Konzeption eines Recherchearbeitsplatzes für ein elektronisches Zeitungsarchiv

Marc Sieburg

Technische Universität München
Institut für Informatik

Diplomarbeit

Konzeption eines Recherchearbeitsplatzes für ein elektronisches Zeitungsarchiv

Marc Sieburg

Aufgabensteller: Prof. Dr. Johann Schlichter

Betreuer: Dietrich Sauter
Institut für Rundfunktechnik

Abgabedatum: 14.05.1997

Ich versichere, daß ich diese Diplomarbeit selbständig verfaßt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. Mai 1997

(Marc Sieburg)

Zusammenfassung

Im Rahmen einer Kooperation des Bayerischen Rundfunks (BR) und des Süddeutschen Verlages (SV) sollen die bestehenden Archive beider Häuser in einem gemeinsamen digitalen Zeitungsarchiv zusammengefaßt werden. In diesem Zusammenhang soll ein Recherchearbeitsplatz konzipiert werden.

Um eine Grundlage für einen Lösungsansatz zu bekommen, wurden zunächst bestehende und mit der Arbeit im Zusammenhang stehende Archive untersucht.

Anschließend wurde eine Anforderungsanalyse für den neuen Recherchearbeitsplatz durchgeführt und die Erfolgsfaktoren für diesen ermittelt. Hierbei wurden besonders die Anforderungen der drei wichtigsten Anwendergruppen - Archivmitarbeiter, Redakteur und externe Kunden - sowie die technischen Rahmenbedingungen beim BR und SV berücksichtigt.

Als die wichtigsten Erfolgsfaktoren erwiesen sich die Erweiterbarkeit des Recherchearbeitsplatzes um bestehende und neue digitale Archive, eine einheitliche, leicht zu erlernende Oberfläche, auf der Crossrecherchen über mehrere Archive schnell und möglichst hochwertig durchgeführt werden können, sowie eine Anbindung des Systems an das Internet.

Auf der Anforderungsanalyse aufbauend wurde ein Lösungsansatz, bestehend aus einem Workflow, einer mehrschichtig, modular aufgebauten Systemarchitektur und einer Benutzerschnittstelle entworfen. Eine Brokerschicht dient zur Verteilung der Rechercheanfragen auf eventuell mehrere Datenbankübersetzer der Übersetzerschicht, die eine Anbindung der Datenbanken an das System ermöglichen.

Beim Entwurf der Benutzerschnittstelle wurde ein besonderes Augenmerk auf Übersichtlichkeit und einfache Bedienung gelegt. Hierzu soll die Verwendung von dem Anwender möglichst vertraute Standardkomponenten beitragen. Um den Anforderungen der verschiedenen Anwendergruppen gerecht zu werden, wurden verschiedene Benutzerprofile, sowie ein Anfänger-, Standard-, und Expertenmodus eingeführt. Durch die Suche über Attributfelder und Ranking soll die Treffermenge eingegrenzt und ein hoher Anteil relevanter Suchergebnisse erreicht werden.

Das Konzept des Recherchearbeitsplatzes wurde anschließend in einen Prototyp umgesetzt.

Im Rahmen dieses Prototyps wurden drei verschiedene Datenbanken angeschlossen. Die Internetsuchmaschine AltaVista, das hostbasierende Musikarchiv des Bayerischen Rundfunks MUSAD, sowie das ebenfalls hostbasierende Zeitungsarchiv des Süddeutschen Verlags REGIS.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Recherchen sind ein wesentlicher Bestandteil der journalistischen Arbeit. Dazu gehören auch häufig Recherchen in Zeitungen. Deshalb werden von vielen Rundfunkanstalten und Zeitungsverlagen umfangreiche Zeitungsarchive gehalten.

In diesen kann der Redakteur jedoch häufig nicht selbst recherchieren. Er gibt seine Recherche an einen Archivar weiter, der für ihn die Recherche durchführt und ihm eine Sammlung von Dokument zusammenstellt.

Die schnelle Verfügbarkeit von Informationen wird mehr und mehr zu einem entscheidenden Wettbewerbsfaktor für Medienunternehmen. Dies gilt besonders für den Nachrichtenbereich. Eine „Digitalisierung" der Archive bietet sich an. Ein weiterer Geschwindigkeitsgewinn wird erzielt, wenn Redakteure von ihrem Arbeitsplatz aus über einen PC auf die Archive zugreifen können. Besonders für Auslandskorrespondenten und in räumlicher Entfernung zum Archiv arbeitende Redakteure ist eine weltweite Verfügbarkeit der Archivdaten über das Internet interessant.

Im Rahmen einer Kooperation des Bayerischen Rundfunks (BR) mit dem Süddeutschen Verlag (SV) wird an einer Digitalisierung des Zeitungsarchivs gearbeitet. Es soll ein elektronisch abfragbares gemeinsames virtuelles Zeitungsarchiv entstehen.

Bei der Realisierung ist neben der Daten-Speicherungs-Seite auf der Seite der Benutzerschnittstellen zwischen zwei Arbeitsplätzen zu unterscheiden:

- Recherche im Archiv (Redaktion, Archivpersonal und externe Kunden)

- Eingabe ins Archiv

Ziel dieser Diplomarbeit ist es, die momentan angewandten Workflows und Vorgehensweisen bei der Recherche zu erfassen, die Anforderungen an einen neuen Recherchearbeitsplatz zu analysieren und darauf aufbauend ein Konzept für diesen zu entwerfen. Das erarbeitete Konzept soll schließlich noch exemplarisch in Form eines Prototyps für eine der aufgelisteten Benutzergruppen umgesetzt werden.

Kapitel 2 Aktuelle Zeitungsarchive beschreibt zunächst einige der mit der Arbeit im Zusammenhang stehende Archive.

Kapitel 3 Anforderungsanalyse erfaßt die Zielgruppen und die technischen Rahmenbedingungen eines Recherchearbeitsplatzes und ermittelt dessen Erfolgsfaktoren.

Kapitel 4 Lösungsansatz versucht die in Kapitel 3 beschriebenen Anforderungen in einem Konzept umzusetzen. Es beschreibt einen Workflow für einen digitalen Rerchercheearbeitsplatz, entwirft eine Systemarchitektur und beschreibt die Benutzerschnittstellen des Systems.

Kapitel 5 Realisierung eines Prototyps für einen Redaktionsarbeitsplatz beschreibt die technische Umsetzung der Systemarchitektur und der Benutzerschnittstelle in einen Prototypen für zukünftige Testrecherchen.

Kapitel 6 Bewertung und Ausblick bewertet die bisher gesammelten Erfahrungen mit dem Prototypen und gibt einen Ausblick auf den weiteren Verlauf des Projekts.

Kapitel 2

Aktuelle Zeitungsarchive

Dieses Kapitel stellt einige der im Augenblick bestehenden und mit der Arbeit im Zusammenhang stehenden Archive näher vor. Es erläutert ihren Aufbau und den typischen Ablauf einer Recherche in dem jeweiligen Archiv.

Folgende Archive wurden für die Betrachtung ausgewählt:

BR-Zeitungsarchiv

SZ-Zeitungsarchiv REGIS

SAD-Datenbanken des BR

Newsroomprojekt der BBC

2.1 Nicht digitalisiertes BR-Zeitungsarchiv

2.1.1 Aufbau

Das Zeitungsarchiv des Bayerischen Rundfunks ist ein Beispiel für ein klassisches Papierarchiv. Jeden Tag werden über 600 neue Zeitungsartikel und Agenturmeldungen im Rahmen von Zeitungsartikeln lektoriert und in das Archiv einsortiert. Die Gliederung erfolgt hierbei durch Themenmappen, wobei ein Artikel in mehrere verschiedene Themenmappen eingeordnet werden kann. In diesem Fall werden entsprechend Fotokopien von diesem Artikel angefertigt. Auf den Artikeln wird der Name der Quelle, das Erscheinungsdatum und die entsprechende Themenmappe als Signatur vermerkt. In den Mappen werden die Artikel nach dem Erscheinungsdatum abgelegt. Das aktuellste Dokument liegt hierbei an oberster Stelle. Bis ein Artikel allerdings endgültig einsortiert wird, oder eventuell auch eine neue Themenmappe für ihn angelegt wird, durchläuft er noch eine mehrstufige Qualitätssicherung, in der noch einmal seine Signatur, seine Formaldaten, seine Vollständigkeit und seine korrekte Einordnung überprüft werden.

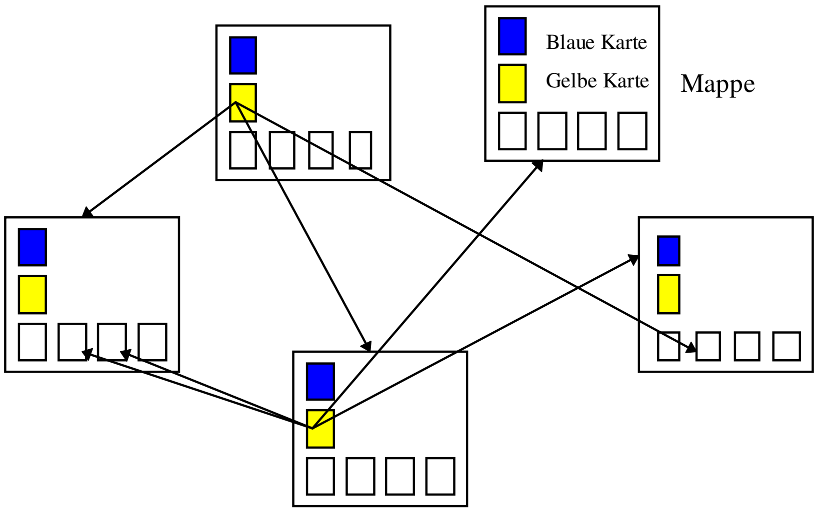

Abbildung 1: Aufbau des BR-Zeitungsarchivs

Die Themenmappen enthalten neben Zeitungsartikeln und Agenturmeldungen weiterhin noch eine gelbe und eine blaue Karte.

Auf der *gelben Karte* werden Verweise auf andere Themenmappen oder auch einzelne Artikel in anderen Mappen vermerkt, deren Inhalt einen Bezug zu der aktuellen Mappe hat (siehe Abbildung 1: Aufbau des BR-Zeitungsarchivs). Diese gelben Karten sind durchaus vergleichbar mit Hyperlinks in HTML-Dokumenten und stellen einen wichtigen Teil der inhaltlichen Erschließung des Archivs dar.

Die *blauen Karten* dienen zur Dokumentation der einzelnen Mappen und können eine kurze Inhaltsangabe, Hintergrundinformationen, und Ergebnisse häufig gestellter Fragen zu dem Thema der aktuellen Mappe enthalten. So wurden zum Beispiel auf der blauen Karte der Themenmappe „Parteitage der SPD" die Termine und Orte aller bisherigen SPD-Parteitage vermerkt.

2.1.2 Workflow einer Recherche

Erhält nun das Zeitungsarchiv des BR eine Anfrage, so sucht ein Archivar nach den ihm geeignet erscheinenden Themenmappen. Hier spielen Erfahrung, Kenntnis des Archivs und Kommunikationsfähigkeit des Archivars eine entscheidende Rolle für die Qualität der Recherche.

Wurde der Rechercheauftrag von einem Mitarbeiter aus dem Funkhaus, in dem sich auch das Archiv befindet, gestellt, so muß in der Regel der Mitarbeiter selbst im Archiv erscheinen. Dort erhält er die Möglichkeit in einem Leseraum die vom Archivar ausgewählten Mappen nach geeigneten Dokumenten durchzusehen und sich für ihn interessante Artikel zu fotokopieren. Artikel oder Mappen im Original auszuleihen ist nur in begründeten Ausnahmefällen möglich.

Stammt der Auftrag dagegen zum Beispiel aus dem räumlich entfernt gelegenen BR-Gelände in Freimann, so sucht der Archivar selbst einige der aussagekräftigsten Artikel heraus, fotokopiert diese und sendet sie anschließend per Fax an den Auftraggeber. Der Arbeitsaufwand einer Recherche für den Archivar ist in diesem Fall häufig erheblich. Für den Auftraggeber führt die Versandart per Fax zu einer relativ schlechten optischen Qualität der erhaltenen Dokumente.

2.2 Teilweise digitalisiertes SZ-Zeitungsarchiv REGIS

2.2.1 Aufbau

Im Zeitungsarchiv des Süddeutschen Verlages (SV) werden in dem System REGIS (regionales Informations-System) seit Mai 1991 die ersten Artikel der Süddeutschen Zeitung als Volltext digital archiviert. Seither wurde das System kontinuierlich erweitert. So werden seit Juli 1993 der Verfasser und formale Daten zu den einzelnen Artikeln, wie zum Beispiel Seite und Ressort mit erfaßt. Seit November 1993 steht das Munzinger-Personenarchiv in REGIS zur Verfügung und seit dem 20.6.1995 ist zusätzlich auch die Zeitschrift „Spiegel" im Volltext in REGIS enthalten. Ältere Artikel sind bisher nur über ein Papierarchiv recherchierbar, das in Mappen strukturiert ist.

Zusätzlich zur Speicherung der SZ-Artikel im Volltext und einer Reihe von Attributfeldern zu jedem Artikel, wird in REGIS eine Infoobjekt-Struktur aufgebaut. Sie soll die Themenmappen eines konventionellen Papierarchivs nachbilden.

Sie hat die Form eines in mehrere Ebenen hierarchisch gegliederten, ungerichteten Graphen. Auf der untersten Ebene befinden sich die Artikel, auf allen übrigen die Infoobjekte. Am Ende eines Weges durch den Graphen befinden in der Regel ein oder mehrere Artikel, die so in einem Infoobjekt zusammengefaßt sind.

In jeder Ebene, außer der untersten Infoobjektebene des Graphen ist es möglich, daß Infoobjekte auf untergeordnete Infoobjekte und zugleich auf Artikel zeigen.

Ein Artikel kann mehreren Infoobjekten, auch auf verschiedenen Ebenen, zugeordnet sein.

Die zwei wichtigsten Kriterien zur Gliederung der Infoobjekte sind die Merkmale Geographie und eine vierstellige Dezimalklassifikation. Steigt man von einem der Merkmale in den Graphen ein, so präsentiert er sich in seiner Struktur als Baum[1].

Eine vereinfachte Übersicht über die Struktur in REGIS gibt die Abbildung 2.

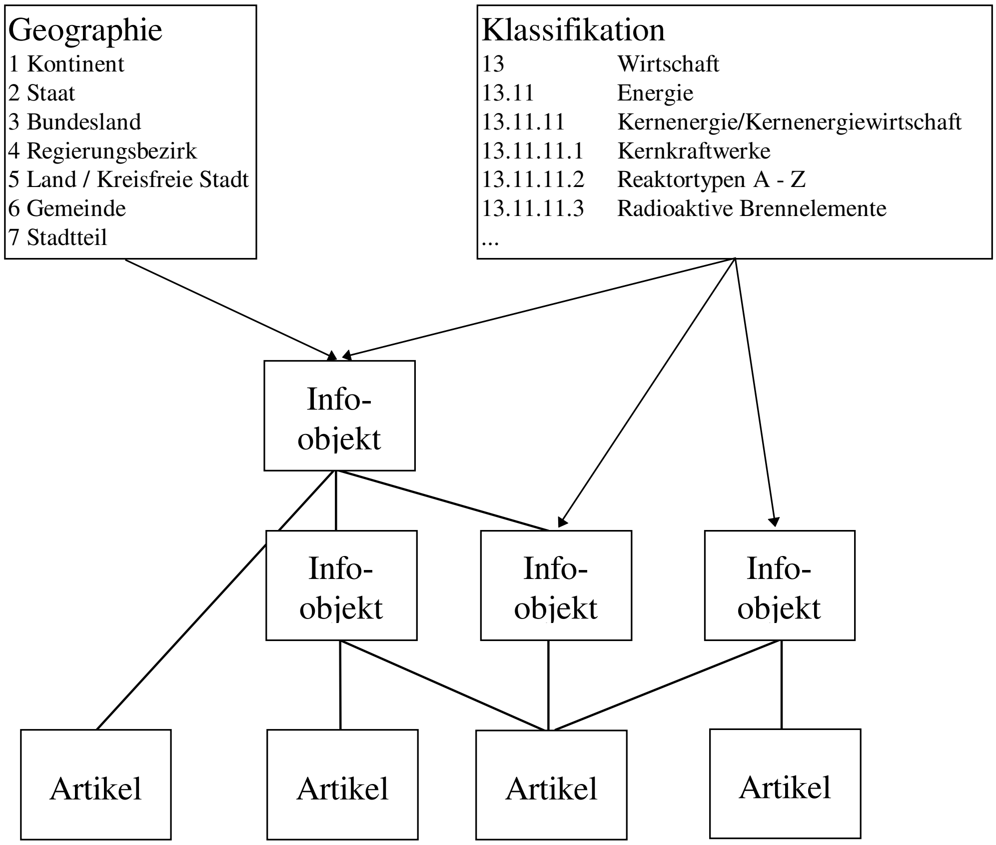

Abbildung 2: Vereinfachter Aufbau von REGIS

Ziel der Infoobjekt-Struktur ist es, mit ihrer Hilfe „bessere“ Rechercheergebnisse zu erreichen. „Besser“ heißt in REGIS, es werden inhaltlich relevante Dokumente gefunden, die mit einer Volltextrecherche alleine nicht gefunden worden wären.

Die Einordnung der Artikel zu den einzelnen Infoobjekten und das Ausfüllen der Attributfelder der Artikel erfolgt durch den Lektor. Eine korrekte Zuordnung ist

[1] Die Infoobjektstruktur könnte somit auch als mehrere übereinanderliegende Bäume betrachtet werden, die teilweise gemeinsame Knoten und Blätter besitzen. Bei einer Recherche über mehrere Merkmale mittels „und“-Verknüpfung, wird genau die Schnittmenge der dazugehörigen Bäume als Ergebnis geliefert, bei „oder“ ihre Vereinigung.

extrem wichtig, damit die in ihrer Realisierung sehr aufwendige Struktur später bei der Recherche tatsächlich bessere Ergebnisse als eine Volltextrecherche liefert und somit von den Anwender auch genutzt wird. Es wird also eine genaue Kenntniss des Lektors über die Archivstruktur benötigt und eine funktionierende Qualitätskontrolle bei der Erfassung. Aber auch der Anwender muß mit der Struktur dieser Objekte gut vertraut sein, um eine erfolgreiche Recherche über die Infoobjekte durchzuführen.

Zur Speicherung der Daten dient eine ADABAS-Datenbank auf einem IBM-Mainframe. Der Zugriff erfolgt über 3270-Terminals unter SNA. Die Benutzeroberfläche wurde in der Programmiersprache Natural entwickelt.

2.2.2 Workflow einer Recherche

Erhält das Zeitungsarchiv des SV eine Anfrage, so kann der Archivar zwischen drei Recherchemöglichkeiten in REGIS wählen:

- Recherche in Artikellisten
- Volltextrecherche / Recherche über Attributfelder
- Recherche in Info-Mappen

Bei der Recherche in Artikellisten erhält der Archivar eine ressortweise alphabetisch geordnete Liste aller gespeicherten Artikel. Zur Einschränkung der angezeigten Liste besteht die Möglichkeit, optional das Datum und das Ressort des gesuchten Dokuments anzugeben. Einzelne Artikel der Liste können markiert, angezeigt und ausgedruckt werden.

Unter Volltextrecherche wird in REGIS sowohl eine klassische Volltextrecherche[2], als auch eine Recherche über Attributfelder angeboten. Für die klassische Volltextrecherche steht das Suchkriterien „Artikeltext“ zur Verfügung. Zusätzlich können weitere, in der Regel durch den Lektor hinzugefügte, oder vom Produktionssystem übernommene Informationen, zum Beispiel unter den Suchkriterien „Überschrift“, „Ressort“, „Person“, „Region“, „Ersch.-Tag“ recherchiert werden.

Die Eingabe der Recherche erfolgt in die in Abbildung 3 dargestellte Maske. Sie stellt 14 Zeilen zur Verfügung, in die jeweils ein Suchkriterium und mehrere Suchwerte eingegeben werden können.

[2] also eine Suche über den Inhalt des gesamten Artikels

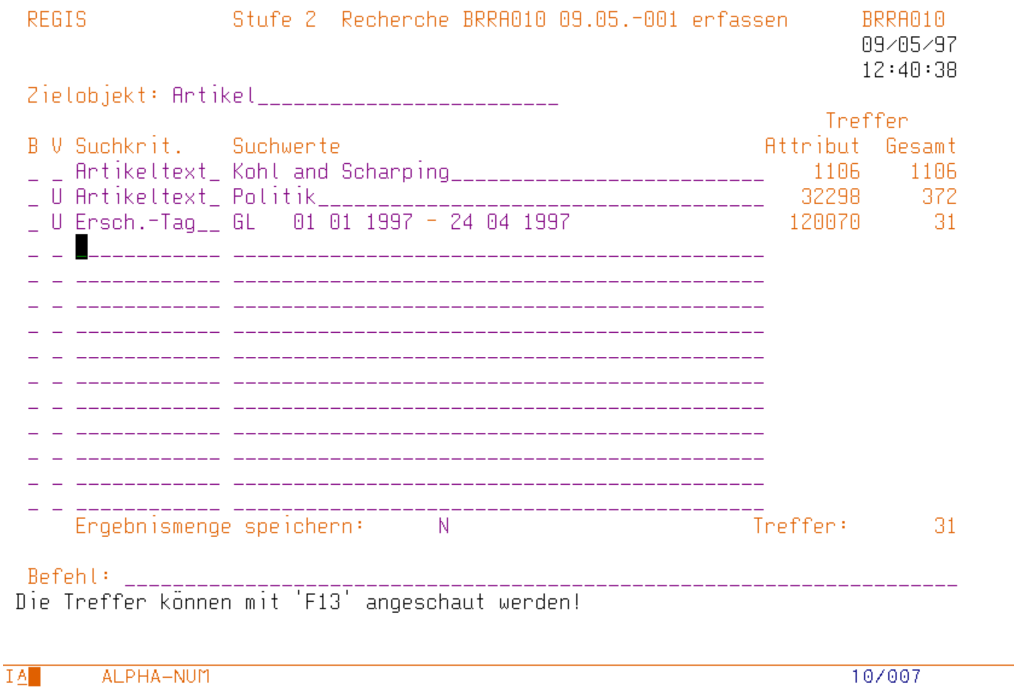

Abbildung 3: Recherchemaske in REGIS

Für die Verknüpfung mehrerer Suchwerte innerhalb einer Zeile stehen die Operatoren „and", „or", „not", „adj" zur Verfügung. Die Angabe der Verknüpfung zwischen den Zeilen erfolgt durch „u" für „and", „n" für „not", „o" für „oder" in der zweiten Spalte der Recherchemaske. Eine Klammerung ist nicht vorgesehen, allerdings kann jede Zeile als ein eingeklammerter Ausdruck gesehen werden. Innerhalb einer Zeile bindet der Operator „or" stärker als „and"[3]. Zwischen den Zeilen dagegen bindet „and" stärker als „or". „Adj" ist ein Abstandsoperator, < Jelzin *Adj5* Banken> liefert alle Artikel, in denen auf „Jelzin" innerhalb von fünf Wörtern „Banken" folgt.

Eine Unterscheidung zwischen Groß- und Kleinschreibung findet nicht statt.

Ein Suche mittels Sternoperator „*" ist möglich, allerdings kann er nur an das Ende eines Suchbegriffs angehängt werden. So ist zum Beispiel „Tele*" zulässig, „Tele*on" dagegen nicht.

Für die Recherche nach Begriffen mit unterschiedlicher Schreibweise stehen noch die Operatoren „syn" für Synonyme und „phonetic" für eine phonetische Suche zur Verfügung. < *syn* Telefon > liefert alle Artikel in denen „Telefon" oder „Telephon" vorkommt. < *phonetic* mirsajanow > findet zum Beispiel Mirsajanow, Mirzajanow, Mirsayanow, Mirzayanow.

---

[3] Begründet wurde dies von Verantwortlichen im Archiv damit, daß es mehr dem üblichen Sprachgebrauch der Mitarbeiter entspräche.

```
REGIS            Stufe 3    Recherche-Ergebnis                        BRRA010
                                                                      24/04/97
                                                                      12:52:54
Artikel                                                                   (31)
                                                                  Seite:     1
B Objektart       Objektname                              Sortierfolge:  DAT<
- --------------- -------------------------------------------------- --------
_ SZ-Artikel      Die besten Karten hätte Gerhard Schröder           08.04.97
                  Umfrage bei Würmtaler SPD-Politikern, welchen Kanz
_ SZ-Artikel      Die besten Karten hätte Gerhard Schröder           08.04.97
                  Umfrage bei Würmtaler SPD-Politikern, welchen Kanz
_ Spie-Artikel    Deutschland SPD: "Wir brauchen starke Nerven"      07.04.97
_ SZ-Artikel      Warten auf einen Gegenkandidaten für Kohl          05.04.97
                  Was Politiker im Landkreis München zur Entscheidun
_ SZ-Artikel      Warten auf einen Gegenkandidaten für Kohl          05.04.97
                  Was Politiker im Landkreis München zur Entscheidun
_ SZ-Artikel      Warten auf einen Gegenkandidaten für Kohl          05.04.97
                  Was Politiker im Landkreis München zur Entscheidun
_ Spie-Artikel    Deutschland KANZLERSCHAFT: Das Werkzeug der Histor 31.03.97
_ Spie-Artikel    DEUTSCHLAND PARTEIEN: "Schmutziges Geschirr"       24.03.97
_ SZ-Artikel      Die Hilflosigkeit der Politik                      22.03.97

Befehl: ______________________________________________________________________
Es gibt noch mehr anzuzeigen!

IA      ALPHA-NUM                                                  08/003
```

Abbildung 4: Trefferliste in REGIS

Die Treffermengen können in einer Liste (Abbildung 4) angesehen, einzelne Dokumente für einen späteren Ausdruck markiert oder sofort angezeigt werden.

Eine Recherche in Info-Mappen erfolgt analog zur Recherche nach Artikeln und in der selben Eingabemaske. Als Zielobjekt wird hierzu in der Maske „Info“ angegeben.

In diesem Suchmodus können mittels der Suchkriterien „Überordnung“ oder „Nachordnung“ die einem einzelnen Treffer in der Infoobjektstruktur direkt über- beziehungsweise untergeordneten Infoobjekte ermittelt werden.

Als Ergebnis erhält der Anwender Info-Mappen. Möchte er die dazugehörigen Artikel, so muß der die aktuelle Recherche abspeichern, als Zielobjekt „Artikel“ angeben und dort eine neue Recherche mit dem Suchkriterium „Recherche“ und dem Namen der Info-Mappenrecherche durchführen.

