Ergebnisaggregation auf der Basis von Web Mining

Diplomarbeit

Universität Rostock, Fachbereich Informatik, Lehrstuhl Datenbank- und Informationssysteme

vorgelegt von:
Dethloff, Christian
geboren am:
10.03.1976 in Waren/Müritz

Gutachter:
Prof. Dr. Andreas Heuer
Prof. Dr. Clemens H. Cap

Betreuer:
Dipl.-Ing. Astrid Lubinski
Dipl.-Inf. Ilvio Bruder

Abgabedatum:
28.02.2002

Zusammenfassung

Das World-Wide Web (WWW) stellt ein Informationsnetzwerk dar, in dem Autoren ihre verfaßten Arbeiten der Öffentlichkeit zugänglich machen können. Diese Informationen im Internet zu lokalisieren, ist trotz einer Vielzahl von Suchmaschinen schwierig, zumal sich die gewünschte Antwort aus Daten in unterschiedlichen Quellen zusammensetzen kann. Für einen eingegrenzten Anwendungsbereich lassen sich viele Verbesserungen finden und einführen. In dieser Arbeit wird eine Systemarchitektur für einen Suchdienst entwickelt und prototypisch implementiert. Als Anwendungsdomäne wird dabei das „Wetter in Deutschland“ betrachtet.

Abstract

The World Wide Web represents an information network, which enables authors to have their released papers made available to the public. Although the existence of search engines, localizing such information in the Internet may be difficult, complicated by the fact that the requested result may be composed of data which is to retrieve from different sources. Provided some restriction upon the range of data to search in there may be found and introduced plenty of improvements. This paper covers the development of a system architecture for a searching service as well as the implementation of a prototype. The search engine approach is based on an ontology and aggregates the hits produced due to a search request in a linked document. The domain of information retrieval used here comprises information concerning the „Weather in Germany“.

CR-Klassifikation:

H.3.3 Information Search and Retrieval
H.2.3 Languages
H.5.4 Hypertext/Hypermedia

Key-Words:

Web Mining, Metadaten, Ergebnisaggregation, Ontologien, WWW, Suchmaschinen

Inhaltsverzeichnis

1 Einleitung **1**

2 Suchmaschinen **3**
2.1 typische Klassifikation 4
2.1.1 Crawler-basierter Ansatz 4
2.1.2 Katalogdienste 12
2.1.3 Meta-Suchmaschinen 13
2.1.4 spezialisierte Suchdienste 15
2.2 Bewertung der Suchdienste 16
2.3 Lösungsansatz 18

3 Grundlagen **19**
3.1 Web Mining 19
3.1.1 Web Content Mining 22
3.1.2 Web Structure Mining 27
3.1.3 Web Usage Mining 27
3.2 Metadaten 28
3.3 Allgemeines über Ontologien 34

4 Realisierung eines domänspezifischen Suchdienstes **39**
4.1 inhaltliche Analyse der Wetterdomäne 41
4.2 persistentes Speichern der Wetter-Ontologie 50
4.3 Systemarchitektur 52
4.4 Initialisierung des Suchdienstes 55
4.4.1 globale Strukturanalyse der Domäne 55
4.4.2 lokale Strukturanalyse der Domäne 63
4.4.3 Web Content Mining-Technik zur Extraktion der Domäneninhalte 65
4.5 Anwenden des Suchdienstes 68
4.5.1 Anfragebearbeitung 69
4.5.2 Ergebnispräsentation 70

5 Implementierungsdetails & Ergebnisse **71**

6 Zusammenfassung & Ausblick **80**

Literaturverzeichnis **82**

1 Einleitung

Das World Wide Web ist ein riesiger Informationspool. Jeder kann informieren und informiert werden. Der Themenvielfalt ist in diesem Netzwerk keine Grenzen gesetzt. Der grandiose wirtschaftliche Aufstieg der New Economy im letzten Jahrzehnt des 20. Jahrhunderts verhalf dem Internet zu einem riesigen Imagegewinn. Vor ein paar Jahren noch größtenteils von Universitäten zu Forschungszwecken gebraucht, wird zusehends der kommmerzielle Charakter des Internets geformt. Viele Unternehmen verkaufen über Plattformen im Web ihre Produkte, liefern Supportlösungen und beraten ihre Kunden. Gleichzeitig wurde das Web auch für den privaten Anwender interessanter. Die Onlinekosten sanken, so daß viele dieses Medium nutzen, um sich zu präsentieren. Dementspechend stark stiegen auch die registrierten WWW-Seiten. Waren es 1998 ungefähr 350 Millionen, wird im Jahr 2000 die Anzahl auf ca. 1 Milliarde geschätzt. Aus diesen Mengen an Seiten für den Nutzer relevante Informationen zu extrahieren, ist die schwierige Aufgabe der Suchmaschinen. Folgende Probleme treten häufig bei herkömmlichen Suchmaschinen auf:

- Die Ergebnisse nicht das gesamte Spektrum an Wissen im Internet bezüglich des spezifizierten Themas des Nutzers ab.
- Die Ergebnismenge enthält nicht relevante Informationen.
- Dem Nutzer eine URL-Liste als Ergebnis präsentiert, die er auswerten muss. Nicht allzu selten erstreckt sie sich über mehrere Seiten, was ein nochmaliges Suchen nach den gewünschten Informationen nach sich zieht.

Ziel dieser Arbeit soll es sein, die eben genannten Probleme von herkömmlichen Suchmaschinen mit einer geeigneten Systemarchitektur zu vermeiden. Metadaten sowie Techniken aus den Bereichen des Web-Minings werden eingesetzt, um eine domänspezifische, hier "Wetter in Deutschland", Suchmaschine zu konzipieren und zu implementieren. Der Schwerpunkt liegt dabei auf Techniken aus dem Bereich des Web-Mining. Diese erlauben ein Aggregieren von relevanten Informationen aus verschiedenen Quellen, um so dem Nutzer die Daten in kompakter Form, ohne Verwenden von URL-Listen zu präsentieren.

Die Arbeit ist, wie folgt erläutert, aufgebaut. Nach der Einleitung beschreibt das zweite Kapitel, ausgehend von der aktuellen Marktsituation, die verschiedenen Typen der Suchmaschinen. Es werden zuerst die Eigenschaften genannt, um dann auf die Probleme einzugehen, die bezüglich der Arbeitsweise, Bedienung und der Architektur auftreten. Ausserdem werden Lösungen vorgeschlagen, wie man die Nachteile beseitigen und umgehen kann. Das dritte Kapitel definiert und beschreibt konkret die Techniken (Web Mining) und Hilfsmittel (Metadaten), die eingesetzt werden können, um die Lösungen

umzusetzen. Im vierten Kapitel wird die Vorgehensweise des Aufbaus einer domänspezifischen Suchmaschine erläutert. Die Systemarchitektur wird vorgestellt, konkrete Algorithmen, die in den verschiedenen Systemprozessen ablaufen, erklärt. Das fünfte Kapitel umfaßt Implementierungsdetails sowie eine Präsentation der Ergebnisse der Implementierung. Abschließend wird eine Zusammenfassung gegeben.

2 Suchmaschinen

Das WWW stellt ein Netzwerk dar, in dem riesige Informationsmengen auf einzelnen Rechnern für den weltweiten Zugriff publiziert werden. Das Wissen der Menschheit verdoppelt sich in einem immer kürzer werdenden Zeitraum, etwa alle 20 Jahre. Um in einem Informationsdienst dieser Größenordnung suchen zu können, bedarf es bestimmter Werkzeuge, der Suchmaschinen. Der Begriff „Suchmaschine“ kann in vier Typenklassen eingeteilt werden:

- **Crawler-basierter Ansatz:** Hierbei unterteilt sich die Architektur der Suchmaschine in verschiedene Komponenten: Der **Crawler** sammelt HTML-Dokumente und extrahiert erste Informationen, z.B. Update-Informationen, von den Seiten. Der **Indizierer** führt auf die gesammelten Dokumente weitere Extraktionsschritte durch, wobei ein Index über die Dokumente angelegt wird, um bei Bedarf schneller auf diese zugreifen zu können. Den Zugriff regelt das **Anfragetool**. Meistens bietet es clientseitig im Browser ein Suchinterface an. In diesem kann der Anwender seinen Suchterm spezifizieren. Anschließend wird dieser zum Server gesendet und ausgewertet. Die Ergebnisse, häufig Listen bestehend aus Links, die auf Dokumente zeigen, die bezüglich des Suchtermes relevant sind, werden dem Nutzer im Browser angezeigt.

- **Katalogdienste:** Redakteure pflegen die Inhalte manuell oder semiautomatisch, die den Nutzern als Kategorienhierarchie präsentiert wird.

- **Meta-Suchmaschinen:** Die Anfrage der Nutzer wird entgegengenommen und an verschiedene Crawler-basierte Suchmaschinen oder Katalogdienste weitergesendet. Die Ergebnisse werden aufbereitet, nach bestimmten Kriterien sortiert und dem Nutzer präsentiert.

- **spezialisierte Suchdienste:** Wie der Name schon sagt, wird sich auf etwas spezialisiert. Die Suche wird eingeschränkt. Dies kann z.B. ein Themengebiet sein oder es können nur Web-Server in die Recherche nach Informationen einbezogen werden, die sich in einer bestimmten geographischen Region befinden.

Speziellere Informationen zu den einzelnen Typen sind im nächsten Abschnitt 2.1 zu finden. Im Kapitel 2.2 werden einige Kriterien vorgestellt, mit denen man Suchdienste bewerten kann. Mit diesem Kriterienkatalog wird versucht die Suchmaschinenarten allgemein zu bewerten, um dann anschließend in Kapitel 2.3 herauszustellen, welche Techniken des Suchens nach relevanten Daten sich für das Anwendungsszenario in dieser Diplomarbeit eignen.

2.1 typische Klassifikation

In den nächsten Abschnitten werden die vier Typen vorgestellt. Die Einteilung basiert auf [Lam01]

2.1.1 Crawler-basierter Ansatz

In diesem Absatz wird der crawler-basierte Ansatz anhand der xFIND Suchmaschine[1] erläutert. Weitere Vertreter sind AltaVista[2], Google[3] oder InfoSeek[4]. Der xFIND Suchdienst wurde am IICM (**I**nstitut für **I**nformationsverarbeitung und **C**omputergestüutzte Neue **M**edien) an der Technischen Universität Graz in der Programmiersprache JAVA entwickelt und ist modular aufgebaut. Eine Beispielarchitektur des Systems wird in der unteren Abbildung gezeigt.

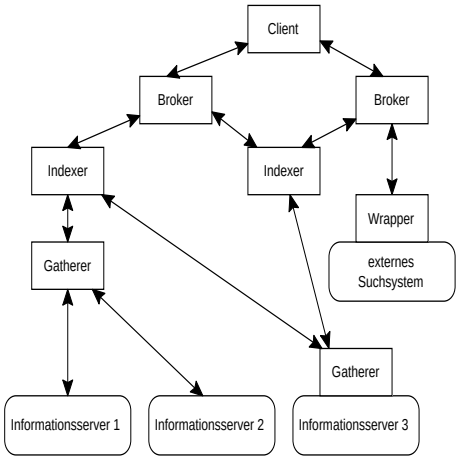


Abbildung 2.1: Beispielarchitektur des xFIND-Suchsystems

Bestandteile des Suchdienstes sind die drei Komponenten:

1. **Gatherer:** Die Crawlerkomponente dient dem Sammeln und Vorverarbeiten von Daten, die auf gewissen Informationsservern bereitgestellt werden. Der Gatherer kann lokal auf einem Informationsserver laufen, um dann das Filesystem zu analysieren und Beschreibungsobjekte des Inhalts zu erstellen.
   Es besteht ein Unterschied zu zentralen Suchdiensten, wie Google oder AltaVista. In diesen Systemen fordert die Crawlerkomponente in regelmäßigen Zeitabständen Web-Seiten an, um sie zentral zu analysieren. Die Folge sind eine unnötig hohe Server- und Netzbelastung, zumal die Anzahl der Dokumente und Suchdienste stark ansteigt und die

[1] http://xfind.iicm.edu/
[2] http://www.altavista.com/
[3] http://www.google.com/
[4] http://www.infoseek.com/

Systeme unabhängig voneinander dieselben Dokumente mehrfach anfordern [Lege99]. Abbildung 2.2 verdeutlicht den wachsenden Datenverkehr anhand einiger bekannter Suchmaschinen und ihrer zugehörigen Datensammler.

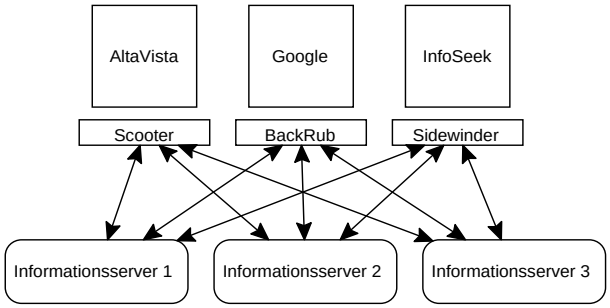


Abbildung 2.2: Gathering zentraler Suchdienste mit Hilfe von Robots und Spiders

Im Unterschied zum obigen Ansatz werden im xFIND-Suchsystem nicht ganze Dokumente, sondern die von den Gatherern erstellten Beschreibungsobjekte versandt. Außerdem entfällt bei lokal betriebenen Crawlern die Serverbelastung.
Ausführlichere Informationen zu der Gatherer-Komponente des xFIND-Systems folgen im Anschluß an der Vorstellung der Systemkomponenten.

2. **Indizierer:** Diese Komponente verarbeitet die Beschreibungsobjekte, die von einem oder mehreren Gatherern erstellt wurden und registriert sie in den von ihm verwalteten Indizes. Dabei kann ein Indizierer auf bestimmte Themenbereiche spezialisiert sein. Die Hauptaufgabe besteht in der Verwaltung von Datentabellen, dem lokalen Cachen der Informationen und in der Beantwortung von Suchanfragen.

3. **Broker:** Dieses Modul repräsentiert die Kontaktstelle für den Benutzer. Wie schon erwähnt, erhält der Indizierer die Daten von einem oder mehreren Gatherern. Dies bedeutet, daß der Indizierer einen mehr oder minder großen Teil des gesamten verfügbaren Wissens verwaltet. Jeder Broker kennt eine Reihe von Indizierern und weiß deshalb, welche Wissensbereiche jeder einzelne von ihnen abdeckt. Die Aufgabe des Brokers besteht darin, die an ihn gestellten Suchanfragen an den Indizierer zu verteilen, die Ergebnisse zu ranken und an den Anwender zurückzusenden. Neben den xFIND-Indizierern können auch externe Quellen, die über das xFIND-API oder einen zwischengeschalteten xFIND-Wrapper ansprechbar sind, abgefragt werden.

Im folgenden Abschnitt wird auf diese drei Komponenten näher eingegangen.

Gatherer (Crawler) Der Datensammler des xFIND-Systems, dort Gatherer genannt, fordert die Web-Seiten von den Informationsservern an. Da sich der Datenbestand aus sehr unterschiedlichen Objekten, z.B. Text- oder Bilddokumente in verschiedenen Datenformaten, zusammensetzen kann, erweist sich eine Vorverarbeitung der Daten für die nachfolgenden Bearbeitungsschritte der Indizierung und Suche als nutzbringend. Wie die Aufbereitung im xFIND-System konkret erfolgen soll, kann konfiguriert werden. Titel, Schlüsselwörter, Links, Metadaten oder beispielsweise eingebettete Multimediaobjekte können aus den gesammelten WWW-Seiten extrahiert und zu Beschreibungsobjekten zusammengefaßt werden. Darüber hinaus werden für jedes Dokument Updateinformationen gesammelt.
Dem Gatherer werden beim Start Initialisierungswerte aus Konfigurationsdateien übergeben, die der Anwender des xFIND-Systems angeben kann. Ein grober Überblick über die Einstellungsmöglichkeiten wird nun gegeben. Ausführliche Beschreibungen finden sich unter *http://xfind.iicm.edu*:

- **Start-URL's:** Der Gatherer startet von diesen URL's und verfolgt die Linkstruktur der besuchten Dokumente.
- **DepthMax:** Anzahl der Tiefe der Links, die verfolgt werden soll.
- **TTL:** Dieser Parameter gibt an, wie lange ein Objekt in der Datenbank nach einem erfolgreichen Update verwaltet werden soll.
- **URLMax:** Spezifiziert die maximale Anzahl von URL's, deren entsprechenden Dokumente eingesammelt werden sollen.
- **HostMax:** Dieser Wert gibt die maximale Anzahl an Hosts an, die besucht werden sollen.
- **AllowDeny:** Der Parameter spezifiziert einen Namen einer Sektion in einer Konfigurationsdatei, in der angegeben wird, welche Protokolle (HTTP, FTP), Dateien und Hosts erlaubt sind oder nicht. Die Angaben erfolgen mit Hilfe von regulären Ausdrücken oder einfachen Strings. Beispielsweise gibt der Eintrag in der Konfigurationsdatei *Deny ftp://.*:.** an, daß alle FTP-Dateien nicht eingesammelt werden sollen.

Mit der Startkonfiguration beginnt der Gatherer das Sammeln mittels eines Breitensuchealgorithmus und die Vorverarbeitung der HTML-Dokumente auf den besuchten Hosts. Das Ergebnis des Crawlvorgangs sind Beschreibungsobjekte je WWW-Seite. Unten werden einige Elemente mit den beispielhaften Werten aufgezählt, die im Beschreibungsobjekt enthalten sind.

1. *Last Modification Time:* 966783465
2. *Content-type:* text/html

3. *Keywords:* Aby, created, guifleisch, john, oliver, page, photopage, this, using, vink
4. *Type:* HTML
5. *Gatherer Name:* TheCorrs
6. *Object URI:* http://www.corrs.de/fan/oliver_oitg/outinthegreen2000.html
7. *MD5:* 73ee74aef2d7267244422df24cec7fe1
8. *Gather Time:* 1006853296
9. *Body:* OLIVER GUIFLEISCH .....

Der Crawler des xFIND-Systems unterstützt nicht das Auslesen des Meta Robots Tags. Dieser HTML-Konstrukt wird von Web Autoren benutzt, um zu kennzeichnen, daß sie ein Finden über öffentliche Suchdienste verhindern wollen:

```
<head>
<meta name="robots" content="noindex">
  <!-- ... andere Angaben im Dateikopf ... -->
</head>
```

Indizierer Die Aufgabe des Indizierers besteht darin, die Beschreibungsobjekte eines oder mehrerer Gatherer zu verarbeiten und in einem geeigneten Format, dem SOIF-Format [SOIF96] abzuspeichern.
Der Indizierer besteht aus vier verschiedenen Teilen:

- **Dokument-Speicher-Part:** Dieser Teil parst die Beschreibungsobjekte, die vom Gatherer erzeugt wurden. Es werden Dokumente registriert, die nicht schon indiziert wurden und in dem SOIF-Format [SOIF96] ins Datei-System abgespeichert. Das SOIF-Format ist leicht durchsuchbar. Der Aufbau von SOIF-Dateien wird weiter unten erläutert.
  Weiterhin werden die exakten Worthäufigkeiten in den Dokumenten festgestellt und in der Datenbank gespeichert. Als Organisationsform wird das invertierte File gewählt. Eine invertierte Datei ist die sortierte Liste aller Schlüsselwörter, wobei für jedes Wort die Objekte in einer sogenannten Posting-Datei vermerkt werden, in denen es auftritt. Vom Index werden Dokumente entfernt, deren TTL-Werte (gibt an, wie lange ein Dokument im Index verwaltet werden soll) nicht mehr gültig sind.
- **Metadata-Speicher-Part:** Das xFIND-Suchsystem bietet ein manuelles Registrieren von WWW-Quellen an. Diese Anmeldung wird akzeptiert, wenn die WWW-Seiten ausreichend vom Anwender beschrieben

werden. Im Rahmen des xFIND-Projektes wurde dazu ein Metadatenformat xQMS (Extensible Quality Metadata Scheme) entwickelt. Auf dieses Format soll hier nicht weiter eingegangen werden.
Diese Metadatenbeschreibung wird neben den entsprechenden beschriebenen Dokumenten ebenfalls abgespeichert. Folgende Abbildung 2.3 zeigt die Speicher-Komponenten

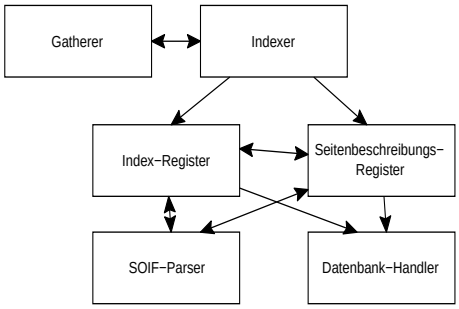


Abbildung 2.3: die Speicher-Komponenten des xFIND-Systems

- **Dokument-Retrieval-Part:** In dieser Komponente werden Abfragen vom Broker akzeptiert und ausgewertet.
- **Metadata-Retrieval-Part:** In dieser Komponente werden analog dem Dokument-Retrieval-Part Anfragen vom Broker akzeptiert und ausgewertet. Angefragte Dokumente, die durch entsprechende Attribute des xQMS-Formats, spezifiziert werden, können durch den Dokument-Retrieval-Part zurückgeliefert werden. Folgende Abbildung 2.4 zeigt die Retrieval-Komponenten

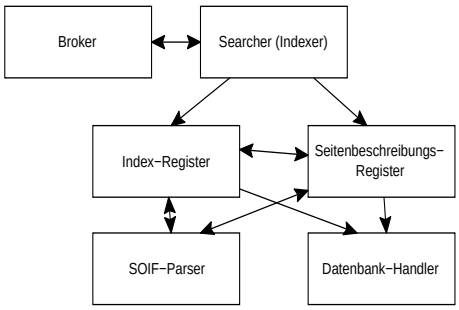


Abbildung 2.4: die Retrieval-Komponenten des xFIND-Systems

In den oberen Ausführungen wurde bereits das SOIF-Format [SOIF96] erwähnt. Dabei handelt es sich um ein Metadatenformat, welches aus gegebenen Objekten, wie z.B. HTML-Dateien, mit Hilfe eines *Summarizers* erzeugt wird. Ein Summarizer ist ein, auf ein bestimmtes Datenformat spezialisiertes, Softwaremodul. Denkbar sind dabei Konvertierer von HTML-,

SGML-, Postscript- und RTF-Dateien in entsprechende SOIF-Objekte. Das SOIF-Format beschreibt keine statische Umsetzung von Dokumenten, es stellt vielmehr einen Rahmen zur Abbildung von Dateien unterschiedlichster Formate in ein einheitliches Objekt dar, wobei die Zuordnung zu den Attributen frei konfigurierbar ist. SOIF-Objekte werden zum Beispiel in den Suchdiensten SWING[5], entwickelt an der Universität Rostock, erzeugt und eingesetzt. Abbildung 2.5 stellt die oberen beispielhaften Werte des Beschreibungsobjektes im SOIF-Datei-Format dar.

```
@FILE { http://www.corrs.de/fan/oliver\_oitg/outinthegreen2000.html
....
gatherer−name{8}: TheCorrs
...
md5{33}: 73ee74aef2d7267244422df24cec7fe1
...
last−modification−time{9}: 966783465
...
content−type{9}: text/html
....
keywords{74}: Aby, created, guifleisch, john, oliver, page, photopage,
this, using, vink
...
type{4}: HTML
...
gather−time{10}: 1006853296
...
body{100}: OLIVER GUIFLEISCH .....
...
}
```

Abbildung 2.5: beispielhafte Werte im SOIF-Format

Jedes SOIF-Objekt beinhaltet die URL des Dokuments und eine Liste von Attribut-Werte-Paaren. Es können nicht nur Teile des Originaldokumentes in den SOIF-Objekten enthalten sein, sondern auch Metainformationen und Linkreferenzen zu anderen HTML-Seiten, die den entsprechenden SOIF-Attributen zugeordnet werden. So ist ein Suchen nach Strings in bestimmten Attributen möglich.

Broker Das dritte Modul ist der Broker. Er empfängt eine Anfrage vom Client und sendet sie an den Indizierer weiter. Der Suchstring kann aus einem einzelnen Schlüsselwort oder auch aus Kombinationen von Schlüsselwörtern, die mit Hilfe der bool'eschen Operatoren AND, OR oder „AND NOT“ verknüpft werden, gebildet werden. Weiterhin besteht die Möglichkeit in den einzelnen SOIF-Attributen eines Objektes zu suchen. Der Broker bekommt die Ergebnisse vom Indizierer und rankt nach den relevantesten Informationen

[5] http://swing.informatik.uni-rostock.de

bezüglich der Suchanfrage. Das Ranking berechnet die Position eines Suchergebnisses innerhalb aller erzielten Ergebnisse. xFIND ist modular aufgebaut. Aus diesem Grund kann ein externes Ranking-System in die Architektur eingebunden werden.
Was bedeutet Ranking der Ergebnisse? Angesichts der Größe und Inhomogenität des WWW reicht für eine erfolgreiche Suche eine bloße Trennung zwischen relevanten und nicht-relevanten Seiten nicht aus. Entscheidend ist die Reihenfolge bei der Präsentation der Treffer, denn nur wenn das erste Dutzend der angezeigten Treffer schon hilfreiche Suchergebnisse enthält, hat die Suchmaschine einen echten praktischen Nutzen. Um dieser Problematik gerecht zu werden, arbeiten Suchmaschinen für das WWW seit jeher mit verschiedenen Heuristiken, um die Treffer einer Suchanfrage zu bewerten und in eine Reihenfolge bringen zu können. Folgende Heuristiken werden beispielsweise von einigen Suchmaschinen angewandt:

- Je mehr Begriffe aus der Suchanfrage im Titel einer Seite auftauchen, desto relevanter scheint die Seite für die jeweilige Anfrage zu sein. Analog kann man dies für gewisse Meta-Tags wie *description* und *keywords* annehmen.

- Je häufiger ein Suchbegriff innerhalb einer Seite vorkommt, desto relevanter scheint diese Seite für diese Anfrage zu sein. Dabei werden üblicherweise die verschiedenen Stellen des Auftretens, z. B. Titel, Meta-Tags, überschriften, Fließtext, unterschiedlich gewichtet.

- Je kürzer eine URL ist, desto bedeutsamer scheint die dazugehörige Webseite zu sein.

Gute, d.h. praxisgerechte Ranking-Algorithmen zeichnen sich durch folgende Eigenschaften aus [BrPa98]:

1. **Geschwindigkeit:** Die Antwortzeit einer Suchmaschine ist, zusammen mit einem guten Ranking, eines der wichtigsten Kriterien für die Nutzer-Akzeptanz. Daher müssen zeitaufwendige Berechnungen offline im Voraus vorgenommen werden. Alle Algorithmen, die online bei der Bearbeitung einer Suchanfrage laufen, müssen schnell ablaufen.

2. **Skalierbarkeit:** Das WWW übertrifft schon jetzt im Umfang alle praktisch relevanten Datenbanken. Die Anzahl der Dokumente im Web verdoppelt sich etwa alle 3 bis 6 Monate. ähnliches gilt für die Nutzerzahlen des WWW. Damit eine Suchmaschine in Zukunft noch nutzbar bleibt, müssen die verwendeten Algorithmen gut skalieren.

3. **Spamresistenz:** Bei Suchmaschinen gibt es zwei Gruppen von Interessenten: die Suchenden und die Gefundenen. Für viele Webseiten ist die

Anzahl der Besucher gleichbedeutend mit Umsatz und Geschäft. Daher setzen die Betreiber dieser Web-Seiten alles daran, die Ranking-Algorithmen der großen Suchmaschinen gut kennen zu lernen und ihre Seiten darauf zu optimieren. Im Extremfall führt das zum sogenannten Index-Spamming: Die „Treffer“ einer Suchmaschine werden unbrauchbar, weil nicht wirklich relevante Seiten als erstes aufgeführt werden, sondern solche, die den Index am erfolgreichsten manipuliert haben. Um das zu verhindern, sollte ein guter Ranking-Algorithmus schwer zu manipulieren, also spamresistent, sein.

4. **Plausibilität:** Das einzige, was für den Anwender letztlich zählt, ist die subjektive Zufriedenheit. Um das zu erreichen, müssen die Prinzipien, nach denen eine Suchmaschine das Ranking der Treffer durchführt, dem Anwender plausibel und sinnvoll erscheinen. Ein theoretisch perfekt durchdachtes Ranking nützt nichts, wenn der Anwender das Ergebnis nicht nachvollziehen kann.

