Thema:

Entwurf und Implementierung eines Webservices und eines passenden Clients zur Klassifikation von Dokumenten

Studienarbeit

Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Prof. Dr. rer. pol. habil. Hans-Knud Arndt

Betreuer: Prof. Dr. rer. pol. habil. Hans-Knud Arndt

Vorgelegt von: Mathias Kant

Abgabetermin: 19.01.07

Danksagung

Ich danke allen Mitarbeitern der Firma Liske-Informationsmanagementsysteme. Besonderer Dank geht dabei an meinen Betreuer Frank Kehring, der mir bei Fragen stets mit Sachverstand zur Seite stand.

Weiterhin danke ich Prof. Dr. Hans-Knud Arndt für die gute Betreuung meiner Studienarbeit an der Otto-von-Guericke-Universität.

Inhaltsverzeichnis

Verzeichnis der Abkürzungen und Akronyme

o. V.	ohne Verfasser
SQL	Structured Query Language
XML	Extensible Markup Language
SOAP	Service-oriented Application Protocol
WSDL	Webservice Definition Language
UDDI	Unify Definition Description Interface
TCP/IP	Transmission Control Protocol/Internet Protocol
HTTP	Hyper Text Transfer Protocol

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

In den letzten Jahren wuchs die Informationsflut in unserer Gesellschaft mehr und mehr, so dass es fast unmöglich ist, so viele Informationen ohne irgendwelche Hilfsmittel gut zu verwalten. Das Ziel vieler Firmen und Organisationen ist es, diese Mengen an Informationen möglichst schnell und günstig zu verwalten, so dass die Informationen in kurzer Zeit am richtigen Ort bzw. Arbeitsplatz sind. Die Systeme zur Verwaltung müssen aber erst eingerichtet werden, wodurch meist schon hohe Kosten entstehen, was viele Firmen abschreckt Systeme anzuschaffen, die die komplexen Strukturen der Informationen effizient und effektiv verwalten. Oft werden einfache Verfahren eingeführt, die günstiger aber größtenteils auch zeitintensiver sind.

Um diese Menge an Informationen verwalten zu können, ist eine Kategorisierung oder Klassifikation hilfreich und notwendig. Für die Klassifikation gibt es einfache Verfahren wie eine alphabetische Sortierung oder auch komplexere, die nach einer bestimmten Formel die Klassifikation durchführen und die Informationen in verschiedene Klassen einordnen.

Ziel

Das Ziel des Praktikums bei der Firma Liske-Informationsmanagementsysteme war es, einen Webservice zu implementieren, der sich mit der Klassifikation von Dokumenten im Allgemeinen beschäftigt. Dieser sollte mit Hilfe eines Clients aufzurufen sein und die Dokumente in guter Qualität und in einer annehmbaren Zeit einem Bereich zuordnen.

Aufbau der Arbeit

Zunächst werden in Kapitel 1 die wichtigen Begriffe wie Klassifikation und Webservice definiert und die in der Arbeit benutzten Formeln, Programme und Sprachen erklärt. Im Anschluss wird im Kapitel 2 die grundlegende Idee beschrieben. In Kapitel 3 wird die Umsetzung der im Entwurf entwickelten Idee betrachtet. Anschließend folgt ein Anwendungsbeispiel, um die Theorie aus der Implementierung zu erklären. Zum Schluss der Arbeit wird alles kurz zusammengefasst und die implementierten Webservices werden in ihrer Realisierbarkeit bewertet und Verbesserungsvorschläge werden genannt.

2 Grundlagen

2.1 XML-Webservice

XML-Webservices sind Softwarekomponenten, welche bestimmte Funktionalitäten über ein Netzwerk bereitstellen. Interaktionen mit diesen erfolgen in den meisten Fällen über gewöhnliche Internetprotokolle wie TCP/IP (Transmission Control Protocol / Internet Protocol), HTTP (Hyper Text Transfer Protocol) oder XML (Extensible Markup Language) sowie neueren Standards wie SOAP (Service-oriented Application Protocol), WSDL (Webservice Definition Language) oder UDDI (Unify Definition Description Interface). Größtenteils sind diese Services keine kompletten Lösungen, sondern nur funktionale Komponenten einer größeren Lösung, die ihre Funktionalitäten über wohl definierte Schnittstellen zur Verfügung stellen. Vorwiegend werden sie in öffentlichen oder privaten Verzeichnissen wie UDDI zur Verfügung gestellt. Die Technologie kann wie in der folgenden Abbildung dargestellt werden:[1]

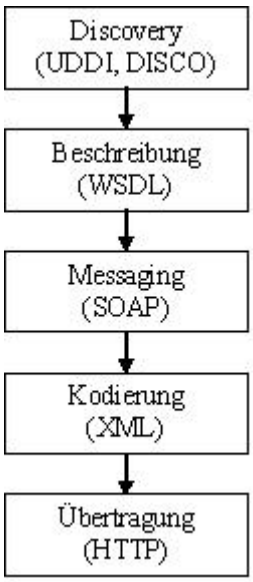

Abb. 2.1: Technologie eines Webservices

[1] Freeman, A.; Jones, A.(2003), Seite 27-28

2.1.1 Vorteile von XML-Webservices

Mit der Nutzung eines Webservices ergeben sich eine Reihe von Möglichkeiten, welche durch folgende Vorteile erreicht werden können:

