches UND und Trunkierung).
- Ausgabe der Kurztrefeferliste: In der Kurztrefeferliste sind die Ergebnisse nach Fundort sortiert unabhängig voneinander hintereinander aufgelistet. Es findet demnach keine Dublettenerkennung statt.

Weitere Dienste: Für die Lokalisierung ist bei elektronischen Zeitschriftenartikeln die EZB, die Elektronische Zeitschriftenbibliothek, mit eingebunden. Dies ist ein System, das standortabhängig über die Verfügbarkeit von Zeitschriften Auskunft gibt und Zugang dazu schafft²²¹.

Technische Realisierung²²²

Vascoda beruht auf der Technik von Web-Services. Vom Benutzer wird das Angebot mit Hilfe eines WWW-Browsers über das WWW genutzt. Auf die beteiligten Datenquellen wird HTTP-basiert zugegriffen. Sie müssen eine SOAP-fähige WWW-Schnittstelle besitzen, die auf eine für vascoda entwickelte WSDL zugreift²²³. Anfragen und Antworten werden in XML codiert. Auf die EZB wird über OpenURL-Syntax zugegriffen.

²¹⁸ Vgl. für diesen Abschnitt auch: „Suchhilfe vascoda“. Online: zugänglich über <http://www.vascoda.de/de/hilfe.html> [Abrufdatum: 15.10.2003, Datei (screenshot): vascoda_3.bmp]

²¹⁹ Diese Information stammt aus E-Mail 2 [Datum: 10.10.2003, Verfasser: Thomas Bausenwein, Karlsruhe, Mitarbeiter des FIZ-Karlsruhe, beteiligt an der technischen Realisierung von vascoda]

²²⁰ Diese Information stammt aus Telefonat 2 [Datum: 16.9.2003, Gesprächspartner: Thomas Bausenwein, Karlsruhe, Mitarbeiter des FIZ Karlsruhe, beteiligt an der technischen Realisierung von vascoda]

²²¹ Vgl. Informationen zur Elektronischen Zeitschriftenbibliothek. Hrsg. EZB. Online: zugänglich über <http://www.bibliothek.uni-regensburg.de/ezeit> [Stand: 22.7.2003, Abrufdatum: 15.10.2003, Datei: ezb.htm]

²²² Vgl. für diesen Abschnitt: Helmes, Leni: Vascoda: technische Umsetzung

²²³ Vgl. Steidl, Nicole: Einsatz von Web Services bei VASCODA. Nicht öffentlich zugängliche elektronische Ressource [Stand: 11.2.2003]

6.7 Renardus²²⁴

Renardus ist ein interdisziplinäres Internetportal, das verschiedene subject gateways Europas über eine einheitliche Oberfläche zugänglich macht. So bietet Renardus Zugang zu Dokumenten, die im WWW verfügbar sind. Der Aufbau von Renardus begann im Jahr 2000 im Rahmen eines groß angelegten Projektes, bei dem gründliche Recherchen über bereits existierende verteilte Informationssysteme und über Möglichkeiten in diesem Bereich gemacht und dokumentiert wurden²²⁵. Die Nutzung ist kostenfrei.

*Architektur*²²⁶

Renardus führt eine verteilte Suche durch. Allerdings greift der Broker nicht direkt auf die subject-gateways zu, sondern auf vorgeschaltete Datenbanken, in denen Metadaten abgelegt sind. Zu den Metadaten gehören nicht nur normierte Daten über Dokumente, sondern auch zusätzliche für die Verwaltung der Bestände wichtige Informationen²²⁷. Diese vorgeschalteten Datenbanken, auch Renardus-Server genannt, müssen von den einzelnen Providern bereitgestellt werden. Für diese Server sind zwei Modelle im Einsatz: entweder ein sogenannter "single server", der nur den Inhalt eines Serviceanbieters zur Verfügung stellt, oder ein "joint server", der die Inhalte mehrerer subject gateways enthält. Eine nicht abgeschlossene Suche wird durch einen Timeout beendet. Die Trefferliste wird erst angezeigt, wenn die Suche beendet ist.

*Funktionen*²²⁸

Rechercheoberfläche: Renardus bietet eine gemeinsame Oberfläche für die Suche und für die Kurzanzeige an. Bei der Kurzanzeige kann der Benutzer zwischen einer ausführlicheren und einer kürzeren Auflistung der Ergebnisse wählen. Für die Vollanzeige wird er direkt auf die Originaloberfläche der entsprechenden Datenquelle geleitet.

Suche:

- Auswahl von Datenquellen: Nach der Voreinstellung werden automatisch alle subject gateways durchsucht, allerdings hat der Benutzer in der er-

²²⁴ Soweit nicht anders gekennzeichnet, beruhen die Angaben in Kapitel 6.7 auf der Erkundung des Informationssysteme durch die Autorin von Anfang September 2003. Zugang: „Homepage von Renardus“. Online: <http://www.renardus.org/> [Abrufdatum: 15.10.2003, Datei (screenshot): renardus_1.bmp]

²²⁵ Vgl. Neuroth, Heike und Petra Lepschy: Das EU-Projekt Renardus. In: Bibliotheksdienst 35 (2001) 1, S. 49ff.