In den vergangenen Jahren wurden zahlreiche Ähnlichkeitsmaße und Rankingstratiegien entwickelt [Note99]:

1. Beim *Coordinate Matching* ist die Häufigkeit des gesuchten Wortes im Dokument für die Berechnung des Rankingwertes ausschlaggebend. Je häufiger es vorkommt, umso relevanter ist es. Die Größe des Dokuments wird nicht berücksichtigt, was sich nachteilig beim Ranking auswirkt. Diesen Umstand beachtet die folgende Strategie.

2. Beim *Term Weighting* wird die Häufigkeit des Wortes auf alle im Dokument enthaltenen Wörter bezogen. Dabei ist die Gefahr des*Spammings* zu beachten. Dabei werden in einem Dokument einzelne Begriffe oft wiederholt, um eine hohe Relevanz vorzutäuschen.

3. Die ersten zwei Maße sind typische Verfahren bei klassischen Information Retrieval (IR) Systemen [BaRi99]. Bei der Bewertung von HTML-Dokumenten bezüglich eines Suchterms müssen zu den herkömmlichen Rankingstrategien der IR-Systeme noch weitere umgesetzt werden, beispielsweise die *Link Popularity*. Für die Berechnung des Relevanzwertes ist die Anzahl der Links von anderen Dokumenten auf das betreffende Dokument entscheidend. Eine Suchmaschine, bei der die Linkpopularität besonders stark in das Ranking der Treffer eingeht, ist Google. Der Algorithmus, der in Google die oben genannte Grundidee aufgreift und erweitert heißt *PageRank* [BrPa98].

Der Algorithmus *PageRank*, der im Suchdienst Google zum Einsatz kommt, wird nun erläutert. Vereinfacht läßt sich der Algorithmus wie folgt beschreiben:

1. Jede Seite wird mit einem Startwert initialisiert. Grundsätzlich können die Startwerte beliebig gewählt werden, da der Algorithmus in (praktisch) jedem Fall konvergiert. Allerdings hat die Wahl der Startwerte wesentlichen Einfluß darauf, wie schnell eine akzeptable Konvergenz erreicht wird. Da man den *PageRank* gerne auch als Wahrscheinlichkeitsmaß auffassen möchte, initialisiert man die Knoten z.B. mit $\frac{1}{Anzahl\ Knoten}$.

2. Aus den Gewichten der Knoten werden die Gewichte der ausgehenden Links der Seiten bestimmt als $\frac{Gewicht\ des\ Knotens}{Anzahl\ Links}$.

3. Aus den Gewichten der eingehenden Links (Backlinks) werden die Knotengewichte neu berechnet als Summe der Kantengewichte.

4. Dieses Verfahren wird ab dem 2. Schritt sooft wiederholt, bis die Knotengewichte konvergiert sind bzw. bis eine hinreichende Annäherung erreicht ist.

Eine stabile Zuordnung von Knoten- und Kantengewichten zeigt die Abbildung 2.6 in einem kleinen Beispielgraphen.

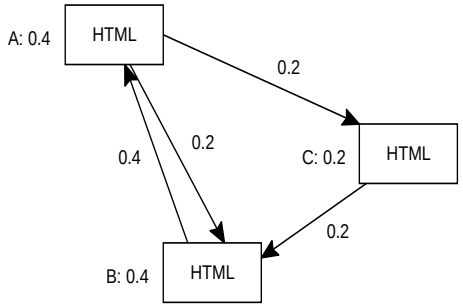


Abbildung 2.6: Ein Linkgraph mit Kanten- und Knotengewichten im Gleichgewicht

Auf eine mathematische Notation des Algorithmus wird verzichtet. Entsprechende Informationen können [BrPa98] entnommen werden.

2.1.2 Katalogdienste

Kataloge sind nach Kategorien hierarchisch gegliederte Linksammlungen. Die Kategorien werden redaktionell erstellt. Die Navigation durch Kataloge kann durch Anklicken der Hauptkategorien und danach der Unterkategorien erfolgen. Oft wird zusätzlich noch eine Volltextsuche angeboten, über die der Datenbestand des Kataloges durchsucht werden kann. Wird kein Treffer im Katalogbestand gefunden, erfolgt bei einigen Katalogen die Weitergabe des Suchwortes an einen Dienst, der auf dem crawler-basierten Ansatz beruht, welcher dann Treffer aus dem eigenen Index anzeigt. Die Aufnahme der eigenen

URL in einem Katalog erfolgt im allgemeinen durch eine manuelle Anmeldung über ein Formular. Die Zuordnung zu einer Kategorie, in der die URL eingeordnet werden soll, erfolgt entweder durch eigene oder durch Zuordnung der Katalogadministration anhand von Stichworten oder einer Kurzbeschreibung. Einige Kataloge setzen ergänzend Roboter zum Auffinden neuer Seiten oder zum Aktualisieren des vorhandenen Datenbestandes ein. Abbildung 2.7 zeigt den Aufbau eines typischen Katalogdienstes, wie z.B. Yahoo[6]

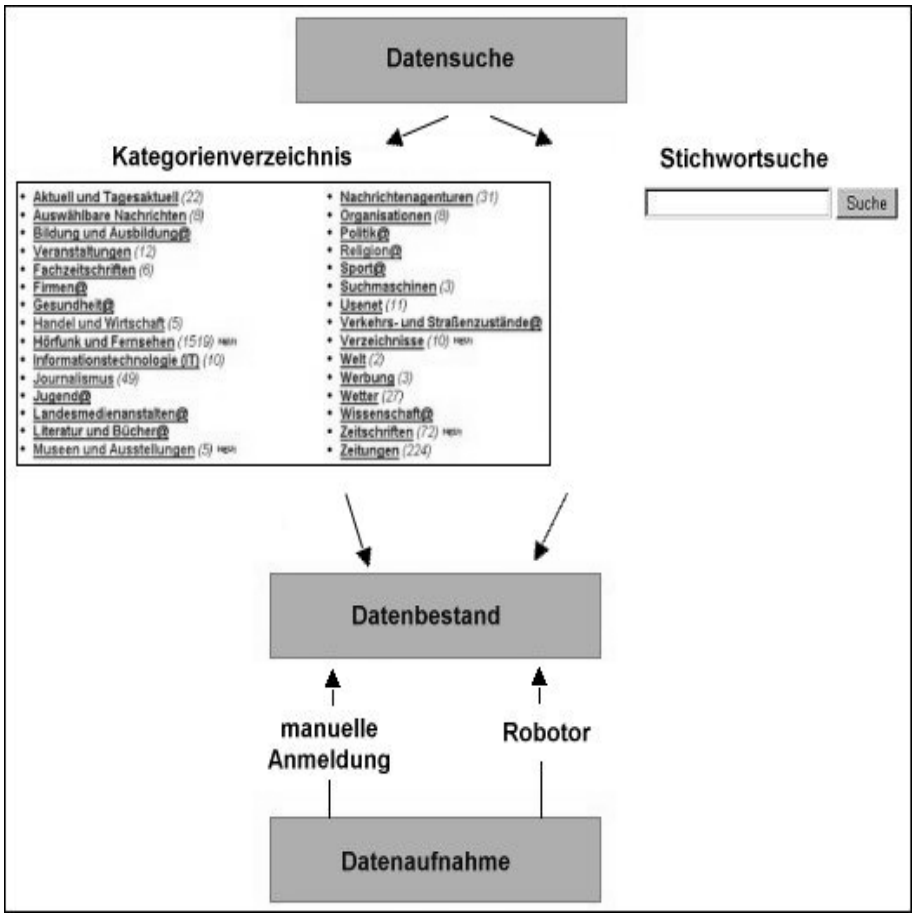


Abbildung 2.7: allgemeine Architektur der Katalogdienste

2.1.3 Meta-Suchmaschinen

Suchdienste, die auf dem crawler-basierten Ansatz beruhen, decken nur einen Teil des Internets ab. Das Web wächst unaufhörlich, die Kluft zwischen erfaßten Webseiten und Suchmaschinenindizies wird immer größer. Die Indizies von crawler-basierten Suchmaschinen sind nicht deckungsgleich. Wie aus nachstehender Abbildung zu ersehen ist, werden nur Teile des Webs gleichzeitig von mehreren Systemen erfaßt. Genau dort setzt die Konzeption der

[6] http://www.yahoo.com

Meta-Suchdienste an. Eine Metasuchmaschine ist ein System, welches den Suchterm des Anwenders entgegennimmt und an verschiedene crawler-basierte Suchdienste weiterleitet. Ein klassischer Vertreter ist MetaGer[7]. Es bietet sich weiterhin an, weitere Quellen wie Enzyklopädien, Wörterbücher, Newsgroups in die Suche einzubeziehen, wie dies z.B. die Meta-Suchmaschine *metor*[8] umsetzt.
Weitere Merkmale sind:

1. **Anfrage übersetzen:** Suchdienste starten mit unterschiedlicher Syntax, Abfragen werden so übersetzt, daß sie von allen beteiligten Suchdiensten gleichermaßen interpretiert werden.

2. **Ranking der Ergebnisse:** In der generierten Liste muß die Platzierung neu bestimmt werden. Das sollte in Abhängigkeit von der Häufigkeit des Vorkommens in anderen Suchdiensten und den dortigen Platzierungen erfolgen.

3. **Aussortieren doppelter Ergebnisse:** Bei der Abfrage mehrerer Suchdienste können natürlich Ergebnisse mehrfach vorkommen. Diese müssen zusammengefaßt und in ihrer Gesamtheit bewertet werden. Ist beispielsweise eine Seite in drei Suchdiensten auf Platz eins, dann soll sie trotzdem nur einmal angezeigt werden.

Untenstehende Abbildung zeigt den typischen Aufbau einer Meta-Suchmaschine

[7] http://www.metager.de
[8] http://www.metor.com

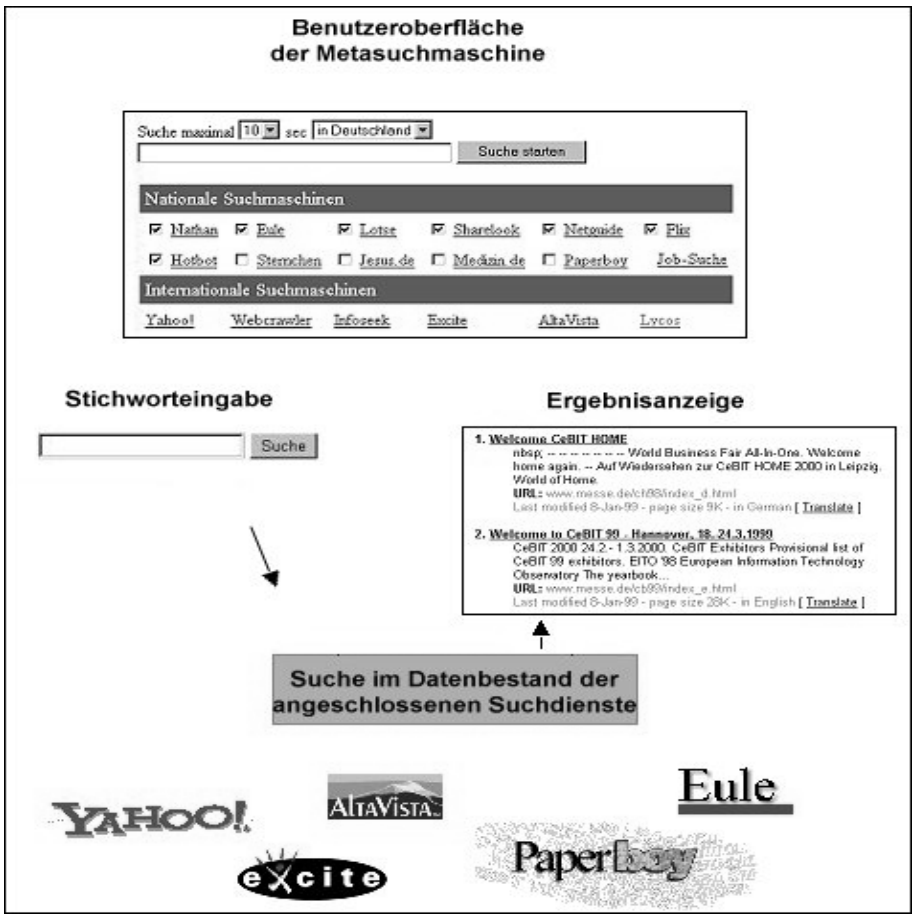


Abbildung 2.8: allgemeine Architektur einer Meta-Suchmaschine

2.1.4 spezialisierte Suchdienste

Zur Zeit existieren keine fundierten Untersuchungen, wie groß das Internet tatsächlich ist. Man kann aber davon ausgehen, daß die Suchdienste immer nur einen kleinen Teil abdecken. Der Suchdienst Google besitzt den wohl zur Zeit größten Index (ca. 2 Milliarden Seiten). Dieser Diskussion entgehen die spezialisierten Suchmaschinen, die ihre durchsuchten Quellen nach bestimmten Kriterien einschränken:

- **themenbezogene Einschränkungen:** Beispielsweise kann man nur Web-Dokumente indizieren, die sich mit Sport oder mit Wetter befassen. *Findolin*[9] hat sich z.B. auf die Suche nach Servern für Newsgroups spezialisiert. Den Nutzern ist es leicht möglich, kostenlose News-Server zu finden, bei denen die gewünschte Newsgroup abonniert wird.

[9] http://www.findolin.de

- **serverbezogene Einschränkungen:** Ist ein Anwender beispielsweise an Informationen interessiert, die sich auf eine geographische Region beschränken, besagt eine Heuristik, daß sich diese Daten auf Web-Servern befinden, die ebenfalls in dieser Region positioniert sind. Als Beispiel sei hier die Suche nach Gaststätten und Hotels im Raum Rostock genannt.

2.2 Bewertung der Suchdienste

In diesem Kapitel werden Kriterien genannt, mit denen man Suchmaschinen bewerten kann. Die Definitionen dieser Kriterien werden im folgenden genauer erklärt.

- **Precision (Genauigkeitskriterium):** Anteil (hoch-)relevanter Suchergebnisse (unter den Top 10). Die Präzision macht also eine Aussage darüber, wieviele irrelevante Dokumente die Suchmaschine liefert.
- **Recall (Vollständigkeitskriterium):** Anteil der gefundenen relevanten Dokumente an der Gesamtheit aller relevanten Dokumente im Internet.
- **Ergebnispräsentation:** Wie werden die Ergebnisse dem Nutzer eines Suchdienstes präsentiert?

Das **Vollständigkeitskriterium** kann von keinem Suchdienst erfüllt werden, weil das Internet zu rasant wächst. Schätzungen[10] gehen von 550 Milliarden Seiten, einschließlich dem sogenannten „hidden web“ aus. Das „hidden web“ bezeichnet die Informationen, die dynamisch dem Anwender präsentiert werden, wenn z.B. per CGI-Skript Anfragen an eine Datenbank gestellt werden, um die aktuelle Temperatur in Rostock oder Schwerin zu bekommen.
Die Erfüllung des **Genauigkeitskriteriums** hängt vom umgesetzten Rankingalgorithmus in den Suchdiensten ab. Da der Markt der Suchmaschinen in den letzten Jahren kommerziellen Charakter angenommen hat und marktwirtschaftlichen Gesetzten unterliegt, ist der eingesetzte Algorithmus ein gut behütetes Geheimnis der Suchdienstbetreiber. Das Google-System mit dem implementierten *PageRank*-Algorithmus erzielt eine „annehmbare“ Genauigkeit. Der *PageRank*-Algorithmus basiert auf der Idee, die Linkstruktur der WWW-Seiten und den damit verbundenen semantischen Informationen zu benutzen. Wenn man z.B. den Namen eines Wissenschaftlers eingibt, so erhält man die URL's, die auf Seiten zeigen, die die Publikationen oder die Homepage dieser Person beinhalten. Wie das letzte Beispiel aber zeigt, hängt die erzielte Genauigkeit der Suchdienste von der Art der Anfrage ab. Möchte man einen Überblick von einem bestimmten Aspekt haben oder in allgemeineren Themen systematisch suchen, kann man auf die Katalogdienste zurückgreifen. Da

[10] http://www.searchenginewatch.com

aber die Inhalte von Katalogdiensten von Redakteuren manuell erstellt werden, wie z.B. Yahoo!, kann es passieren, daß für speziellere Suchanfragen keine Einträge vorhanden sind. Recherchen in spezielleren Gebieten unterstützen crawler-basierte Suchdienste sehr gut, da sie den Text auf den WWW-Seiten indizieren und im Falle von Google zusätzlich die Linkstruktur der WWW-Seiten untereinander betrachten. Deshalb gehen immer mehr Dienste dazu über, beide Suchmaschinentypen zu verbinden. Werden für eine Suchanfrage im Katalog keine Einträge gefunden, so erfolgt eine Weiterleitung zu einem crawler-basierten Suchdienst. Unternehmen, die auf diese Art zusammenarbeiten, sind z.B: **Yahoo & Google, Web.de & Fast Search, Allesklar & Fireball, Dino-Online & Inktomi**.
Meta-Suchmaschinen decken die Indizes von mehreren crawler-basierten Suchdiensten ab, und so ist anzunehmen, daß sie mehr Informationen liefern können als die jeweiligen crawler-basierten Suchdiensten. Wie genau aber eine Anfrage von einer Meta-Suchmaschine beantwortet werden kann, hängt wiederum vom verwendeten Ranking-Algorithmus ab. Da hier keine fundierten wissenschaftlichen Informationen vorliegen, kann man keine Aussage über die Genauigkeit dieser Dienste geben.
Bei den spezialisierten Suchdiensten dagegen kann man annehmen, daß sie bezüglich der Einschränkung (Thema oder Server, siehe 2.1.4) die genauesten Ergebnisse von den vier vorgestellten Suchmaschinenarten zurückliefern.
Die **Ergebnispräsentation** wird oft als Liste von URL's gestaltet. Es werden noch zusätzliche Informationen mit angegeben, wie die untere Abbildung eines Ergebnisses des Suchdienstes Google zeigt.


Wetter nach dem Hundertjährigen **Kalender**
... für das Jahr **2002** In den alten ... dem Titel "**100-jähriger Kalender**". Bis zum ... gem nach dem **100**-jährigen **Kalender** gegriffen, denn ... **Februar**. 1. - 6 ...
www.altmuehltal.de/hundert.htm - 15k - Im Archiv - Ähnliche Seiten

Morgenweb: wetter
... Sonnt sich die Katz im **Februar**, liegt sie im März hinterm Ofen ... **100-jähriger Kalender**.
Trübseliges Winterwetter. Vereinzelt kommen ein paar ...
www.morgenweb.de/aktuell/wetter/ - 30k - Im Archiv - Ähnliche Seiten

100 jähriger Kalender
Das Wetter für **2002** nach dem Hundertjährigen **Kalender** ... **Februar**: 1. - 6. trüb, Regen, Nebel, Wind. 7. hell und ziemlich kalt. 8. - 11. trüb mit Regen ...
private.freepage.de/cgi-bin/feets/freepage_ext/41030x030A/ rewrite/probol/hobby/kalender2002.htm - 6k - Im Archiv - Ähnliche Seiten

Abbildung 2.9: Ausschnitt aus einer Ergebnispräsentation des Suchdienstes Google

Textausschnitte aus den referenzierten Dokumenten, in denen die Suchterme vorkommen, werden abgebildet. über den Typ des Dokumentes (PDF, PS, HTML...) wird informiert und außerdem werden Seiten, die vom gleichen Host kommen, optisch gruppiert. Gerade bei allgemeineren Suchanfragen ist die URL-Liste so lang, daß eine nachträgliche Suche in den Ergebnissen nach den Dokumenten, die die relevanten Informationen enthalten, durchgeführt werden muß.

2.3 Lösungsansatz

Im Rahmen der Diplomarbeit soll ein Suchdienst aufgebaut werden, der für einen eingegrenzten Anwendungsbereich genaue Ergebnisse liefert. Der Suchdienst soll sehr spezielle Anfragen an die Domäne zulassen und mit einer sehr hohen Genauigkeit antworten. Das sind Eigenschaften der spezialisierten Suchdienste.
Es kann der Fall auftreten, daß sich die relevanten Informationen bezüglich eines Suchterms in verschiedenen HTML-Dokumenten auf unterschiedlichen Hosts befinden. Diese Informationen müssen aggregiert und in ein verlinktes Dokument integriert werden. Besondere Aufmerksamkeit wird den Ranking-Algorithmen, besonders unter Berücksichtigung der *Link Popularity*, gegeben, die die relevantesten Ergebnisse am höchsten einstufen. Der Fakt, daß der *PageRank*-Algorithmus in Google umgesetzt wurde, macht den crawler-basierten Ansatz interessant. Der Suchdienst, der in dieser Arbeit umgesetzt wird, ist auf ein Thema spezialisiert und beruht auf dem crawler-basierten Ansatz.

3 Grundlagen

Im letzten Kapitel wurden Suchmaschinen untersucht. Dabei wurde eine typische Klassifikation von Suchdiensten vorgestellt, bewertet und herausgestellt, welche Arten von Suchmaschinen (crawler-basierter Ansatz und die spezialisierte Suchmaschine) die Basis für einen neuen Suchdienst darstellen, der im Rahmen dieser Arbeit entwickelt wird. In den Komponenten des Systems der „neuen“ Suchmaschine kommen Algorithmen aus dem Bereich des Web Mining zum Einsatz. Außerdem werden Metadaten, verwendet, um die Ergebnisse einer Suchanfrage qualitativ zu erhöhen. Aus diesem Grund werden in diesem Kapitel näher auf die Themengebiete Web Mining und Metadaten eingegangen.

3.1 Web Mining

Das Medium Internet ist für zwei unterschiedliche Gruppen von Benutzern interessant. Die erste umfaßt die Gruppe von Nutzern, die sich durch das Surfen und dem Suchen über spezielle Themen im Internet informieren wollen. Sie erwarten folgendes vom World-Wide Web:

- **relevante Informationen finden:**
  Wie oben angesprochen, benutzt diese Gruppe von Nutzern Suchmaschinen oder versucht durch eigenständiges Verfolgen von Links, auch Browsing genannt, „gute Daten“ im Internet zu lokalisieren.

- **neues Wissen aus den Informationen generieren:**
  Dieser Punkt kann als Sub-Problem des oberen betrachtet werden. Bezeichnet oberer Punkt einen query-triggered Prozeß, bei dem Dokumente bezüglich einer Suchanfrage oder durch das Surfverhalten des Nutzers ausgewählt werden, umfaßt diese Kategorie einen data-triggered Pozeß. Man versucht aus einer Menge von schon erhaltenen Web-Dokumenten neues Wissen zu generieren. Data Mining-Techniken können demzufolge verwendet, wie auch in [CDF+98],[Cohe99],[MBNL99] motiviert und umgesetzt wurde.

- **Personalisierung der Informationen:**
  Dieses Kriterium differiert von Nutzer zu Nutzer, und ist sehr subjektiv zu betrachten. Jeder möchte seine bestimmten Vorlieben und seinen Geschmack z.B. bezüglich der Präsentation der Ergebnisse einer Suchmaschine umgesetzt wissen. Natürlich gibt es auch unterschiedliche Auffassungen über den Inhalt der spezifizierten Web-Seiten. Zwei Nutzer, die z.B. die gleichen Schlüsselwörter einem Suchdienst übergeben, erwarten nicht die gleiche Antwort oder sind dementsprechend mit unterschiedlichen Ergebnissen zufrieden.

Die zweite Gruppe von Nutzern sind die Service-Provider, die Dienste und Informationen der Öffentlichkeit zur Verfügung stellen. Sie sind daran interessiert, zu erfahren, wie ihr Angebot bei dem „Rest“ der Web-Community ankommt.
Techniken aus dem Bereich des Web Mining versuchen gerade diese oben aufgezeigten Bedürfnisse mit geeigneten Verfahren zu befriedigen. Die Eigenschaften der oberen Kriterien begünstigen den Einsatz von Techniken anderer Forschungsgebiete, wie Information Retrieval (IR), Information Extraktion (IE), Machine Learning (ML), Data Mining und Datenbanktechniken. Web Mining wird in Anlehnung an Data Mining wie folgt definiert [Etzi96]:

> „Web Mining bezeichnet das Benutzen von Data Mining-Techniken, um automatisch Informationen von Web-Dokumenten zu entdecken und zu extrahieren.“

Das Themengebiet des Web Mining ist mittlerweile groß und unterliegt den Interessen unterschiedlicher Forschungsrichtungen, die eigene Projekte umgesetzt oder geplant haben. Dieser Umstand macht es nicht einfach, korrekt festzulegen, welche anderen Forschungszweige das Gebiet des Web Mining's beeinflussen.
In [KoBl00] wird ähnlich Etzioni [Etzi96] ein Zerlegen der Aufgabe des Web Mining in Teilaufgaben vorgeschlagen:

1. **Finden der Quellen:**
   Die Aufgabe des Erhaltens der relevanten Web-Dokumente

2. **Selektion von Informationen und Vorverarbeitung:**
   Automatische Selektierung und Pre-Processing von spezifischen Informationen der erhaltenen Web-Ressourcen

3. **Generalisierung:**
   Automatisches Entdecken allgemeiner Pattern auf individuellen Web-Seiten sowie auf mehreren Seiten

4. **Analyse:**
   Validierung und/oder Interpretation der extrahierten Pattern

Die erste Teilaufgabe umfaßt das Suchen von und Zugreifen auf Online- oder Offline-Dokumenten, wie zum Beispiel elektronische Newsletter, elektronische Journale, Einträge aus Newsgroups, sowie der Inhalt von HTML-Dokumenten. Auch sind damit Dokumente gemeint, die beispielsweise durch CGI-Anfragen erhalten werden.
Schritt zwei läßt sich mit IR-Techniken durchführen. Der Transformationsschritt enthält beispielsweise das Entfernen von Stop-Wörtern, der Technik des Stemmings oder es wird nach Phrasen gesucht, die das Gewünschte beschreiben oder enthalten.

Bei der Generalisierung kommen Techniken des Data Mining oder des maschinellen Lernens zum Einsatz. Das Web ist ein interaktives Medium. Daher spielt bei der Validierung und bei der Interpretation der Pattern im Schritt vier der Mensch eine bedeutende Rolle.

Die eben genannte Definition und die genannten Aufgaben des Web Mining lassen einen Vergleich mit dem Prozeß des KDD [FaPS96] (Knowledge Discovery in Databases) zu. Man könnte Web Mining als Erweiterung des KDD-Prozesses betrachten.
Außerdem wird Web Mining oft mit IR, IE und mit maschinellem Lernen, welches sich auf das Web bezieht, gleichgesetzt. Diese Sichtweise ist aber falsch. Die nächsten drei Gegenüberstellungen verdeutlichen dies:

1. **Web Mining vs. IR:**
   Information Retrieval hat das primäre Ziel, Text zu indizieren, die Suche nach nützlichen Dokumenten zu unterstützen, Dokumentenklassifikation, Dokumentenkategorisierung, Datenvisualisierung, Filtertechniken anzubieten und darüber hinaus User-Interfaces bereitzustellen, die eine intuitive Suche ermöglichen. Eine detailliertere Übersicht ist in [BaRi99], [Rijs79] zu finden. Die Kategorisierung und Klassifikation von Web-Dokumenten kann als eine Instanz des Web Mining betrachtet werden, welche zur Indizierung genutzt werden kann. Bezüglich dieser Sichtweise ist Web Mining ein Teil des (Web)-IR-Prozeß.