- **Basieren auf Standards:** Das heißt jeder Webservice verwendet die gleichen Protokolle und kodiert die Daten konsistent.
- **Herstellerunabhängig:** Die Standards, auf denen sie basieren, wurden von Technologie- und sonstigen Unternehmen sowie akademischen Institutionen gemeinsam entwickelt.
- **Einfach:** Webservices sind einfach, weil Programmierer im Gegensatz zu anderen Technologien für verteiltes Rechnen nicht alle Standards und Protokolle verstehen müssen. Diese werden durch verschiedene Entwicklungstools wie Microsoft Visual Studio .NET abstrahiert.
- **Sprach- und plattformunabhängig:** Sie verlangen weder die Benutzung einer bestimmten Programmiersprache noch eines bestimmten Betriebssystems.
- **Funktionale Abstraktion:** Die Funktionalität des Webservices muss nicht direkt zur darunter liegenden funktionalen Implementierung zuzuordnen sein. So können sie beispielsweise Schnittstellen bieten, die die Funktionalität verschiedener darunter liegender Systeme zusammenfasst.
- **Auffindbar:** Webservices sind bei Providern registriert. Von dort aus können sie die Anwender finden und benutzen.
- **Verkürzte Entwicklungszeit:** Webservices sind wieder verwendbar, wodurch die Entwicklungszeit von neuen Webservices erheblich verkürzt werden kann.[2]

2.1.2 Aufbau von XML-Webservices

Im ersten Schritt sucht der Client im UDDI –Verzeichnis einen Webservice, wo er eine URL erhält. Anschließend fordert der Client das Discovery-Dokument an, in welchem eine URL zu einem Beschreibungsdokument steht. Danach fordert er dieses Beschreibungsdokument an und erzeugt einen Webservice-Proxy. Durch den Proxy braucht der Programmierer nicht die gesamte Interaktion mit dem Webservice manuell

[2] Freeman, A.; Jones, A.(2003), Seite 32-34

durchführen. Im vierten und letzten Schritt interagiert der Client über den Proxy mit dem Webservice.[3]

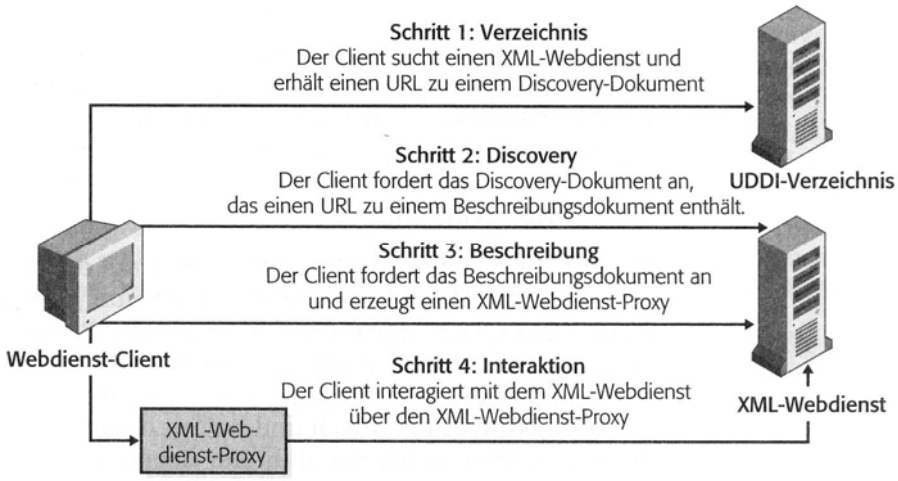

Abb. 2.2: Aufbau eines XML-Webservices[4]

2.2 SQL (Structured Query Language)

SQL ist eine deklarative Datenbanksprache für relationale Datenbanken, wobei die Datenstrukturen durch die relationale Algebra definiert werden. Mittels SQL können Datenbestände manipuliert werden durch Anfügen, Bearbeiten und Löschen von Datensätzen. Eine der Hauptfunktionen ist jedoch die Anfrage von Daten.[5]

SELECT-, FROM-, WHERE –Anfrage

Der so genannte SFW-Block dient der Anfrage von Daten aus einer Datenbank. Ein sehr allgemeines Beispiel ist folgendes:

```
select distinct Vorname, Nachname
from Studenten
where Alter=´22´
order by Nachname asc
```

Die `select`-Klausel gibt an, welche Spalten einer oder mehrerer Tabellen als Ergebnis angezeigt werden sollen.

[3] Freeman, A.; Jones, A.(2003), Seite 43-45
[4] Freeman, A.; Jones, A.(2003), Seite 43
[5] http://de.wikipedia.org/wiki/SQL

Die `from`-Klausel spezifiziert die zu verwendenden Relationen, die durch ein kartesisches Produkt verknüpft werden. Ebenfalls können Umbenennungen durchgeführt werden.

Um Selektionsbedingungen zu den Relationen der `from`-Klausel einzufügen, wird die `where`-Klausel verwendet.

Mit Hilfe der `distinct`-Klausel können mehrfach auftretende Einträge gelöscht werden.

Die `order by`-Klausel sortiert eine ungeordnete Liste von Ergebnissen. Dabei bedeutet `asc` aufsteigende Sortierung und `desc` absteigende Sortierung. [6]

INSERT – Anweisung

Die `insert`-Anweisung ermöglicht das Einfügen eines oder mehrerer Tupel in eine Relation.

```
insert
into Prüfung (Fach, Note)
values (´Managementinformationssysteme´, 2)
```

Attribute, die nicht aufgeführt werden, werden auf `null` gesetzt. Die Zuordnung der Werte zu den Attributen wird durch die Reihenfolge festgelegt. Falls die Attributliste fehlt, müssen alle Attribute in der Reihenfolge angegeben werden, wie sie waren, als die Relation erstellt wurde.[7]

