Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Informatiker (FH)

an der

Fachhochschule Konstanz

-Hochschule für Technik, Wirtschaft und Gestaltung-

im Fachbereich Informatik / Wirtschaftsinformatik

Thema :	Evaluierung und prototypische Implementierung eines Suchdienstes für wissenschaftliche Bibliotheken auf Basis strukturierter Metadaten
Diplomand:	Tarik Varli Mannheimer Str. 18 78467 Konstanz
Firma:	Bibliothek Service Zentrum (BSZ) Fritz-Arnold-Str. 4 a 78467 Konstanz
Betreuer:	Professor Dr. Klein Fachhochschule Konstanz Dipl.-Bibl.(FH) Dipl.-Inf.Wiss. Andreas Lehmann BSZ – Konstanz
Eingereicht:	28.02.2000

Abstract

Die vorliegende Arbeit beschäftigt sich im zweiten Kapitel mit Metadaten und ihrem Einsatz im Bibliothekswesen. Nach einer Übersicht über die verschiedenen Metadatenentwürfe gehe ich besonders auf das Dublin Core Metadata Element Set (DCMES) ein. Diese Gruppe von 13 weiter spezifizierbaren Element ist im WWW und der internationalen bibliothekarischen Fachwelt als einheitliches Kurzformat für die Beschreibung einer elektronischen Ressource akzeptiert. Das BSZ entschied sich früh für die Produktion und Haltung solches Daten, so daß auch die zu implementierende Suchmaschine darauf aufbauen mußte. Einen besonderer Schwerpunkt wird also das gesamte Spektrum der Verwendung des DCMES beim BSZ bilden.

In Kapitel drei werde ich auf verschiedene funktionsbestimmte Gattungen von Suchdiensten eingehen: mit den Suchmöglichkeiten auf einzelnen Servern und Datenbanken über spezielle Gateways, eigene Seitenindexierung oder eine systematische Verzeichnisstruktur begann die Entwicklung. Ihnen folgten bald übergreifende roboterbasierte Dienste, oder solche, die einen Verzeichnisdienst mit den Daten verbinden, die über einen Crawler gesammelten werden. Einen weiteren Schritt der Entwicklung bilden die übergeordneten Suchdienste (Metasuchmaschinen), die eine Suchanfrage an mehrere "untergeordnete" Suchmaschinen weiterreichen, aber das Ergebnis in eine einheitliche Trefferanzeige übersetzen. Einen Ausblick auf die weitere Entwicklung bietet der Abschnitt über mobile Agenten, die unter dem Einsatz künstlicher Intelligenz "lernend" arbeiten sollen.

Mit einem Vergleich verschiedener, angebotener Suchdienste in Kapitel 4 wird die Entscheidung des BSZ für sein Angebot eines qualitätsorientierten Suchdienstes auf Harvestbasis untermauert, der der Aufgabenstellung dieser Einrichtung entspricht.

Das fünfte Kapitel gibt eine ausführliche Übersicht über das Harvest-System, das Gatherer, Broker und Webinterface verbindend am BSZ zum Einsatz kommt. Es dokumentiert im einzelnen die Schritte der Installation, Konfiguration und Anpassung an die technischen und fachlichen Gegebenheiten. Als Resultat wird der Dienst SWIB (**S**uchdienst der **wi**ssenschaftlichen **B**ibliotheken) angeboten.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Eine der Stärken von Bibliothekskatalogen ist das gezielte Retrieval, also die Suche mittels eines strukturierten und weitgehend kontrollierten Vokabulars unter Einbeziehung von Normdaten. Normdaten werden in zentralen Datenbanken gepflegt und verwaltet. Es gibt im deutschsprachigen Raum z.B. die Personennamendatei (PND), die Gemeinsame Körperschaftsdatei (GKD) und die Schlagwortnormdatei (SWD). Sie enthalten die normierte Ansetzung für eine Person, eine Körperschaft, ein Schlagwort und alle zugehörigen Verweise.

Die derzeitigen Suchdienste, in der Umgangssprache allg. als Suchmaschinen bezeichnet, und die ihnen zugrunde liegende Datenbasis, erlauben kein gezieltes Retrieval, wie dies bei Bibliothekskatalogen der Fall ist (Siehe Abb. 1 und Abb.2).

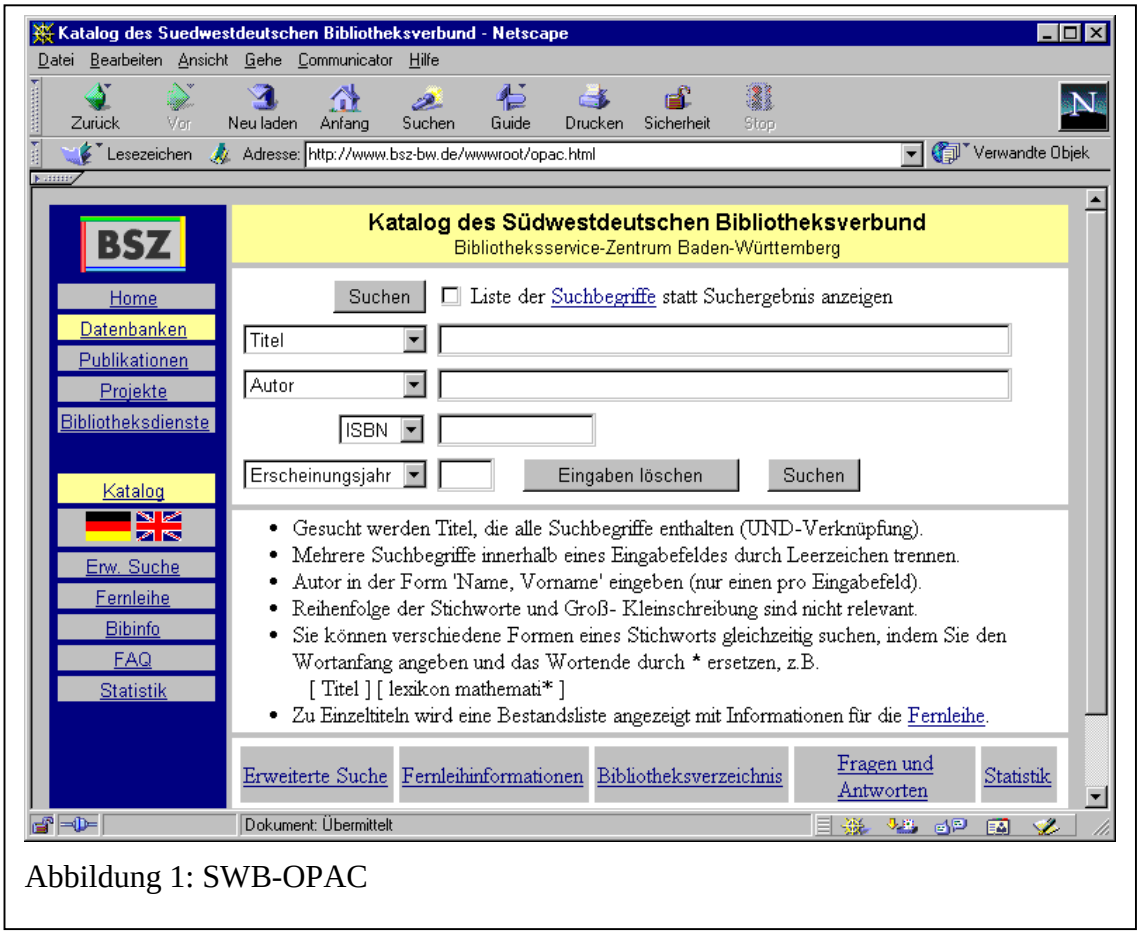

Abbildung 1: SWB-OPAC

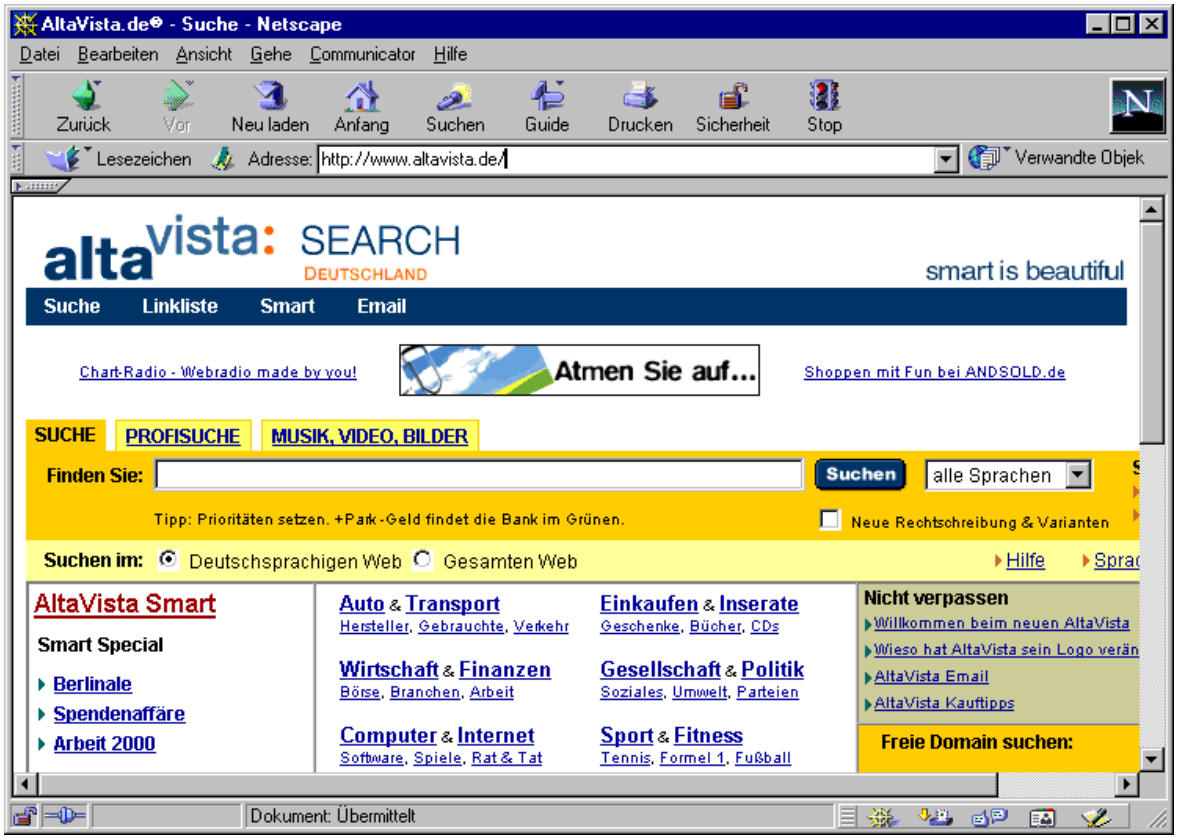

Abbildung 2: Startseite des AltaVista.de

In der Abbildungen 1 und 2wird sofort ersichtlich, daß WWW-OPAC[1] (http://www.bsz-bw.de/wwwroot/opac.html), für die Recherche im Katalog des Südwestdeutschen Bibliotheksverbund(SWB[2]), mit ihren Felder wie Titel, Autor ein geziehltere Recherche ermöglicht, was bei AltaVista nicht der Fall ist.

Im Internet gibt es keine Kontrolle oder zentrale Koordination. Dadurch ist jede Person, die einen Internetzugang besitzt, in der Lage seine eigene Information zu veröffentlichen. Hierdurch wird eine Fülle der Informationsangebote garantiert. Eine Information im Internet zu finden hängt oft davon ab, ob Informationsinhaber eine Interesse daran haben, sie zu veröffentlichen. Jede im Internet gestellte Information kann auch jederzeit wieder aus dem Internet entfernt werden. Es hängt vom Seitenanbieter ab, welche In-

[1] Der WWW-OPAC (Online Public Access Catalog) erlaubt als WWW-Schnittstelle den Zugriff auf die Bestände des SWB-Katalog.
[2] Der SWB wurde 1983 als kooperative Einrichtung der Universitäten des Landes Baden-Württemberg gegründet. Die Verbundzentrale wurde an der Universität Konstanz als zentrale Einrichtung unabhängig von der Universitätsbibliothek eingerichtet (Vgl. BSZ-Kompakt Seite 3).

formationen weiter im Internet präsentiert werden sollen. Hierdurch entsteht eine gewisse Beliebigkeit.

Durch die täglich hinzukommende Angebote im Internet steigt die Informationsvielfalt. Bestehende Angebote können verschwinden, verändert oder verschoben werden. Während bei einem Bibliothekskatalog die Verantwortlichen dafür sorgen, das veraltete Informationen gar nicht zugänglich sind, existiert im Internet keine solche Instanz, die dafür sorgt, daß die Informationen auf dem aktuellen Stand sind. Im Internet Texte, Grafiken, ganze Bücher, oder eine Datenbanken veröffentlicht werden.

Aus den genannten Gründen sind herkömmliche Suchdienste – gemessen an „klassischen“ bibliothekarischen Nachweisinstrumenten - für wissenschaftliche Zwecke nur bedingt einsetzbar.

Das Retrieval in Suchdiensten des Internets erweist sich aus zahlreichen weiteren Gründen als problematisch:

- Strukturierte Metadaten (wie bspw. das Dublin Core Metadata Element Set (DCMES) (Siehe Kap.2.4)) werden von den meisten Suchmaschinen nicht unterstützt.

- Viele Suchergebnisse sind irrelevant, der Nutzer droht in der Flut der Antworten zu „ertrinken“, d.h. Recherchen liefern einen hohen *Recall* bei sehr geringer *Precision*. Bspw. ergab eine Suche nach „bill gates“ bei AltaVista über 14.000 Treffer, von denen die ersten 50 entweder kaum einen Bezug zur gesuchten Person hatten oder deren URL nicht mehr gültig war.

- Viele Datenbankeinträge sind veraltet, da offenbar zu selten ein vollständiges Update erfolgt.

- Die Suchmaschinen unterstützen den Nutzer nur wenig oder gar nicht bei der Formulierung von Suchanfragen. Als Standard werden bei vielen Suchmaschinen alle eingegebenen Begriffe mit ODER verknüpft und es besteht keine Möglichkeit, an-

zugeben, in welchem Kontext der Begriff gesucht wird (also z.B. Suche nach dem Titel oder Autor einer Internet-Ressource).

Aus diesem Grund soll im Rahmen dieser Diplomarbeit eine prototypische Suchmaschine implementiert und evaluiert werden, die die genannten Probleme löst oder zumindest Lösungsansätze aufzeigt.

Der Suchdienst solte folgende Anforderungen erfüllen:

- er soll den Einsatz strukturierter Metadaten, speziell des DCMES unterstützen,
- er soll - durch den Einsatz dieser Metadaten, sowie durch eine ausgewählte (d.h. eingeschränkte) Datenbasis - dem Benutzer ein gezielteres Retrieval ermöglichen,
- er soll - durch regelmäßige Aktualisierung in kurzen Intervallen sowie durch die Beschränkung auf WWW-Angebote, deren langfristige Verfügbarkeit garantiert ist, dem Benutzer die Gewähr bieten, daß die gefundenen Objekte tatsächlich verfügbar sind und
- der Benutzer soll bei der Formulierung von Suchanfragen unterstützt werden, ähnlich wie bspw. in online verfügbaren Bibliothekskatalogen (OPACs).

Auf dem Markt ist eine Vielzahl von kommerziellen und nicht-kommerziellen Systemen verfügbar, mit denen ein Internet-Suchdienst aufgebaut werden kann. Für den beim Bibliotheksservice-Zentrum Baden-Württemberg (BSZ) aufzubauenden **S**uchdienst der **wi**ssenschaftlichen **B**ibliotheken (SWIB)[3] entschied man sich für den Einsatz des Systems *Harvest.*

[3] Vgl. http://www.bsz-bw.de/diglib/medserv/projekt/swib/swib.html

2 Metadaten und deren Einsatz im Bibliothekswesen

Unter Metadaten (Beschreibung von Daten durch Daten) versteht man strukturierte Daten, mit deren Hilfe eine Informationsressource beschrieben wird. Sie beschreiben die Dokumente in inhaltlicher und formaler Hinsicht; sie enthalten Informationen, wie z.B. Angaben über Autor, Titel, Abstract oder Zeitpunkt der Veröffentlichung, die die Verwaltung von Dokumenten oder Objekten gewährleisten, und den Zugriff auf sie optimieren oder überhaupt ermöglichen. Sie stellen damit das, was an Erschließungsarbeit in den Bibliotheken seit jeher geleistet wurde, dar. Wie jedes bibliothekarische Regelwerk setzt auch der effektive Einsatz von Metadaten obligatorisch eine gewisse strukturierte Standardisierung voraus[4].

Ein mit strukturierten Metadaten beschriebenes Dokument kann im Internet besser identifiziert und dadurch auffindbar gemacht werden. Deshalb verwendet man die Metadaten für die digitalen Dokumente und Objekte, die in den unterschiedlichsten, z.T. nicht miteinander kompatiblen Formen und Formaten vorliegen (z.B. als Textdateien, Bilddateien, Sounddateien, Videodateien). Angesichts des explosionsartigen Anwachsens der Datenmengen im Internet ist der Einsatz von Metadaten notwendiger denn je geworden.

Metadaten werden sowohl für digitale Dokumente als auch für physikalische Objekte (Bücher, Gegenstände, etc.) benutzt, die selbst nicht digital gespeichert, aber digital beschrieben werden (z.B. Titelaufnahmen der Bestände in Bibliotheken). Somit können inhaltlich verbundene, unterschiedliche Objekte (digital, konventionell) zu einem Thema oder Fachgebiet durch eine Recherche zusammengeführt werden.

Der Begriff „Metadaten“ geht zwar ebenso wie deren Verwendung dem Web-Zeitalter voraus (man denke bspw. an Datenbanken, Bibliothekskataloge usw.), wird aber in letzter Zeit zunehmend auch im Zusammenhang mit dem Internet –genauer gesagt mit dem WWW- verwendet.

[4] Vgl. http://www2.sub.uni-goettingen.de/metall.html

2.1 Überblick über verschiedenartige Metadaten[5]

Es existieren eine Vielzahl von Metadatenformaten die beschreiben, wie Metainformationen anzugeben und abzubilden sind, z.B.:

Text Encoding Initiative (TEI)

Die Text Encoding Initiative (TEI) wurde 1994 auf SGML-Strukturen zur Identifikation bibliographischer Informationen beim Electronic Text Center in Virginia/USA entwikkelt. Dieser Metadaten-Satz wurde hauptsächlich zur Anwendung von digitalisierten Texten von gedruckten Vorlagen konzipiert. Deshalb ist er stark auf Textdokumente beschränkt.
(Siehe http://www.tei-c.org/).

Encoding Archive Description (EAD)

Dieser Metadaten-Satz wurde auf SGML-Basis aufgebaut und dient auch als Suchkriterium für Archivdaten.
(Siehe http://lcweb.loc.gov/rr/ead/eadhome.html).

Government Information Locator Service (GILS)

Den US-Regierungsbehörden wurde vorgeschrieben, alle Dokumente mit Metadaten zu versehen. Da seit einigen Jahren alle amtlichen Druckschriften der Regierung vom Government Printing Office auch in elektronischer Form angeboten werden, fungiert dieser Metadaten-Satz als eine Art Norm für alle US-amtlichen Druckschriften.
(Siehe http://www.access.gpo.gov/su_docs/gils/whatgils.html).

US-MACHINE-READABLE CATALOGUE (USMARC)

USMARC ist das Austauschformat für die elektronische Verarbeitung bibliographischer Daten, das vom Network Office der Library of Congress in den sechziger Jahren ent-

[5] Vgl. http://www.mpib-berlin.mpg.de/dok/metadata/metaifbs.htm

wickelt wurde. Dieses Metadaten-Format enthält weitaus mehr und differenziertere Felder als jeglicher andere Metadaten-Satz
(Siehe http://www.tlcdelivers.com/tlc/crs/gen0001.htm).

Das Maschinelle Austauschformat für Bibliotheken 2 (MAB2)

Das deutsche Pendant zu USMARC, MAB2, dient auch als Metadaten-Satz, der im Umfang ähnlich ist wie USMARC, jedoch mit einfachen Feldern, während USMARC-Felder oft in Unterteilungen (Subfelder) strukturiert ist
(Siehe http://www.de.freebsd.org/~wosch/lv/diplom/html/node142.html).

Das Dublin Core Metadata Element Set (DCMES)

Der Dublin Core stellt den „Kern“ der inhaltlichen und formalen Erschließungsmerkmale, die sonst für die bibliothekarische und inhaltliche Erschließung benutzt worden sind. Von diesen Metadatenformaten ist, insbesondere im bibliothekarischen Bereich, das DCMES ein weitverbreiteter Ansatz geworden. Auch im BSZ wird das DCMES bereits seit mehreren Jahren eingesetzt und ist eine der Grundlagen für den zu entwikkelnden Suchdienst.

2.2 Das Dublin Core Metadata Element Set

Wie bereits in der Einleitung erwähnt sind Internet-Suchmaschinen in vielerlei Hinsicht problematisch. Um die Suchverfahren zu präzisieren, d.h. im Internet relevante Ressourcen gezielt zu finden, wurde eine Initiative gegründet, die eine Reaktion auf die Herausforderungen durch das Internet darstellt und einen zukunftsweisenden Ansatzpunkt für eine verbesserte Informationssuche im Internet bietet.

Die internationale Dublin-Core-Initiative unter Leitung des Online Computer Library Center (OCLC) in Dublin, Ohio besteht aus Vertretern der verschiedensten „Communities“, darunter Vertreter aus Archiven, Museen, Dokumentationsstellen, Bibliotheken, Verlagen und aus zahlreichen wissenschaftlichen Disziplinen. Diese Initiative hat zum

Ziel, ein einfach zu verwendendes, interdisziplinär einsetzbares Konzept für strukturierte Metadaten zu entwerfen, mit dem Internet-Ressourcen strukturiert beschreiben werden können, um diese gezielt suchbar und selektierbar zu machen.

Das von der Initiative vorgeschlagene Konzept, das Dublin Core Metadata Element Set (DCMES) wurde 1995 als eine Liste von 13 Datenelementen zur Beschreibung von digitalen Dokumenten bzw. Ressourcen vorgestellt. Dieses Konzept, Metadaten zur Erschließung von digitalen Ressourcen einzusetzen, ist in den letzten Jahren gereift und hat sich international unter der Schirmherrschaft der Dublin Core Initiative durchgesetzt[6]. Seit 1995 wurde das DCMES in verschiedenen Arbeitsgruppen (Siehe http://purl.oclc.org/dc/groups/index.htm) weiterentwickelt und besteht derzeit aus 15 Elementen zur Ressourcenbeschreibung. 1998 wurde das Konzept als RFC2413 veröffentlicht.
(Siehe http://www.ietf.org/rfc/rfc2413.txt).

Wichtig für die erste Entwicklung des DCMES war der fachübergreifende Konsens, daß die Erschließungs- und Retrievalaspekte für digitale Objekte mit der Beschreibung durch Dublin Core Metadaten zu präzisieren seien.

2.2.1 Zielsetzung

Das Ziel ist es, ein kostengünstiges, leicht handhabbares und effektives Verfahren zu entwickeln, das auch vor der großen Zahl an Dokumenten in digitalen Netzen nicht kapitulieren muß. Darüber hinaus soll die Anbindung an andere, zumeist komplexere Formate möglich werden (interoperability).

Die Grundidee besteht darin, bereits bei der Erstellung der Ressourcen die strukturierten Metadaten einzubinden.

DCMES ist bewußt einfach gehalten, damit Entwickler von Dokumenten entsprechende Metadaten gegebenenfalls selbst generieren und aus ihrer Sicht interpretieren können,

[6] Vgl. http://www.dbi-berlin.de/dbi_pub/bd_art/97_04_08.htm

ohne dabei auf aufwendige und teure Verfahren durch geschultes Personal zurückgreifen zu müssen.