2. **Web Mining vs. IE:**
   Das Ziel von Information Extraction (IE) ist das Transformieren einer Dokumentensammlung, gewöhnlich mit der Hilfe eines IR-Systems, in Informationen, die sich leichter analysieren lassen [CoLe96]. Es werden relevante Informationen extrahiert. Es gibt verschiedene Ansichten darüber, wie Web Mining und IE zusammenhängen und wie man beide Techniken in einem Gesamtkontext darstellen kann. Eine Meinung, die hier vertreten wird, ist der Einsatz von Web Mining, um den Web IE-Prozeß zu verbessern. Es gibt zwei Typen von IE: die Extraktion von unstrukturierten Texten und von semi-strukturierten Daten [Musl99]. Die IE-Aufgaben, die bei Vorliegen von unstrukturierten Dokumenten erledigt werden müssen, unterscheiden sich erheblich von den Aufgaben, die beim Bearbeiten von semi-strukturierten oder strukturierten Texten auftreten. Beim ersten Fall kommen Verfahren aus der Computer-Linguistik zum Einsatz, bevor Verfahren aus dem Bereich des Data Mining eingesetzt werden können [CoLe96], [Wilk97]. Mit der steigenden Popularität des Web's stiegen die Ansprüche an IE-Systeme, die Informationen aus semi-strukturierten Texten, den HTML-Dokumenten, extrahieren können, z.B. Meta-Informationen, wie die HTML-Tags [Soder96]. Bezüglich des Mediums „Internet“ , welches sehr dynamisch ist und

Inhalte auf sehr unterschiedliche Art und Weise bereitstellt, ist es sehr umständlich, IE-Systeme manuell aufzubauen [MuMK98]. Es gibt aber durchaus auch Systeme, die im Rahmen solcher Projekte entstanden sind [CGH+94],[AtMe97],[HaGa97]. Andere wiederum benutzen Techniken aus den Bereichen maschinelles Lernen und Data Mining, um Extraktionspattern oder Extraktionsregeln automatisch oder semiautomatisch für Web-Dokumente zu lernen [Kusk99]. Einige Beispiele sind in [KuWD97],[Frei98],[HsDu98],[MuMK98],[GrMe99],[Soder96] zu finden.

3. **Web Mining vs. webbezogenes maschinelles Lernen:**
   Es gibt Applikationen, die mit Hilfe von Techniken aus dem Bereich des maschinellen Lernens implementiert wurden und keinen Bezug zum Web Mining besitzen. Ein Beispiel ist eine Technik, die benutzt wird, um einen Crawler aufzubauen, der die Web-Daten, die einer speziellen Domäne angehören, effizient sammelt [ReMc99],[MNRS99]. Verfahren aus dem Bereich des maschinellen Lernens können in den Prozeß des Web Minings integriert werden. [Mlad99] zeigt die Verbesserung der Textklassifikation mit Hilfe einer Technik des maschinellen Lernens gegenüber traditioneller IR-Techniken. Man kann aus diesem Grund keine klare Trennlinie zwischen diesen beiden Forschungsrichtungen ziehen.

Nachdem im oberen Abschnitt Web Mining definiert wurde und der Zusammenhang mit anderen Forschungsrichtungen (IR, IE, maschinelles Lernen) diskutiert wurde, sieht dieser Abschnitt eine Aufteilung des Web Mining in Kategorien vor. Je nachdem, welches Ziel des Minings verfolgt wird, kann man folgende Aufteilung, ähnlich [MBNL99],[BoLe99], vornehmen: *Web Content Mining*, *Web Structure Mining*, und *Web Usage Mining*, um Inhalt, Struktur und Benutzerverhalten zu „minen“ .

3.1.1 Web Content Mining

Web Content Mining beschreibt das Extrahieren von nützlichen Informationen aus dem Web. Das können Web-Daten, Web-Dokumente und Web-Inhalte sein. Das WWW schließt eine Menge von Diensten und Quellen, wie Gopher, FTP, Usenet, ein, die über das Internet nutzbar sind. Außerdem existieren digitale Bibliotheken, die eine Menge an Dokumenten, z.B. Postscript-Dateien, im Web bereitstellen. Auch von Unternehmen und staatlichen Institutionen kann man über dieses Medium eine Menge erfahren. Sie stellen sich auf eigenen Web-Seiten vor und ihre Datenbanken Öffentlichkeit zugänglich gemacht. Natürlich sind einige Daten nicht indizierbar. Diese werden entweder automatisch durch Anfragen generiert oder sind privat. Die nützlichen Informationen können auf unterschiedliche Art und Weise in verschiedenen Typen, textueller, multimedialer Natur, präsentiert werden. Außerdem können dies

Meta-Informationen oder Hyperlinks sein. Ein spezieller Forschungszweig des Data Mining wird Multimedia Data Mining [ZHL+00] genannt und kann als Teilgebiet des Web Content Mining betrachtet werden. Dieser Bereich erfährt aber weniger Beachtung als der Bereich, der sich mit den Hypertextinhalten auseinandersetzt [ZHL+00],[Mitc99]. Der Web-Inhalt besteht aus unstrukturierten Bereichen (freie Texte), semi-strukturierte Daten (HTML-Dokumente) und strukturierteren Inhalten (Daten in Tabellen oder generierten HTML-Seiten). Den Forschungsbereich, der sich mit den unstrukturierten Inhalten befaßt, nennt man KDT (Knowledge Discovery in Text) [FeDa95], Text Data Mining [Hear99] oder Text Mining [Tan99]. Folglich kann man Text Mining als eine Instanz des Web Content Mining's betrachten. Text Mining wird später in einem separaten Abschnitt behandelt.
Man kann den Forschungszweig Web Content Mining aus zwei Blickwinkeln betrachten: Aus der Sicht des Information Retrieval und der Datenbanktechniken. Die nächsten Abschnitte stellen die unterschiedlichen Sichtweisen dar.

Web Content Mining aus der IR-Sicht Im letzten Abschnitt wurden schon die unterschiedlichen Arten des Web-Inhaltes erwähnt, hauptsächlich die unstrukturierten und die semi-strukturierten Dokumente. Die nächsten beiden Abschnitte diskutieren die Behandlung dieser Daten mit Techniken aus dem IR-Bereich.

- **unstrukturierte Dokumente:**
  Es gibt eine Reihe von Applikationen, mit denen unstrukturierte Dokumente mit IR-Techniken bearbeitet werden können, z.B. Klassifikation von Texten, Kategorisierung von Texten, Finden von Schlüsselwörtern und Phrasen usw. Daniel Billsus und Michael J.Pazzani entwickelten 1999 ein Tool, welches Nachrichten klassifiziert [BoLe99]. Sie mußten vor der Klassifikation die unstrukturierten Dokumenten derart durch eine Methode konvertieren, daß sie untereinander vergleichbar sind. Die Texte werden dabei in TF-IDF Vektoren (term-frequency / inverse-document-frequency) umgewandelt und dann das Kosinusmaß zwischen ihnen und den Vektoren von Trainingsdokumenten verwendet, um eine Gleichheit messen und dementsprechend klassifizieren zu können. Nähere Informationen zu der TF-IDF-Methode sind in (Salton, G. (1989). Automatic Text Processing. Addison-Wesley.) zu finden.
  Die Vektoren bestehen aus einem Merkmal je Zeile. Merkmale werden von den Trainingsdokumenten festgelegt. Ein Merkmal kann z.B. ein Wort sein. In den zu untersuchenden Dokumenten wird das Merkmal repräsentiert, z.B. in [BoLe99] durch die TF-IDF-Darstellung. Es wird die Häufigkeit des Wortes in den Dokumenten bestimmt. Eine andere Möglichkeit ist die Boolean-Darstellung, bei der festgehalten wird, ob das Wort (Merkmal) vorhanden ist oder nicht.
  Außer den genannten sind noch weitere Ansätze möglich, Merkmale

zu definieren. Beispielsweise kann gleichzeitig noch die Wortposition bestimmt werden [Cohe95],[AHKV98],[FPW+99], oder Wortsequenzen bis zu einer festgelegten Länge n, die n-gram-Repräsentation genannt [HKLK97],[KaHS97], z.B. „Weihnachten schneit es “ ist ein Tri-Gram. [KoBl00] zeigt in einer Übersicht weitere Möglichkeiten auf.

- **semi-strukturierte Dokumente:**
  Semi-strukturierte Dokumente bestehen aus strukturierten und unstrukturierten Teilen. Applikationen, die auf semi-strukturierten Dokumenten aufsetzen, um relevante Informationen zu extrahieren, haben zusätzlich zu den Verfahren für die Bearbeitung von unstrukturierten Dokumenten die Möglichkeit eben die strukturierten Elemente zu erfassen und auszuwerten. Bei HTML-Dokumenten beispielsweise sind die Hyperlinks und die Tags Elemente, die den Dokumenten gewisse Struktur geben, die benutzt werden kann, um Daten und semantische Informationen über die Dokumente zu extrahieren. [KoBl00] gibt einen Überblick über Web Content Mining für semi-strukturierte Dokumente.

Web Content Mining aus der DB-Sicht In den letzten Jahren wurden Datenbanktechniken verwendet, um webbezogene Probleme zu lösen [FlLM98]. Die Aufgaben wurden dabei nach [FlLM98] in drei Klassen, Modellierung und Anfragebearbeitung von Web-Daten, Information-Extraktion und -Integration, Konstruktion und Rekonstruktion von Web-Seiten, partitioniert. Bevor jedoch diese Aufgaben mit Datenbanktechniken gelöst werden können, müssen Daten-Repräsentations-Modelle für das World-Wide Web und seinen Daten entwickelt werden. Ziel einiger Techniken und Verfahren, die in diese Kategorie fallen, ist es, das Schema, welches den Web-Dokumenten unterliegt, zu bestimmen. Das Extrahieren und die Entdeckung von Schemata [Wang99],[Toiv99] oder das Erstellen von DataGuides [GoWi97],[NeAM97a],[GoWi99] sind typische Aufgaben von entsprechenden Applikationen. DataGuides sind eine strukturiere Zusammenfassung von semi-strukturierten Datenbanken, die meistens auch abgeschätzt werden [Abit97],[GoWi99]. DataGuides können darüber hinaus zur Anfrageformulierung und Anfragebearbeitung in einem semi-strukturierten Datenbank-Managementsystem eingesetzt werden [MAG+97]. Ein semi-strukturiertes Datenbank-Managementsystem wird gewöhnlich durch einen gerichteten Graphen modelliert [Abit97],[Bune97]. Ein oft verwendetes Modell zur Darstellung von semi-strukturierten Daten ist das Object Exchange Model (OEM) [CGH+94],[PaGW95]. Dieses Modell ermöglicht das Darstellen von Objekten als Knoten. Kanten, die annotiert sind, zwischen den Objekten stellen die Beziehung zwischen ihnen dar. Jedes Objekt wird durch eine ID eindeutig spezifiziert und besitzt einen Wert, der *atomar* (integer, gif, html,...) oder vom Typ *set*, bestehend aus mehreren Objekten.

Eine andere Herangehensweise ist das Erstellen von Multi-Layered Databases (MLDB) [ZaHa98]. Eine MLDB besteht aus mehreren Schichten. Auf der untersten Ebene sind beispielsweise in einem Web-Repository die kompletten Web-Dokumente abgespeichert. In den höheren Schichten werden die Daten in einer relationalen Datenbank gespeichert. Die Informationen werden von unten nach oben verallgemeinert. Dies geschieht beispielsweise durch Eliminieren von redundanten Tupeln und dem Benutzen von Konzepthierarchien [ZaHa98].

Im folgenden Abschnitt werden nun die drei Aufgabenbereiche erläutert:

1. **Modellierung und Anfragebearbeitung von Web-Daten:**
   Es wurden viele Anfragesprachen, wie z.B. Lorel [AQM+97],UnQL [PDHS96] oder MSL [CGH+94] für semi-strukturierte Daten und nicht explizit für Web-Daten entwickelt.
   Die Anfragesprachen für das World-Wide Web teilen sich zwei Generationen:

   (a) **1.Generation:**
   Die Anfragesprachen der 1.Generation versuchen, Textinhalte innerhalb der Dokumente kombiniert mit strukturellen Graphenmustern, die die Linkstruktur der Seiten beschreiben, anzufragen. Vertreter sind *WebSQL* [MeMM96] und *WebLog* [LaSS96].
   Im folgenden soll WebSQL näher erläutert werden. WebSQL versucht, das Web relational mit zwei virtuellen Relationen, den Dokumenten und den Anchors, zu modellieren. Die Dokumenten-Tabelle enthält für jedes Dokument ein Tupel, analog enthält die Anchor-Relation für jeden Anchor im Dokument ein Tupel. Diese Abstraktion des Webs erlaubt es, SQL als Anfragesprache zu benutzen, um Informationen aus dem WWW zu erhalten. Weil diese Relationen nur virtuell vorliegen, kann man nicht direkt auf ihnen operieren. In der *FROM*-Klausel der SQL-Anfrage werden dagegen die gewünschten Dokumente und Anchor materialisiert. Das grundsätzliche Verfahren, um einen kleinen Teil des Internets zu materialisieren besteht im Navigieren von bekannten URL's aus, wie die folgende Anfrage zeigt:

```
SELECT d.url,e.url,a.label
FROM Document d SUCH THAT
       "www.basketmc.de" ->* d,
     Document e SUCH THAT d => e,
     Anchor   a SUCH THAT a.base = d.url
WHERE a.hef = e.url
```

Die *FROM*-Klausel instantiiert zwei Dokumente d und e und einen Anchor. $d \rightarrow e$ bedeutet einen Link auf ein Dokument, welches auf dem gleichen Server ist, wie das referenzierende Dokument, ein $d \Rightarrow e$ bedeutet hingegen, daß das referenzierte Dokument auf einem anderen Server liegt.
Das Ergebnis der obigen Anfrage ist eine Liste von Tripeln der Form (d1,d2,label), wobei d1 auf dem lokalen Server wie die Startseite www.basketmc.de liegt, d2 ein Dokument ist, was irgendwo gespeichert sein kann und d1 referenziert d2 mit der Linkbeschreibung *label*.

(b) **2. Generation:**
Die zweite Generation wird auch „Web data manipulation language“ genannt. Sie unterscheiden sich von den Sprachen der 1. Generation in folgenden Fakten. Sie können die Struktur der Objekte zugreifen, die sie manipulieren, sie modellieren die innere Struktur der Dokumente sowie die externen Links, die auf die Web-Dokumente zeigen. Typische Vertreter sind WebOQL [ArMe00] und StruQL [FFLS97]

2. **Information-Extraktion und -Integration:**
Die Aufgabe von Information-Integration-Systemen besteht darin, Anfragen zu beantworten, die das Extrahieren und Kombinieren der Daten von verschiedenen WWW-Quellen fordern. Die Systeme kann man in zwei grundlegende Verfahren teilen, dem *Warehousing Verfahren* und dem *Virtual Verfahren*. Beim *Warehousing Verfahren* werden Daten aus unterschiedlichen Verfahren in ein Warehouse geladen. Werden nun Anfragen an das System gestellt, wird nur das Warehouse kontaktiert. Der Vorteil liegt in der adäquaten Performance zur Anfragezeit, der Nachteil liegt in der Aktualität der Daten. Das Warehouse wird ein Update gestartet, wenn sich die Daten ändern. Beim *Virtual Verfahren* mußsen bei Anfragebearbeitung die Daten aus den entsprechenden unterschiedlichen Dokumenten extrahiert werden. Der Vorteil liegt darin, daß die Daten immer aktuell sind, der Nachteil jedoch in der nicht gegebenen adäquaten Performance liegt.
Ein System welches heterogene Informationsquellen integriert, wurde im TSIMMIS-Projekt [CGH+94] entwickelt. In diesem Projekt wurde auch das OEM-Modell entwickelt. Die Architektur, welche unten ab-

gebildet wird, zeigt die Informationsquellen, welche unterschiedlichste Systeme und Formate (relationale Datenbanken, Flat-File-Repositories oder HTML-Dokumente) darstellen können. Die Wrapper, die in der *Wrapper Specification Language* (WSL) formuliert werden, kapseln die heterogenen Quellen. Die Mediatoren wiederum greifen mit der *Mediator Specification Language* (MSL) auf die Wrapper zu, um Informationen lokalisieren, selektieren und kombinieren zu können.
Die Wrapper in TSIMMIS werden durch Spezifikationen der Sprache WSL erzeugt. Mit diesen grammatikalischen Regeln kann eingegeben werden, welche Strukturelemente von HTML-Dokumenten beispielsweise selektiert werden sollen. Ein weitere Möglichkeit Wrapper zu erzeugen, stellt das *World-Wide Web Wrapper Factory* (W4F) [SaAz01] dar. Dieses Toolkit wird bei der Umsetzung des Systems in dieser Arbeit benutzt (siehe Kapitel 4.3).

3. **Konstruktion und Rekonstruktion von Web-Seiten:**
   Datenbanktechniken können auch eingesetzt werden, um Web-Seiten zu erstellen [FlLM98] Eine typische Herangehensweise ist das Erzeugen von HTML-Seiten, in dem eine Vielzahl von verschiedenen Datenquellen von Wrappern analysiert werden. Mediatoren setzen auf den Wrappern auf und erzeugen eine einheitliche Sicht auf die Daten, die dann durch eine geeignete Spezifikationssprache in HTML-Seiten konvertiert werden können [FlLM98].

3.1.2 Web Structure Mining

Die Verfahren, die dieser Kategorie angehören, versuchen, das Modell zu entdecken, welches die Linkstruktur des WWW beschreibt [CDG+99]. Das Modell basiert auf der Topologie der Hyperlinks mit oder ohne ihrer Beschreibung. Weiterhin können mit dem Modell die Web-Seiten kategorisiert werden. Beispielsweise werden die Experten- und Fan-Seiten mit einem Algorithmus [Klei99] identifiziert, die einem inhaltlichen Thema, z.B. Wetter in Deutschland, zugeordnet werden können. Verbesserungen von HITS wurden im Clever-System [CDG+99] und [BhHe98] vorgeschlagen. In dieser Arbeit spielt der HITS-Algorithmus eine besondere Rolle und wird in Kapitel 4.2.1 ausführlicher behandelt.

3.1.3 Web Usage Mining

Diese Kategorie [CoMS97] versucht die Daten zu interpretieren und zu analysieren, die während der Onlinesession eines Web-Surfers in den Logdateien des Web- und Proxyservers gespeichert werden. Die Daten können auch als Graphen repräsentiert werden [BBA+99],[PPK+00]. Die Applikationen dieser Kategorie können in zwei Klassen eingeteilt werden, das Lernen von Benut-

zerprofilen [Lang99] und das Lernen von Navigationsmustern [Spil99]. Web Usage Mining spielt in dieser Arbeit keine Rolle, deshalb wird an dieser Stelle interessierten Lesern beüglich dieser Kategorie folgende weiterführende Ausarbeitungen angeboten: [Sriv00],[MaSp00],[Cool00].

3.2 Metadaten

Das World-Wide Web hat sich in den letzten Jahren zum weltgrößten Datenspeicher entwickelt. Dadurch werden die Möglichkeiten zum weltweiten sofortigen Zugriff auf Informationen immer weiter verbessert, andererseits entstehen aber auch zunehmend Probleme dabei, in der Fülle der angebotenen Daten die tatsächlich relevanten Informationen zu finden (siehe Kapitel 2). Diese Probleme sind unter anderem auch dadurch bedingt, daß die Informationen im WWW zwar in maschinenlesbarer Form vorliegen, aber nicht maschinenverständlich sind. Der Einsatz von Metadaten, die Autoren die Möglichkeit geben, ihre Ressourcen, z.B. private Homepages im WWW, zu beschreiben und in einem flexiblen und maschinenverständlichen Datenformat abgespeichert werden, lösen das Problem. Die nachfolgende Definition der Metadaten basiert auf [Bern97]:

- **Metadaten sind maschinenverständliche Informationen über Web-Ressourcen oder andere Dinge:**
  Die Betonung liegt auf der Maschinenverständlichkeit, die durch eine wohldefinierte Semantik und Struktur der Metadaten erreicht werden soll. Der Ursprung der Bezeichnung Metadaten rührt daher, daß sie Daten über Daten darstellen. Mit fortschreitender Entwicklung der Metadaten-Sprachen, wie das Resource Description Framework (RDF) [RDFMS99],[RDFS99], und der Anwendungen, die Metadaten verarbeiten, werden sie zu einer starken Basis für ein Web aus maschinenverständlichen Informationen über verschiedenste Arten von Dingen, Konzepten und Ideen. Zum Beispiel kann ein Autor seine WWW-Ressource beschreiben, in dem er Angaben über sich, das Erstellungsdatum und dem Thema angibt. RDF wird in einem unten folgenden Abschnitt erklärt.

- **Metadaten sind Daten:**
  Informationen über Informationen sind in jeder Hinsicht wieder als Informationen zu betrachten. Metadaten können wie andere Daten gespeichert werden. Es gibt drei verschiedene Varianten der Verfügbarmachung von Metadaten [AlTu99],[Fiel94]:

  1. Die Metainformation wird extern vom Dokument gespeichert. Sie kann unabhängig abgerufen werden.

2. Die Metainformation und das Dokument sind in einem Container verpackt, welcher die Informationen bereitstellt, wenn sie benötigt werden.
3. Die Metainformation ist im Dokument selbst enthalten.

Eine Ressource kann Informationen über sich selbst enthalten sowie Informationen über andere Ressourcen. Durch das World-Wide Web Consortium[11] (W3C), wurden technische Umsetzungen standardisiert, z.B. ist die Angabe von Metadaten im HEAD-Teil eines HTML-Dokuments mittels der sogenannten <*META* >-Tags, die in der HTML 4.0-Spezifikation definiert wurden [RaLJ98], möglich. Mit Hilfe von PICS [PICS96] können andererseits auch Metadaten in einem zweiten Dokument abgelegt werden. Dieses Datenformat bietet die Möglichkeit, Aussagen über den Inhalt von Web-Seiten zu treffen. Primär ist PICS allerdings zur Beurteilung des Inhalts von Web-Seiten nach verschiedenen Kriterien geschaffen worden, so daß es Eltern beispielsweise möglich ist, Tools zu verwenden, die automatisch Web-Seiten herausfiltern, deren Inhalt nicht dem Alter ihrer Kinder entsprechen.

- **Metadaten können Metadaten beschreiben:**
  Metadaten können ihrerseits wieder bestimmte Eigenschaften haben, wie Ablaufdatum oder Urheberschaft.

In den obigen Punkten wurde schon auf technische Implementierungen des W3C für die Beschreibung von Web-Ressourcen Bezug genommen, die zum Standard vorangetrieben wurden. Genannt wurden die <*META* >-Tags und die PICS. Im unteren Abschnitt wird RDF in den Grundzügen vorgestellt. RDF wurde gewissermaßen als Nachfolger von PICS vom W3C als einen Standard für Web-Metadaten veröffentlicht.
Das fehlende Bindeglied zwischen der technischen Implementierung und der tatsächlichen Verwendung von Metadaten sind Vereinbarungen darüber, welche Informationen (Kriterien) über Daten man mit welchen Vokabeln und dazugehörenden Eigenschaften beschreibt. Solche Vereinbarungen, Schemata genannt, sind beispielsweise RDF-Schema, Dublin Core [Dubl99], MARC [Heer96], SOIF [SOIF96], LOM [LOM+98], IMS [IMSP99]. Außer dem RDF-Schema werden die restlichen Schemata in dieser Ausarbeitung nicht behandelt, da sie in dieser Arbeit nicht relevant sind.
Im folgenden Kapitel wird kurz auf RDF eingegangen.

Das Resource Description Framework (RDF)

RDF erlaubt das Austauschen von Metadaten zwischen verschiedenen Anwendungen und soll jetzt näher vorgestellt werden. Im W3C wurde dazu

[11] http://www.w3.org

eine Arbeitsgruppe gebildet, die nach [Lass97] aus Netscape, Microsoft, IBM, Nokia, OCLC und weiteren Mitgliedern zusammengesetzt ist und die Entwicklung des RDF zum Standard vorantreibt. Zunächst soll die Frage beantwortet werden, warum RDF überhaupt notwendig ist, existiert doch mit XML [XML98] ein universelles Datenformat. Eine Antwort gibt Tim Berners-Lee in [Bern98]. Als Beispiel soll die Aussage *„Der Autor dieses Papers ist Andreas“* dienen. In XML kann diese Aussage unterschiedlich repräsentiert werden:

<author >
    <uri >paper </uri >
    <name >Andreas </name >
</author >
oder
<document href= „paper“ >
    <author >Andreas </author >
</document >
oder auch
<document >
    <details >
        <uri >href= „paper“ </uri >
        <author >
            <name >Andreas </name >
        </author >
    </details >
</document >

Alle drei Varianten haben für lesende Personen die gleiche Bedeutung. Für einen Computer, der die Daten auswerten soll, stellen diese Repräsentationen völlig unterschiedliche XML-Graphen dar. Wenn die Daten semantisch interpretiert werden sollen, kommt es deshalb zu Problemen. Wenn die Frage „Wer ist der Autor dieses Papers“ beantwortet werden soll, müssen erst Abbildungen geschaffen werden, die die verschiedenen syntaktischen Darstellungen auf eine gemeinsame semantische Darstellung überführen. RDF definiert gewisse Einschränkungen bezüglich der zu verwendenden Datenstruktur und hilft damit, diese Mehrdeutigkeiten zu beseitigen. Dies wird mit RDF-Schema [RDFS99] erreicht. Im nun folgenden Abschnitt soll ein Einblick in das Datenmodell und der entsprechenden Syntax von RDF gegeben werden. Die vollständige Darstellung ist in [RDFMS99] gegeben.

Datenmodell und Syntax Das grundlegende Datenmodell basiert auf drei Arten von Objekten:

1. Ressourcen sind alle Dinge, die durch RDF-Ausdrücke beschrieben werden. Das sind Objekte, die über eine URI identifiziert werden können.

Außerdem können auch Objekte außerhalb des WWW, wie z.B. gedruckte Bücher, beschrieben werden.

2. Eigenschaften (Properties) sind spezielle Aspekte, Charakteristiken, Attribute oder Beziehungen, die eine Ressource beschreiben. Der Wert einer Eigenschaft kann ein Literal oder der Verweis auf eine andere Ressource sein. Jede Eigenschaft hat eine spezielle Bedeutung, definiert erlaubte Werte, die Arten von Ressourcen, die sie beschreiben kann und ihre Beziehungen zu anderen Eigenschaften. Diese Charakteristiken der Eigenschaft werden in Schemas definiert.

3. Aussagen (Statements) sind zusammengesetzt aus einer Ressource, einer Eigenschaft und ihrem Wert. Diese Teile können als Subjekt, Prädikat und Objekt der Aussage betrachtet werden.

Das RDF-Datenmodell ist von der Syntax der Datenrepräsentation unabhängig und die im folgenden verwendete XML-Syntax ist nur eine Möglichkeit der Umsetzung.

Einfache Aussagen: Eine einfache Aussage, wie „Die Universität von Waikato ist der Besitzer der WWW-Seite http://www.cs.waiakto.ac.nz/ml/weka/“ hat die folgenden Teile:

1. Subjekt (Ressource): http://www.cs.waikato.ac.nz/ml/weka/

2. Prädikat (Eigenschaft): Besitzer

3. Objekt (Wert, Literal): „Die Universität von Waikato“

Diese Aussage läßt sich eindeutig auf das RDF-Datenmodell abbilden. In der unteren Abbildung werden die Teile graphisch dargestellt.

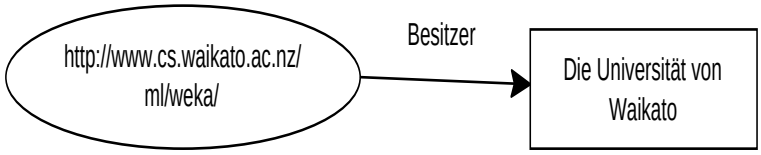


Abbildung 3.1: Modellierung einer einfachen Aussage

Ein Oval für die Ressource (http://www.cs.waikato.ac.nz/ml/weka/), ein Pfeil für die Eigenschaft (Besitzer) und ein Rechteck für das Objekt („Die Universität von Waikato“). In XML-Syntax würde diese Aussage folgendermaßen aussehen:

```
1. <?xml version="1.0"?>
2. <rdf:RDF
```

```
3.   <xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns\#"
4.   <xmlns:s="http://description.org/schema/">
5.   <rdf:Description about="http://www.cs.waikato.ac.nz/ml/weka/">
6.     <s:Besitzer>Die Universit"at von Waikato</s:Besitzer>
7.   </rdf:Description>
8. </rdf:RDF>
```

In den Zeilen 3 und 4 werden Namespaces deklariert: der RDF-Namespace wird dem Präfix rdf zugewiesen und ein Schema dem Präfix s. Ein Namespace [BrHL99] dient in XML zur Unterscheidung verschiedener Vokabulare. In dem Beispiel oben wird über die URL http://www.w3.org/1999/02/22-rdf-syntax-ns#, die ein Dokument mit der Beschreibung des RDF-Vokabulars identifiziert, eine eindeutige Beschreibung für den Namespace rdf zur Verfügung gestellt und festgelegt, welche syntaktischen Elemente dieser Namespace enthält. Auf diese Weise werden syntaktische Überschneidungen von verschiedenen Vokabularen verhindert. Jedes syntaktische Element wird durch seinen zugeordneten Namespace eindeutig. Mit *Description* in Zeile 5 beginnt die Beschreibung. Über *about* kann auf die zu beschreibende Ressource Bezug genommen werden. In diesem Fall wird der Besitzer spezifiziert.