Bei der Recherche in Trefferlisten erhält der Anwender eine Liste aller gespeicherten Artikel. Eine Eingrenzung der Liste mittels Datum und Ressort ist möglich. Er kann durch die Liste blättern, sich einzelne Artikel anzeigen lassen (siehe Abbildung 5) und für eine spätere Weiterverarbeitung markieren.

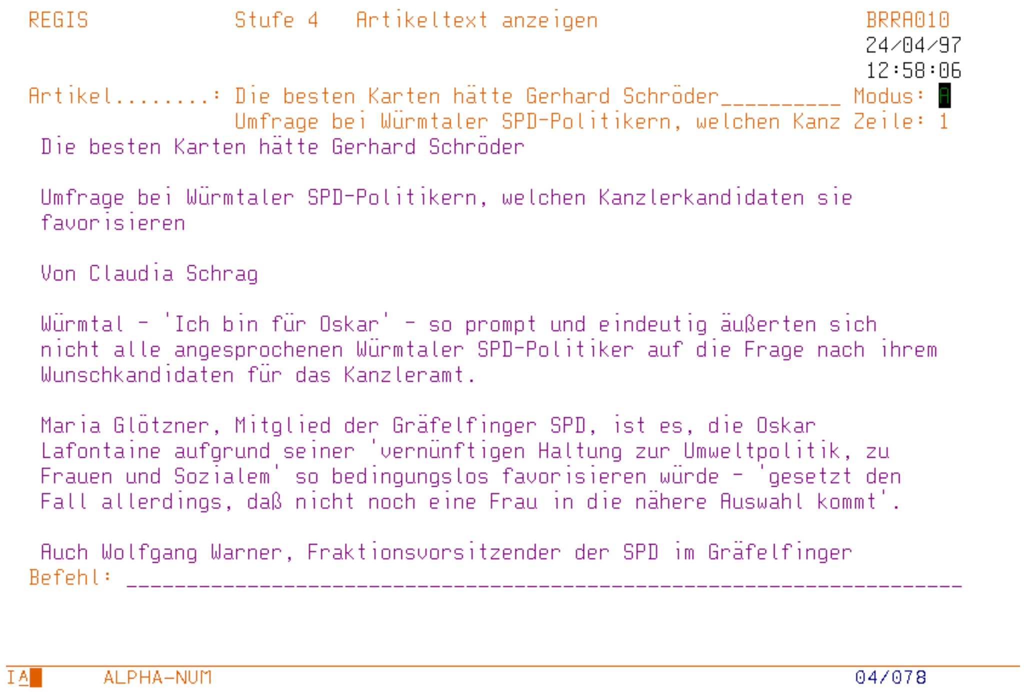

Abbildung 5: Artikelanzeige in REGIS

Die fertige Recherche wird ausgedruckt und dem Auftraggeber ausgehändigt. Info-Mappen, die sich noch im konventionellen Papierarchiv befinden und Artikel enthalten, die noch nicht digital gespeichert sind, werden ebenfalls an den Auftraggeber ausgehändigt, sofern er sich im selben Haus befindet wie das Archiv. Sie müssen aber wieder zurückgebracht werden. In diesem Fall muß sich das Archiv auf einen sorgfältigen Umgang des Auftraggebers mit den Archivdokumenten verlassen.

2.3 SAD-Datenbanken des BR

Beim Bayerischen Rundfunk finden sich eine Reihe von digitalen Archiven in Form der SAD-Datenbanken. Dazu gehören:

- FESAD Fernseh-Archiv- und Dokumentationssystem
- MUSAD Musik-Archiv- und Dokumentationssystem
- WOSAD Wort- und Schriftgut-Archiv- und Dokumentationsystem
- Venus Nachrichtenarchiv
- PRESTO Das Munzinger Personenarchiv

- Biblio Bibliotheksdatenbank des Bayerischen Rundfunks
- Piano Notenarchiv des Bayerischen Rundfunks

2.3.1 Aufbau

Bei allen Archiven handelt es sich um reine textbasierte Systeme, die (noch) keine Bild- oder Tondokumente liefern. Bei MUSAD, WOSAD oder FESAD[4] erhält der Anwender zum Beispiel neben einer Reihe von informativen Daten zu den einzelnen Dokumenten die Archivnummer, unter der das jeweilige Dokument im Archiv bestellt werden kann.

Zur Speicherung der Daten dient eine STAIRS[5]-Datenbank auf einem IBM-Mainframe. Der Zugriff erfolgt über ein 3270-Terminal unter SNA.

Aufgrund des ähnlichen Aufbaus der einzelnen SAD-Archive ist eine Crossrecherche über mehrere Archive möglich. Aus technischen Gründen ist die maximale Anzahl der gleichzeitig in einer Crossrecherche ansprechbaren Datenbanken jedoch auf 16 beschränkt. Dies kann eine gravierende Einschränkung darstellen, besonders wenn man berücksichtigt, daß allein MUSAD aus 4 Datenbanken besteht.

2.3.2 Workflow einer Recherche

Der Ablauf einer Recherche soll exemplarisch an MUSAD vorgeführt werden, eine Recherche in anderen SAD-Datenbanken, oder eine Crossrecherche erfolgt grundsätzlich analog.

[4] Eine Beschreibung von MUSAD, WOSAD, FESAD findet sich in [HUCK 89]
[5] Storage and Information Retrieval System, ein Datenbanksystem von IBM

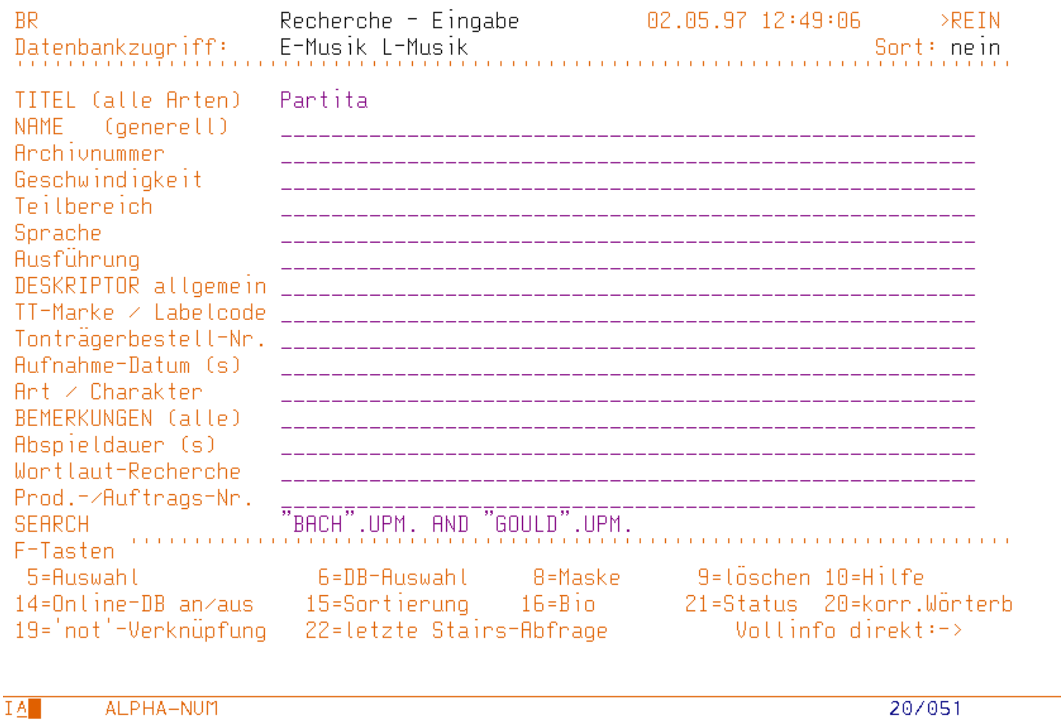

Abbildung 6: Beispiel einer Recherchemaske in MUSAD

Hat der Anwender das Werke-Archiv von MUSAD für seine Recherche ausgewählt, so erhält er die Möglichkeit, die zu recherchierenden Datenbanken zu bestimmen. Hierbei können mehrere Datenbanken angegeben werden. So kann der Anwender zum Beispiel gleichzeitig in der Werkdatenbank für ernste und leichte Musik recherchieren. Eine Veränderung dieser Auswahl zu einem späteren Zeitpunkt der Recherche ist möglich.

Für die Recherche stehen eine Reihe von Suchattributen, wie zum Beispiel *Person*, *Titel* oder *Sprache* zur Verfügung. Die einzelnen Attribute sind zeilenweise aufgeführt. Da wesentlich mehr Suchattribute zur Verfügung stehen, als Zeilen auf dem 3270-Terminal, kann der Anwender sich in einem Menü die benötigten Suchattribute für seine Recherche zusammenzustellen. Dabei besteht auch die Möglichkeit eine „SEARCH“-Eingabezeile auszuwählen, in die eine Suchanfrage in STAIRS[6] eingegeben werden kann. Ein Beispiel für eine Recherchemaske in MUSAD zeigt Abbildung 6.

[6] Eine einfache Anfragesprache des STAIRS-System. Ein Beispiel ist in Abbildung 6 zu sehen.

```
BR                    Recherche - Kurzinformation            02.05.97 12:46:26

   1   86/20583 000  S Partita für Klavier Nr. 6 e-moll,          38  23'15 EMUS
       79409           BWV 830
       CBS             aus: Klavierübung Teil I
                       Bach, Johann Seb / Gould, Glenn
   2   86/20582 000  S Partita für Klavier Nr. 4 D-dur,           38  25'15 EMUS
       79409           BWV 828
       CBS             aus: Klavierübung Teil I
                       Bach, Johann Seb / Gould, Glenn
   3   85/20598 000  S Partita für Klavier Nr. 2 c-moll,          38  15'20 EMUS
       77289           BWV 826                                     B
       CBS             Bach, Johann Sebastian
                       Gould, Glenn
   4   84/05695 000  S Partita für Klavier Nr. 1 B-dur,           38  11'40 EMUS
       77289           BWV 825
       CBS             aus: Klavierübung Teil I
                       Bach, Johann Seb / Gould, Glenn

F-Tasten
1=vor  2=rück   3=Druck          5=Auswahl   7=Recherche  9=lö      ====> KURZINFO
10=Hilfe       13=Panorama     14=Vollinfo alle          16=Bio   Kü+DFG=Vollinfo
Gefundene Dokumente:     7

IA      ALPHA-NUM                                                      03/002
```

Abbildung 7: Trefferliste in MUSAD

Die Treffermenge kann in einer Liste angesehen werden (Abbildung 7). Sie enthält neben dem Titel, der Bestellnummer, einer Takenummer[7], der Abspielzeit und einem Kürzel für die Datenbank in der das Dokument gefunden wurde, eine je nach Dokument bis zu drei Zeilen lange Kurzinformation.

Einzelne Dokumente können angewählt und in einer meist mehrseitigen Vollinformation (Abbildung 8) angesehen, ausgedruckt oder auch, falls es die Rechte des Anwenders zulassen, korrigiert werden. Gesichtete Dokumente werden in der Trefferliste gekennzeichnet. Es besteht auch die Möglichkeit direkt in der Vollinformation durch die Treffermengen zu blättern.

Mit Hilfe der in MUSAD recherchierten Daten kann der Anwender die gewünschten Tonträger beim Archiv bestellen.

[7] Dies ist die Nummer des Dokuments auf dem zu einer Bestellnummer gehörenden Tonträger. Es können durchaus mehrere hundert Tondokumente auf einem Tonträger, zum Beispiel ein Tonband oder eine CD, gespeichert sein.

```
 BR  MUSIK-DB   Recherche: Vollinformation           Seite 01 von 02  >RVOL01
Archiv-Nr: 86/20583 000     Produktions-Nr. i000755          02.05.97 12:48:02
Obertitel  : Partita Nr. 4 6 e-moll, BWV 830
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

LC-0000    : CBS                                   instrumental  38  Stereo
Bestell-Nr: 79409
E-Musik / Solistische Musik  / Partita

KomponistIn   :     Bach, Johann Sebastian (1685-1750)

Titel (DBTI)  :     Partita für Klavier Nr. 6 e-moll, BWV 830            23'15
Satztitel     :     1. Satz: Toccata                                      9'50
                    2. Satz: Allemande                                    2'05
                    3. Satz: Courante                                     2'20
                    4. Satz: Air                                          0'40
                    5. Satz: Sarabande                                    3'40
                    6. Satz: Menuet I - Menuet II                         0'50
                    7. Satz: Gigue                                        3'30
Über.Werktitel:     aus: Klavierübung Teil I

F-Tasten '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
3=rück          4=Tonträger          5=Auswahl     6=Restinfo       ====> RVOL01
7=Recherche   12=Kurzinformation   16=Bio        17=nächstes Dokument
18=vorhergehendes Dokument         DFG=weiter

IA       PROTECTED                                                  01/001
```

Abbildung 8: Dokumentanzeige in MUSAD

2.4 Newsroomprojekt der BBC

2.4.1 Aufbau

Das sich seit 1994 in der Entwicklung befindende „Electronic News and Production System“ ENPS der BBC steht zum Zeitpunkt der Ausarbeitung der Diplomarbeit kurz vor der Einführung. Im Endausbau soll es auf über 6000 PCs installiert sein und allen journalistischen Mitarbeitern der BBC weltweit zur Verfügung stehen. Die PCs sind über Server untereinander vernetzt. Ein Server soll ungefähr 50 Clients bedienen. Als Protokoll wird TCP/IP verwendet.

ENPS wurde speziell für Journalisten entwickelt und soll innerhalb weniger Sekunden Suchanfragen auf einem im System enthalten Archiv beantworten. Dieses Archiv speichert Nachrichten mehrerer Nachrichtenagenturen für einen Zeitraum von ca. 90 Tagen.

Das System soll für einen Journalisten intuitiv erlernbar sein. Ziel ist ein Schulungszeitraum von unter zwei Tagen. Hierzu wurde versucht, eine auf das wesentliche reduzierte, für alle Anwender gleiche Oberfläche mit sehr geringen

Konfigurationsmöglichkeiten[8] zu entwerfen. Sie enthält auf der linken Seite ein (nicht verschiebbares) Fenster mit einem Inhaltsverzeichnis. Rechts unten sind drei Icons (Drucker, Mailbox, Papierkorb) plaziert. Oben ist ein zweizeiliges Feld zur Anzeige aktueller Nachrichten und Mails angebracht. Der übrige Bereich dient zur Anzeige der Ergebnisse. Hier können Texte sowie Audio und Videodaten angezeigt werden. Ein Schnittsystem soll in Zukunft integriert werden.

Die Benutzeroberfläche beruht auf einem selbst entworfenen graphischen Fenstersystem. Es stellt unter DOS eine graphische Oberfläche mit Mausbedienung und „Drag & Drop“-Fähigkeiten zur Verfügung. Dies bringt jedoch einige Nachteile mit sich, so ist das System zum Beispiel nicht in einem Web-Browser lauffähig, neue Fähigkeiten eines modernen Betriebsystems können nicht genutzt werden und der Anwender kann Erfahrungen mit der Bedienung von Standardsoftware nicht immer ohne weiteres auf das Newssystem übertragen.

2.4.2 Workflow für die News

Möchte der Redakteur eine Recherche nach bestimmten Nachrichtenmeldungen durchführen, so erhält er eine Recherchemaske, in der ihm neben einer Volltextrecherche in Überschriften- und Nachrichtentext noch mehrere Zeitfilter zur Verfügung stehen. Weiterhin ist noch eine Auswahl der Nachrichtenagenturen und einiger Nachrichtenkategorien, wie zum Beispiel „international News“ oder „Economics / Business“ möglich.
Die Ergebnisse einer Recherche werden im Inhaltsverzeichnis auf der linken Seite des Bildschirms angezeigt. Einzelne Dokumente können geladen, editiert und anschließend in verschiedenen Bereichen abgelegt werden. Insgesamt können bis zu 40 Dokumente lokal vorgehalten werden.

[8] Journalisten sollen auf diese Weise durch einfache Positionsangabe ein Bedienelement definieren und so Erfahrungen und Tips zur Bedienung austauschen können.

Kapitel 3

Anforderungsanalyse

Dieses Kapitel analysiert die Anforderungen an einen Recherchearbeitsplatz für ein digitales Archiv. Dazu werden zunächst die potentiellen Zielgruppen des Archivs untersucht. Anschließend wird die bestehende und zu erwartende Hard- und Softwareumgebung der Recherchearbeitsplätze analysiert. Darauf aufbauend erfolgt schließlich die Herleitung der Erfolgsfaktoren für den Recherchearbeitsplatz.

3.1 Zielgruppen

Grundsätzlich lassen sich zwei verschiedene Zielgruppen feststellen, die der Anwender und die der System-Administratoren und Systementwickler. Für letztere sei hier nur erwähnt, daß eine leichte Wartbarkeit des Systems eine der wichtigsten Anforderungen darstellt. Genauer wird hierauf in Kapitel 3.3.7 eingegangen.

In diesem Abschnitt beschränkt sich die Betrachtung auf die Zielgruppe der Anwender. Die drei wichtigsten Zielgruppen sind hierbei:

- Redakteure
- Archivare
- externe Kunden

3.1.1 Redakteure

Die Zielgruppe der Redakteure ist äußerst heterogen bezüglich der vorhandenen Computerkenntnisse, der verwendeten Hardwareplattformen, der eingesetzten Betriebssysteme und der zu recherchierenden Daten. So wird zum Beispiel ein

Musikredakteur häufiger Daten in der Form benötigen, wie sie zur Zeit MUSAD bietet. Ein Nachrichtenredakteur benötigt eher aktuelle Agenturmeldungen und eine Recherche für eine Dokumentation greift häufiger auf Daten zurück, wie sie das BR-Zeitungsarchiv oder REGIS bieten.

Für den zu entwerfenden Recherchearbeitsplatz muß davon ausgegangen werden, daß ein Redakteur häufig nur unzureichend mit der Struktur der seiner Recherche zugrunde liegenden Datenbanken vertraut ist. Zur Zeit lauten zum Beispiel häufige Anfragen im BR-Zeitungsarchiv: „Geben Sie mir alles über das Thema XY“. Eine solche Anfrage würde bei einer Onlinerecherche häufig zu einer unüberschaubaren Menge an Treffern führen. Dies muß vom System unbedingt abgefangen, beziehungsweise durch geschickte Benutzerführung, soweit möglich, schon bei der Eingabe vermieden werden.

Um eine hohe Akzeptanz der Anwender zu erreichen, ist eine möglichst einfache, leicht zu erlernende Oberfläche, ein wesentlicher Erfolgsfaktor für den Recherchearbeitsplatz. Wird ein Redakteur häufiger durch komplexe Benutzerführung, lange Wartezeiten oder unbefriedigende Rechercheergebnisse frustriert, so wird er immer schneller dazu tendieren, die Recherchen nicht mehr selber durchzuführen, sondern sie gleich im Archiv in Auftrag zu geben.

3.1.2 Archivmitarbeiter

Zum Zeitpunkt der Einführung des Systems kann bei Archivaren nicht unbedingt von besseren Computerkenntnissen als bei Redakteuren ausgegangen werden. Durch die häufigere Nutzung des Systems sollten sie jedoch sehr schnell auch mit einer komplexeren Rechercheoberfläche arbeiten können. Aufgrund der meist sehr guten Kenntnisse der Archivare bezüglich der ihrer Recherche zugrunde liegenden Daten wird von vielen eine möglichst professionelle und vielseitige Oberfläche, die auch komplizierte Spezialanfragen erlaubt, explizit gewünscht.

Erfahrungen mit bestehenden Datenbanken haben jedoch gezeigt, daß zu komplexe Recherchemöglichkeiten, die sehr großes Hintergrundwissen des Recherchierenden erfordern, häufig nur wenig genutzt werden. Viele Archivare tendieren in diesen Fällen eher zu einer Volltextrecherche und nutzen dadurch die häufig mit sehr großem Aufwand in das Archiv implementierten Zusatzinformationen nur unzureichend.

3.1.3 Externe Kunden

Das Benutzerprofil kann hier, mangels empirischer Daten, noch nicht genauer bestimmt werden. Es ist jedoch anzunehmen, daß es dem eines Redakteurs relativ ähnlich ist. Im Vordergrund stehen hier sehr stark juristische, sicherheits- und abrechnungstechnische Fragen. So ist es den Pressearchiven im Augenblick in der Regel nicht gestattet, Daten nicht hauseigener Produktionen an externe Kunden weiterzugeben. Weiterhin sollen und dürfen häufig einige hausinterne Informationen zu einzelnen Datensätzen nicht nach außen weitergegeben werden. Eine genaue Spezifikation sämtlicher Zugriffsrechte wird somit für das Recherchesystem unabdingbar.

3.2 Hard- und Softwareumgebung

3.2.1 Arbeitsplatzrechner

Die wohl im Augenblick am häufigsten verwendete Plattform ist der Windows-PC. Ein Umstieg auf Windows 95 ist häufig noch nicht erfolgt. Einige Unternehmen planen im Augenblick dieses System zu überspringen und ihre Rechner auf NT 4.0 umzustellen. Es werden weiterhin Arbeitsplatzrechner auf MAC-OS und UNIX-Basis eingesetzt.

Der zu entwerfende Recherchearbeitsplatz muß mittelfristig sicherlich alle diese Systeme unterstützen also möglichst plattformunabhängig sein um Mehrfachentwicklungen zu vermeiden.

3.2.2 Netzwerke

Im Bayerischen Rundfunk wie im Süddeutschen Verlag besitzen die meisten Arbeitsplatzrechner noch keinen TCP/IP-Anschluß an ein hausinternes Netz. Häufig bestehen in den einzelnen Abteilungen einzelne nicht miteinander verbundene Netze auf Basis der unterschiedlichsten Protokolle. Eine hausinterne Vernetzung dieser Rechner auf TCP/IP-Basis ist jedoch fest geplant und kann als Voraussetzung für das Recherchesystem angenommen werden. Der Zugang zum Internet erfolgt im Augenblick an den meisten Arbeitsplätzen noch über Modem und einen Provider wie zum Beispiel Compuserve.

3.2.3 Datenbanken

Viele der bestehenden Datenbanken wie zum Beispiel REGIS oder die SAD-Datenbanken besitzen im Augenblick nur einen 3270-Zugang über SNA. Der Zugriff auf diese Datenbanken erfolgt bisher über ein 3270-Terminal oder eine 3270-Terminalemulation über das SNA-Protokoll. Bei einigen Datenbanken ist ein direkter Zugang über TCP/IP noch in diesem Jahr geplant.

Eine neu zu gründende Datenbank für die Archivkooperation wird vermutlich auch eine JDBC, ODBC, OLE oder Web-Schnittstelle zur Verfügung stellen und über TCP/IP zu erreichen sein.

3.3 Erfolgsfaktoren für den Recherchearbeitsplatz

Die im folgenden Abschnitt aufgeführten Erfolgsfaktoren für einen Recherchearbeitsplatz beruhen auf einer Umfrage bei Redakteuren - sie wurde bereits vor Beginn der Diplomarbeit vom IRT durchgeführt - und auf Gesprächen mit Mitarbeitern der Archive des Bayerischen Rundfunks, des Süddeutschen Verlags und der Deutschen Welle.

Als die wichtigsten Erfolgsfaktoren haben sich dabei ergeben:

- leicht zur erlernende einheitliche Oberfläche
- schnelle Antwortzeiten
- hohe Qualität der Recherche
- Einbindung bestehender oder neuer digitaler Archive
- Anbindung an das Internet
- Zugriffskontrolle, Copyright
- leichte Wartbarkeit
- Erweiterbarkeit als Konzept

Auf die einzelnen Punkte soll nun in den folgenden Unterabschnitten genauer eingegangen werden.

3.3.1 Leicht zu erlernende und einheitliche Oberfläche

Die Rechercheoberfläche sollte möglichst einfach und intuitiv zu bedienen sein. Ziel ist die Schulungszeiten auch für Redakteure mit nur relativ geringen Computerkenntnissen unter vier Stunden zu halten. Nach dieser Zeit sollte ein Redakteur in der Lage sein, einfache Recherchen selbständig durchzuführen. Eine möglichst einfache und übersichtliche Benutzeroberfläche, bei der ein Überladen mit Funktionalität vermieden wird, ist daher äußerst wichtig. Weiterhin ist bei dem Entwurf zu berücksichtigen, daß viele Arbeitsplätze noch mit einer Bildschirmauflösung von 800 x 600 Pixel arbeiten. Wenn in diesem Fall bei der

Eingabe häufiges Scrollen nötig wird oder wichtige Bedienelemente nicht mehr sichtbar sind, so stört dies den Arbeitsablauf unter Umständen erheblich.

Die verwendeten Bedienelemente sollten sich soweit möglich an bekannte Standards halten, so daß dem Anwender ihre Funktionalität bereits aus anderen Anwendungen vertraut ist. Auf jeden Fall muß dem Benutzer ein umfangreiches Online-Hilfesystem zur Verfügung stehen, das ihm die Funktion jedes Bedienelements erklärt und ihm mit Beispielen weiterhilft.

Für den Einsteiger sollte ein spezieller *Anfängermodus* zur Verfügung stehen, der ihn Schritt für Schritt durch die Recherche führt und ihm mit Erläuterungen und Beispielen zur Seite steht. Allerdings muß auch hierbei darauf geachtet werden, daß dieser Hilfemodus nicht mit Information überfrachtet wird und so einen Anfänger eher irritiert als unterstützt. Sobald der Anwender sich mit der Oberfläche vertraut fühlt muß er den Assistenten abschalten und in den *Standardmodus* wechseln können.

In diesem Modus findet der Benutzer eine einheitliche Oberfläche für alle zur Verfügung stehenden Datenbanken vor. Daß in diesem Fall nicht auf allen Datenbanken gleichzeitig optimal recherchiert werden kann, ist aufgrund der häufig sehr unterschiedlichen Archivstrukturen sehr wahrscheinlich, wird jedoch zugunsten einer einfachen Bedienbarkeit in Kauf genommen werden. Der Rechercharbeitsplatz muß so konzipiert werden, daß eine möglichst optimale Umsetzung der Recherche auf die einzelnen Archive gewährleistet wird. Berücksichtigt man an dieser Stelle, daß die meisten Informationen in einem Archiv in der Regel weit mehr als einmal gespeichert sind, so ist an dieser Stelle anzunehmen, daß für die meisten Standardrecherchen ein für den Recherchierenden durchaus befriedigendes Ergebnis erzielt wird, auch wenn nicht alle Treffer in den einzelnen Archiven erfaßt werden. Eine tatsächliche Lieferung *aller* Informationen zu einem Thema, würde im Gegenteil den Benutzer eher überfordern, als ihm nützlich sein. Inwieweit diese Annahmen korrekt sind, wird sich allerdings erst im Laufe von Testrecherchen am Prototyp herausstellen.