DELETE – Anweisung

Mit der `delete`-Anweisung wird das Löschen eines oder mehrerer Tupel aus einer Relation realisiert.

```
delete from Studenten
where MatrNr=´176426´
```

Es werden alle Tupel aus der betreffenden Relation gelöscht, die die Selektionsbedingungen `where` erfüllen.[8]

---

[6] Heuer, A.; Saake, G.(2000), Seite 340-351, 363-369
[7] Heuer, A.; Saake, G.(2000), Seite 373-374
[8] Heuer, A.; Saake, G.(2000), Seite 372-373

UPDATE – Anweisung

```
update Studenten
set Vorname = ´Mathias´
where Nachname = ´Kant´
```

Die `update`-Anweisung ermöglicht ein Ändern eines oder mehrerer Tupel einer Relation. In allen Tupeln der Relation, die die Bedingung `where` erfüllen, werden die Werte des Attributs wie in der `set`-Klausel angegeben ersetzt.[9]

2.3 Klassifikation

2.3.1 Definition

„Klassifikation im herkömmlichen Sinn bedeutet die Einteilung in bestimmte Kategorien oder Klassen. Es werden also Zugehörigkeitsentscheidungen basierend auf verfügbaren Daten getroffen. Ein Klassifikationsprozess ist demnach die wiederholte Anwendung dieser Entscheidungsfindung in neuen Situationen [Mitchie et al. 1994]. Sinn und Zweck der Klassifikation ist es, ein logisches System zu schaffen, damit Objekte jederzeit wieder auffindbar sind, und somit das Chaos minimieren. [Walther 2001]“[10]

Allgemein kann Klassifikation als Oberbegriff für alle Methoden gesehen werden, die sich mit der Einordnung von Daten und Objekten beschäftigen. In dieser Arbeit wird der Begriff Klassifikation als Einteilung von Daten in vorher bereits bekannte Klassen definiert.[11]

2.3.2 Funktionen zur Klassifikation

Um Dokumente zu klassifizieren, können eine Vielzahl von Verfahren und Funktionen angewendet werden. Dabei wird unterschieden in lernende (statistische) und nicht lernende (regelbasierte) Verfahren. Regelbasierte Verfahren führen die Klassifikation immer nach den gleichen Methoden und mit dem gleichen Material durch. Die statistischen Verfahren müssen erst mit Trainingsdokumenten trainiert werden. Anhand dieser klassifiziert die jeweilige Software dann die Daten.

---

[9] Heuer, A.; Saake, G.(2000), Seite 371-372
[10] Hoffmann, R. (2002), Seite 30
[11] Hoffmann, R. (2002), Seite 51,52

2.3.2.1 TF-IDF-Formel

$$w_{ij} = t_{ij} * \log\left(\frac{N}{f_j}\right)$$

w_{ij} Gewicht von Term j in Dokument i

t_{ij} Zahl des Auftretens von Term j in Dokument i

N Zahl der Dokumente

f_j Zahl der Dokumente die den Term j ein oder mehrmals enthalten

Bei dieser Formel spielt das Auftreten von einem Term j im Dokument i eine wichtige Rolle. Je häufiger ein Term in einem Dokument vorkommt, desto relevanter ist er für die Klassifikation. Taucht einer dieser Terme in mehreren Dokumenten auf, sinkt die Bedeutung für die Klassifikation. Die TF-IDF-Formel berücksichtigt diesen Fall und ist somit nützlich, um die Klassifikation zu unterstützen.[12]

2.3.2.2 Kosinus-Maß

Die Distanz ist der Kosinus des Winkels α(x,y) zwischen den Vektoren x und y. Dabei werden zwei Vektoren miteinander verglichen. Das Ergebnis gibt die Ähnlichkeit der beiden Vektoren wieder. Je höher dieser Wert ist, desto ähnlicher sind sie. Die maximale Ähnlichkeit liegt bei 1 und die minimale bei -1.

$$d(x,y) = \cos\alpha(x,y) = \frac{x * y}{|x| * |y|} = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}}$$

2.4 Microsoft Visual Studio

Die Webservices in dieser Arbeit wurden mit dem Entwicklungstool Microsoft Visual Studio geschrieben. Dabei handelt es sich um eine integrierte Entwicklungsumgebung für Hochsprachen. Das Tool bietet die Möglichkeit, in verschiedenen Programmiersprachen wie Visual Basic, C/C++, C# oder Java zu programmieren. Hier wurde die Sprache Visual Basic verwendet.

[12] Brückner, T.; Dambeck, H. (2003) Seite 194

3 Entwurf

3.1 Einarbeitung

Am Anfang meines Praktikums sollte ich mich mit dem Thema Klassifikation beschäftigen. Das langfristige Ziel war, einen Webservice, der Dokumente in bestimmte Klassen einordnet, zu entwerfen und zu implementieren. Beim Recherchieren ergaben sich viele Möglichkeiten, Dokumente zu klassifizieren. Ich fand vorgegebene käuflich zu erwerbende Software sowie einzelne Methoden wie das Kosinus-Maß oder die TF-IDF-Formel. Ein Beispiel für käuflich zu erwerbende Software ist SERglobalBRAIN[13]. Bei dieser Software werden vorher Klassen definiert und während der Klassifikation werden die Dokumente dann nach einem Klassenprofil in eine bestimmte Klasse eingeordnet. Software wie Content Surveyor[14] führt eine sehr komplexe Klassifikation durch, welche sehr viele Dateiformate verarbeitet und alles sehr anschaulich visualisiert.