Strukturierte Werte: Die Verwendung von strukturierten Werten soll anhand des Beispiels „Der Besitzer der WWW-Seite http://www.cs.waikato.ac.nz/ml/weka/ ist die Universität von Waikato und hat die E-mail wekasupport@cs.waikato.ac.nz“ erklärt werden. Die Eigenschaft Besitzer hat einen strukturierten Wert, wie die unten stehende Abbildung zeigt.

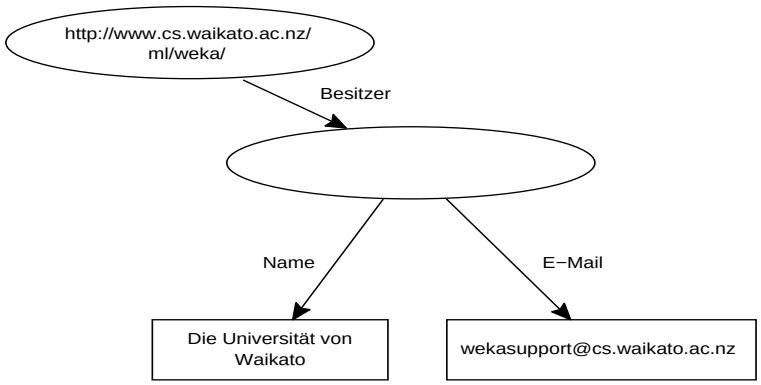


Abbildung 3.2: Modellierung einer strukturierten Aussage

Analog der obigen Vorgehensweise kann auch hier eine entsprechende XML-Syntax angegeben werden [RDFMS99].
Container: In RDF existieren drei Arten von Containern:

1. **Bag** ist eine ungeordnete Liste von Ressourcen oder Literalen

2. **Sequence** ist eine geordnete Liste von Ressourcen oder Literalen
3. **Alternative** ist eine Liste von Ressourcen oder Literalen, die Alternativen darstellen

Beispiele und weitergehende Aussagen über Container sind in [RDFMS99] zu finden.

Schemadefinition Das RDF-Modell beschreibt Ressourcen durch Eigenschaften, über die jedoch zunächst keine hinsichtlich ihrer Charakteristiken näheren Aussagen gemacht werden. Diese Charakteristiken der Eigenschaften werden über Schemas definiert. In diesen Schemas wird für jede Eigenschaft festgelegt, was sie für eine Bedeutung hat, welche Werte für diese Eigenschaft erlaubt sind, welche Arten von Ressourcen diese Eigenschaft besitzen und welche Beziehungen sie zu anderen Eigenschaften hat. Das RDF-Schema-System ist ähnlich dem Klassensystem objektorientierter Sprachen. Allerdings besteht ein wesentlicher Unterschied: Bei den objektorientierten Sprachen wird eine Klasse von Objekten dadurch definiert, welche Eigenschaften sie besitzt. Bei den RDF-Schemas wird von der Idee ausgegangen, daß Klassen von Objekten auch identifizierbar sind, ohne daß ihnen sofort bestimmte Eigenschaften zugeordnet werden. Dagegen wird für jede Eigenschaft festgelegt, für welche Klassen von Objekten sie anwendbar ist. Dies stellt eine wesentlich flexiblere Lösung dar, da die interessierenden Eigenschaften ganz wesentlich vom Anwendungsbereich abhängen. Mit dem RDF-Schema-System können bei Bedarf jederzeit neue Eigenschaften definiert und bestimmten Klassen zugeordnet werden, ohne daß bestehende Anwendungen beeinträchtigt werden. Außerdem können so definierte Eigenschaften mit ihrer Charakteristik wie Bedeutung, Wertebereich usw. sofort für eine andere Klasse unter Beibehaltung dieser Charakteristik verwendet werden. In dem Dokument [RDFS99] wird nun nicht ein Schema definiert, in dem verschiedene Klassen und Eigenschaften festgelegt werden, sondern es wird eine „Schema Definition Language“ definiert, mit deren Hilfe die eigentlichen Schemas definiert werden. Diese eigentlichen Schemas werden auch als Vokabulare bezeichnet, da sie definieren, welche Eigenschaften mit welcher Bedeutung für Beschreibungen zu Verfügung stehen. Unten stehende Abbildung verdeutlicht den Zusammenhang.

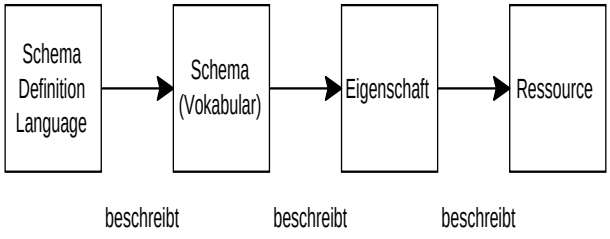


Abbildung 3.3: Das RDF-System

Klassen und Eigenschaften: Wie in [RDFMS99] beschrieben, kann eine Ressource eine Instanz einer oder mehrerer Klassen sein, was durch die Eigenschaft *rdf:type* ausgedrückt wird. Die Bildung von Klassenhierarchien ist ebenfalls möglich, dazu kann die Eigenschaft *rdfs:subClassOf* verwendet werden. In [RDFS99] werden außerdem verschiedene Ressourcen definiert, die es ermöglichen, Aussagen über Beschränkungen für die Verwendung von Eigenschaften und Klassen zu machen. So wird zum Beispiel ermöglicht, gültige Werte für Eigenschaften zu definieren und festzulegen, welchen Klassen eine Eigenschaft sinnvollerweise zugeordnet werden kann. Allerdings wird kein Mechanismus festgelegt, der diese Beschränkungen durchsetzt. Ob und wie die Beschränkungen durchgesetzt werden, bleibt den einzelnen Anwendungen überlassen. Es wird lediglich vorgeschlagen, diese Informationen z.B. bei einem Validierungs-Programm dafür zu verwenden, Fehler zu finden, oder bei einem Programm zur Bearbeitung von Metadaten daraus Vorschläge für sinnvolle Werte einer Eigenschaft zu generieren.

Constraints: Weiterhin können den Klassen und Eigenschaften Beschränkungen zugeordnet werden. Insbesondere werden damit die Konzepte von Domain und Range umgesetzt. Ausführliche Ausführungen können [RDFS99] entnommen werden.

3.3 Allgemeines über Ontologien

Der Begriff der Ontologie bezeichnete ursprünglich eine philosophische Disziplin, der seit Anfang der 90er Jahre in Teilgebieten der KI, z.B. Knowledge Engineering (KE), durch [Grub93a],[Grub93b] etabliert wurde. Ontologien beschreiben Wissen in strukturierter, computerverständlicher Form und können so in verschiedenen Bereichen für die Lösung von Aufgaben eingesetzt werden:

1. Katalogdienste, z.B. Yahoo [LaFi99]
2. Intelligente Suchmaschinen, z.B. Ontobroker [DEFS99], SHOE [LuSR96], Getess [SBB+99], OntoSeek [GuMV99]

Die Anwendungspalette ist, wie oben gezeigt, sehr breit. Interessant für die Umsetzung der Konzepte ist der Punkt „Intelligente Suchmaschinen“. Die Suchmaschine Getess zum Beispiel stellt dem Anwender eine Suche nach Informationen in der Domäne Tourismus zur Verfügung.
In Anlehnung an die Definition in [Erdm01] läst sich folgende Definition einer Ontologie formulieren: „Eine Ontologie ist eine formale und explizite Spezifikation der Begriffe eines Ausschnittes der Welt“. Auf eine mathematische Definition der Ontologie soll hier verzichtet und stattdessen auf [Erdm01] verwiesen werden. An dieser Stelle sollen die wichtigsten Bestandeile einer Ontologie genannt werden.

1. **Konzept- oder Klassenbezeichner:** Das sind Strukturelemente der Ontologie, die Begriffe des modellierten Wissens einer Domäne darstellen.

2. **Attributbezeichner:** Attribute können Konzepte mit anderen Konzepten als auch mit atomaren Werten in Beziehung setzen.

Inhalte einer Domäne können meistens durch die Konzepte allein nicht ausreichend beschrieben werden. Es muß noch ein Domänenlexikon erstellt werden. Zu jedem Konzept werden Ausprägungen ins Lexikon gespeichert, sozusagen Synonyme zu den Konzepten. Verwiesen sei an dieser Stelle auf das Getess-Projekt [SBB+99], bei dem unter anderem auch ein Domänenlexikon erstellt wurde.

maschinenverständliche Repräsentation der Ontologie Die Struktur eines Anwendungsbereiches kann mittels einer Ontologie dargestellt werden. Die Ontologie kann durch Modellierungssprachen maschinenverständlich repräsentiert werden. Diese Sprachen stellen syntaktische Elemente zur Verfügung, um die Ontologie zu beschreiben und darzustellen. Eine weit verbreitete Sprache, DAML+OIL, sei an dieser Stelle erwähnt und erläutert: DAML+OIL wird eigentlich nur DAML (DARPA Agent Markup Language) genannt. Diese Modellierungssprache hat aber viele Aspekte von OIL (*Ontology Inference Layer* oder auch *Ontology Interchange Language*) [FHH+00a],[FHH+00b] geerbt.
DAML+OIL, gehört wie OIL zu der Sprachfamilie des *Semantic Web*. Das *Semantic Web* ist eine Vision des World-Wide Web-Erfinders und W3C-Direktors Tim Berners-Lee. Die Grundidee beruht auf der Aussage über die im Web zugänglichen Informationsquellen. Sie sollten mehr sein als verlinkte Texte, vielmehr ein Netz miteinander verknüpfter und durch Metadaten angereicherter Informationseinheiten bilden, das für automatische Recherchen oder inhaltliche Verknüpfungen zugänglich sind. Für die Realisierung der Vision vom *Semantic Web* kommt eine W3C-Entwicklung in Form des RDF-Datenmodells (siehe Kapitel 3.2) zum Einsatz. Durch RDF-Schema können die Objekte und Properties von RDF zu Klassen bzw. Property-Typs zusammengefaßt und in einem Schema definiert werden. Für das *Semantic Web* ergibt sich dadurch eine Schichtung von aufeinander aufbauenden Sprachen, die semantisch immer ausdrucksstärker werden (XML, RDF, RDFS, ...). Die folgende Abbildung stellt die Sprachen und ihre Abhängigkeiten dar.

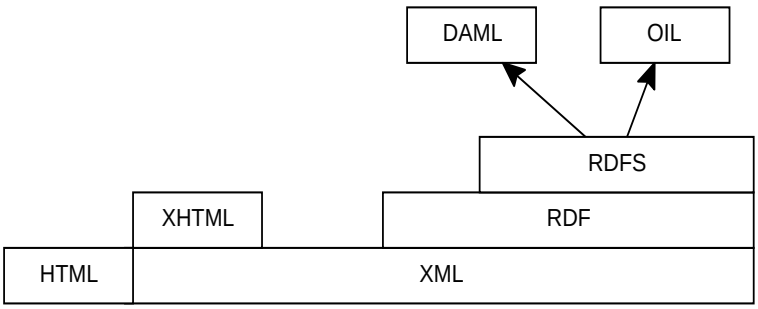


Abbildung 3.4: Die XML-Sprachfamilie und ihre Abhängigkeiten

An dieser Stelle sei jetzt kurz auf OIL eingegangen, bevor die Eigenschaften von DAML erläutert werden.
OIL ist eine Web-basierte Repräsentation und Inference Layer, der Ebene des Schlußfolgerns, für Ontologien. OIL kann als Kombination mehrerer Aspekte unterschiedlicher Disziplinen betrachtet werden:

1. Frame-basierte Sprachen:
   Die zentralen Elemente der Modellierung sind Frames (Klassen) und Slots (Attribute). Die Frames bilden eine Klassenhierarchie. OIL verwendet die grundlegenden Primitiven Frame-basierter Systeme in seiner Sprache. Zum einen das Konzept der Klassen und der Definition von Superklassen und Attributen. Außerdem können Relationen dabei auch als eigenständige Entitäten definiert werden. Aus diesem Grund können Relationen eigene Attribute haben und in einer Hierarchie angeordnet werden.

2. Deskription Logik (DL):
   DLs beschreiben Wissen in Form von Konzepten und Rollen. Die Semantik von Ausdrücken der DL kann mathematisch präzise beschrieben werden, wodurch zum Beispiel automatisches Schließen möglich wird. OIL übernimmt diese formale Semantik aus der DL und damit auch die Unterstützung für automatisches Schließen.

3. Web-Standards:
   Die syntaktischen Elemente von OIL sind in den Standards des W3C definiert, zum Teil basieren sie auf einer XML DTD bzw. eines XML Schemas. Außerdem ist OIL eine Erweiterung von RDF. Einige der Modellierungprimitiven von OIL können direkt in RDF(S) ausgedrückt werden, die anderen werden in einem eigenen Namensraum neu definiert.

DAML+OIL ist eine semantische Markupsprache für Web-Ressourcen. Es baut auf RDF und RDF Schema auf und erweitert diese Sprachen mit mächtigeren Modellierungsprimitiven. Für eine ausführliche Beschreibung von DAML sei auf [HaPH01] verwiesen. Nachfolgend seien einige Erweiterungen genannt:

1. Einschränkungen von Properties

```
<rdf:RDF
  ...
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  ...
>
...
<daml:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#hatVater"/>
      <daml:toClass rdf:resource="#Person"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
...
```

2. Angabe von speziellen Konstrukten, um zusätzliche Informationen für Inferenzsysteme zu modellieren, wie z.B. inverse Relationen

```
<rdf:RDF
  ...
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  ...
>
...
<daml:ObjectProperty rdf:ID="hatKind">
  <daml:inverseOf rdf:resource="#hatEltern"/>
</daml:ObjectProperty>
...
```

3. Verwenden von boole'schen Operatoren bei Klassendefinitionen, wie zum Beispiel Vereinigung (daml:unionOf), Schnitt (daml:intersectionOf) oder Komplement (daml:complementOf)

```
<rdf:RDF
  ...
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  ...
>
...
```

```
<daml:Class rdf:ID="Auto">
  <rdfs:subClassOf>
    <daml:Class>
      <daml:complementOf rdf:resource="#Person"/>
    </daml:Class>
  </rdfs:subClassOf>
</daml:Class>
...
```

4. Definition von eigenen Datentypen ist mittels XML-Scheme möglich (Somit kann man außer dem einzigen RDF-Datentyp Literal noch weitere benutzen)

```
<rdf:RDF
  ...
  xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
  ...
>
...
<xsd:simpleType name="über12">
  <xsd:restriction base="xsd:decimal">
  <xsd:minInclusive value="13"/>
  </xsd:restriction>
</xsd:simpleType>
...
```

5. DAML erbt von OIL die wohldefinierte Semantik der Description Logics

4 Realisierung eines domänspezifischen Suchdienstes

In Kapitel 2 wurden Suchmaschinen untersucht und eine typische Klassifikation, in den crawler-basierten Ansatz, den Katalogdiensten, den Meta-Suchmaschinen und den spezialisierten Diensten, vorgenommen.
In diesem Kapitel wird nun die konkrete Umsetzung eines Suchdienstes vorgestellt. Dieser Dienst soll domänspezifische Anfragen beantworten. Das Themengebiet, welches von der erstellten Suchmaschine abgedeckt werden soll, umfaßt das

„Wetter in Deutschland“

Die Ergebnisse von Anfragen, wie beispielsweise „Erstelle den aktuellen Wetterbericht von Rostock“ oder „Bitte die Bauernregeln für den Januar auflisten“ beinhalten relevante Informationen, die aus verschiedenen Quellen aggregiert und integriert werden und in Form eines verlinkten Dokumentes, dem Anwender des Systems, angezeigt werden sollen.
In dieser Anforderung stecken viele Fragen, die vor der Implementierung des Systems, konzeptionell erfaßt und beantwortet werden müssen.

1. *Wie ist die Domäne inhaltlich aufgebaut?*
2. *Wie werden die Wissensstrukturen der Domäne im World-Wide Web lokalisiert und markiert?*
3. *Wie verarbeitet das System die Suchanfragen des Anwenders?*
4. *Mit welchen Mitteln können die Ergebnisse aggregiert werden und in einem verlinkten Dokument dargestellt werden?*

In den nächsten Abschnitten werden diese Fragen beantwortet.
Kapitel 4.1 beinhaltet die Domänenanalyse. Ausgehend von WWW-Seiten, deren Autoren inhaltliche Informationen, nachfolgend auch Konzepte der Wetterdomäne genannt, über das **„Wetter in Deutschland“** publiziert haben, werden diese Daten analysiert. *Eigenschaften* dieser Inhalte werden erfaßt, z.B. welche Daten muß man darstellen können, um das Biowetter zu erfassen. Das sind beispielsweise diverse Krankheiten, wie Migräne, Bronchitis, Depressionen, die regional zu bestimmten Zeiten und Wettersituationen bei Personen auftreten können. Weiterhin werden *strukturelle Zusammenhänge* abgeleitet, um festzulegen, welche Konzepte aus der Domäne hierarchisch angeordnet werden können. Örtliche Angaben lassen sich z.B. in einer Hierarchie darstellen (Land, Bundesland, Landkreis, Ort). Als letzten Punkt müssen die Beziehungen zwischen den Inhalten in der Domäne entdeckt werden. Eine mögliche Beziehung ist die zeitliche, örtliche Angabe beim Wetterbericht („aktuelle Wetterlage in Rostock“). Das Erfassen der Konzepte und ihrer Eigenschaften, das

Darstellen der Konzepte in einer Hierarchie und die Beziehungen zwischen den Konzepten, können in einer Ontologie modelliert werden (siehe Kapitel 3.3). Abschließend muß ein umfangreiches Lexikon aufgebaut werden, welches zu den erfaßten Konzepten Ausprägungen beinhaltet, die auf den HTML-Dokumenten von den Autoren verwendet wurden, um inhaltliche Informationen zu den Konzepten darzustellen. Beispielsweise bedeuten „Biowetter“ und „Gesundheitswetter“ oder „Medizinwetter“ das gleiche.
In Kapitel 4.2 wird vorgestellt, wie die Struktur der Ontologie persistent in einer Datenbank gespeichert werden kann.
Die Systemarchitektur wird in Kapitel 4.3 vorgestellt und beschrieben. Dabei wird auf die zwei Teile der Funktionalität des Systems näher eingegangen und erläutert, an welchen Stellen im System die Prozesse ablaufen.

1. **Initialisierung des Suchdienstes:** Auf Serverseite wird die Initialisierung (Kapitel 4.4) vorgenommen. Dabei werden mit einem geeigneten Verfahren die WWW-Seiten im Internet lokalisiert, die inhaltliche Informationen der Domäne „Wetter in Deutschland“ beinhalten. Dies kann mit dem HITS-Algorithmus [Klei99] (Kapitel 4.4.1) erreicht werden. Dieses Verfahren ist anfrage-abhängig, d.h., zu einem gegebenen Suchterm wird in einem ersten Schritt eine fixe Anzahl von HTML-Dokumenten von einer Suchmaschine bestimmt und in folgenden Schritten die relevantesten HTML-Seiten (Expertenseiten) bezüglich eines Rankingwertes berechnet. Der Algorithmus von Kleinberg wird im Rahmen der globalen Strukturanalyse (Kapitel 4.4.1) in dem System ablaufen.
   Die lokale Strukturanalyse (Kapitel 4.4.2) beinhaltet die Untersuchung der WWW-Seiten, die vom HITS-Algorithmus als die relevantesten Web-Dokumente, die Expertenseiten, markiert wurden. Ergebnis dieser lokalen Strukturanalyse sollen HTML-Strukturen sein, die sehr oft auf diesen WWW-Seiten markierbar sind, deren Inhalte Wissensstrukturen der Domäne „Wetter in Deutschland“ darstellen.
   Kapitel 4.4.3 umfaßt einen Algorithmus, der mit Hilfe des W4F-Toolkit's [SaAz01] die Wissensstrukturen aus den Expertenseiten extrahiert. Die HTML-Strukturen, die in der lokalen Strukturanalyse entdeckt wurden, werden von den Expertenseiten angefragt und der enthaltene Inhalt in einer Datenbank gespeichert.

2. **Anwenden des Suchdienstes:** Der Anwender hat nach der Initialisierung die Möglichkeit, die Suchmaschine zu verwenden (Kapitel 4.5). Er kann auf Client-Seite in einem Web-Browser eine Anfrage formulieren, die dann auf Server-Seite vom System verarbeitet wird (Kapitel 4.5.1). Das Ergebnis der Anfrage wird dem Anwender im Browser in Form eines verlinkten Dokumentes angezeigt (Kapitel 4.5.2).

4.1 inhaltliche Analyse der Wetterdomäne

Die Aufgabe der Diplomarbeit lautet, einen Suchdienst zu entwickeln, der für einen eingegrenzten Anwendungsbereich Anfragen genau beantworten soll und die Ergebnisse von solchen Anfragen in Form eines verlinkten Dokumentes darstellen soll. Es wird die Domäne „Wetter in Deutschland“ gewählt, aber man kann den Suchdienst genauso gut für andere Anwendungsbereiche, wie z.B. Sportnachrichten, Nachrichten über Musikgruppen, allgemeine Wirtschaftsnachrichten oder Informationen rund um die Börse, enwickeln.
Es gibt eine Reihe von Anbietern im Internet, die über das Wettergeschehen in Deutschland informieren. Die Domänenanalyse macht es notwendig, viele von ihnen inhaltlich zu untersuchen und letztendlich festzulegen, welche Informationen der Suchdienst erfaßt und anfragbar macht. Diese Ergebnisse werden in einer Ontologie (siehe Kapitel 3.3) modelliert.

Identifikation von Konzepten Im World-Wide Web wurden 35 „geeignete“ Web-Seiten und Domains identifiziert und analysiert. Unter dem Begriff des „meteorologischen Aspektes“ kann man eine Menge von Themengebieten identifizieren, die als Konzepte in einer Ontologie modelliert werden können und auf den untersuchten Web-Seiten und Domains inhaltlich erschlossen werden:

1. **Wetterbericht:**

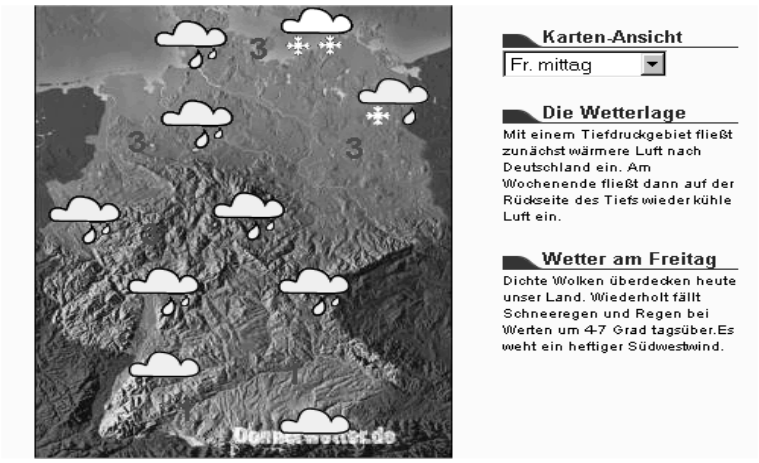


Abbildung 4.1: Wetter in Deutschland (Quelle: donnerwetter.de)

Abhängig von einem bestimmten Zeitpunkt, üblicherweise die aktuellen Informationen, werden Wetterdaten für Deutschland (Abbildung 4.1) gezeigt, oder aber der Anwender hat die Möglichkeit, Daten von bestimmten Orten in Deutschland anzufordern, indem er die Örtlichkeit an den Server per Formular sendet (Abbildung 4.2). Diese Wetterdaten

können z.B. Temperatur, Bewölkung, Niederschlagswahrscheinlichkeit, Windrichtung und Windstärke charakterisieren.

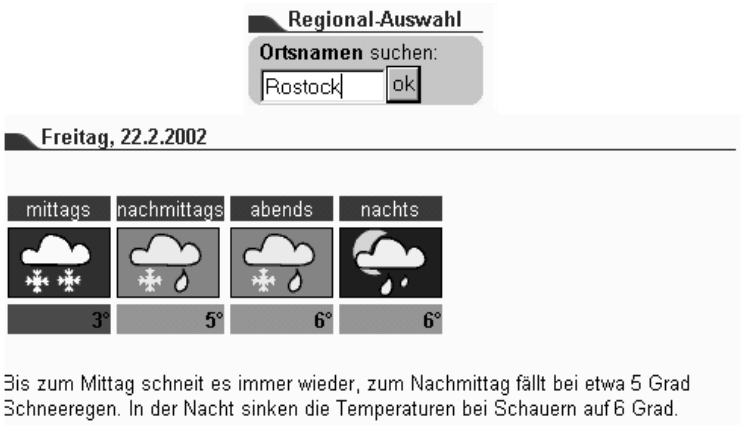


Abbildung 4.2: Wetterdaten für Rostock (Quelle: donnerwetter.de)

Üblich ist außerdem, Wetteraussichten für die kommenden Tage anzugeben, z.B. morgiges Wettergeschehen oder die 3-Tage-Aussichten. Diese Informationen sind in der nächsten Abbildung 4.3 dargestellt.

Abbildung 4.3: Wetteraussichten für Rostock (Quelle: donnerwetter.de)

2. **Biowetter:** Differenzierter gestaltet sich die Strukturierung der Informationen für das Biowetter. Auf einigen WWW-Seiten findet man nur Informationen über bestimmte Regionen Deutschlands. Beispielsweise werden auf der WWW-Seite der Domaine *http://www.wetteronline.de*, die unten abgebildet ist, nur Daten für Norddeutschland, Westdeutschland, Süddeutschland, Ostdeutschland und dem Alpenraum im Internet publiziert.

Abbildung 4.4: Biowetter für Regionen Deutschlands (Quelle: wetteronline.de)

Andere hingegen (Abbildung 4.5), bieten, wie auch bei dem Wetterbericht die Möglichkeit an, für bestimmte Städte Deutschlands die Biowetter-Informationen anzufordern.

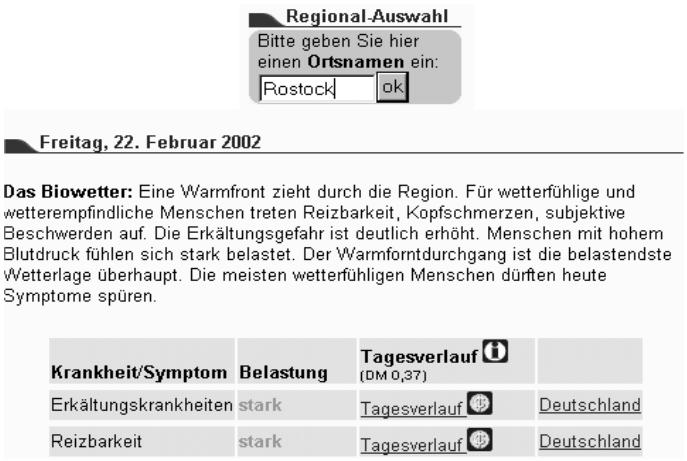


Abbildung 4.5: Biowetter für Rostock (Quelle: donnerwetter.de)

Daten des Biowetters sind hauptsächlich eine Menge von Krankheiten, die bei bestimmten Wetterbedingungen auftreten können. Ein bekanntes Beispiel ist die *Migräne*. In der Abbildung 4.6 wird ein Formular auf einer WWW-Seite gezeigt. Man hat die Möglichkeit, eine Krankheit aus einer vorgegebenen Liste auszuwählen. Daraufhin wird dann die Belastung und das Auftreten dieser Krankheit in Deutschland durch Symbole graphisch dargestellt.