Unterstützend können noch Benutzerprofile zum Einsatz kommen, die einen Redakteur zum Beispiel noch zusätzlich mit für sein Sachgebiet relevanter Information, wie zum Beispiel Einblendung aktueller Nachrichtenmeldungen versorgen. Dies kann beispielsweise als Nachrichtenticker auf der Oberfläche oder als Bildschirmschoner realisiert werden.

Sollte ein Redakteur mit der Standardoberfläche nicht zu dem gewünschten Ergebnis kommen, so muß es ihm möglich sein, seine Anfrage an einen Experten im Archiv, zusammen mit einem Kommentar über sein gewünschtes Thema, weiterzuleiten. Falls er bereits zu den fortgeschrittenen Anwendern zählt, auf einem speziellem Archiv recherchieren will und eine persönliche

Zugangsberechtigung zu diesem Archiv besitzt, kann er zusätzlich auch direkt im *Expertenmodus* auf dieses Archiv zugreifen. Für MUSAD würde ihm dann zum Beispiel in der Rechercheoberfläche ein 3270-Terminalemulator zur Verfügung gestellt, oder eine speziell für dieses Archiv entworfene Eingabemaske. Weiterhin sollte es im Expertenmodus auch über die Standardoberfläche möglich sein, komplexere Anfragen in boolscher Logik einzugeben.

Eine hohe Konfigurierbarkeit der Oberflächengestaltung durch den Anwender im Anfänger oder Standardmodus wurde von vielen Seiten als eher kontraproduktiv eingeschätzt. In der Vergangenheit wurde die Erfahrung gemacht, daß viele Anwender ihre Erfahrungen und Tips zur Benutzung eines Programmes durch einfache Ortsangaben der entsprechenden Bedienelemente miteinander austauschen. Auch für die EDV-Beratung wird die Arbeit durch einheitliche Oberflächen erleichtert.

3.3.2 Schnelle Antwortzeiten

Auch wenn bei einem digitalen Archiv die Antwortzeiten auf eine Recherche meist erheblich kürzer sind, als in einem klassischen Papierarchiv, so ist die Bereitschaft, längere Wartezeiten an einem Rechner in Kauf zu nehmen, sehr viel geringer. Schnelle Antwortzeiten sind ja gerade eine der Stärken eines digitalen Archivs. Daher muß diesem Kriterium bei der Konzeption des Arbeitsplatzes ausreichend Beachtung geschenkt werden.

Zum einen sollte eine hohe Netzbelastung vermieden werden. Da in modernen Client-Server basierten Netzwerken auf der Clientseite meist ausreichend Rechenleistung vorhanden ist, kann dies unter anderem erreicht werden, indem möglichst viele Aufgaben lokal auf dem Rechner des Anwenders erledigt werden. Wichtig ist zusätzlich die zu übertragenden Datenmengen möglichst gering zu halten. Daher sollten zwischen den an einer Recherche beteiligten Rechnern möglichst nur für die Ausführung der Recherche notwendige Daten ausgetauscht werden. Durch lokales Caching auf den einzelnen Rechnern kann hier sehr viel Netzlast eingespart werden. Speziell bei den in zukünftigen Archiven auftretenden Multimedia-Daten, wie Musik, Filme, Bilder, Faksimile, um hier einige zu nennen, ist auf ein kompaktes Datenformat zu achten. Hier muß mit Sicherheit ein Kompromiß zwischen der Anforderung an höchste Qualität und kürzeste Übertragungszeit gefunden werden.

Weiterhin sollte ein häufiger Verbindungsaufbau zwischen den Rechnern vermieden werden. Bei jedem Verbindungsaufbau, speziell über Weitverkehrsnetze, entstehen in der Regel Verzögerungen, die sich zu einer spürbaren Wartezeit aufsummieren können.

3.3.3 Hohe Qualität der Recherche

An dieser Stelle muß unterschieden werden zwischen der inhaltlichen und der optischen Qualität des Rechercheergbnisses.

Als inhaltlich hochwertig ist eine Recherche anzusehen, wenn sie in kompakter Form alle vom Anwender gewünschten Informationen enthält. Der Anteil der relevanten Dokumente in der Treffermenge soll möglichst hoch sein. Im Idealfall enthält sie alle inhaltlich relevanten und keine falschen Dokumente[9].

In der Regel wird man jedoch bei einer erfolgreichen Anfrage mehr als nur die notwendigen Artikel erhalten. Vor allem bei eher unpräzisen Suchanfragen kann die Treffermenge regelrecht explodieren. Jedoch kann auch eine Treffermenge, die nur aus inhaltlich relevanten Dokumenten besteht, eine für den Anwender unzumutbare Größe annehmen. An dieser Stelle müssen von dem Recherchesystem Sicherheitsmechanismen eingebaut werden, die den Nutzer vor solchen riesigen Ergebnismengen und den damit häufig verbundenen Wartezeiten schützen. Dies kann durch einfache Beschränkung der maximalen Trefferanzahl geschehen, auch wenn hierbei der Verlust von Information in Kauf genommen werden muß. Diese eher als Notbremse zu bezeichnende Art der Eingrenzung der Treffermenge kann jedoch durchaus noch zu einem vernünftigen Ergebnis führen. Speziell, wenn die Treffermenge vorher noch nach bestimmten Qualitätskriterien sortiert wird (Ranking). Möglichkeiten des Rankings sind zum Beispiel Aktualität des Dokuments oder Häufigkeit der Suchbegriffe in den einzelnen Dokumenten[10].

Sollte eine solche Kürzung der Treffermenge erfolgt sein, ist der Anwender davon durch entsprechende Hinweise in der Trefferliste in Kenntnis zu setzen. Insgesamt ist eine solche Kürzung einer zu großen Treffermenge vorzuziehen, die, speziell von dem Anfänger oder Standardanwender, nicht mehr überblickt werden kann und bei diesem eher zu einer Ablehnung des Systems führt.

Durch geschickte graphische Aufbereitung und Strukturierung der Trefferlisten können hier für den Anwender größere Treffermengen, wie sie speziell bei Crossrecherchen über mehrere Archive häufiger vorkommen, sehr viel überschaubarer gehalten werden. Hier bieten sich Baumstrukturen an, wie sie zum Beispiel aus vielen Dateimanagern bekannt sind.

Eine weitere Eingrenzung der Treffermenge erfolgt noch durch Übernahme semantischer Information aus der Anfragemaske in die Anfrage an die Datenbank. Sind Eingabefelder nach speziellen inhaltlichen Kriterien, wie zum Beispiel die

[9] Eine ausführliche Behandlung des Themas findet sich in [Frakes, Baeza-Yates 92]

[10] Eine weiteres Sortierkriterium kann zum Beispiel die Zugriffshäufigkeit auf ein bestimmtes Dokument sein. Einige Redakteure greifen in konventionellen Papierarchiven, wenn sie vor einer großen Auswahl von Dokumenten stehen, gerne zu abgegriffenen Dokumenten. Sie vermuten dort inhaltlich „relevantere" Dokumente. Dieses Auswahlkriterium läßt sich durch eine Sortierung nach Zugriffshäufigkeiten nachbilden.

Unterscheidung in Personen und Schlagwörtern, unterteilt, so kann dies, falls dies von den entsprechenden Datenbanken unterstützt wird, zu einer deutlichen Eingrenzung der Treffermenge und einem inhaltlich wesentlich aussagekräftigeren Ergebnis führen.

Eine optisch hochwertige Qualität der Treffer ermöglicht dem Anwender wesentlich schneller, den Inhalt eines Dokuments zu erfassen. Häufig sucht der Anwender bei seinen Recherchen ein ganz spezielles Dokument, das er schon anhand seiner äußeren Form sofort wiedererkennt. Bei Textdokumenten kann hier Faksimile sehr hilfreich sein. Bei Faksimiledokumenten ist jedoch auf eine für den Bildschirm abgestimmte Auflösung zu achten. Graustufen oder Farben können durch Antialiasing-Effekte die Lesbarkeit erheblich erhöhen, führen jedoch meist auch zu einem deutlich höheren Datenvolumen. Bei reinem Schwarz-Weiß-Faksimile kann der Text, speziell wenn ungeschickte Auflösungen gewählt werden, extrem unleserlich werden. In solchen Fällen ist ein optisch aufbereiteter Fließtext vorzuziehen. Als weitere Alternative für Textdokumente sei hier noch Quasifaksimile[11] erwähnt, das bei relativ geringem Datenvolumen dem Anwender ebenfalls eine hervorragende optische Qualität der Daten liefern kann.

3.3.4 Einbindung bestehender oder neuer digitaler Archive

Im Rahmen der ARD und des Süddeutschen Verlages bestehen bereits mehrere digitale Archive, deren Aufbau und Pflege mit enormen Investitionen verbunden war. Eine Anbindung dieser Archive wird schon aus rein wirtschaftlichen Gründen von den beteiligten Unternehmen gewünscht. Außerdem sind ihre Inhalte eine große Bereicherung und teilweise sogar unverzichtbar für ein Recherchesystem im Medienumfeld. Einige der Archive werden auch noch weiter ausgebaut, so daß beispielsweise in den Audio-Archiven[12] mittel- bis langfristig nicht nur Informationen zu den archivierten Daten sondern zusätzlich auch das Audio-Dokument selbst über das Netz geliefert werden kann. Ähnliches ist sicherlich auch bei Bildern und Filmen zu erwarten. Bei einigen Archiven wird voraussichtlich jedoch kaum noch etwas an den Schnittstellen zu diesen Archiven geändert, so daß der Zugriff auf diese Systeme von dem neuen Recherchesystem über bestehende Zugänge, wie zum Beispiel 3270-Terminalemulationen erfolgen muß. Dies muß für den Standardanwender im Rahmen der einheitlichen Benutzeroberfläche verborgen bleiben. Für den Experten soll jedoch ein direkter 3270-Zugang über das Recherchesystem ermöglicht werden.

---

[11] Quasifaksimile ist ein optisch aufbereiteter (Fließ-) Text. Er soll für den Anwender (fast) wie ein Faksimile erscheinen. Es werden dem Original ähnliche Fonts und eine ähnliche Formatierung verwendet. Der tatsächliche Zeilenumbruch kann jedoch Unterschiede zum Original aufweisen.

[12] Dazu gehört zum Beispiel MUSAD

Der Zugriff auf kommerzielle, kostenpflichtige Datenbanken von externen Firmen über dieses System soll für einige Anwender möglich sein. Hierbei sind Budgets und Abrechnungssystem für die einzelnen Anwender zu implementieren. Hierbei sind auch Sicherheitsmechanismen gegen unbeabsichtigte zu teure Recherchen und unberechtigte Zugriffe wichtig.

Weiterhin ist im Rahmen der Archivkooperation mit dem Aufbau eines neuen digitalen Zeitungsarchivs zu rechnen, dessen Aufbau, Schnittstellen und Dokumentenformat erst nach dem Abschluß dieser Arbeit festliegen werden.

Somit ist die Möglichkeit, bestehende und neue digitale Archive in das Recherchesystem mit aufzunehmen, ein wesentlicher Bestandteil des zu entwerfenden Konzepts.

3.3.5 Anbindung an das Internet

Bei der Internet-Anbindung müssen zwei Richtungen betrachtet werden. Zum einen eine Anbindung des Recherchesystems an das Internet und umgekehrt eine Anbindung des Internets an das Recherchesystem.

Der erste Fall ist zum Beispiel interessant, um Auslandskorrespondenten weltweit, Mitarbeitern von Zuhause aus, oder externen Kunden gegen Bezahlung einen Zugriff auf das Recherchesystem zu ermöglichen. In diesen Fällen kommt Zugriffskontrollen und verschlüsselter Datenübertragung eine besondere Rolle zu, um unfreundliche Angriffe oder Abhören des Datenstroms zu vermeiden. Insbesondere bei externen Kunden entstehen Copyrightprobleme. Auf sie wird im nächsten Abschnitt detaillierter eingegangen. Zusätzlich muß in diesem Fall ein Abrechnungssystem implementiert werden.

Eine Anbindung des Internets und Internetdatenbanken wie zum Beispiel AltaVista an das Recherchesystem kann durchaus eine große Bereicherung der Recherchemöglichkeiten für den Anwender darstellen. Im Hochschulbereich ist zum Beispiel das Internet als Informationsquelle kaum noch wegzudenken. Aus diesem Grund ist sie auf jeden Fall im Recherchesystem mit zu berücksichtigen, auch wenn eine tatsächliche Anbindung aufgrund fehlender Sicherheitskonzepte in vielen unternehmensinternen Netzen noch nicht möglich ist.

3.3.6 Zugriffskontrolle, Copyright

In der Regel ist es nicht jedem Anwender in gleicher Weise gestattet in allen Archiven zu recherchieren. Zum Beispiel müssen bei externen, kostenpflichtigen Archiven Zugriffsbeschränkungen möglich sein. So muß für jeden Anwender oder jede Anwendergruppe eine Kennung und ein Paßwort bestehen, das vom

Recherchesystem verwaltet wird und anhand derer es die Zugriffsrechte zu den einzelnen Archiven bestimmen kann. Hinzu kommt, daß nicht jedes Dokument, das recherchiert werden darf, gleichzeitig auch ausgedruckt, weiterverarbeitet oder veröffentlicht werden darf. Dies hat seinen Ursprung in teilweise sehr komplizierten Copyrightbestimmungen. Erschwerend kommt hier noch dazu, daß in der Archivkooperation Bayerischer Rundfunk Süddeutscher Verlag ein öffentlich-rechtliches und ein privates Unternehmen zusammenarbeiten. Hier müssen Juristen praktikable Lösungen finden. Von dem zu entwerfenden Recherchesystem ist diese rechtliche Problematik durch Zugriffskontrolle in Abhängigkeit vom jeweiligen Anwender, nicht nur auf die einzelnen Datenbanken, sondern auch auf der Dokumentenebene zu berücksichtigen. Hierzu muß zum Beispiel die Export-, Speicher- oder die Druckfunktion bei einzelnen Dokumenten abgeschaltet werden, beziehungsweise müssen entsprechende Hinweise auf die Nichtveröffentlichbarkeit eines Dokumentes hinzugefügt werden.

Verstärkt tritt diese Copyrightproblematik bei dem Verkauf von Recherchediensten an externe Kunden oder dem Anbieten dieser Dienste im Internet auf. Hier müssen Informationen, die nur für den hausinternen Gebrauch bestimmt sind, aus den einzelnen Dokumenten herausgefiltert werden. Bei kostenpflichtigen Diensten muß ein entsprechendes Abrechnungssystem implementiert werden.

3.3.7 Leichte Wartbarkeit

Betrachtet man die bisher gestellten Anforderungen an das Recherchesystem so gelangt man schnell zu der Einsicht, daß leichte Wartbarkeit kein triviales Problem mehr ist und bereits im Konzept mit verankert werden muß. So müssen die Verwaltung von Benutzerlisten, Zugriffsbeschränkungen und die Zugänge zu den Datenbanken hier ebenso berücksichtigt werden, wie ein leichtes Updating einzelner Komponenten des Systems. Dies alles soll natürlich während des Betriebs und ohne dessen Unterbrechung geschehen. Weiterhin müssen Cacheverwaltung, Nutzungsstatistiken, Nachrichten an den Anwender und Pflege des Hilfesystems auf einfache Art und Weise möglich sein.

3.3.8 Erweiterbarkeit als Konzept

Das in dieser Arbeit konzipierte Recherchesystem wird auch nach seiner Inbetriebnahme einem permanenten Veränderungsprozeß unterworfen sein. Es kommen neue Datenbanken hinzu, oder alte werden entfernt. Eine Erweiterung um neue Dokumenttypen oder Profildienste muß - mit vertretbarem Aufwand - möglich sein. Weiterhin soll das Konzept eine möglichst große Zukunftssicherheit bieten. Dies alles kann nur erreicht werden, wenn die Möglichkeit zur Erweiterbarkeit bereits im Konzept mit verankert ist. Proprietäre, nicht auf Standards beruhende Insellösungen sollten ebenso vermieden werden, wie die Verwendung vermeintlicher Standardsoftware, deren spätere Erweiterung gar nicht oder nur unter großem Kostenaufwand möglich ist.

Kapitel 4

Lösungsansatz

Im folgenden Kapitel wird versucht, einen Lösungsansatz zu beschreiben, der die im vorhergehenden Kapitel aufgeführten Anforderungen möglichst umfassend erfüllen soll.

Dazu wird zunächst der Workflow einer Recherche beschrieben und anschließend eine Systemarchitektur entworfen, mit deren Hilfe die Realisierung des gewünschten Recherchesystems möglich wird. Abschließend folgt der Entwurf einer Benutzerschnittstelle, die den Bedürfnissen der einzelnen Anwendergruppen gerecht werden soll.

4.1 Workflow für eine Recherche

Grundsätzlich besteht zwischen dem Ablauf einer Recherche beim einem Archivar, einem Redakteur und einem externen Kunden kein großer Unterschied. Die Unterschiede finden sich überwiegend in den Zugriffsrechten und Benutzerprofilen der einzelnen Anwendergruppen. Daher gilt der in diesem Abschnitt beschriebene Workflow, sofern dies nicht ausdrücklich anders vermerkt ist, für alle Anwendergruppen des Recherchearbeitsplatzes. Eine Übersicht über den im folgenden geschilderten Ablauf einer Recherche zeigt Abbildung 9.

Zu Beginn seiner Recherchen muß der Benutzer sich mittels seiner Kennung und seines Paßwortes beim System anmelden (1)[13]. Anhand dieser Kennung ermittelt das System das Benutzerprofil und die Zugriffsrechte des Anwenders auf die an das System angeschlossenen Suchmaschinen.

[13] Die in Klammern gesetzten Ziffern beziehen sich auf die entsprechenden Nummern in Abbildung 9.

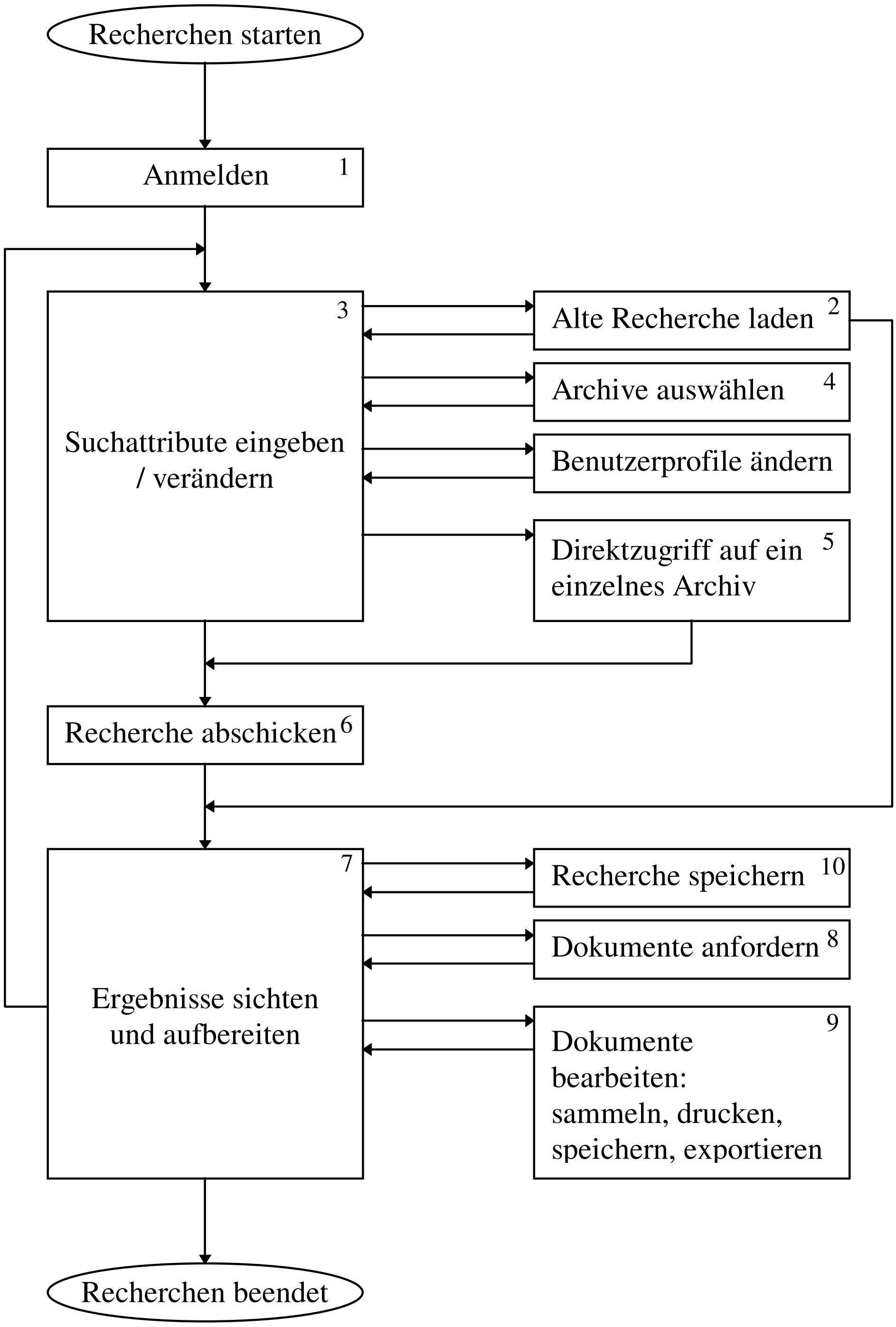

Abbildung 9: Workflow einer Recherche

Er erhält nun die Möglichkeit, eine neue Recherche zu formulieren, oder eine bereits bestehende Recherche zu laden (2) und zu verändern. Dies kann zum Beispiel eine zu einem früheren Zeitpunkt gespeicherte Recherche oder ein von einem Redakteur an einen Archivmitarbeiter übermittelter Rechercheauftrag sein.

Für die Eingabe der Recherche erhält der Anwender eine für alle Suchmaschinen einheitlich Recherchemaske, in die er die für seine Recherche notwendigen Suchbegriffe eingeben oder verändern kann (3). Die Recherchemaske kann je nach Anwendergruppe Unterschiede aufweisen[14].

Weiterhin kann der Benutzer aus der Liste der ihm zur Verfügung stehenden Archive einzelne Archive für die aktuelle Recherche aktivieren beziehungsweise deaktivieren (4).

Wenn es die Zugriffsrechte erlauben und sich der Anwender bei seiner Recherche auf ein Archiv beschränken möchte, hat er zusätzlich die Möglichkeit direkt, das heißt mit einer speziell für dieses Archiv entworfenen Oberfläche, auf ihm zu recherchieren (5).

Eine fertig formulierte Recherche kann der Anwender an das System abschicken (6). Sie wird dann vom System bearbeitet und die Ergebnisse werden als Trefferlisten, versehen mit einigen Kurzinformationen angezeigt und können vom Benutzer gesichtet und aufbereitet werden (7).

Er erhält hierzu die Möglichkeit, einzelne Dokumente anzufordern (8), diese zu sammeln, zu strukturieren, und falls es seine Zugriffsrechte zulassen, diese auch zu drucken und zur Weiterverarbeitung zu speichern, oder zum Beispiel in ein Produktionssystem zu exportieren (9).

Weiterhin kann er eine Recherche zur späterem Verwendung abspeichern oder zum Beispiel als Archivar die Ergebnisse an den Auftraggeber weiterleiten (10).

Sollte eine Recherche nicht die gewünschten Ergebnisse geliefert haben, so kann sie modifiziert und erneut abgeschickt oder versehen mit einem Kommentar an einen Spezialisten gesendet werden.

4.2 Systemarchitektur

In diesem Abschnitt wird das Konzept für die Systemarchitektur zunächst in der Übersicht dargestellt. Anschließend wird in den Unterabschnitten detailliert auf die einzelnen Komponenten und ihre Funktion eingegangen.

[14] Hierauf wird in Kapitel 4.3 genauer eingegangen

4.2.1 Übersicht

Ziel der im folgenden vorgestellten Architektur ist eine vollständige Entkoppelung der Rechercheoberfläche von den darunterliegenden Archivsystemen.

Der Anwender muß auf mehreren verschiedenen Archiven simultan recherchieren können, ohne mit der Struktur dieser Recherchesysteme vertraut zu sein. Trotzdem soll dem Benutzer ein möglichst hochwertiges Rechercheergebnis angeboten werden. Das heißt, die Trefferanzahl muß überschaubar bleiben, die Trefferlisten entsprechend graphisch aufbereitet werden, die Treffer sollten möglichst aussagekräftig sein und die Antwortzeiten müssen kurz bleiben.

Ein fortgeschrittener Anwender muß, wenn er dies wünscht, auch direkt auf einzelne spezielle Datenbanken, zum Beispiel über eine 3270-Terminalemulation, oder eine speziell für diese Datenbank entworfene Eingabemaske, zugreifen können.

Die Systemarchitektur gliedert sich in fünf logische Schichten (Abbildung 10):

- Benutzerschnittstelle
- Anfrageobjekt
- Broker
- Übersetzer
- Archivsysteme

Davon sollten, abhängig von der für die Realisierung eingesetzten Technik, zum Beispiel die ersten beiden Schichten lokal auf dem Rechner des Benutzers, der Broker auf dem entsprechenden Web-Server und die Übersetzer auf den Gateways zu den einzelnen Archivsystemen, oder einem Rechner der in einer netzwerktechnisch möglichst günstigen Verbindung damit steht, ausgeführt werden.

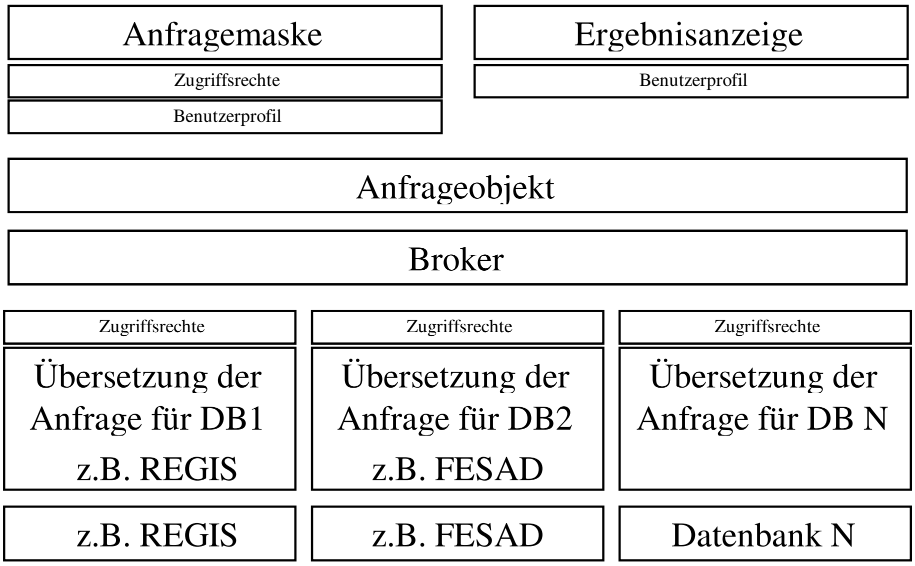

Abbildung 10: Systemarchitektur Übersicht

Es wird versucht die anfallende Netzlast auf ein Minimum zu reduzieren, indem die einzelnen Module möglichst viele Aufgaben lokal erledigen, häufigen Verbindungsaufbau vermeiden, die notwendig zu übertragenden Datenmengen gering halten, Caching einsetzen und Flaschenhälse vermeiden.