Im Mittelpunkt steht das Bestreben, einen Minimalsatz von Erschließungselementen zu definieren, die zur verbesserten Präzision und Retrievalfähigkeit digitaler Dokumente bei Recherchen im Internet verhelfen können[7]. HTML-Dokumente sollen mit eingebetteten formalen und inhaltlichen „Metatags“ im Header des Dokuments versehen werden, die jedoch nicht beim normalen Display durch den Browser angezeigt werden. Sie sind nur bei der Anzeige des Quelltextes für das menschliche Auge zu sehen, können aber von den sog. Suchmaschinen, Robotern und Gatherern interpretiert und weiterverarbeitet werden.

Um auch die multimedialen Möglichkeiten des Internets nutzen zu können, wurde das Konzept so erweitert, daß bspw. auch Bild-, Video- und Audiodateien mit eingeschlossen wurden.

Durch das DC-Metadatenmodell können inhaltliche und formale Anfragen bessere Suchergebnisse liefern.

2.2.2 Die Elemente des DCMES

Die derzeit aktuelle Version des DCMES, Version 1.1, enthält die folgenden 15 Elemente[8]:

1. **Title** (DC.TITLE): A name given to the resource.

2. **Creator** (DC.CREATOR)**:** An entity primarily responsible for making the content of the resource.

3. **Subject** (DC.SUBJECT)**:** The topic of the content of the resource.

[7] Vgl. http://www.mpib-berlin.mpg.de/dok/metadata/metaifbs.htm
[8] Vgl. http://purl.org/DC/documents/rec-dces-19990702.htm

4. **Description** (DC.DESCRIPTION)**:** An account of the content of the resource.

5. **Publisher** (DC.PUBLISHER)**:** An entity responsible for making the resource available

6. **Contributor** (DC.CONTRIBUTORS)**:** An entity responsible for making contributions to the content of the resource.

7. **Date** (DC.DATE)**:** A date associated with an event in the life cycle of the resource.

8. **Type** (DC.TYPE)**:** The nature or genre of the content of the resource.

9. **Format** (DC.FORMAT)**:** The physical or digital manifestation of the resource.

10. **Identifier** (DC.IDENTIFIER)**:** An unambiguous reference to the resource within a given context.

11. **Source** (DC.SOURCE)**:** A Reference to a resource from which the present resource is derived.

12. **Language** (DC.LANGUAGE)**:** A language of the intellectual content of the resource.

13. **Relation** (DC.RELATION)**:** A reference to a related resource.

14. **Coverage** (DC.COVERAGE)**:** The extent or scope of the content of the resource.

15. **Rights** (DC.RIGHTS)**:** Information about rights held in and over the resource.

Für die detaillierte Beschreibung (Siehe Anhang A)

2.2.3 DCMES-Elemente in HTML-Seiten

Metadaten in HTML

Im HTML-Standard 3.2 wurden konkrete Angaben zur Einbindung von Metadaten gemacht. In aktuellen HTML-Standard 4.0 existieren diese nicht mehr. Es wird lediglich syntaktisch beschrieben, wie die HTML-Dokumente mit Metadaten versehen werden. An einer Standardisierung für die Einbindung von Metadaten arbeitet das W3-Konsortium, als Beispiel sei das Ressource Description Framework (RDF)[9] genannt.

Die Metadaten wie *description, author, keywords, date,* die in der HTML-Spezifikation u.a. erwähnt werden, sind am weitesten verbreitet und werden im Allgemeinen von Suchmaschinen erkannt bzw. interpretiert.

Diese Metadaten werden wie im folgenden Beispiel im Header der HTML-Datei eingetragen:

```
<head>
<meta name="description" content="Eine Inhaltsbeschreibung des
                                  Dokuments">
<meta name="author" content="Autoren des Dokument">
<meta name="keywords" content="wichtige Begriffe, die in dem Dokument
                               vorkommen">
<meta name="date" content="26.01.00">
... weitere Angaben...
</head>
```

DCMES-Metadaten

DCMES-Elemente werden in den HTML-Seiten in gleicher Form in den Header-Teil eingefügt. Allerdings wird Metadaten, die einem bestimmten öffentlichen System angehören (so wie hier dem System von Dublin Core) eine Kurzbezeichnung für den Herausgeber der Metadaten vorangestellt. Bei Dublin Core sind dies die Initialen DC. Dahinter folgt, durch einen Punkt getrennt, der Name der Metadaten-Angabe.

Ein Dublin-Core-Eintrag in allgemeiner Darstellung hat innerhalb einer HTML-Seite folgende Syntax:

```
<META NAME="DC.element name" CONTENT="value of element">
z.B.
<META NAME="DC.TITLE" CONTENT="Kreieren eines DC-Metadaten-Eintrags">
```

2.2.4 Dublin Core Qualifiers: Scheme und Type

Das aktuelle DCMES besteht aus 15 Basis-Elementen. Diese 15 Elemente dürften die meisten der Metadaten-Attribute abdecken, die benötigt werden, um eine qualifiziert und gezielt recherchierbare Beschreibung von elektronischen Dokumenten oder Objekten im Internet zu ermöglichen.

Bei nur 15 Elementen treten jedoch oftmals Mehrfachbelegungen mit unterschiedlichen Inhalten auf. Um diese differenzieren, interpretieren und voneinander unterscheiden zu können, ist die Nutzung von Qualifizierungselementen (Dublin Core Qualifiers) vorgesehen.

Die Grundidee ist es, die 15 Basis-Elemente mittels der Qualifizierungselemente SCHEME und TYPE näher zu differenzieren. Die Qualifizierungselemente bestehen aus einem Namen (identifier) und einem Wert (value) und sollen darüber informieren, wie ein DCMES- Element bzw. dessen Inhalt zu interpretieren ist, d.h. sie dienen einer Spezifizierung der Semantik.

Sowohl Namen als auch Wert der Qualifizierungselemente müssen einem kontrollierten Vokabular entstammen, um die maschinelle Interpretation von Daten unterschiedlichster Herkunft zu erleichtern.

[9] Vgl. http://www.w3.org/RDF/

- **Qualifizierungselement TYPE**

Das Qualifizierungselement TYPE wird zur Spezifizierung des Basiselements verwendet. Beispielsweise kann das Element CREATOR in einem Metadaten-Eintrag mehrfach vorkommen bzw. in unterschiedlicher Bedeutung vorkommen (beispielsweise für Personennamen, Körperschaften, die zugehörige E-Mail-Adresse, Telefon-Nr. usw.).

Syntax des DCMES-Elements, in dem das Qualifizierungselement TYPE verwendet wird:

```
<META NAME="DC.element name.TYPE identifier" CONTENT="value of element">
```

Beispiele

```
<META NAME="DC.CREATOR.NAME" CONTENT="Hinterhofer, Xaver">
<META NAME="DC.CREATOR.EMAIL" CONTENT="Hinterhover@uni-adorf.de">
```

- **Qualifizierungselement SCHEME**

Das Qualifizierungselement SCHEME dient zur Interpretation des Inhalts eines Basiselements. Mit SCHEME werden die bei einigen DCMES Basiselementen verwendeten Code-Systeme identifiziert.

Syntax des DC-Elements, in dem das Qualifizierungselement SCHEME verwendet wird:

```
<META NAME="DC.element name" SCHEME="identifier" CONTENT="value of element">
```

Beispiele:

```
<META NAME="DC.DATE.CURRENT" SCHEME="ANSI.X3.30-1985" CONTENT="11970721">
<META NAME="DC.DATE.CURRENT" SCHEME="ISO31" CONTENT="1197-07-21">
<META NAME="DC.FORMAT" SCHEME="IMT" CONTENT="text/html">
```

2.3 Der Virtuelle Medienserver und der Einsatz des DCMES beim BSZ

Bereits seit mehreren Jahren beteiligt sich das BSZ an der Erzeugung von Metadaten und ihrer Einführung in den deutschen Bibliotheken. Alle HTML-Dateien im virtuellen Medienserver werden nach dem Dublin-Core-Standard strukturiert beschrieben.

Der Virtuelle Medienserver des BSZ dient der Speicherung, Erschließung und Bereitstellung von Dokumenten. Hierbei handelt es sich um Objekte wie Texte, Images und Audio-Files, die ins Internet gestellt werden und allgemein zugänglich sind. Auch die Einbindung anderer Objekttypen ist möglich und für die Zukunft geplant.

Diese Objekte lassen sich inhaltlich und formal folgendermaßen unterscheiden: Hochschulpublikationen (z.B. Dissertationen, Diplomarbeiten, Forschungsberichte, Preprints), E-Reprints, E-Journals, Datenbanken, unselbständige Werke (z.B. Artikel aus der Zeitschrift Bibliotheksdienst), VD 17-Daten[10] (Text- und Image-Spiegelungen), Metainformationen (Rezensionen, Abstracts, Textproben, Inhaltsverzeichnisse etc.).

[10] Im Projekt Verzeichnis der im deutschen Sprachraum erschienenen Drucke des 17. Jahrhunderts, kurz VD17, wird eine Nationalbibliographie für den Zeitraum von 1601 bis 1700 erstellt (Vgl. http://www.VD17.de).

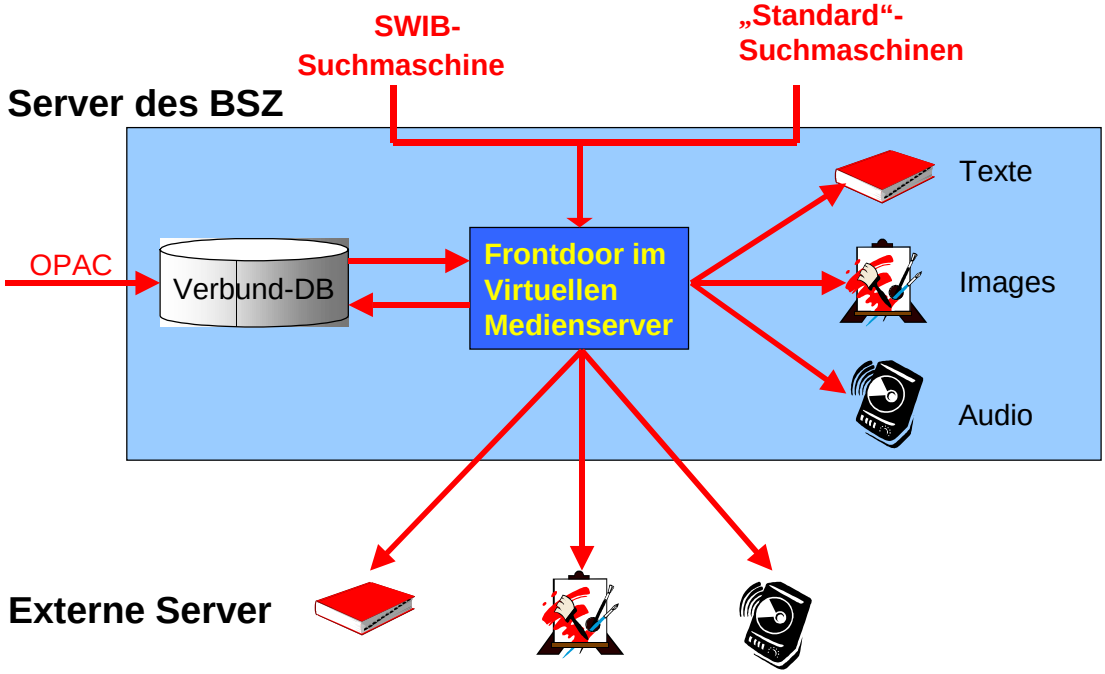

Abbildung 3: Der Virtuelle Medienserver des BSZ

Der Virtuelle Medienserver (Siehe Abb. 3) ist seit 1995 im Betrieb und stellt als geographisch verteiltes Depot (auch SWB-E-Depot oder einfach E-Depot genannt) die integrierende Plattform der Bibliotheksregion für elektronisch verfügbare Objekte und neue Dienstleistungen dar: Die Online-Ressourcen der Universitätsbibliotheken Chemnitz-Zwickau, Karlsruhe, Konstanz, Kaiserslautern, Mannheim, Stuttgart bzw. auf dem BSZ-eigenen Server für die gesamte Region vorgehaltene Text-, Bild-, und Audioobjekte werden im Medienserver mit vielfältigen Recherche-Einstiegen angeboten. Der Einbezug weiterer Bibliotheken (Tübingen, Heidelberg, Freiburg, Stuttgart-Hohenheim, Dresden, Leipzig, Saarbrücken) steht kurz bevor. Über diese vollständig verfügbaren Objekte hinaus werden zu mehr als 90.000 Dokumenten weiterführende, direkt verknüpfte Informationen wie Abstracts, Rezensionen oder Inhaltsverzeichnisse angeboten. *„Damit stellt der Virtuelle Medienserver in der Südwestdeutschen Bibliotheksregion eine der größten ohne Zugangsbeschränkung im Internet verfügbaren Dokumentsammlungen des deutschen Bibliothekswesen aller Fachdisziplinen dar“*[11].

[11] Vgl. BSZ-Kompakt, Auflage 1999

2.4 Frontdoors auf Basis von DC-Elementen

Über den Virtuellen Medienserver des BSZ ist es bereits seit mehreren Jahren möglich, aus Titelaufnahmen in der Verbunddatenbank[12] auf Objekte im WWW zuzugreifen. Diese Objekte können sowohl beim BSZ vorgehalten als auch auf Servern anderer Einrichtungen gespeichert werden. Der Zugriff auf die Objekte erfolgt dabei nicht direkt aus der Titelaufnahme, sondern über dazwischengeschaltete HTML-Dateien, sogenannte Frontdoors. Diese Frontdoors erfüllen mehrere Funktionen:

- Sie entkoppeln die Titelaufnahme vom eigentlichen Objekt,
- Sie bieten Vorabinformationen für den Benutzer (Abstracts, Inhaltsverzeichnisse u.ä.) und
- Sie sind mit DC-Metadaten versehen, wodurch sie und damit auch die Objekte im WWW (z.B. durch den SWIB-Suchdienst) recherchierbar werden.

Beim BSZ wurden Verfahren entwickelt, um aus Dublin Core Metadaten sowohl Titelaufnahmen für die Verbunddatenbank (die noch einer fachlichen Prüfung durch Bibliothekare (Hochkatalogisierung) - unterliegen) als auch Frontdoors generieren zu können.

Das Verfahren sieht im Detail folgendermaßen aus (Siehe Abb. 4):

- Die zu den Objekten gehörenden Metadaten werden auf den Server des BSZ transferiert.
- Aus diesen Metadaten wird ein Format für den Import in die Verbunddatenbank generiert.
- Nach dem Import werden die Titelaufnahmen korrigiert
- Die korrigierten Titelaufnahmen werden in die lokalen OPACs eingespielt

[12] Die Verbunddatenbank ist die bibliographische Datenbank des Südwestdeutschen Bibliotheksverbundes, die vom BSZ betrieben wird. Sie enthält derzeit 8 Mio. Titel von Büchern, Zeitschriften usw. sowie 22 Mio. Bestandsnachweise von wissenschaftlichen Bibliotheken in Baden-Württemberg, Rheinland-Pfalz und Sachsen.

- Aus den korrigierten Titelaufnahmen werden - ergänzt durch Informationen aus den gelieferten Metadaten, die in der Verbunddatenbank nicht abgebildet werden können Frontdoors erstellt.

Aus dem SWIB-Suchdienst heraus ist dann über diese Frontdoors, die von SWIB-Suchdienst indiziert werden, der Zugriff auf die Objekte möglich[13]. Zusätzliche Zugriffsmöglichkeiten: OPACs, andere Suchmaschinen...

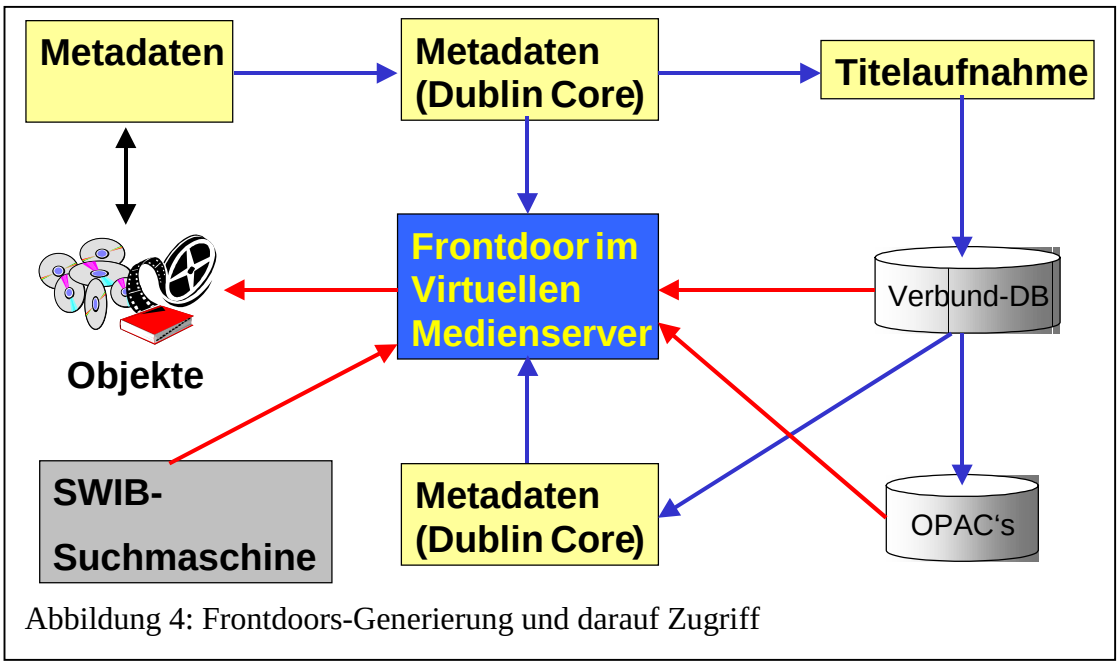

Abbildung 4: Frontdoors-Generierung und darauf Zugriff

[13] Diese Indexierung der Frontdoors stellt den zentralen Aufgabe des SWIB-Suchmaschine, die im Rahmen dieser Diplomarbeit realisiert wird.

In der Regel haben Frontdoors folgende Aufbau:

Abbildung 5: Aufbau eines Frontdoors

Der Header mit der DCMES-Metadaten für diese Frontdoor sieht folgendermaßen aus:

```
<HEAD>
<TITLE>Empfehlungen zum Aufbau eines Servernetzes f&uuml;r elektroni-
sche Hochschulpublikationen</TITLE>
...
<META NAME="DC.DATE.CURRENT" SCHEME="ANSI.X3.30-1985"
                                        CONTENT="19980911">
<META NAME="DC.CREATOR.NAME"            CONTENT="Lehmann, Andreas">
<META NAME="DC.CREATOR.CORPORATE"        CONTENT="Arbeitsgruppe Volltex-
te und Hochschulpublikationen / Baden-W&uuml;rttemberg">
<META NAME="DC.TITLE"                     CONTENT="Empfehlungen zum Auf-
bau eines Servernetzes f&uuml;r elektronische Hochschulpublikationen">
<META NAME="DC.PUBLISHER"                    CONTENT="Bibliotheksservice-
Zentrum Baden W&uuml;rttemberg">
<META NAME="DC.TYPE"                    CONTENT="ResearchPaper">
<META NAME="DC.FORMAT"   SCHEME="IMT"   CONTENT="Text/html">
<META NAME="DC.LANGUAGE" SCHEME="NISOZ39.53" CONTENT="GER">
<META NAME="DC.SOURCE"                  CONTENT=" SWB-IDNR/06853597">
<META NAME="DC.IDENTIFIER"              CONTENT="(SCHEME=URL)
....
</HEAD>
```

2.5 DC-Elemente vom BSZ

Die 15 Elemente des Dublin Core Metadata Element Set, die beim BSZ verwendet werden, lehnen sich an das Element Set, welches von der Dublin Core Metadata Initiative als DC Version 1.0 im Dezember 1996 veröffentlicht worden ist.

Interpretation der DC-Elemente beim BSZ[14]

1. **title**

   Titel, der vom Verfasser, Urheber oder Verleger vergebene Name (=value) der Ressource.

   Alternative Titel können in weiteren Elementen aufgeführt werden, z.B. für Subtitel, übersetzter Titel usw.

2. **creator**

   Autor (empfohlen: „Familienname, Vorname“ als normierte Schreibweise) oder Körperschaft. Die Person(en) oder Organisation(en), die den intellektuellen Inhalt verantworten.

   Im Falle mehrerer Autoren jeder weitere in einem zusätzlichen Meta-Element.

3. **subject**

   Schlagwörter aus einem kontrollierten Vokabular. Das Thema der Ressource bzw. Schlagwörter oder Phrasen, die das Thema oder den Inhalt des Dokuments beschreiben.

4. **description**

   Inhaltliche Beschreibung. Eine textuelle Beschreibung des Ressourceninhalts inklusive eines Referats (Abstract) bei dokumentähnlichen Ressourcen oder Inhaltsbeschreibungen bei graphischen Ressourcen.

5. **publisher**

   Verleger / Herausgeber. Einrichtung, die verantwortet, daß diese Ressource in dieser Form zur Verfügung steht, wie z.B. ein Verleger, ein Herausgeber, eine Universität oder ein Unternehmen.

6. **contributors**

[14] Vgl. http://www.bsz-bw.de/diglib/medserv/konvent/metadat/dcsyntax.html

Sonstige Beteiligte. Zusätzliche Person(en) und Organisation(en) zu jenen, die im Element 2 (creator) genannt wurden, die einen bedeutsamen intellektuellen Beitrag zur Ressource geleistet haben, deren Beitrag aber sekundär im Verhältnis zu denen im Element 2 (creator) zu betrachten ist (z.B. Herausgeber, Übersetzer, Illustratoren, auch Konferenzleiter, Moderatoren).

7. date

Datum, an dem die Ressource in der gegenwärtigen Form zugänglich gemacht wurde, in normierter Schreibweise.

8. type

Ressourcenart, Art der Publikation wie z.B. Dissertation, Diplomarbeit, Gedicht, Homepage, Essay usw.. Verwendet wird **ein kontrolliertes Vokabular** aus einer Liste zugelassener Bezeichnungen.

9. format

Datentechnisches Format der Ressource, z.B.: Text/HTML, ASCII, Postscript-Datei, ausführbare Anwendung, JPEG-Bilddatei usw..

Die Angabe für dieses Element muß einem kontrollierten Vokabular entnommen werden, z.B. den angemeldeten Internet Media Types (MIME Types).

10. identifier

Ressourcenidentifikation, Zeichenkette oder Zahl, die eine eindeutige Identifikation des Dokuments ermöglicht (z.B. ISBN, ISSN). Bei Internet-Ressourcen sind bspw. URLs und URNs vorgesehen.

11. source

Quelle, SWB-ID-Nr[15]. der Titelaufnahme des Dokuments.

12. language

Sprache, Sprache(n) des intellektuellen Inhalts der Ressource als Sprachcode, in der der Inhalt des Dokuments vorliegt.

Darstellung 3-stellig aus dem normierten Sprachcode ISO 639-2.

13.relation

Verhältnis zu anderen Ressourcen.

14.coverage

Abdeckungsspektrum, Abdeckungsaspekte (zeitliche, örtliche, flächenhafte etc.

[15] Jede Titelaufnahme in der SWB-Verbunddatenbank erhält eine eindeutige Identifikations-Nr

Aspekte, die zur Charakterisierung des Objekts sinnvolle Ergänzungen geben, z.B. bei Karten, Bildern etc.).