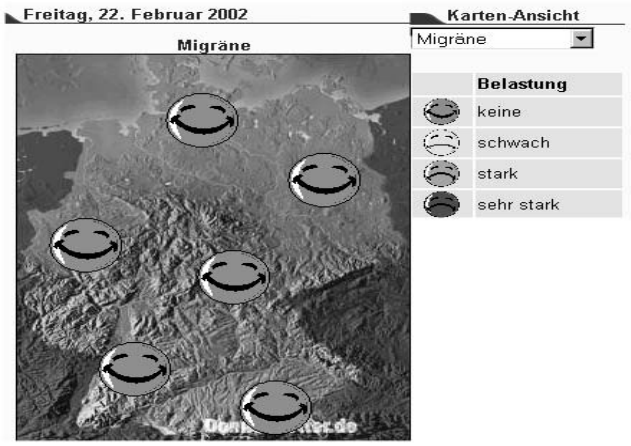


Abbildung 4.6: Biowetterdaten (Krankheiten) (Quelle: donnerwetter.de)

3. **Bauernregeln:** In den zurückliegenden Jahrhunderten mußten die Menschen, die in der Argrarwirtschaft tätig waren, die „Zeichen der Natur“ deuten können. Sie versuchten, aus den Wetter-Intensitäten der verschiedenen Jahreszeiten, wie z.B. des Winters, abzuleiten, wie warm oder wie kalt es in den darauffolgenden Jahreszeiten, im Frühling in diesem Beispiel, wird. Das taten sie keineswegs aus Langeweile. Früher gab es noch keine meteorologischen Stationen und wissenschaftlichen Wetterbeobachtungen. Die Wetterdeutungen beruhten auf jahrelangen Erfahrungen der Bauern, um bestmögliche Ergebnisse bei den Ernten zu erzielen oder sogar Ernteausfälle bei eventuellen Wetterkapriolen (Eisheilige im Mai) zu vermeiden. Es entstanden Bauernregeln, die von Generation zu Generation weitergegeben wurden. Mit der Zeit wurden viele verfälscht. Einige aber erwiesen sich als richtig.
   Einige Web-Anbieter stellen Informationen über Bauernregeln bereit. So gibt es Autoren, die die Bauernregeln des gerade aktuellen Monats darstellen (Abbildung 4.7).

Abbildung 4.7: einige Bauernregeln für den Februar (Quelle: http://home.arcor.de/wetterstation_grosserkmannsdorf/index.htm)

Andere stellen dagegen HTML-Formulare zur Verfügung, die es erlauben, sich bei Bedarf beliebige Bauernregeln anzeigen zu lassen (Abbildung 4.8).

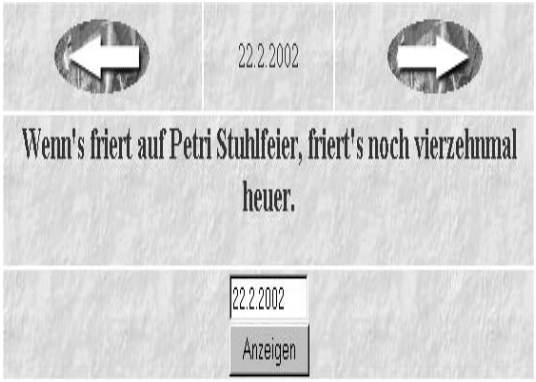


Abbildung 4.8: eine beliebige Bauernregel (Quelle: agri.ch)

4. **100-jähriger Kalender:** In den Jahren 1652 bis 1658 beobachtete Mauritius Knauer tagtäglich das Wetter. Keine astronomische, klimatische oder atmosphärische Erscheinung entging ihm. Irgendwann erkannte der Abt, daß er sein Wissen vielen Menschen zugänglich machen mußte. Knauer nannte seine Schrift **„Galendarjum Oeconomicum Practicum Perpetuum“** . Er glaubte, daß sieben Beobachtungsjahre für eine dauerhafte Wettervorhersage ausreichten, da sich nach seinen astrometeorologischen Ansichten die Witterungsabläufe entsprechend der Planetenfolge Mond, Saturn, Jupiter, Mars, Sonne, Venus, Merkur wiederholten.
   Dr. Christoph von Hellwig aus Thüringen hatte sich schon längere Zeit mit astrologischen und medizinischen Schriften befaßt. Als er die Bekanntschaft von Dr. Mauritius Knauer machte, witterte er sofort ein einträgliches Geschäft. Er verkürzte die vom Abt erstellte und berechnete Planetentafel von 1600 bis 1912 auf hundert Jahre, nämlich von 1701 bis 1800, und ließ den Kalender 1704 drucken. Im Jahre 1720 versah der Verleger Weinmann aus Erfurt die Schrift mit dem Titel ”100-jähriger Kalender”. Bis zum Jahre 1860 wurde dieser Kalender in über 180 Auflagen gedruckt und verbreitet.
   In Abbildung 4.9 sind als Beispiel Einträge aus dem 100-jährigen Kalender für den Monat März gezeigt.


März

1. - 7. rauh, kalt und windig
8. - 17. sehr kalt, aber kein Schnee
18. - 19. Wind, Schnee und Regen
20. - 21. hell, vormittags sehr kalt und es gefriert
22. - 25. kalt, doch wenig Schnee, aber Wind und zuletzt Nebel
26. - 31. sind die Tage aber meistenteils schön

Abbildung 4.9: 100-jähriger Kalender für den März 2002 (Quelle: altmuehltal.de)

Ohne genaue Temperaturen zu nennen, geht der Autor auf bestimmte Zeiträume im Monat ein und beschreibt für sie das prognostizierte Wettergeschehen.

Die vier Konzepte sind nachfolgend noch einmal strukturell dargestellt.

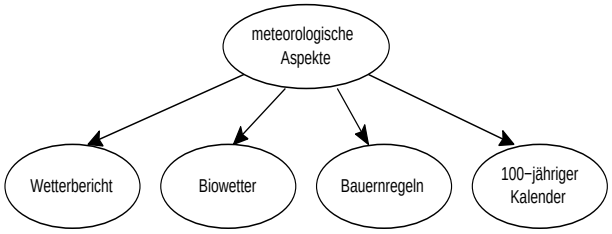


Abbildung 4.10: hierarchische Darstellung des meteorologischen Aspektes

Eigenschaften der Konzepte Während des Vorstellens der Konzepte der Domäne Wetter wurden schon die Eigenschaften genannt, die das Konzept näher beschreiben. Beispielsweise sind für den Wetterbericht Angaben über Temperatur und Bewölkung beschreibende Werte. Folgende Tabelle stellt die Eigenschaften der vier Konzepte dar.

Konzept	Eigenschaften
Wetterbericht	Temperatur, Zeit1, Region1, Bewölkung, Bericht1
Wetterbericht	Niederschlagsmenge, Windstärke, Windrichtung
Biowetter	Krankheit, Zeit2, Region2, Bericht2
Bauernregeln	Zeit3, Bericht3
100-jähriger Kalender	Zeit4, Bericht4

In der nächsten Tabelle sind mögliche Werte der Eigenschaften aufgelistet:

Eigenschaften	Werte
Temperatur	20 °C, -2 °C, ...
Zeit1	heute, morgen, 3-Tage-Aussichten, 15.02.2002, ...
Region1	Mecklenburg/Vorpommern, Rostock, ...
Bewölkung	bedeckt, heiter, wolkenlos, ...
Bericht1	Atlantische Tiefs bringen viel Regen..., ...
Niederschlagsmenge	8mm, ...
Windstärke	4, ...
Windrichtung	SSO, NW, O, ...
Krankheit	Migräne, Gliederschmerzen, ...
Zeit2	heute, morgen, 3-Tage-Aussichten, 15.02.2002, ...
Region2	Mecklenburg/Vorpommern, Rostock, ...
Bericht2	Empfindliche Menschen können mit den..., ...
Zeit3	Januar, Februar, 15.02., 15.Februar, 15.02.2002, ...
Bericht3	Kräht der Hahn auf dem Mist..., ...
Zeit4	Januar, Winter, 01.01.2002 - 15.01.2002, ...
Bericht4	trüb und mittelmäßig kalt, unbeständig mit Wind, ...

Beziehungen zwischen Konzepten Die Eigenschaften Zeit aller vier Konzepte und die Eigenschaft Region der Konzepte *Wetterbericht* und *Biowetter* stellen Werte dar, die komplexerer Struktur sind als beispielsweise die *Temperatur*. Die Region kann z.B. vieles sein, das Land Deutschland oder das Bundesland Niedersachsen oder gar eine Stadt. Genauso verhält es sich mit der Zeit. Es kann ein Zeitpunkt, 15.02.2002, oder ein Zeitintervall im Fall des 100-jährigen Kalenders angegeben werden, 01.01.2002 - 15.01.2002. Aus diesem Grund werden die beiden Konzepte *zeitlicher Aspekt* und *geographischer Aspekt* als Konzepte in die Ontologie aufgenommen und strukturiert. In der Ontologie wird deshalb kenntlich gemacht, daß folgende Konzepte in Beziehung stehen:

1. Wetterbericht & zeitlicher Aspekt
2. Wetterbericht & geographischer Aspekt
3. Biowetter & zeitlicher Aspekt
4. Biowetter & geographischer Aspekt
5. Bauernregeln & zeitlicher Aspekt
6. 100-jähriger Kalender & zeitlicher Aspekt

In der folgenden Abbildung wird die jeweilige Hierarchie der beiden Konzepte, *zeitlicher Aspekt* und *geographischer Aspekt* dargestellt.

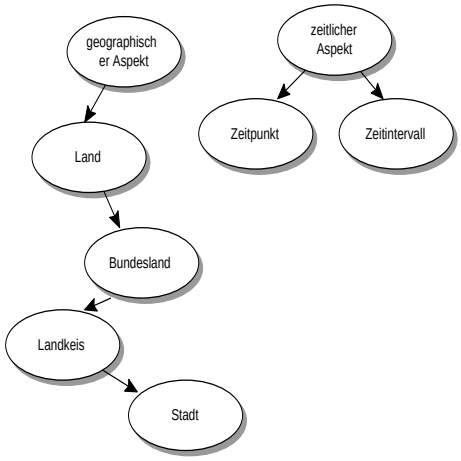
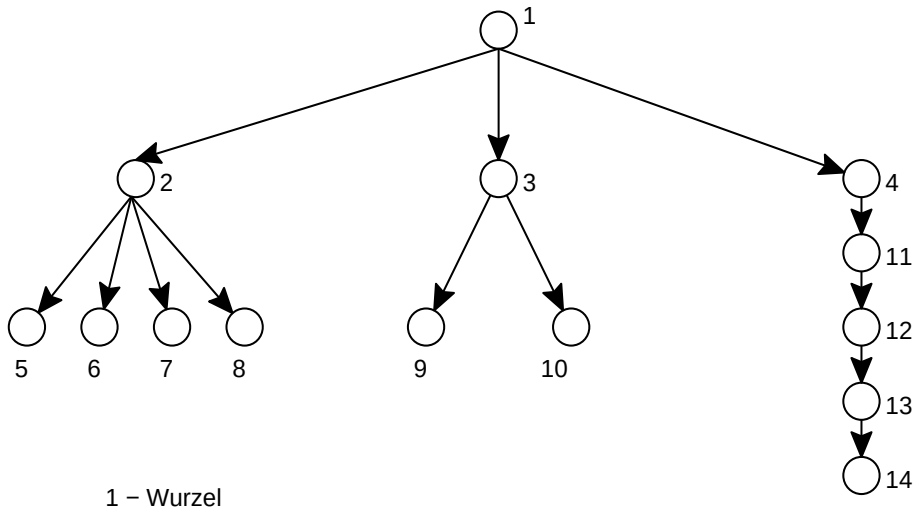


Abbildung 4.11: hierarchische Darstellung des geographischen und zeitlichen Aspektes

Nachdem die Konzepte, die dazugehörigen Eigenschaften und die Beziehungen zwischen den Konzepten herausgearbeitet wurden, können die Ergebnisse in Form einer Ontologie zusammengefaßt und modelliert werden.



1 – Wurzel
2 – meteorologischer Aspekt
3 – zeitlicher Aspekt
4 – geographischer Aspekt
5 – Wetterbericht
6 – Biowetter
7 – Bauernregeln
8 – 100–jähriger Kalender
9 – Zeitpunkt
10 – Zeitintervall
11 – Land
12 – Bundesland
13 – Landkreis
14 – Ort

Abbildung 4.12: Ontologie zur Modellierung der Wetterdomäne

Auf eine Darstellung der Eigenschaften der Konzepte und der Beziehungen unter den Konzepten wird verzichtet, da sonst die Abbildung zu unübersichtlich wird.

Ausprägungen der Konzepte Die Ontologie gibt Auskunft darüber, wie das Wissen in der Wetterdomäne strukturiert ist. Mit dieser Struktur kann man aber noch nicht alleine die Konzepte mit ihren Eigenschaften aus dem Internet von den relevanten HTML-Dokumenten automatisch extrahieren. Man kann und darf nicht davon ausgehen, daß z.B. alle Web-Autoren den String „Biowetter“ benutzen, um die sich anschließenden Informationen auf der Web-Seite als Eigenschaften des Konzeptes *Biowetter* zu kennzeichnen. Die folgende Abbildung 4.13 stellt einen Link mit der Beschreibung „Gesundheitswetter“ dar. Die Linkadresse zeigt auf ein HTML-Dokument, welches Eigenschaften zum Konzept *Biowetter* enthält.

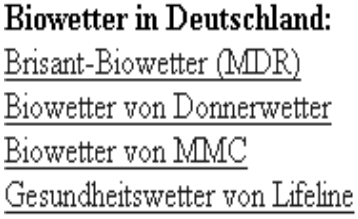


Abbildung 4.13: Darstellung der Ausprägung „Gesundheitswetter“ (Quelle: lifeline.de)

Es ist also erforderlich alle Ausprägungen der Konzepte der Ontologie zu identifizieren und in einem Lexikon zu sammeln. Die Tabelle gibt eine kleine Menge der Ausprägungen wieder, die zu den jeweiligen Konzepten auf den analysierten Web-Seiten und Domains verwendet werden:

Konzept	Ausprägungen
Wetterbericht	Deutschlandwetter, Prognosen, ...
Biowetter	Gesundheitswetter, Medizinwetter, ...
Bauernregeln	Tagesregeln
100-jähriger Kalender	-
Zeitpunkt	heute, morgen, 15.02.2002, 15.02., ...
Zeitintervall	15.02. - 28.02., ...
Land	Deutschland
Bundesland	Niedersachsen, Bremen, ...
Landkreis	Landkreis Demmin, ...
Ort	PLZ, Ortschaft, Stadt, ...

4.2 persistentes Speichern der Wetter-Ontologie

Im letzten Kapitel wurde erklärt, was die Ergebnisse einer webbezogenen Inhaltsanalyse der Domäne waren. Es wurde herausgestellt, wie die Autoren die Wissensstrukturen bezüglich der Wetterinhalte auf ihren Web-Seiten publiziert haben. Das Ergebnis waren:

1. Konzepte, wie z.B. *Biowetter*, die in einer Hierarchie angeordnet werden können,

2. Ausprägungen der Konzepte, die helfen, das Wissen (Eigenschaften) der Konzepte zu identifizieren. Ein Autor verwendet beispielsweise die Zeichenkette „Biowetter“ als Linkbeschreibung auf seiner Web-Seite, um kenntlich zu machen, daß durch Verfolgen des Links auf eine neue Web-Seite zugegriffen wird, die das Wissen des Konzeptes *Biowetter* darstellen. Ein anderer Autor hingegen greift auf die Linkbeschreibung „Gesundheitswetter“ zurück, um das gleiche auszudrücken.

3. Eigenschaften der Konzepte, beispielsweise die *Temperatur* des Konzeptes *Wetterbericht*, die das eigentliche Wissen der Wetterdomäne darstellen,

4. Beziehungen zwischen den Konzepten, die komplexere Eigenschaften der Unterkonzepte des Konzeptes *meteorologischer Aspekt* ausdrücken sollen. Beispielsweise besitzt das Konzept *Biowetter* die Eigenschaft Zeit, welche als ein Zeitpunkt (*aktuell*) oder ein Zeitintervall (*3-Tage-Aussichten*) strukturiert werden kann.

Thema in diesem Kapitel ist die persistente Speicherung dieser Informationen in eine Datenbank.
Untenstehende Tabelle verdeutlicht, wie die Hierarchie der Konzepte relational dargestellt werden kann.

Konzept	Oberkonzept
Wurzel	NULL
meteorologische Aspekte	Wurzel
zeitliche Aspekte	Wurzel
geographische Aspekte	Wurzel
Wetterbericht	meteorologische Aspekte
Biowetter	meteorologische Aspekte
Bauernregeln	meteorologische Aspekte
100-jähriger Kalender	meteorologische Aspekte
Zeitpunkt	zeitliche Aspekte
Zeitintervall	zeitliche Aspekte
Land	geographische Aspekte
Bundesland	Land
Landkreis	Bundesland
Ort	Landkreis

Die obere Relation enthält zwei Attribute, das Konzept und das Oberkonzept. Es wird das Konzept und sein unmittelbares Vaterkonzept jeweils als Tupel abgespeichert, z.B. *Wetterbericht* & *meteorologische Aspekte*.
Die nun folgende Tabelle zeigt die Relation, welche die Ausprägungen jedes Konzeptes speichert. Sie besteht wiederum aus zwei Attributen.

Konzept	Ausprägung
Wetterbericht	Deutschland
Wetterbericht	Prognosen
...	...
Biowetter	Gesundheitswetter
...	...

Die eben dargestellten Relationen enthalten Attributwerte, die besonders in der Phase der „Initialisierung des Suchdienstes“ benutzt werden. In Kapitel 4.4.1 wird ein graphenbasierter Algorithmus vorgestellt, der es ermöglicht, die relevanten Web-Seiten (Expertenseiten) bezüglich der Wetterdomäne zu identifizieren. Dabei werden die Expertenseiten je Konzept, nachfolgend Expertenkonzept genannt, festgestellt und in der Datenbank gespeichert.
Thema des Kapitels 4.4.3 ist die Extraktion von Web-Inhalten aus den markierten Expertenseiten mit Hilfe des Expertenkonzeptes + Ausprägungen und den Konzepten *zeitlicher Aspekt* + Ausprägungen, *geographischer Aspekt* + Ausprägungen. Die möglichen Formate des Web-Inhaltes werden genauer in Kapitel 4.4.3 spezifiziert. Die Web-Inhalte stellen die Eigenschaften des Expertenkonzeptes dar, z.B. einen tabellarischen Überblick über die Daten des Wetterberichtes, wie Temperatur, Bewölkung, Niederschlagsmenge. Zusätzlich spielen beim Extraktionsschritt auch noch die Konzepte *zeitlicher Aspekt* + Ausprägungen, *geographischer Aspekt* + Ausprägungen eine Rolle. Beispielsweise ist es möglich, beim Identifizieren des Web-Inhaltes bezüglich eines Expertenkonzeptes nach örtlichen und zeitlichen Informationen zu suchen, z.B. „aktuelle Biowetter für Rostock“ oder „das aktuelle Biowetter für Deutschland“ . Aus diesem Grund haben die folgenden Tabellen, die Relationen zum Abspeichern der Web-Inhalte darstellen, folgende Struktur:

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Expertenseite
Wetterbericht	Ort	Zeit	Web-Inhalt	Experte
Biowetter	Ort	Zeit	Web-Inhalt	Experte

und

Expertenkonzept	Konzept2	Web-Inhalt	Expertenseite
Bauernregel	Zeit	Web-Inhalt	Experte
100-jähriger Kalender	Zeit	Web-Inhalt	Experte

Die *Bauernregeln* und *100-jähriger Kalender* werden getrennt vom *Wetterbericht* und *Biowetter* gespeichert, da über *Bauernregeln* und *100-jähriger Kalender* keine örtlichen Informationen vorliegen.

4.3 Systemarchitektur

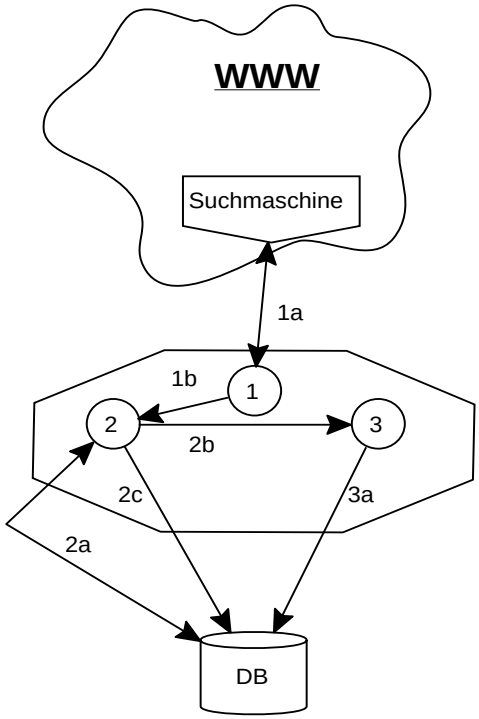
Nachdem in den letzten beiden Kapiteln der eingegrenzte Anwendungsbereich des Suchdienstes in Form einer Ontologie modelliert wurde und das Vorgehen bei der persistenten Speicherung der Ontologie erläutert wurde, wird nun die Systemarchitektur der Suchmaschine erklärt.
Bevor der Anwender das System verwenden kann müssen die relevantesten Wetter-Informationen im Internet markiert und gespeichert werden. Folgende zwei grundlegende Prozesse können vom System gesteuert werden:

1. **Identifizieren der relevanten Wetterinformationen und deren Speicherung:**
   Dieser Prozeß kann auch als Initialisierung des Suchdienstes verstanden werden. Es wird eine Applikation gestartet. Ziel dieser Applikation ist es:

   (a) pro Unterkonzept des Konzeptes *meteorologische Aspekte* sogenannte Expertenseiten im Internet zu finden. Expertenseiten sind HTML-Dokumente, auf denen Werte der Eigenschaften eines Konzeptes, beispielsweise dem *Biowetter*, *sehr gut* dargestellt werden. Unter dem Begriff *sehr gut* versteht man einen hohen Rankingwert, der von einem graphenbasierten Algorithmus (siehe Kapitel 4.4.1) berechnet wird. Startmenge des Algorithmus ist eine Anzahl von HTML-Dokumenten, die die Antwortmenge einer herkömmlichen Suchmaschine, wie z.B. Google, bezüglich eines Suchterms sind. Dieser Suchterm ist ein bool'escher Ausdruck, der sich aus dem Konzept, für das die Expertenseiten identifiziert werden sollen, den entsprechenden Ausprägungen des Konzeptes und dem String *Deutschland*, der die Wetterdaten auf Deutschland begrenzt, zusammensetzt. Ein Suchterm für das Konzept *Biowetter* hat die Form: *Deutschland AND (Biowetter OR Gesundheitswetter OR Medizinwetter)*. Die Antwortmenge setzt sich aus den 200 besten Treffern, die die Suchmaschine zurückgeliefert hat, zusammen. Durch eine geeignete Erweiterung dieser Menge und anschließender Analyse der Linkstruktur der Dokumente erhält man die Expertenseiten (Kapitel 4.4.1).
   Analog werden Expertenseiten für die Konzepte *Wetterbericht*, *Bauernregeln* und *100-jähriger Kalender* identifiziert.

   (b) die Web-Inhalte auf den Expertenseiten zu extrahieren. Dabei wird ein Algorithmus verwendet, der Ergebnis der lokalen Strukturanalyse in Kapitel 4.4.2 ist. Es werden auf den Expertenseiten nach HTML-Strukturen gesucht, die häufig vorkommen und wetterspezifische Daten, bezüglich des jeweiligen Konzeptes der Expertenseite, beinhalten. Beispielsweise wird die Formularstruktur auf den Expertenseiten des Konzeptes *Biowetter* verwendet, um Biowetterdaten eines spezifizierten Ortes zu bekommen. Ausführliche Betrachtungen werden in Kapitel 4.4.3 gegeben.

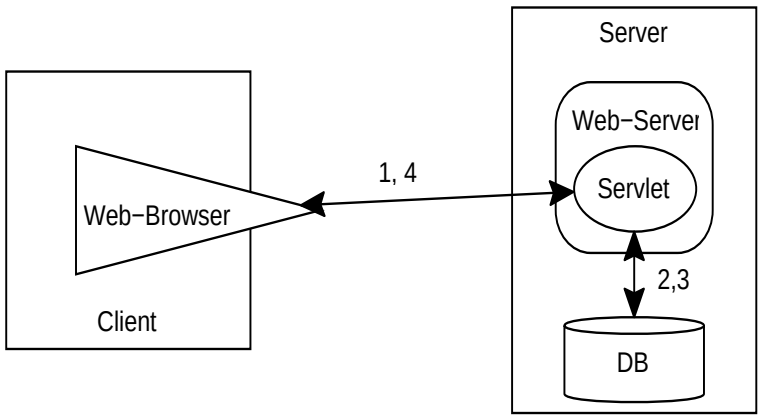
   Zusammenfassend soll die Abbildung 4.14 den Ablauf der Initialisierung des Suchdienstes verdeutlichen.



1a – Spezifizieren der Startmenge (200 URL's)
1b – Startmenge dem Algorithmus übergeben
2a – Holen der Konzepte und entsprechenden Ausprägungen
2b – Expertenseiten werden dem Algorithmus der lokalen Strukturanalyse übergeben
2c – Abspeichern der Expertenseiten
3a – nach der Extraktion Webinhalte werden diese gespeichert

Abbildung 4.14: Initialisierung des Suchdienstes

2. **Anfagebearbeitung und Ergebnispräsentation:**
   Nach der Initialisierung des Systems kann es angewendet werden. Der Anwender sitzt an seinem Rechner, nachfolgend Client bezeichnet, und hat einen Web-Browser geöffnet. Eine Suchmaske wird von einem anderen Rechner, auch als Server bezeichnet, angefordert. Auf dem Server läuft ein Web-Server mit einer integrierten Servlet-Engine. Die Suchmaske enthält Formularelemente, die zur Spezifikation einer Suchanfrage dienen. Aufbau und Benutzen der Maske ist Thema des Kapitels 4.5.1. Die Anfrage wird in Formulardaten verpackt und von einem Servlet auf dem Web-Server verarbeitet. Die angefragten Daten werden aus einer Datenbank extrahiert und in ein verlinktes Dokument integriert, welches als Ergebnis dem Anwender in dem Web-Browser auf dem Client angezeigt wird. Die Abbildung 4.15 stellt den Ablauf der Anfragebearbeitung und Ergebnispräsentation noch einmal graphisch dar.



1 − Suchterm an das Servlet senden
2 − die entspechenden Daten in DB anfragen
3 − Daten aus DB in ein verlinktes Dokument integrieren
4 − verlinktes Dokument an den Client zurücksenden

Abbildung 4.15: Anwenden des Suchdienstes

In den nächsten beiden Kapiteln, 4.4 und 4.5, wird die Initialisierung und das Anwenden des Suchdienstes ausführlich erläutert.

4.4 Initialisierung des Suchdienstes

Das Wissen der Domäne „Wetter in Deutschland“ befindet sich auf bestimmten Web-Seiten im Internet. In diesem Kapitel wird unter anderem diskutiert, mit welchen Mechanismen man diese Dokumente (Expertenseiten) im World-Wide Web lokalisieren kann. Das vorgestellte Verfahren fällt unter die „globale Strukturanalyse“ (siehe 4.4.1). *Global* deshalb, weil die Linkstruktur der WWW-Seiten untersucht wird, und deshalb bei der Analyse ein Graph betrachtet wird, der viele Web-Seiten umfaßt.
In Kapitel 4.4.2 wird der Aufbau jeder einzelnen Expertenseite untersucht. Das Ergebnis dieser lokalen Strukturanalyse sollen HTML-Strukturen sein, die sehr oft auf diesen WWW-Seiten markierbar sind, deren Inhalte Wissensstrukturen der Domäne „Wetter in Deutschland“ darstellen.
Thema des Kapitels 4.4.3 ist das Extrahieren der Web-Inhalte von den Expertenseiten.