Nach einer Besprechung mit meinem Betreuer in der Firma Liske[15] kristallisierten sich zwei Hauptaufgaben heraus. Eine war einen Posteingang zu klassifizieren, um die Post in Gruppen einzuteilen. Dieser Webservice wurde in zwei Varianten unterteilt, einerseits mit einer Bereichsvorgabe, was eine Abfrage von Dokumenten eines bestimmten Bereichs (z.B. „Rechnungen“) aus der Eingangspost bedeutet, und andererseits ohne eine Bereichsvorgabe, so dass die Post automatisch in die passenden Ordner kopiert wird. Hierzu werden nur die ersten 200 Zeichen des Dokuments betrachtet, da Schlüsselbegriffe wie „Rechnung“ oder „Auftrag“ meist weit oben stehen. Die andere Hauptaufgabe war es, Webseiten zu klassifizieren. Dies hat den Sinn, gewisse Seiten zu einem bestimmten Thema aus einer Vielzahl von Webseiten automatisch zu selektieren. Beispielsweise sollte der Webservice aus den Webseiten www.n-tv.de oder auch www.spiegel.de alle Teilseiten ausgeben, die sich mit dem Thema „Fußball“ beschäftigen. Somit wird dem Nutzer langes Suchen nach Seiten, die ihn interessieren könnten, erspart. Im Laufe der Arbeit entstand noch die Aufgabe einen normalen Text zu klassifizieren. Dabei sollten die Möglichkeiten bestehen, einen Text einzugeben oder eine Textdatei zu öffnen und danach zu klassifizieren. Die gesamte Klassifikation sollte mit Hilfe eines Thesaurus durchgeführt werden, welcher später in Kapitel 3.3 beschrieben wird.

---

[13] http://www.itassistance.biz/docu/Prospekte/SERglobalBrain%20ApplSheet%202-p%20ger%20V1.31.pdf
[14] http://www.neuropower.de/product/CS/descr.ge.html
[15] Liske-Informationsmanagementsysteme: www.liske.de

Dazu sollte ein Client implementiert werden, mit dem die Webservices benutzt bzw. verwaltet werden können. Mit diesem Client sollte die Möglichkeit bestehen, sowohl den Posteingang als auch die Webseiten nach verschiedenen Verfahren zu klassifizieren. Zusätzlich sollte es noch möglich sein, den Thesaurus einzusehen und auch zu editieren.

3.2 Entwicklungsumgebung

Implementiert werden sollte der Webservice mit Visual Studio. Die Oberfläche dieses Entwicklungstools ist in der **Abb. 3.1** zu sehen. Ich entschied mich in Visual Studio für die Programmiersprache Visual Basic, da sie relativ intuitiv ist und sie mir bereits bekannt war. Mit Visual Studio können sowohl ein Webservice als auch der passende Windows Client implementiert werden. Dies erspart den Wechsel zwischen unterschiedlichen Programmen mit möglicherweise unterschiedlichen Programmier-sprachen und -umgebungen.