²²⁶ Vgl. für diesen Abschnitt: Renardus: project deliverable D1.5: Carpenter, Leona: Architectural Model for the Renardus system. Online: http://www.renardus.org/about_us/deliverables/d1_5/d1-5_final.doc [Stand: 25.06.2002, Abrufdatum: 13.10.2003, Datei: renardus_d1_5.doc]

²²⁷ Vgl. Renardus: project deliverable D2.2: Technical requirements for participating organizations. Anders Ardö, Colm Doyle, Johan Hallenberg [u.a.] Online: http://www.renardus.org/about_us/delivables/d2_2/D2_2_final.html [Stand: 8.7.2001, Abrufdatum: 13.10.2003, Datei: renardus_d2_2.htm]

²²⁸ Vgl. für diesen Abschnitt auch: „Suchhilfe Renardus“. Online: zugänglich über <http://www.renardus.org> [Abrufdatum: 15.10.2003, Datei: renardus_2.htm]

weiterten Suche die Möglichkeit, spezielle Datenquellen auszuwählen. Für die Zukunft ist ein query routing angedacht²²⁹.

- Sucheinstiege, Retrievalfunktionen, Feldinhalte: Es gibt eine einfache und eine erweiterte Suche. Da die Daten in den Renardus-Servern normiert vorliegen, stehen einige Datenfelder und Retrievalfunktionen (Boolesches UND, ODER, Trunkierung) zur Verfügung, die jedoch nur komplett in der erweiterten Suche genutzt werden können. Bei der einfachen Suche steht nur ein einziges allgemeines Suchfeld zur Verfügung. Felder und deren Inhalte sind in dem Renardus Datenmodell festgelegt, an das sich alle beteiligten Provider halten müssen²³⁰. Als weiterer Sucheinstieg ist das Browsing möglich. Angeordnet sind die Dokumente entsprechend der DDC (Dewey Decimal Classification). In dem Renardus-Server müssen nachgewiesene Dokumente nach der DDC erschlossen vorliegen. Dafür müssen einzelne Provider, wenn sie ursprünglich nicht die DDC verwenden, eine Transformation von ihrem Klassifikationssystem durchführen²³¹.
- Ausgabe der Kurztrefeferliste: In der Kurztrefeferliste sind die Ergebnisse alphabetisch hintereinander aufgelistet. Dubletten werden nicht erkannt. Die Trefferliste bietet mit dem Link "refine search" die Möglichkeiten, die Suchanfrage zu modifizieren.

Technische Realisierung²³²

Der Benutzer kommuniziert über einen Web-Browser mit dem Broker. Zwischen dem Broker und den einzelnen Renardus-Servern wird für die Suche das Protokoll Z39.50 verwendet. Demnach müssen alle Renardus-Server eine Z39.50-Schnittstelle besitzen. Administrative Daten, wie zum Beispiel Angaben über die Sammlung einzelner subject-gateways, werden über das Protokoll OAI-PMH an den Broker übergeben. Dieses kann auch bei der Pflege eines "joint servers" eingesetzt werden, indem die Metadaten einzelner Gateways regelmäßig auf den Renardus-Server überspielt werden.

²²⁹ Renardus: project deliverable D4.1: Documentation of the procedures and control mechanisms. Leona Carpenter; Johan Hallenberg; Sara Kjellberg [u.a.]. Nicht öffentlich zugängliche elektronische Ressource [Stand: 30.6.2002]

²³⁰ Vgl. Renardus: project deliverable D2.2: Technical requirements for participating organizations.

²³¹ Vgl. ebd.

²³² Sofern nicht anders angegeben vgl. für diesen Abschnitt: Renardus: project deliverable D1.5: Carpenter, Leona: Architectural Model for the Renardus system

7 SCHLUSS

Verteilte bibliografische Informationssysteme werden bei Recherchen von einer großen Anzahl von Personen ganz selbstverständlich genutzt. Dabei ist wohl den wenigsten klar, welche Unterschiede zwischen einzelnen Datenquellen im Hintergrund zu überwinden sind. Dies ist verständlich, da es ja gerade das Ziel verteilter Systeme ist, die Heterogenität einzelner Datenquellen zu verbergen und dem Endnutzer sie als Einheit erscheinen zu lassen.

Allerdings scheint diese Unkenntnis sich nicht nur auf Bibliotheksbenutzer und andere fachfremde Personen zu beschränken. Selbst in der Welt der Informationsspezialisten herrscht zum Teil Unkenntnis darüber, wie verschiedene Systeme funktionieren. So fand in der Diskussionsliste INETBIB im September 2003 ein Austausch über den KVK und die DigiBib statt, in dem behauptet wurde, dass beide Systeme mit Z39.50 arbeiten würden²³³. Zu dieser falschen Aussage meldete sich kurze Zeit später ein Bibliothekar aus New Haven in Connecticut zu Wort und stellte klar, dass der KVK nicht mit Z39.50-Schnittstellen arbeite²³⁴. Daraufhin meldete auch das HBZ, „dass die DigiBib keineswegs auf Z39.50 beschränkt“ sei²³⁵, sondern auch andere Protokolle eingesetzt würden.

Dieser Austausch ist nicht verwunderlich, da wenig über die Technik verteilter bibliografischer Informationssysteme verbreitet wird. Technisch detaillierte Berichte sind oft nicht frei zugänglich. Durch diese Arbeit ist ein Überblick über einige verschiedene technische Realisierungsmöglichkeiten gegeben worden. Der Schwerpunkt lag jedoch nicht allein auf der technischen Verknüpfung von Datenquellen, sondern auch auf inhaltlichen Aspekten, die in verteilten Informationssystemen zu berücksichtigen sind. Auch wenn in der Praxis oft eine Trennung zwischen technischen Teams und Teams, die sich um die inhaltlichen Belange kümmern, gemacht wird, ist dies meines Erachtens nicht sinnvoll. Bei verteilten Informationssystemen ist es kaum möglich, die technische und inhaltliche Ebene zu trennen. Erst durch die Vermischung beider Bereiche kann ein gutes System entstehen.