4.4.1 globale Strukturanalyse der Domäne

Im Kapitel 4.1, die Inhaltsanalyse, wurde mehrfach der Begriff „geeignete Seite“ gebraucht, um auszudrücken, daß die Inhalte auf dieser Seite Wissen der Domäne „Wetter in Deutschland“ darstellen. Wie kann man diese Seiten im

World-Wide Web markieren und identifizieren? Diese Frage wird in diesem Kapitel beantwortet.
Wie im Kapitel 2 schon erläutert, unterscheiden sich Internetsuchdienste, wie z.B. Google von klassischen IR-Systemen in der Aufbereitung der Suchergebnisse. Bei der Bewertung von HTML-Dokumenten bezüglich eines Suchterms werden zu den herkömmlichen Rankingstrategien der IR-Systeme noch weitere umgesetzt. Relevant für die globale Strukturanalyse ist die *Link Popularity*. Sie gibt an, wieviele Links von anderen Dokumenten auf das betreffende Dokument zeigen. Mit dieser Strategie wird die Wichtigkeit der WWW-Seite im Internet festgelegt. Zum Beispiel ist die Yahoo! Webseite intuitiv wichtiger als die Webseite der Stadt Rostock, weil mehr Seiten auf Yahoo! als auf die Homepage der Stadt zeigen. Der Rankingwert von einer Seite A kann als die Anzahl der Seiten, die auf A verweisen, definiert werden und kann benutzt werden um die Ergebnisse einer Suche zu gewichten. Diese Grundidee funktioniert aber nicht so gut, besonders gegen *Spamming*. Spamming wurde im zweiten Kapitel in Bezug auf das Ranking in IR-Systemen eingeführt. Hier bedeutet es folgendes: Es wird eine Menge von Pseudoseiten erstellt, die auf eine gewünschte Seite verweisen und um so die Wichtigkeit dieser Seite künstlich zu erhöhen.
Eine Suchmaschine, bei der die Linkpopularität besonders stark in das Ranking der Treffer eingeht, ist Google. Der Algorithmus, der in Google die oben genannte Grundidee aufgreift und erweitert heißt *PageRank* [BrPa98]. Die Erweiterung erfolgt indem auch die „Wichtigkeit“ der Webseiten, die zu einer gegebenen Seite verweisen, berücksichtigt wird. Eine Webseite ist demzufolge viel wichtiger, wenn z.B. Yahoo! auf sie verweist. Die ausführliche Beschreibung des *PageRank*-Algorithmus ist in Kapitel 2 zu finden.

Der HITS-Algorithmus Es wurden in der Vergangenheit ähnliche Algorithmen entwickelt, die die Linkstrukturen der WWW-Seiten ausnutzen, um relevante Informationen im World-Wide Web lokalisieren zu können. Ein Vertreter ist z.B. das HITS-Verfahren von Kleinberg [Klei99]. Analog dem PageRank-Algorithmus basiert HITS auf folgende Annahmen:

1. Die Autoren von Webseiten machen implizit eine Aussage über ihre (subjektive) hohe Meinung von gewissen anderen Webseiten, auf die sie mit Links verweisen.

2. Die Gesamtheit der subjektiven Kritiken von Web-Autoren kann man als objektive Bewertung einer Seite auffassen.

3. Je mehr Links auf eine bestimmte Seite verweisen, desto „wichtiger“ scheint diese Seite zu sein.

4. Je weniger Links eine Seite enthält, desto „wichtiger“ ist jeder einzelne Link.

5. Je „wichtiger“ eine Seite ist, desto bedeutender sind die auf ihr enthaltenen Links (z. B. Seiten, die von Yahoo! referenziert werden).

6. Je „wichtiger“ die Links sind, die auf eine bestimmte Seite zeigen, desto „wichtiger“ scheint diese Seite zu sein.

Im Unterschied zu der *PageRank*-Technik, die jeder Seite einen globalen Rankingwert zuweist, ist der HITS-Algorithmus eine anfrageabhängige Rankingstrategie. Das Ergebnis sind zwei Seitentypen, die sogenannten „Hubs“ und „Authorities“ . Eine Seite ist ein „Hub“ für eine Anfrage Q[12], falls sie viele Links auf Seiten enthält, welche für Q relevant sind. Eine Seite ist ein „Authority“ für eine Anfrage Q, falls sie für Q relevant ist, d.h., entsprechende Informationen zur Verfügung stellt. Typischerweise kann man „Hubs“ und „Authorities“ aufgrund der Linkstruktur erkennen, wie die folgende Abbildung zeigt.

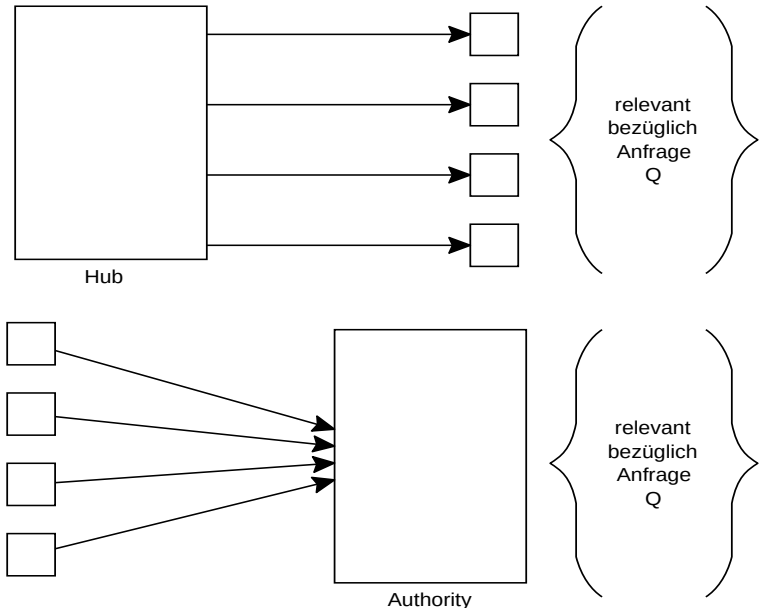


Abbildung 4.16: typischer „Hub“ und „Authority“

Ferner gilt auch:

1. Ein guter „Hub“ zeigt auf gute „Authorities“.

2. Eine gute „Authority“ wird von guten „Hubs“ referenziert.

Aufgrund von diesen Beziehungen zwischen „Hubs“ und „Authorities“ können auch relevante Dokumente lokalisiert werden, welche die Anfrageterme nicht enthalten. Beispielsweise führt eine Anfrage, wie „Autohersteller“ kaum auf

[12]Suchterm, der der Suchmaschine übergeben wird

die Web-Seiten von Honda, VW oder Ford. Mit der Analyse der Beziehung können solche Anfragen sinnvoll beantwortet werden.
Bevor die Schritte des Algorithmus erläutert werden können, muß vorab noch eine Definition erfolgen. Das WWW kann nämlich als gerichteter Graph interpretiert werden. Der Algorithmus erzeugt unter anderem einen Sub-Graphen, der die Teilmenge von Web-Seiten von der gesamten Menge an Seiten umfaßt.

Definition: Gegeben ist eine Menge von Web-Seiten. V, E ist die Menge der gerichteten Kanten (Links) zwischen diesen Seiten. Das Paar (V, E) formt einen ungewichteten Di-Graphen. Für Seiten $p, q \in V$ kennzeichnet man einen Link von p *nach* q *mit* $p \rightarrow q$.

Die Abarbeitung des Verfahrens erfolgt in vier Schritten:

1. Für eine Anfrage (Suchterm) Q werden die ersten t, beispielsweise 200, Dokumente mit einer Suchmaschine (Google oder AltaVista) bestimmt. Diese Menge von Dokumenten entspricht dem „root set“. Für dieses erste Resultat gilt im Allgemeinen:

   (a) im „root set“ sind viele relevante WWW-Seiten enthalten

   (b) im „root set“ sind nicht alle guten „Hubs“ und „Authorities“ enthalten

2. Das „root set“ wird um Dokumente erweitert, welche von den Seiten im „root set“ referenziert werden. Weiterhin wird das „root set“ um Dokumente erweitert, die die Seiten im „root set“ referenzieren. Die so erhaltene Menge wird „base set“ genannt. Folgende Abbildung zeigt den Erweiterungsschritt.

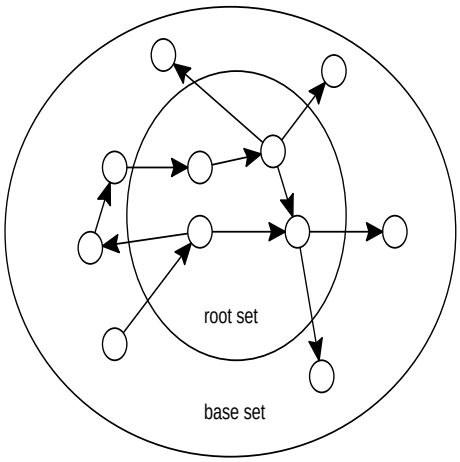


Abbildung 4.17: Erweiterung der „root set“ zur „base set“

Damit diese Basismenge nicht zu viele Dokumente enthält, werden pro Dokument höchstens d, beispielsweise 50, Dokumente hinzugefügt, welche auf dieses zeigen. Links innerhalb derselben Domäne werden entfernt, da diese häufig nur als Navigationshilfen dienen.

3. In diesem Schritt werden die „Hub“ $h(p)-$ und „Authority“ $a(p)-$ Gewichte für ein Dokument p berechnet. Dabei spielen die Anzahl eingehender und ausgehender Links eine zentrale Rolle. Außerdem wird die Idee: „ein guter Hub zeigt auf gute Authorities und eine gute Authority wird von guten Hubs referenziert. Dies führt zu einer rekursiven Definition von $a(p)$ und $h(p)$:

   (a) $a(p)$ und $h(p)$ seien stets normalisiert, d.h.,

   $$\sum_{p \in V} a(p)^2 = 1 \text{ und } \sum_{p \in V} h(p)^2 = 1$$

   (b) **Initialisierung:** Alle Dokumente haben die gleichen Werte $a(p)$ und $h(p)$.

   (c) **Iteration:** Die neuen Gewichte werden aus den alten wie folgt berechnet:

   $$a(p) = \sum_{(q,p) \in E} h(q) \text{ und } h(p) = \sum_{(p,q) \in E} a(q)$$

   (d) Wiederhole die Iteration bis zur Konvergenz. Kleinberg hat berechnet [Klei99], daß nur wenige Schritte, ungefähr 10 bis 20, notwendig sind, um die Konvergenz zu erreichen.

4. Im letzten und vierten Schritt wird das Resultat berechnet. Falls Übersichtsseiten, auch Fanseiten genannt, gewünscht werden, werden die k besten „Hubs“ zurückgeliefert. Das sind gerade die Dokumente mit den höchsten $h(p)$-Werten. Wenn stattdessen Inhaltsseiten, auch Expertenseiten, interessanter sind, werden die k besten „Authorities“ als Ergebnis präsentiert, für die die höchsten $a(p)$-Werte berechnet wurden.

In den letzten Jahren hat dieser Algorithmus viele Erweiterungen erfahren [BhHe98] und [CDG+98], weil das oben beschriebene Verfahren drei Probleme kennt:

1. Falls alle Seiten einer Domäne eine einzelne, externe Seite referenzieren, so wird diese Seite zu stark als „Authority“ gewichtet. Genauso umgekehrt: Falls eine Seite zu viele Seiten derselben Domäne referenziert, so wird diese Seite zu stark als „Hub“ betrachtet.

2. Automatisch erzeugte Links, wie beispielsweise Werbebanner, führen zu falschen „Authorities“

3. Anfragen, wie beispielsweise „Auto Opel“ führen dazu, daß generelle Seiten über Autos und Linkseiten über verschiedene Marken im Resultat dominieren. Der Term „Auto“ dominiert den Term „Opel“ .

Im folgenden werden nun die Verbesserungen, also das Lösen der drei oben beschriebenen Probleme mittels Vorschlägen von [BhHe98] und [CDG+98], erläutert.

1. Problem: Der gleiche Autor einer WWW-Seite kann nur eine „Stimme“ für eine externe Seite abgeben. Analog das Gegenteil: Ein Dokument kann insgesamt nur eine „Stimme“ für die Seiten einer Domäne abgeben.

   (a) Falls k Seiten p_i einer Domäne ein Dokument q referenzieren, so wird das Gewicht $aw(p_i, q) = \frac{1}{k}$ für jeden Link (p_i, q) gesetzt.

   (b) Falls es von einer Seite p l Links zu Seiten q_i einer anderen Domäne gibt, so wird das Gewicht $hw(p, q_i) = \frac{1}{l}$ für jeden Link (p, q_i) gesetzt.

   (c) Damit wird der Iterationsschritt wie folgt geändert:

   $$a(p) = \sum_{(q,p)\in E} aw(q,p) \cdot h(q) \text{ und } h(p) = \sum_{(p,q)\in E} hw(p,q) \cdot a(q)$$

2. Problem und 3. Problem: Lösung nach [BhHe98]
   Zur Lösung dieser Probleme werden Knoten aus dem Graphen entfernt, welche nichts oder nur wenig mit der Anfrage zu tun haben. Zu diesem Zweck wird eine „künstliche“ Anfrage aus den Dokumenten im „root set“ geformt und die Ähnlichkeit der Dokumente zu dieser Anfrage bestimmt:

   (a) Die Anfrage setzt sich aus den ersten Wörtern, beispielsweise 1000, aller Dokumente im „root set“ zusammen

   (b) Anfrage und Dokument werden mit der $tf - idf$ Gewichtung in Vektoren transformiert .

   (c) Die Ähnlichkeit zwischen Dokument und Anfrage, $s(p)$, wird mit dem Kosinusmaß bestimmt.

   (d) Für einen gegebenen Grenzwert t werden alle Knoten (Dokumente) aus dem Graph entfernt, für welche $s(p) < t$ gilt. Auf die Bestimmung des Grenzwertes wird hier nicht näher eingegangen (siehe [BhHe98])

   Dieser „pruning“ Schritt erfolgt zwischen Schritt 2 und 3 im HITS-Algorithmus. Außerdem können die $s(p)$-Werte bei der Berechnung der Hub- und Authoritygewichte benutzt werden:

$$a(p) = \sum_{(q,p)\in E} aw(q,p) \cdot s(q) \cdot h(q)$$
und
$$h(p) = \sum_{(p,q)\in E} hw(p,q) \cdot s(q) \cdot a(q)$$

3. Problem und 2. Problem: Lösung nach [CDG+98]
   Bei dem ursprünglichen Algorithmus von Kleinberg werden alle Kanten (Links) im Graphen des „base set“ mit 1 bewertet. Die semantischen Informationen, die die Links bieten, werden nicht ausgenutzt. Die Semantik haben [CDG+98] mit berücksichtigt. Die Links, deren *Anchor Text* nicht den Suchterm enthalten, werden geringer gewichtet, als die Links, die diese Bedingung erfüllen. Das gleiche trifft für die Linkumgebung zu. Berechnungen und Experimente haben ergeben, daß die Umgebung 50 Bytes vor und nach dem Link relevant für die Untersuchungen sind. Enthält die Umgebung den Suchterm, dann wird der entsprechende Link höher gewichtet.

[CDG+98] hat noch mehr Erweiterungen vorgenommen, die für die Arbeit nicht relevant waren.
Bei der Umsetzung des Algorithmus wurden die Ideen von Kleinberg und die Lösung des ersten Problems nach [BhHe98] und die Lösung des zweiten und dritten Problems nach [CDG+98] implementiert.

Nachdem im oberen Teil des Kapitels der HITS-Algorithmus erklärt wurde, wird nun erläutert, wie dieser Algorithmus im Prozeß der Initialisierung des Suchdienstes verwendet wird, um relevante Web-Seiten im Internet zu identifizieren, die Wissen der Domäne „Wetter in Deutschland“ darstellen.
Für die Konzepte *Biowetter, Wetterbericht Bauernregeln* und *100-jähriger Kalender* wird jeweils die Linkanalyse durchgeführt. Dies geschieht derart, daß die Konzepte zusammen mit ihren Ausprägungen und dem geographischen Aspekt „Deutschland“ jeweils in einer Suchanfrage formuliert werden:

1. Deutschland AND (Biowetter OR Gesundheitswetter OR Medizinwetter)
2. Deutschland AND (Wetterbericht OR „aktuelle Vorhersage“ OR ...)
3. Deutschland AND (Bauernregeln OR ...)
4. Deutschland AND („100-jähriger Kalender“ OR ...)

Diese Suchterme werden jeweils von einer Suchmaschine, in diesem Fall Google, beantwortet. Danach wird der Algorithmus gestartet. Beispielsweise werden für den Suchterm *Deutschland AND (Biowetter OR Gesundheitswetter OR Medizinwetter)* die ersten 200 URL's, die *root set*, aus der Ergebnismenge extrahiert. Eine URL z.B. zeigt auf eine HTML-Seite, die in der Abbildung 4.18 dargestellt wird. Diese Seite enthält eine Auflistung von Links, die auf Dokumente zeigen, auf denen das Biowetter der Themenschwerpunkt ist.

Abbildung 4.18: ein Ausschnitt einer HTML-Seite aus der „root set“ bzgl. Konzept *Biowetter*

Im zweiten Schritt des Algorithmus wird die *root set* zur *base set* um Dokumente erweitert, die von den Seiten im „root set“ referenziert werden. Außerdem wird das „root set“ um Dokumente erweitert, die die Seiten im „root set“ referenzieren. Die resultierende Menge ist das „base set“ .
Das Dokument, welches ausschnittsmäßig in der Abbildung 4.18 gezeigt wird, wird z.B. von dem Dokument in der Abbildung 4.19 referenziert.

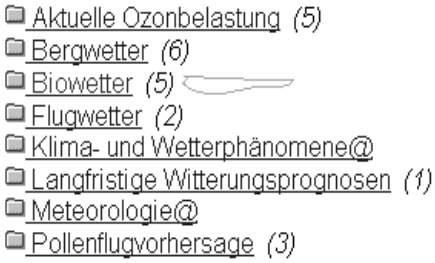


Abbildung 4.19: gekennzeichneter Link verweist auf das Dokument in Abbildung 4.18

Weiterhin besitzt z.B. die HTML-Seite (Abbildung 4.18) einen Link, der das Dokument (Ausschnitt in Abbildung 4.20) referenziert.

Abbildung 4.20: gekennzeichneter Link in Abbildung 4.18 verweist auf dieses Dokument

Die letzten Schritte des Algorithmus dienen dem Berechnen der „Hub“ - und „Authority“ -Gewichte der Web-Seiten in der „base set“. Die besten Expertenseiten bezüglich der einzelnen Konzepte sind unten dargestellt.

- **Biowetter:** http://www.donnerwetter.de/biowetter/
- **Wetterbericht:** http://www.wetter.net/deutschland.html
- **Bauernregeln:** http://www.bauernregeln.de/
- **100-jähriger Kalender:** http://www.altmuehltal.de/hundert.htm

4.4.2 lokale Strukturanalyse der Domäne

Die lokale Stukturanalyse dient dem Entdecken und Identifizieren von HTML-Elementen auf wetterspezifischen WWW-Seiten. Dieser Prozeß wurde manuell vollzogen. Die zu untersuchenden WWW-Seiten sind das Ergebnis der globalen Strukturanalyse, die Expertenseiten, (siehe 4.4.1) gewesen. Vier HTML-Elemente waren das Ergebnis dieser Analyse:

1. **Frames:**
   Mit Hilfe von Frames kann man den Anzeigebereich des Browsers in verschiedene, frei definierbare Segmente aufteilen. Jedes Segment kann eigene Inhalte enthalten. Die einzelnen Anzeigesegmente, also die Frames, können wahlweise einen statischen Inhalt„non scrolling regions“ , oder einen wechselnden Inhalt haben. Verweise in einem Frame können Dateien aufrufen, die dann in einem anderen Frame angezeigt werden. Frames werden in der Wetter-Domäne sehr oft eingesetzt.

2. **eingebettete Frames:**
   Eingebettete Frames erzeugen keine Aufteilung des Bildschirms, wie bei „normalen“ Frames, sondern sind ähnlich wie Grafiken Bereiche innerhalb einer HTML-Datei, in denen fremde Quellen, vor allem andere HTML-Dateien angezeigt werden können.

3. **Tabellen:**
   Man kann mit speziellen syntaktischen HTML-Elementen Tabellen definieren, um tabellarische Daten darzustellen, oder um Text und Grafik attraktiver am Bildschirm zu verteilen. Obwohl Tabellen natürlich vornehmlich zur Darstellung tabellarischer Daten geschaffen wurden, sind sie in der heutigen Praxis des Web-Designs vor allem als Grundgestaltungsmittel für Seitenlayouts nicht mehr wegzudenken. Gerade der letzte Punkt wurde von vielen Designern der Wetterseiten im Internet aufgegriffen und so stellen die Tabellen in der Wetter-Domäne sehr oft eine Alternative zu den Frames dar, um den Anzeigebereich des Browsers zu gestalten. Folgende Abbildung 4.21 zeigt ein Beispiel.

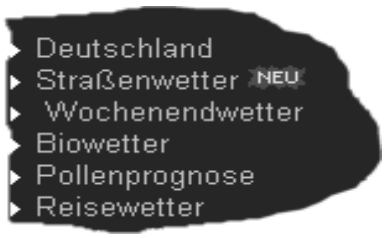


Abbildung 4.21: Ausschnitt von einer HTML-Seite (Links = Navigationsleiste) (Quelle: wetter.net)

4. **Formulare:**
   Mit der Markup-Sprache HTML hat man die Möglichkeit Formulare zu erstellen. In Formularen kann der Anwender Eingabefelder ausfüllen, in mehrzeiligen Textfeldern Text eingeben, aus Listen Einträge auswählen usw. Wenn das Formular fertig ausgefüllt ist, kann der Anwender auf einen Button klicken, um das Formular abzusenden. Formulare können sehr unterschiedliche Aufgaben haben. So werden sie zum Beispiel eingesetzt:

   (a) um bestimmte, gleichartig strukturierte Auskünfte von Anwendern einzuholen,

   (b) um Anwendern das Suchen in Datenbeständen zu ermöglichen,

   (c) um Anwendern die Möglichkeit zu geben, selbst Daten für einen Datenbestand beizusteuern,

   (d) um dem Anwender die Möglichkeit individueller Interaktion zu bieten, etwa um aus einer Produktpalette etwas Bestimmtes zu bestellen.

   In der Domäne „Wetter in Deutschland“ dienen die Formulare auf den entsprechenden Web-Seiten dem zweiten Zweck, dem Suchen in Datenbeständen, z.B. die aktuelle Temperaturangabe für Rostock.

Der Autor des Dokumentes (Abbildung 4.22), welches eine Expertenseite bezüglich des Konzpetes *Biowetter* darstellt, benutzt z.B. Tabellen, um auf der linken Seite eine Navigationsleiste, unter anderem mit dem Link „Biowetter“ anzubieten und um die Inhalte, die von diesen Links auf der Navigationsleiste referenziert werden, darstellen zu können. In der Abbildung 4.22 wird das „heutige Biowetter für Deutschland“ dargestellt.

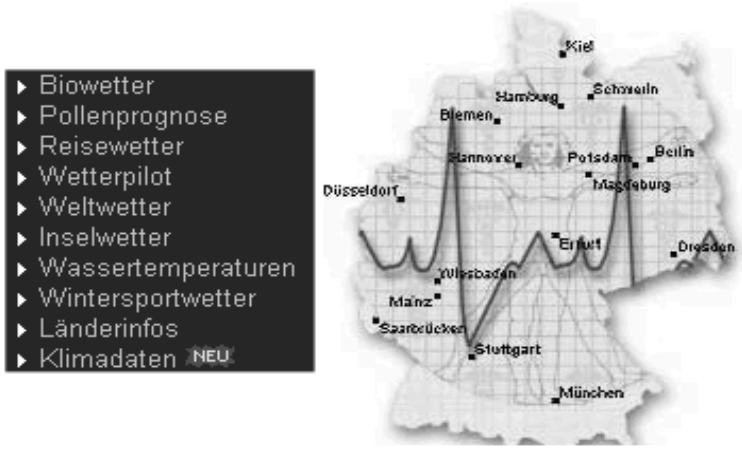


Abbildung 4.22: Ausschnitt von einer Expertenseite (Links = Navigationsleiste) (Quelle: wetter.net)

Das Ergebnis der lokalen Strukturanalyse dient als Hilfsmittel bei der Entwicklung des Algorithmus, des Extrahierens von Wetterdaten. Dieser Algorithmus wird im nächsten Kapitel 4.4.3 ausführlich erklärt.

4.4.3 Web Content Mining-Technik zur Extraktion der Domäneninhalte

Der letzte Prozeßschritt der Initialisierung des Suchdienstes umfaßt das Extrahieren der Web-Inhalte aus den Expertenseiten und das Speichern dieser Inhalte in die Datenbankrelationen, die in Kapitel 4.2 vorgestellt wurde. Das Extrahieren der Web-Inhalte führt ein Algorithmus aus, der in diesem Kapitel erläutert wird. Dabei werden die Ergebnisse der lokalen Strukturanalyse (Kapitel 4.4.2) in den Ablauf einfließen.
Das eigentliche Parsen der HTML-Seiten wird mit dem Wrapper-Toolkit W4F [SaAz01] durchgeführt. In Kapitel 5 wird dieses Werkzeug an einem Beispiel vorgestellt, in dem interne Spezifika des Toolkits, wie z.B. einige Regeln, mit denen man festlegen kann, welche HTML-Inhalte erfaßt werden sollen, vorgestellt werden.
Der generelle Ablauf des Algorithmus wird nun dargestellt. Danach wird zur Illustration eine Expertenseite vorgestellt und erklärt, welche Inhalte der Seite von dem Algorithmus extrahiert werden.
In Abbildung 4.23 ist der Ablauf des Algorithmus dargestellt.

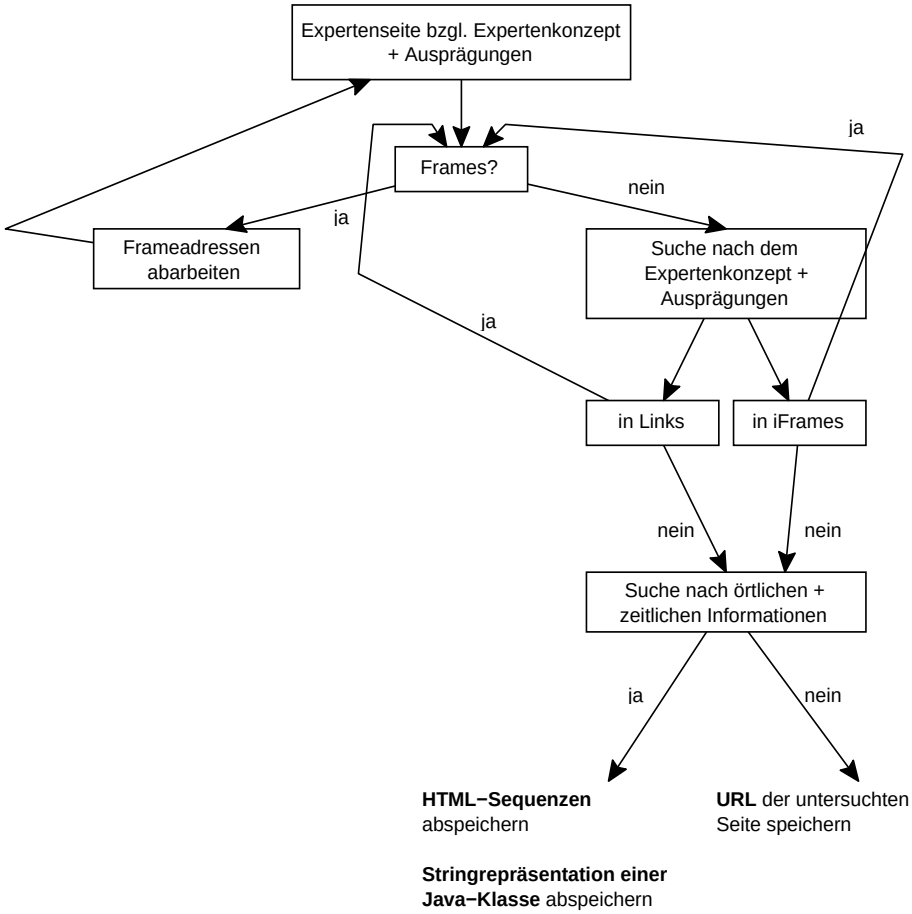


Abbildung 4.23: Algorithmus zur Extraktion der Web-Inhalte aus den Expertenseiten

Ausgehend von den Expertenseiten, bezüglich der Expertenkonzepte *Wetterbericht, Biowetter, Bauernregeln* und *100-jähriger Kalender* wird der Algorithmus je Expertenseite abgearbeitet:

1. Zuerst wird überprüft, ob die Seite Framesets enthält. Wenn ja, werden die Frameadressen aus der Expertenseite extrahiert. Für jede Frameadresse wird der Algorithmus separat gestartet. Wenn keine Frameadressen vorhanden sind, wird sofort der zweite Schritt durchgeführt.