15.rights management

Rechtliche Bedingungen, vorgesehen für den Inhalt dieses Elements ist ein Link (URL oder ein anderer passender URI) zu einem Urhebervermerk, ein "Rights-Management"-Vermerk über die rechtlichen Bedingungen oder ggf. zu einem Server, der solche Informationen dynamisch erzeugt.

3 Internet-Suchdienste und deren Funktionsweise

Eine zielgerichtete Suche im Internet war bei der Entwicklung der Standards und der Strukturen des Internets nicht unbedingt berücksichtigt worden. Es konnte auch niemand ahnen, daß sich das Internet so sprungartig verbreitete wie in den letzten Jahren. Durch den explosionsartigen Anwuchs ist ein Kommunikations- und Informationsangebot von nie dagewesener Größe entstanden. Schätzungen über die Zahl der dieses Angebot nutzenden Menschen verbieten sich von selbst angesichts des rasanten Wachstums. Das Gleiche gilt für Schätzungen über die „Größe“ des Internets oder zur Zahl der angebotenen Seiten. Bei den neu angeschlossen Rechnern ans Internet hat sich die Wachstumsrate jedes Jahr ungefähr verdoppelt.

Um sich in so einer sehr dynamischen, wachsenden, unübersichtlichen Umgebung zu Recht zu finden sind die Suchdienste unabdingbar. Allerdings, die generelle Suchdienste reihen für die gezielte suche im Internet, bedingt der Schwerigkeiten bei der Recherche im Internet und dessen Struktur, nicht aus.

3.1 Suchdienst in lokalen WWW-Servern / Gateways zu Datenbanken

„Die Stichwortsuche innerhalb eines WWW-Servers war eine der ersten Möglichkeiten, dem Benutzer die gezielte Suche nach Informationen im WWW zu ermöglichen. Dabei handelt es sich um eine einfache Stichwortsuche, die auf das Dokumentverzeichnis des lokalen WWW-Servers zugreift. Eine der ersten gezielten Suchmöglichkeit war die Stichwortsuche innerhalb eines WEB-Servers; es handelt sich um eine Stichwortsuche, die auf das Dokumentverzeichnis des lokalen WEB-Servers zugreift. „Dieses einfache Suchmöglichkeit besteht seit der Entstehung von WEB durch HTML und HTTP; über das HTML-Element <ISINDEX> wird die Eingabe von Suchworten innerhalb einer HTML-Seite definiert, die auf den lokalen WEB-Server befindet. Dort eingegebenen Suchbegriffe werden nach Auslösen der Suche mit einem vorangehenden „?“ und durch ein „+“ voneinander getrennt an die Dokumentadresse angehängt und die Anfrage an den Server gesendet. Wurden z.B. Hypertext und Information als Suchbegriffe eingege-

ben, so hängt der WWW-Browser „?hypertext+information“ in seinem Request an die Dokument-URL an. Darauf sucht der WWW-Server in seinem Datenbestand nach Dokumenten , in denen die Suchbegriffe vorkommen, und liefert eine HTML-Seite mit entsprechenden Verweisen zurück[16]*“*.

Bei Suchabfragenbearbeitungen, die auf einer gewissen Dokumentstruktur, bestimmter Felder oder Relevanzgrade beruhen sollen, müssen zusätzliche Softwarekomponenten in Verbindung mit Datenbanken auf der Server-Seite eingesetzt werden. Hierfür eignet sich die CGI-Schnittstelle. Es gibt momentan so gut wie keine Java-Variante auf dem Gebiet.

In HTML-Formulare eingetragene Suchbegriffe können vom Browser an die CGI-Schnittstelle weitergegeben werden und von dort an spezielle Hintergrundsprogramme weitergeleitet werden. Information Retrieval-Methoden können bei diesem Verfahren beinahe uneingeschränkt eingesetzt werden. Auf diesem Prinzip beruht auch die WWW-Schnittstelle der Harvest-System (Siehe Kap. 5).

Um den Nutzer ein möglichst professionelle Suche im lokalen Datenbestand zu ermöglichen, bieten inzwischen fast alle WWW-Server den sog. Gateway an. Das Suchabfragen bei dieser Gateway-Lösung nicht nur auf den lokalen Datenbestand beschränkt sind, bildet einen weiteren Vorteil. Inzwischen nutzen viele Datenbankanbieter, Bibliotheken, Warenhäuser diese Technik um ihre Bestände über komfortable WWW-Schnittstellen anzubieten.

Eine Beispiel ist die Literaturrecherchen in Bibliotheken; über ein Gateway wird zu den vorhandenen OPAC-Katalogen eine Suche unter Verwendung diverser literaturüblicher Suchfelder (Autor, Titel, Verlag, Erscheinungsjahr usw.) ermöglicht (Siehe Abb. 1).

[16] Vgl. http://www.inf-wiss.uni-konstanz.de/suche/such_tutorial.html

3.2 Katalog- und verzeichnisbasierte Suchdienste[16]

Sind bestimmte Internetressourcen nach Themengebieten geordnet und in einem „Katalog“ zusammengefasst werden, so spricht man von einer Katalog- und verzeichnisbasierten Suche. Bei diesem Suchverfahren navigiert sich der Nutzer durch hierarchisch, thematisch aufgebauten Verzeichnissen. Die in geeignete Themen unterteilten Verzeichnisse, sind in Form von Oberkategorien (wie z.B. Wirtschaft, Politik, Sport, Kultur, Bildung etc.) aufgebaut (Siehe Abb. 6). Sie werden als Einstiegsknoten zur weiteren Verzweigung benutzt. Wenn der Suchende die passende Kategorie zu seinem Thema gefunden hat, verfügt er über einen guten Ausgangspunkt zu vielen Verweisen zu seinem Thema. Anbieter von Suchmaschienen muessen Ihre Informationsangebote sinnvoll gliedern, damit der Suchende nicht von der Vielzahl der Information erschlagen wird. Hierbei kommen dann auch die sog. Unterkategorien ins Spiel, die wiederum sinnvoll untergliedert sein können.

Diese Verzeichnisse dienen für den Einstig in Recherchen zu beliebigen Themen. Analog zu den „Gelben Seiten“ sind Kataloge besonders zu Einstieg in ein gewisses Thema geeignet. Der Nutzer kann ohne das er nach einem konkreten Stichwort sucht, sich in das von ihm gewünschte Thema einarbeiten. Durch diese Eigenschaft kann der Nutzer an Informationen gelangen ohne zuvor bestimmte Begriffe zu wissen. Bei einem Einstieg in eine Thematik entsteht bei dieser Suchart ein gewisser Entdeckungseffeckt, der bei einer gezielten Stichwortsuche eher ausbleibt.

Durch das explosionsartig gestiegene Informationsangebot erreichten die Kataloge sehr schnell eine Größe, durch die das navigationsartige Browsen eher mühsam wurde und nicht mehr für tauglich schien. Aus diesem Grunde wurde die Stichwortsuche durch diverse Suchfunktionen, wie z. B. Boole’sche Ausdrücke ausgestattet. Da diese Suchart allerdings nicht auf der Volltextsuche beruht, sondern auf Link-Texten der referenzierten Dokumenten basiert, ist sie meist für den Suchenden sehr unzufriedend.

Die Anbieter von Katalogen unterscheiden sich in der Strenge bei der Aufnahme neuer Einträge in ihre Verzeichnisse. Je nach Konzept des Anbieters, werden neue Informa-

tionen ausschließlich von ihm selber gesucht oder die Kunden besitzen die Möglichkeit Informationsneuigkeiten selber in den Katalog einzubringen.

Abbildung 6: Einstiegseite von Yahoo-Deuschland (http://www.yahoo.de)

Weitere Beispiele für thematischen Verzeichnisse:

- Yahoo, „http://www.yahoo.com“
- Web.de, „http://web.de“
- DINO, „http://www.dinoonline.de“
- WWW Virtual Library „http://vlib.org/Overview.html“

3.3 Roboterbasierte Suchdienste

Durch den Einsatz von Suchmaschinen besitzt der Nutzer die Möglichkeit mit einer einzigen Abfrage große Teile des Internets zu durchsuchen. Hierbei werden die Suchbegriffe aus dem Feld der Einstiegsseite eingelesen, abgearbeitet und die Suchergebnisse in Form einer Liste von Links dargestellt. Dieses Ergebnis wird durch sog. Robotor bzw. Gatherer (Siehe Kap. 5) erreicht. Sie durchsuchen das Internet, wobei sie alle erreichbaren Dokumente herunterladen, um sie anschließend aufzuarbeiten und ganz oder auszugsweise in einer Datenbank abzuspeichern.

Eine leistungfähige Suchmaschine anzubieten, die eine grosse Datenbasis aufweisst, sie regelmäsig aktualisiert und die Abfragen auf dieser Datenbasis schnell bearbeitet, erfordert eine umfangreiche Hard- und Software Ausstattung.

Die Hardwareausstatung des Meta-Suchdienstes Acoon (http://www.acoon.de) sieht folgendermaßen aus[17]:

- 3 unter Windows NT 4.0 laufenden PCs.
- Alle drei Computer sind mit jeweils zwei 400MHz-Pentium-II CPUs, einer 18gb SCSI-Festplatte sowie einer 100MBit-Netzwerkkarte ausgestattet.#
- Computer Nummer 1 beinhaltet den eigentlichen Web-Robot. Dieser Computer ist mit 512 MB RAM ausgestattet.
- Computer Nummer 2 enthält 1 GB RAM sowie 400 GB Festplattenkapazität aufgeteilt in 3 RAID-5 Arrays und ein RAID-0 Array für temporäre Dateien. Dieser Computer enthält die Rohdatenbank. Er ist außerdem für die tägliche Neuerstellung des Suchindex verantwortlich.
- Computer Nummer 3 hat dieselbe Ausstattung wie Nummer 2. Jedoch sind hier die 400 GB Plattenkapazität auf 2 RAID-5 Arrays aufgeteilt, die jeweils den Suchindex enthalten. Sollte dieser Computer einmal ausfallen, so kann Computer Nummer 2 seine Aufgaben übernehmen.

„Diese Computer sind in der Lage über 2 Millionen URLs pro Tag zu bearbeiten. Eine Leistung die ausreicht um den kompletten Datenbestand jeweils etwa alle 10-14 Tage

komplett zu überprüfen, was natürlich eine hohe Aktualität des Datenbestandes bewirkt.“

Aconn setzt selbst entwickelte Software mit Inprise Delphi unter WIN-NT ein.

Eine Beispiel für die Suchmaschinen: die Lycos(http://www.lycos.de/):

Abbildung 7: Suchmaschine Lycos

Weitere Beispiele für die Suchmaschine:

- AltaVista „http://altavista.com[.de]“

[17] Vgl: http://www.acoon.de/infos.html

- Excite „http://www.excite.com[.de]“
- Webcrawler „http://www.webcrawler.com“
- Businesswebsource „http://www.businesswebsource.com/“

3.3.1 Arbeitsweise der Suchmaschinen[18]

Die Erstellung der *indizierten lokalen Datenbank* mittels Abruf der *Remote Informationressourcen* erfolgt bei allen roboterbasierten Suchmaschinen ähnlich:

- Identifizierung von Informationsressourcen
- Abholen bzw. Herunterladen der identifizierten Information
- Analyse, Extrahierung und Indizierung der heruntergeladenen Information
- Aktualisierung der indizierten Datenbasis
- Suchanfragen entgegennehmen und Abarbeiten

Der technische Aufbau einer Suchmaschine am Beispiel des Suchdienstes WebCrawler (http://www.webcrawler.com):

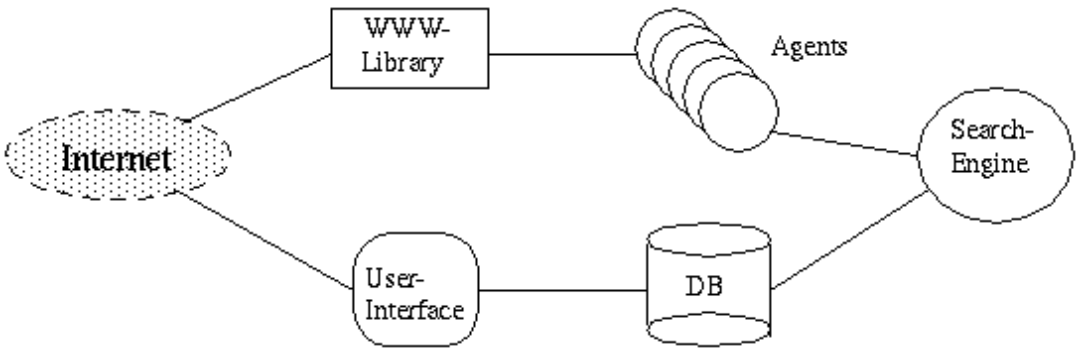

Abbildung 8:Systemaufbau der Suchmaschine WebCrawler16

„Die Suchmaschine betrachtet das Web als riesigen gerichteten Graphen, wobei Knoten WWW-Dokumente und gerichtete Kanten die Verweise darstellen, die von einem Dokument ausgehen. Von einem bestimmten Knoten aus wird dann der Graph entlang den Kanten abgearbeitet. Bei jedem so erreichten Dokument wird von der Suchmaschine eine lexikalische Analyse durchgeführt, bei der inhaltsrelevante Terme aus dem Dokument extrahiert und in der Datenbank (DB) abgelegt werden. Das Abrufen der WWW-Dokumente erledigen parallel laufende Agenten-Prozesse. Diese geben der Suchmaschine entweder das gewünschte HTML-Dokument oder eine entsprechende Fehlermeldung, warum auf das gewünschte WWW-Dokument nicht zugegriffen werden konnte. Ferner nutzen die Agenten den Katalog WWW Virtual Library (http://vlib.org/Overview.html) für die Dokumentbesorgung. Der Benutzer greift auf das System über den Suchserver (User-Interface) zu, der zwecks Anfragenbearbeitung auf die von der Suchmaschine aufgebaute Datenbank zugreift[16].“

Das Einsammeln der Daten (Gatherer)

Durch sog. „Robots“, „Spiders“ oder „Gatherer“ (Siehe Kap. 5) werden die Daten eingesammelt. Dies erfolgt in bestimmten Zeitabschnitten.
Bei dieser Aufgabe machen sie von der Webstruktur des Internets gebrauch. Sie gehen von Dokumenten, die sie bereits kennen aus und verfolgen die in diesen Dokumenten enthalten Links zu anderen Dokumenten. Diese neu gefundenen Dokumenten werden wiederum auf Links und Querverweise geprüft. Ein Robot bewegt sich so schrittweise rekusionsartig durch das Netz und identifiziert dabei neue Informationsangebote.

Robots finden sehr schnell Webserver, die neu an das Internet angeschlossen worden sind. Hierzu muß lediglich in einem dem Robot bereits bekannten Dokument, ein Links zu einer Webseite des neuen Servers eingefügt werden. Ist dies geschehen wird der Robot auch den Rest des neuen Servers durchsuchen und dessen Inhalt indizieren.

Sämtliche Robots besitzen ein gemeinsames Merkmal: ausgehend von einer bereits bekannten Anzahl von Informationsangeboten werden die eigenen Datenbanken stets mit neuen Inhalten erweitert.

[18] Vgl. Effektive Suche im Internet, Ulrich Babiak

Analyse und Indexierung der eingehenden Daten

Nach dem Einsammeln der Dokumente müssen diese für die Aufnahme in die Datenbank aufbereitet werden. Wieviel Aufwand für die Ausarbeitung und Analyse der Dokumente investiert wird, hängt von den einzelnen Suchmaschinen ab. Sie werden allerdings stetig weiterentwickelt um ihren Leistungsstand zu erweitern. Die hierfür neu entwickelten Methoden werden von den meisten Anbietern geheim gehalten.

Gemäß Algorithmen und Methoden der jeweiligen Suchmaschine werden die neu gefundenen Dokumente für den Datenbankeintrag aufgearbeitet. Um die Datenbank schlanker zu halten, speichern einige Suchmaschinen nur die Titel, Überschriften und die ersten paar Sätze. Die meisten Suchmaschinen wie auch das Harvest-System erfassen die neuen Dokumente im Volltextmodus.

Die Analyse der Daten zeichnet sich dadurch aus, dass einzelne Wörter aus den gesammelten Daten extrahiert und ihre Textposition festgehalten wird. Dadurch kann unterschieden werden, ob ein Wort im Titel, im Textteil, in der Abschnittsüberschrift, in einer URL oder in einem anklickbaren Hyperlink vorkommt. Je nach Einstellung der Suchmaschine können auch weitere Elemente wie z.B. eingefügte Bilder usw. erfaßt werden. Die Genauigkeit der Suchanfrage hängt im allgemeinen von der Genauigkeit dieser Systemeinstellung ab.

Die Sortierung der Treffer nach ihrer Relevanz (Ranking)

Um den wichtigsten Suchtreffer als erstes anzuzeigen, verwenden die Suchmaschinen unterschiedliche Methoden zur Relevanzfeststellung. Dabei ist meistens die Wortstellung im Dokument entscheidend. Leider wird hierbei in den meisten Fällen keine Praxistauglichkeit erreicht.

Da je nach Spezifizierung der Suchanfrage eine Vielzahl von Treffern erhalten werden kann und schon das Durchforsten der ersten 10 – 20 Treffern einen enormen Zeitaufwand bedeutet, verwenden die Suchmaschinen sog. Rankings. Alle Treffer werden in einer Liste dargestellt, wobei der wichtigste Treffer ganz oben in der Liste erscheint.

Durch diesen gewichteten Aufbau der Liste soll erreicht werden, dass die vom Nutzer gesuchte Information möglichst schnell gefunden werden kann. Die Wichtigkeit des gefundenen Treffers hängt dabei davon ab, inwiefern der Suchtreffer mit der Suchanfrage übereinstimmt. Je größer diese Übereinstimmung ist, um so wichtiger ist der Treffer. Idealerweise würde man also keine lange Liste angezeigt bekommen, sondern nur das direkt gesuchte Dokument.

Berechnung der Relevanz von Suchergebnissen

Bei der Berechnung der Relevanz verwenden die Suchmaschinen unterschiedliche Verfahren, die in der Regel nicht veröffentlicht werden.

Es gibt aber eine Konsens über die Kriterien, die bei der Berechnung der Relevanz zugrunde liegen:

- **Die Anzahl der gefundenen Wörter**

  Je mehr Suchworte in einem Dokument gefunden werden, desto höher wird es eingestuft. Daher bei der Trefferlisten sind die wichtigen, d.h. die am meisten den Suchbegriff beinhalten, stehen am Anfang der Trefferliste. Werden mehrere Suchworte verknüpft, so werden Ergebnisse, die alle oder viele der gesuchten Begriffe oder Phrasen enthalten, als relevanter eingestuft.

- **Die Position der Wörter**

  Die Position des Wortes im Text gilt als Anhaltspunkt für seine Wichtigkeit.

  - ***Domain und URL:*** Auf Systemen, die lange Dateinamen zulassen, werden Dokumente oft unter einem aussagekräftigen Namen gespeichert. Die Suchmaschine wertet das Dokument bei Übereinstimmung mit dem Suchwort als besonders relevant. Das gilt ganz besonders, wenn es sich um den Domainnamen handelt.

- ***Titel:*** Ein Dokument mit dem Suchwort im Titel hat gute Chancen auf einen vorderen Platz; es ist relevanter als ein Dokument, in dem der Suchbegriff nur im Fließtext vorkommt

- ***Überschrift:*** Enthält eine Überschrift das gesuchte Wort, befaßt sich das gesamte Dokument oder ein wesentlicher Teil damit. Häufig werden aus ästhetischen Gründen Überschriften als Grafiken eingebunden. Die Indexierung und Gewichtung wird dadurch erschwert.

- ***Meta-Tag:*** Die Maschinen, die Meta-Tags auswerten, ordnen Dokumente, die den Begriff im Content (Inhalt) oder Keywords (Schlüsselworte) Tag führen, höher ein.

- ***Dokumentenanfang:*** Je früher das Wort im Dokument auftaucht, desto relevanter für das Suchergebnis wird es gewichtet.

- **Der Abstand der Suchbegriffe im Dokument**

  Wenn in einem Dokument einige der Suchbegriffe nahe beieinander liegen, so wird angenommen, dieses Dokument sei relevanter als ein Dokument, in dem die Suchbegriffe zwar auch, aber mit großen Abstand zueinander enthalten sind.

- **Die Häufigkeit von Suchbegriffen innerhalb von Dokumenten**

  Kommt ein Suchbegriff mehrmals in einem Dokument vor, wird dieses Dokument höher eingestuft als Dokumente, in denen der Suchbegriff nur einmal vorkommt

- **Die Gesamthäufigkeit einzelner Wörter**

  Wörter, die in der Datenbank insgesamt besonders häufig vorkommen, wie z.B. „Internet“, sind weniger dazu geeignet, ein Dokument besonders zu kennzeichnen, und fallen darum weniger ins Gewicht. Wörter, die in der Suchdatenbank seltener sind, werden als spezifischer angesehen und deshalb höher bewertet.

Diese Kriterien fließen mit unterschiedlicher Gewichtung ein, wenn die Suchergebniss bzw. die Trefferreihenfolge berechnet werden. Dabei wird für alle Treffer ein Score berechnet, die entweder in Prozenten oder in absoluten Zahlen erfolgt.

„Als einzige mir bekannte Maschine veröffentlicht Web.de den Algorithmus zur Relevanzbewertung der Suchtreffer[19]*:*

- *Jeder Treffer im Namen:* ***5 Punkte***
- *auf einer Wortgrenze:* ***1 Punkt zusätzlich***
- *am Beginn des Namens:* ***2 Punkte zusätzlich***
- *Jeder Treffer im Info Teil:* ***1 Punkt***
- *auf einer Wortgrenze:* ***1 Punkt zusätzlich***
- *am Beginn des InfoTeils:* ***1 Punkt zusätzlich“***

Rankingservice

Jeder der in Internet eine Seite veröffentlicht, hat eine Interesse daran, auch durch Suchmaschinen gefunden zu werden. Die Suchdienste bieten meistens kostenlose Anmeldung der neuen Seiten an. Nun wer dazu noch möchte, dass seine Internetseiten auch bei der potentialen Suchanfragen in der Suchtrefferliste die höheren Plätze belegt, sollte bei der Vorbereitung ihrer Internetseiten oben erwähnten Kriterien des Ranking in Betracht zu ziehen. Es gibt mittlerweile kommerzielle Anbieter „Rankingservices“, die ihrem Kunden eine hohen Stellenwert innerhalb der Suchtrefferliste versprechen; sie schalten bei der Anmeldung an der Suchmaschinen ihre spezielle auf die Seiten abgestimmten Rankingseiten ein, die auf die auf die eigentliche Seite verweisen.

[19] Vgl.http://suchfibel.de/3allgem/index.htm

3.4 Hybride-Suchmaschinen

Hybride-Suchmaschinen beinhalten eine Kombination von mehreren Suchverfahren, wobei eine Suchmaschine mit einen Katalog verbunden wird. Über Gateways kann auch eine Verknüpfung zu verschiedenen Informationsquellen erstellt werden:

- email-Verzeichnisse
- Telefon- / Adressbücher
- Verzeichniss von Unternehmen und Organisationen (Gelbe Seiten)
- Zugriff auf kommerzielle Datenbanken

Diese Verknüpfung basiert meist auf Kooperationen oder Fusionen von Suchdienstbetreibern[16].

3.5 Metasuchmaschinen

Erlaubt eine Suchmaschine die gleichzeitige Suche bei verschiedenen Suchdiensten, so spricht man von einer Metasuchmaschine (Multi-Search). Dabei werden mehrere Suchdienste über ein einziges Suchformular angesprochen, z.B. sucht die Metasuchmaschine MetaGopher (Siehe Abb. 9) gleichzeitig in GoTo.com, Yahoo, Hotbot, Go Network, Webcrawler, Looksmart, und Altavista.