Für einen Informationsspezialisten, der Endnutzer verteilter Angebote ist, ist es wichtig, sein "Handwerkszeug" zu kennen. Ein professioneller Umgang mit verteilten bibliografischen Informationssystemen ist erst möglich, wenn sowohl Kenntnisse über den inhaltlichen als auch über den technischen Aufbau zu Grunde liegen. Diese Arbeit soll einen Beitrag dazu leisten, beide Bereiche zu verknüpfen.

²³³ Vgl. Hoppe, Frank <inetbib@ub.uni-dortmund.de> „Re: KVK und Digibib“ 9.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22240.html> [Abrufdatum: 13.10.2003, Datei:inet_1.htm]

²³⁴ Vgl. Askey, Dale <inetbib@ub.uni-dortmund.de> „Re: KVK und Digibib“ 9.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22244.html> [Abrufdatum: 13.10.2003, Datei:inet_2.htm]

²³⁵ Vgl. Jansen, Heiko <inetbib@ub.uni-dortmund.de> „Re: KVK und Digibib“ 9.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22245.html> [Abrufdatum: 13.10.2003, Datei:inet_3.htm]

8 LITERATURVERZEICHNIS

- Allen J. und M. Mealling: RFC 2651: The Architecture of the Common Indexing Protocol (CIP). Online: <http://www.faqs.org/rfcs/rfc2651.html>[Stand: 10/1999, Abrufdatum: 8.10.2003, Datei: cip.htm]
- Allers, Heinrich <inetbib@ub.uni-dortmund.de> „Fremddaten für mehrbändige Werke (war: KVK und Digibib)“ 10.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22356.html>[Abrufdatum: 13.10.2003, Datei: inet_5.htm]
- Arc - An OAI Service Provider for Digital Library Federation. Xiaoming Liu; Kurt Maly; Mohammad Zubair [u.a.]. In: D-Lib Magazine 7 (2001) 4. Online: <http://www.dlib.org/dlib/april01/liu/04liu.html> [Stand: 4/2001, Abrufdatum: 8.10.2003, Datei: arc.htm] Architektur von Web-Informationssystemen.
- Gerti Kappel; Werner Retschitzegger; Birgit Pröll [u.a.]. In: Web & Datenbanken: Konzepte, Architekturen, Anwendungen. Hrsg.: Erhard Rahm und Gottfried Vossen. Heidelberg: dpunkt-Verl. 2003, S. 101-134
- Askey, Dale <inetbib@ub.uni-dortmund.de> „Re: KVK und Digibib“ 9.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22244.html> [Abrufdatum: 13.10.2003, Datei: inet_2.htm]
- The Bath Profile: an international Z39.50 specification for library applications and resource discovery: release 1.1. Hrsg.: Carrol Lunau; Paul Miller und William E. Moen. Online: <http://www.nlc-bnc.ca/bath/bp-current.htm> [Stand: 02/2001, Abrufdatum: 13.10.2003, Datei: bath.htm]
- Begriffe im Kontext des Information Retrieval: Web-Lehrsammlung HU-Berlin. Online: <http://www.ib.hu-berlin.de/~is/web-lehrsammlung/Begriffe/Retrieval.htm> [Stand: 02.05.1999, Abrufdatum: 04.09.2003, Datei: ir_begriffe.htm]
- Das Dateienkonzept in Retrievalsystemen: Web-Lehrsammlung HU-Berlin. Online: <http://www.ib.hu-berlin.de/~is/Grundlagen/Dateikonzept.htm> [Stand: 01.05.2002, Abrufdatum: 04.09.2003, Datei: ir_dateien.htm]
- Dempsey, Lorcan; Rosemary Russell und Robin Murray: A utopian place of criticism: brokering access to network information. In: Journal of Documentation 55 (1999) 1, S. 33-70
- Diedrichs, Reiner: iPort als Portal für Bibliotheken im GBV. Online: <http://www.gbv.de/du/pdf/mupo.pdf> [Stand: 27.05.2003, Abrufdatum: 07.10.2003, Datei: gbv.pdf]
- Dierolf, Uwe: Karlsruher Virtueller Katalog mit über 30 Millionen Büchern und Zeitschriften: bundesweit erstmalige Dienstleistung der UB Karlsruhe. Online: <http://www.uni-karlsruhe.de/~presse/Veroeffentlichungen/Unikath/Unikath96/ub.html> [Stand: 27.11.1996, Abrufdatum: 13.10.2003, Datei: kvk_1.htm]
- Dierolf, Uwe: KVK goes Salt Lake City. In: B.I.T. Online 5 (2002) 4, S. 333-336
- Dierolf, Uwe und Michael Mönnich: Karlsruher Virtueller Katalog (KVK): Neue Dienstleistung im World Wide Web. In: Bibliotheksdienst 30 (1996) 8/9, S. 1395-1401
- Dierolf, Uwe und Michael W. Mönnich: KVK - Karlsruher Virtueller Katalog: zwei Jahre virtuell. In: B.I.T. Online (1998) 3, S. 159-168