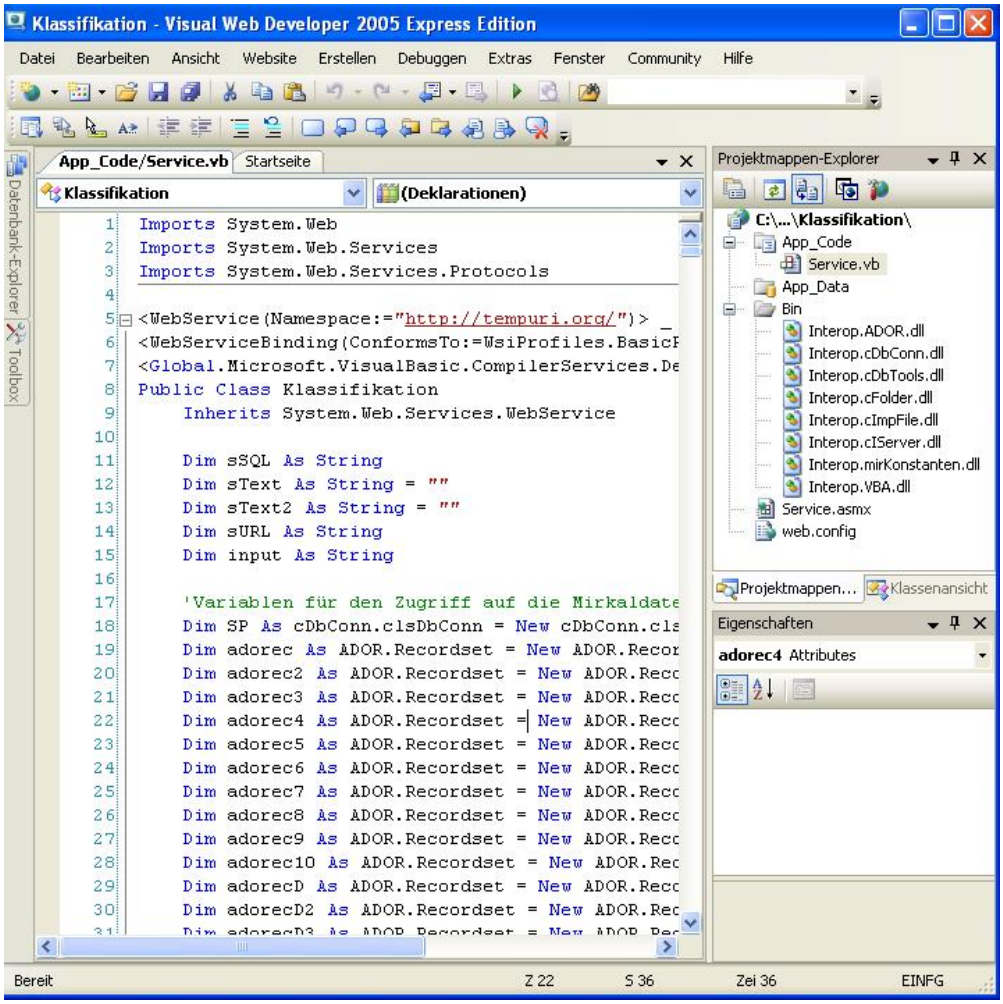

Abb. 3.1: Entwicklungstool Visual Web Developer von Visual Studio

Dabei wurde Mirakel als Datenquelle benutzt. Mirakel ist ein multimediales Dokumenten- und Informationsverwaltungssystem. Es wurde von der Firma Liske entwickelt und unterstützt die Archivierung und Verwaltung von Text- und Bilddateien. In Mirakel besteht die Möglichkeit, die zu erfassenden Informationen in verschiedene Archive einzupflegen und dort verschiedene Ordner einzurichten. Die Dokumente stehen dem Nutzer als Originalansicht, wie in **Abb. 3.2** zu sehen ist, und als Textansicht zur Verfügung. Mirakel hält die Dokumente in Form einer Access-Datenbank bereit. Diese Access-Tabellen ermöglichen es, auf die Texte und die entsprechenden Attribute zuzugreifen. In Mirakel konnte ich auch sehr gut beobachten, wie meine Implementierung funktioniert.[16]

Abb. 3.2: Mirakel von Firma Liske Informationsmanagementsysteme

[16] www.liske.de

3.3 Thesaurus

Unter einem Thesaurus wird im Allgemeinen eine geordnete Zusammenstellung von Begriffen verstanden, wobei er sich zum Indexieren, Speichern und Wiederauffinden eignet. Es wird auch eine Reihe von Beziehungen zwischen den Begriffen dargestellt und er dient der Vokabularkontrolle sowie der terminologischen Kontrolle.[17]

Zur Klassifikation benutzte ich einen Thesaurus. Dieser beinhaltet Begriffe, die Attribute besitzen und hierarchisch aufgebaut sind, wie in **Abb. 3.3** zu sehen ist. Ich erstellte in einer Access-Datenbank zwei Tabellen. In der einen Tabelle, die ich TBA_THESAURUS_LIST nannte, wurden die Hauptbegriffe wie z.B. „Sport“, „Politik“ oder „Wirtschaft“ meines Thesaurus verwaltet. In der anderen Tabelle, die TBA_THESAURUS_ENTRIES heißt, wurden dann die Ober- und die Unterbegriffe verwaltet. Zu jedem Hauptbegriff gibt es mehrere Oberbegriffe. Zu dem Hauptbegriff „Sport“ gibt es dann zum Beispiel „Fußball“, „Handball“ oder „Tennis“. Die Unterbegriffe sind die Begriffe, die auch am häufigsten in den Dokumenten auftreten. Zu dem Oberbegriff „Fußball“ gibt es dann beispielsweise die Unterbegriffe „Tor“, „Ballack“ oder „Bundesliga“.

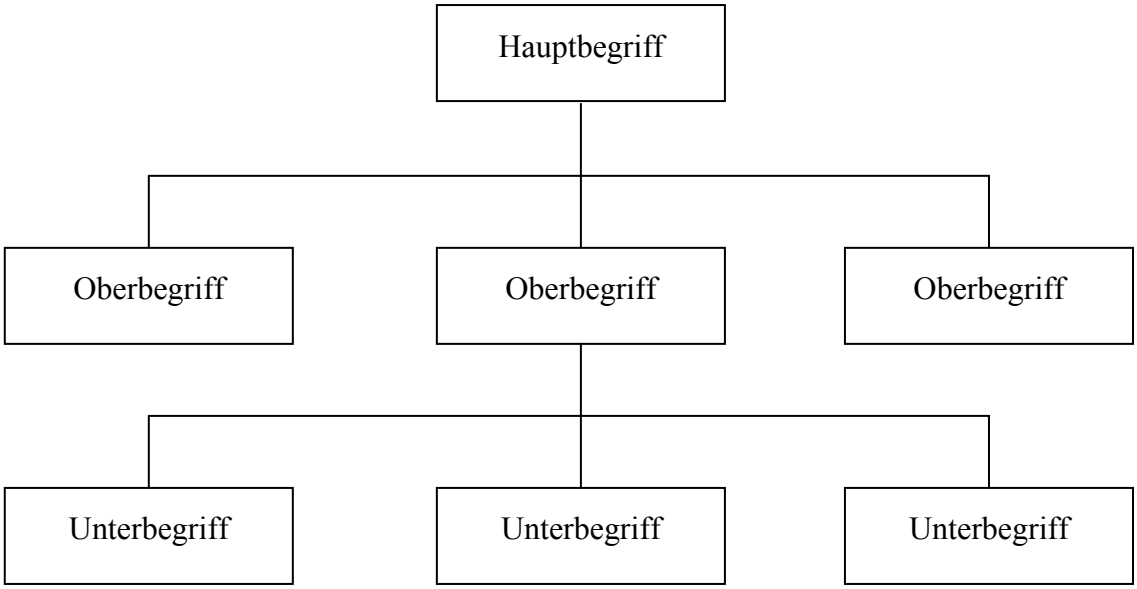

Abb. 3.3: Thesaurus-Modell

Die Tabelle TBA_THESAURUS_LIST enthält die Spalten TL_PKEY, TL_NAME, TL_BEM, TL_LDAT und TL_ODAT. TL_PKEY ist hierbei der Primärschlüssel, welcher mir erlaubte, die Hauptbegriffe mit den Begriffen der anderen Tabelle zu

[17] http://www.bui.haw-hamburg.de/pers/ulrike.spree/grundthes/grundthes.htm

verknüpfen. Die Spalte TL_NAME beinhaltet den Namen des Hauptbegriffs. TL_BEM, TL_LDAT und TL_ODAT sind für Bemerkungen, das letzte Änderungsdatum und das Erstellungsdatum gedacht.