In der Praxis werden einfache WWW-Seiten mit mehreren Suchmasken verschiedener Suchdienste ebenfalls als Metasuchdienste bezeichnet. Dies ist allerdings nicht korrekt. Unter diesen „unechten“ Metasuchdienste verbirgen sich trotzdem einige Vorteile:

- Suchformulare mehrerer Suchmaschinen werden auf einer WWW-Seite angeboten
- Erspart das Laden der Homepages einzelner Suchmaschinen
- Guter Überblick über verschiedene Suchmaschinen

„Echte“ Metasuchdienste weisen folgende Charakteristiken auf:

- *Mehrere Suchdienste, in der Regel meist nur Suchmaschinen und Kataloge, werden automatisch über eine Schnittstelle (Suchformular) befragt*
- *Die verschiedenen Suchdienste werden vorgegeben, können manchmal aber auch vom Benutzer ausgewählt werden*
- *Funktionalität und Operatoren der verschiedenen Suchdienste werden verwendet. Hierbei wird eine Anpassung der Anfrage auf die einzelnen Suchdienste vorgenommen.*
- *Eliminierung von Mehrfachtreffern aus den Ergebnissen der verschiedenen Suchdienste Zeitvorgaben und maximale Treffergrenzen können gesetzt werden"*[16]

Durch die gemischte Treffermenge der einzelnen Suchdienste wird das Ranking zum zentralen Problem der Metasuche. Da die Kriterien des Rankings wie bereits erwähnt nicht bekannt ist, beschränken sich Metasuchdienste auf die Gruppierung der einzelnen Treffer nach der Suchmaschine. Mehrfachtreffer werden zur Zeit nur durch den Vergleich der URLs bereinigt. Gleiche Dokumente mit unterschiedlicher URL werden bisher noch nicht eleminiert, da eine inhaltliche Analyse bis jetzt nicht stattfindet.

Es gibt zwei unterschiedliche Techniken zur Befragung der Suchdienste durch die Metasuchmaschine: Der sequentielle und der parallele (gleichzeitige) Zugriff:

Sequentielle Suche in mehreren Suchdiensten
Der Metasuchdienst befragt mehrere Suchdienste nacheinander, wobei die Trefferausgabe erst nach der Abfrage des letzten Suchdienstes erfolgt.

Parallele Suche in mehreren Suchdiensten
Der Metasuchdienst befragt meherer Suchdienste parallel, wobei die Trefferausgabe sofort nach dem Ergebnis des ersten Suchdienstes erfolgt.

Metasuchdienste sind i.d.R. auf dem neuesten Stand. Sie vereinen auch Suchdienste und Datenbanken, die nicht so bekannt sind. Hier liegt ein grosser Vorteil der Metasuchdienste. Gerade bei speziellen Suchanfragen, bei denen ein einzelner Suchdienst nur

wenige Treffer anbieten kann, wird durch einen Metasuchdienst eine größere Informationsauswahl erreicht.

Abbildung 9: Metasuchmaschine MetaGopher (http://metagopher.com)

Weitere Beispiele für die Metasuchdienste:

- MetaXplorer „http://www.ubka.uni-karlsruhe.de/suchmaschinen/metaxa/“
- MetaGer „http://meta.rrzn.uni-hannover.de/“
- SavvySearch „http://www.savvysearch.com“

3.6 Intelligente / Mobile Agenten

Die Komplexität des Internet vergrößerte sich im Laufe der Zeit durch das sprunghafte Anwachsen der weltweiten Vernetzung. Hierdurch wurde die Entwicklung neuer Methoden, die dem Internetnutzer ein effizientes und zielgerichtete Recherche ermöglicht, erforderlich.

Aus diesem Grunde werden weiterführende Konzepte aus dem Bereich der künstlichen Intelligenz erprobt. Man spricht hier von sog. „Intelligenten“ oder „mobilen“ Agenten. Dabei handelt es sich um Programme, die den Suchenden mit einem gewissen Grad an Eigenständigkeit unterstützt, um einen möglichst großen Erfolg zu erzielen.

„Es gibt momentan keine allgemein gültige und akzeptierte Definition des Begriffs »Agenten« jedoch gibt es doch einige Charakteristiken, die solche Agenten gemeinsam haben sollten:

Ziel
Der Benutzer muß in der Lage sein, dem intelligenten Agenten komplexe Anfragen stellen zu können. Die Entscheidung, wie die Aufgabe in Teilaufgaben zerlegt wird, wo und wann die erforderlichen Informationen zu finden sind, obliegt dann dem Agenten. Der Benutzer braucht sich dann nur um das „Was“ und nicht um das „Wie“ seiner Anfrage zu kümmern.

Flexibilität
Die Aktionen eines Agenten zum Lösen einer Aufgabe sind zu keinem Zeitpunkt festgelegt. Der Agent ändert sein Verhalten, wenn ihn äußere Umstände dazu zwingen oder wenn die Teillösung einer Aufgabe neue Aspekte hervorbringt, die eine geänderte Vorgehensweise zum Erreichen eines Zieles nahelegen. Zu den äußeren Umständen gehört z.B. eine Veränderung der Softwareumgebung oder ein nicht erreichbarer Server im Internet.

Mitarbeit

Der Agent nimmt nicht nur blind Kommandos entgegen, sondern rechnet auch damit, daß der menschliche Benutzer Fehler macht, wichtige Informationen ausläßt oder daß Mehrdeutigkeiten aufgedeckt werden müssen. Diese Unklarheiten müssen dann durch geeignete Mittel, wie zum Beispiel Nachfrage beim Benutzer, Heranziehen einer Wissensbasis u.a., beseitigt werden.

Selbststart

Ein Agent ist in der Lage, eine Aktion zu starten, ohne unmittelbar vom Benutzer angesprochen worden zu sein. So könnte er beispielsweise automatisch eine Suche im WWW beginnen, wenn die Netzbelastung relativ niedrig ist. Dies kann auch geschehen, wenn der Benutzer gerade nicht am Computer eingeloggt ist.

Kommunikationsfähigkeit

Ein intelligenter Agent muß in der Lage sein, auf effektive Art und Weise Informationen beschaffen zu können. Dazu dient die Kommunikation mit der Softwareumgebung, mit anderen Agenten und nicht zuletzt mit dem Benutzer. Zur Inter-Agenten-Kommunikation dienen sogenannte Agent-Communication-Languages (ACL). Damit sind Agenten nicht nur fähig, Wissen und Informationen von anderen Agenten zu bekommen, sondern können auch gemeinsam an der Lösung einer Aufgabe arbeiten. Zur Kommunikation mit dem Benutzer dienen heute meist noch Dialogboxen; natürlichsprachliche Zugänge werden erforscht.

Anpassungsfähigkeit

Agenten sollen sich an den Benutzer anpassen können, d.h. seine Gewohnheiten und seine Arbeitsweise kennen.“[16]

Die oben genannten Eigenschaften können mit dem heutigen Stand der Technik noch nicht realisiert werden. Ein Beispiel für mobilen Agent ist BargainFinder unter URL: http://bf.cstar.ac.com/bf/ zu sehen.

3.7 Entwicklung der verschiedenen Typen von Suchdiensten[20]

Die Anzahl der Suchdienste hat seit 1994 stetig zugenommen. Sie bekommen kaum noch neue Konkurrenten. Sie werden von Privatfirmen betrieben, die meisten sind jedoch für die Endnutzer kostenlos.

Darüberhinaus existieren noch mehrere Tausend Suchdienste und Datenbanken für spezielle einzelne Typen von Web Ressourcen wie email Adressen und andere Informationsprotokolle (z.B. FTP).

„Globale Listen- oder formularbasierte Dienste, hauptsächlich auf das manuelle Registrieren von Ressourcen aufbauend, verlieren in ihrer Mehrzahl stark an Bedeutung wegen wesentlicher Schwächen was Grösse, Deckungsbereich, Aktualisierung und Retrievalmöglichkeiten angeht. Sie werden wohl nur überleben, wenn enorme Investitionen getätigt werden und wenn sie sich mit roboterbasierten Diensten zusammentun, was auch teilweise schon geschehen ist. Positiv sind dagegen die speziellen Auswahl- und Strukturierungsmöglichkeiten, die sich aus der manuellen Arbeit und der Beteiligung der Nutzer ergeben."

Regional bzw fachspezifischen Dienste, die hauptsächlich von öffentlichen Institutionen (Universitäten, Bibliotheken usw.)angeboten werden, nehmen in den letzten Jahren stark zu. Es gibt sie sowohl als robotergenerierte Datenbanken als auch als themenorientierte Browsingkataloge.

Sie bieten dadurch eine verbesserte Suche, da sie schon durch ihren regionalen oder fachlichen Deckungsbereich eine Eingrenzung bieten. Die Grundfunktion einer regionalen Eingrenzung bieten insofern auch Dienste, die die Begrenzung der Suche auf Internet Domänen, Domänengruppen oder Länder ermöglichen.

„Ganz im Gegensatz zu der grossen Zahl und dem frühen Erscheinen von fachlich spezialisierten Linklisten und Directories jedweder Art, steht die bisher immer noch geringe Zahl von robotergenerierten Datenbanken für ein Fachgebiet. Das ist natürlich zum

[20] Vgl.: http://www.lub.lu.se/tk/demos/DGD97.html

Teil auch methodischen Problemen geschuldet, wie ein Internet Harvesting Robot gleichzeitig ein Fachgebiet weitestmöglich erfassen und möglichst auf für das Fach relevantes Material beschränkt werden soll.“

4 Vergleich von Suchdienste

Alle Suchdienste stellen die gesammelten Daten aus dem Internet lokal in ihrem eigenen Datenbank.

Der Suchdienst besteht hauptsächlich aus den folgenden funktionellen Teilen:

Harvesting: Erfassung der Informationen Entweder aus allen Dokumenten, oder nur beschränkt auf die Home- oder Top-Seiten.

Da es mehrere Wochen dauern kann, die Dokumente wieder zu besuchen, sind die Datenbestände oft veraltet. Nur wenige Dienste benutzen Methoden, die bestimmte Teile der Datenbanken sehr häufig aktualisieren.

Indexierung und Retrieval: Die meisten Dienste versuchen, den Volltext der Originaldokumente zu indexieren. Metadaten werden nur von ganz wenigen der grösseren Suchdienste überhaupt indexiert und nicht korrekt zur Verbesserung des Retrievals genutzt.

Anwendungsoberfläche : Die Anwenderoberflächen sind immer mit HTML geschrieben unter Verwendung von HTTP Gateways zur Datenbank.

Eine Tendenz ist die, immer mehr, redaktionell aufgearbeitete Kataloge zusammen zu stellen. Diese sind wesentlich kleiner als robotergenerierte Datenbanken, lassen sich jedoch durch den Benutzer themenorientiert durchsuchen.

Zusätzlich bieten die meisten Dienste auch spezielle Datenbanken zu Usenet News, eMail-Adressen, etc. an.

Viel Werbung und eine Menge von Unterhaltungsangeboten stört die Anwendbarkeit für den seriösen Nutzer. Die Folge ist Unübersichtlichkeit und in den Hintergrund treten eigentlichen Nutzungsfälle.

Die Ursache liegt darin, dass die Neuentwicklung der Suchmöglichkeiten und Angeboten nicht mehr aus wissenschaftlich/experimentellen Gründen geschieht, sondern aus Konkurrenzgesichtspunkten auf dem kommerziellen Massenmarkt. Man optimiert ganz einfach die Bedingungen für den Anzeigenverkauf[20].

4.1 Allgemeine Suchdienste

In diesem Kapitel werden einige allgemeine Internet Suchdienste (roboterbasierte und katalog- verzeichnisbasierte) repräsentativ ausgewählt und beschrieben:

- **AltaVista**
  http://www.altavista.de
  http://www.altavista.com

Die Suchmaschine AltaVista wurde von der Firma Digital entwickelt. AltaVista gehört zu den bekanntesten Suchmaschinen. Neben der Web-Seiten werden auch Beiträge in Newsgroups indiziert.

Nach Angaben der Betreiber weist AltaVista auf über 30 Millionen Webseiten. Indizierung erfolgt im Volltext. Die AltaVista Software ist in C geschrieben. Das Programm besteht aus zwei Hauptkomponenten:

- *Scooter* ist eine sehr schnelle Roboter und besucht täglich ca. drei Millionen Web-Seiten
- *Indexer* kann täglich bis zu einem Gigabyte Daten verarbeiten. Die indizierte Datenbank ist ca. 40 Gigabyte groß und kann mehrere Anfragen gleichzeitig bedienen.

Die Suchbegriffe können mit den booleschen Operatoren AND, OR, AND NOT und NEAR verknüpft werden. Außerdem gibt es eine automatische Phrasensuche; häufig gesuchte Suchbegriffe, die aus mehreren Wörtern bestehen, werden automatisch zu Phrasen verknüpft.

AltaVista bietet auch die Suche in Feldern. Dazu stellen man den Feldnamen vor den Suchbegriff und trennt ihn mit einem Doppelpunkt. Damit wird bestimmt, welcher Teil eines Dokumentes bei der Suche berücksichtigt wird.
z.B.

Suchbegriff	Funktion
image: Name	Image findet den Dateinamen eines Bildes: gif oder jpg.

applet:class / Art	Applet findet Seiten mit einem speziellen Java Applet
host:name	Host findet Seiten eines bestimmten Servers
text:Text	Text findet alle Seiten, die den gesuchten Ausdruck benutzen

- **Aladin**
  http://www.aladin.de

Die Suchmaschine Aladin(Allgemeines Archiv deutschsprachiger Inhalte im Netz) indiziert den ganzen Inhalt deutschsprachiger Seiten im Volltext unter allen Domains (also auch .at, .ch,.com, .net, .org, usw.).

Nach Betreiberangaben durchsucht Aladin mit seinem Robot automatisch rund um die Uhr das gesamte Internet exklusiv nach neuen Inhalten in deutscher Sprache. Aladin kombiniert einen intern verwalteten Stichwortkatalog mit deutscher Spracherkennung. Alle 30 Sekunden besucht Robot eine neue deutschsprachige Seite im Internet. Dadurch werden monatlich 18.000 bis 27.000 weitere deutschsprachige Seiten im Volltext indiziert.

Suchbegriffe können mit UND und ODER verknüpft werden, wobei UND ist voreingestellt.

- **Lycos.de**
  http://www.lycos.com
  http://www.lycos.de

Die Suchmaschine Lycos (Lycosidae) steht als einer der ersten Suchdienst seit 1994 zur Verfügung. Lycos gehört zu der bekannteste Suchdienste. Laut eigenen Angaben hat Lycos mittlerweile einen Index mit ca. 51 Millionen Einträgen aufgebaut. Allerdings sind nicht alle Dokumente im Volltext indiziert, sondern teilweise nur deren Zusammenfassung.

Lycos unterscheidet bei der Such nicht zwischen Groß- und Kleinschreibung. Bei mehreren Suchbegriffen ist die ODER-Verknüpfung voreingestellt. Weitere booleschen Operatoren UND, NOT , EXOR werden unterstützt.

Auch ein paar Felder lassen sich bestimmen: neben der Suche im gesamten Dokument kann man nur innerhalb einer Domain, nur im Titel oder der URL suchen.

- **Crawler.de**
  http://www.crawler.de

Crawler.de ist eine Suchmaschine der Firma Schlund, basiert auf Software der Suchengine Glimpse der Universität von Arizona, die auch ein Teil der Harvest-System ist.

Bei der Suche werden boolesche Operatoren AND, OR, NEAR unterstützt

Interessant ist, das zu jedem Ergebnis eine Drop-Down-Liste mit allen Worten präsentiert wird, die verwandten Begriffe zu dem Suchwort enthalten. So kann auf einen Klick nach verwandten Begriffen gesucht werden. Neben jedem Wort ist die Zahl der Fundstellen aufgeführt.

- **Excite.de**
  http://www.excite.com
  http://www.excite.de

Die leistungsfähige Suchmaschine Excite wurde 1993 von sechs Studenten der Stanford Universität entwickelt. Der Betreiber von Excite, Excite INC, ist einer der führenden Internet-Suchdienstanbieter. Dieser Suchdienst bietet neben Web-Inhalten auch Usernet Newsgroups und Städteinformationen einschließlich Stadtplänen vieler Städte in aller Welt an.

Die Suchmaschine dieses Dienstes basiert auf ICE (Intelligent Concept Extraction) und arbeitet mit einer Datenbank, die nach Betreiberangaben um 60% größer als die der engsten Konkurrenten ist. Aktualisierung der Datenbasis erfolgt wöchentlich.

Excite bietet bei der Suche neben der Verknüpfungen mit AND, ODER und NOT Operatoren weitere Suchoptionen mit (+, -, ^()), mit denen das Suchwort weiter spezifiziert werden kann, und dadurch die Trefferquote erhöht wird, an.

Z.B. Excite erlaubt das Verknüpfen von Suchbegriffen mit einfachen +/- Zeichen. Alle Begriffe mit einem vorgesetzten + Zeichen müssen vorkommen. Das -Zeichen vor einem Suchwort kennzeichnet, daß dieses Wort in keinem der gefundenen Dokumente enthalten sein darf.

- **Fireball.de**
  http://www.fireball.de (früher: Flipper, Kitty)

Fireball ist eine Weiterentwicklung der Suchmaschine Flipper. Zwischenzeitlich hieß die Maschine auch Kitty, nach der Projektgruppe KIT (Künstliche Intelligenz und Textverstehen) an der TU Berlin, die das Projekt gestartet hatte.

Fireball hat sich auf deutsche Seiten spezialisiert und durchsucht auch deutschsprachige Seiten auf ausländischen Servern, die nicht auf einer .de - Domain liegen. Eine spezielle Software zur Sprachanalyse extrahiert die entsprechenden Dokumente aus dem Internet. Die Datenbasis nach eigenen Angaben besteht aus ca. 8 Millionen Einträgen und wird wöchentlich aktualisiert.

Fireball erlaubt das Verknüpfen von Suchbegriffen mit einfachen +/- Zeichen. Alle Begriffe mit einem vorgesetzten + Zeichen müssen vorkommen. Boolesche Operatoren AND, OR und NEAR und Wildcards werden unterstützt

Beim NEAR Verknüpfung dürfen im gesuchten Dokument maximal 10 Worte voneinander entfernt stehen.

- **Intersearch.de**
  http://www.intersearch.de

Intersearch ist eine Suchmaschine für den deutschsprachigen Bereich. Die Datenbasis umfaßt nach eigenen Angaben ca. 3 Millionen Seiten aus Deutschland und Österreich.

Bei der Suche dienen +/- Zeichen zur Spezifizirung. Die Suche nach bis zu 10 nebeneinander liegenden Begriffen kann mit dem Operator NEAR erfolgen. Zusätzlich kann die Suche auf die Felder Titel, Meta-Tag Inhalt (description) und Schlüsselworte (keywords), URL, und Email eingegrenzt werden.

Die Operatoren wie AND und ODER werden selbstverständlich unterstützt

- **Web.de**
  http://www.web.de

Web.de ist ein manuell bearbeiteter deutschsprachiger Katalog mit derzeit laut eigenen Angaben über 180.000 Einträgen in über 7000 Kategorien. Die Datenbasis wird täglich aktualisiert. Laut Betreiberangaben wird der Datenbestand regelmäßig kontrolliert und überarbeitet.

Bei der Standardsuche wird zwischen Groß- und Kleinschreibung nicht unterschieden. Mehrere Suchbegriffe werden erst mit UND-Verknüpfung, dann mit ODER-Verknüpfung durchsucht.

Die erweiterte Suche bietet Verknüpfungsmöglichkeiten mit UND und ODER. Die Suchbegriffe werden automatisch trunkiert. Somit werden auch Wortbestandteile gefunden. Auch die Phrasensuche ist mit den üblichen Anführungszeichen möglich.

- **Allesklar Katalog**
  http://www.allesklar.de

Allesklar ist ein deutschsprachiger Katalog, der in zwei Versionen zur Verfügung steht. Die eine basiert vollständig auf der Programmiersprache Java und setzt ein Java-fähige Browser voraus. Zudem müssen beim ersten Besuch der Seite Java-Classes (70 kB) vom Server geladen werden. Dafür müssen später bei der Übertragung von Suchergebnissen vom Server am Client kleinere Datenmengen übertragen werden.

Laut Betreiberangaben umfaßt allesklar.de über 200.000 manuell verzeichnete Seiten in mehr als 10.000 Kategorien. Die Kategorien entfalten sich in Listen, die über vier Ebenen komplett sichtbar bleiben und so ein Zurückblättern in der gesamten Hierarchie erleichtern.

Suchbegriffe können mit UND und ODER verknüpft werden. Interessant ist die Möglichkeit geographischer Suche nach Orten und Postleitzahlen. So kann mit jedem Suchwort ein Ort oder eine Postleitzahl verknüpft werden. Das funktioniert für deutsche, österreichische, und schweizer Seiten.

- **Dino**
  http://www.dino-online.de

Dino (Deutsch InterNet Organisationssystem) wird von der Firma AIS (Axon Internet Service GmbH) aus Göttingen betrieben. Seiner manuell erstellter Katalog umfaßt deutschsprachige Seiten und hat einen qualitativ guten und vollständig redaktionell bearbeiteten Datenbestand. Laut Betreiberangaben werden 200.000 Links zu deutschsprachigen Web-Seiten auf ca. 10.000 Seiten verwaltet. Aktualisierung der Datenbasis erfolgt permanent.
Neben dem eigentlichen Katalog gibt es ein Branchenbuch, den Dino Surf-Tip, eine ausführliche Seite mit einer Sammlung internationaler TOP-Links, die Worlds Best Sites, einen Reuters Nachrichtenbereich und einige redaktionelle Themen.

Bei mehreren Suchbegriffen wird standardmäßig die UND-Verknüpfung eingesetzt. Außerdem werden die OR und NOT Operatoren und Wildcards unterstützt.

Die Regionalsuche ermöglicht das Auffinden von Seiten mit lokalem Bezug.

4.2 lokale Suchdienste[21]

Neben einer Vielzahl von globalen Suchdiensten entstehen immer mehr lokale Suchmaschinen. Die lokalen Suchmaschinen sind meistens gut gepflegt und aktuell.
Sie erlauben einen bestimmten, begrenzten Suchraum, dafür aber gezielt und qualitativ höhere Ergebnisse. Der lokale Anbieter kann seine Datenbankaktualisierung individuell und nach eigenem Bedürfnis gestalten. Dies erlaubt den Zugang – im Gegensatz zu globalen Suchdiensten - zu aktuellen Informationen. Auf dem Markt gibt es eine Reihe von Software zur Einrichtung lokaler Suchdiensten. Die meiste davon sind als zwar als Free-Ware erhältlich, jedoch erfordern sie einen gewisse Aufwand bei der Einrichtung.

Seit einigen Jahren gibt es „lokalen“ Suchmaschinen für die verschiedenen UNIX-Derivate. Diese sind in der Regel in der Programmiersprache C und Perl geschrieben. Die Kernfunktionalität und die Algorithmen sind meist in C geschrieben und die Konfiguration, Parametrisierung und die Abfrageschnittstelle erfolgt in Perl (Vgl. Kapitel 5).

Funktionsweise der lokalen Suchmaschinen unterscheidet sich nicht von der globaler Suchmaschinen.

Weiterhin spielt die Arbeitsweise des Gatherers (Datensammler) eine wesentliche Rolle:

Dateisystembasierte Indizierung: der Gatherer bearbeitet ausgehend von einem Wurzelverzeichnis rekursiv alle Dateien. WWW-serverseitige Einstellungen, wie URL-Rewriting oder Zugangssperren, ignoriert er. Dafür müssen die Dateien nicht über den HTTP-Server angefordert werden, was den Indiziervorgang beschleunigt und Rechenleistung spart.