-
- Dierolf, Uwe und Michael W. Mönnich: Mehrwertdienste durch Virtuelle Kataloge: 5 Jahre Karlsruher Virtueller Katalog. In: B.I.T. Online 4 (2001) 3, S. 261-266
- Die Digitale Bibliothek NRW: Technisches Konzept. Margarete Groos; Josef Hardt; Artur Nold [u.a.]. Online: <http://www.ub.uni-bielefeld.de/digibib-nrw/techkon.htm> [Stand: 28.4.1998, Abrufdatum: 13.10.2003, Datei: digibib_1.htm]
- DIMDI Datenbanken: Benutzerhandbuch. Köln: DIMDI 2002. Online: <http://www.dimdi.de/de/db/dbinfo/dbhandbuchg.pdf> [Abrufdatum: 15.10.2003, Datei: dimdi_5.pdf]
- DIMDI Kurzporträt. Hrsg.: DIMDI. Online: <http://www.dimdi.de/de/dimdi/dd/index.htm> [Abrufdatum: 15.10.2003, Datei: dimdi_4.htm]
- Dovey, Matthew J.: JISC Information Environment Service Level Descriptions. Online: <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/service-description/study/jiscie-sld.pdf> [Stand: 9.4.2003, Abrufdatum: 9.10.2003, Datei: protokolle.pdf]
- Dovey, Matthew J.: So you want to build a union catalogue? In: Ariadne 23 (2000) März. Online: <http://www.ariadne.ac.uk/issue23/dovey/intro.html> [Stand: 3/2000, Abrufdatum: 8.10.2003, Datei: katalog.htm]
- Eberhart, Andreas und Stefan Fischer: Java-Bausteine für E-Commerce-Anwendungen: verteilte Anwendungen mit Servlets, CORBA und XML. München: Hanser 2000
- Eimeren, Birgit von; Heinz Gerhard und Beate Frees: Internetverbreitung in Deutschland: unerwartet hoher Zuwachs. In Media Perspektiven (2003) 5, S. 338-358
- Elmasri, Ramez und Shamkant B. Navathe: Grundlagen von Datenbanksystemen, 3. Aufl. München: Pearson Studium 2002
- Entwicklungsprojekt „KOBV Informationsportal“: Teilprojekt der 1. Stufe: Monika Kuberek; Monika Lill; Stefan Litsche [u.a.]. Online: <http://www.zib.de/kuberek/publications/ZR-01-41.pdf> [Stand: 10/2001, Abrufdatum: 13.10.2003, Datei: kobv_4.pdf]
- Eversberg, Bernhard: Was sind und was sollen bibliothekarische Datenformate? Online: <http://www.zdb-opac.de/> [Stand: 25.1.1999, Abrufdatum: 18.10.2003, Datei: format.htm]
- Fasbender, Andreas und Otto Spaniol: [Artikel] WWW. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S.808
- Gale Directory of Databases: Volume 1: Online Databases: 2002 Part 1. Hrsg.: Erin Nagel. Detroit: Gale Group 2001
- Gardner, Tracy; Paul Miller und Rosemary Russell: MIA Functional Model, version 0.3. Online: <http://www.ukoln.ac.uk/dlis/models/requirements/func/> [Stand: 13.09.1999, Abrufdatum: 07.10.2003, Datei: mia_functional.htm]
- Gardner, Tracy; Paul Miller und Rosemary Russell: The MIA Logical Architecture, version 0.3. Online: <http://www.ukoln.ac.uk/dlis/models/requirements/arch/> [Stand: 13.09.1999, Abrufdatum: 08.10.2003, Datei: mia_logical.htm]
- Gaus, Wilhelm: Dokumentations- und Ordnungslehre: Theorie und Praxis des Information Retrieval. 3. Aufl. Berlin: Springer 2000
- Gusik, Antonius <inetbib@ub.uni-dortmund.de> „Re(2): KVK und Digibib“ 10.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22248.html> [Abrufdatum: 13.10.2003, Datei: inet_4.htm]

-
- Heinisch, Christian: Zur Usability von anbieterunabhängigen Metasuchoberflächen für die Fachinformationsrecherche im Invisible Web: Aufgabenstellung und Realisierung. In: ABI-Technik 21 (2001) 4, S. 312-317
- Helmes, Leni: Vascoda: technische Umsetzung. Online: <http://www.dl-forum.de/Foren/Vascoda/volltextdokumente/helmes.pdf> [Stand: 6.3.2003, Abrufdatum: 20.10.2003, Datei: vascoda_4.pdf]
- „Homepage der Digitalen Bibliothek NRW, Standort FHB Köln“. Online: <http://www.digibib-nrw.de/> [Abrufdatum: 15.10.2003, Datei (screenshot): digibib_2.bmp]
- „Homepage der Heinisch Informationsbewältigung e.Kfm“. Online: <http://www.informationsbewaeltigung.de/> [Stand: 2003, Abrufdatum: 8.10.2003, Datei: heinisch.htm]
- „Homepage der KOBV-Suchmaschine“. Online: <http://search.kobv.de/> [Abrufdatum: 16.10.2003, Datei: kobv_6.htm]
- „Homepage der Zeitschriftendatenbank (ZDB)“. Online: <http://www.zdb-opac.de/> [Abrufdatum: 18.10.2003, Datei: zdb.htm]
- „Homepage der Z39.50 Standard National Agency“. Online: <http://www.loc.gov/z3950/agency/> [Stand: 14.10.2003, Abrufdatum: 21.10.2003, Datei: maintenance.htm]
- „Homepage des DIMDI“. Online: <http://www.dimdi.de/> [Abrufdatum: 15.10.2003, Datei: dimdi_3.htm]
- „Homepage des KOBV“. Online: <http://www.kobv.de/> [Abrufdatum: 16.10.2003, Datei: kobv_5.htm]
- „Homepage des KVK“. Online: <http://kvk.uni-karlsruhe.de/> [Abrufdatum: 16.10.2003, Datei: kvk_2.htm]
- „Homepage für die Suche im Angebot des DIMDI“. Online: <http://www.dimdi.de/de/db/recherche.htm> [Abrufdatum: 15.10.2003, Datei: dimdi_6.htm]
- „Homepage von MedPilot“. Online: <http://www.medpilot.de/> [Abrufdatum: 15.10.2003, Datei (screenshot): medpilot.bmp]
- „Homepage von Renardus“. Online: <http://www.renardus.org/> [Abrufdatum: 15.10.2003, Datei (screenshot): renardus_1.bmp]
- „Homepage von vascoda“. Online: <http://www.vascoda.de/> [Abrufdatum: 15.10.2003, Datei (screenshot): vascoda_1.bmp]
- „Homepage von ZACK: deutsche Bibliotheken“. Online: <http://opus.tu-bs.de/zack/cgi/zack?ds-de> [Abrufdatum: 16.10.2003, Datei: zack_1.htm]
- Hoppe, Frank <inetbib@ub.uni-dortmund.de> „Re: KVK und Digibib“ 9.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22240.html> [Abrufdatum: 13.10.2003, Datei: inet_1.htm]
- Humphrey, Susanne M. und Biagio John Melloni: Databases: a primer for retrieving information by computer. New Jersey: Prentice-Hall 1986
- Hypertext Transfer Protocol - HTTP/1.1. Fielding, R.; J. Gettys; J. Mogul [u.a.]. Online: <http://www.ietf.org/rfc/rfc2616.txt> [Stand: 6/1999, Abrufdatum: 8.10.2003, Datei: http_1.txt]