Ein weiterer wichtiger Aspekt dieser Architektur ist, daß durch seine hohe Modularität und klar definierten Schnittstellen, leichte Erweiterbarkeit, Wartbarkeit und Skalierbarkeit bereits, soweit möglich, im Konzept verankert sind.

Im Folgenden werden die einzelnen Module in ihrem Aufbau und ihrer Funktionsweise genauer vorgestellt

4.2.2 Anfrageobjekt

Das zentrale Objekt auf Benutzerseite ist das Anfrageobjekt, es regelt die Kommunikation mit der Benutzerschnittstelle und dem Broker, enthält als Daten die Anfrage und die Trefferlisten. Es läßt sich speichern und wieder laden. Somit besteht für den Benutzer, die Möglichkeit eine Anfrage lokal, oder falls dies technisch nicht möglich ist, über den Broker zu speichern und zu einem späteren Zeitpunkt wieder zu bearbeiten. Auch läßt sich eine Anfrage von einem Redakteur zur weiteren Bearbeitung an einen Archivmitarbeiter weiterleiten, z.B. wenn eine Anfrage nicht die gewünschten Ergebnisse geliefert hat.

4.2.3 Anfragemaske

Abbildung 11: Objekt Anfragemaske

Das Anfrageobjekt ruft, beim Login unter Prüfung der Zugriffsrechte und der Benutzerprofile das Objekt Anfragemaske auf. Dieses enthält als Daten die Anfrage. Sie kann vom Benutzer verändert werden, oder er kann eine neue Anfrage eingeben. Diese wird dann an das Anfrageobjekt gesendet, das eine Verbindung zum Broker aufbaut und die Anfrage dorthin weiterleitet.

Die Anfragemaske und die Ergebnisanzeige sind die einzigen Objekte, die eine für den Benutzer sichtbare Repräsentation auf dem Bildschirm besitzen. Alle übrigen Objekte dieser Architektur sind für den Benutzer nicht direkt wahrnehmbar.

4.2.4 Broker

Das Brokerobjekt enthält eine Liste der für den Benutzer, in Abhängigkeit von seinen Zugriffsrechten, zur Verfügung stehenden Datenbankübersetzer. An diese leitet es die vom Anfrageobjekt erhaltene Anfrage parallel weiter und erhält von ihnen die Ergebnisse als Trefferliste in Form einer Kennung, mit der das jeweilige Dokument angefordert werden kann, eines Titels und einer kurzen (ca. zweizeiligen) Inhaltsangabe zu dem jeweiligen Dokument. Die Liste der Übersetzer muß jederzeit während des Betriebes ohne großen Aufwand veränderbar sein.

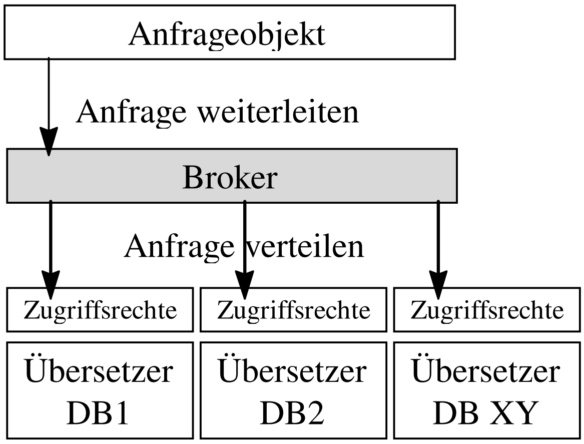

Abbildung 12: Objekt Broker

Die Kommunikation mit den einzelnen Übersetzern erfolgt über voneinander unabhängige Threads. Falls neue Treffer vorliegen, erfolgt eine Aktualisierung der Trefferliste beim Anfrageobjekt durch den Broker in sinnvollen Zeitabständen. Auf diese Weise werden dem Anwender schon Ergebnisse angeboten, wenn die ersten Archive Treffer liefern und nicht erst nach dem Abschluß der Recherche auf allen ausgewählten Archiven.

4.2.5 Übersetzer

Das Übersetzerobjekt empfängt die Anfrage vom Broker, prüft die Zugriffsrechte und übersetzt die Anfrage in eine für das darunterliegende Datenbanksystem verständliche Form. Hierbei ist auf eine semantisch möglichst schlüssige Umsetzung der Anfrageattribute auf die von der jeweiligen Datenbank zur Verfügung gestellten Attribute zu achten. Ein Veränderung der Umsetzung der einzelnen Attribute, muß jederzeit leicht während des Betriebes möglich sein.

Das Übersetzerobjekt übernimmt die gesamte Kommunikation mit der zu ihm gehörenden Datenbank. Hierbei ist zu beachten, daß es sich bei der Kommunikation mit bereits bestehenden Archivsystemen eventuell nicht mehr um die bisher verwendeten Kommunikationsstandards handelt. Als Beispiel sei hier ein 3270-Zugang über SNA erwähnt.

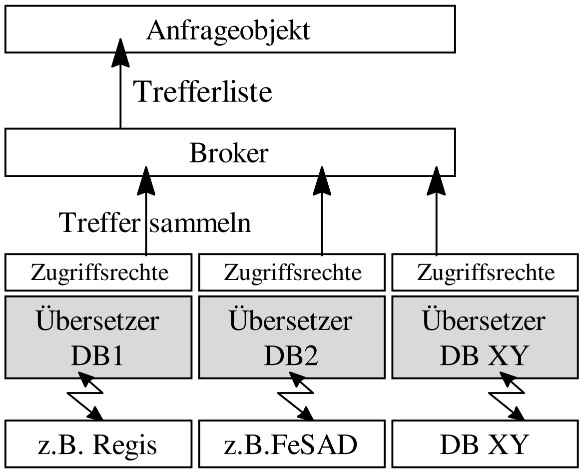

Abbildung 13: Objekt Übersetzer

Die von der Datenbank erhaltenen Treffer leitet der Übersetzer an den Broker weiter, der sie von den einzelnen Übersetzern sammelt und an das Anfrageobjekt weiterleitet.

4.2.6 Ergebnisanzeige

Das Anfrageobjekt leitet die Trefferliste an das Objekt Ergebnisanzeige weiter.

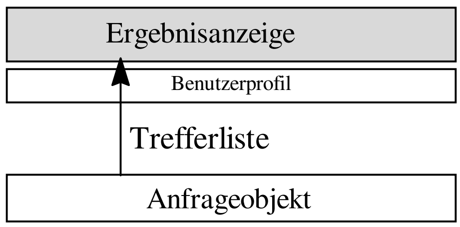

Abbildung 14: Objekt Ergebnisanzeige

Dort werden die Trefferlisten strukturiert und sortiert in einem Trefferbaum[15] angeboten. Über die Ergebnisanzeige kann der Benutzer einzelne Dokumente von den Archiven anfordern. Dazu wird analog zur Recherche eine Verbindung über das Anfrageobjekt zum Broker aufgebaut, der dann, über den Übersetzer der entsprechenden Datenbank, das Dokument anfordert. Es ist jedoch auch möglich, daß bei einigen Dokumenten, zum Beispiel von einem Web-Server mittels einer

[15] eine Beschreibung des Trefferbaums erfolgt in Kapitel 4.3.2

HTML-Adresse, das entsprechende Dokument direkt von der Ergebnisanzeige angefordert werden kann.

Durch Caching der Dokumente auf Broker, Übersetzer oder Anwenderebene kann die Arbeit teilweise noch erheblich beschleunigt und die Netzlast entsprechend reduziert werden.

4.3 Benutzerschnittstelle

Die Benutzerschnittstelle ist für den Anwender in der Regel die einzige sichtbare Schicht des Recherchesystems und wird von ihm häufig mit diesem identifiziert. Die tiefer liegenden Schichten der Systemarchitektur werden dem Standardanwender meist verborgen bleiben. Er sollte seine Recherchen ohne genaue Kenntnis dieser Architektur durchführen können. Dies gilt nicht unbedingt für den professionellen Rechercheur im Archiv, der eventuell sein Wissen über die Struktur der Archive und den Aufbau des Systems für seine Suche ausnützen möchte.

4.3.1 Prinzipielle Entwurfskriterien

Um dem Anwender eine leicht zu erlernende und ergonomische Oberfläche zur Verfügung zu stellen, sind folgende Kriterien bei dem Entwurf der einzelnen Komponenten der Rechercheoberfläche besonders berücksichtigt worden:

Alle wichtigen Bedienelemente sind schnell „auf einen Blick" zu erkennen, die Oberfläche wird nicht mit Funktionalität überladen. Die Wege, die mit der Maus zur Bedienung der Oberfläche zurückgelegt werden müssen, werden möglichst kurz gehalten, die Anzahl der notwendigen Aktionen seitens des Anwenders zur Benutzung der Oberfläche wird möglichst gering gehalten. Dazu gehört auch, daß Scrolling der Oberfläche, speziell bei der Eingabe der Recherche vermieden wird. Diese Kriterien sollen ab einer Bildschirmauflösung von 800 x 600 Punkten erfüllt werden.

In Anlehnung an die Philosophie des Newsroom-Projekts der BBC wird auf eine zu hohe Konfigurierbarkeit der Oberfläche seitens des Anwenders im Anfänger und Standardmodus verzichtet.

Die Aufteilung des Bildschirms erfolgt in drei Bereiche. Vorzugsweise auf der linken Seite befindet sich der Trefferbaum, oben eine Funktionsleiste. Der größte Bereich ist zur Anzeige der Anfragemaske beziehungsweise der

Rechercheergebnisse vorgesehen (siehe Abbildung 17 auf Seite 43 und Abbildung 18 auf Seite 47).

Eine Ähnlichkeit zu vielen bekannten Dateimanagern und Browsern wurde hier bewußt gesucht, um dem Anwender eine möglichst vertraute und daher leichter zu erlernende Oberfläche anzubieten. Durch die Verwendung vertrauter Standardelemente wird eine intuitive, schnell zu erlernende Bedienung möglich.

Für den fortgeschrittenen Anwender, dem eine Bedienung der Oberfläche mit der Maus in manchen Fällen zu umständlich erscheinen mag, ist zusätzlich eine Bedienung der Oberfläche mittels Tastenkommandos vorgesehen.

4.3.2 Trefferbaum

Der Trefferbaum steht sowohl während der Eingabe einer Recherche als auch bei der Sichtung der Suchergebnisse zur Verfügung.

Während der Eingabe einer Anfrage dient er zur Darstellung und Auswahl aller recherchierbaren Archive. Es werden nur solche Archive angezeigt, die mit den aktuellen Benutzerrechten auch tatsächlich recherchierbar sind. Die Deaktivierung oder Aktivierung einzelner Archive ist durch einfachen Mausklick auf das entsprechende Archivsymbol möglich. Es muß auf eine deutlich sichtbare Markierung des ausgewählten Status geachtet werden.

Abbildung 15: Darstellung der recherchierbaren Archive

Durch Doppelklick auf ein Archiv kann der Anwender, wenn er entsprechende Zugangsberechtigungen besitzt, direkt auf diesem recherchieren. Dies kann über eine speziell für dieses Archiv entworfenen Suchmaske erfolgen. Es ist auch möglich, speziell für den professionellen Anwender, an dieser Stelle die Originaloberfläche des jeweiligen Recherchesystems, wie zum Beispiel ein 3270-Terminal zur Verfügung zu stellen. Falls in der allgemeinen Suchmaske bereits Einträge vorhanden sind, wäre eine Übernahme dieser Einträge in die datenbankspezifische Maske wünschenswert.

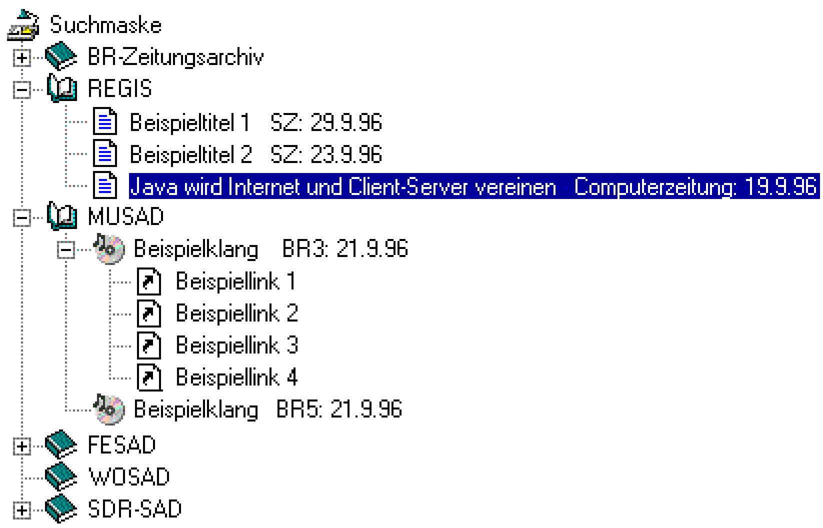

Abbildung 16: Trefferbaum nach einer erfolgreichen Recherche

Während eine Suchanfrage vom System bearbeitet wird, ist für den Anwender anhand der Darstellung der Archiveinträge im Trefferbaum[16] erkennbar, von welchen Archiven bereits Ergebnisse vorliegen, und welche sich noch in der Bearbeitung befinden. Sobald die ersten Suchergebnisse vorliegen, können diese über den Trefferbaum bearbeitet werden.

Die Treffermengen werden hierarchisch nach Archiven sortiert. Eine optionale Sortierung nach Quellen, Datum oder ähnlichen Kriterien ist ebenfalls möglich. Der Anwender kann sich , durch auf- und zuklappen einzelner Unterbäume, nur einzelne Teile der Trefferliste anzeigen zu lassen. Einträge im Trefferbaum, hinter denen sich nicht sichtbare Unterbäume verbergen, sind durch ein „+“-Zeichen vor dem entsprechenden Eintrag gekennzeichnet. Durch einfachen Klick auf das „+“-Zeichen läßt sich der entsprechende Unterbaum aufklappen und durch einen Klick auf das dann erscheinende „-“-Zeichen wieder zuklappen.

Um eine bessere Übersicht über die Trefferlisten zu erhalten und häufiges Scrolling zu vermeiden, klappt, falls sich der Mauszeiger über einem nur teilweise sichtbaren Baumeintrag befindet, der nicht sichtbare Anteil des Eintrags über die Begrenzungen des Trefferbaumes hinaus auf.

Verharrt der Mauszeiger einige Zeit, zum Beispiel eine Sekunde, über einem Eintrag, so erscheint unter ihm in einem Popupfenster einige Zusatzinformation.

[16] Dies kann zum Beispiel erreicht werden, indem Archive, die sich noch in der Bearbeitung befinden, grau dargestellt werden.

Dies kann bei einer Datenbank Information zu diesem Archiv sein, bei einem Dokument zum Beispiel der Untertitel oder eine kurze, circa zweizeilige Inhaltsangabe.

Die Auswahl eines Dokuments erfolgt durch Doppelklick auf den entsprechenden Eintrag. Der Trefferbaum fordert dann über das Anfrageobjekt das Dokument an und stellt es im Anzeigebereich dar. Tondokumente werden, falls vorhanden, über den Audiokanal des Rechners ausgegeben. Zusätzlich wird der Eintrag im Trefferbaum als gelesen markiert. Der Anwender kann Dokumente im Trefferbaum kennzeichnen, die ihm besonders wichtig erscheinen. Der Dokumentstatus, wie zum Beispiel „normal", „wird geladen", „gelesen" oder „wichtig", kann durch verschiedene Icons im Trefferbaum dargestellt werden.

Eine Rückkehr in die Suchmaske ist jederzeit durch Anwahl des Eintrags „Suchmaske" im Trefferbaum möglich. Der aktuelle Trefferbaum bleibt jedoch bestehen, bis entweder eine neue Anfrage vom Anwender abgeschickt, die alte Anfrage gelöscht, oder die Anfragemaske durch Doppelklick im Trefferbaum aufgerufen wird. Bevor die Trefferlisten aus dem Suchbaum entfernt werden, muß eine Dialogbox erscheinen, die den Anwender auf den Verlust der Rechercheergebnisse hinweist und ihm die Möglichkeit gibt, die aktuelle Recherche abzuspeichern oder die Aktion abzubrechen.

4.3.3 Funktionsleiste

In der Funktionsleiste werden einige wichtige Funktionsknöpfe angeboten. Ein Überladen dieser Leiste mit zu vielen Buttons, wie es zum Beispiel bei vielen Office-Systemen unter Windows heute üblich ist, wird jedoch vermieden.

Auf jeden Fall enthalten sind ein Button zum Speichern und Laden einer Recherche. Weiterhin ein Button zum Drucken des Dokuments, ein Hilfebutton und ein Button zum Einstellen von Optionen, wie zum Beispiel den Benutzermodus oder verwendete Schriftgrößen.

Für jeden Knopf steht ein Hilfetext zur Verfügung, der automatisch erscheint, wenn der Mauszeiger etwas länger über einem Knopf verweilt. Diese „Bubblehelp"-Funktion läßt sich über das Optionen-Menü abschalten. Falls eine Statuszeile vorhanden ist, erscheint zusätzlich eine Kurzinformation, die sofort die Funktion des Buttons erklärt, wenn sich die Maus über ihm befindet.

Bei der Darstellung von Faksimile-Dokumenten, Bildern oder Videos können zusätzlich zwei Knöpfe zum Vergrößern und Verkleinern des Dokumentes, oder alternativ ein Popup mit verschiedenen Darstellungsgrößen in Prozent, angeboten werden.

4.3.4 Suchmaske

Hier sind gemäß der Anforderungsanalyse drei verschiedene Benutzermodi vorgesehen. Dies sind der Anfänger-, Standard- und der Expertenmodus. Im Folgenden wird zunächst auf den Standardmodus eingegangen und anschließend werden die Besonderheiten des Anfänger- und Expertenmodus erläutert.

Standardmodus

Bei dem Entwurf einer einheitlichen Suchmaske für mehrere zum Teil sehr unterschiedlich aufgebaute Archive ist es kaum möglich eine alle Anwender zufriedenstellende Lösung zu finden. So wird es in den meisten inhaltlich aufbereiteten Datenbanken wie zum Beispiel REGIS oder MUSAD immer Funktionalität geben, die von der Anfragemaske für den Standardanwender nicht abgedeckt wird. Umgekehrt würde es bei einer sehr komplexen Suchmaske meist Suchattribute geben, die auf einigen Archiven nicht umgesetzt werden können. Es ist also nicht Aufgabe der Suchmaske, alle verwendeten Datenbanken optimal abzubilden, sondern ein leicht zu bedienendes und trotzdem vielen einfachen Recherchen genügendes Werkzeug sein.

Die einfachste und von vielen Crossrecherchesystemen im Internet verwendete Möglichkeit wäre, nur eine Volltextrecherche anzubieten. Sie ist leicht auf die meisten Datenbanken zu übertragen. Dies wäre eine eher unbefriedigende Lösung, da hierbei ein Großteil der häufig mit hohem finanziellen Aufwand in die Archive implementierten Zusatzinformation verloren ginge. Erfahrungen[17] haben gezeigt, daß eine reine Volltextsuche bei sehr großen Archiven häufiger zu einer sehr großen unübersichtlichen Treffermenge führt, als eine Suche über inhaltlich von einem Lektor aufbereitete Zusatzinformationen. So würden zum Beispiel bei einer Anfrage nach „*Kohl*" sowohl Artikel mit dem Politiker, als auch eine Menge von Kochrezepten geliefert. Ein weiteres, häufig in diesem Zusammenhang vorgetragenes Beispiel ist der Satz: „*Herr Langer fährt mit dem Golf zum Golf an den Golf.*" Bei einer reinen Volltextrecherche ist es dem System nicht möglich zwischen den verschiedenen Wortbedeutungen von „*Golf*" zu unterscheiden. In vielen Zeitungsarchiven wird versucht diese Problematik durch Lektorierung der Artikel zu umgehen. In konventionellen Papierarchiven ist dies sogar unverzichtbar. Ein nicht thematisch korrekt eingeordnetes Dokument gilt dort als unwiederbringlich verloren. Es sei angemerkt, daß dies in digitalen Archiven keineswegs mehr gilt. Ein Artikel kann hier über Volltextrecherchen jederzeit wieder erreicht werden. Eine thematisch stärkere Eingrenzung bei oben beschriebenen Mehrdeutigkeiten kann häufig durch Angabe eines weiteren Suchbegriffs erreicht werden. Die Aufgaben eines Lektors können sich daher in

[17] Diese Aussage beruht auf Erfahrungsberichten aus dem Zeitungsarchiv des Axel Springer Verlages

einem digitalen Archiv also durchaus von denen in einem konventionellen Archiv unterscheiden. So können in einem digitalen Archiv neue Dienstleistungen bereitgestellt werden, die in einem konventionellen Archiv aus technischen beziehungsweise finanziellen Gründen nicht möglich waren. Diese Tatsache findet in dieser Arbeit unter anderem in der Möglichkeit der Einbindung von Nachrichten- und Profildiensten ihre Berücksichtigung.

Aufgrund der oben geschilderten Problematik bei der Volltextrecherche ist also die Aufnahme zusätzlicher semantischer Information in die Suchanfrage sinnvoll und auch notwendig. Dies erfolgt durch Attributfelder in der Anfragemaske, deren korrekte Umsetzung auf die einzelnen Datenbanken eine keineswegs triviale Aufgabe der Übersetzer ist.

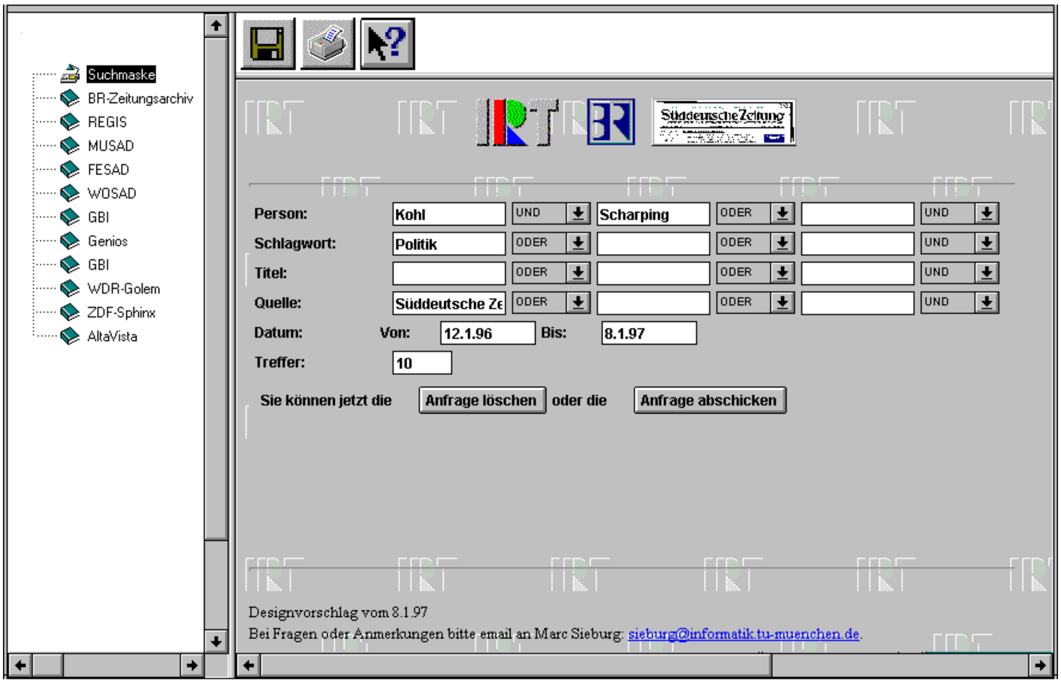

Abbildung 17: Suchmaske im Standardmodus

Ein wichtiges Attribut ist das Datum. Es steht auf fast allen Archiven, die diese Arbeit betreffen, zur Verfügung, und stellt ein starkes und häufig benötigtes Kriterium zur Eingrenzung der Treffermenge dar.

Als weitere wichtige Attribute haben sich Personen, Dokumententitel, Dokumentenquelle und übrige Schlagwörter erwiesen. Die Suchbegriffe innerhalb eines Attributes stehen in einer Zeile und können untereinander mit „`und`“, „`oder`“, „`nicht`“ gemäß der bool'schen Algebra und „`near`“ verknüpft werden. Wobei „`nicht`“ in diesem Fall als „`und nicht`“ zu verstehen ist. Die einzelnen Suchbegriffe werden jeweils in ein eigenes Feld eingegeben, die Verknüpfung kann - wie in Abbildung 17 dargestellt - über ein Popupmenü ausgewählt werden.

Da in der Standardoberfläche Klammern vorerst nicht vorgesehen sind, tritt automatisch das Problem von Mehrdeutigkeiten auf. Hier muß eine für den Anwender nachvollziehbare Lösung gefunden werden. Eine einfache Lösung ist, „`und`“ stärker als „`oder`“ zu betrachten[18]. Die Verknüpfung „`nicht`“ kann hier, da sie als „`und nicht`“ interpretiert wird, als gleichrangig mit „`und`“ betrachtet werden[19]. „`Near`“ als nicht bool'scher Ausdruck bindet am stärksten.

Weiterhin können die einzelnen Attributzeilen selbst wieder mit „`und`“, „`oder`“, „`nicht`“ verknüpft werden. Eine Verknüpfung mit „`near`“ ist hier nicht immer sinnvoll und sollte daher nicht in jedem Fall mit in die Verknüpfungsliste aufgenommen werden.