[21] Vgl. http://www.heise.de/ix/artikel/1999/02/089/

HTTP-basierte Indizierung: der Gatherer fordert alle zu indizierenden Dokumente vom WWW-Server an und erfaßt so auch nur die für den normalen Anwender zugänglichen Dokumente. Hierbei entstehen bei regelmäßiger Indizierung nicht unerhebliche Datenmengen, die eventuell in der Abrechnung des Providers mit auftauchen. Sollte dies nicht der Fall sein, lohnt sich eine HTTP-basierte Indizierung, zumal eine solche Suchmaschine nicht nur eigene WWW-Server indizieren kann.

Neben der Methode der Dateiübermittlung stellen die erkannten und konfigurierbaren Dateiformate ein weiteres Kriterium dar: während einfachere Produkte nur Text und HTML indizieren können, sind fortgeschrittenere Indizierer auch in der Lage, Informationen aus einer Vielzahl anderer Dateiformate zu verarbeiten und durch eine Suchfunktion zur Verfügung zu stellen.

Zu letzterer Kategorie gehören beispielsweise der Netscape Compass Server und das Harvest-System, die unter anderem Dateien, die im PDF-, PostScript- oder Word-Perfect-Format vorliegen, mit in den Index aufnehmen können.

Über die Brauchbarkeit einer Suchmaschine entscheidet in erster Linie die Bedienbarkeit des Brokers, was bei der Integration und Anpassung der mitgelieferten Suchformulare an das verwendete Design beginnt und bei einer verständlichen Ergebnissausgabe endet.

Die meiste Programme liefern eine einfache HTML-Seite mit und rufen ein eigenes CGI-Programm auf. Etwas aufwendiger ist meist die Anpassung der Ausgabe der Suchergebnisse. Bei einfacheren Programmen muß man dazu manchmal sogar an dem entsprechenden CGI-Skript selbst Hand anlegen.

Im Folgenden wird eine Übersicht einiger am Markt erhältlichen Lokalen Suchdienste gegeben:

WebGlimpse

WebGlimpse basiert im Kern auf dem Indizierer Glimpse, der für sich allein genommen lediglich Textdateien, die in einem lokalen Dateisystem vorliegen, indizieren kann und so eine schnelle Suche nach bestimmten Stichworten in diesen Dateien ermöglicht.

WebGlimpse versorgt Glimpse mit Textdateien, die auf einem HTTP-Server liegen, und bietet ein Web-Interface für Suchanfragen an. Es erweitert Glimpse vom Indizierer für lokale Dateisysteme zu einem für das Web.

ht://Dig

Diese Programmpaket zählt zu den HTTP-basierten Indizierern und arbeitet sich rekursiv durch einen oder mehrere WWW-Server. Zur Erhöhung der Trefferquote lassen sich bei ht://Dig eigene, zusätzliche META-Tags in die HTML-Seiten integrieren, die dann entsprechend in die Datenbank aufgenommen werden.
ht://Dig liegt als Quelltext vor und ist unter den Rahmenbedingungen der GNU Public License (GPL) frei verfügbar.

Webinator

Webinator von der Firma Thunderstone ist ein kommerzielles Produkt, von dem es eine kostenlose - im Funktionsumfang eingeschränkte - Version gibt, die für kleinere Webserver durchaus ausreichend sein kann. Webinator hat einige Funktionen, die man bei anderen Indizierern nicht findet. So verwaltet er beispielsweise den Index als SQL-Datenbank, so daß die indizierten Daten mit Hilfe von SQL-Kommandos nachträglich beabeitet werden können. Weiterhin besticht die Ausgabe der Suchergebnisse: Sie enthält neben einer grafischen Prozentanzeige für die Relevanz des gefundenen Dokumentes auch Informationen über dessen Dateigröße und einige weitere interessante Angaben. Dies ist ebenso wie das Interface für die Eingabe der Suchangabe frei konfigurierbar.

Harvest

Bei Harvest handelt es sich um ein komplexes, dafür aber auch leistungsfähiges System. Es benötigt eine langen Einarbeitungszeit und viele Konfigurationsdateien.
Siehe Kap. 5

Netscape Compass Server

Der Netscape Compass Server ist das Nachfolgeprodukt zum Netscape Catalog Server, der wiederum eine kommerzielle Weiterentwicklung des Harvest-Systems ist. Daher ist der Kern des Compass Server ähnlich aufgebaut ist wie dieses.

Aufbau des Netscape Compass Server: neben der Suchfunktion werden die Nutzer des Systems auch über neu entdeckte Informationen zu Themen, die sie interessieren, durch den My Compass Newsletter automatisch unterrichtet.

Von Harvest geerbt hat der Compass Server die Fähigkeit, mit sehr vielen Dateitypen umzugehen – es sind sogar noch einige hinzugekommen, unter anderem die Microsoft Office Dateiformate.

Zusätzlich zum automatisch generierten Suchindex kann man mit dem Compass Server auch einen baumartig aufgebauten Informationskatalog, ähnlich wie man ihn von Yahoo kennt, aufbauen, so daß ein weiterer Weg zum Finden von Informationen als Alternative zu der Eingabe eines Suchbegriffes entsteht. Zu den weiteren Besonderheiten des Netscape Compass Server zählt die Fähigkeit, allen Personen, die an einem bestimmten Thema interessiert sind, eine Nachricht zukommen zu lassen, wenn ein Gatherer ein neues Dokument hierzu gefunden hat. Diese Nachrichten werden entweder in einer täglichen Email, dem My Compass Newsletter, zusammengefaßt oder sind über eine personalisierte Web-Seite für den Nutzer zugreifbar.

Swish-E

Swish-E erlaubt die Indizierung sowohl auf Dateisystemebene als auch über das HTTP-Protokoll. Interessant an Swish-E dürfte vor allem die hohe Geschwindigkeit bei der Beantwortung von Suchanfragen und die relativ geringe Größe der erzeugten Indizes sein. Diese soll lediglich 1 bis 5 % des Umfangs der zugrundeliegenden HTML-Dokumente haben.

Excite for Web Servers

Kostenlos und ohne jeglichen Support bietet Excite, Betreiber der gleichnamigen Suchmaschine fürs Web, Excite for Web Servers an, verlangt aber eine detaillierte Registrierung. Es gibt Versionen für nahezu alle UNIX-Derivate.

Dank seines geringen Umfangs sowie der optisch ansprechenden und variablen Präsentation der Suchergebnisse bietet sich Excite für Projekte an, bei denen schnell ein funktionsfähiges Ergebnis präsentiert werden muß.
Ein langfristiger Einsatz ist nicht ratsam, da Excite nicht weiterentwickelt wird.

Verity Information Server

Der Verity Information Server ist ein kommerzielles Produkt, daß für mittlere und größere Unternehmen interessant sein soll. Dafür würden vor allem die große Anzahl von verarbeiteten Dateiformaten als auch die Möglichkeit, die indizierten Daten nach Themen zu ordnen und so eine Struktur in die im Unternehmensnetz vorhandenen Informationen zu bringen, sprechen. Es sollen sich bei der Ausgabe der Suchergebnisse direkt auch die zu einem Suchergebnis thematisch zugehörigen Dokumente aufspüren lassen.

5 Harvest – Implementierung beim BSZ

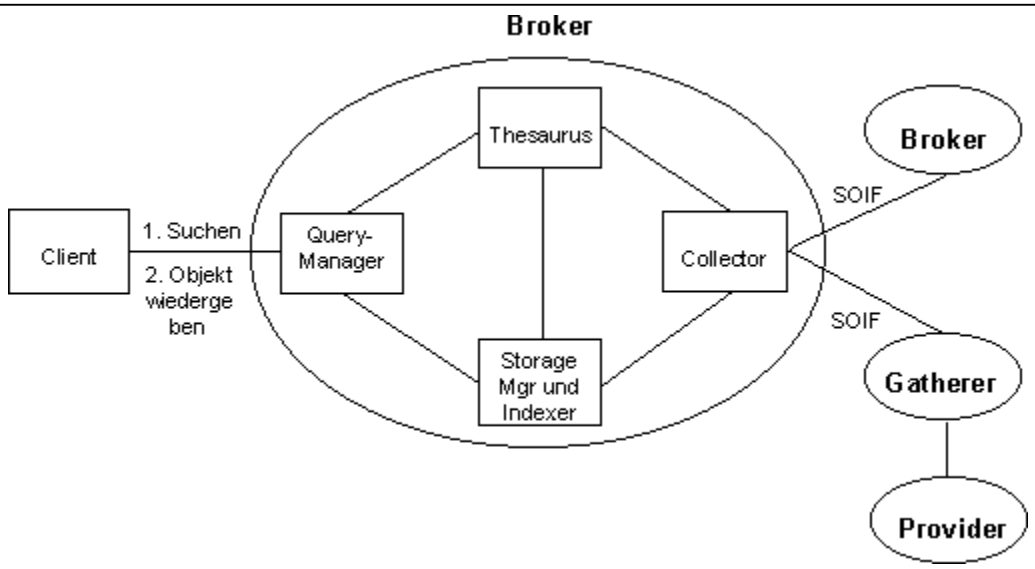

Abbildung 10:Harvest-Komponente (Vgl. Harvest Manual S. 2)

Harvest ist ein integriertes Softwaresystem zum Aufbau eines Internet-Suchdienstes. Das System besteht aus mehreren Komponenten (Siehe Abb. 10), die zum Sammeln, Extrahieren, Organisieren, Replizieren, Indizieren, Suchen und Cachen (Puffern) von verteilten Informationen im Internet dienen. Verschiedene Arten der Indizierungen sind möglich (Siehe Kap. 5.2).Durch die „verteilten“ Indizierung kann die Netzlast erheblich reduziert werden.[22]

Harvest ist ein frei verfügbares Softwaresystem, das von den Universitäten University of Colorado, University of Arizona und University of South California entwickelt wurde und das aus zahlreichen konfigurierbaren Tools besteht. Mit einem relativ geringen Aufwand an Installation und Grundkonfiguration steht bereits ein funktionsfähiges Suchsystem zur Verfügung. Die Anpassung an eigene Bedürfnisse erfordert jedoch umfangreiche Änderungen an diesem Standardsystem, das ein flexibles Grundgerüst darstellt.

[22] Harvest Users Manual seite 2

5.1 Harvest-Komponenten und deren Funktionsweise

Harvest setzt sich, wie in Abb. 10 dargestellt, aus mehreren unabhängig voneinander arbeitenden Komponenten zusammen.

1. Gatherer

Der Gatherer sammelt die zu indizierenden Informationen (wie Schlüsselwörter, Titel, Metadaten usw.) aus dem Internet, extrahiert und analysieren (z.B. Umsetzen in in einen SOIF-Record), und legt sie in eine Gatherer-Datenbank in Summary Object Interchange Format (SOIF) (Siehe Kap. 0) ab. Mit welchen Servern (HTTPs, FTPs usw.) die Verbindung hergestellt werden soll, von denen die Informationen bezogen werden, liest der Gatherer aus einem Konfigurationsdatei (Siehe Kap. 5.5.1), in dem manuell die Server-URLs eingetragen sind.

Der Gatherer besteht aus mehreren Programmen, die Teilaufgaben wie Sammeln, Extrahieren, Zusammenfassungen in SOIF- Format und den Broker zugänglich machen übernehmen. Das *Gatherer-Programm* selbst steuert dabei den gesamten Prozeß (Siehe Abb. 11):

- *Enumerator*

  Das *Enumerator-Programm* greift, mit einer Anzahl von Standard-Zugriffs-Methoden (FTP, Gopher, HTTP, NNTP[23], lokale Dateisystem) auf eine Reihe von Providern (URLs in der Konfigurationsdatei), zu und überträgt die Informationsquellen. Es ermittelt rekursiv den gesamten Inhalt vom Ziel-Server, indem er beispielweise Links auf einem WWW-Server rekursiv verfolgt. Welche Dateien zum Gatherer übertragen werden sollen, welche Links oder Verzeichnisse nicht weiter verfolgt werden sollen, kann man in einem Konfigurationsfile (für jeden zu besuchenden Server die Zugangsmethode, Suchbreite und) festlegen.

- *Essence*

  Das *Essence-Programm* extrahiert die gesammelten Objekte. Dabei erkennt er den Datentyp (html, ps usw.). Falls die Datei komprimiert ist (z.B. tar) wird sie dekomprimiert.
  Der Dateiname und der Dateityp dienen als Auswahl (durch Konfiguration kann man einige Dateinamen sowie Typen vom Indizierung ausschließen).
  Zum Schluß wird anhand des Datentyps das passende *Summarizer-Programm* aufgerufen.

- *Summarizer*

  Das vom *Essence-Programm*, aufgerufene *Summarizer-Programm* (z.B. HTML.sum, PostScript.sum) wertet den Objektinhalt aus und faßt, eine Liste von Attribut-Wert-Paare, im SOIF-Format zusammen.

- *Gatherd*

  Der Hintergrundprozeß *gatherd* stellt dann die Gatherer-Datenbank den Brokern zur Verfügung. Gatherd wird, sobald der Prozeß des Datensammelns abgeschlossen ist, gestartet und steht als Server für den Verbindungsaufbau der Brokern zur Verfügung.

- *Gather*

  Mit dem Programm G*ather* wird die Verbindung zum g*atherd-Server* aufgebaut, um die Daten aus der Gatherer-Datenbank abzurufen. Es kann von der Kommandozeile aus ausgeführt werden (z.B. zum Testen `$gather localhost 8500 | more`) und wird vom Broker benutzt.

[23] Network News Transfer Protokoll

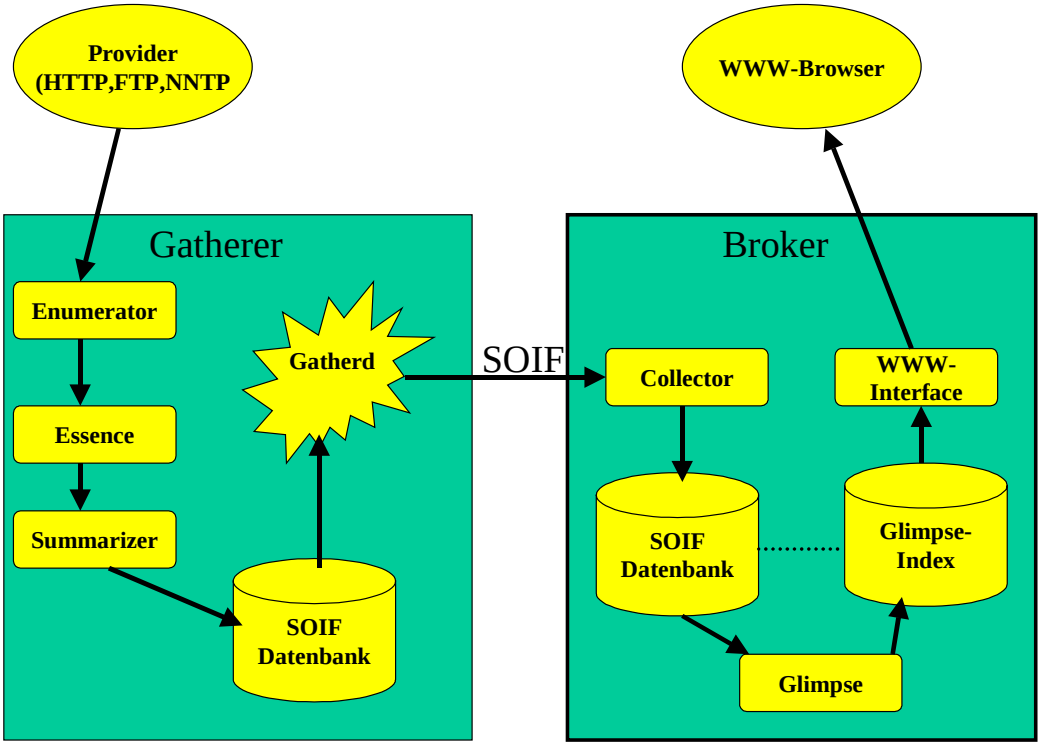

Abbildung 11: Funktionsweise von Gatherer & Broker

Damit stehen die Informationen, die zur Indizierung aufgearbeitet worden sind, den Harvest-Brokern oder anderen Suchdienste zur Verfügung, um sie in durchsuchbarer Form zu indizieren und über das WWW-Interface zugreifbar zu machen.

2. Broker

Der Broker holt die gesammelten und aufbereiteten Informationen (je nachdem wie das System eingerichtet bzw. konfiguriert ist) von einen oder mehreren Gatherer sowie von anderen Brokern, bereinigt sie (z.B. Löschen von mehrfach vorhandenen Informationen) und indexiert sie inkrementell. Darüber hinaus stellt er eine Schnittstelle (in Form eines HTML-Formulares mit CGI-Anbindung) zur Verfügung, mit der Suchanfragen durchführt werden können (Siehe Abb. 11).

Der Broker ist von seiner Architektur und Funktionsweise her ist wesentlich einfacher als der Gatherer und besteht aus einem *Verbindungstool* (Collector), einer *Indizierungsschnittstelle* mit SOIF-Datenbank (Glimpse) und ein *WWW-Interface* (HTML-Form):

- *Verbindungstool (Collector)*

Der Broker-Collector stellt die Verbindung zum gewählten Gatherer und anderen Broker her, holt die in SOIF-Format erstellten Dateien und fügt sie in die SOIF-Indexdatenbank inkrementell ein. Außerdem löscht er die doppelten SOIF-Einträge, die durchaus vorkommen können, wenn die Daten aus mehreren Gatherer und/oder Broker stammen.

- *Indizierungsschnittstelle Glimpse*

Der Harvest-Broker besitzt eine flexible Indizierungsschnittstelle, die verschiedene Search-Engines zur Indizierung und Anfrage an den Index einsetzen kann. Als Standard Seach-Engine wird Glimpse genutzt. Sobald die Daten vom Collector geholt worden sind, wird Glimpse gestartet, der seinerseits die SOIF-Datenbank indiziert und die Indexdaten in einem Indexdatei ablegt und verwaltet. Diese erstellte Indexdatei wird, wenn neue Daten vorlegen, aktualisiert.

Glimpse unterstützt bei der Formulierung von Suchanfragen folgende Eigenschaften[24]:

- case-sensitive und case-insensitive Anfragen
- Patternmatching: exakt, Teilwörter oder Ausdrücke mit mehreren Wörtern
- boolesche Kombinationen von Wörtern (mit AND/OR)
- approximatives Patternmatching, Schreibfehler können erlaubt sein
- strukturierte Anfragen, Treffer können an bestimmte Attribute gebunden werden
- Darstellung von einzelnen Zeilen oder ganzen Datensätzen, in denen ein Treffer vorkommt, beispielsweise Zitate

- Begrenzung der Trefferanzahl

- vereinfachte reguläre Ausdrücke (Wild Cards).

Die verschiedenen Optionen können über WWW Interface eingestellt werden.

- *WWW-Interface*

Das WWW-Interface bietet eine komfortable Möglichkeit Suchanfragen an das Harvest-System zu stellen. Anhand einer HTML-Formular werden die Benutzeranfragen über den HTTP-Server an den Broker gestellt, der seinerseit die Abfrage durch eine CGI-Programm (Query-Manager) bearbeitet und deren Ergebnisse zurückliefert.

3. Replicator

Der Replicator ist ein Tool, mit dessen Hilfe Broker repliziert werden können. Er gehört nicht zu der Harvest-Standarddistribution. Er wird bei der Bildung von Harvest-Netzen eingesetzt. Ein Replicator kopiert die gesamten SOIF-Datenbank eines Brokers auf andere Rechner im Internet, die dann somit ein Replikatoren-Netz bilden. Die Bearbeitung der Suchanfragen wird beschleunigt, da die Bearbeitung von Suchanfragen an den Broker durch den Replicator auf mehreren Rechner verteilt wird.

„Der Replicator verwaltet einen Verzeichnisbaum aus Dateien. Ein Server muß als Master Copy ausgezeichnet sein. Updates davon werden auf alle anderen Repliken verbreitet und die Master Copy überschreibt jede lokale Änderung, die an individuellen Repliken gemacht wurde. Es ist möglich, eine replizierte Kollektion so zu konfigurieren, daß eine andere Master Copy getrennte Unterbäume verwaltet, um die Verantwortung zu verteilen. Jede replizierte Kollektion wird durch eine einzelne oder hierarchisch geschachtelte Replication Group exportiert, für die es eine Access Control List gibt. Wenn eine Replik zu einer

[24] http://www.uni-stuttgart.de/rus/Bi/1996/4/File4.html

Replication Group hinzugefügt wird, beginnt sie sich mit Daten zu füllen. Wächst eine Replication Group auf hunderte oder Tausende von Repliken an, kann eine neue Gruppe erzeugt werden, um die Verwaltung zu erleichtern[25]*“.*

4. Cache

Mit dem Einrichten eines Object-Cache-Systems kann der Zugriff auf häufig angefragte Informationen optimiert werden. Er wird mittlerweile separat distribuiert und ist nicht zwingender Bestandteil von Harvest. Er kann ohne Harvest-Installation, z.B. bei der Einrichtung eines Proxy-Servers für WWW-Server eingesetzt werden.

„Der Object Cache erlaubt Benutzern, FTP, Gopher und HTTP Daten schnell und effizient zu holen, ohne das Internet zu belasten. Der Harvest Cache ist mehr als eine Größenordnung schneller als der CERN Cache und andere bekannte Internet Caching-Systeme. Dies wird dadurch erreicht, daß für WWW- und Gopher-Zugriffe nicht geforkt wird, die Ein-/Ausgabe sowie DNS Lookups nichtblockierend erfolgen, Meta-Information, oft gefragte Objekte und DNS Lookups im RAM gecached werden.
Der Cache kann in zwei verschiedenen Modi betrieben werden:

1.als Proxy Object Cache und

2.als httpd Accelerator[17]*“.*

5. Harvest-Server Registry (HSR)

HSR ist ein Broker, der an der University of Colorado in Boulder läuft und eine Datenbasis über alle im Internet installierten Broker, Gatherer, Replikatoren und Object-Cache-Systeme verwaltet.

[25] http://www.uni-stuttgart.de/rus/Bi/1996/4/File4.html

Beim BSZ wurden nur die zentralen Komponenten des Harvest-Systems (der Gatherer und der Broker) eingesetzt. Die detaillierten Informationen über den Einsatz vom Replicator- und Cache-Komponenten sind in Harvest-Manual (Seite 54ff) zu finden.

Summary Object Interchange Format (SOIF)

SOIF dient als harvesteigenes Protokoll zur Kommunikation zwischen dem Gatherer und dem Broker. Wie oben erwähnt erzeugen der Gatherer aus den gesammelten Informationen Inhaltszusammenfassungen in SOIF-Format. Diese SOIF-Objekte werden von Broker zur Indizierung abgeholt.

SOIF bietet verschiedene Möglichkeiten Objektsammlungen zu klammern, damit der Harvest Broker SOIF-Inhaltszusammenfassungen von einem Gatherer für viele Objekte in einem einzigen effizient komprimierten Stream holen können. Mit dem Befehl `$gather localhost 8500` können alle Objekte von Brokern abgeholt werden. Der Broker stellt dann diese SOIF-Objeckte mit strukturierten Attribut-Wert-Paaren für die verschieden Anfragetypen zur Verfügung.
Die Harvest-Standarddistribution sieht folgenden Attribute vor (Harvest-Manual, Seite 77):

ATTRIBUTE	DESCRIPTION
Abstract	Brief abstract about the object.
Author	Author(s) of the object.
Description	Brief description about the object.
File-Size	Number of bytes in the object.
Full-Text	Entire contents of the object.
Gatherer-Host	Host on which the Gatherer ran to extract information from the object.
Gatherer-Name	Name of the Gatherer that extracted information from the object. (eg. Full-Text, Selected-Text, or Terse).
Gatherer-Port	Port number on the Gatherer-Host that serves the Gatherer's information.
Gatherer-Version	Version number of the Gatherer.