-
- Iannella, Renato: Technical Review of RDNC Subject Gateway Services. Online: <http://www.rdn.ac.uk/publications/studies/technical-review/report.pdf> [Stand: 10.11.1999, Abrufdatum: 13.10.2003, Datei: rdnc.pdf]
- Informationen zur Elektronischen Zeitschriftenbibliothek. Hrsg.: EZB. Online: zugänglich über <http://www.bibliothek.uni-regensburg.de/ezeit> [Stand: 22.7.2003, Abrufdatum: 15.10.2003, Datei: ezb.htm]
- Internet: eine Einführung in die Nutzung der Internet-Dienste. Hrsg.: RRZN/ Universität Hannover. 8. Aufl. Hannover: RRZN 2000
- Jakobs, Kai und Otto Spaniol: [Artikel] Protokoll. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S.559
- Jansen, Heiko <inetbib@ub.uni-dortmund.de> „Re: KVK und Digibib“ 9.9.2003. Online: <http://www.ub.uni-dortmund.de/listen/inetbib/msg22245.html> [Abrufdatum: 13.10.2003, Datei: inet_3.htm]
- John Kirriemuir, Dan; Susan Welsh und Martin Jon Knight: Cross-searching subject gateways: the query routing and forward knowledge approach. In: D-Lib Magazine 4 (1998) 1. Online: <http://www.dlib.org/dlib/january98/01kirriemuir.html> [Stand: 1/1998, Abrufdatum: 8.10.2003, Datei: routing.htm]
- Jost, Michael: EULER-TAKEUP: Report on Sustainable Service and Business Model. Online: <http://www.emis.de/projects/EULER/takeup/d2.html> [Stand: 28.11.2002, Abrufdatum: 8.10.2003, Datei: euler_1.htm]
- Keane, Shirley: UKOLN: mission statement. Online: <http://www.ukoln.ac.uk/ukoln/mission/> [Stand: 24.4.2003, Abrufdatum: 17.10.2003, Datei: ukoln_1.htm]
- Keane, Shirley: UKOLN staff. Online: <http://www.ukoln.ac.uk/ukoln/staff/> [Stand: 9.10.2003, Abrufdatum: 18.10.2003, Datei: ukoln_2.htm]
- Kluck, Michael; J. Krause und M.N.O. Müller: Virtuelle Fachbibliothek Sozialwissenschaften: Projektskizze. Bonn: IZ Sozialwissenschaften 2000
- Der Kooperative Bibliotheksverbund Berlin Brandenburg. Martin Grötschel; Monika Kuberek; Stefan Lohrum [u.a.]. Online: <ftp://ftp.zib.de/pub/zib-publications/reports/SC-99-52.pdf> [Stand: 12/1999, Abrufdatum: 16.10.2003, Datei: kobv_1.pdf]
- Kowalski, Gerald: Information retrieval systems: theory and implementation. Boston, Kluwer Academic Publishers, 1997
- Kuberek, Monika: Dublettenbehandlung (Match- und Merge-Verfahren) in der KOBV-Suchmaschine: Grundlagen. Online: http://www.kobv.de/deutsch/content/wir_ueber_uns/kobv_projekt_1997-2000/docs/sc-99-16.pdf [Stand: 1999, Abrufdatum: 8.10.2003, Datei: kobv_3.pdf]
- Kuberek, Monika: Gemeinsamer KOBV-Standard. Online: <http://www.zib.de/kuberek/publications/SC-99-14.pdf> [Stand: 1.6.1999, Abrufdatum: 13.10.2003, Datei: KOBV_7.pdf].
- Kuberek, Monika; Stefan Lohrum und Raluca Rascanu: Endnutzerbasierte Online-Fernleihe im KOBV: Kurzbeschreibung. In: ZIB-Report 02-55 (2002). Online: <http://www.zib.de/kuberek/publications/ZR-02-55.pdf> [Stand: 12/2002, Abrufdatum: 16.10.2003, Datei: kobv_8.pdf]