Feldname	Datentyp	Beschreibung
TL_PKEY	Number PK	Index
TL_NAME	Text	Name des Hauptbegriffes
TL_BEM	Text	Allgemeines Bemerkungsfeld
TL_LDAT	DateTime	Datum der letzten Bearbeitung
TL_ODAT	DateTime	Erstellungsdatum

Tab. 3.1: Tabellenbeschreibung TBA_THESAURUS_LIST

TBA_THESAURUS_ENTRIES beinhaltet die Spalten TE_PKEY, TE_THE, TE_RefID, TE_NAME, TE_WERT, TE_TYP, TE_BEM, TE_LDAT und TE_ODAT. TE_PKEY ist hierbei der Primärschlüssel. Zu welchem Hauptbegriff der Ober- bzw. der Unterbegriff gehört, gibt TE_THE an. Mit der TE_RefID wird angegeben, zu welchem Oberbegriff der Unterbegriff gehört. Ist es ein Oberbegriff, steht dort eine „0“. Zur möglichen Festlegung der Relevanz habe ich die Spalte TE_WERT eingefügt. Einen bestimmten Typ wie z.B. Begriff, Übersetzung oder Synonym kann mit TE_TYP festlegt werden. TE_BEM, TE_LDAT und TE_ODAT haben die analogen Funktionen wie in der Tabelle TBA_THESAURUS_LIST.

Feldname	Datentyp	Beschreibung
TE_PKEY	Number PK	Index
TE_THE	Number	Hauptbegriffsnummer des Begriffs (referenziert auf TL_PKEY in TBA_THESAURUS_LIST)
TE_RefID	Number	Oberbegriffsnummer des Begriffs (referenziert auf TE_PKEY)
TE_NAME	Text	Name des Thesaurusbegriffs
TE_WERT	Number	Wertigkeit des Begriffs
TE_TYP	Text	Typ des Begriffs
TE_BEM	Text	Allgemeines Bemerkungsfeld
TE_LDAT	DateTime	Datum der letzten Bearbeitung
TE_ODAT	DateTime	Erstellungsdatum

Tab. 3.2: Tabellenbeschreibung TBA_THESAURUS_ENTRIES

3.4 Klassifikationsmethoden

Insgesamt benutzte ich folgende drei Methoden bzw. Formeln, um ein Dokument zu klassifizieren:

- **Normal:** Bei der „normalen“ Methode gibt es verschiedene Varianten. Eine davon ist:

$$w = \frac{AnzahlderVorkommen}{AnzahlderWörter} * 100$$

  Bei dieser Formel ergeben sich theoretisch Werte zwischen 0 und 100. Da Texte häufig mehrere 100 Wörter haben, liegen die Werte meist zwischen 0 und 1. Deswegen habe ich für die Ergebnisse dieser Formel einen Wertebereich von 0 bis 1 festgelegt. Alle Werte über 1 werden abgefangen und auf 1 gesetzt. So ergeben sich Werte, die sich gut analysieren und bewerten lassen.

- **TF-IDF-Formel:** Diese Formel wird zur Klassifikation der Eingangspost benutzt. Der Vektor wird mit den passenden Thesauruseinträgen, in dem Fall die Einträge zur Eingangspost (u.a. „Rechnung“ oder „Auftrag“), gefüllt. Diesen Einträgen wird als Wert die Anzahl der Dokumente angehängt, in denen der Eintrag vorkommt. In der Formel ist dies f_j. Mit der Auswertung dieser Vektoren kann dann die Formel benutzt werden. Der Thesauruseintrag mit dem höchsten w_{ij} wird dem Dokument zugeordnet.

$$w_{ij} = t_{ij} * \log\left(\frac{N}{f_j}\right)$$

- **Kosinus-Maß:** Mit dieser Formel wurde die Eingangspost mit Bereichsvorgabe klassifiziert. Hier wurden zwei Vektoren benutzt, um einerseits das Dokument und andererseits den ausgewählten Bereich zu repräsentieren. Diese beiden wurden dann verglichen mit dem Ergebnis der Ähnlichkeit.

$$d(x,y) = \cos\alpha(x,y) = \frac{x * y}{|x| * |y|} = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}}$$

4 Implementierung

4.1 Beginn

Zu Beginn der Implementierung probierte ich mich an einigen einfachen Beispielen von Webservices zum Thema Klassifikation. Ich benutzte zu Anfang fertige Funktionen, um einen Webservice zu implementieren, wobei ich mich mit der Auswertung und der Ausgabe der Klassifikation beschäftigte. Ich entschied mich für XML als Ausgabeformat des Webservices. Danach versuchte ich eine eigene Klassifikation zu implementieren. Um später auch den Thesaurus zu verwalten, implementierte ich, wie im nachfolgenden Kapitel 4.2 zu lesen ist, einen Webservice Thesaurus.

4.2 Webservice Thesaurus

Vor der Implementierung des eigentlichen Webservices zur Klassifikation von Dokumenten beschäftigte ich mich mit der Verwaltung des Thesaurus, welcher die Grundlage der Klassifikation sein sollte.

Der Webservice zur Verwaltung des Thesaurus beinhaltet die Funktionen, um die Hauptbegriffe, Ober- und Unterbegriffe ändern und löschen zu können, sowie Funktionen, um neue Hauptbegriffe, Ober- und Unterbegriffe einzufügen. Im Folgenden werden die elementaren Funktionen des Webservices beschrieben.