Update-Time	The time that Gatherer updated the content summary for the object.
Keywords	Searchable keywords extracted from the object.
Last-Modification-Time	The time that the object was last modified.
MD5	MD5 16-byte checksum of the object.
Refresh-Rate	The number of seconds after Update-Time when the summary object is to be regenerated. Defaults to 1 month.
Time-to-Live	The number of seconds after Update-Time when the summary object is no longer valid. Defaults to 6 months.
Title	Title of the object.
Type	The object's type. Some example types are: Archive, Audio, Awk, Backup, Binary, C, Cheader, Command, Compressed, CompressedTar, Configuration, Data, Directory, DotFile, Dvi, FAQ, FYI, Font, FormattedText, GDBM, GNUCompressed, GNUCompressedTar, HTML, Image, Internet-Draft, MacCompressed, Mail, Makefile, ManPage, Object, OtherCode, PCCompressed, Patch, Perl, PostScript, RCS, README, RFC, SCCS, ShellArchive, Tar, Tcl, Tex, Text, Troff, Uuencoded, and WaisSource
Update-Time	The time that the summary object was last updated. REQUIRED field, no default.
URL-References	Any URL references present within HTML objects.

„Es ist für einen Harvest-Server-Betreiber möglich, weitere (oder weniger) Attribute zu definieren und nach Filetypen zu variieren. Grundsätzlich gilt natürlich, ***daß nicht vorhandene Information auch nicht erzeugt werden kann.*** *So kann etwa aus einem PostScript-File das Attribut „author" nicht gewonnen werden kann. Auf die Auswahl der Keywords besteht Einflußmöglichkeit"*[26].

[26] Vgl. Harvesting Mathematics, Judith Plümer, Roland Schwänzl, Fachbereich Mathematik Universität Osnabrück

Beispielsweise werden die HTML-Tags, bei der Erzeugung von SOIF-Objekte, durch HTML-Summarizer (HTML.sum) nach folgenden Schema in SOIF Attribute übersetzt (Vgl. Harvest-Manual Seite 21):

HTML ELEMENT	SOIF ATTRIBUTES
<A>	keywords,parent
<A:HREF>	url-references
<ADDRESS>	address
<B>	keywords,parent
<BODY>	body
<CITE>	references
<CODE>	ignore
<EM>	keywords,parent
<H1>	headings
<H2>	headings
<H3>	headings
<H4>	headings
<H5>	headings
<H6>	headings
<HEAD>	head
<I>	keywords,parent
<IMG:SRC>	images
<META:CONTENT>	$NAME
<STRONG>	keywords,parent
<TITLE>	title
<TT>	keywords,parent
<UL>	keywords,parent

Nach obigen Schema wird z.B. folgender HTML-Tag,

```
<TITLE>BSZ Baden-Wuerttemberg - Homepage des BSZ</TITLE>

<A HREF="http://www.bsz-bw.de">Home</A>
```

folgendermaßen in das SOIF-Format übersetzt:

```
title{13}: BSZ Baden-Wuerttemberg - Homepage des BSZ

url-references{32}: http://www.bsz-bw.de
```

Die von Gatherer erzeugten SOIF-Objekte haben folgende Schema:

```
@FILE { url1
        Attribute-Name-1:        DATA
        Attribute-Name-2:        DATA
        ...
        Attribute-Name-n:        DATA
        }
```

Weitere Beispiele: Die Homepage von BSZ mit folgendem Quelleausschnitt

```
<HTML>
<HEAD>
...
<META NAME="DC.creator.name"      CONTENT="Heymans, Wolfgang">
<META NAME="DC.subject"
   CONTENT="(SCHEME=SWD)       Bibliotheksservice-Zentrum       Baden-
W&uuml;rttemberg, S&uuml;dwestdeutscher Bibliotheksverbund, Homepage">
<META NAME="DC.description"
   CONTENT="Homepage     des     Bibliotheksservice-Zentrum     Baden-
W&uuml;rttemberg (BSZ),
...
</HEAD>
...
```

wird folgendermaßen in einem SOIF-Objekt abgelegt(Siehe Anhang C):

```
SOIF Object for: http://www.bsz-bw.de/index.html

@FILE { http://www.bsz-bw.de/index.html
update-time{9}: 949389971
last-modification-time{9}:      949389971
time-to-live{7}:        2419200
refresh-rate{6}:        604800
gatherer-name{23}:      SWIB-Suchdienst vom BSZ
gatherer-host{17}:      thurgau.bsz-bw.de
gatherer-version{3}:    1.5
type{4}:        HTML
file-size{4}:   7349
md5{32}:        dc0bc33ead20315f01fb24d5eaac15fc
...
dc.publisher{45}:       Bibliotheksservice-Zentrum Baden-Wuerttemberg
dc.creator.name{17}:    Heymans, Wolfgang
dc.date.current{33}:    (SCHEME=ANSI.X3.30-1985) 20000118
dc.description{128}:    Homepage des Bibliotheksservice-Zentrum Baden-
                        Wuerttemberg (BSZ),
                        Suedwestdeutscher Bibliotheksverbund (SWB) und
                        Zentralkatalog
...
}
```

5.2 Harvest als verteilte Suchmaschine

Durch die Verteilung der zwei wichtigsten Komponenten von Harvest im Netz (Gatherer und Broker), wird die Realisierung eines verteilten Suchdienstes möglich.

Gatherer und Broker können auf unterschiedliche Weise konfiguriert werden (Siehe Abb. 12). Bei der Indizierung von lokalen Daten können auf einem Rechner mehrere Gatherer einen oder mehrere Broker bedienen. Gatherer, die verteilt im Netz betrieben werden, können einen oder mehrere Broker, die auf der indizierenden Maschine laufen, bedienen (Siehe Abb. 12, rechte Konfiguration). Harvest kann natürlich auch Server indizieren, auf denen keine Gatherer laufen. Hierbei holt ein lokal installierter und eingerichteter Gatherer mit Standard-Zugriffs-Protokollen, wie FTP, Gopher, HTTP oder NNTP, die Daten remote vom Provider (Siehe Abb. 12, links). Die stärkeren Linien kennzeichnen die höhere Server- und Netzbelastung, die dafür in Kauf genommen werden muß. Einen Gatherer lokal zu betreiben, ist zwar wesentlich effizienter (Siehe Abb. 12, rechts), jedoch ist ein nicht lokaler Gatherer besser als viele Server, die unabhängig

voneinander Index-Informationen sammeln, weil viele Broker oder andere Suchdienste den verteilt erzeugten Index mitbenutzen können.

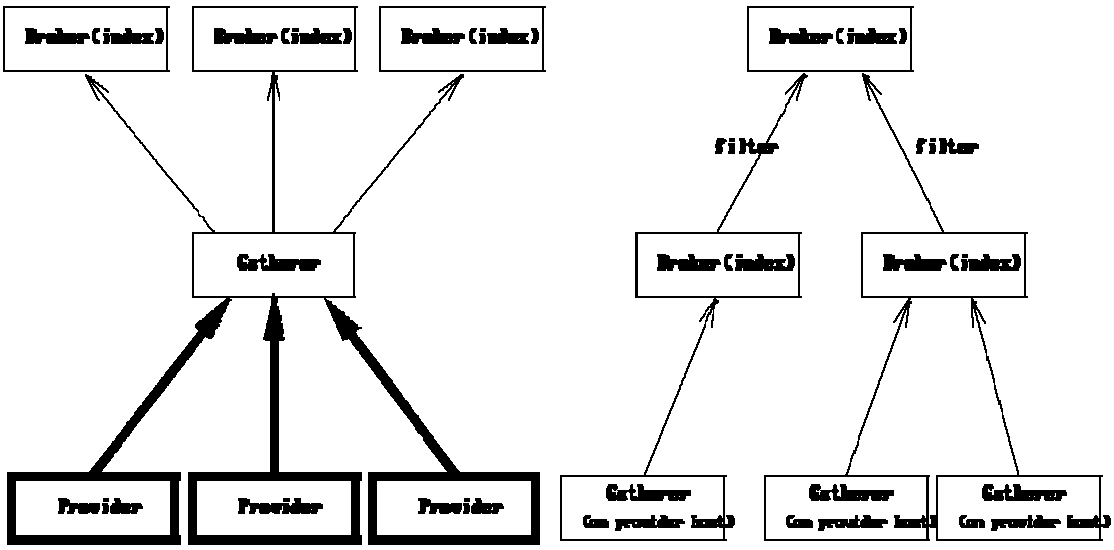

Abbildung 12: Konfigurationsmöglichkeiten des Harvest-Systems (Vgl. Harvest-Manual S. 3)

Ein Broker kann so konfiguriert werden, daß er nicht nur Gatherer abfragt, sondern auch von anderen Brokern Daten beziehen kann, um einen Index aus verteilten Informationressourcen zu erstellen. Die Netzbelastung wird dadurch maßgeblich reduziert, da der Indizierungprozeß verteilt abläuft. Das Query-Interface, über das der Broker die Information holt, läßt das Filtern oder Verfeinern der Information zu - und damit eine Erhöhung des Informationswertes durch Einarbeitung von Kontextinformation -. [27]

5.3 Installation

5.3.1 Hardware- / Plattformanforderungen

Ein typischer Harvest-Server setzt, um optimale Leistung zu erbringen, folgende Hardware-Ausstattung voraus:

[27] http://www.uni-stuttgart.de/rus/Bi/1996/4/File4.html#HDR0

- Einen schnellen Prozessor (z.B. **Sun Sparc 5**, **DEC-Alpha**, **Intel-Pentium**),
- 1-2 GB freien Speicherplatz auf der Festplatte und
- 64 MB RAM.

Eine langsamere CPU verlangsamt den Harvest-Server. Wichtiger als die CPU-Geschwindigkeit ist jedoch die Speichergröße. Bei Harvest-Servern laufen einige Prozesse ständig im Hintergrund ab (z.B. Glimpse-Serverprozeß, DNS-Serverprozesse). Wenn nicht genügend Hauptspeicher vorhanden ist, verringert die Verlagerung (Paging) auf die Festplatte drastisch die Leistung[28].

Der Arbeitsspeicher spielt auch eine wichtige Rolle bei der Abarbeitung von Benutzerabfragen an den Broker.

Harvest läuft prinzipiell auf jedem UNIX-Rechner. Folgende Plattformen werden explizit unterstützt:

- DEC: OSF/1 2.0 und 3.0
- SUN: SunOS 4.1.x und Solaris 2.3

Für einige andere Systeme (HP-UX, AIX, Linux usw.) stehen speziell abgestimmte Sourcecodes zur Verfügung.

Hardware-Ausstattung beim BSZ:

- UNIX-Maschine: Sun E 4000
- Prozessor : 3 Ultra-Sparc II 336 MHz
- Festplatten : 1 9GB Systemplatte,10 9GB als raid 5 in einem A5000-Photon

[28] Harvest Users Manual

- RAM : 2560 MB

➢ Betriebssystem Solaris 2.6

5.3.2 Software

Zu den oben genannten Plattformanforderungen müssen folgende Programme vorhanden sein:

- Perl, ab Version 4.0 (Perl 5.0 wird empfohlen)
- GNU Zip ab Version 1.2.4
- ein HTTP Server

Falls Harvest als Quellcode-Version und nicht als binäre (also vorkompilierte) Version installiert werden soll, braucht man zusätzlich noch:

- GNU-gcc ab Version 2.5.8
- Flex v2.4.7 und Bison v1.22

Software beim BSZ

Auf dem Rechner ist

- Perl 5.0
- GNU-Zip und
- WWW-Server Apache 1.3.11

vorhanden.

5.3.3 Installationsablauf

Es gibt zwei Formen von Harvest Distributionen: Zum einen den reinen Sourcecode, zum anderen eine binäre Version. Beim BSZ wurde die Sourcecode-Version 1.5 eingesetzt.

Die Harvest Software wird auf zahlreichen FTP-Servern angeboten
(Siehe http://www.tardis.ed.ac.uk/harvest/download.html)

Sourcecode-Version

Bei der Installation der Sourcecode-Version, geht man folgendermaßen vor:

I. Die Distribution muß mit dem folgendem Befehl entpackt werden:

```
$gzip -dc harvest-src.tar.gz | tar xf -
```

Beim Entpacken wird im aktuellen Verzeichnis das Verzeichnis *harvest-1.5* angelegt, in dem u.a. folgende Installationsskripte und der Sourcecode stehen:

- Makefile
  Top-Level Makefile für die Harvest-Distribution (da die Komponenten auch einzeln installieren lassen, haben ihre entsprechende Make-File).
- SetupComponent
  In diesem Shell-Skript stehen für die Installationsroutine alle zu installierende Komponente und deren Pfade. Es ermöglicht Komponente hinzufügen oder herauszunehmen.

- Src
  In diesem Verzeichnis stehen alle Source-Code, Help-File und Installations und Standard-Konfigurationsdateien

- Components

In diesem Verzeichnis befinden sich die Komponenten, die zur Standard-Distribution gehören, u.a. Gatherer und Broker

Die Standard-Installation erfolgt unter */usr/local/harvest*.

Wenn ein anderes Verzeichnis gewünscht ist, dann müssen folgende Schritte durchgeführt werden ($ ist in diesem Fall UNIX-Eingabezeichen):

- `$vi Makefile`
  Standardeintrag: `prefix = /usr/local/harvest`
  in
  `prefix = [eigener PATH]`
  ändern.

- `$make reconfigure`
  (damit die Änderung in allen weiteren Verzeichnissen (z.B. src, components) wirksam wird).

II. Mit dem Befehl

- `$make install`

kann dann die Harvest-Installation gestartet werden.

Hinweis
Mit der Installation werden unter dem angegebenen Installationsverzeichnis (bei der BSZ-Installation /usr/local/harvest) u.a. folgende Files und Verzeichnisse angelegt.

- **RunHarvest**
  Mit diesem Shell-Skript wird die Standardinstallation gestartet
- **brokers**
  In diesem Verzeichnis werden die Broker stehen, die angelegt werden

- **gatherers**
  In diesem Verzeichnis stehen die Gatherer
- **bin**
  hier befinden sich die exe files(wie z.B. broker, CreateBroker, gather).
- **cgi-bin**
  CGI-Skripte, die über den HTTP-Server aufgerufen werden, (z.B. „nph-search-SWIB.cgi“, mit dem die Suchanfrage an das Harvest-Server gestellt werden).
- **lib**
  in diesem Verzeichnis werden alle Bibliotheksdateien für das Harvest-System abgelegt, (z.B. Perl-Module, Broker und Gatherer Bibliotheksdateien)

Damit sind alle Schritte, die zur Installation nötig sind, getan. Nun kann Harvest eingerichtet und konfiguriert werden.

5.4 Konfiguration des Harvest-Suchdienstes

5.4.1 HTTP-Server

Der WWW-Server muß so konfiguriert sein, daß

- das Verzeichnis „harvest“ auf *$HARVEST_HOME* zeigt und
- er auf die CGI Programme im Verzeichnis *$HARVEST_HOME/cgi-bin/* zugreifen und diese ausführen kann.

Hinweis
Alle CGI-Skripte gehen davon aus, daß Perl unter */usr/local/bin* liegt.

Für die Konfiguration des WWW-Servers (Apache 1.3.11) beim BSZ, wurden in die Konfigurationsdatei *httpd.conf* folgende Angaben über Harvest eingetragen:

```
#
#   Eintraege fuer Harvest
#

Alias /harvest/ /usr/local/harvest/
Alias /Harvest/ /usr/local/harvest/
ScriptAlias /harvest/cgi-bin/ /usr/local/harvest/cgi-bin/


<Directory /usr/local/harvest/>
    AllowOverride None
    Options Indexes FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>

<Directory /usr/local/harvest/cgi-bin/>
    AllowOverride None
    Options ExecCGI Indexes
    Order allow,deny
    Allow from all
</Directory>
```

Über die Alias-Anweisung (`Alias URL-Pfad Dateipfad`) wird ein Verzeichnis oder eine Datei einem URL zugewiesen.

Die Anweisungen

```
Alias /harvest/ /usr/local/harvest/
Alias /Harvest/ /usr/local/harvest/
```

bewirken, daß bei einem Zugriff auf den URL http://www.bsz-bw.de/harvest[Harvest]/ greift der WWW-Server Apache intern auf das Verzeichnis */usr/local/harvest/* zu.

Mit `<Directory ...> und </Directory>` werden Konfigurationsanweisungen eingeschlossen, die für die /harvest und /harvest/cgi-bin Verzeichnisse gelten sollen.

5.4.2 Run Harvest

Nun soll Harvest gestartet bzw. eingerichtet werden. Dies geschieht am einfachsten durch den Aufruf von RunHarvest. RunHarvest startet das shell-skript *harvest* im Verzeichniss *$HARVEST_HOME /bin.* Dieses Skript stellt dem Anwender eine Reihe von Fragen, um festzustellen, in welcher Umgebung Harvest laufen soll und in welchen Verzeichnissen der Gatherer und der Broker generiert werden sollen. Nachdem alle

notwendigen Angaben gemacht wurden, werden u.a. der Gatherer und der Broker angelegt und gestartet. (Siehe Anhang B).

Das Programm benötigt folgende Angeben:

I. Angaben über den lokalen WWW-Server

- Auf welchem Host läuft der WWW-Server? [www.bsz-bw.de]
- Auf welchem Port läuft der WWW-Server? [80]

II. Auswahl einer Configuration zur Indexierung

- Was soll indiziert werden? [Auswahl aus 3 Möglichkeiten:
  1. Index your entire WWW site.
  2. Index an entire WWW site (or sites).
  3. Index selected parts of WWW, FTP, or Gopher sites.]

III. Konfiguration des Harvest-Servers

- Der Name für des Harvest-Servers, der eingerichtet wird. [none]
- Verzeichnis, in dem der Gatherer installiert werden soll. [/usr/local/harvest/gatherers/HTS]:
- Auf welchem Port soll der Gatherer laufen? [8500]
  meistens ist es im Bereich 8500-9500
- Verzeichnis, in dem der Broker installiert werden soll.
  [/usr/local/harvest/brokers/HTS]
- Auf welchem Port soll der Broker laufen? [8501]
  Der Broker-Port ist standardmäßig auf "Gatherer-Port +1" eingestellt.
- Zukünftiges Broker-Passwort (für die Wartung per Browser)
- Eingabe der URL-Liste (Siehe Kap. 5.1) auf deren Grundlage der Gatherer die Daten einsammeln soll.

IV. Starten des Harvest-Servers

- Jetzt wird der Harvest-Server nach den oben gemachten Angaben eingerichtet und der Indizierungsvorgang wird gestartet (der Gatherer, um die Informationen zu sammeln und anschließend der Broker, um diese zu indizieren).

Nach Durchführung dieser vier Schritte verfügt man über ein lauffähiges Harvest-System, das jetzt noch individuell angepaßt werden kann.

Bemerkung
Beim Einrichten des Harvest-Servers wird ein Broker und ein Gatherer automatisch erzeugt. Es ist möglich, das System nachträglich zu erweitern, z.B. um einen weiteren Broker zu erstellen (hierfür steht der Befehle *$HARVEST_HOME/bin/CreateBroker* zur Verfügung). Die Komponenten können auch einzeln aufgebaut werden, z.B. kann man den Gatherer folgendermaßen einrichten:

```
$cd src
$vi Makefile (falls man das Prefix ändern will) (Siehe Kap. 5.3.3)
$make reconfigure (falls man das Prefix geändert hat)
$make gatherer install-gatherer (Gatherer wird eingerichtet)
```

Weitere detaillierte Informationen finden Sie im Harvest-Manual (http://swing.informatik.uni-rostock.de/doc/user-manual/node6.html)

5.5 Anpassungen beim BSZ

Es wurden folgende Anpassungen durchgeführt:

- Gatherer
  - Einstellungen in der Config-Datei, um auch numerischen Werte zu indizieren
  - Ersetzen der Datei „HTML.sum“, um mangelhafte Indizierung der Standard-Distribution zu beheben

- Anpassen an deutsche Umlaute

- Broker
  - Die HTML-Abfrage-Seite (query) wurde um die spezielle Felder erweitert
  - Entsprechende Änderungen wurden im CGI-Skript gemacht
  - Die Ergebnis-Ausgabe wurde zuerst auf 25 beschränken

- Administration
  - Skripte zur Automatisierung

5.5.1 Gatherer Anpassungen und dessen Konfiguration

- **SWIB.cf**

Standardmäßig wird (mit RunHarvest) als Konfigurationsdatei für einen Gatherer die Datei

/usr/local/harvest/gatherers/<gatherername>/<gatherername.cf>

angelegt.

Die Konfigurationsdatei für den SWIB-Gatherer

/usr/local/harvest/gatherers/SWIB/SWIB.cf

hat folgendes Aussehen:

```
#
#  SWIB.cf - configuration file for the
#  SWIB-Suchdienst vom BSZ Gatherer
#
#  Created by  on Tue Nov 16 08:56:36 MET 1999
#
Gatherer-Name:    SWIB-Suchdienst vom BSZ
Gatherer-Port:    8500
#Time-To-Live:    2419200
Top-Directory:    /volume/www/local/harvest/gatherers/SWIB

<RootNodes>
# Enter URLs for RootNodes here
http://www.bsz-bw.de/depot/dokersch/lilis.html URL=3000 Depth=1
</RootNodes>

<LeafNodes>
# Enter URLs for LeafNodes here
</LeafNodes>
```

- **RootNodes**

Im Abschnitt **<RootNodes>** werden alle Server eingetragen, die vom Gatherer besucht werden sollen, wobei das Robot Exclusion Protocol[29] beachtet wird. Mit diesem Protokoll kann der Webmaster der besuchten Server selbst steuern, auf welche Bereiche der Gatherer zugreifen darf. Als Voreinstellung werden von jedem WWW-Server maximal 250 (URL-Max) Seiten geholt, und dies in einem Abstand (Delay) von 1 Sekunde. In den SWIB-Konfiguration wurde die maximale Seitenzahl auf 3000 erhöht, und die rekursive Zugriffstiefe (Depth) wurde auf 1 eingestellt. Eine Standard Delay-Einstellung von 1 Sekunde kann u.U. notwendig sein, da ein gut ausgestatteter WWW-Server leicht mehrere Tausend Seiten beinhalten kann und der Sekundenabstand Server und Leitung recht stark belasten würde (ein Erhöhung des Abstandes, z.B. auf 60 (delay=60) wäre dann sinnvoll)

In Harvest-Manual (Seite 3ff) sind weitere RootNode Spezifikationen angegeben.

- **LeafNodes**

Im Abschnitt **<LeafNodes>** werden URLs eingetragen, die vom Gatherer einfach abgeholt werden. Hierbei erfolgt keine rekursive Verfolgung der Links. Dateien , die keine

Links enthalten, also komprimierte Dateien (wie z.B. tar-files) und Postscriptdateien, würde man deswegen sinnvollerweise unter „LeafNodes“ eintragen.

- **Time-To-Live**

Beim Lauf des SWIB-Gatherers werden alle eingesammelten SOIF-Objekte in die Gatherer-Datenbank, die im Verzeichnis

/usr/local/harvest/gatherers/SWIB/data/...

liegt, eingetragen.

Diese Objekte haben eine Lebenszeit (Time-To-Live), die Standardmäßig auf 4 Wochen (2419200 Sekunde) eingestellt ist und individuell erhöht bzw. gemindert werden kann. Harvest überprüft regelmäßig, ob Objekte ihre Lebenszeit überschritten haben und löscht diese aus der Datenbank.