-
- Lagoze, Carl und Herbert Van de Sompel: The Open Archives Initiative: building a low-barrier interoperability framework.
Online: <http://www.openarchives.org/documents/oai.pdf> [Abrufdatum: 8.10.2003, Datei: oai_1.pdf]
- Lynch, Clifford A: Metadata Harvesting and the Open Archives Initiative. In: ARL Bi-monthly Report 217 (2001). Online: <http://www.arl.org/newsltr/217/mhp.html> [Stand: August 2001, Abrufdatum: 8.10.2003, Datei: oai_2.htm]
- Lynch, Clifford A.: The Z39.50 Information Retrieval Standard: Part I: A strategic view of its past, present and future. In: D-Lib Magazine (1997) 4.
Online: <http://www.dlib.org/dlib/april97/04lynch.html> [Stand: 4/1997, Abrufdatum: 9.10.2003, Datei: z3950_3.htm]
- Medizininformation: umfassend und unabhängig: Deutsches Institut für Medizinische Dokumentation und Information: Jahresbericht 2002. Hrsg.: DIMDI.
Online: <http://www.dimdi.de/de/dimdi/public/jahr/jahr2002.pdf> [Stand: 2003, Abrufdatum: 13.10.2003, Datei: dimdi_1.pdf]
- Memokarte des Bibliotheksverbunds Bayern. Hrsg.: DIMDI.
Online: <http://www.dimdi.de/de/db/dbinfo/dbmemo/1a00ger.pdf> [Stand: 10/2001, Abrufdatum: 13.10.2003, Datei: dimdi_2.pdf]
- Miller, Paul: Z39.50 for all. In: Ariadne 21 (1999).
Online: <http://www.ariadne.ac.uk/issue21/z3950/intro.html> [Stand: September 1999, Abrufdatum: 9.10.2003, Datei: z3950_2.htm]
- Müller, Matthias N.O.: Integration unterschiedlich erschlossener Datenbestände am Beispiel der Virtuellen Fachbibliothek Sozialwissenschaften. In: Competence in content: 25. Online-Tagung der DGI: Frankfurt am Main, 3. bis 5. Juni 2003: Proceedings. Frankfurt am Main, Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis e.V. 2003
- Müller, Uwe und Klotz-Berendes, Bruno: Tutorium: die Open Archive Initiative. Online: http://eldorado.uni-dortmund.de:8080/bib/2003/workshop_oai.html;internal&action=buildframes.action [Stand: 8.5.2003, Abrufdatum: 9.10.2003, Datei: oaipmh_3.pdf]
- Ness, Andreas J: Eine Systemarchitektur für die Gestaltung und das Management verteilter Informationssysteme. Band 143 aus der Reihe IPA-IAO Forschung und Praxis. Berlin: Springer 1990
- Netzwerke: Grundlagen. Hrsg.: RRZN/ Universität Hannover. 3. Aufl. Nackenhei: Herdt 2000
- Neuroth, Heike und Petra Lepschy: Das EU-Projekt Renardus. In: Bibliotheksdienst 35 (2001) 1, S. 49-63
- Ollig, Hans: Einführung in DigiBib und Fernleihe. Online: http://www.hbz-nrw.de/produkte_dienstl/DigiBib/Vortraege/22012003/FH_DigiBib_220103.pdf [Abrufdatum: 18.10.2003, Datei: digibib_5.pdf]
- The Open Archives Initiative Protocol for Metadata Harvesting, version 1.1.
Online: <http://www.openarchives.org/OAI/1.1/openarchivesprotocol.htm> [Stand: 20.6.2001, Abrufdatum: 8.10.2003, Datei: oaipmh_1.htm]
- OpenURL Standard: a draft American national standard developed by the National Information Standards Organization. Online:
http://library.caltech.edu/openurl/Documents/OpenURL_Draft_Standard_Edited_03.doc [Stand: 2003, Abrufdatum: 14.10.2003, Datei: openurl_1.doc]

-
- Penke, K.; D. Scharna und J. Willenborg: Bibliotheksverzeichnisse zur Auswahl von Bibliotheksbeständen in verteilten Informationssystemen: Untersuchung und Einsatz im Kooperativen Bibliotheksverbund Berlin-Brandenburg.
Online: http://www.kobv.de/deutsch/content/wir_ueber_uns/docs/biblioverz.pdf
[Abrufdatum: 8.10.2003, Datei: kobv_2.pdf]
- Pieper, Dirk und Friedrich Summann: Die Entwicklung des Zugangssystems der Digitalen Bibliothek NRW. In: NfD 50 (1999), S. 397-405
- Powell, Andy: DNER portal architecture, 1999.
Online: <http://www.rdn.ac.uk/publications/mia/> [Stand: 20.6.2002, Abrufdatum: 14.10.2003, Datei: diagramm.htm]
- Powell, Andy und Liz Lyon: JISC Information Environment Architecture: Functional Model Online: <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/functional-model/>
[Stand: 09.04.2003, Abrufdatum: 08.10.2003, Datei: jisc_1.htm]
- Powell, Andy und Liz Lyon: The DNER Technical Architecture: scoping the information environment. Online: <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/dner-arch.html> [Stand: 18.05.2001, Abrufdatum: 07.10.2003, Datei: mia_dner.htm]
- Powell, Andy und Liz Lyon: The JISC Information Environment and Web services. In: Ariadne 31 (2002). Online: <http://www.ariadne.ac.uk/issue31/information-environments/>
[Stand: 04/2002, Abrufdatum: 15.10.2003, Datei: jisc_2.htm]
- Das Prinzip der Invertierung. Web-Lehrsammlung HU-Berlin. Online: <http://www.ib.hu-berlin.de/~is/Grundlagen/Invertierung.htm> [Stand: 01.05.2002, Abrufdatum: 04.09.2003, Datei: ir_invertierung.htm]
- Professional Java Serverprogramming. Danny Ayers; Hans Bergstein; Michael Bogovich [u.a.]. Birmingham: Wrox Press 1999
- Quernheim, Ulrich und Otto Spaniol: [Artikel] OSI-Referenzmodell. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S. 515-516
- Realizing the Hybrid Library. Stephen Pinfield; Jonathan Eaton; Catherine Edwards [u.a.]. In: D-Lib Magazine 4 (1998) 10.
Online: <http://www.dlib.org/dlib/october98/10pinfield.html> [Stand: Oktober 1998, Abrufdatum: 08.10.2003, Datei: agora.htm]
- Renardus: project deliverable D1.1: Evaluation report of existing broker models in related projects. Michael Day; Anders Ardö, Matthew J. Dovey [u.a.].
Online: http://www.renardus.org/about_us/deliverables/d1_1/D1_1_final.pdf
[Stand: 27.04.2000, Abrufdatum: 08.10.2003, Datei: renardus_d1_1.pdf]
- Renardus: project deliverable D1.4: Carpenter, Leona: Architectural Model for the Renardus System. Online: http://www.renardus.org/about_us/deliverables/d1_4/D1_4_final.pdf
[Stand: 11.12.2000, Abrufdatum: 8.10.2003, Datei: renardus_d1_4.pdf]
- Renardus: project deliverable D1.5: Carpenter, Leona: Architectural Model for the Renardus system. Online: http://www.renardus.org/about_us/deliverables/d1_5/d1-5_final.doc
[Stand: 25.06.2002, Abrufdatum: 13.10.2003, Datei: renardus_d1_5.doc]
- Renardus: project deliverable D2.1 Technical standards: Whois++ Data access protocol.
Online: http://www.renardus.org/about_us/deliverables/d2_1/WHOIS++.html
[Stand: 6/2000, Abrufdatum: 20.10.2003, Datei: renardus_d2_1b.htm]