Zwischen den Zeilen gilt, um eine einheitliche Regelung für den Anwender anzubieten, ebenfalls die oben vorgeschlagene Regelung „`und`“ vor „`oder`“, wobei jede Zeile für sich als ein eingeklammerter Ausdruck gesehen wird. Sollte der Anwender zum Beispiel eine Recherche der Art „`(Person oder Schlagwort) und Titel`“ wünschen, so bleibt ihm allerdings in dieser Oberfläche nur die Möglichkeit, zuerst die Recherche „`Person oder Schlagwort`“ durchzuführen und anschließend, auf dieser Recherche aufbauend, nach „`Titel`“ zu recherchieren. Diese Form der Recherche ist jedoch aufgrund ihrer Komplexität im Augenblick dem Expertenmodus vorbehalten.

Welche Attribute angeboten werden sollen, kann im Rahmen dieser Arbeit nicht abschließend geklärt werden. Die hier aufgeführten Attribute haben eher Vorschlagscharakter. Es kann durchaus sein, daß sie im Laufe des Betriebs des Recherchesystems angepaßt werden oder bei verschiedenen Anwendergruppen, wie zum Beispiel Zeitungsredakteuren im Süddeutschen Verlag oder Rundfunkredakteuren im Bayerischen Rundfunk, Unterschiede aufweisen.

Grundsätzlich sollte auch die Anfragemaske nicht mit zu vielen verschiedenen Attributen überladen werden. Bei ihrer Auswahl sollte daher von den Bedürfnissen der Anwender ausgegangen werden und nicht so sehr der Wunsch im Vordergrund stehen, möglichst viele Fähigkeiten einzelner Archive in die Rechercheoberfläche zu übertragen. Diese Fähigkeiten sollen erst dem Experten im Expertenmodus umfassend zur Verfügung stehen.

---

[18]Es ist wichtig, daß dem Anwender an dieser Stelle im Hilfesystem durch viele, leicht verständliche Beispiele geholfen wird, um Verständnisproblemen und Frustration, bei der Eingabe und durch unerwartete Rechercheergebnisse, vorzubeugen. Hierbei können auch Grafiken, wie sie in der Mengenlehre in der Schule verwendet werden hilfreich sein.

[19] Aufgrund der Assoziativgesetze gilt $(a\wedge b)\wedge(\neg c) = a\wedge(b\wedge(\neg c))$

Expertenmodus

Auch im Expertenmodus steht für den Anwender eine für alle Datenbanken gleiche Recherchemaske zur Verfügung. Sie erhält zusätzlich noch eine Eingabezeile, in der eine Suchanfrage direkt als bool'scher Ausdruck mit Klammern, oder auch Kommandotext eingegeben werden kann.

Zusätzlich besteht in diesem Modus noch die Möglichkeit über die Originaloberflächen der einzelnen Archive auf diese zuzugreifen. Dies kann zum Beispiel bei REGIS eine 3270-Terminalemulation sein oder eine speziell für die jeweilige Datenbank entworfene Suchmaske. Sie wird anstatt der einheitlichen Suchmaske in das Recherchesystem eingeblendet, wenn der Anwender über den Trefferbaum den entsprechende Datenbankzugang ausgewählt hat. Bei einem 3270-Terminal sollte das Recherchesystem, soweit möglich, dem Anwender hierbei unter Prüfung der Zugriffsrechte, das Login abnehmen.

Optional sollte eine bereits in der Standardmaske eingegeben Recherche mit in die Suchumgebung der einzelnen Archive übernommen werden, um dem Experten mehrfache Arbeit zu ersparen und ihm einen besseren Einblick in das Verhalten der Übersetzer zu ermöglichen.

Anfängermodus

Im Anfängermodus steht dem Benutzer permanent ein elektronischer Hilfeassistent zur Seite. Er führt ihn bei seiner Recherche Schritt für Schritt durch die Suchmaske und erläutert ihm die einzelnen Felder. Hierbei sollte auch eine Reihe von einfachen Beispielen gezeigt werden, da diese einem schnelleren Verständnis sehr förderlich sein können.

Es wurde erwogen, in diesem Modus immer nur die aktuell notwendigen Bedienelemente anzuzeigen. Dies können zum Beispiel die aktuelle Attributzeile der Suchmaske oder am Ende der Eingabe einer Anfrage der Knopf zum Abschicken der Recherche sein. In diesem Fall würde allerdings die Übersichtlichkeit sehr leiden.

Eine andere Möglichkeit ist es, dem Anfänger bereits die Standardmaske zu präsentieren und die bei dem aktuellen Schritt notwendigen Bedienelemente optisch deutlich hervorzuheben. Der Hilfeassistent könnte in Form einer Sprechblase, die auf das aktuelle Feld zeigt, erscheinen. Auf diese Weise ist es dem Anfänger leichter möglich, den Überblick über bereits getätigte Eingaben und noch folgende Eingabefelder zu behalten. Der Umstieg auf den Standardmodus wird ebenfalls erheblich erleichtert.

Einige erläuternde, nicht zu lange Texte vor dem Erscheinen der Suchmaske und besonders, wenn Wartezeiten bei Suchanfragen auftreten, können den Einstieg in das System für den Anfänger zusätzlich vereinfachen.

4.3.5 Ergebnisanzeige

Die Ergebnisse einer Suchanfrage werden, abgesehen von der Trefferliste im Trefferbaum, im selben Bereich dargestellt, in dem vorher die Suchmaske angezeigt wurde[20].

Im Expertenmodus sollte es noch zusätzlich möglich sein, wenn der Anwender dies wünscht, mehrere Treffer in verschiedenen Fenstern parallel anzuzeigen. In einem dieser Fenster könnte dann zum Beispiel auch die Suchmaske während der Ergebnisanzeige geöffnet bleiben.

Nach dem Abschicken einer Recherche erscheint in der Ergebnisanzeige zunächst eine Anzeige über den aktuellen Status der Recherche. Diese kann je nach Benutzermodus verschieden ausführlich ausfallen oder sogar ganz ausgeschaltet werden. Vor allem im Anfänger- und Standardmodus ist sie jedoch wichtig, um das Überbrücken von eventuellen Wartezeiten zu erleichtern.

Sobald die ersten Ergebnisse eingetroffen sind, können diese über den Trefferbaum selektiert und in der Ergebnisanzeige dargestellt werden (Abbildung 18).

Die Ergebnisse können die Trefferlisten der einzelnen Archive und nach augenblicklichem Stand Volltextdokumente, formatierte Volltextdokumente, Quasifaksimile, Faksimile, Bilder, Filme, Animationen, Musik[21] oder Mischformen dieser Dokumente sein.

[20] Wenn dies nicht ausdrücklich vermerkt wird, beziehen sich alle Angaben in diesem Kapitel sowohl auf den Standard- als auch auf den Anfänger- und Expertenmodus

[21] Diese wird , falls vorhanden und aktiviert, über den Audiokanal des Computers ausgegeben.

Die Anbieter haben objektorienterte Frontend-Anwendungen zu lange vernachlässigt

Java wird Internet und Client-Server vereinen

Das Interesse an objektorientierter Entwicklung steigt. Grund dafür ist der Erfolg von Java und die Anziehungskraft des Internet.

Die Anbieter objektorientierter Technologien mußten während des Internet-Booms festellen, daß sie sich auf die falsche Seite konzentriert haben. Ihre Tools zielen häufig auf das Back-End und Anwendungen für Server. Durch das World Wide Web ist die Entwicklung heute aber Endanwender getrieben. Im Front-End-Bereich fehlt es jedoch oft an Anwendungen.

Die plattformunabhängige Internet-Technologie Java hat das Potential, diese Lücke zu schließen. Vor allem ist Java viel einfacher zu programmieren als frühere objektorientierte Programmiersprachen wie C++. So müssen sich Anwendungsentwickler nicht mehr auf eine Betriebssystemplattform festlegen, wenn ... dungen wie von SAP und Oracle später nachziehen.

Die größte Herausforderung im Umfeld objektorientierter Technologien ist die Integration der verschiedenen verteilten Architekturen. Hierzu gehören nach Ansicht von Jean-Marie Chauvet von Neuron Data das traditionelle Client-Server-Computing, die beiden konkurrierenden Objekttechnologien Object Management Architecture (OMA) von dem Standardisierungsgremium OMG und OLE/DCOM von Microsoft sowie die Web-basierte Verteilung von Anwendungen und Daten. Nach Ansicht von Chauvet besitzt die Objekttechnologie die notwendige Flexibilität, die verschiedenen Technologien zusammenzubringen.

Entwicklungs-Tools in der Pipeline

Große Erwartungen stecken ...

Eine andere Form der Internet-Objekt-Integration plant Apple. Durch seine Cyberdog-Technologie in Netscapes Web-Browser stellt das Unternehmen die Verbindung auf Desktop-Ebene her. Der Browser kann als Fragment arbeiten, das andere Opendoc-basierende Anwednungen einsetzen. Über den Erfolg der objektorientierten Technologien entscheidet nicht zuletzt die Fähigkeit, unternehmenskritische Anwendungen zu erstellen. Eine Schlüsselrolle hat hierbei die Unified Modelling Language (UML) von Rational, die eine einheitliche Methodologie für die Entwicklung bietet. Auch Java-Programmierer werden UML nutzen müssen, wenn sie komplexe Anwendungen erstellen wollen. Ob Java allein die Anforderungen an unternehmenskritische Anwendungen erfüllen kann, ist aber noch offen. So erwartet Katy Ring von Ovum, daß die Nutzung vieler Applets, die über das Netz geladen werden und auf Serverdienste zugreifen, einen Object Request Browser erfordern werden, um die Kommunikation mit dem Backend-Server sicherzustellen.

Elspeth Wales/ Thomas Kaneshige

Abbildung 18: Ergebnisanzeige eines Faksimiledokuments

Für einzelne Dokumentarten können, während sie angezeigt werden, zusätzliche Bedienelemente in der Funktionsleiste, oder wenn sie ohne Scrolling erreicht werden können, in der Ergebnisanzeige, zur Verfügung stehen. Für Filme, Animationen und Musik sind dies die Knöpfe Start, Stop, Vor- und Zurückspulen und Pause, wie sie von vielen Video- oder Musikrecordern her bekannt sind.

Die Darstellungsgröße des Dokuments läßt sich, wie in Kapitel 4.3.3 beschrieben, über die Funktionsleiste einstellen. Zusätzlich sollte ist es noch möglich, dies über die Tasten „+" und „-" zu erreichen.

Für Dokumente, die über die Ränder der Ergebnisanzeige herausragen, werden Schiebebalken zur Verfügung gestellt. Darüber hinaus ist eine Positionierung des Dokuments über die Pfeiltasten ist möglich.

In der Ergebnisanzeige dargestellte Dokumente können, falls es dessen Copyrightbestimmungen, die Dokumentart und die Zugriffsrechte des Anwenders zulassen, lokal gespeichert, gedruckt oder über das Clipboard des Systems in einem externen Programm weiterverarbeitet werden.

Sollten Dokumente in einem Produktionssystem weiterverarbeitet werden, so ist noch zu berücksichtigen, daß speziell Bilder, Musik, Faksimile und Video in der Regel in einem für den Bildschirm optimierten, möglichst kompakten Format an die Ergebnisanzeige geliefert werden. Aus diesem Grund steht für den Export eines Dokumentes noch die Option „Export für Produktionssystem" zur Verfügung. In diesem Fall wird das Dokument vom Archiv in einer für das Produktionssystem qualitativ ausreichenden Form angefordert und analog zu oben exportiert. Sollte ein Archiv ein Dokument noch nicht in dieser hochwertigen Form online zur Verfügung stellen, so kann über diese Option der Wunsch nach diesem Dokument an das Archiv übermittelt werden. Die Zustellung erfolgt dann auf traditionellem Weg.

Es ist, vor allem in der Anfangsphase des Projekts, durchaus noch möglich, daß auch dies von einzelnen Archiven noch nicht unterstützt wird. In diesem Fall bleibt nur der traditionelle Weg einer telefonischen Anforderung des Dokuments mit einer Bestellnummer oder anderen zu einer Bestellung notwendigen Angaben. Die hierzu notwendigen Schritte und Informationen sollten in der Ergebnisanzeige erscheinen, falls der „Export für Produktionssystem" ausgewählt wurde.

Auch in der Ergebnisanzeige wird der Anwender im Anfängermodus durch einen elektronischen Hilfeassistenten unterstützt, der ihn durch die Bedienung der Ergebnisanzeige führt, ihm Tips zu aktuellen Feldern gibt und ihm Vorschläge zum weiteren Vorgehen liefert.

4.3.6 Hilfesystem

Das Hilfesystem gliedert sich in mehrere Stufen, mit zunehmenden Umfang der Hilfetexte. Teile dieses Hilfesystems wurden bereits im Anfängermodus der Suchmaske (Kapitel 4.3.4) und in der Funktionsleiste (Kapitel 4.3.3) beschrieben.

Eine Kurzinformation zu jedem Bedienelement erscheint automatisch in einer Statuszeile, sobald sich der Mauszeiger über diesem befindet. Die Statuszeile kann sich zum Beispiel im unteren Fensterrand des Recherchesystems oder alternativ auch rechts in der Funktionsleiste der Rechercheoberfläche befinden.

Eine etwas ausführlichere Hilfe erhält der Anwender bei der, optional abschaltbaren, „Bubblehelp". Hier erscheint, wenn der Mauszeiger eine gewisse Zeit über einem Bedienelement verweilt, neben dem Zeiger ein Textrahmen mit einer kurzen Erläuterung, der wieder ausgeblendet wird, sobald der Mauszeiger dieses Element wieder verläßt.

Nach Drücken des Hilfeknopfs in der Funktionszeile, kann der Anwender alternativ Hilfe zu einzelnen Elementen der Rechercheoberfläche, Hilfe zu dem aktuell aktiven Element oder allgemeine Hilfetexte erhalten.

Im ersten Fall wird der Mauszeiger mit einem Fragezeichen versehen, und der Anwender erhält durch einfachen Klick auf ein Element der Rechercheoberfläche den dazugehörigen Hilfetext.

Im zweiten Fall, der auch durch Drücken der F1-Taste erreicht werden kann erhält der Anwender ausführliche Hilfetext, Tips und Beispiele zu dem aktuell aktiven Element. Dies kann zum Beispiel ein Feld der Recherchemaske sein, in dem sich gerade der Cursor befindet oder ein geöffnetes Dialogfenster. Kann kein aktives Element ermittelt werden, so erscheint die allgemeine Hilfe.

Im dritten Fall erhält er eine Auswahl der Hilfetexte mit Beispielen und Tips zu Themen wie zum Beispiel Durchführung einer Recherche oder die Sichtung und Weiterverarbeitung von Rechercheergebnissen.

Als letztes steht noch eine elektronischer Hilfeassistent zur Verfügung, der speziell für den Anfängermodus entworfen, wie in Kapitel 4.3.4 und 4.3.5 beschrieben, den Benutzer Schritt für Schritt durch das Recherchesystem führt.

Es ist wichtig, daß alle Hilfetexte, Erläuterungen und auch Fehlermeldungen in einer möglichst höflichen, leicht verständlichen und den Benutzer ansprechenden Form gehalten werden[22]. Der Anwender darf sich nicht von dem System allein gelassen oder angegriffen fühlen. Fehlverhalten des Systems, sei es durch fehlerhafte Eingaben oder durch tatsächliche Fehler im System, führt bei Anwendern häufig zu Ablehnung oder sogar Aggressionen. Diese, soweit möglich, abzufangen ist eine wichtige Aufgabe und muß beim Verfassen der Texte berücksichtigt werden.

[22] Dies ist durchaus noch nicht immer eine Selbstverständlichkeit. Fehlermeldungen der Art „Das ist aber total falsch! Hier muß A, B, C stehen.“, wie sie in manchen elektronischen Archiven noch vorkommen, können den Anwender vor den Kopf stoßen und , vor allem wenn sie sich häufen, eine negative Grundeinstellung gegenüber dem System zur Folge haben.

Kapitel 5

Realisierung eines Prototyps für einen Redaktionsarbeitsplatz

In diesem Kapitel wird die Realisierung eines Prototyps für einen Recherchearbeitsplatz für Redakteure beschrieben. Aufgabe dieses Prototyps ist es zu zeigen, daß mit der im letzten Kapitel vorgeschlagenen Architektur die gleichzeitig Anbindung verschiedener Datenbanken möglich ist und eine einfache Crossrecherche über eine einheitliche Oberfläche durchgeführt werden kann. Eine vollständige Implementierung aller in der Anforderungsanalyse oder im Konzept der Oberfläche geforderten Fähigkeiten des Recherchesystems kann nicht Aufgabe des Prototyps sein.

5.1 Umsetzung des Entwurfs in einen Prototyp

Der im letzten Kapitel vorgestellte Lösungsansatz wurde in Bezug auf die verwendete Systemumgebung bewußt allgemein gehalten. Es fand weder eine Festlegung auf eine spezielle Netzwerktechnik, noch auf ein Betriebssystem oder spezielle Entwicklungswerkzeuge oder Software zur Implementierung des Systems statt.

Wie bereits in der Anforderungsanalyse dargestellt, ist dies aufgrund der äußerst heterogenen Systemumgebung im BR und SV in Bezug auf das Betriebsystem und den Zugang zu den elektronischen Archiven auch unbedingt zu vermeiden.

Es muß eine Lösung gefunden werden, die Anwendern auf verschiedenen Betriebsystemen möglichst einfach und kostengünstig ein einheitliches Recherchesystem bietet. Da zumindest bei den Arbeitsplatzrechnern der Anwender in Zukunft eine Vernetzung auf TCP/IP-Basis vorausgesetzt werden darf, bietet sich für die Implementierung der Benutzeroberfläche ein Web-

Browser[23] an. So steht zum Beispiel der Browser von Netscape auf allen in der Anforderungsanalyse aufgeführten Betriebsystemen zur Verfügung.

Um jedoch den Anforderungen des Konzepts für den Recherchearbeitsplatz gerecht zu werden, reicht eine reine Implementierung der Oberfläche auf HTML-Basis nicht mehr aus. Zur Realisierung des Trefferbaums und der interaktiven Hilfe ist weitere Funktionalität notwendig. Diese kann auf einfache Art und Weise in Java[24] programmiert werden.

Durch die Kombination von Web-Browsern und Java sind eine Vielzahl der in der Anforderungsanalyse und im Lösungsansatz enthaltenen Forderungen sehr einfach zu erfüllen. Dazu gehören unter anderem die Plattformunabhängigkeit, ein Großteil des Cachings, das Vermeiden von großen Datenmengen, die lokale Ausführung möglichst vieler Aufgaben, Teile der interaktiven Hilfe, und die leichte Wartbarkeit.

So werden zum Beispiel die Rechercheoberfläche, oder bereits vom Anwender geladene Ergebnisse vom Web-Browser in einem lokalen Cache gespeichert. Bei einer Modifikation einer dieser Komponenten auf dem Web-Server werden diese jedoch beim nächsten Aufruf des Anwenders automatisch auf den neuesten Stand gebracht.

Weiterhin kann durch die Realisierung des Anfrageobjekts als ein für den Anwender nicht sichtbares Applet, die Häufigkeit des Verbindungsaufbaus zum Web-Server auf ein Minimum beschränkt und durch die lokale Ausführung auf dem Client-Rechner der Web-Server entlastet werden.

Auf die vielen Vorteile und Möglichkeiten die Java in Zusammenhang mit einem Web-Browser bietet soll hier nicht weiter eingegangen werden. Es sei hier an die Unmengen von Berichten in der (Fach-) Presse verwiesen.

Auch für die Realisierung der übrigen Komponenten der Systemarchitektur wurde Java gewählt. Gründe hierfür sind unter anderem die dadurch gewährleistete Plattformunabhängigkeit, die in Java extrem einfach zu implementierende Kommunikation zwischen den einzelnen Komponenten durch Socket-Verbindungen, das Threadkonzept und seine Objektorientiertheit.

Insgesamt ergibt sich für den Prototyp eine Verteilung der einzelnen Komponenten wie sie in Abbildung 19 dargestellt ist.

[23] Eine einfache Einführung in das *World Wide Web* (WWW) bietet sum Beispiel [Klute 96]
[24] Eine Einführung in Java bietet [Flanagan 96] oder [Goslin, Joy, Steele 96]

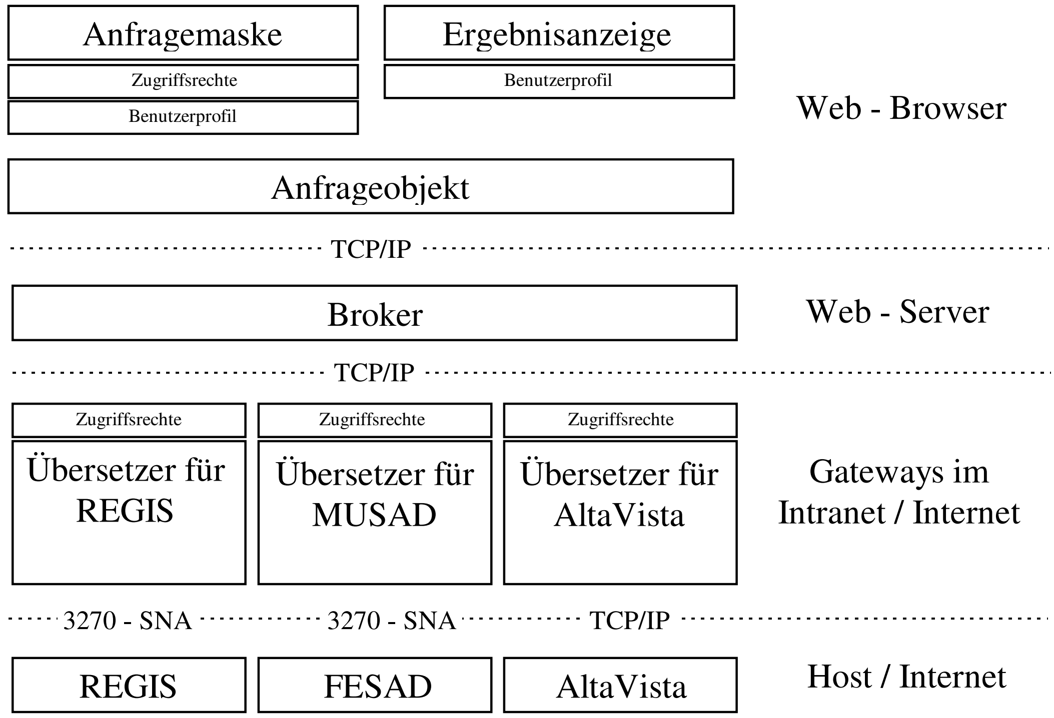

Abbildung 19: Umsetzung der Architektur im Prototyp

5.1.1 Benutzerschnittstellen

Im Prototyp wurde exemplarisch die Oberfläche für den Standardmodus implementiert. Ziel war es, einen Eindruck über die Art und Weise zu vermitteln, mit der über eine einheitliche Rechercheoberfläche parallel auf mehreren Archiven gleichzeitig recherchiert werden kann und ein Werkzeug für spätere Testrecherchen zur Verfügung zu stellen. Es ist nicht die Aufgabe des Prototyps bereits die volle, im Lösungsansatz aufgeführte Funktionalität bezüglich des Hifesystems, der Benutzerprofile und der Bedienung der einzelnen Komponenten der Benutzerschnittstellen bereitzustellen.

Die Rechercheoberfläche gliedert sich, wie in Abbildung 20 dargestellt, in folgende HTML-Frames:

- Funktionsleiste
- Anfragemaske / Ergebnisanzeige
- Trefferbaum

wobei die in der Funktionsleiste abgebildeten Knöpfe im Prototyp noch keine Funktionalität besitzen. Sie werden nur angezeigt, um zu verdeutlichen, über welche Bedienelemente in einem zukünftigen Recherchesystem das Laden, Speichern, Drucken von Dokumenten oder Rechercheanfragen und das Aufrufen des Hilfesystem möglich sein wird.

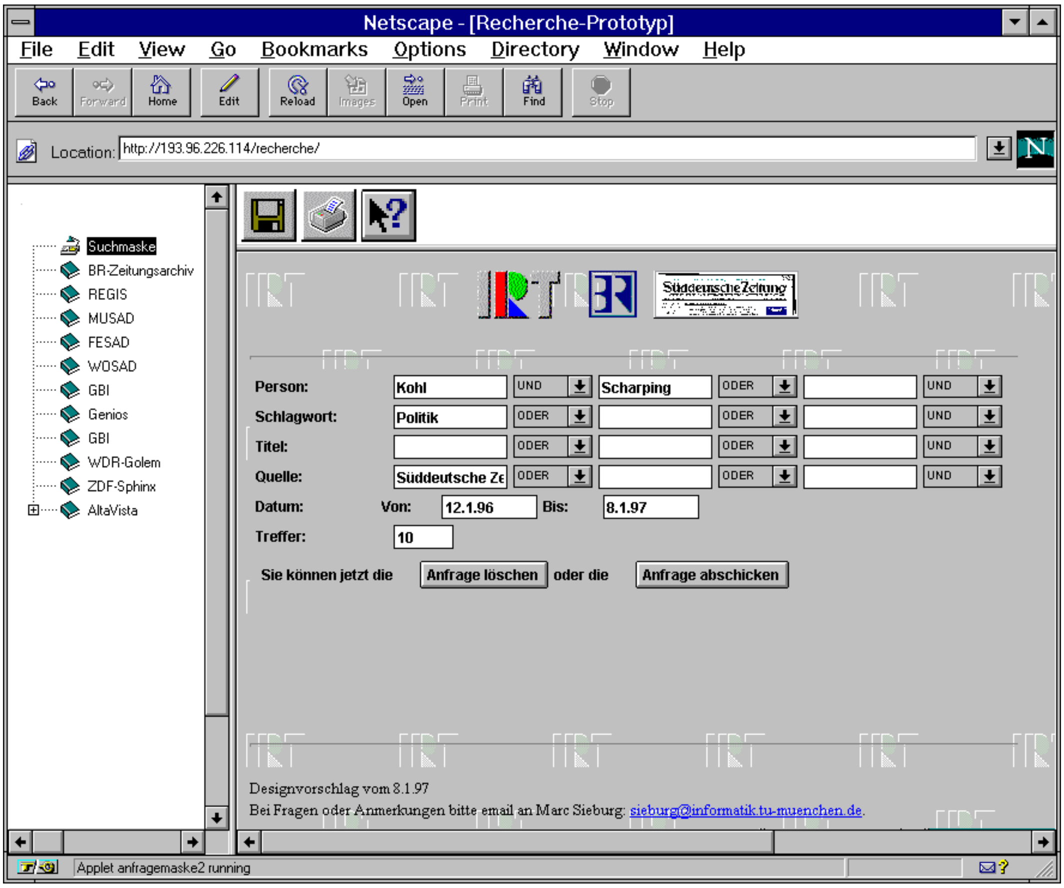

Abbildung 20: Prototyp der Rechercheoberfläche unter Netscape

Anfragemaske:

Die Implementierung der Anfragemaske erfolgte ebenfalls in Java[25]. Prinzipiell wäre eine Realisierung des Eingabeformulars in HTML sehr einfach möglich gewesen. Hier wurde jedoch Java[26] vorgezogen, da auf diese Weise eine direkte Kommunikation der Anfragemaske mit dem Anfrageobjekt möglich ist.