- **HTML-Summarizer (HTML.sum)**

Nach dem Start des SWIB-Suchdienstes wurde festgestellt, daß einige HTML-Seiten, deren Links unter RootNode eingetragen waren, nicht indiziert wurden. Nach Anfragen bei einigen Harvest-Betreibern habe ich festgestellt, daß das Problem an dem Standard-HTML-Summarizer (*/usr/local/harvest/lib/gatherer/HTML.sum*) liegt. Dieser wurde durch einen neuen entwickelten Summarizer[30] ersetzt und damit war das Problem behoben.

- **Anpassen an deutsche Umlaute**

Das Problem mit deutschen Umlauten wird im HTML Summarizer mittels Anhängen eines e an das Grundvokal gelöst, z.B. wird der ä durch ae ersetzt:

[29] http://info.webcrawler.com/mak/projects/robots/exclusion.html
[30] HTML.sum ist von Vincent Winczewski, Konrad-Zeus-Zentrum für Informationstechnik, Berlin

```
....
s/\334/Ue/g;        # LATIN CAPITAL LETTER U WITH WITH DIAERESIS
s/\\334/Ue/g;
s/%DC/Ue/g;
s/&#220;/Ue/g;

s/\337/ss/g;        # LATIN SMALL LETTER SHARP S
s/\\337/ss/g;
s/%DF/ss/g;
s/&#223;/ss/g;

s/\340/a/g; # LATIN SMALL LETTER A WITH GRAVE ACCENT
s/\\340/a/g;
s/%E0/a/g;
s/&#224;/a/g;
...
s/\344/ae/g;        # LATIN SMALL LETTER A WITH DIAERESIS
s/\\344/ae/g;
s/%E4/ae/g;
s/&#228;/ae/g;
...
```

5.5.2 Broker Anpassungen / Erweiterungen und dessen Konfiguration

Der Broker bezieht in regelmäßigen Abständen Daten von Gatherern. In welchen Abständen dies geschieht wird neben anderen Dingen (Broker-Hompage (WWW-Interface), Admin-Password, Web-Server, Web-Path, Broker-Port) in der Datei:

/usr/local/harvest/brokers/SWIB/admin/broker.conf

eingestellt.

Der aktuelle Eintrag ist wie folgt angegeben:

```
Collection-Rate 86400
```

d.h. alle 24 Stunden werden die Daten vom Gatherer abgeholt und damit die Broker SOIF-Files aktualisiert (86400 Sek = 24Uhr).

Wie dieser Datenbezug vonstatten geht, wird von der Datei

/usr/local/harvest/brokers/SWIB/admin/Collection.conf

gesteuert.

- **Numerische Suche**

Um die Indizierung nach numerischen Werte zu erzwingen, wurde folgender Eintrag in den *broker.conf* Datei gemacht:

```
GlimpseIndex-Flags –n   # -n für die nummerische Werte
```

Anpassung des Abfrageformulars / WWW-Interface (vmquery.html)
Der Pfad der Broker-Hompage bzw. des WWW-Interfaces ist in der Datei *broker.conf* mit dem Eintrag „`/swib/vmquery.html`“ angegeben.

In dem standardmäßigen Abfrageformular *query.html* ist nur ein Feld zum Abfragen vorgegeben:

```
<STRONG>Query:</STRONG> <INPUT NAME="query" TYPE="text" SIZE="50">
```

Dieses Formular habe ich durch *vmquery.html* ersetzt, bei der die notwendigen DCMES-Felder hinzugefügt wurden, um eine Recherche nach den entsprechenden Doubline-Core-Metadaten zu ermöglichen.

Die Ergänzung sehen wie folgt aus:

```
...
<table align=center border=2 cellspacing=1 cellpadding=1 >
<tr>
  <td><b><FONT FACE="Arial">Titel</FONT></b></td>
  <td  colspan="3"  align="left"><input  name="title"  type="text"  si-
ze="50"></td>
<tr>
   <td><b><FONT FACE="Arial">Autor</FONT></b></td>
   <td  colspan="3"  align="left"><input  name="author"  type="text"  si-
ze="50"></td>
<tr>
   <td><b><FONT FACE="Arial">Herausgeber</FONT></b></td>
   <td colspan="3" align="left"><input name="publisher" type="text" si-
ze="50"></td>

<tr>
   <td><b><FONT FACE="Arial">Quelle</FONT></b></td>
   <td  colspan="3"  align="left"><input  name="source"  type="text"  si-
ze="50"></td>

<tr>
   <td><b><FONT FACE="Arial">Publikationsart</FONT></b></td>
   <td colspan="3" align="left"><select name="type">
        <OPTION VALUE="" SELECTED> alle
        <OPTION VALUE="Manual"> Anleitung (Manual)
        <OPTION VALUE="Article"> Aufsatz
        <OPTION VALUE="Image"> Bild (Image)
        <OPTION VALUE="Book"> Buch (Monographie)
        <OPTION VALUE="Dataset"> Dataset (Datensatz, Programm)
                  ...
        </select>
        </td>
<tr>
    <td><b><FONT FACE="Arial">Erscheinungsjahr</FONT></b></td>
    <td  colspan="3"  align="left"><input  name="date"  type="text"  si-
ze="4"></td>
</tr>
<tr>
    <td><b><FONT FACE="Arial">freie Suche</FONT></b></td>
    <td  colspan="3"  align="left"><input  name="freeQuery"  type="text"
size="50"></td>
</tr>
</table>
...
```

Anpassungen an das WWW-Interface

Die Definitionen zur Gestaltung der Ausgabe einer Recherche befinden sich in der Datei

/usr/local/harvest/cgi-bin/lib/BrokerQuery.cf.

Nach diesen Angaben baut das Standard CGI-Skript

/usr/local/harvest/cgi-bin/nph-search.cgi

die Ausgabeseite auf.

Das Standard-CGI-Skript wurde durch ein eigenes CGI-Skript

/usr/local/harvest/cgi-bin/nph-search-SWIB.cgi

und einige Perl-Programme ersetzt. Diese Perl-Programme (*tv_bildeAbfrage.pl, tv_attributHaendler.pl, tv_anpDeutschUml.pl, tv_fuhreAbfrageAus.pl*) beinhalten Sub-Routinen, die vom CGI-Skript *nph-search-SWIB.cgi* aufgerufen werden.

Von dem Standard-Skript wurden jedoch, die Standardfehlerbehandlung, Debugging und Socket-Verbindungsroutine übernommen, die problemlos weiterfunktionierten.

Hintergrundablauf einer Benutzerabfrage an den Broker

(1) Suchbegriff eingeben

Der Benutzer gibt die zu suchende Suchbegriffe in das HTML-Formular (vmquery.html) ein und betätigt die submit-taste. Dadurch wird das Skript *nph-search-SWIB.cgi* mit Benutzerangaben (request) parametrisiert aufgerufen:

```
http://www.bsz-bw.de/harvest/cgi-bin/nph-search-
SWIB.cgi?title=&author=lehmann&publisher=&source=&type=&date=&freeQue
ry=&broker=SWIB&firstquery=yes&caseflag=on&wordflag=on&errorflag=0&op
aqueflag=on&maxobjflag=25&maxlineflag=5&maxresultflag=10000
```

(2) **Request Einlesen**

Der Request im CGI-Skript wird mit der Anweisung „`%RQ = &get_request`“, in die Hash-Tabelle %RQ(Perl-Syntax) eingelesen. Der Einlesevorgang erfolgt in der sub.funktion *get_request*, der in dem Perl-Skript *tv_bildeAbfrage.pl* in folgenden Codesegment definiert ist:

```
sub get_request {

   if ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN, $request, $ENV{'CONTENT_LENGTH'});
    } elsif ($ENV{'REQUEST_METHOD'} eq "GET" ) {
    $request = $ENV{'QUERY_STRING'};
    }
    #$request =~ tr/A-Z/a-z/;
    $cgiQuery = $request;
    %F = split(/[&=]/, $request);

#Leere Eingabe Felder loeschen, da cgiparser nicht damit zurecht kommt

    if ($F{'author'} eq "")
    {
      $request =~ s/author=&//;
    }

    if ($F{'title'} eq "")
    {
      $request =~ s/title=&//;
    }
...

@F = split(/[&=]/, $request);
  &url_decode(@F);
}
```

(3) Deutsche Umlaute Anpassen

Dann wird mit dem Aufruf von `&setGermanSpecifChar()`, falls die Benutzerabfrage Umlaute enthält (ä und ü werden z.B. durch ae und ue ersetzt). Die Sub-Routine „*setGermanSpecifChar*“ ist im „tv_anpDeutschUml.pl“ definiert und sieht wie folgt aus:

```
sub setGermanSpecifChar()
{
  foreach $key (keys %RQ)
  {
    $RQ{$key} =~ s/\n/ /g;

    $RQ{$key} =~ s/\374/ue/g;
    $RQ{$key} =~ s/\334/Ue/g;

    $RQ{$key} =~ s/\344/ae/g;
    $RQ{$key} =~ s/\304/Ae/g;
    ...
}
}#end setGermanSpecifChar()
```

(4) Benutzerabfrage (userquery) bilden

Dann wird die Routine *„buildUserQuery“*, im *tv_bildeAbfrage.pl* aufgerufen, in dem die Benutzerabfrage (userquery) in einer vom Glimpse auswertbaren Form aufgebaut wird. Z.B. wird der Titelfeldinhalt mit `dc.titel=titelfeldinhalt` und die einzelnen Felder und die Feldinhalte, die ein blank enthalten, mit dem UND-Operator standarmäßig aneinander angehängt. Ausserdem wird in einer sub-routine (check_Query), nach allgemeinen Syntax-Fehlern geprüft.

```
sub buildUserQuery
{
   $title      = $RQ{'title'};
   $author      = $RQ{'author'};
   $publisher   = $RQ{'publisher'};
  ...
  if ($title ne "")
    {
          $title = &checkField($title);

          $tmpTitle    = $title;
          $tmpAlter    = $title;
          $titleDC     = "\"dc\.title\"".":";
          $titleAltDC  = "\"dc\.title\.alternative\"".":";

          $tmpTitle    = &anpasAndOr($tmpTitle,$titleDC);
          $tmpAlter    = &anpasAndOr($tmpAlter,$titleAltDC);

          $titleQuery        =    "(".$titleDC.$tmpTitle.")"."    OR
"."(".$titleAltDC.$tmpAlter.")";

           if ($userquery ne "")
           {
              $userquery               =   "(".$userquery.")"."   AND
"."(".$titleQuery.")";
           }
           else
           {
             $userquery = $titleQuery;
           }
       }
...
}#End buildUserQuery
```

(5) Attribute anhängen

Nach der Bildung der Benutzerabfrage wird die Routine *„pushAttr“*, die im *tv_attributHaendler.pl* definiert ist, aufgerufen. Diese Routine hängt die weiteren Attribute an das Attributenfeld @atts, das ein Bestandteil der Broker-Abfrage ist. Der Broker liefert bei den gefundenen Objekten alle Attribute zurück, die im @atts-Feld angegeben wurden. Diese Attribute, die das SOIF-Objekt beschreiben, werden dann zur besseren Übersicht mit ausgegeben:

```
sub pushAttr()
{      # hange zusatzlich attribut

  push(@atts,"dc.title") unless grep (/^dc.title$/,@atts);
  push(@atts,"dc.title.alternative")           unless           grep
(/^dc.title.alternative$/,@atts);
  push(@atts,"dc.type") unless grep (/^dc.type$/,@atts);
  push(@atts,"dc.description")              unless              grep
(/^dc.description$/,@atts);
  push(@atts,"dc.subject") unless grep (/^dc.subject$/,@atts);
  push(@atts,"dc.source") unless grep (/^dc.source$/,@atts);
  push(@atts,"dc.creator.name")              unless              grep
(/^dc.creator.name$/,@atts);
  push(@atts,"dc.creator.corporate")           unless           grep
(/^dc.creator.corporate$/,@atts);
}
```

(6) Brokerabfrage aufbereiten

Die Brokerabfrage, die letztendlich an den Glimpse gestellt wird und u.a. die Benutzerabfrage und die Attribute enthält, wird in folgenden Zeilen aufbereitet:

```
# BUILD QUERY STRING
#
$query = "";
$query .= $userclass . " AND " if ($userclass ne "");
$query .= $userquery;

# BUILD BROKER QUERY
#
$bquery = "#USER ";
$bquery .= "#opaque "                if ($opaqflag);
$bquery .= "#desc "                 if ($descflag);
$bquery .= "#index timeout $lifetime "        if ($lifetime ne "");
$bquery .= "#index error $errors "        if ($errors ne "");
$bquery .= "#index maxresult $maxresult "    if ($maxresult ne "");
$bquery .= "#index maxfiles $maxfiles "        if ($maxfiles ne "");
$bquery .= "#index maxlines $maxlines "        if ($maxlines ne "");
$bquery .= "#index case ";
$bquery .= $caseflag ? "insensitive " : "sensitive ";
$bquery .= "#index matchword "            if ($wordflag);
$bquery .= $attributes;
$bquery .= "#END ";
$bquery .= $query;
```

(7) Broker-Abfrage duchführen und das Suchergebnis ausgeben

Die Broker-Abfrage in der Variablen `$bquery` wird mit der Anweisung

```
&do_query ($SOCK, $bquery)
```

an das Socket übergeben. Die Sub-Routine *do_query* ist im *tv_fuhreAbfrageAus.pl* definiert. In dieser Sub-Routine wird das Ergebnis über das Socket eingelesen. Falls die Anzahl der Suchergebnisse über 25 liegt, werden nur die ersten 25 ausgegeben. Will man sich jedoch alle Treffer anzeigen lassen, ist dafür am Ende der Trefferliste ein Link ausgegeben. Durch die Beschränkung der Ergebnisanzeige werden zum einen die Treffer schneller angezeigt und zum anderen der Netzwerkverkehr entlastet. Da die qualitativ besseren Treffer am Anfang der Ergebnisliste stehen (Ranking) wird es in den meisten Fällen sowieso nicht nötig sein, sich alle Treffer anzeigen lassen. Die Ergebnisausgabe wird in der Sub-Funktion *„printObjekt“* aufbereitet und dem Browser zurückgeliefert.

```
while (<$S>)    #S = Socket
           {
                chop $_;
                next if (/^$/o);
                print "|$_|\n" if ($debug);
                next if (/^200/o);
                if (/^126 -/o || /^120 -/o || /^103 -/o || /^111 -/o)
                {
                    $OBJ[++$#OBJ] = $_ . "\n";
                    last if (/^103 -/o || /^111 -/o);
                }
                else
                {
                    $OBJ[$#OBJ] .= $_ . "\n";
                }
            }

           close ($S)  || &fatal ("socket: $!\n");

           $object_count = (scalar (@OBJ)) – 2
           print "<H3> Es wurden $object_count Treffer gefunden: </H3>";
           print "<p> </p>";
            if ($firstquery eq "non" || $object_count <= 25)
            {
               &printObjects();
            }
            else
            {
            $letzteElem    = $OBJ[$#OBJ
            $vorletzteElem = $OBJ[-2];
            @tmpOBJ = @OBJ[0..24]
            ...

                  &printObjects;
          }
…
```

Somit ist nach diesen Schritten die Benutzerabfrage durch den Broker abgearbeitet, und das Ergebnis wird im Browser des Benutzers präsentiert.

5.5.3 Administration

Wichtige Dateien für administrative Zwecke

$HARVEST_HOME = usr/local/harvest

- *$HARVEST_HOME/cgi-bin/ :*

  Enthält die CGI-Skripte wie z.B. *nph-search-SWIB.cgi*

- *$HARVEST_HOME/gatherers/... :*
  Enthält in Unterverzeichnissen die einzelnen Gatherer.

- *$HARVEST_HOME/gatherers/SWIB/ :*
  enthält u.a.
- *SWIB.cf* ist die Konfigurationdatei des Gatherers (Name, Port, Verzeichnis, Suchgebiet).
- *RunGatherd* Startet den Gatherer-Demon.
- *RunGatherer* Startet den Gatherer.
- *log.errors* enthält die während eines Gatherer-Durchlaufs aufgetretenen Fehlermeldungen.
- *log.gatherer* enthält die während eines Gatherer-Durchlaufs aufgetretenen Meldungen.

- *$HARVEST_HOME/brokers/... :*
  Enthält in Unterverzeichnissen u.a. die einzelnen Broker und die Datei:
- *Brokers.cf* die Auflistung aller erzeugten Broker (mit Port- und Hostnummer).

- *$HARVEST_HOME/brokers/SWIB/ :*
- *RunBroker* startet den Broker.
- *stats.html* enthält eine Statistik für den Broker.

- *$HARVEST_HOME/brokers/SWIB/admin/ :*
- *admin.html* mit Hilfe dieser Datei läßt sich der Broker komfortabel über einen WWW-Browser verwalten.
- *Brokers.conf* enthält die Konfiguration des Brokers (u.a. auch das Paßwort).
- *Collection.conf* Hier läßt sich einstellen, wie und was der Broker sammeln soll.

- *$HARVEST_HOME/brokers/SWIB/objects/ :*

Enthält in Unterverzeichnissen die gesammelten SOIF-Daten; muß vor jeder „von Hand“ gestarteten Indizierung komplett gelöscht werden.

Starten von Gatherer und Broker

Beim Booten des Rechners, auf dem das Harvest-System lauft, muß der Gatherer-Demon *Gatherd* mit dem Befehl *RunGatherd*

```
$HARVEST_HOME/gatherers/SWIB/RunGatherd
```

und der Broker-Demon mit *RunBroker*

```
$HARVEST_HOME/brokers/SWIB/RunBroker
```

gestartet werden.

Diese Prozesse erfolgen über Cron-Jobs.

Datensammlung durch den Gatherer

Der Gatherer soll vor dem Ablauf der eingestellten „Lebenszeit“ (Time-To-Live) der SOIF-Objekte zur Aktualisierung des Datenbestandes neu gestartet werden. Dies geschieht mit dem Shell-Skript RunGatherer:

```
$HARVEST_HOME/gatherers/SWIB/RunGatherer
```

Eine manueller Start entfällt hier auch, da dies über Cron-Job erledigt wird.

Manueller Start des SWIB-Brokers zur Indizierung

Der SWIB-Broker ist momentan so eingestellt, daß er alle 24 Stunden Daten aus dem Gatherer abholt und anschließend seine Index-Datei aktualisiert.
Falls der Broker Manuell gestartet werden sollte, müssen folgende Schritte durchgeführt werden:

I. folgende Dateien im Verzeichnis */usr/local/harvest/brokers/SWIB* sind zu löschen:

```
$rm .gl*                    #Alle Glimpse-Dateien
$rm broker.out              #Info über vorherige Brokerablauf
$rm ./admin/LOG             #LOG-Datei von vorigen Ablauf
$rm ./admin/Registry        #Registry-Datei
$mv objects objects.old     #Alle vorhandene Objeskte
$rm -rf objects.old &
```

Hierfür habe ich ein Shell-Skript *startBroker* geschrieben, mit dem alle diese Dateien gelöscht werden können.

II. Der Broker-Demon muß „gekillt“ werden:

Mit dem Befehl

```
$ps -ef |grep broker
```

sind die PID der Prozesse Broker und Glimpseserver zu ermitteln. Die ermittelten Prozesse sind dann mittels

```
$kill -9 (PID#)
```

zu entfernen.

III. Jetzt kann der Broker mit folgendem Befehl gestartet werden:

```
$broker ./admin/broker.conf –new
```

Insgesamt fällt die Administration des laufenden SWIB-Suchdienstes kaum ins Gewicht, die nötigen Prozesse sind durchschaubar strukturiert und einfach anzustossen.

5.6 Ein Beispiel-Recherche an SWIB-Suchdienst

Mit der Einstiegsseite (http://www.bsz-bw.de/swib/vmquery.html) werden dem Benutzer Felder angeboten, in denen er seine Suchbegriffe eingeben kann. Als Angebot bestehen Titel Autor, Herausgeber, Quelle, Publikationsart, Erscheinungsjahr. Hinterlegt ist dabei eine Übersetzung, die gezielt auf die verschiedenen Indices zugreift, die mit den einzelnen DC-Elementen gebildet wurden. Darüber hinaus besteht die Möglichkeit, im Feld „freie Suche“ auch nach Begriffen zu recherchieren, die nicht mit DCMES-Elementen beschrieben sind (Siehe Abb. 13).

Gesucht wird im Beispiel nach dem Autor "Lehmann". Nachdem die Suchbegriffe eingegeben wurden, wird der Recherchevorgang mit dem Button „Suchen“ ausgelöst. An die Suchmaschine übergeben wird die Anfrage nach "dc.creator.name":lehmann als Person oder nach "dc.creator.corporate":lehmann als Körperschaft.

Abbildung 13:SWIB-Einstiegsseite

Abbildung 14:Darstellung der Suchergebnissen

Als Antwort des Harverstsystems werden die Suchergebnisse präsentiert (Siehe Abb. 14) mit Angabe der vorher automatisch erzeugten Suchanfrage. In diesem Fall sind fünf Treffer gefunden worden, wobei eine getrennt durchgeführte Kontrolle zu Recall und Precision ergeben hat, dass alle relevanten Dokumente gefunden wurden.

6 Ausblick

Unabsehbar ist, wie lange noch die Problematik der gezielten Recherche im Internet für die wissenschaftlichen Institutionen (Bibliotheken, Archive, Volltextanbieter, Universitäten usw.) bestehen bleibt. Gewiß ist aber, daß diese Institutionen für ihre Nutzerschaft immer mehr auch die Ressourcen des WWW in ihr Angebot dauerhaft integrieren müssen, und eben nicht nur solange, als die für Wisschenschaft, Forschung und Bildung relevante Dokumentmenge weiter steigt. Ein Lösungsversuch der Dokumentanbieter stellt die Vergabe strukturierter Metadaten dar, denn mit ihnen kann eine gezielte Recherche auf Objekte des WWW eröffnet werden, die der Präzision und dem Retrieval der traditionellen bibliographischen Datenbanken für gedruckte Publikationen entspricht. Die Verbindung der gleichzeitigen Recherche auf Datenbanken für gedrucktes Material und in Datenangeboten von Internetressourcen ist über die Metadatennutzung bereits erfolgt (Digitale Bibliothek Nordrheinwestfalen http://www.hbz-nrw.de/DigiBib).

Zu hoffen ist nicht, daß die allgemeinen Suchdienste für die wissenschaftlichen Zwecke in naher Zukunft deutlich verbessert werden: Ihre kommerzielle Orientierung, die Konkurrenz um Werbung, die wenig spezifische Auswahl ihrer Datenquellen werden diese Dienste auch zukünftig für die forschenden Institutionen und Wissenschaftler nur bedingt brauchbar erscheinen lassen. Marketingstrategien stellen nicht die Objektivität, Informationsqualität und Vertrauenswürdigkeit im Vordergrund, sondern gewinnbringende Maßnahmen. Die entwickelten Rankingsservices, die einen höheren Rangplatz in der Treffernliste von Suchmaschinen versprechen, müssen kritisch betrachtet werden: die Bezahlung eines höheren Rankings durch den Objektanbieter beim Suchdienst verfälscht die Gleichbehandlung des Angebots und die Qualität des Informationsgehalts nachhaltig. Niemand kann solche Absprachen zwischen kommerziellen Suchdiensten und sogenannten Rankingservice-Firmen mit Gewißheit und auf Dauer ausschließen.

Die Schwächen der allgemeinen Suchdiensten können durch den Einsatz lokaler, auf den speziellen Bedarf eines Nutzerkreis zugeschnittener Suchdienste ausgeglichen werden. Die Probleme mit den Datentypen, die sich nicht direkt indizieren lassen, die Ex-

trahierung der unterschiedliche Datenformaten bei der Indizierung, die Auflösung des Zusammenhangs von Kontext und Struktur beim Indizieren sind auf technischer Basis, sicher auch unter Einsatz computerlinguistischer Modelle zu lösen. Die zunehmende und konsequente Beschreibung der Objekte mit Metadaten und die Standardisierung dieser Metadaten selbst, sind Aufgaben, die von den Informationsanbietern in Bibliotheken, Archiven und Rechenzentren im Verein mit den Entwicklungen in der Informatik zukünftige Aufgabe bleiben.