-
- Renardus: project deliverable D2.1 Technical Standards: Z39.50 and profiles. Online: http://www.renardus.org/about_us/deliverables/d2_1/Z3950.html [Stand: 20.6.2000, Abrufdatum: 8.10.2003, Datei: renardus_d2_1a.htm]
- Renardus: project deliverable D2.2: Technical requirements for participating organizations. Anders Ardö, Colm Doyle, Johan Hallenberg [u.a.]
Online: http://www.renardus.org/about_us/delivables/d2_2/D2_2_final.html
[Stand: 8.7.2001, Abrufdatum: 13.10.2003, Datei: renardus_d2_2.htm]
- Renardus: project deliverable D4.1: Documentation of the procedures and control mechanisms. Leona Carpenter; Johan Hallenberg; Sara Kjellberg [u.a.]. Nicht öffentlich zugängliche elektronische Ressource [Stand: 30.6.2002]
- Rösch, Hermann: Portale in Internet, Betrieb und Wissenschaft: Marktplatz und Instrument des Kommunikations- und Wissensmanagements. In: BIT Online (2001) 3. Online: <http://www.b-i-t-online.de/archiv/2001-03/fach1.htm> [Stand: 2001, Abrufdatum: 17.10.2003, Datei: portal.htm]
- Salton, Gerard und Michael McGill: Information Retrieval: Grundlegendes für Informatikwissenschaftler. Hamburg: McGraw Hill 1987
- A scalable architecture for harvest-based Digital Libraries: The ODU/Southampton Experiments. Xiaoming Liu; Tim Brody; Stevan Harnad [u.a.]. In: D-Lib Magazine 8 (2002) 11. Online: <http://www.dlib.org/dlib/november02/liu/11liu.html> [Stand: 11/2002, Abrufdatum: 8.10.2003, Datei: harvesting_1.htm]
- Schmidt, Joachim W. und Gerald Schröder: [Artikel] Daten. In: VDI-Lexikon Informatik und Kommunikationstechnik. Hrsg.: Manfred Broy und Otto Spaniol. Berlin: Springer 1999, S.160
- Schneider, Wolfram: Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls: Diplomarbeit am Fachbereich Informatik der Technischen Universität Berlin. Berlin 1999
- SFX context sensitive linking: kontext-sensitives Linking mit einem unabhängigen Linkserver. Hrsg.: Ex Libris. ExLibris 2003
- Simple Object Access Protocol (SOAP) 1.1: W3C Note.
Online: <http://www.w3.org/TR/SOAP/> [Stand: 8.5.2000, Abrufdatum: 9.10.2003, Datei: soap_1.htm]
- Sompel, Herbert van de; Jeffrey A. Young und Thomas B. Hickey: Using the OAI-PMH...Differently. In: D-Lib Magazine 9 (2003) 7/8.
Online: <http://www.dlib.org/dlib/july03/young/07young.html> [Stand: 8/2003, Abrufdatum: 20.10.2003, Datei: oaipmh_2.htm]
- SRW-Search/Retrieve Web Service: Z39.50 International: next generation: background. Hrsg.: Z39.50 Maintenance Agency.
Online: <http://www.loc.gov/z3950/agency/zing/srw/> [Stand: 25.11.2002, Abrufdatum: 9.10.2003, Datei: srw_1.htm]
- SRW-Search/Retrieve Web Service: Z39.50 International: next generation: implementors. Hrsg.: Z39.50 Maintenance Agency.
Online: <http://www.loc.gov/z3950/agency/zing/srw/implementors.html> [Stand: 5.6.2003, Abrufdatum: 14.10.2003, Datei: srw_2.htm]