2. Der zweite Schritt umfaßt das Suchen von Zeichenketten in der Linkadresse, Linkbeschreibung und Linkumgebung und in den HTML-iFrame's. Die Zeichenketten beinhalten das Expertenkonzept, wie *Biowetter*, mit den entsprechenden Ausprägungen, wie „Gesundheitswetter“ oder „Medizinwetter“ . Es werden nur Linkadressen und eingebettete HTML-Quellen, meistens in Form eines Verweises auf eine HTML-Seite, betrachtet, die nicht aus der Domain, in der sich die Expertenseite befindet, herausführen. Wenn die Zeichenketten gefunden werden,

wird der entsprechende Link verfolgt, beziehungsweise die eingebettete HTML-Quelle analysiert und rekursiv vorgegangen. Es wird als erstes in den Quellen der neuen HTML-Seiten nach Framesets gesucht usw.
Wenn keine Zeichenketten gefunden werden, wird der rekursive Vorgang abgebrochen und der dritte Schritt des Algorithmus ausgeführt.

3. Im dritten Schritt wird abhängig vom Expertenkonzept nach örtlichen und zeitlichen Informationen gesucht. Liegt eine Expertenseite bezüglich der Konzepte *Biowetter* und *Wetterbericht* vor, werden nach Orts- und Zeitangaben gesucht, im Falle der *Bauernregeln* und dem *100-jährigen Kalender* dagegen nur nach zeitlichen Angaben. Die örtlichen und zeitlichen Daten sind Ausprägungen der Konzepte *zeitliche Aspekte* und *geographische Aspekte*, die im Rahmen der Analyse der Wetterdomäne, erläutert in Kapitel 4.1, erarbeitet wurden.
Bei nicht erfolgreichem Suchvorgang wird die URL des durchsuchten HTML-Textes in die Datenbank als Web-Inhalt gespeichert, wie in Kapitel 4.2 gezeigt. Die Tabelle zeigt diesen Fall:

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Expertenseite
Wetterbericht	Deutschland	aktuell	URL	Experte

Ist hingegen der Suchprozeß erfolgreich, wird versucht, die HTML-Elemente, die die Suchzeichenkette umgeben (Start- und End-HTML-Tag), z.B. ein Formular, Tabellenzeilen oder Tabellenzellen, zu extrahieren und zu analysieren. Dabei kommt den HTML-Formularen besondere Bedeutung zu. Bei den anderen HTML-Elementen wird der Text zwischen dem Start- und End-Tag extrahiert und als Web-Inhalt, in diesem Beispiel eine *HTML-Sequenz*, in die Datenbank abgespeichert. Folgende Tabelle zeigt ein Beispiel.

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Expertenseite
Wetterbericht	Deutschland	aktuell	HTML-Sequenz	Experte

Bei einem Formular werden die Parameterdefinitionen ausgelesen. Mit Hilfe dieser Parameter kann man mit Hilfe des HTML-Wrapper-Toolkits W4F eine Java-Klasse generieren. Der *main*-Methode eines Objektes der Klasse können Argumente übergeben werden, die den Parametern des HTML-Formulars entsprechen. Die Funktionalität eines Objektes der generierten Java-Klasse entspricht dem des HTML-Formulars:

   (a) Übergabe von Formulardaten

(b) Formulardaten werden an den entsprechenden Web-Server gesendet

(c) Der Server generiert automatisch eine HTML-Seite und sendet sie an den Client zurück

Es wird die *Stringrepräsentation der Java-Klasse* gespeichert. Wenn die Klasse beispielsweise Test heißt, wird die Zeichenkette „Test“ in die Datenbank gespeichert. Untere Tabelle zeigt einen Beispieleintrag.

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Expertenseite
Biowetter	Ort	heute	Test	Experte

Der Beispieleintrag in der Spalte „Konzept2“ wurde vorgenommen, weil innerhalb der HTML-Formular-Umgebung Ausprägungen des Konzeptes *geographischer Aspekt* gefunden wurden, die darauf schließen lassen, daß der Parameter einen Ort innerhalb Deutschlands spezifizieren muß. Wird dieser Web-Inhalt vom Anwender des Suchdienstes angefragt, beispielsweise mit dem Suchterm „Biowetter von Rostock“ , dann wird der Ort „Rostock“ als Argument der *main*-Methode eines Objektes der Klasse *Test* übergeben und die dynamisch generierte Web-Seite dem Anwender angezeigt. Eine Eigenschaft der HTML-Dokumente bezüglich der Domäne „Wetter in Deutschland“ ist: Wenn Biowetter- oder Wetterbericht-Daten per Formular angefragt werden, sind die Ergebnisse aktuell. Aus diesem Grund wird in der Spalte *Konzept3* die Zeichenkette „heute“ eingetragen. Die Zeichenkette „heute“ wird im weiteren Verlauf *zeitlicher Default*-Wert genannt. Analoges gilt für die örtlichen Angaben. Dort ist auf den Expertenseiten der Konzepte *Biowetter* und *Wetterbericht* der örtliche Bezug zu „Deutschland“ gegeben, bevor der Extraktionsalgorithmus gestartet wird, um von diesen Seiten detailiertere Informationen zu bekommen. Die Zeichenkette „Deutschland“ stellt somit den *örtlichen Default*-Wert dar.
Mit dem *Refelection*-Konzept der Programmiersprache Java kann man den Objekttyp zur Laufzeit bestimmen. Mit welchen Mitteln das System erkennen kann, daß die Zeichenkette „Rostock“ einen Ort darstellt, ist Thema des Kapitels 4.5.1.

In Kapitel 5 wird ein ausführliches Fallbeispiel des Algorithmus gezeigt.

4.5 Anwenden des Suchdienstes

Während der Initialisierung des Suchdienstes wurden alle notwendigen Prozesse abgearbeitet, die das System anwendbar machen.
In diesem Kapitel wird beschrieben,

1. welche Art von Anfragen an das System gestellt werden können und wie die Anfragen verarbeitet werden (Kapitel 4.5.1),

2. wie die Ergebnisse aggregiert werden und dem Anwender als verlinktes Dokument präsentiert werden (Kapitel 4.5.2).

4.5.1 Anfragebearbeitung

In Kapitel 4.2 wurde die persistente Speicherung der Ontologie behandelt. Außerdem wurde auch festgelegt, wie die Web-Inhalte, die von den Expertenseiten extrahiert werden, relational verwaltet werden. Untere zwei Tabellen stellen die Relationen dar, die die Speicherung der Web-Inhalte repräsentieren.

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Expertenseite
Wetterbericht	Ort	Zeit	Web-Inhalt	Experte
Biowetter	Ort	Zeit	Web-Inhalt	Experte

und

Experten-konzept	Konzept2	Web-Inhalt	Experten-seite
Bauernregel	Zeit	Web-Inhalt	Experte
100-jähriger Kalender	Zeit	Web-Inhalt	Experte

In beiden Relationen sind keine NULL-Werte erlaubt, womit sich dann vier verschiedene Anfragekombinationen ergeben:

- Wetterbericht & Ort & Zeit
- Biowetter & Ort & Zeit
- Bauernregenl & Zeit
- 100-jähriger Kalender & Zeit

Der Anwender kann z.B. Anfragen, der Form „aktuelles Biowetter in Rostock“, angeben.
Durch eine Anfrage können mehrere Web-Inhalte markiert werden, da pro Expertenkonzept 15 Expertenseiten identifiziert wurden. Es wird aber bloß der Web-Inhalt zurückgegeben, deren zugehörige Expertenseite den größten Rankingwert hat. Der Rankingwert der Expertenseite wurde während der Initialisierung des Suchdienstes beim Ablauf des HITS-Algorithmus berechnet. Die restlichen möglichen Web-Inhalte werden mit dem relevantesten Web-Inhalt in einem verlinkten Dokument integriert und an den Anwender gesendet. Wie das geschieht, wird im nächsten Kapitel 5.2 dargestellt.

4.5.2 Ergebnispräsentation

Im letzten Kapitel wurde erklärt, welche Anfragen möglich sind und wie die Anfragen vom System verarbeitet werden.
Thema in diesem Kapitel ist die Art und Weise, wie das Ergebnis dem Anwender präsentiert wird. Der Suchterm, der vom Anwender eingegeben wird, spezifiziert eine Anzahl von Web-Inhalten. Es ist aber nur der Web-Inhalt am relevantesten, dessen Expertenseite, von denen der Web-Inhalt extrahiert wurde, den größten Rankingwert hat. Es kann vorkommen, daß mehrere Web-Inhalte den gleichen entsprechenden Rankingwert besitzen. Diese Web-Inhalte werden dann gleichwertig im Web-Browser des Anwenders angezeigt.
Der Anwender hat die Möglichkeit, bevor er den Suchterm an den Server sendet, anzugeben, wieviele Ergebnisse maximal angezeigt werden sollen. Dementsprechend gestaltet sich auch die Anzeige der Ergebnisse im Web-Browser des Anwenders. Ein Beispiel ist in der unteren Abbildung angegeben.

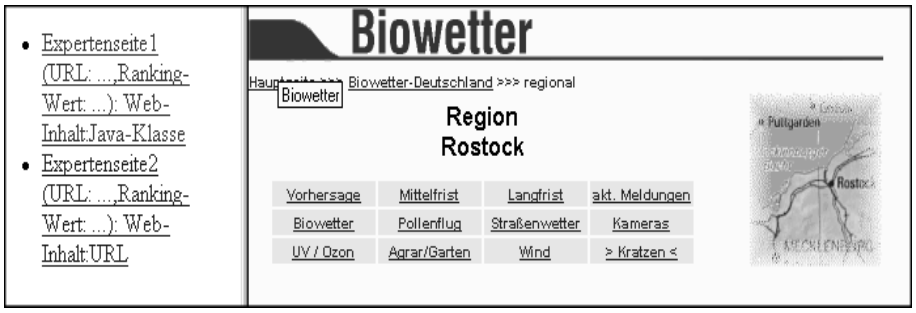


Abbildung 4.24: Ergebnisanzeige des Suchdienstes

Ein linker Frame dient dem Navigieren durch die Ergebnisse, die absteigend nach dem Rankingwert sortiert sind. Diese Ergebnisse werden nicht direkt angezeigt, sondern sind durch einen Link referenziert. Der relevanteste Web-Inhalt wird in dem rechten Frame angezeigt. Wenn mehrere Ergebnisse angezeigt werden, weil die Web-Inhalte den gleichen Rankingwert besitzen, wird der rechte Anzeigebereich des Web-Browsers in mehrere Frames geteilt.
Es kann vorkommen, daß der Anwender Web-Inhalte mit seinem Suchterm anfragen will, die nicht in der Datenbank hinterlegt sind. Das System muß entsprechend reagieren und die Ausgabe dieser Fehlermeldung anzeigen.

5 Implementierungsdetails & Ergebnisse

Im letzten Kapitel wurden die Konzepte zur Realisierung einer domänspezifischen Suchmaschine besprochen.
Ausgangspunkt war eine umfassende webbezogene Inhaltsanalyse der Domäne „Wetter in Deutschland“ , deren Ziel die strukturelle Erfassung des Wissens auf relevanten Web-Dokumenten war. Die Modellierung der Wissensstrukturen erfolgte mit einer Ontologie.
Die relevanten Web-Dokumente, deren Autoren inhaltliche Fakten bezüglich der Domäne publizieren, werden anhand eines graphenbasierten Algorithmus identifiziert. Diese sogenannten Expertenseiten werden in einer Datenbank gespeichert. Im Rahmen der Diplomarbeit wurde das Datenbankmanagmentsystem (DBMS) MySQL verwendet. Es ist aber kein Problem, die Daten auf ein anderes DBMS, wie DB2 oder Oracle, zu portieren.
Die Expertenseiten werden einzeln mit einem Algorithmus analysiert, um inhaltliches Wissen zu extrahieren. Dieser Prozeß wird mit dem W4F-Toolkit [SaAz01] vorgenommen. Die Algorithmenschritte wurden ausführlich in Kapitel 4.4.3 erläutert und sollen hier exemplarisch durchgeführt werden.

Beispiel 1: Extraktion von Web-Inhalten Abbildung 5.1 zeigt einen Ausschnitt einer Expertenseite bezüglich des Expertenkonzeptes *Biowetter*.

Abbildung 5.1: Biowetter-Experte

Die nächsten Schritte demonstrieren den Ablauf des Algorithmus.

1. Diese Seite enthält ein Frameset. Aus diesem Grund werden einzeln die Frameadressen, *http://www.donnerwetter.de/navi.mv*, *http://www.donnerwetter.de/aktuell.mv* und *http://www.donnerwetter.de/themen.mv* analysiert.

2. **http://www.donnerwetter.de/navi.mv:**

   (a) Es werden in Verweisen (Linkadressen, Linkbeschreibungen und Linkungebungen) und in iFrame-HTML-Umgebungen nach dem Expertenkonzept und den Ausprägungen gescannt. Es wird also nach den Zeichenketten „Biowetter“ , „Gesundheitswetter“ und „Medizinwetter“ gesucht. Das Ergebnis ist der Link *http://www.donnerwetter.de/biowetter/menu.hts*

   (b) Die HTML-Seite, auf die die geparste Linkadresse *http://www.donnerwetter.de/biowetter/menu.hts* zeigt, wird

analysiert. Der Inhalt dieses Dokumentes wird wieder nach den Zeichenketten durchsucht. Es wird kein Link-HTML- und iFrame-HTML-Tag gefunden, der diese Zeichenketten einschließt. Aus diesem Grund greift der dritte Schritt des Algorithmus.

(c) In diesem letzten Schritt wird nun nach örtlichen und zeitlichen Informationen gescannt. Das sind Ausprägungen der Konzepte *zeitlicher Aspekt* und *geographischer Aspekt*, beispielsweise der aktuelle Wochentag und der aktuelle Monat, beide als Zeichenkette. Örtliche Informationen können z.B. „Region“ , „Ort“ oder „PLZ“ sein. Im Kapitel 4.4.3 wurde anhand der Eigenschaften und dem Aufbau der Expertenseiten bezüglich der Expertenkonzepte *Wetterbericht* und *Biowetter* festgelegt, daß es Defaultwerte für örtliche und zeitliche Informationen gibt: „Deutschland“ und „heute“ . Wenn die Expertenseiten keine weiteren örtlichen und zeitlichen Daten präsentieren, werden in der Datenbank neben dem entsprechenden Web-Inhalt „Deutschland“ und „heute“ gespeichert.
In diesem Beispiel tritt dieser Fall bei den zeitlichen Daten nicht auf. Die Terme „Freitag“ , „Februar“ und „2002“ können als zeitliche Informationen im HTML-Code

```
<B>Freitag, 22. Februar 2002</B>
```

gefunden werden. Es ist nicht möglich, ausgehend von diesem HTML-Code, eine HTML-Sequenz aus dem Dokument zu extrahieren, die die Biowetterdaten zu dem zeitlichen Bezug darstellt. Aus diesem Grund wird die URL dieser HTML-Seite als Web-Inhalt in die Datenbank gespeichert. Im zweiten Beispiel unten wird die Extraktion einer HTML-Seite vorgestellt, bei der HTML-Sequenzen identifiziert werden können und extrahiert werden. Die extrahierte zeitliche Information „Freitag“ stellt den aktuellen Tag dar und wird als zeitlicher Wert „heute“ in der Datenbank gespeichert.
Örtliche Informationen können ebenfalls lokalisiert werden. Diese Informationen sind Teil eines HTML-Form-Elements.

```
<FORM action=/region/ortrubrik.mv method=get>
```

Dieses Formular wurde vom Autor der HTML-Seite verwendet, um dem Anwender die Möglichkeit zu geben, dynamisch Biowetterdaten von Orten Deutschlands zu bekommen. Aus diesem Grund können zwei Web-Inhalte aus dieser HTML-Seite abgespeichert werden. Ein Web-Inhalt ist die oben schon erwähnte URL der analysierten HTML-Seite, die Informationen zum „aktuellen Biowetter in Deutschland“ referenziert. Der örtliche Bezug „Deutschland“ ist Ergebnis der globalen Linkanalyse und gibt einen Defaultwert (siehe Kapitel 4.4.3) an.

Untenstehende Tabelle gibt einen Überblick über das Speichern des ersten Web-Inhaltes.

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Expertenseite
Biowetter	Deutschland	heute	URL	Experte

Der zweite Web-Inhalt ist eine Zeichenkette, die eine Java-Klasse bezeichnet, die verwendet wird, um dynamisch eine HTML-Seite zu generieren, die aktuelle Biowetterdaten eines spezifizierten Ortes beinhaltet. Die Java-Klasse wird mit dem HTML-Wrapper-Toolkit W4F generiert. Es werden die wichtigen HTML-Formulardaten ausgelesen:

```
<FORM action=/region/ortrubrik.mv method=get>
  <INPUT size=10 name=search>
  <INPUT type=submit value=ok name=B1>
</FORM>
```

Das erste INPUT-Element definiert ein Textfeld der Länge 10, in dem der Anwender den Ort eintragen kann. Das zweite INPUT-Element stellt den SUBMIT-Button dar, der beim Drücken einen Prozeß, im *action*-Attribut spezifiziert, startet, der die Zeichenkette im Textfeld an den Server sendet.
W4F bietet eine Spezifikationssprache, die man benutzen kann, um unter anderem *RETRIEVAL_RULES*, *EXTRACTION_RULES* und *JAVACODE* zu spezifizieren. Aus unterem Konstrukt innerhalb der *RETRIEVAL_RULES* wird die Java-Methode *getOrt* erzeugt. Diese Methode gehört zu der Klasse *BiowetterOrt*, die als Stringrepräsentation in der Datenbank als Web-Inhalt abgespeichert wird.

```
EXTRACTION_RULES {
  source = html.src;
}

RETRIEVAL_RULES {
  getOrt(String ort)
    {
     METHOD: GET;
     URL: "http://www.donnerwetter.de/region/ortrubrik.mv";
     PARAM: "search"=ort,
            "B1"="ok";
    }

JAVACODE {
  public static void main(String[] args)
    throws Exception {
    BiowetterOrt erg = BiowetterOrt.getOrt(args[0]);
    //generierte Seite in erg.source
```

```
  }//end of main
}
```

Die Methode *getOrt* hat einen String-Parameter, der den Ort spezifiziert. Der Ausdruck innerhalb der *EXTRACTION_RULES*-Sektion gibt an, daß man den gesamten Quelltext der resultierenden HTML-Seite bekommen möchte, auf der die Biowetterdaten für den Ort publiziert sind. In der *JAVACODE*-Sektion wird die *main*-Methode der Klasse implementiert. In ihr wird ein Objekt der Klasse *BiowetterOrt* erzeugt und dann die Methode *getOrt* mit dem Parameter *args[0]*, welcher den Ort spezifiziert, aufgerufen. In der Datenbank wird die Stringepräsentation der Java-Klasse, der Defaultwert „heute“ bezüglich der zeitlichen Information und der String „Ort“ für die örtliche Information, der signalisiert, daß ein Ort einer Java-Klasse übergeben werden kann, abgespeichert.

Experten-konzept	Konzept2	Konzept3	Web-Inhalt	Exper-tenseite
Biowetter	Ort	heute	„BiowetterOrt“	Experte

3. **http://www.donnerwetter.de/aktuell.mv:** Das ist die gleiche Adresse wie die Adresse des analysierten Links. Die Analyse führt deshalb zu demselben Ergebnis.

4. **http://www.donnerwetter.de/themen.mv:** Von der HTML-Seite, auf die der Link zeigt, können keine domänenspezifische Inhalte extrahiert werden.

Im nächsten Beispiel wird die Extraktion einer HTML-Seite gezeigt, in der HTML-Sequenzen identifiziert und abgespeichert werden können.

Beispiel 2: Extraktion von Web-Inhalten Die HTML-Seite, dargestellt in einem Ausschnitt in Abbildung 5.2, ist eine Expertenseite bezüglich des Konzeptes *100-jähriger Kalender*.

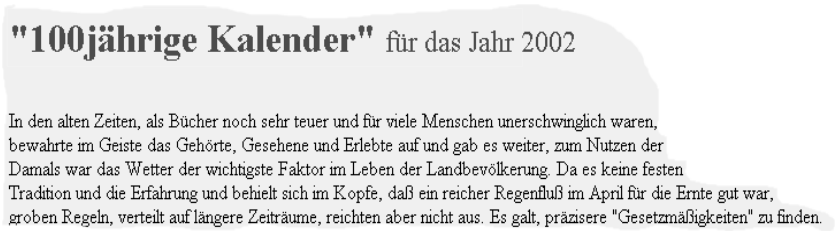


Abbildung 5.2: Kalender-Experte

Analog zu dem oberen Beispiel wird auch hier die Anwendung des Algorithmus auf diese Expertenseite gezeigt.

1. Diese Seite enthält keine Frameset.

2. Ziel des zweiten Schrittes ist die Extraktion von Links und HTML-iFrame-Umgebungen, die aber in diesem Beispiel nicht vorhanden sind.

3. Während des dritten Schrittes wird nach zeitlichen Informationen gesucht. Im Gegensatz zu den Konzepten *Biowetter* und *Wetterbericht* kommt auf den Expertenseiten bezüglich der Konzepte *Bauernregeln* und *100-jähriger Kalender* keine örtlichen Informationen vor (siehe Kapitel 4.4.3).
   Zeitliche Daten, die Zeichenkette „2002“ , können auf dieser Expertenseite lokalisiert werden.

```
...
<B>"100jährige Kalender" </B>
</FONT><FONT color=red size=+2>für das Jahr 2002</B>
...
```

   Ausgehend von dieser Zeichenkette kann keine HTML-Sequenz extrahiert werden, so daß in diesem Fall die URL (= URL der Expertenseite) als Web-Inhalt abgespeichert wird.

Expertenkonzept	Konzept2	Web-Inhalt	Expertenseite
100-jähriger Kalender	2002	URL	Experte

   Außerdem können noch alle zwölf Monate jeweils mit den entsprechenden Daten aus dem 100-jährigen Kalender durch eine Tabellenstruktur dargestellt werden. Im unteren Beispiel werden Ausschnitte aus einer HTML-Tabelle gezeigt, in der Kalenderdaten für den Februar eingebettet sind.

```
<TABLE cellSpacing=0>
  <TBODY>
    <TR>
      <TD></TD>
      <TD>
        <FONT color=green size=+2>
          <B><I>Februar</I>
          </B>
        </FONT>
      </TD>
    </TR>
    <TR>
      <TD align=right>1. - 6. </TD>
      <TD>trüb, Regen, Nebel, Wind</TD>
    </TR>
  <TR>
  ...
</TABLE>
```

Es ist also möglich, zwölf HTML-Sequenzen zu extrahieren. In der Tabelle sind für das Attribut *Web-Inhalt* die Werte *HTML-Tabelle* angegeben. Dieser Eintrag soll andeuten, daß die HTML-Sequenz eine Tabellenstruktur darstellt.

Experten-konzept	Konzept2	Web-Inhalt	Experten-seite
100-jähriger Kalender	Januar 2002	HTML-Tabelle	Experte
...	...	...	...
100-jähriger Kalender	Dezember 2002	HTML-Tabelle	Experte

Beispiel 1: Anfragebearbeitung In diesem Beispiel wird die Suchmaske, vorgestellt und Beispielanfragen angegeben. Dann wird gezeigt, wie die Anfrage beantwortet wird und in einem verlinkten Dokument dem Anwender zurückgesendet wird.

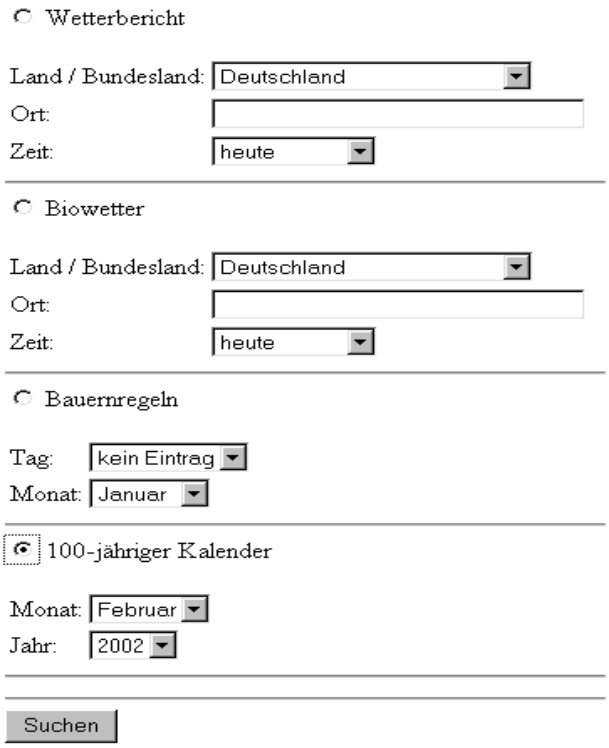


Abbildung 5.3: Suchmaske des Suchdienstes

Die vier Expertenkonzepte, *Biowetter*, *Wetterbericht*, *Bauernregeln* und *100-jähriger Kalender*, werden als Radiobuttons dargestellt und können wahlweise vom Anwender ausgewählt werden. Zusätzlich kann man noch weitere Daten eingeben, die zeitliche und örtliche Informationen beinhalten. Im Beispiel des

100-jährigen Kalenders, in der Abbildung 5.4 selektiert, ist es möglich, das Jahr und den Monat auszuwählen. In der Abbildung 5.4 lautet die Anfrage: „Gib die Daten des 100-jährigen Kalenders für den Februar 2002“ . Diese Anfrage wird an den Server gesendet und von einem Servlet, welches im Web-Server TomCat läuft, beantwortet. Dabei werden in der Datenbank die Web-Inhalte angefragt, die den spezifizierten Suchtermen in der Suchmaske entsprechen, im Beispiel: „100-jähriger Kalender“ „„Februar“ ,und „2002“ . In der Tabelle

Experten-konzeot	Konzept2	Web-Inhalt	Expertenseite
100-jähriger Kalender	Februar 2002	HTML-Tabelle	http://www.altmuehltal.de/hundert.htm

wurde der Web-Inhalt, eine HTML-Tabelle von der Expertenseite *http://www.altmuehltal.de/hundert.htm* abgespeichert. Der Wert des Attributes *Expertenseite* ist ein Fremdschlüssel zu einer Relation, welche die Daten zu dieser Expertenseite enthält. Unter anderem ist dort auch der Ranking-Wert aufgeführt, der während der globalen Strukturanalyse (HITS-Algorithmus) berechnet wurde. Diese Expertenseite hat bezüglich des Konzeptes *100-jähriger Kalender* den größten Ranking-Wert. Aus diesem Grund wird die obige HTML-Tabelle als HTML-Sequenz im Browser des Anwenders angezeigt. Das Ergebnis ist auszugsweise in der unteren Abbildung angegeben. Links ist die Navigationsleiste zu sehen, die ein Browsen durch die anderen Web-Inhalte ermöglicht. Im rechten Fenster wird dann der Web-Inhalt angezeigt.

Abbildung 5.4: Ergebnis der Anfrage „100-jähriger Kalender im Februar 2002“

Die Suchmaschine Google liefert bei der Anfrage: „100-jähriger Kalender Februar 2002“ folgendes Ergebnis:

Abbildung 5.5: Ergebnissausschnitt der Suchmaschine Google

Es wird eine URL-Liste generiert, die dem Anwender im Browser angezeigt wird. Um an die relevanten Informationen zu gelangen, muß der Anwender erst die Links verfolgen und dann auf den entsprechenden Seiten nach diesen Daten zu suchen, sie werden ihm also nicht gleich angezeigt.
Ein anderes Beispiel soll diese unterschiedliche Präsentation der Ergebnisse noch einmal verdeutlichen. Es soll vom Suchdienst die Anfrage: „Biowetter heute Rostock“ beantwortet werden. In der Suchmaske werden folgende Werte eingetragen:

Abbildung 5.6: Suchangaben bezüglich des Biowetters

In der Tabelle, in denen die Biowetter-Inhalte gespeichert sind, werden die Daten mit dem größten Ranking-Wert angefragt.

Expertenkonzept	Konzept2	Konzept3	Web-Inhalt	Exper-tenseite
Biowetter	Ort	heute	„BiowetterOrt“	Experte

Dieser Web-Inhalt, „BiowetterOrt“ stellt die Stringrepräsentation einer Java-Klasse dar (siehe oben). Es wird ein Objekt der Klasse erzeugt, dessen *main*-Methode den Parameter „Rostock“ übergeben bekommt. Das Ergebnis wird dynamisch erzeugt (Abbildung 5.7).