[25] Diese erfolgte zu Beginn der Diplomarbeit in Zusammenarbeit mit einem Fopra von Christoper Schmid [Schmid 96], in dem von ihm diese Anfragemaske programmiert wurde.

[26] Entwicklungswerkzeuge, wie zum Beispiel Visual Cafe von Symantec, ermöglichen es inzwischen solche Eingabemasken auf extrem einfache Art, sogar ohne daß der Entwickler Kenntnisse in Java besitzen muß, als Java-Applets zu realisieren.

Die Maske besteht, wie im Lösungsansatz in Kapitel 4.3.4 beschrieben, aus mehreren Attributzeilen. In jeder Zeile stehen mehrere Felder zur Verfügung, in denen die Suchbegriffe eingegeben und durch „und“, „oder“, „nicht“ über die Popups zwischen diesen Feldern verknüpft werden können.

Eine Besonderheit dieser Maske ist, daß sie, sobald der Anwender ein Feld in der letzten Spalte ausfüllt, eine weitere Spalte zur Verfügung stellt.

Ist die Anfrage fertig eingetragen worden, so kann sie durch drücken des „Anfrage abschicken“-Knopfes abgeschickt werden. Die Anfragemaske baut dann eine Verbindung zum Anfrageobjekt auf und sendet ihm eine Anfrage der Form

```
<Anfrage>::=

     <Attributzeile>{<Verknüpfung><Attributzeile>}*

<Attributzeile>::=
     Attribut=<String>{<Verknüpfung><String>}*

<Verknüpfung>::=    & | ⊥ | ~ | !

<String>::=         String of char
```

Während der Bearbeitung der Recherche durch das System wird der Knopf „Anfrage abschicken“ bis zum Eintreffen der letzten Ergebnisse deaktiviert.

In dieser Zeit kann der Anwender jedoch alle übrigen Funktionen des Systems weiter bedienen, also sich zum Beispiel die ersten Treffer ansehen, Dokumente laden, oder bereits eine neue Recherche eintragen.

Ergebnisanzeige:

Hier werden, soweit möglich, die Fähigkeiten des Web-Browsers ausgenützt. Sämtliche Ergebnisse werden als HTML-Seiten geladen. Somit stehen dem Anwender die gesamten Multimediafähigkeiten des Browsers zur Verfügung.

Trefferbaum:

Bei seiner Initialisierung baut er eine Verbindung zum Anfrageobjekt auf und fordert über die Funktion `BrokerGetDB()`[27] von diesem die anzuzeigenden Archive an. Sollte das Anfrageobjekt nicht zu erreichen sein, so wird ein Fenster mit einer Fehlermeldung geöffnet und im Trefferbaum nur der Eintrag für die

[27] `BrokerGetDB()` wird in Kapitel 5.1.2 Anfrageobjekt beschrieben.

Suchmaske angezeigt. Der Trefferbaum bleibt, ebenso wie die Funktionsleiste, während der gesamten Arbeitszeit mit dem Recherchesystem sichtbar.

Für jeden Baumeintrag können für Einfach- und Doppelklick unterschiedliche Aktionen implementiert werden. Die abstrakte Basisklasse eines Baumeintrags „TreeNode" stellt hierfür die Funktionen `public abstract void SingleClick()` und `public abstract void DoubleClick()` zur Verfügung. Durch Ausnutzung der Polymorphie-Eigenschaften von Java kann hierdurch eine extreme Flexibilität und Erweiterbarkeit des Trefferbaums erreicht werden.

Im Prototyp wurde von TreeNode die Klasse URLNode abgeleitet, die für Einfach- und Doppelklick zwei eventuell verschiedene Dokumente lädt. Es wird erkannt, ob es sich bei dem jeweiligen Dokument um eine URL-Adresse[28] für eine HTML-Seite oder eine Dokumentkennung für eine Anforderung beim Anfrageobjekt handelt.

Der Trefferbaum dient während der Eingabe einer Recherche als Auswahlliste für die Archive. Über ihn können einzelne Archive, falls diese angeschlossen sind und eine eigene HTML-Seite besitzen, direkt angewählt werden[29].

Während der Durchführung einer Recherche wird der Trefferbaum laufend aktualisiert. Für den Anwender ist so der aktueller Fortschritt seiner Anfrage zu erkennen.

Eingetroffene Ergebnisse können von dem Anwender bearbeitet werden. Ein Einfachklick auf den Archivnamen zeigt die Trefferliste zu dem jeweiligen Archiv in HTML-Form in der Ergebnisanzeige. Ein Klick auf das +/- Zeichen vor dem Archivnamen öffnet, beziehungsweise schließt den Unterbaum des Archivs.

Durch Einfachklick auf einen Eintrag der Trefferliste wird die HTML-Trefferliste in der Ergebnisanzeige ab der Position des angewählten Eintrags angezeigt. Bei einem Doppelklick wird das entsprechende Dokument in die Ergebnisanzeige geladen.

Das in diesem Abschnitt geschilderte Verhalten des Trefferbaums für Einfach- und Doppelklick auf einzelne Einträge ist stark abhängig von den im Broker erzeugten und an das Anfrageobjekt gesendeten Trefferlisten, aus denen dieser Baum aufgebaut wird, und von denen die Struktur des Baumes und die zu ladenden Dokumente bestimmt werden.

---

[28] URLs (Uniform Resource Locators) dienen zum Auffinden von Dokumenten im World Wide Web.

[29] In einer späteren Version des Recherchesystems kann hier auch die Anzahl der zu recherchierenden Archive eingegrenzt werden. In dem in dieser Arbeit implementierten Prototypen ist diese Funktion noch nicht in den Trefferbaum implementiert.

5.1.2 Anfrageobjekt

Das Anfrageobjekt läuft als für den Anwender nicht sichtbares Applet im Browsing-Fenster auf der linken Seite, in dem auch der Trefferbaum ausgeführt wird. Die gesamte Kommunikation mit dem Broker erfolgt ausschließlich über dieses Applet. Die dazu notwendige Informationen, wie die IP-Adresse des Brokers, seine Portnummer und das Basisverzeichnis der Ergebnisse der Anfragen auf dem Web-Server (`URLPath`) erhält das Anfrageobjekt als Parameter von seiner ihn startenden HTML-Seite (Abbildung 21). Auf diese Weise läßt es sich sehr einfach an verschiedene Web-Server, beziehungsweise Broker anpassen.

```
<HTML>
<HEAD>
<TITLE>Browse</TITLE>
</HEAD>
<BODY>
<APPLET CODE="AnfrageObject.class" WIDTH=0 HEIGHT=0>
    <param name=BrokerIPAdr   value="194.172.230.114">
    <param name=BrokerPort    value="3170">
    <param name=URLPath       value="/BRSV/recherche/">
</APPLET>
<APPLET CODE="TreeApplet.class" WIDTH=800
HEIGHT=600></APPLET>
</BODY>
</HTML>
```

Abbildung 21: HTML-Seite des Anfrageobjekts

Da die verschiedenen Applets der Rechercheoberfläche zum Teil in verschiedenen Frames des Web-Browsers ablaufen, ist ein direkter Datenaustausch zwischen diesen nicht ohne weiteres möglich. Das Problem kann durch Einführen einer Klasse mit statischen Variablen, im Folgenden `Mediator`[30] genannt, gelöst werden. Sie stehen während der Laufzeit des Web-Browsers allen Applets[31] zur Verfügung. In diesem Prototyp werden über den `Mediator` eine Schnittstelle zur

[30] Der Name wurde aus einem Beispielprogramm für Inter-Applet-Kommunikation des Entwicklungssystems „Symantec Cafe" übernommen.

[31] Bei unvorsichtiger Nutzung dieser Eigenschaft von Java können eventuell gravierende Sicherheitslücken entstehen. So sollten zum Beispiel sicherheitsrelevante Informationen, die nicht für die Öffentlichkeit bestimmt sind, nicht als öffentliche statische Variablen gespeichert werden, da sie von jedem Applet im Web-Browser gelesen werden können. (Vorausgesetzt, es kennt den Klassen und den Variablennamen)

Steuerung des Trefferbaums und zur Kommunikation mit dem Anfrageobjekt zur Verfügung gestellt.

Auf diese Weise wurde ein einfacher browserinterner RPC-ähnlicher Mechanismus implementiert. Der `Mediator` tritt als Vermittler zwischen den Clients, dies sind hier meist die Applets der Benutzerschnittstelle und dem Server, dies ist in der Regel das Anfrageobjekt, auf[32].

Für diese Kommunikation stellt das Anfrageobjekt die Klasse `AnfrageThread` zur Verfügung. Bei ihrer Erzeugung trägt sie sich beim `Mediator` ein und kann damit von jedem Applet der Rechercheoberfläche verwendet werden.

Sie stellt folgende Funktionen zur Verfügung**:**

`public void BrokerGetDB()`

Über diese Funktion können die beim Broker für den Anwender verfügbaren Datenbanken abgefragt werden. Das Anfrageobjekt sendet an den Broker eine Anfrage der Form:

```
[GET_DB]

[USER]=UserId[Pwd]=UserPwd
```

Dabei ist „UserId“ die beim Start der Rechercheoberfläche abgefragte Benutzerkennung und „UserPwd“ das dazugehörige Paßwort. Sie sind beim Anfrageobjekt gespeichert und werden von diesem eingefügt.

Sollte der Broker nicht erreichbar sein oder während der Kommunikation mit diesem ein Fehler auftreten, so wird ein Fenster mit einer Fehlermeldung angezeigt und die Anfrage abgebrochen[33].

Das Ergebnis der Anfrage wird automatisch in den Trefferbaum eingetragen.

`public void BrokerRecherche( String Anfrage )`

Sendet eine Rechercheanfrage an den Broker. Sie hat folgenden Aufbau:

```
[RECHERCHE]

[USER]=UserId[Pwd]=UserPwd

Anfrage
```

[32] Eine Einführung in Remote Procedure Calls (RPC) findet sich zum Beispiel in [Borghoff,Schlichter 95] Seite 26 ff

[33] Dies gilt für alle Funktionen, in denen das Anfrageobjekt eine Verbindung zum Broker aufbaut.

`[/RECHERCHE]`

„Anfrage“ ist hierbei der meist mehrzeilige Anfragestring, wie er von der Suchmaske geliefert wird.

Die Ergebnisse der Recherche werden automatisch vom Anfrageobjekt in den Trefferbaum und die Ergebnisanzeige eingetragen.

`public void BrokerGetDocument( String Doc, String DB )`

Fordert ein einzelnes Dokument „Doc“ von einer Datenbank „DB“ beim Broker wie folgt an:

```
[GET_DOC]

[USER]=UserId[Pwd]=UserPwd

Doc

DB

[/GET_DOC]
```

Das vom Broker gelieferte Dokument wird automatisch in der Ergebnisanzeige angezeigt.

Alle drei Funktionen bauen eine Socketverbindung zum Broker auf, senden ihm die notwendigen Daten und verarbeiten das eintreffende Ergebnis in der Funktion „`AnalyseBrokerIn()`“. Eine Übersicht über diesen Ablauf zeigt Abbildung 22.

Am Ende der Kommunikation wird die Verbindung wieder abgebaut. Erst danach kehrt die jeweilige Funktion zurück. Das aufrufende Programm erhält so die Möglichkeit, zum Beispiel während der Durchführung einer Recherche, einzelne Menüpunkte als belegt zu markieren und diese danach wieder freizugeben. Sollte von dem aufrufenden Programm eine Ausführung im Hintergrund gewünscht werden, so braucht es nur einen neuen Thread zu erzeugen, der eine dieser Funktionen aufruft, nach ihrer Terminierung eventuell noch einzelne Aktionen durchführt und sich dann wieder beendet.

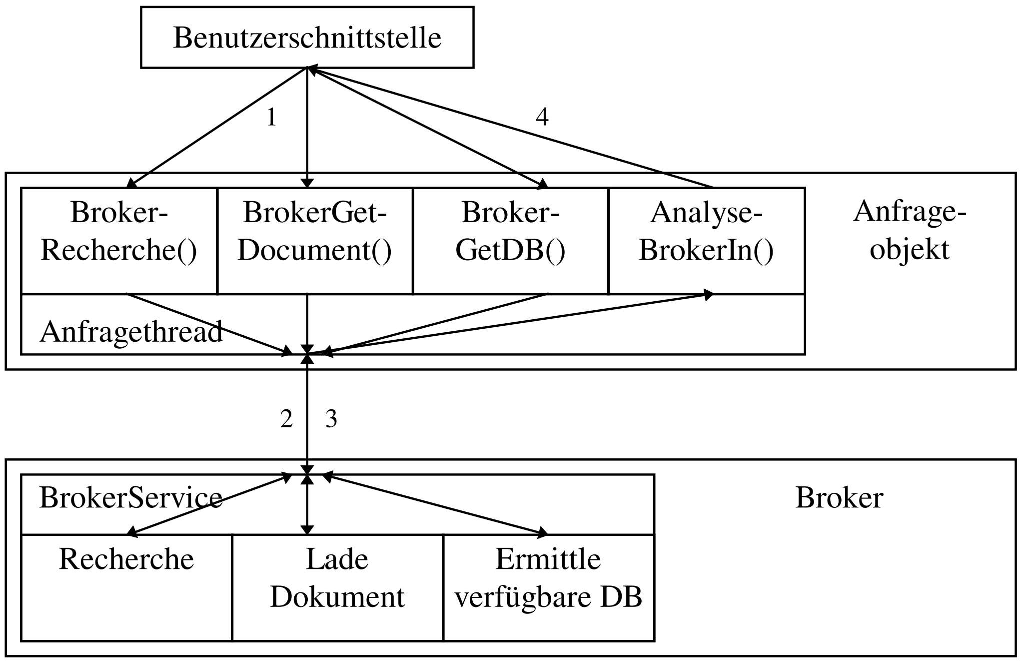

Abbildung 22: Datenfluß beim Bearbeiten einer Anfrage

Die Antworten des Broker werden bei allen drei Funktionen in der Funktion

`private void AnalyseBrokerIn( ... ) throws IOException`

ausgewertet.

Eine sofort anzuzeigende HTML-Seite sendet der Broker in der Form:

`[htmlErgebnis]=Pfad/Seite.html`

Diese wird vom Anfrageobjekt in die Ergebnisanzeige geladen und sofort dargestellt. Die HTML-Adresse wird hierbei aus den Parametern `BrokerIPAdr + URLPath + htmlErgebnis` zusammengesetzt. Es handelt sich bei diesen Seiten also ausschließlich um auf dem gleichen Web-Server zur Verfügung stehende Dokumente, auf dem auch der aktuell verwendete Broker installiert ist. Von beliebigen Web-Servern zu ladende Dokumente werden unter einer anderen Kennung vom Broker übergeben.

Etwas komplizierter ist der Aufbau der Trefferlisten, da diese nach Archiven sortiert in einer Baumstruktur vom Broker geliefert werden[34]:

`<TOC>::={ <Item> | <Subtree> }*`

[34] Die Beschreibung erfolgt in BNF-Form. Siehe dazu [Broy 92] Seite 74 ff

```
<Subtree>::=   <Item>
               ⌊ITEMS⌋
               {<Item> | <Subtree>}+
               ⌊/ITEMS⌋
<Item>::=      ⌊ITEM⌋
               <Attribut>
               ⌊/ITEM⌋
<Attribut>::=  ⌊Title⌋=<String>
               {⌊Subtitle⌋=<String>}
               {⌊URL1⌋=<String>}
               {⌊URL2⌋=<String>}
               {⌊Image⌋=<String>}
               {⌊OpenImage⌋=<String>}
<String>::=    String of char
```

Anhand dieser Trefferlisten wird der für den Benutzer sichtbare Trefferbaum aufgebaut, beziehungsweise aktualisiert.

Für jeden Baumeintrag (`Item`) muß eine Titel (`Title`) angegeben werden. Weiterhin kann er einen Untertitel (`Subtitle`), bis zu zwei Dokumentadressen (`URL1` und `URL2`), sowie bis zu zwei Symbole (`Image` und `OpenImage`) enthalten. `Image` und `OpenImage` sind die relativen[35] URL-Adressen der Images, die vor dem jeweiligen Baumeintrag, in Abhängigkeit von dessen Zustand, angezeigt werden sollen. URL1 und URL2 können jeweils eine URL-Adresse oder der zur einer Dokumentanforderung beim Broker mittels der Funktion `BrokerGetDocument( String Doc, String DB )` notwendige Anfragetext „Doc" sein. Bei Einfachklick des Anwenders auf einen Baumeintrag wird die zu diesem Eintrag gehörende URL1 ausgewertet, bei Doppelklick entsprechen URL2.

Abbildung 23 zeigt ein kurzes Beispiel für eine Trefferliste. Die Einrückungen der einzelnen Zeilen erfolgen hier zur besseren Übersichtlichkeit. Sie finden bei der tatsächlichen Datenübertragung nicht statt.

[35] es sind die URL-Adressen relativ zu der URL-Adresse des Anfrageobjekts.

```
[TOC]
   [ITEM]
       [Title]=REGIS
       [Subtitle]=Zeitungsarchiv des Süddeutschen Verlags
       [URL1]=results/REGIS/Results42.html
       [URL2]=REGIS.html
       [Image]=images/book.gif
       [OpenImage]=images/openbook.gif
    [/ITEM]
    [ITEMS]
        [ITEM]
            [Title]=Erster Treffer
            [Subtitle]=Ein reines Testdokument
            [URL1]=results/REGIS/Results42.html#T1
            [URL2]=[DOCU]=Erster Treffer[DATE]=01.04.1997
            [Image]=images/page.gif
        [/ITEM]
        [ITEM]
            [Title]=Zweiter Treffer
            [Subtitle]=Auch ein reines Testdokument
            [URL1]=results/REGIS/Results42.html#T2
            [URL2]=[DOCU]=Zweiter Treffer[DATE]=01.04.1997
            [Image]=images/page.gif
        [/ITEM]
    [/ITEMS]
    [ITEM]
        [Title]=AltaVista
        [Subtitle]=Internet Suchmaschine AltaVista
        [URL2]= http://www.altavista.com
        [Image]=images/book.gif
        [OpenImage]=images/openbook.gif
    [/ITEM]
[/TOC]
```

Abbildung 23: Beispiel einer Trefferliste

5.1.3 Broker

Der Broker muß theoretisch beliebig viele Anfragen gleichzeitig bearbeiten können. Er tritt gegenüber dem Anfrageobjekt als Server auf, bei den Übersetzern stellt er dagegen einen Client dar[36]. Für jede Anfrage müssen unter Umständen zu mehreren Übersetzern parallel Verbindungen aufgebaut werden. Die von den Übersetzern eintreffenden Daten sollen ebenfalls parallel empfangen und schließlich voneinander möglichst unabhängig an das Anfrageobjekt weitergeleitet werden. Der Anwender soll also die ersten Ergebnisse der einzelnen Archive präsentiert bekommen, sobald sie vorliegen und nicht erst dann, wenn alle Übersetzer ihre Übertragung beendet haben.

Diese Forderungen wurde mit Hilfe eines mehrstufiges Threadkonzepts, wie es in Abbildung 24 dargestellt ist, umgesetzt.

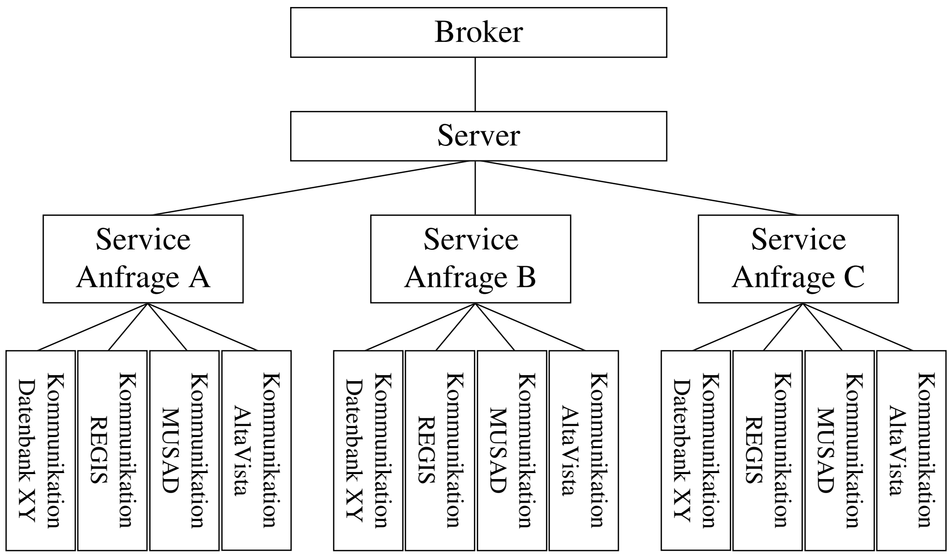

Abbildung 24: Threadkonzept des Brokers

Der Broker wurde als Konsolen-Applikation realisiert, um eine Ausführung auf einem Web-Server zu vereinfachen. Sämtliche Protokolldaten werden über `System.out` ausgegeben.

Bei der Initialisierung des Brokers wird die Datei „Broker.conf“ eingelesen, in der sich die Parameter für den Broker und die Liste der verfügbaren Übersetzer befinden. Die Datei hat die Form:

```
<Broker.conf>::=    {<BrokerInfo>}{<ÜbersetzerInfo>}*
```

[36] Eine Einführung in das Client-Server-Modell findet sich zum Beispiel in [Borghoff Schlichter 95] Seite 17 ff

```
<BrokerInfo>::=          ≤Broker≥
                              {⌊Port⌋=<String>}
                              {⌊Path⌋=<String>}
                              {⌊SubPath⌋=<String>}
                         ≤/Broker≥
<ÜbersetzerInfo>::= ≤Übersetzer≥
                              ⌊Name⌋=<String>
                              ⌊IPAdr⌋=<String>
                              ⌊Port⌋=<String>
                              {⌊SubPath⌋=<String>}
                              {⌊Info⌋=<String>}
                              {⌊URL⌋=<String>}
                         ≤/Übersetzer≥
<String>::=    String of char
```

In `BrokerInfo` ist `Port` die Portnummer, unter der der Broker installiert werden soll, `Path` der Basispfad, in dem der Broker alle zu erzeugenden HTML - Seiten ablegen soll, `SubPath` ein optional anzugebendes Unterverzeichnis für Dokumente, die keinem Übersetzer zugeordnet sind.

Die aufgeführten Übersetzer werden in einer Liste im Broker mit ihrem Namen, der IP-Adresse und Portnummer, unter der sie erreicht werden können gespeichert. Weiterhin können für jedes Archiv noch der relative Pfad `SubPath`[37], unter dem die Ergebnisse des Übersetzers gespeichert werden sollen, ein Informationstext `Info` zu dem jeweiligen Archiv und eine URL mit einer archivspezifischen HTML-Seite, angegeben werden.

Nach dem Einlesen der Konfigurationsdatei „Broker.conf“ startet der Broker einen Serverthread, der einen Server unter der in dieser Datei als Parameter eingetragenen Portnummer beim System anmeldet und anschließend auf Anfragen wartet.

[37] relativ zum Basispfad „`Path`“ des Brokers

Erhält der Serverthread eine Anfrage, so startet er einen neuen Servicethread, übergibt diesem die zu der Kommunikation notwendigen Socketdaten und steht für die nächste Anfrage zur Verfügung.

Der Servicethread analysiert und bearbeitet die eintreffende Anfrage. Im Prototyp werden drei verschiedene Anfragetypen unterstützt.

- Ermittle verfügbare Datenbanken
- Recherche
- Lade Dokument

Der Aufbau dieser Anfragen wurde bereits in Kapitel 5.1.2 *Anfrageobjekt* dargestellt. Im folgenden sollen nun die wesentlichen Aktionen des Brokers beschrieben werden, die bei ihrer Bearbeitung erfolgen.

Ermittle verfügbare Datenbanken:

Der Broker erzeugt eine Liste der für den Anwender verfügbaren Datenbanken und sendet sie an das Anfrageobjekt. Hierbei werden die Parameter `Name`, `Info`, und `URL` aus der Datei Broker.conf in eine für das Treeapplet verwertbare Form gebracht.

Im Prototyp erhält der Anwender mehr als die ihm tatsächlich zur Verfügung stehenden Übersetzer aufgelistet. Ihm soll durch die Liste ein Beispiel für einige an das Recherchesystem anschließbare Archive gegeben werden. Tatsächlich wurden im Prototyp nur die Übersetzer für AltaVista, MUSAD und REGIS realisiert.

Recherche:

Der Servicethread liest die vollständige Anfrage ein. Anschließend erzeugt er für jeden angeschlossenen Übersetzer einen Kommunikationsthread, wartet bis alle Threads ihre Arbeit abgeschlossen haben, schließt die Kommunikation mit dem Anfrageobjekt ab und beendet sich schließlich selbst.

Die Kommunikationsthreads arbeiten voneinander unabhängig und senden ihre Ergebnisse über eine vom Servicethread bereitgestellt synchronisierte Funktion[38] direkt an das Anfrageobjekt. Auf diese Weise treffen beim Anwender die ersten

[38] Mit Hilfe des `synchronized`-Operators lassen sich in Java sehr einfach Monitore realisieren. Der wechselseitiger Ausschluß der Kommunikationsthreads beim Zugriff auf die Socketverbindung zum Anfrageobjekt ist für die Informationsübertragung zwingend notwendig. Bei der Realisierung der Kommunikationsthreads wurde darauf geachtet, diesen Monitor so kurz wie möglich zu belegen, um eine maximale Nebenläufigkeit der einzelnen Threads zu gewährleisten. Siehe hierzu auch [Tanenbaum 95] Seite 58 ff.

Ergebnisse ein, sobald sie beim Broker bereitliegen und nicht erst, wenn die Recherche auf allen Archiven abgeschlossen ist.

Die einzelnen Kommunikationsthreads bauen eine Socketverbindung zu dem jeweiligen Übersetzer auf, senden ihm die Anfrage zusammen mit der Kennung des Anwenders und warten auf die Ergebnisse. Sie treten gegenüber den Übersetzern als Clients auf.