-
- SRW-Search/Retrieve Web Service: Z39.50 International: next generation: SRU: URL access mechanism. Hrsg.: Z39.50 Maintenance Agency.
Online: <http://www.loc.gov/z3950/agency/zing/srw/sru.html> [Stand: 25.11.2002, Abrufdatum: 9.10.2003, Datei: sru_1.htm]
- Steidl, Nicole: Einsatz von Web Services bei VASCODA. Nicht öffentlich zugängliche elektronische Ressource [Stand: 11.2.2003]. Der Text wurde nach Abschluss der vorliegenden Arbeit leicht bearbeitet veröffentlicht: Helmes, Leni und Nicole Steidl: Webservices praktisch angewandt: FIZ Karlsruhe entwickelt Automatisierung der Informationsinfrastruktur. In: Information, Wissenschaft und Praxis 54 (2003) 7, S. 421-428
- Stock, Wolfgang G: Checkliste für Retrievalsysteme: Qualitätskriterien von Suchmaschinen. In: Password (2000) 5, S. 22-31
- „Suchhilfe Digitale Bibliothek“. Online: zugänglich über <http://www.digibib-nrw.de> [Abrufdatum: 15.10.2003, Datei (nur Inhalt der Seite): digibib_3.doc]
- „Suchhilfe DIMDI“. Online: zugänglich über <http://www.dimdi.de/de/db/recherche.htm> [Abrufdatum: 15.10.2003, Datei: dimdi_7.htm]
- „Suchhilfe KOBV-Suchmaschine“. Online: zugänglich über <http://search.kobv.de> [Stand: 3.2.2003, Abrufdatum: 16.10.2003, Datei (nur Inhalt der Seite): kobv_9.doc]
- „Suchhilfe KVK“. Online: http://www.ubka.uni-karlsruhe.de/hylib/kvk_help.html [Abrufdatum: 16.10.2003, Datei: kvk_3.htm]
- „Suchhilfe Renardus“. Online: zugänglich über <http://www.renardus.org> [Abrufdatum: 15.10.2003, Datei: renardus_2.htm]
- „Suchhilfe vascoda“. Online: zugänglich über <http://www.vascoda.de> [Abrufdatum: 15.10.2003, Datei (screenshot): vascoda_3.bmp]
- „Suchhilfe ZACK“. Online: <http://opus.tu-bs.de/zack/help.html> [Abrufdatum: 16.10.2003, Datei: zack_2.htm]
- „Testzugang zu SFX“. Online: <http://www.sfxit.com/trysfx/index.html> [Stand: 6.2.2003, Abrufdatum: 17.10.2003, Datei: sfx_1.htm]
- Töteberg, Ingrid: Die Digitale Bibliothek - das universale Portal des Wissens. Online: http://www.hbz-nrw.de/produkte_dienstl/DigiBib/ [Stand: 6.12.2002, Abrufdatum: 16.10.2003, Datei: digibib_4.htm]
- Up and running: Implementing Z39.50: proceedings of a symposium sponsored by the State Library of Iowa November 26, 1996, Ames, Iowa. Hrsg.: Sara L. Randall. Bethesda: NISO Press 1997
- Vascoda: über uns. Online: zugänglich über <http://www.vascoda.de> [Abrufdatum: 15.10.2003, Datei (screenshot): vascoda_2.bmp]
- Veen, Theo van: Harvesting metadata and the use of application profiles: ELAG 2003: Cross language applications and the web, workshop 1.
Online: http://www.elag2003.ch/ws/ws_1.pdf [Stand: 4/2003, Abrufdatum: 9.10.2003, Datei: harvesting_2.pdf]
- Web Services. Markus Keidl; Alfons Kemper; Stefan Seltzsam [u.a.]. In: Web & Datenbanken: Konzepte, Architekturen, Anwendungen. Hrsg.: Erhard Rahm und Gottfried Vossen. Heidelberg: dpunkt-Verl 2003, S. 293-331

XML Schema: W3C Architecture Domain. Hrsg.: W3C.

Online: <http://www.w3.org/XML/Schema> [Stand: 4/2000, Abrufdatum: 7.10.2003, Datei: xml.htm]

Z39.50 –1995: Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. Hrsg.: Z39.50 Maintenance Agency.

Online: <ftp://ftp.loc.gov/pub/z3950/official/part1.pdf> [Stand: Juli 1995, Abrufdatum: 8.10.2003, Datei: z3950_1.pdf]

ZiNG SRU/SRW WSDL Specification, version 1.0.

Online: <http://www.loc.gov/z3950/agency/zing/srw/srw-bindings.wsdl> [Stand: 2.12.2002, Abrufdatum: 21.10.2003, Datei: srw_wsdl.xml]

Weitere Quellen:

Die im Folgenden aufgeführten E-Mails liegen bei der Autorin vor. Ebenso Gesprächsprotokolle der Telefonate.

E-Mail 1 [Datum: 8.9.2003, Verfasser: Peter Mayr, Köln, Mitarbeiter in der Technik der Digitalen Bibliothek im HBZ NRW]

E-Mail 2 [Datum: 10.10.2003, Verfasser: Thomas Bausenwein, Karlsruhe, Mitarbeiter des FIZ-Karlsruhe, beteiligt an der technischen Realisierung von vascoda]

E-Mail 3 [Datum: 30.9.2003, Verfasser: Susanne Breuer, Köln, Mitarbeiterin des DIMDI, Verantwortliche für die Öffentlichkeitsarbeit]

E-Mail 4 [Datum: 2.10.2003, Verfasser: Olaf Ritter, Köln, Mitarbeiter des DIMDI]

Telefonat 1 [Datum: 16.10.2001, Gesprächspartner: Stefan Lohrum, Berlin, Bereichsleiter Informationstechnik am Zuse Institut Berlin, Mitarbeit am KOBV]

Telefonat 2 [Datum: 16.9.2003, Gesprächspartner: Thomas Bausenwein, Karlsruhe, Mitarbeiter des FIZ Karlsruhe, beteiligt an der technischen Realisierung von vascoda]