Funktion TableInsertHB

Diese Funktion beschäftigt sich mit dem Einfügen von Hauptbegriffen inklusive den Attributen des Begriffs. Vor dem Einfügen eines neuen Eintrags überprüft die Funktion, ob der Begriff schon vorhanden ist. Falls dies nicht der Fall ist, wird der Begriff mit der SQL-Anweisung `insert` in die Tabelle TBA_THESAURUS_LIST eingefügt. Diese Funktion gibt bei erfolgreichem Einfügen den boolean-Wert `true` aus.

Funktion	Variablen	Output	Kurzbeschreibung
TableInsertHB	sName, sBem, ldat, odat	Boolean	Einfügen von Hauptbegriffen
TableDeleteHB	sName	Boolean	Löschen von Hauptbegriffen
TableChangeHB	salterName, sneuerName, sBem, ldat	Boolean	Ändern von Hauptbegriffen
TableInsertThes	sTableName, sRefName, sName, sBem, styp, iWert, ldat, odat	Boolean	Einfügen von Oberbegriffe / Unterbegriffe
TableChangeThes	sTableName, sRefName, salterName, sneuerName, sBem, styp, iWert, ldat	Boolean	Ändern von Oberbegriffe / Unterbegriffe
TableDeleteThes	sName	Boolean	Löschen von Oberbegriffe / Unterbegriffe
Thesaurus		XmlDocument	Thesaurus laden
Hauptbegriffe		XmlDocument	Alle Hauptbegriffe laden
alleOberbegriffe	HB	XmlDocument	Alle Oberbegriffe zu einem Hauptbegriffe laden
Oberbegriffe		XmlDocument	Alle Oberbegriffe zu „Eingangspost“ laden
Deleteaktuell		Boolean	Löscht Ergebnisordner für die Eingangspost
VerwaltungsError	Err, Zeile	Boolean	Verwaltung der Fehlermeldungen

Tab. 4.1: Funktionen des Webservice „Thesaurus“

Funktion TableDeleteHB

Hier werden ausgewählte Hauptbegriffe aus dem Thesaurus gelöscht. Nach einer Prüfung der Existenz des zu löschenden Eintrags werden mit dem Hauptbegriff auch automatisch alle Ober- und Unterbegriffe gelöscht, die sich auf den Hauptbegriff beziehen. Das heißt, wenn der Hauptbegriff „Sport“ gelöscht wird, dann werden auch automatisch die Oberbegriffe wie „Fussball“, „Tennis“ oder „Handball“ gelöscht. Zusätzlich werden auch die entsprechenden Unterbegriffe gelöscht („Ballack“ oder „Bundesliga“ zu „Fussball“). Dies wird über die SQL-Anweisung `delete` ausgeführt. Bei erfolgreichem Löschen gibt die Funktion ein `true` aus.

Funktion TableChangeHB

Mit dieser Funktion können Hauptbegriffe und deren Attribute aus dem Thesaurus geändert werden. Zuerst wird geprüft, ob der Eintrag überhaupt vorhanden ist.

Anschließend wird der Hauptbegriff und dessen Attribute mittels der SQL-Anweisung `update` geändert. Wie unter anderem diese Funktion aufgebaut ist, ist in der **Abb. 4.1** zu sehen.

```
 1:     'Funktion um Hauptbegriffe (TBA_THESAURUS_LIST) zu ändern
 2:     <WebMethod()> _
 3:     Public Function TableChangeHB(ByVal salterName As String, ByVal sneuerName As
            String, ByVal sBem As String, ByVal ldat As DateTime) As Boolean
 6:
 7:         'zu ändernden Eintrag suchen
 8:         sSQL = "SELECT TL_PKEY, TL_NAME FROM TBA_THESAURUS_LIST WHERE TL_NAME='" &
                salterName & "'"
10:         adorec = SP.GetRecordset(sSQL:=sSQL)
11:
12:         If adorec Is Nothing Then
13:             VerwaltungError(SP.sLastDbError, 134)
14:             Return False
15:         End If
16:
17:         'wenn Eintrag nicht vorhanden ist
18:         If adorec.EOF = True Then
19:             VerwaltungError("Eintrag nicht vorhanden", 142)
20:             Return False
21:         Else
22:
23:             'Änderung des Eintrags
24:             sSQL = "Update TBA_THESAURUS_LIST " & _
25:                  "SET TL_NAME='" & sneuerName & "', TL_BEM='" & sBem & "',
                     TL_LDAT='" & ldat & "' " & _
27:                  "WHERE TL_PKEY=" & adorec.Fields(0).Value
28:             SP.ExecuteSQL(sSQL:=sSQL)
29:
30:             Return True
31:         End If
32:     End Function
```

Abb. 4.1: Codeausschnitt „TableChangeHB“

Funktion TableInsertThes

Diese Funktion behandelt das Einfügen eines Ober- bzw. eines Unterbegriffes in den Thesaurus bzw. in die Tabelle TBA_THESAURUS_ENTRIES. Am Anfang der Funktion wird überprüft, ob der Eintrag schon vorhanden ist. Falls dies nicht der Fall ist, wird der Hauptprozess dieser Funktion gestartet. Dort wird unterschieden, ob es ein Ober- oder ein Unterbegriff ist. Handelt es sich um einen Oberbegriff, so wird in der Tabellenspalte TE_RefID eine „0“ eingetragen. Sollte es jedoch ein Unterbegriff sein, wird in derselben Spalte die TE_PKEY von dem Oberbegriff eingetragen, auf den sich der Unterbegriff bezieht. „Fussball“ hat beispielsweise eine „0“ in der Spalte TE_RefID und „Bundesliga“ dann die TE_PKEY von „Fussball“. Das Einfügen wird dann analog zu der Funktion TableInsertHB mit der SQL-Anweisung `insert` realisiert.