Die Ergebnisse sind aufgebaut wie folgt:

```
<Ergebnis> ::=      {<htmlText>|<TocItem>}*

<htmlText>::=  [htmlText]=<String>

<TocItem>::=   [TocTitle]=<String>

               {[TocSubTitle]=<String>}

               {[TocHTMLLabel]=<String>}

               [TocHTMLAdr]=String>

<String>::=    String of char
```

Aus den Einträgen in `<htmlText>` baut der Kommunikationsthread eine HTML-Seite auf, die später von der Rechercheoberfläche in die Ergebnisanzeige geladen werden kann. Als Namen erhält sie den Namen des Übersetzers, versehen mit einer eindeutigen Nummer. In der Regel handelt es sich im Prototyp bei dieser Seite um eine in HTML aufbereitete Trefferliste, eventuell mit einigen Zusatzinformationen oder Fehlermeldungen.

`[TocHTMLLabel]` stellt einen Verweis auf einen Eintrag in dieser HTML-Seite dar. Er wird als `[URL1]` in die Trefferliste für das Anfrageobjekt eingetragen und ist somit später für den Anwender durch Einfachklick auf den entsprechenden Eintrag im Trefferbaum zu erreichen. Der Name des Eintrags steht hinter dem Parameter `[TocTitle]`.

`[TocHTMLAdr]` wird als `[URL2]` in die Trefferliste eingetragen. Sie kann von dem Anwender im Trefferbaum der Rechercheoberfläche mittels Doppelklick erreicht werden. In der Regel stellt sie eine URL-Adresse des jeweiligen Treffers oder einen eindeutigen Identifikationsstring dar, mit dessen Hilfe das Dokument vom Anfrageobjekt über den Broker beim Übersetzer angefordert werden kann.

`[TocSubtitle]` ist die Zusatzinformation zu jedem Treffer, die erscheint, sobald der Mauszeiger eine gewisse Zeit über dem entsprechenden Baumeintrag im Trefferbaum der Rechercheoberfläche verweilt.

Der Kommunikationsthread sammelt alle Ergebnisse des zu ihm gehörenden Übersetzers. Nach dem letzten Eintrag wird die Verbindung zu diesem abgebaut, die Trefferliste an das Anfrageobjekt gesendet und der Thread beendet.

Lade Dokument:

Das Laden eines Dokumentes über den Broker erfolgt gemäß dem in Abbildung 25 dargestellten Diagramm.

Wird ein Dokument mit seinem Identifikationsstring[39] vom Anfrageobjekt beim Broker angefordert (1), so prüft dieser zunächst, ob das Dokument bereits aufgrund einer früheren Anfrage lokal gespeichert ist. In diesem Fall entfallen die Schritte 2-5 und die URL des geforderten Dokumentes wird an das Anfrageobjekt übermittelt (7). Im anderen Fall wird eine Verbindung mit dem für das geforderte Dokument zuständigen Übersetzer aufgebaut und das Dokument angefordert(2). Der Übersetzer fordert das Dokument bei der ihm zugeordneten Datenbank an (3) und schickt das erhaltene Dokument (4) zum Broker (5).

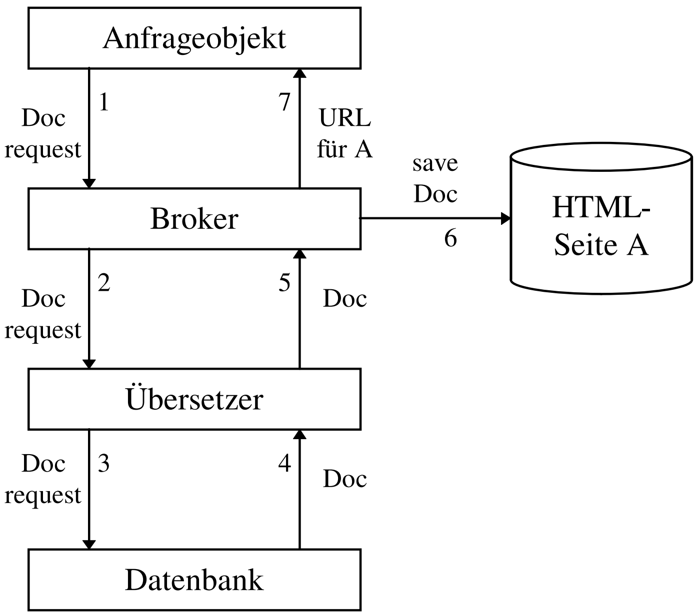

Abbildung 25: Datenfluß beim Laden eines Dokuments

Das Dokument wird lokal beim Broker als HTML-Dokument im Verzeichnis `Path + SubPath`[40] gespeichert (6). Als Namen erhält es dem Namen des

[39] dabei handelt es sich <u>nicht</u> um eine URL-Adresse, da diese bereits vom Anfrageobjekt ohne Zuhilfenahme des Brokers geladen wird, sondern um einen String mit dem das Dokument von einem Übersetzer angefordert werden kann. Wie sich dieser String zusammensetzen kann, wird zum Beispiel in Kapitel 5.2.2 auf Seite 75 beschrieben.

[40] Dies ist der `SubPath`, wie er in der `ÜbersetzerInfo` des Übersetzers und der `Path`, wie er in `BrokerInfo` in der Datei Broker.conf angegeben wurde.

Übersetzers konkateniert mit dem Identifikationsstring[41] des angeforderten Dokuments.

Die relative URL-Adresse (zum Basispfad des Brokers) des HTML-Dokuments wird als `[htmlErgebnis]` an das Anfrageobjekt geschickt (7) und kann dann von diesem in die Ergebnisanzeige geladen werden.

Bei hostbasierten Datenbanken wie MUSAD oder REGIS führt das Speichern der HTML-Dokumente beim Broker zu einem erheblichen Geschwindigkeitsgewinn, falls dasselbe Dokument wiederholt angefordert wird.

5.1.4 Übersetzer

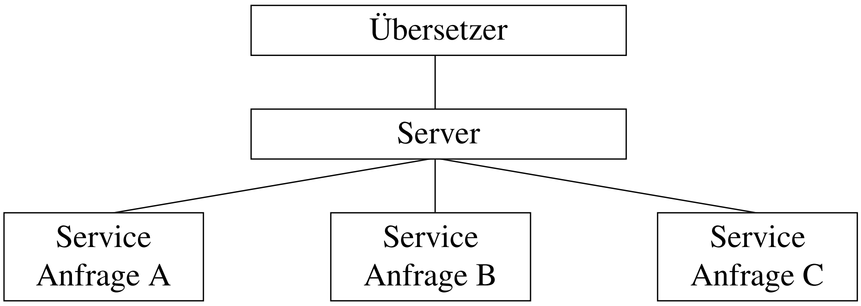

Abbildung 26: Threadkonzept der Übersetzer

Alle Übersetzer wurden, analog zum Broker, multithreaded programmiert. Auch hier wartet eine Serverthread und startet, sobald er eine Anfrage erhalten hat, einen Servicethread. Dieser analysiert die Anfrage und erledigt die weitere Kommunikation mit dem Broker und der angeschlossenen Datenbank. Der Serverthread steht sofort wieder für die nächste Anfrage zu Verfügung. Somit kann ein Übersetzer theoretisch beliebig viele Anfragen gleichzeitig bearbeiten.

Eine genauere Beschreibung der Funktionsweisen der einzelnen im Prototyp realisierten Übersetzer liefert das nächste Kapitel:

5.2 Anbindung von bestehenden Datenbanken

Für die Anbindung an den Prototyp wurden drei Datenbanken ausgewählt:

- AltaVista

[41] Da der Identifikationsstring des Dokuments Zeichen enthalten kann, die in einem Dateinamen unter Umständen nicht erlaubt sind, müssen diese vor dem Speichern noch durch Leerzeichen ersetzt werden. Diese Zeichen sind \ ? : * " < > | / [ ]

- SAD-Datenbanken
- REGIS

Auf diese Weise kann exemplarisch gezeigt werden, daß sowohl Recherchen auf hostbasierenden Systemen, als auch auf einem Client-Server System, gleichzeitig und für den Benutzer transparent, innerhalb eines Web-Browsers möglich sind.

5.2.1 Internetsuchmaschine AltaVista

Bei AltaVista handelt es sich um eine der bekanntesten Suchmaschinen im WWW.

AltaVista recherchiert durch Suchroboter im WWW. Über alle von diesen gefundenen Seiten kann der Benutzer nun Volltextrecherchen durchführen. Dabei können die Suchbegriffe durch AND, OR, NOT, NEAR und Klammern miteinander verknüpft werden. Weiterhin können der Suchbereich noch zeitlich eingegrenzt und die Treffer - nach Häufigkeit der gefundenen Suchbegriffe oder Datum sortiert - ausgegeben werden.

Die Anfrage wird als CGI-Anfrage[42] abgeschickt. Die Trefferlisten werden als HTML-Dokumente zurückgegeben. In der Regel enthält eine HTML-Seite bis zu zehn Treffer und, falls vorhanden, Verweise auf die Fortsetzung der Liste. Eine Begrenzung der Anzahl der Treffer auf eine bestimmte Obergrenze findet nicht statt.

Der in diesem Prototyp realisierte Übersetzer wandelt die Anfrage der Rechercheoberfläche in eine CGI-Anfrage um. Da es sich bei AltaVista um eine - bis auf das Datumsfeld - reine Volltextdatenbank handelt, geht hierbei die semantische Information der einzelnen Attributfelder der Rechercheoberfläche verloren. Als Sortierkriterium für die Trefferliste werden die Suchbegriffe angegeben, um eine Sortierung nach der Häufigkeit der gesuchten Schlagworte zu erreichen. Eine weitere Möglichkeit wäre noch die Sortierung nach dem Datum. Es ist allerdings zu beachten, daß es sich bei dem Datumsfeld in AltaVista um das Speicherdatum des HTML-Dokuments handelt und nicht um das Entstehungsdatum des in dieser Seite gespeicherten Dokuments. Auf diese Problematik wird unten bei der Beschreibung einiger Testrecherchen noch weiter eingegangen.

Erhält der Übersetzer nun die erste Seite der Trefferliste seiner Anfrage an AltaVista, so liest er die Kurzinformationen der einzelnen Treffer aus diesem HTML-Dokument, übermittelt an den Broker die relevanten Daten und fordert bei

[42] CGI (Common Gateway Interface)

AltaVista, falls vorhanden, die nächste Trefferseite an, um mit dem Auslesen der Trefferliste fortzusetzen, bis entweder das Ende der Liste oder die maximale Trefferzahl erreicht ist. Diese wird entweder vom Benutzer in der Eingabemaske angegeben, oder defaultmäßig vom Übersetzer beschränkt[43].

An den Broker werden für jeden Treffer als Informationen für den Ergebnisbaum der Rechercheoberfläche die URL-Adresse, der Titel der entsprechenden HTML-Seite und die erste Textzeile dieser Seite übermittelt. Hierbei werden noch die Umlaute vom HTML-Format in Java-Zeichenkode gewandelt. Weiterhin wird die komplette Kurzinformation zu jedem Treffer zur Erstellung einer Trefferliste im HTML-Format an den Broker gesendet.

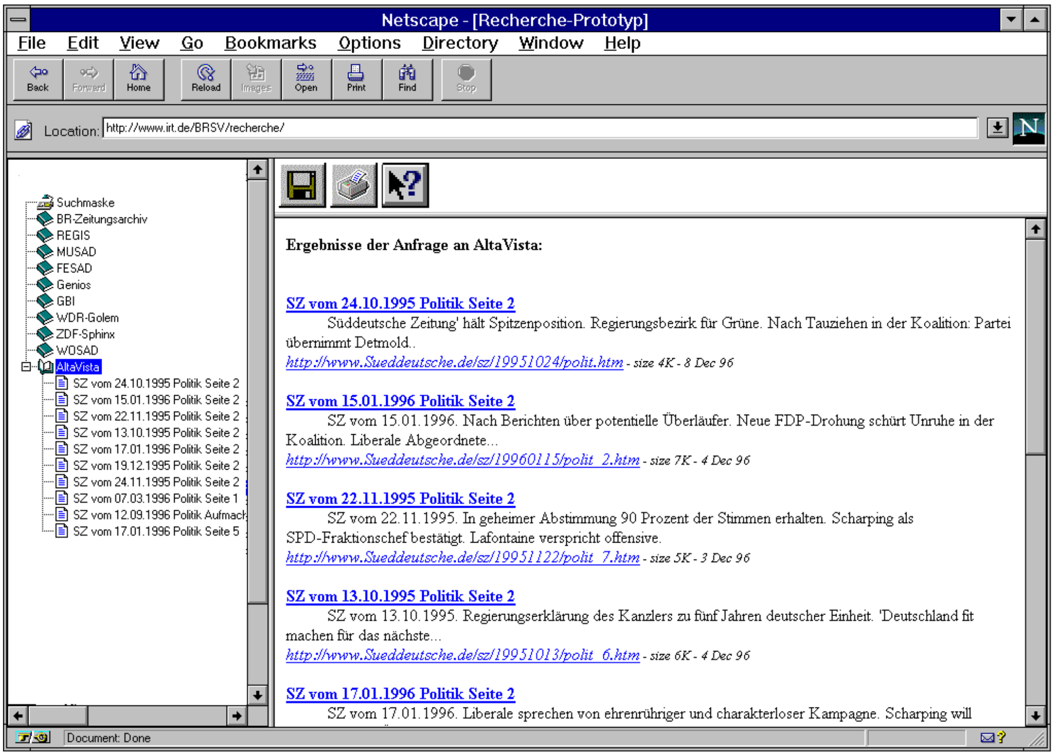

Abbildung 27 Trefferliste von AltaVista

Dieser erzeugt aus den Kurzinformationen ein HTML-Dokument mit einer Übersicht über alle ermittelten Treffer und leitet dessen URL zusammen mit den Informationen für den Trefferbaum an das Anfrageobjekt weiter. Dort werden diese Treffer von dem Anfrageobjekt in den Ergebnisbaum eingetragen.

[43] Im Prototyp wurden 100 Treffer als Obergrenze gewählt

Der Benutzer erhält nun durch Einfachklick den entsprechenden Eintrag der Kurzinformation in der vom Broker erzeugten Trefferübersicht im Dokumentenfenster dargestellt(siehe Abbildung 27). Durch Doppelklick auf den entsprechenden Eintrag im Trefferbaum oder anklicken der URL-Adresse in der Trefferübersicht wird das angeforderte Dokument in das Anzeigenfenster geladen (siehe Abbildung 28). Aus Effizienzgründen erfolgt dies nicht mehr über den Broker, sonder es wird vom Trefferbaum über den Java-Befehl `java.applet.AppletContext. showDocument(URL url, String target)` direkt vom Trefferbaum oder beim Anwählen des Links in der Trefferübersicht durch den Webbrowser geladen.

Abbildung 28 Ergebnisanzeige von AltaVista

Möchte ein Anwender direkt auf AltaVista recherchieren, so erhält er durch Doppelklick auf das AltaVista-Symbol im Trefferbaum die Originaloberfläche von AltaVista im Dokumentenfenster dargestellt und er kann mit ihr auf dieser Datenbank recherchieren. In diesem Fall erfolgt die Kommunikation mit AltaVista ohne die unter der Benutzerschicht liegenden Module des Recherchesystems.

5.2.2 SAD-Datenbanken beim BR

In diesem Abschnitt wird, anhand eines hostbasierenden Systems vorgeführt, daß der Zugriff auf diese Systeme, den die Anwender bisher meist nur über 3270-

Terminals oder 3270-Terminalemulationen durchführen konnten, ohne Änderungen am Hostsystem, über die oben vorgestellte einheitliche Benutzeroberfläche möglich ist.

Die Anbindung der hostbasierenden SAD Datenbanken wird in diesem Prototyp beispielhaft an MUSAD durchgeführt. MUSAD ist das aktuelle Musikarchiv des Bayerischen Rundfunks. Der Zugang erfolgt im Augenblick durch 3270-Terminals über ein Token-Ring-Netzwerk. Als Netzwerkprotokoll wird SNA verwendet. Da der hier realisierte Prototyp jedoch TCP/IP voraussetzt, wurde vom Rechenzentrum des BR ein Gateway von SNA nach IP, mit Hilfe von Novell SAA Server[44], eingerichtet. Dieser Gateway stellt sich für den Prototyp als TN3270-Server dar.

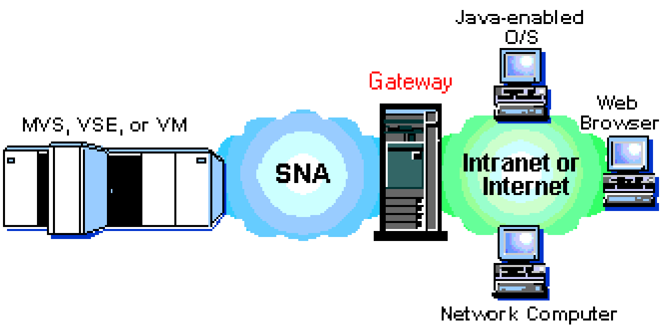

Abbildung 29: SNA-Anbindung[45]

Für die Recherche sollen prinzipiell zwei Möglichkeiten des Zugriffs möglich sein.

Zum einen der Zugang über die für alle Datenbanken einheitliche Benutzeroberfläche. Hier muß also die Struktur der Datenbank für den Benutzer transparent bleiben, die in der Anfragemaske der Rechercheoberfläche eingegebene Anfrage muß vom System bestmöglich in eine MUSAD-Anfrage umgewandelt werden.

Zum anderen muß es für den Benutzer möglich sein, auf Wunsch und wenn es seine Nutzungsrechte zulassen, über eine 3270-Terminalemulation innerhalb des Webbrowsers, direkt auf MUSAD zu recherchieren.

Die Anbindung von MUSAD an den Prototyp erfolgt somit auf zwei Arten:

---

[44] Andere Produkte mit denen dies möglich gewesen wäre sind zum Beispiel:
- IBM Communications Server with Host On-Demand
- SUN Netra j

[45] Grafik aus http://www.networking.ibm.com/hex/white_paper_en.html

- Durchschalten der Terminalemulation
- Kommunikation mit MUSAD durch einen Übersetzer

Beide Varianten wurden im Rahmen der Diplomarbeit in den Prototyp des Recherchesystems implementiert.

Durchschalten der Terminalemulation

In diesem Modus wird innerhalb des Browsers eine 3270-Terminalemulation durchgeführt. Der Prototyp verwendet hierfür ein für Forschungszwecke frei im Internet erhältlichen TN3270-Terminalemulator, der als Java-Applet realisiert wurde und damit in einem Web-Browser ausgeführt werden kann. Allerdings mußte dieser Emulator erst noch umfangreich modifiziert werden, da viele der von MUSAD benötigten 3270-Funktionen nicht oder nur fehlerhaft unterstützt wurden.

Der TN3270-Emulator verwendet für die Verbindung zum Host die Socket-Klasse `java.net.socket` und setzt somit das TCP/IP-Protokoll voraus. Zusätzlich erlauben die Sicherheitsrestriktionen in einem Web-Browser, wie Netscape oder Microsoft Internetexplorer, einem Applet nur eine Socketverbindung zu dem Web-Browser aufzubauen, von dem es heruntergeladen wurde.

Aus diesen Gründen baut auch die Terminalemulation innerhalb des Webbrowser eine Verbindung zum Broker auf, der jedoch in diesem Fall nur einen Kommunikationskanal zu dem entsprechenden TN3270-Server durchschaltet. Der Anwender erhält daraufhin ein in ein HTML-Dokument eingebettetes 3270-Terminal und kann sich, falls er eine Zugangskennung auf eine der SAD-Datenbanken besitzt, in diese einloggen.

In der Endversion des Recherchesystems sollten vom Broker noch die Zugriffsrechte geprüft und die Login-Prozedur in MUSAD automatisch durchgeführt werden. Weiterhin könnte eine vom Benutzer eventuell schon in der Standardoberfläche ausgefüllte Anfrage optional in die MUSAD-Oberfläche eingesetzt werden. Diese Punkte wurden in diesem Prototyp nicht mehr implementiert.

Kommunikation durch einen Übersetzer

Auch in diesem Modus wird eine TN3270-Terminalemulation in Java durchgeführt. Allerdings findet diese, für den Benutzer nicht sichtbar, auf Übersetzerebene statt. Hier war ein fehlerfreies Funktionieren der Terminalemulation besonders wichtig, da der Übersetzer in der Regel weniger flexibel auf Fehler in der Emulation reagieren kann als ein Anwender.

Der Übersetzer verwaltet sämtliche Ein- und Ausgaben auf dem 3270-Terminal. Für den Anwender sind sie nicht sichtbar.

Um bei Anfragen unnötige Wartezeiten zu sparen, meldet sich beim Start des Übersetzers ein eigener Kommunikationsthread sofort bei MUSAD an. Hierzu benötigt er eine eigene Zugangskennung und ein Paßwort. Anschließend durchläuft er mehrere Eingabemasken, bis er die Recherchemaske von MUSAD erreicht hat. An dieser Stelle wartet er dann auf Anfragen.

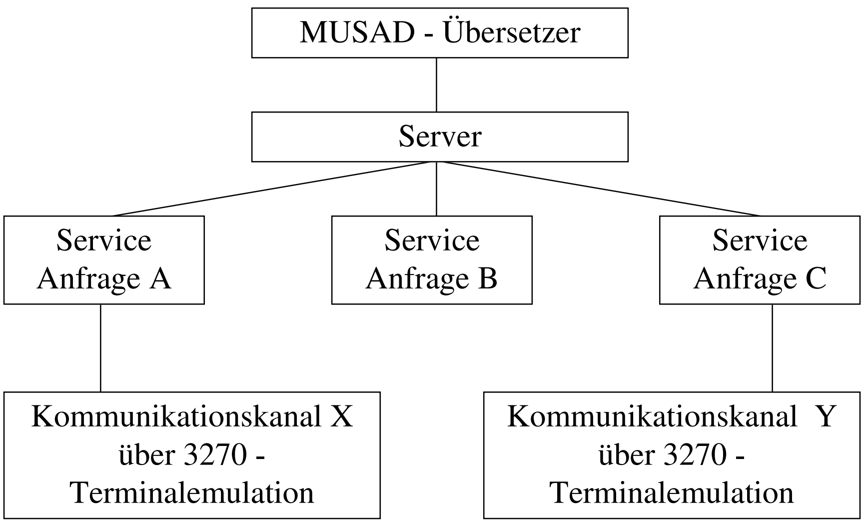

Abbildung 30: Threadkonzept für MUSAD

Der Übersetzer muß es ermöglichen, daß mehrere Anwender auf der gleichen Zugangskennung quasi parallel auf MUSAD recherchieren. Für den Prototyp im Rahmen der Diplomarbeit stand zum Beispiel nur eine Kennung für MUSAD zur Verfügung. Mit Hilfe des `synchronized` Statements in Java konnte hier auf einfache Art und Weise eine Verwaltung dieser begrenzt zur Verfügung stehenden Kommunikationskanäle implementiert werden.

Erhält der MUSAD-Übersetzer eine Anfrage vom Broker, so startet analog zum Altavista-Übersetzer ein neuer Thread, der die Anfrage bearbeitet. Sind zu diesem Zeitpunkt alle Verbindungen zu MUSAD belegt, so wird dieser Thread blockiert, bis ein Kommunikationskanal zu MUSAD frei geworden, also eine andere Anfrage abgearbeitet worden ist. In der Abbildung 30 gilt dies zum Beispiel für den Servicethread B. Er bleibt blockiert, bis einer der beiden Kommunikationskanäle X oder Y frei geworden ist.

Der Übersetzer selbst steht jedoch sofort wieder für neue Anfragen des Brokers zur Verfügung.

In die Felder des im Übersetzer simulierten und für den Benutzer nicht sichtbaren 3270-Terminals werden schließlich die Daten der Anfrage eingetragen und an MUSAD abgeschickt. Hierzu wird die in MUSAD optional zur Verfügung stehende STAIRS-Eingabezeile verwendet.

Wichtig ist hierbei eine plausible, das heißt eine semantisch möglichst korrektes Umsetzung der in der Oberfläche angebotenen Suchattribute in die entsprechenden Anfrageattribute in MUSAD. Für den Prototyp wurde folgende Übersetzung gewählt:

Attributname Anfragemaske	Attributname MUSAD	Attributname STAIRS
Person	Urheber, Produktion, Mitwirkung	.UPM.
Schlagwort	Abstract	.ABS.
Titel	Titel	.TITEL.
Quelle	Tonträger	.BTT.

Eine Anfrage der Form:

Person=Bach&Richter&

Schlagwort=Klavier&

Titel=Partita&

Quelle=CD

würde übersetzt in die STAIRS-Anfrage („BACH“.UPM. AND „RICHTER“.UPM. AND „KLAVIER“.ABS AND „PARTITA“.TITEL AND

„CD".BTT.) und unter anderem alle CD-Aufnahmen der Partiten von J.S. Bach mit S. Richter als Pianisten liefern[46].

Somit ist in MUSAD eine relativ schlüssige Übernahme der Attributinformationen aus der Rechercheoberfläche in die tatsächlich durchgeführte Recherche möglich. Es sei noch angemerkt, daß das Feld Urheber sowohl Komponisten als auch Interpreten abdeckt.

Eine Verfeinerung dieser Tabelle wird mit Sicherheit im Laufe des Betriebes des Recherchesystems noch erfolgen. Sie wird auf den dann gesammelten Erfahrungen basieren, ist jedoch nicht mehr Gegenstand dieser Diplomarbeit.

Die Ergebnisliste der Anfrage an MUSAD wird anschließend bis zu der vom Benutzer angegebenen maximalen Trefferzahl ausgelesen. Die Kurzinformationen der einzelnen Treffer werden, zusammen mit einer Nummer, unter der sie später angefordert werden können, an den Broker gesendet. Diese Nummer setzt sich zusammen aus der Archivnummer des Tonträgers[47], der Abspieldauer des Dokuments und einer Take-Nummer - dies ist die Aufnahmenummer des Dokuments auf dem Tonträger. Mit diesen drei Angaben kann der Übersetzer später das Dokument eindeutig auf MUSAD recherchieren.

Wurde die Trefferliste fertig ausgelesen und an den Broker übermittelt, so wird die Verbindung zu diesem geschlossen, der Kommunikationsthread des Übersetzers kehrt zurück zur Recherche-Eingabemaske von MUSAD, löscht die Eingaben der aktuellen Anfrage und ist bereit für die nächste.

Ein Beispiel für eine Trefferliste und ihre Darstellung in der Rechercheoberfläche zeigt Abbildung 31.

Erfolgt nun vom Broker eine Anforderung nach einem bestimmten Dokument mit Hilfe der oben beschriebenen Identifikationsnummer, so wird mit dieser Nummer analog zu oben über das interne 3270-Terminal eine Recherche in Form einer STAIRS-Anfrage durchgeführt. Zu dem entsprechenden Dokument wird die Vollinformation ausgelesen, in HTML konvertiert und an den Broker weitergeleitet. Da MUSAD noch keine Audio-Daten zur Verfügung stellt, handelt sich hierbei um reine Textdokumente[48].