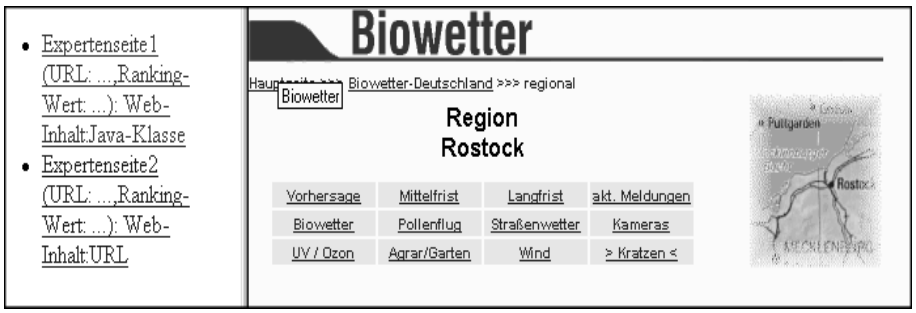


Abbildung 5.7: Ergebnis der Anfrage „heutiges Biowetter für Rostock“

Google hingegen liefert wieder eine URL-Liste als Antwort, ohne aber dynamisch generierte HTML-Seiten dem Anwender anzuzeigen.

Abbildung 5.8: Ergebnissausschnitt der Suchmaschine Google

Im entwickelten Suchdienst kann noch folgender Fall eintreten: Angefragte Web-Inhalte besitzen den gleichen Rankingwert.
Dann werden diese Daten im rechten Frame-Fenster des Browsers dem Anwender präsentiert.

6 Zusammenfassung & Ausblick

Das Ziel dieser Arbeit war es, einen Suchdienst zu entwickeln, der Anfragen bezüglich der Domäne „Wetter in Deutschland“ sehr genau beantworten soll. Es wurde ein Konzept entwickelt und vorgestellt, mit dem der Suchdienst prototypisch umgesetzt wurde.
Ausgangspunkt der konzeptionellen Arbeit war eine webbezogene Inhalts- und Strukturanalyse der Domäne, um die Wissensstrukturen auf relevanten Wetter-Seiten zu identifizieren, diese Strukturen dann mit Hilfe einer Ontologie zu modellieren und die relevantesten HTML-Seiten (Expertenseiten) dieser Domäne im Internet zu lokalisieren. Die inhaltlichen und strukturellen Metadaten werden von einem Web Content Mining-Algorithmus benutzt. Dieses Verfahren ist eine Wrappertechnik, die Inhalte von den Expertenseiten extrahiert und speichert.
Der Anwender kann diese Inhalte anfragen. Entsprechend dem Suchterm werden Daten angefordert, die dem Nutzer integriert, in einem verlinkten Dokument, präsentiert werden.

Der Schritt der Domänenanalyse ist sehr umfangreich. Es ist deshalb nicht ohne große Anpassungen möglich, diesen Suchdienst für einen anderen Anwendungsbereich, z.B. Sportnachrichten, zu konzipieren. Folgende Fragen müssen beantwortet werden:

1. Wie ist das Wissen der Domäne auf den HTML-Seiten strukturiert? (inhaltliche Metadaten)

2. Wie kann dementsprechend die Domäne modelliert werden? (Aufbau einer Ontologie)

3. Welche signifikanten HTML-Elemente werden auf den Web-Seiten der Domäne verwendet, um das Wissen darzustellen? (strukturelle Metadaten)

Die Antworten können in unterschiedlichen Domänen sehr verschieden sein. Vor allem bei den strukturellen Metadaten kann es vorkommen, daß ein HTML-Element in verschiedenen Domänen anders verwendet wird, um das Wissen darzustellen. Deshalb ist es erforderlich, den Web Content Mining-Algorithmus den Eigenschaften jeder einzelnen Domäne anzupassen. Dieser Algorithmus verwertet gerade die Ergebnisse aus der Inhaltsanalyse (inhaltiche Metadaten), der globalen Strukturanalyse (Expertenseiten) und der lokalen Strukturanalyse (strukturelle Metadaten).

Der Schwerpunkt der Arbeit lag auf Techniken des Web Minings. Dieser Forschungsbereich wird von vielen anderen Richtungen der Informatik beeinflußt, wie z.B. Information Extraction und das maschinelle Lernen. So

konnte in der Arbeit nur ein kleiner Ausschnitt des Web Minings konkreter umgesetzt werden. Aufbauend auf dieser Arbeit kann man sicherlich andere Web Mining-Verfahren umsetzen, um eine domänenspezifische Suchmaschine aufzubauen und die Ergebnisse miteinander zu vergleichen.

Literatur

[Abit97] S. Abiteboul: Querying semi-structured data. In F. N. Afrati and P. Kolaitis, editors, Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings, volume 1186 of Lecture Notes in Computer Science, pages 1-18. Springer, 1997.

[AHKV98] H. Ahonen, O. Heinonen, M. Klemettinen, A. Verkamo: Applying data mining techniques for descriptive phrase extraction in digital document collections. In Advances in Digital Libraries (ADL'98), Santa Barbara, California, USA, April 1998, 1998.

[AlTu99] L. Altman, S. Tuomela: WWW Metadata & Collaboration. 1999. http://ils.unc.edu/altml/collaboration/

[AmFu99] B. Amann, I. Fundulaki: Integrating Ontologies and Thesauri to build RDF Schmemas. in: Research and Advanced Technology for Digital Libraries. Proceedings of Third European Conference on Digital Libraries, ECDL-99, Paris, France, September 1999. LNCS 1696.

[AQM+97] S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. L. Wiener: The lorel query language for semistructured data. Int. J. on Digital Libraries, 1(1):68-88, 1997.

[ArMe00] G. O. Arocena, A. O. Mendelzon: Weboql: Restructuring documents, databases, and webs. Theory and Practice of Object Systems, 5(3):127-141, 1999. SIGKDD Explorations. Copyright c ACM SIGKDD, July 2000. Volume 2, Issue 1 - page 10

[AtMe97] P. Atzeni, G. Mecca: Cut & paste. In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona, pages 144-153. ACM Press, 1997.

[BaRi99] R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval. Addison-Wesley Longman Publishing Company, 1999.

[BBA+99] A. Büchner, M. Baumgarten, S. Anand, M. Mulvenna, J. Hughes: Navigation pattern discovery from internet data. In Proceedings of the WEBKDD'99 Workshop on Web Usage Analysis and User Profiling, August 15, 1999, San Diego, CA, USA, 1999.

[Bern97] T. Berners-Lee: Metadata Architecure, 1997, http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html

[Bern98] T. Berners-Lee: Why RDF model is different from XML model, 1998, http://www.w3.org/DesignIssues/RDF-XML.html

[BhHe98] K. Bharat, M. R. Henzinger: Improved algorithms for topic distillation in a hyperlinked environment. In Proceedings of the 21st annual

international ACM SIGIR conference on Research and development in information retrieval August 24 - 28, 1998, pages 104-111, Melbourne Australia, 1998.

[BiPa99] D. Billsus, M. Pazzani: A hybrid user model for news story classification. In Proceedings of the Seventh International Conference on User Modeling (UM '99), Banff, Canada, 1999.

[BoAk97] W. N. Borst, J. M. Akkermanns: Engineering Ontologies. In: International Journal of Human-Computer Studies. 46(2/3), 1997. pp. 365-406

[BoLe99] J. Borges, M. Levene: Data mining of user navigation patterns. In Proceedings of the WEBKDD'99 Workshop on Web Usage Analysis and User Profiling, August 15, 1999, San Diego, CA, USA, pages 31-36, 1999.

[Boye01] C. Boyens: OntoKick Ein Werkzeug zur Unterstützung von ontologiebasiertem Wissensmanagement, Institut für Angewandte Informatik und Formale Beschreibungsverfahren der Universität Karlsruhe (TH), 2001

[BrHL99] Bray, Hollander, Layman: World Wide Web Consortium Recommendation, 1999, http://www.w3.org/TR/1999/REC-xml-names-19990114

[BrPa98] S. Brin, L. Page: The anatomy of a largescale hypertextual Web search engine. Proc. 7th WWW Conf., 1998.

[BuCG99] O. Buyukkokten, J. Cho, H. Gracia-Molina: Exploiting Geographical Location Information of Web Pages, 1999

[Bune97] P. Buneman: Semistructured data. In Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Tucson, Arizona, May 1997. Tutorial

[CaKa97] J. Carri'ere, R. Kazman: WebQuery: Searching and visualizing the Web through connectivity. Proc. 6th WWW Conf., 1997.

[CDF+98] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery. Learning to extract symbolic knowledge from the world wide web. In Proceedings of the Fifteenth National Conference on Articial Intellligence (AAAI98), pages 509-516, 1998.

[CDG+98] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, S. Rajagopalan: Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. Proceedings of the 7th World Wide Web conference, 1998. Elsevier Sciences, Amsterdam.

[CDG+99] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins: Mining the link structure of the world wide web. IEEE Computer, 32(8):60-67, 1999.

[CGH+94] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom: The tsimmis project: Integration of heterogeneous information sources. In Proceedings of the 10th Meeting of the Information Processing Society of Japan, pages 7-18, 1994.

[ChDI98] S. Chakrabarti, B. Dom, P. Indyk: Enhanced hypertext classification using hyperlinks. Proc. ACM SIGMOD, 1998.

[Cohe95] W. W. Cohen: Learning to classify english text with ilp methods. In Advances in Inductive Logic Programming (Ed. L. De Raedt), IOS Press, 1995.

[Cohe99] W. W. Cohen: What can we learn from the web? In Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99), pages 515-521, 1999.

[CoLe96] J. Cowie, W. Lehnert: Information extraction. Communications of the ACM, 39(1):80-91, 1996.

[CoMS97] R. Cooley, B. Mobasher, J. Srivastava: Web mining: Information and pattern discovery on the world wide web. In Proceedings of the 9th IEEE International Conference on Tools with Arti

cial Intelligence (ICTAI'97), 1997.

[Cool00] R. W. Cooley: Web Usage Mining: Discovery and Application of Interesting Patterns from Web data. PhD thesis, Dept. of Computer Science, University of Minnesota, May 2000.

[Dahl95] K. Dahlgren: A linguistic ontology. in: Internationel Journal of Human-Computer Studies. vol.43 no. 5/6, 1995, pp. 809-818

[DDES97] S. Decker, M. Daniel, M. Erdmann, R. Studer: An Enterprise Reference Scheme for Integrating Model Based Knowledge Engineering and Enterprise Modelling. in: Proceedings of the 10th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW'97), Lecture Notes in Artificial Intelligence LNAI 1319, Springer Verlag, 1997.

[DEFS99] S. Decker, M. Erdmann, D. Fensel, R. Studer: Ontobroker - Ontology based Access to Distributed and Semi-structured Information. in: Database Semantics: Semantic Issues in Multimedia Systems. Proceedings TC2/WG 2.6 8th Working Conference on Database Semantics (DS-8), Rotorua, Nw Zealand. Kluwer Academic Publisher, Boston 1999, pp. 351-369

[Dubl99] Dublin Core Metadata Initiative, Dublin Core Metadata Element Set, Version 1.1: Reference Description. 1999. http://purl.org/dc

[Erdm01] M. Erdmann: Ontologien zur konzeptionellen Modellierung der Semantik von XML, Dissertation, Universität Karlsruhe, 2001

[Etzi96] O. Etzioni: The world wide web: Quagmire or gold mine. Communications of the ACM, 39(11):65-68, 1996.

[FaPS96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth: From data mining to knowledge discovery: An overview. In Advances in Knowledge Discovery and Data Mining, pages 1-34. AAAI Press, 1996.

[FeDa95] R. Feldman, I. Dagan: Knowledge discovery in textual databases (kdt). In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), pages 112-117, Montreal, Canada, 1995.

[FFLS97] M. F. Fernandez, D. Florescu, A. Y. Levy, D. Suciu: A query language for a web-site management system. SIGMOD Record, 26(3):4-11, 1997.

[FHH+00a] D. Fensel, I. Horrocks, F. Harmelen, S. Decker, M. Erdmann, M. Klein: OIL in a Nutshell. http://www.cs.vu.nl/ dieter/ftp/spool/oil.nutshell.pdf.

[FHH+00b] [D. Fensel, I. Horrocks, F. Harmelen, S. Decker, M. Erdmann, M. Klein: A brief Introduction to OIL. http://www.cs.vu.nl/ dieter/ftp/paper/oil.cuba.pdf.

[Fiel94] R. Fielding: Maintaining Distributed Hypertext Infostructures: Welcome to MOMspider's Web. Chapter 6, The Need for Visible Metainformation. 1994. http://www.ics.uci.edu/pub/websoft/MOMspider/ WWW94/meta.html

[FlLM98] D. Florescu, A. Y. Levy, A. O. Mendelzon: Database techniques for the world-wide web: A survey. SIGMOD Record, 27(3):59-74, 1998

[FPSU96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy: editors Advances in Knowldege Discovery and Data Miningg. AAAIPress / The MIT Press, 1996

[FPW+99] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, C. G. Nevill-Manning: Domain-specific keyphrase extraction. In Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99, pages 668-673, 1999.

[Frei98] D. Freitag. Information extraction from html: Application of a general learning approach. In Proceedings of the Fifteenth Conference on Artificial Intelligence AAAI-98 (1998), pages 517-523, 1998.

[Glus99] R. J. Gluschko, J. M. Tenenebaum, B. Metzler: An XML Framework for Agent-based E-commerce. in: Communications of the ACM 42(3) March 1999. pp. 106-114

[GrMe99] S. Grumbach, G. Mecca: In search of the lost schema. In Database Theory - ICDT '99, 7th International Conference, pages 314-331, 1999.

[Grub93a] T. R. Gruber: A translation Approach to Portable Ontology Specifications. In: Knowledge Acquisition . 5(2), Academic Press, 1993. pp. 199-220

[Grub93b] T. R. Gruber: Towards Principles for the Design of Ontologies Used for Knowledge Sharing, *http://www-ksl.stanford.edu/knowledge-sharing/papers/onto-design.ps*

[GoWi97] R. Goldman, J. Widom: Dataguides: Enabling query formulation and optimization in semistructured databases. In M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, editors, VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece, pages 436-445. Morgan Kaufmann, 1997.

[GoWi99] R. Goldman, J. Widom: Approximate dataguides. In Proceedings of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats, 1999.

[GuMV99] N. Guarino, C. Masalo, G. Vetere: OntoSeek - Content-based Access to the Web. in: IEEE Intelligent Systems 14(3), May/June 1999. pp. 70-80 http://www.ladseb.pd.cnr.it/infor/Ontology/ Papers/OntoSeek.pdf

[HaGa97] J. Hammer, H. Garcia-Molina, J. Cho, A. Crespo, R. Aranha: Extracting semistructured information from the web. In Proceedings of the Workshop on Management of Semistructured Data, pages 18-25, 1997.

[HaPH01] F. Harmelen, P.Patel-Schneider, I.Horrocks: Reference description of the DAML+OIL (March 2001) ontology markup language, 2001. http://www.daml.org/2001/03/reference.html

[Hear99] M. A. Hearst: Untangling text data mining. In Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics, 1999.

[Heer96] R. Heery: Review of Metadata Formats. In: Program, Bd. 30, Nr. 4, S. 345-373, October 1996. http://www.ukoln.ac.uk/metadata/review.html

[HKLK97] T. Honkela, S. Kaski, K. Lagus, T. Kohonen: Websom - self-organizing maps of document collections. In Proc. of Workshop on Self-Organizing Maps 1997 (WSOM'97), pages 310-315, 1997.

[HsDu98] C. N. Hsu, M. T. Dung: Generating finite-state transducers for semi-structured data extraction from the web. Information Systems, 23(8):521-538, 1998.

[IMSP99] IMS Project: IMS Meta-Data Specification Draft. 1999. http://www.imsproject.org/metadata.html

[KaHS97] H. Kargupta, I. Hamzaoglu, B. Stafford: Distributed data mining using an agent based architecture. In Proceedings of Knowledge Discovery And Data Mining, pages 211-214. AAAI Press, 1997.

[Kash99] V. Kashyap: Design and Creation of Ontologies fpr Environmental Information Retrieval. in: Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW-99), Banff, Canada, October 1999. http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/ Kashyap1/kaskyap.pdf

[Kess95] M.Kesseler: A Schema Based Aprroach to HTML Authoring. in: Proceedings of the 4th Int. World Wide Web Conference (WWW-4). Boston, December 1995

[Klei99] J. Kleinberg. Authoritative sources in a hyperlinked environment. J. of the ACM, 1999, to appear. Also appears as IBM Research Report RJ 10076(91892) May 1997.

[KoBl00] R. Kosala, H. Blockeel: Web Mining Research: A Survey SIGKDD Explorations. Copyright c ACM SIGKDD, July 2000.

[KRR+00] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Sivakumar, E. Upfal: The Web as a graph

[KRRT99] S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins: Trawling emerging cybercommunities automatically. Proc. 8th WWW Conf., 1999.

[Kusk99] N. Kushmerick: Gleaning the web. IEEE Intelligent Systems, 14(2):20-22, 1999.

[KuWD97] N. Kushmerick, D. Weld, R. Doorenbos: Wrapper induction for information extraction. In Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-97, pages 729-737, 1997.

[LaFi99] Y. Labrou, T. W. Finin: Yahoo! As an Ontology: Using Yahoo! Categories to Describe Documents. in: Proceedings of the 1999 ACM CIKM International Conference on Information and Knowledge Management, Kansas City, Missouri. November, 1999. ACM Press. pp. 180-187

[Lam01] S. Lam: The Overview of Web Search Engines, Department of Computer Science, University of Waterloo, 2001

[LaSS96] L. Lakshmanan, F. Sadri, I. Subramanian: A declarative language for querying and restructuring the web. In Proceedings of 6th. International Workshop on Research Issues in Data Engineering, RIDE '96, pages 12-21, 1996.

[Lass97] O. Lassila: Introduction to RDF Metadata, W3C Note, 1997, http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html

[Lang99] P. Langley: User modeling in adaptive interfaces. In Proceedings of the Seventh International Conference on User Modeling, pages 357-370, 1999.

[Lege99] H. Legenstein: Qualitätsaspekte zur Wissensauffindung un Testimplementierung für das xFIND-System, Diplomarbeit an der Technischen Universität Graz, Östereich

[LOM+98] IEEE Learning Technology Standards Committee (LTSC), Learning Object Metadata (LOM) Draft Document v2.1, 1998. http://ltsc.ieee.org/doc/wg12/LOMdoc2_1.html

[LuSR96] S. Luke, L. Spector, D. Rager: Ontology-based Knowledge Discovery on the World-Wide Web. in: Proceedings of the Workshop on Internet-based Information Systems at AAAI-96. Portl, Oregon 1996

[MAG+97] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, J. Widom: Lore: A database management system for semistructured data. SIGMOD Record, 26(3):54-66, September 1997

[MaSp00] B. Masand, M. Spiliopoulou: Webkdd-99: Workshop on web usage analysis and user profiling. SIGKDD Explorations, 1(2), 2000.

[MBNL99] S. K. Madria, S. S. Bhowmick, W. K. Ng, E. P. Lim: Research issues in web data mining. In Proceedings of Data Warehousing and Knowledge Discovery, First International Conference, DaWaK '99, pages 303-312, 1999.

[MeMM96] I. O. Mendelzon, G. A. Mihaila, T. Milo: Querying the world wide web. In Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems, pages 80-91, 1996.

[Mill90] G. A. Miller: Wordnet. An On-Line Lexical Database. in: International Journal of Lexicography 3-4, 1990.pp. 235-312

[Mitc99] T. M. Mitchell: Machine learning and data mining. Communications of the ACM, 42(11):30-36, 1999.

[MKIS98] E. Mena, V. Kashyap, A. Illaramendi, A. Sheth: Domain Specific Ontologies for Semantic Information Brokering on the Global Informatione Infrastructure. in: Proceedings of the first International Conference on Formal Ontologies in Information Systems (FOIS-98), Trento, Italy, Frontiers in Artificial Intelligence and Applications, vol. 46, IOS-Press, June 1998

[Mlad99] D. Mladenic: Text-learning and related intelligent agents. IEEE Intelligent Systems, 14(4):44-54, 1999. SIGKDD Explorations.

[MNRS99] A. McCallum, K. Nigam, J. Rennie, K. Seymore: A machine learning approach to building domain-specific search engines. In Proceedings of

the International Joint Conference on Artificial Intelligence IJCAI-99, pages 662-667, 1999.

[MuMK98] I. Muslea, S. Minton, C. Knoblock: Wrapper induction for semistructured, web-based information sources. In Proceedings of the Conference on Automatic Learning and Discovery CONALD-98, 1998.

[Musl99] I. Muslea: Extraction patterns for information extraction tasks: A survey. In AAAI-99 Workshop on Machine Learning for Information Extraction, 1999.

[NeAM97a] S. Nestorov, S. Abiteboul, R. Motwani: Infering structure in semistructured data. SIGMOD Record, 26(4), 1997.

[NeAM97b] S. Nestorov, S. Abiteboul, R. Motwani: Extracting schema from semistructured data. In L. M. Haas and A. Tiwary, editors, SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA, pages 295-306. ACM Press, 1998.

[Note99] G. R. Notess: Search Engine Showdown, 1999

[PaGW95] Y. Papakonstantinou, H. Garcia-Molina, J. Widom: Öbject Exchange Across Heterogeneous Information Sources,Ïn Proceedings of Eleventh International Conference on Data Engineering, Taipei, Taiwan, 251-260, 1995

[PDHS96] P. Buneman, S. B. Davidson, G. G. Hillebrand, D. Suciu: A query language and optimization techniques for unstructured data. In H. V. Jagadish and I. S. Mumick, editors, Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996, pages 505-516. ACM Press, 1996.

[PICS96] PICS Label Distribution Label Syntax and Communication Protocols, Version 1.1, W3C Recommendation, 1996; http://www.w3.org/TR/REC-PICS-labels

[Posc00] Z. Poscai: Ontolgiebasiertes Wissensmanagement für die Produktentwicklung. Forschungsberichte aus dem Institut für Rechneranwendung in Planung und Konstruktion der Universität Karlsruhe. Band 3/2000. ShakerVerlag 2000.

[PPK+00] G. Paliouras, C. Papatheodorou, V. Karkaletsis, P. Tzitziras, C. D. Spyropoulos: Large-scale mining of usage data on web sites. In AAAI 2000 Spring Symposium on Adaptive User Interfaces, 2000.

[RaLJ98] D. Raggett, A. Le Hors, I. Jacobs (ds.): HTML 4.0 Specification W3C Recommendation, 24. April 1998. http://www.w3.org/TR/REC-html40

[RDFMS99] Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation 22 February 1999; http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

[RDFS99] Resource Description Framework (RDF) Schema Specification W3C Proposed Recommendation 03 March 1999; http://www.w3.org/TR/1999/PR-rdf-schema-19990303

[ReMc99] J. Rennie, A. McCallum: Using reinforcement learning to spider the web efficiently. In Proceedings of the 16th International Conference on Machine Learning ICML-99, 1999.

[Rijs79] C. J. van Rijsbergen: Information Retrieval. Butterworths, 1979.

[SaAz01] A. Sahuguet, F. Azavant: Building intelligent Web applications using lightweight wrappers. in: Data and Knowledge Engineering, Special Issue „Heterogeneous Information Resources Need Semantic Access“. Volume 36, Issue 3, Elsevier, March 2001.pp. 283-316

[SBB+99] S.Staab, C.Braun, I.Bruder, A.Düsterhöft, A.Heuer, M.Klettke, G.Neumann, B.Prager, J.Pretzel, H.-P. Schnurr, R.Studer, H.Uszkoreit, B.Wrenger: A System for Facilitating and Enhancing Web Search. in: Engineering Applications of Bio-Inspired Artificial Neural Networks (IWANN-99) - Proceedings of International Working Conference on Artificial and Natural Neural Networks. Volume 2. Alicante, Spain, June 1999. LNCS 1607, Springer, Berlin, 1999. pp.706-714

[Schö01] V. C. Schöch: Die Suchmaschine Google, Institut für Informatik Freie Universität Berlin vschoech@inf.fu-berlin.de 19. Juni 2001

[Soder96] S. Soderland: Learning information extraction rules for semi-structured and free text. Machine Learning, 34(1-3):233-272, 1996.

[SOIF96] Wessels, Duane: The Summary Object Interchange Format (SOIF). 1996. http://www.tardis.ed.ac.uk/harvest/docs/old-manual/node151.html

[Sper97] E. Spertus: ParaSite: Mining structural information on the Web. Proc. 6th WWW Conf., 1997.

[Spil99] M. Spiliopoulou: Data mining for the web. In Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '99, pages 588-589, 1999.

[Sriv00] J. Srivastava, R. Cooley, M. Deshpande, P. N. Tan: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations, 1(2), 2000.

[StBF98] R. Studer, V. R. Benjamins, D. Fensel: Knowledge engineering, principles and methods. In Data and Knowledge Engineering, 25(1-2), S. 161-197, 1998. ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/DKE98.ps

[Tan99] A. H. Tan: Text mining: The state of the art and the challenges. In Proc of the Pacific Asia Conf on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases, pages 65-70, 1999.

[Toiv99] H. Toivonen: On knowledge discovery in graphstructured data. In Workshop on Knowledge Discovery from Advanced Databases (KDAD'99), pages 26-31, 1999.

[WaLi97] K. Wang, H. Liu: Schema discovery for semistructured data. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97), pages 271-274, 1997.

[Wang99] H. L. K. Wang. Discovering association of structure from semistructured objects. To appear in IEEE Transactions on Knowledge and Data Engineering, 1999.

[Wies92] G.Wiederhold: Mediators in the architecture of future information systems. in: IEEE Computer 25(3) 1992, pp. 38-49

[Wilk97] Y. Wilks: Information Extraction as a core language technology, volume 1299 of Lecture Notes in Computer Science, chapter In M-T. Pazienza (ed.), Information Extraction, pages 1-9. Springer, 1997.

[XML98] Extensible Markup Language (XML) 1.0, W3C Recommendation, http://www.w3.org/TR/REC-xml

[ZaHa98] O. Zaiiane, J. Han: Webml: Querying the world-wide web for resources and knowledge. In Proc. ACM CIKM'98 Workshop on Web Information and Data Management (WIDM'98), pages 9-12, 1998.

[ZHL+00] O. R. Zaiane, J. Han, Z. N. Li, S. H. Chee, J. Chiang: Multimediaminer: a system prototype for multimedia data mining. In Proc. ACM SIGMOD Intl. Conf. on Management of Data, pages 581-583, 1998. SIGKDD Explorations. Copyright c ACM SIGKDD, July 2000. Volume 2, Issue 1 - page 15

Thesen

1. Bei der Realisierung einer spezialisierten Suchmaschine ist eine inhaltliche sowie eine strukturelle Domänenanalyse notwendig.

2. Innerhalb der strukturellen Domänenanalyse dient die globale Strukturanalyse dem Lokalisieren von Expertenseiten im World-Wide Web. Ergebnis der lokalen Strukturanalyse ist die Markierung von HTML-Elementen, die inhaltliche Wissensstrukturen enthalten. Ein komplexer Aufbau von HTML-Seiten erlaubt nicht immer eine fehlerfreie Strukturanalyse.

3. Das Wrapper-Toolkit W4F, in Java implementiert, stellt mächtige Funktionen bereit, um statischen und dynamischen HTML-Code zu extrahieren. Beim Einsatz von W4F werden Retrieval-, Extraktionsregeln und Mappingmechanismen spezifiziert.

4. Durch das Analysieren der Linkstrukturen auf Web-Seiten erhält man wertvolle semantische Informationen, mit deren Hilfe Suchmaschinen sehr genaue Antworten auf Anfragen geben können.

5. Für die Anwendungsdomäne „Wetter in Deutschland“ existieren mehrere Expertenseiten, die für den hier prototypisch umgesetzten Suchdienst genügend Informationen bereitstellen.

6. Mit Hilfe von Ontologien können Themengebiete modelliert werden. Zum Aufbau von Ontologien eignet sich die Sprachfamilie des *Semantic Web*. In der hier vorgestellten Realisierung wird die Ontologie auf relationale Datenstrukturen abgebildet.

7. Die Technologie der Suchmaschinen läßt sich in die vier Typen: crawlerbasierter Suchdienst, Katalogdienst, Meta-Suchmaschine und spezialisiertes Suchsystem unterteilen. Der umgesetzte Suchdienst dieser Arbeit setzt auf dem crawler-basierten Ansatz auf und zeichnet sich zudem durch Eigenschaften einer spezialisierten Suchmaschine aus.

Erklärung

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig und nur unter Vorlage der angegebenen Literatur und Hilfsmittel angefertigt habe.

Rostock,den 28.02.2000 Christian Dethloff