Für die Betreiber von Internet Suchdiensten im wissenschaftlichen Bereich, insbesondere bei der Universitäten, hat sich das Harvest-System praktisch durchgesetzt. Harvest ist zwar relativ komplex, benötigt nicht zu unterschätzenden Aufwand zur Anpassung an die eigenen Bedürfnisse und verwirrt zunächst durch die wechselnden Bezeichungnen des Query-Interfaces im Broker. Diese Mühe wird mit seiner Flexibilität und Stabilität, seiner leichten Administrierbarkeit und vor allem mit seiner verteilten Indizierungsmöglickeit belohnt. Das Fehlverhalten z.B. bzgl. der deutschen Umlaute wird hoffentlich in den nächsten Versionen behoben: an vielen Universitäten wird an Erweiterungen, Anwendungen und Pages gearbeitet. Damit besteht ein starkes Interesse an der Weiterentwicklung des Harvest-Systems.

Gelungen ist die Implementation des Harvestsystems in die Systemumgebung des BSZ und damit der Aufbau des SWIB-Suchdienstes. Aber auch hier ist die Entwicklung damit nicht abgeschlossen, denn die technischen Möglichkeiten des Harvestpaketes werden nicht ausgeschöpft: bislang greift SWIB ausschließlich auf die im virtuellen Medienserver des BSZ angebotenen digitalen Medien zurück.

Eine Erweiterung der SWIB-Datenbasis ist vorgesehen:

- eine Ausweitung des „Anbieterkreises für elektronische Medien“ durch weitere Bibliotheken und wissenschaftliche Einrichtungen ist in Planung. Insbesondere muß das Augenmerk gerichtet werden auf den Einbezug internationaler Angebote. Hierfür bestehen mit der Beteiligung am Cooperative Online Cataloging Project des Online Computer Library Center (OCLC) in den USA als dem weltgrößten Anbieter bibliographischer Daten wichtige

Kontakte: unter Nutzung von Metadaten des DCMES wird hier eine internationale Datenbank für Webressourcen aller Art aufgebaut. Ihre Nachnutzung in vielfältiger Form steht denen offen, die am Projekt beteiligt sind.

- Die Ausweitung auf weitere Medientypen ist erforderlich: die Vollständigkeit im Angebot von Hochschulschriften, die online verfügbar sind, geht einher mit einem weitgehenden Verzicht auf den Nachweis z.B. von Homepages fremder WWW-Server, einzelnen Seiten und Angeboten auf WWW-Servern, die nicht zum institutionellen Kundenkreis des BSZ zählen. Wenn dies sicher auch nur in strenger relevanzbezogener Auswahl erfolgen kann, kann es doch nicht in Zukunft vernachlässigt werden, soll eine Trennung in Informationsvermittlung digitaler und konventioneller Medien vermieden werden.

- Das Harvestsystem stellt eine Anwendung dar, die vor allem für textuelle Objekte und Strukturen entwickelt wurde. Vorhandene und entstehenden multimediale Objekte können über ihre Beschreibung einbezogen werden, weitere Suchangebote stossen aber noch auf große Schwierigkeiten. Die Beteiligung des BSZ an einem Projekt zu dem auf multimediale Inhalte ausgelegten Produkt IBM Digital Library wird hier in Zukunft interessante Fortschritte bringen können.
- von bibliothekarischem Interesse ist der Einbezug von bibliographischen Normdaten: die in ihnen existierenden Verweisungsformen und semantischen Netze können in Rechercheprozessen zu einer deutlichen Ergebnisverbesserung beitragen. Ihre noch nicht erfolgte Integration in die Metadatenstrukturen und Metadatenangebote wird bibliothekarische Aufgabe sein, die ohne Begleitung der Informatik nicht sinnvoll möglich sein wird.

Vergessen werden darf aber nicht, daß bei großen Dokumentenzahlen (>65000) eine mächtigere Datenbank als Glimpse[31] unterlegt werden muß. Und damit schließt sich der Kreis mit der nächsten Aufgabe.....

[31] Vgl. Harvesting Mathematics, Judith Plümer, Roland Schwänzl, Fachbereich Mathematik Universität Osnabrück

Literaturverzeichnis

Babiak, Ulrich: Effektive Suche im Internet; 1. Aufl. O'Reilly Verlag, Köln 1997

Bekavac, **Bernard:** Tutorial zur Suche im WWW/Internet (1.2), Informationswissenschaft - Universität Konstanz, URL: http://www.inf-wiss.uni-konstanz.de/suche/such_tutorial.html (letzter Zugriff 25.02.00)

Bibliotheksservice-Zentrum Baden-Württemberg [Hrsg.]: BSZ kompakt, 3. Aufl. 1999

Borggraefe, Stefan / Schade, Oliver: Programmpakete zur Indizierung der eigenen Website, URL: http://www.heise.de/ix/artikel/1999/02/089/ (letzter Zugriff 25.02.00)

Das Verzeichnis der im deutschen Sprachraum erschienenen Drucke des 17. Jahrhunderts, URL: http://www.VD17.de (letzter Zugriff 25.02.00)

Dublin Core in der Interpretation des BSZ Teil 1: Syntax, URL: http://www.bsz-bw.de/diglib/medserv/konvent/metadat/dcsyntax.html (letzter Zugriff 25.02.00)

Dublin Core Metadata Element Set, Version 1.1: Reference Description, URL: http://purl.org/DC/documents/rec-dces-19990702.htm (letzter Zugriff 25.02.00)

Göbel, Oliver / Gabriele Mayer: Mit Harvest durchs Internet, URL: http://www.uni-stuttgart.de/rus/Bi/1996/4/File4.html (letzter Zugriff 25.02.00)

Hardy, Darren R. / Schwartz, Michael F. / Wessels, Duane: Harvest User's Manual; University of Colorade, Boulder 1996

Hayer, Hans / Kolbeck, Rainer: Erfolgreiche Internetsuche; Markt und Technik, Haar bei München 1996

Karzauninkat, Stefan: Die Suchfibel, Wie findet man Informationen im Internet?, URL: http://suchfibel.de/3allgem/index.htm(letzter Zugriff 25.02.00) Ernst Klett Verlag Leipzig

Koch, Traugott: Verbesserung der Recherchemöglichkeiten im Internet - internationaler Überblick, URL: http://www.lub.lu.se/tk/demos/DGD97.html (letzter Zugriff 25.02.00)

Lamprecht, Stephan: Professionelle Recherche im Internet; 2. Aufl. Hanser, München 1999

Niedersächsische Staats- und Universitätsbibliothek Göttingen 1997/98, Einführung in Metadaten, URL: http://www2.sub.uni-goettingen.de/metall.html (letzter Zugriff 25.02.00)

Plümer, Judith / Schwänzl,Roland: Harvesting Mathematics, Fachbereich Mathematik Universität Osnabrück, Tecnical Report 1996

Resource Description Framework (RDF), URL: http://www.w3.org/RDF/ (letzter Zugriff 25.02.00)

Rusch-Feja, Diann: Entwicklungsgeschichte des Dublin Core, BIBLIOTHEKS-DIENST Heft 4, 97, URL: http://www.dbi-berlin.de/dbi_pub/bd_art/97_04_08.htm (letzter Zugriff 25.02.00)

Rusch-Feja, Diann: Mehr Qualität im Internet - Entwicklung und Implementierung von Metadaten, URL: http://www.mpib-berlin.mpg.de/dok/metadata/metaifbs.htm (letzter Zugriff 25.02.00)

Suchmaschine Acoon, Info: http://www.acoon.de/infos.html (letzter Zugriff 25.02.00)

SWIB: Suchdienst Wissenschaftliche Bibliotheken: BSZ-Konstanz, URL:http://www.bsz-bw.de/diglib/medserv/projekt/swib/swib.html (letzter Zugriff 25.02.00)

Anhang A

Die Elemente des DCMES

Die derzeit aktuelle Version des DCMES, Version 1.1, enthält die folgenden 15 Elemente[32]:

Element: Title (DC.TITLE)	Name:	Title
	Identifier:	Title
	Definition:	A name given to the resource.
	Comment:	Typically, a Title will be a name by which the resource is formally known.
Element: Creator (DC.CREATOR)	Name:	Creator
	Identifier:	Creator
	Definition:	An entity primarily responsible for making the content of the resource.
	Comment:	Examples of a Creator include a person, an organisation, or a service.Typically, the name of a Creator should be used to icate the entity.
Element: Subject (DC.SUBJECT)	Name:	Subject and Keywords
	Identifier:	Subject
	Definition:	The topic of the content of the resource.
	Comment:	Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
Element: Description (DC.DESCRIPTION)	Name:	Description
	Identifier:	Description
	Definition:	An account of the content of the resource.
	Comment:	Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.
Element: Publisher (DC.PUBLISHER)	Name:	Publisher
	Identifier:	Publisher
	Definition:	An entity responsible for making the resource available

[32] http://purl.org/DC/documents/rec-dces-19990702.htm

	Comment:	Examples of a Publisher include a person, an organisation, or a service. Typically, the name of a Publisher should be used to indicate the entity.
Element: Contributor (DC.CONTRIBUTORS)	Name:	Contributor
	Identifier:	Contributor
	Definition:	An entity responsible for making contributions to the content of the resource.
	Comment:	Examples of a Contributor include a person, an organisation, or a service.Typically, the name of a Contributor should be used to indicate the entity.
Element: Date (DC.DATE)	Name:	Date
	Identifier:	Date
	Definition:	A date associated with an event in the life cycle of the resource.
	Comment:	Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and follows the YYYY-MM-DD format.
Element: Type (DC.TYPE)	Name:	Resource Type
	Identifier:	Type
	Definition:	The nature or genre of the content of the resource.
	Comment:	Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the working draft list of Dublin Core Types [DCT1]). To describe the physical or digital manifestation of the resource, use the FORMAT element.
Element: Format (DC.FORMAT)	Name:	Format
	Identifier:	Format
	Definition:	The physical or digital manifestation of the resource.
	Comment:	Typically, Format may include the media-type or dimensions of the resource. Format may be used to determine the software, hardware or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [MIME] defining computer media formats).
Element: Identifier (DC.IDENTIFIER)	Name:	Resource Identifier
	Identifier:	Identifier

	Definition:	An unambiguous reference to the resource within a given context.
	Comment:	Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Example formal identification systems include the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).
Element: Source (DC.SOURCE)	Name:	Source
	Identifier:	Source
	Definition:	A Reference to a resource from which the present resource is derived.
	Comment:	The present resource may be derived from the Source resource in whole or in part. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.
Element: Language (DC.LANGUAGE)	Name:	Language
	Identifier:	
	Definition:	A language of the intellectual content of the resource.
	Comment:	Recommended best practice for the values of the Language element is defined by RFC 1766 [RFC1766] which includes a two-letter Language Code (taken from the ISO 639 standard [ISO639]), followed optionally, by a two-letter Country Code (taken from the ISO 3166 standard [ISO3166]). For example, 'en' for English, 'fr' for French, or 'en-uk' for English used in the United Kingdom.
Element: Relation (DC.RELATION)	Name:	Relation
	Identifier:	Relation
	Definition:	A reference to a related resource.
	Comment:	Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.
Element: Coverage (DC.COVERAGE)	Name:	Coverage
	Identifier:	Coverage
	Definition:	The extent or scope of the content of the resource.
	Comment:	Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity).

		Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [TGN]) and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges.
Element: Rights (DC.RIGHTS)	Name:	Rights Management
	Identifier:	Rights
	Definition:	Information about rights held in and over the resource.
	Comment:	Typically, a Rights element will contain a rights Management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the Rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource.

Anhang B

Beispiel zur Harvest-Installations-Ablauf mit RunHarvest

```
$ RunHarvest

  ########################################################################
  #                          Welcome to Harvest!                         #
  ########################################################################

   This program will create, configure, and run a Harvest Broker and
   Gatherer.  It will allow you to index one or more WWW, FTP, or Gopher
   servers, using a set of defaults for content extraction and indexing.
   We offer 3 standard configurations:

         1. Index your entire WWW site.
         2. Index an entire WWW site (or sites).
         3. Index selected parts of WWW, FTP, or Gopher sites.

   NOTE: When this program asks you a question, you can accept the
         default answer shown within the square brackets [ ]'s by
         pressing ENTER, or you can enter your own answer.

Do you want to continue? [yes]:

   You will be asked a few questions about what you want Harvest to do,
   and about a few basic details of your WWW, FTP, or Gopher server(s).
   You'll need several megabytes of disk space to run these Harvest
   servers; the amount of disk space needed depends on the size of the
   servers that you're indexing.  If you are uncertain these questions,
   then stop now and contact your server administrator.

   This program is broken down into 4 question and answer sessions.
   Based on your answers, it will create and configure Harvest servers.
   It will also set up your system to run the Harvest servers regularly.

   A comprehensive user's manual is available for Harvest via WWW at:

         http://www.tardis.ed.ac.uk/harvest/docs/

Do you want to continue? [yes]:
------------------------------------------------------------------------
STEP 1: Describe your local WWW server.
------------------------------------------------------------------------

   The Harvest Broker requires that you have access to a WWW server
   (e.g., httpd).  If you don't currently run a WWW server, then you
   will need to install and configure one before you can run Harvest.

On which host does your WWW server run? [birnau.bsz-bw.de]:
On which port does your WWW server run? [80]:

------------------------------------------------------------------------
STEP 2: Select a standard configuration.
------------------------------------------------------------------------

   We offer 3 standard configurations:

         1. Index your entire WWW site.
         2. Index an entire WWW site (or sites).
         3. Index selected parts of WWW, FTP, or Gopher sites.

Please select a configuration [1]: 3

------------------------------------------------------------------------
```

```
STEP 3: Configure your new Harvest servers.
------------------------------------------------------------------------

   To configure your Harvest Broker and Gatherer, answer the following:

Enter a short description of this Harvest server [none]: Harvest Test Server
Enter a one-word description of this Harvest server [none]: HTS
Where do you want to install the Gatherer?:
        [/usr/local/harvest/gatherers/HTS]:
On which port should the Gatherer run? [8500]:
Where do you want to install the Broker?:
        [/usr/local/harvest/brokers/HTS]:
On which port should the Broker run? [8501]:
Enter a password for the Broker administrative commands []: pw_varli

Enter the list of URLs for the collections that you'd like to index.
The URLs that you enter below will be classified as 'RootNodes' and will
be enumerated (e.g. by recursive FTP directory listings for FTP URLs).
Terminate this list by entering a period ('.') on a line by itself.
Enter URL: http://www.bsz-bw.de/virtuelleServer/depot/dokersch.html
Enter URL: .

   To configure your Harvest Broker and Gatherer, answer the following:

Enter a short description of this Harvest server [none]: Harvest Test Server
Enter a one-word description of this Harvest server [none]: HTS
Where do you want to install the Gatherer?:
        [/usr/local/harvest/gatherers/HTS]:
On which port should the Gatherer run? [8500]:
Where do you want to install the Broker?:
        [/usr/local/harvest/brokers/HTS]:
On which port should the Broker run? [8501]:
Enter a password for the Broker administrative commands []: pw_varli

Enter the list of URLs for the collections that you'd like to index.
The URLs that you enter below will be classified as 'RootNodes' and will
be enumerated (e.g. by recursive FTP directory listings for FTP URLs).
Terminate this list by entering a period ('.') on a line by itself.
Enter URL: http://www.bsz-bw.de/virtuelleServer/depot/dokersch.html
Enter URL: .

------------------------------------------------------------------------
STEP 4: Create and run the Harvest servers.
------------------------------------------------------------------------

   Now, this program will create Harvest servers based on your input.
   Then, it will run the Harvest servers.

Creating the Gatherer...
Successfully created the Gatherer!
Would you like to edit the Gatherer's workload specification?:
        [no]:
Creating the Broker...
Successfully created the Broker!
Running the Gatherer.
WARNING:  For large sites, this may take several hours...
Done running the Gatherer!
Running the Broker.
WARNING:  For large sites, this may take several hours...
Done running the Broker!
Done.

   Your Harvest Servers are now running.  To access them, refer to

       http://birnau.bsz-bw.de:80/Harvest/brokers/HTS/summary.html
```

Anhang C

Folgendes SOIF-Objekt wurde vom SWIB-Gatherer aus der BSZ-Hompage (http://www.bsz-bw.de) erzeugt:

```
SOIF Object for: http://www.bsz-bw.de/index.html

@FILE { http://www.bsz-bw.de/index.html
update-time{9}: 949389971
last-modification-time{9}:      949389971
time-to-live{7}:        2419200
refresh-rate{6}:        604800
gatherer-name{23}:      SWIB-Suchdienst vom BSZ
gatherer-host{17}:      thurgau.bsz-bw.de
gatherer-version{3}:    1.5
type{4}:        HTML
file-size{4}:   7349
md5{32}:        dc0bc33ead20315f01fb24d5eaac15fc
dc.creator.corporate{45}:       Bibliotheksservice-Zentrum Baden-Wuerttemberg
title{64}:      Bibliotheksservice-Zentrum Baden-Wuerttemberg - Homepage des BSZ
url-references{1448}:   http://www.bsz-bw.de
http://www.bsz-bw.de
/wwwroot/s10000_d.html
/wwwroot/home_swb.html
http://www.bsz-bw.de/projekte.html
http://www.bsz-bw.de/bibldienste
http://www.bsz-bw.de/links
http://www.bsz-bw.de/eu/bodensee.html
/wwwroot/s73000_d.html
http://www.bsz-bw.de/Excite/AT-Gesamtquery.html
http://www.bsz-bw.de/swib/vmquery.html
http://www.bsz-bw.de/aktuell/new.html
http://www.bsz-bw.de/aktuell/
http://www.bsz-bw.de/statistik/
http://www.bsz-bw.de/bibldienste/bsz-karte.html
http://www.bsz-bw.de/download
/wwwroot/s10000_d.html
/cgi-bin/opacform.cgi
http://webpac.bsz-bw.de/webpac-cgi/wgbroker?new+-access+top
/bibinfo/
/subito/
/bibldienste/redi.html
/projekte.html
/verbundsys/
/lokalsys/
/diglib/
/swib/vmquery.html
/wwwroot/home_swb.html
/wwwroot/s73000_d.html
/wwwroot/s70000_d.html
/wwwroot/text/zkhome.html
/download
ftp://ftp.swbv.uni-konstanz.de/pub
/bibldienste/listen.html
/bibldienste
/bibldienste/bibliotheken.html
/bibldienste/deutsch.html
/bibldienste/europa.html
/bibldienste/verzeichnis.html
/bibldienste/verbuende.html
/depot/dokersch/3400000/3421000/3421308k.html
/bibldienste/arbeit.html
/links
```

```
/links/presse.html
/links/kiosk.html
/links/buecher.html
/wwwroot/text/verlage.html
/links/literatur.html
/links/suchdienste.html
/links/wetter.html
/links/verkehr.html
/wwwroot/text/fabio.html
/koondaba
/eu/bodensee.html
/links/internet.html
/links/medien.html
/links/kultur.html
/links/vermischt.html
mailto:webmaster@bsz-bw.de
dc.publisher{45}:       Bibliotheksservice-Zentrum Baden-Wuerttemberg
dc.creator.name{17}:    Heymans, Wolfgang
dc.date.current{33}:    (SCHEME=ANSI.X3.30-1985) 20000118
description{128}:       Homepage des Bibliotheksservice-Zentrum Baden-Wuerttemberg
(BSZ),
 Suedwestdeutscher Bibliotheksverbund (SWB) und Zentralkatalog
dc.subject{106}:                (SCHEME=SWD)  Bibliotheksservice-Zentrum  Baden-
Wuerttemberg, Suedwestdeutscher Bibliotheksverbund, Homepage
images{56}:     /wwwroot/gif_01/bszlogo60.gif
/wwwroot/gif_01/new-kl.gif
dc.title{64}:   Bibliotheksservice-Zentrum Baden-Wuerttemberg - Homepage des BSZ
keywords{840}:

adressen aktuelles aller aus baden bibinfo bibliothekarische bibliotheken biblio-
theksadressen bibliotheksdienste bibliotheksservice bodensee bsz buecher
datenbank datenbanken des deutschland digital diskussionslisten dokumentliefer-
dienst download erreichen euregio europa fabio fachbibliographien fahrplaene
ftp fuer home ifb information informationen informationsmittel internet kataloge
kebweb koondaba kultur kulturraum landkarte library links literatur lokalsystem
medien nationalbibliotheken neues online onlinedatenbanken opac projekte publika-
tionen redi regionale server sie statistik subito suchdienst suchdienste suche
suchen swb swib testbetrieb uns verbundsystem verbundsysteme verlage vermischtes
webmaster webpac welt weltweit werkzeuge wetter wissenschaftliche
wuerttemberg www zeitschriften zeitungen zentralkatalogs zentrum

dc.identifier{24}:      (SCHEME=URL) /index.html
dc.description{128}:    Homepage des Bibliotheksservice-Zentrum Baden-Wuerttemberg
(BSZ),
 Suedwestdeutscher Bibliotheksverbund (SWB) und Zentralkatalog
dc.language{23}:        (SCHEME=NISOZ39.53) GER
dc.format{22}:  (SCHEME=imt) text/html
body{1276}:

        Home

        Datenbanken

        Publikationen

        Projekte

        Bibliotheksdienste

        Links

        KEBweb



        Adressen
```

```
        Suchen

        SWIB Suche

        Aktuelles



        Statistik

        Landkarte

        Download

Bibliotheksservice-Zentrum Baden-Wuerttemberg
Serviceangebote im WWW

Datenbanken

SWB-Datenbank (WWW-OPAC)
SWB-Datenbank (WebPAC im Testbetrieb)
 Bibliotheksadressen BIBINFO
SUBITO - Dokumentlieferdienst
ReDI - Regionale Datenbank-Information

Projekte

Neues Verbundsystem
Neues Lokalsystem
Digital Library
SWIB: Suchdienst Wissenschaftliche Bibliotheken

Informationen + Publikationen

So erreichen Sie uns
 Publikationen des SWB
Publikationen des Zentralkatalogs
Download   *
FTP-Server
Diskussionslisten

Bibliotheksdienste

Bibliotheken & Online-Kataloge:
Deutschland *
Europa *
Weltweit
Verbundsysteme & Nationalbibliotheken
IFB: Informationsmittel fuer Bibliotheken
Bibliothekarische Werkzeuge

Links aus aller Welt

Zeitungen *
Zeitschriften *
Buecher *
Verlage *
Literatur *
Suchdienste *
Wetter  *
Fahrplaene

FabiO Fachbibliographien     *
KoOnDaba Onlinedatenbanken

KEBweb - Kulturraum EUREGIO Bodensee
```

```
und vieles andere aus den Rubriken
Internet *
Medien *
Kultur *
Vermischtes

    webmaster@bsz-bw.de
     18.01.2000


}
```

Ehrenwörtliche Erklärung

Hiermit erkläre ich, Tarik Varli, geboren am Geb. 15.10.1969 in Kemah, ehrenwörtlich,

(1) daß ich meine Diplomarbeit mit dem Titel:

„Evaluierung und prototypische Implementierung eines Suchdienstes für wissenschaftliche Bibliotheken auf Basis strukturierter Metadaten“

im Bibliothek Service Zentrum Konstanz unter Anleitung von **Professor Dr. Klein** und **Dipl.-Inf.Wiss.Andreas Lehmann** selbständig und ohne fremde Hilfe angefertigt habe und keine anderen als in der Abhandlung angeführten Hilfen benutzt habe;

(2) daß ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe.

Ich bin mir bewußt, daß eine falsche Erklärung rechtliche Folgen haben wird.

Konstanz, 28.02.2000