[46] Tatsächlich wurden bei dieser Recherche auch keine anderen Treffer geliefert. Es wäre durchaus denkbar gewesen, hier auch noch Partiten von zum Beispiel E. Bach zu erhalten.
[47] Beispiele für Tonträger sind Schallplatte, Tonband, CD
[48] Ein Beispiel für ein solches Dokument zeigt Abbildung 32.

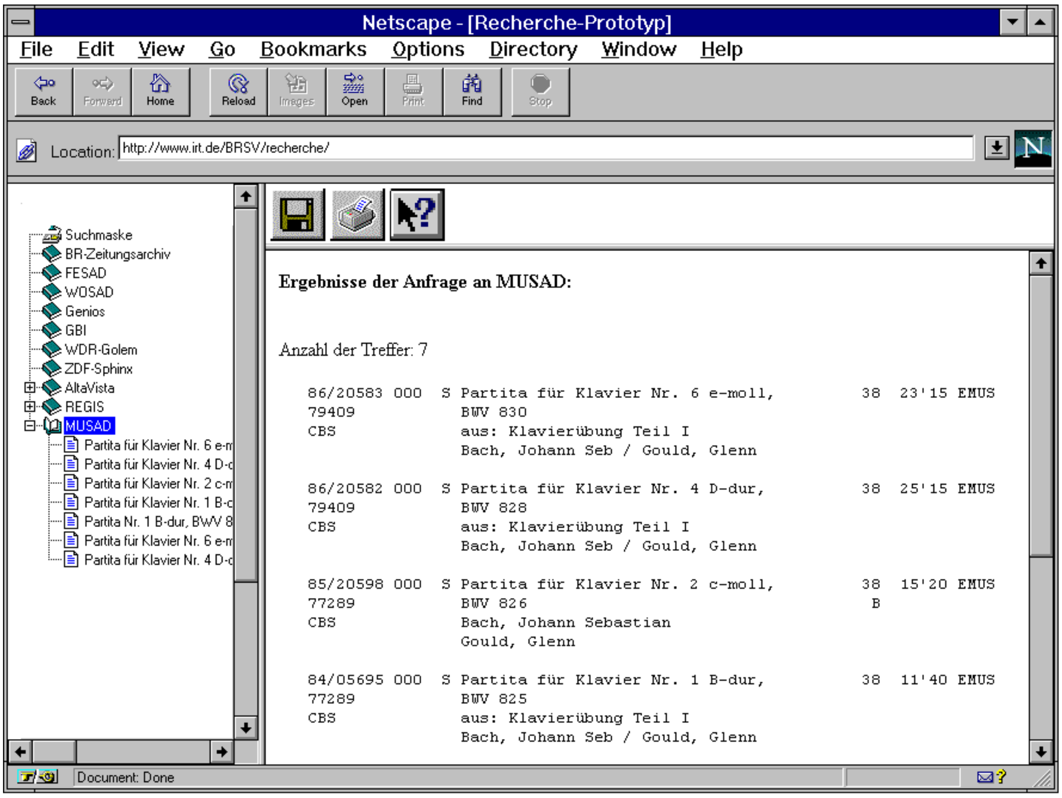

Abbildung 31: Trefferliste von MUSAD

Da in MUSAD die Take-Nummer nicht recherchiert sondern nur ausgelesen werden kann, kommt es jedoch vor, daß der Übersetzer auf die Anfrage nach der gesuchten Archivnummer und der gesuchten Abspieldauer, mehr als einen Treffer erhält. In diesem Fall muß er alle gefundenen Kurzinformationen überprüfen, bis er das Dokument mit der richtigen Nummer erhält. Insgesamt tritt dieser Fall, daß sich auf einem Tonträger zwei oder mehr Aufnahmen mit exakt der gleichen Dauer befinden, eher selten auf, mußte jedoch berücksichtigt werden. Findet eine Eingrenzung der Treffermenge mit der zur eindeutigen Identifizierung eigentlich nicht notwendigen Abspieldauer nicht statt, so kann es zu erheblichen Verzögerungen kommen. Tonträger mit 20 oder sogar über hundert Aufnahmen sind keine Seltenheit und müßten vom Übersetzer erst analysiert werden. Die Abspieldauer hat sich hier als sehr kompaktes und wirkungsvolles Kriterium erwiesen.

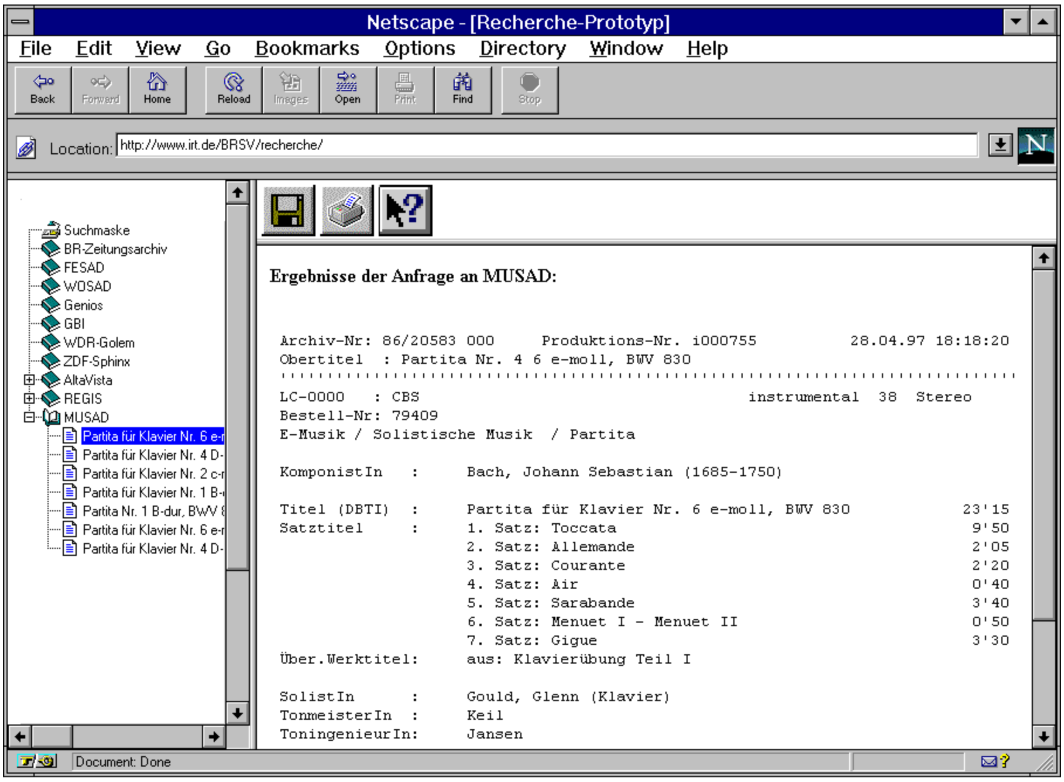

Abbildung 32: Dokument von MUSAD

5.2.3 REGIS-Datenbank der SZ

Für eine reine Demonstration der Fähigkeit des in dieser Arbeit konzipierten Recherchearbeitsplatzes, hostbasierende Datenbanken via 3270-Terminal anzusprechen, wäre die Anbindung einer weiteren Datenbank dieses Typs an den Prototyp nicht mehr unbedingt notwendig gewesen. Erschwerend für eine Umsetzung kommt hinzu, daß REGIS eine wesentlich komplexere Rechercheoberfläche zur Verfügung stellt als zum Beispiel MUSAD.

Außerdem läßt sich unter REGIS für einen maschinellen Übersetzer zum Teil nur schwer herausfinden, ob zum Beispiel eine Recherche nun mit null Treffern beendet ist oder sich noch in der Ausführung befindet. REGIS stellt hier im Unterschied zu MUSAD keine eindeutige Bildschirmausgabe zur Verfügung.

Allerdings stellt das umfangreiches Zeitungsarchiv REGIS eine erhebliche Bereicherung für den Prototyp dar. Dies gilt erst recht im Hinblick auf die später mit dem Prototypen durchzuführenden Testrecherchen.

Im Prinzip unterscheidet sich die Anbindung von REGIS an den Prototyp nicht wesentlich von der Anbindung von MUSAD. Das Durchschalten der

Terminalemulation für den fortgeschrittenen Anwender erfolgt sogar exakt identisch.

Die Kommunikation über einen Übersetzer verläuft ähnlich zu der in MUSAD. Auch in REGIS meldet sich der Übersetzer bereits beim Starten des Programmes an und verbleibt, wenn er die Rechercheoberfläche von REGIS erreicht hat in einem Wartezustand, beziehungsweise führt dann bereits schon anstehende Anfragen aus.

Auch in REGIS wäre eine direkte Umsetzung der Attributfelder der Anfragemaske in entsprechende Attribute der Datenbank möglich. REGIS liefert bei nicht eindeutiger Angabe zum Beispiel der Person eine Auswahlliste der bekannten Personen, die zu bearbeiten für den Übersetzer relativ aufwendig wäre und in dem knappen Zeitrahmen einer Diplomarbeit nicht durchgeführt werden konnte. Um diese Auswahllisten zu umgehen, verwendet der Übersetzer in diesem Prototyp nur die Attribute ARTIKELTEXT und ERSCH.-TAG.

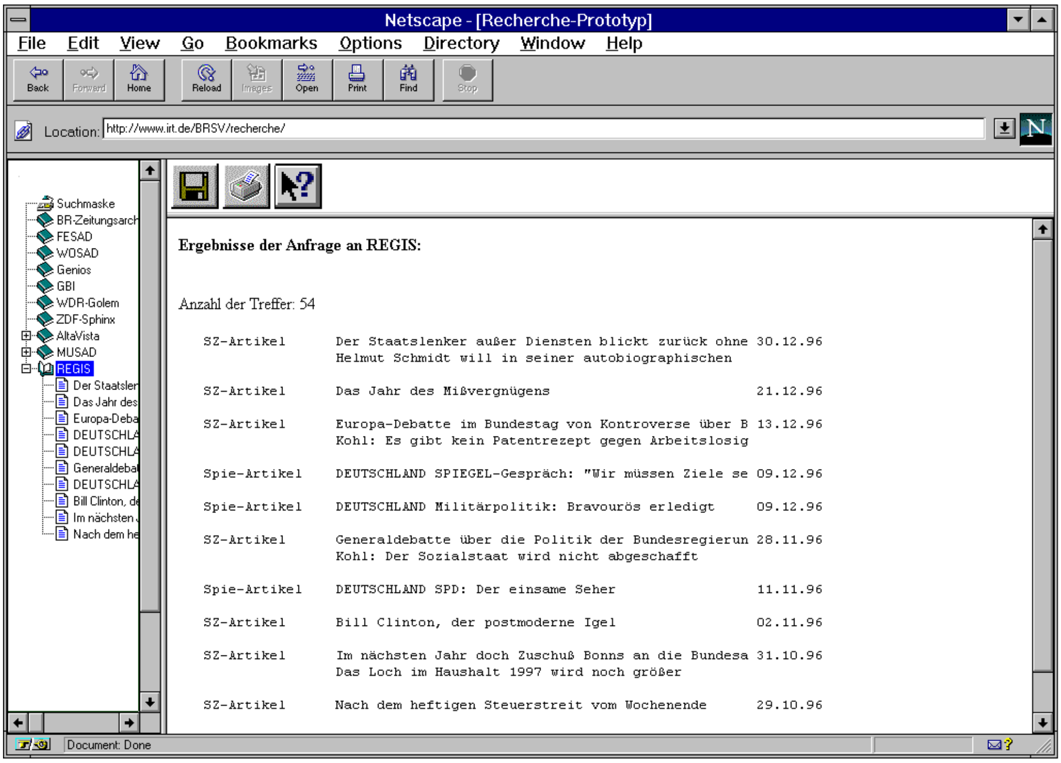

Abbildung 33: Trefferliste von REGIS

Analog zu MUSAD werden die Kurzinformationen vollständig oder bis zu einer maximalen Trefferanzahl ausgelesen und an den Broker weitergeleitet[49]. Zur späteren Identifikation der einzelnen Artikel dient hier die Überschrift und das

[49] Abbildung 33 zeigt ein Beispiel für die Darstellung einer Trefferliste von REGIS in der Ergebnisanzeige.

Datum des Artikels. Eine eindeutige Identifikationsnummer steht leider nicht zur Verfügung. Weiterhin muß die Überschrift noch bis zu dem letzten vollständige Wort vor dem 42. Buchstaben gekürzt werden, da längere Rechercheeingaben in REGIS nicht möglich sind. Mit Hilfe dieser Überschrift und des Erscheinungstags kann, analog zu MUSAD, nun von dem Anwender über den Trefferbaum der Rechercheoberfläche der vollständige Artikeltext angefordert werden. Hierbei werden durch den Übersetzer auf dem Feld ÜBERSCHRIFT und ERSCH.-TAG die entsprechenden Recherchen durchgeführt. Die Wahrscheinlichkeit über diese Recherche zwei inhaltlich verschiedene Artikel zu erhalten wurde von mehreren Mitarbeitern im Archiv des Süddeutschen Verlages als „praktisch null“ bezeichnet. Tatsächlich ist dieser Fall bisher auch noch nicht aufgetreten. Allerdings wurden auf diese weise schon zwei gleiche, beziehungsweise zwei ähnliche Artikel gefunden. Zum Beispiel wenn der selbe Artikel in zwei verschiedenen Regionalteilen der SZ erschienen ist. Im Rahmen dieses Prototyps übermittelt der Übersetzer einfach einen der beiden Artikel als Ergebnis an den Broker. Eine befriedigende Lösung für REGIS kann hier jedoch erst gefunden werden, wenn REGIS einen eindeutigen Identifikationsschlüssel für die einzelnen Dokumente zur Verfügung stellt.

Eine graphische Aufbereitung der Vollinformation durch den Übersetzer findet im Rahmen dieses Prototyps nur sehr begrenzt statt. Die vor jedem Artikel von REGIS gelieferten Informationen zu dem Artikel, wie zum Beispiel Ressort, Autor, Erscheinungstag werden in nichtproportionaler Schrift gesetzt, die Überschriften und der Artikeltext selbst in proportionaler Schrift. Auf den von einigen Anwendern in einer Endversion geforderten Spaltensatz wurde hier noch verzichtet. Eine Eindeutige Erkennung der Überschriften durch den Übersetzer war nicht möglich, daher wird sie in der gleiche Schriftart und Größe dargestellt wie der restliche Artikeltext.

Das so erzeugte HTML-Dokument übermittelt der Übersetzer an den Broker. Dort wird es gespeichert und kann dann vom Anfrageobjekt in die Ergebnisanzeige geladen werden (Abbildung 34).

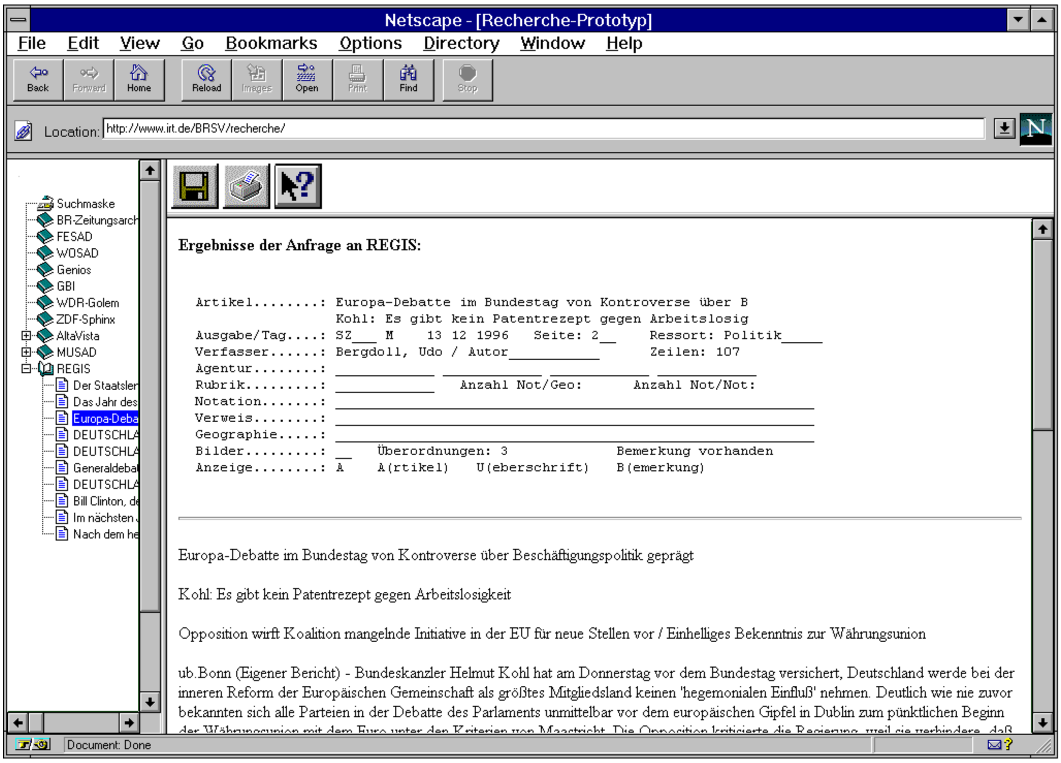

Abbildung 34: Dokument von REGIS

5.3 Testrecherchen am Prototyp

Mit dem Prototyp sollen umfangreiche Testrecherchen mit Redakteuren und Archivmitarbeitern des Bayerischen Rundfunks und des Süddeutschen Verlages durchgeführt werden.

Sie haben die Analyse folgender Punkte zum Ziel:

- Akzeptanz durch Anwender
- Überprüfung der bereitgestellten Suchattribute
- Verfeinerung der Übersetzer
- Praxistauglichkeit
- allgemeine Verbesserungsvorschläge
- Performance

Für Testrecherchen steht der Prototyp unter *http://www.irt.de/BRSV/recherche/* zur Verfügung. Beim Laden der Oberfläche werden Kennung und Paßwort abgefragt. Für nicht registrierte Anwender steht der volle Umfang der Oberfläche zur Verfügung, auf Übersetzerebene wird der Zugriff für nicht registrierte Anwender jedoch verweigert. Eine Ausnahme stellt der Übersetzer von AltaVista dar. Eine Recherche auf dieser Suchmaschine ist für alle Anwender möglich.

Da für viele Redakteure noch kein Internetzugang besteht, wurde eine Zugriffsmöglichkeit auf den Web-Server des IRT über ISDN realisiert.

Sämtliche Recherchen werden von dem System auf Übersetzer- und Brokerebene zur späteren Analyse protokolliert.

Im Rahmen der Diplomarbeit konnten diese Recherchen nicht mehr durchgeführt werden. Aus zeitlichen Gründen erfolgten nur Testrecherchen mit am Projekt direkt beteiligten Personen zur Überprüfung der Funktionsfähigkeit des Prototypen und zur Vorbereitung der sich unmittelbar an die Arbeit anschließenden Testrecherchen. Zu einzelnen der oben aufgeführten Punkte konnten bereits erste Tendenzen ermittelt werden:

Überprüfung der Suchattribute

Als kritisch er erweist sich das Kriterium Quelle. Während es im Zeitungsumfeld mit dem Namen der Zeitung (Spiegel, Süddeutsche Zeitung, etc.) identifiziert wird, versteht man in MUSAD, WOSAD und FESAD darunter die Art des Bild- und Tonträgers (Videoband, CD, Schallplatte, Tonband, etc.). Inwieweit durch Herausfiltern der für das jeweilige Archiv sinnlose Quellenangaben eine für den Anwender befriedigende Lösung gefunden werden kann, werden erst weiterführende Tests ergeben.

Praxistauglichkeit

In REGIS mußte eine Reihe von Sonderfällen eingebaut werden. So müssen aus allen Überschriften für die Anforderung eines Dokuments Hochkommas entfernt werden, Ziffern dagegen müssen in Hochkommas gesetzt werden. Die Länge von 42 Zeichen für die Eingabe der Suchbegriffe in REGIS erwies sich für eine eindeutige Identifikation als zu kurz. Zu lange Titel von Serien, deren Seriennummer erst nach dem 42. Zeichen aufgeführt wurde ließen sich nicht mehr recherchieren. In einigen Fällen wurden Überschriften in REGIS nicht oder nicht in der Form, in der sie in der Treffeliste dargestellt werden, erfaßt. Dies gilt insbesondere für Spiegel-Artikel und für Bildunterschriften[50]. Eine wirklich

[50] Spiegel-Artikel und Bildunterschriften wurden übrigens von REGIS als Rechercheergebnis geliefert, obwohl der Übersetzer als Zielkriterium SZ-Artikel angibt. Dieses im REGIS-Handbuch aufgeführte Zielkriterium wurde von REGIS ignoriert und automatisch in Artikel umgewandelt. Offensichtlich werden auch Bildunterschriften in REGIS als Artikel behandelt. Inwieweit dies ohne die dazugehörigen Bilder sinnvoll ist, sei dahingestellt.

zuverlässige Dokumentenanforderung durch den Übersetzer wird - mit vertretbarem Aufwand - erst durch eine Zugriffsmöglichkeit auf die interne Dokumentennummer in REGIS realisierbar.

MUSAD und AltaVista[51] funktionieren recht zuverlässig. Allerdings besteht in AltaVista die Möglichkeit einzelne „verwaiste Links“ als Rechercheergebnis zu erhalten. Dies ist ein bekanntes systembedingtes Problem des WWW.

Um ein längerfristig stabiles Funktionieren der Übersetzer von REGIS/MUSAD zu gewährleisten, hat es sich als vorteilhaft erwiesen, wenn sich die Übersetzer nach längerer Untätigkeit (z.B. 10 Minuten) wieder bei REGIS/MUSAD abmelden und die Verbindung beenden.

Performance

Die Performance-Angaben beziehen sich auf Test innerhalb des Hausnetzes des IRT beziehungsweise eine Anbindung des Arbeitsplatzrechners des Anwenders über ISDN an das IRT. REGIS ist über IBM-Connect mit 9600 Baud an das Netz des BR angeschlossen.

AltaVista erwies sich in den meisten Fällen als schnellste Suchmaschine, gefolgt von MUSAD und REGIS. Eine Recherche auf REGIS und MUSAD benötigte für ihre Bearbeitung im Durchschnitt ca. 30 Sekunden. Eine Dokumentanforderung je nach Länge des Dokuments ungefähr genauso lange. Sollte erst ein Verbindungsaufbau zu REGIS oder MUSAD durch den Übersetzer nötig sein, so verlängert sich die Anfragezeit um ca. eineinhalb Minuten. Ist ein Dokument bereits beim Broker lokal gespeichert, so verkürzt sich die Anforderungszeit in der Regel auf deutlich unter drei Sekunden.

[51] Einige Zeit lieferte die Recherche „Person=Kohl&Scharping&Schlagwort=Politik&Quelle=Süddeutsche Zeitung“ in AltaVista sogar mehr Artikel der SZ als eine entsprechende Volltextrecherche in REGIS. Die Ursachen hierfür konnten durch den Systemadministrator von REGIS nicht geklärt werden, ein Fehlverhalten der Volltextrecherche in REGIS wurde nach seiner Aussage jedoch schon beobachtet.

Kapitel 6

Bewertung und Ausblick

Das in dieser Arbeit entwickelte Konzept wurde bereits in mehreren Vorträgen einem breiteren Publikum vorgestellt. Die Resonanz war außerordentlich positiv. Besonders die Möglichkeit, über eine einfache und einheitliche Oberfläche von einem Web-Browser aus auf Datenbanken wie REGIS oder MUSAD zugreifen zu können schien ein wichtiges Kriterium zu sein. Eine Erweiterung des Prototypen um weitere hostbasierende Systeme ist fest geplant.

Ein Zugriff auf diese Systeme ist unter Umständen in Zukunft noch einfacher in Java zu realisieren. So hat IBM einen direkten SNA-Zugang mit Java angekündigt. Weiterhin wurden von IBM eine CICS- und eine CPIC-Bibliothek in Java angekündigt[52]. Für hostbasierende Systeme auf die auch in Zukunft nur eine Zugriff über eine 3270-Recherchemaske möglich ist, bleibt jedoch nur die in MUSAD und REGIS beschriebene Vorgehensweise.

Als weiteres wichtiges Kriterium hat sich die einfache Erweiterbarkeit des Systems erwiesen. Der Wunsch, Nachrichtensysteme an das Recherchesystem anzuschließen, die den Anwender automatisch mit den neuesten Nachrichten versorgen, wurde gegen Ende der Diplomarbeit häufiger geäußert. Eine Erweiterung des Recherchearbeitsplatzes um solche Nachrichtensysteme ist leicht möglich, jedoch nicht mehr Gegenstand dieser Arbeit.

[52] Erste Versionen standen gegen Ende der Diplomarbeit bereits im Internet zur Verfügung.

Abbildungsverzeichnis

Literaturverzeichnis

[Borghoff Schlichter 95] Borghoff, Uwe M., Schlichter, Johann H.:
Rechnergestützte Gruppenarbeit: eine Einführung in verteilte Anwendungen.
Springer Verlag (1995)

[Broy 92] Broy, Manfred:
Informatik, Eine grundlegende Einführung, Teil1
SpringerVerlag (1992)

[Flanagan 96] Flanagan, David:
Java in a Nutshell
O'Reilley & Associates (1996)

[Frakes, Baeza-Yates 92] Frakes, William B., Baeza-Yates, Ricardo:
Information Retrieval: Data Structures & Algorithms
Prentince Hall PTR(1992)

[Goslin, Joy, Steele 96] Goslin, James, Joy, Bill, Steele, Guy:
The Java Tutorial: Object-Oriented Programming for the Internet
Addison Wessley (1996)

[Huck 89] Huck, Frank R.:
Kooperationsprojekte deutsche Rundfunkanstalten beim Aufbau von Archivdatenbanken (Fernsehen, Wort und Musik)
in: INFO Jahrgang 4, Heft 2 1989
Nomos Verlagsgesellschaft Baden Baden

[Klute 96] Klute, Rainer:
Das World Wide Web: Web-Server und Clients, HTML 2.0/3.0, HTTP
Addison-Wesley (1996)

[Schmid 96] Schmid, Christopher:

Fopra: Realisierung einer Recherchekomponente für einen Archivarbeitsplatz
Aufgabensteller: Prof. Dr. Johann Schlichter
Technische Universität München, Institut für Informatik
Lehrstuhl Angewandte Informatik - Verteilte Systeme

[Tanenbaum 95] Tanenbaum, Andrew S.:
Moderne Betriebsysteme
2. Auflage
Hanser Verlag (1995)

Konzeption eines Recherchearbeitsplatzes für ein elektronisches Zeitungsarchiv

Marc Sieburg