Funktion TableChangeThes

Bei der Funktion kann ein Ober- oder Unterbegriff mit den dazu gehörenden Attributen geändert werden. Es wird der Begriff gesucht und dann, abhängig davon, ob es ein Ober- oder ein Unterbegriff ist, mit der SQL-Anweisung `update` geändert.

Funktion TableDeleteThes

Falls der Eintrag vorhanden ist, wird der Ober- bzw. der Unterbegriff mit all seinen Attributen gelöscht. Bei einem Oberbegriff werden auch die Unterbegriffe, die auf diesen referenzieren, gelöscht.

Funktion Thesaurus

Diese Funktion liest den kompletten Thesaurus aus und stellt ihn als Xml-Dokument für den Client bereit. Jeder Hauptbegriff aus der Tabelle TBA_THESAURUS_LIST und dessen Oberbegriffe aus TBA_THESAURUS_ ENTRIES werden in das Xml-Dokument geschrieben. Anschließend werden alle referenzierenden Unterbegriffe ebenfalls eingetragen. Dies wird für jeden Hauptbegriff durchgeführt und somit entsteht eine hierarchische Struktur im Xml-Dokument.

4.3 Webservice Klassifikation

Nachdem der Webservice Thesaurus fertig gestellt war, implementierte ich den Webservice Klassifikation, welcher das Kernstück meiner Arbeit darstellt. Dieser hält verschiedene Funktionen, die im Nachstehenden kurz erläutert werden, bereit, um Dokumente auf verschiedene Art und Weise zu klassifizieren.

Funktion automatKlass

Mit dieser Funktion können die Webseiten klassifiziert werden. Sie benötigt als Inputparameter ein Xml-Dokument und einen String. In dem Dokument werden die Webseiten bereitgestellt, die der Anwender klassifiziert haben möchte, wobei der String den Klassifikationsbereich angibt (z.B. „Sport“). Der Bereich kann aus der Menge der Hauptbegriffe des Thesaurus gewählt werden. Am Anfang werden die Webseiten in

Mirakel gespeichert und anschließend mit den Ober- bzw. Unterbegriffen zu dem gewählten Hauptbegriff verglichen. Überschreitet die Anzahl des Auftretens eines Begriffes in einer Teilseite einen bestimmten Wert, so wird diese zur Weitergabe an den Client in ein Xml-Dokument geschrieben, welches am Ende ausgegeben wird.

Funktion automatKlassEingPost1

Diese Funktion dient der Klassifikation der Eingangspost mit einer Bereichsvorgabe. Jedes Dokument der Eingangspost wird mit den Begriffen des ausgewählten Bereichs (z.B. „Rechnungen“) aus dem Thesaurus verglichen. Falls ein Begriff des Bereichs im Dokument vorkommt, wird dieses in den Ergebnisordner in Mirakel kopiert und gleichzeitig wird dessen Pfad in ein Xml-Dokument geschrieben, welches der Client weiter verarbeiten kann. Die Ergebnisse der Funktion können in Mirakel betrachtet werden.

Funktion automatKlassEingPost2

Analog zur vorherigen Funktion wird auch hier die Eingangspost klassifiziert mit dem Unterschied, dass diese in alle Bereiche (Oberbegriffe) gleichzeitig eingeordnet wird. Das heißt, es werden die Dokumente mit allen Ober- und Unterbegriffen verglichen. Bei Übereinstimmung wird das passende Dokument in den entsprechenden Ordner in Mirakel kopiert, wodurch Ordner wie „Rechnung“ oder „Auftrag“ entstehen. Die Pfade der Dokumente werden so in ein Xml-Dokument geschrieben, dass erkennbar ist, zu welchem Bereich es gehört.

Funktion Klass

Diese Funktion klassifiziert einen normalen Text, indem er mit der gesamten Tabelle TBA_THESAURUS_ENTRIES verglichen wird. Die Anzahl des Auftretens eines Begriffes und die Wortanzahl im Text werden dazu benutzt, um die Relevanz des Begriffes in dem Dokument mit der „normalen“ Formel zu bestimmen. Überschreitet diese einen Schwellenwert, so wird er zur Weiterverarbeitung in ein Xml-Dokument geschrieben.

Funktionen automatKlassEingPost1Vektor & automatKlassEingPost2Vektor

Diese beiden Funktionen zur Klassifikation der Eingangspost (mit und ohne Bereichsvorgabe) unterscheiden sich von den Originalen in dem Maße, dass sie zur Berechnung der Relevanz die TF-IDF-Formel benutzen.

Funktion KlassVektor

Mit dieser Funktion wird wie in „Klass“ ein Text klassifiziert. Die Relevanzberechnung erfolgt über Vektoren, welche für die Anzahl des Auftretens der jeweiligen Thesaurusbegriffe benötigt werden. Anschließend wird die Berechnung analog zur Funktion „Klass“ durchgeführt.

Funktion	Variablen	Output	Kurzbeschreibung
automatKlass	Bereich, xmldoc	XmlDocument	Klassifikation von Webseiten
automatKlassEingPost1	Bereich	XmlDocument	Klassifikation der Eingangspost zu einem Bereich
automatKlassEingPost2		XmlDocument	Klassifikation der Eingangspost in verschiedene Bereiche
Klass	sText	XmlDocument	Klassifikation eines Textes
automatKlassVektor	Bereich, xmldoc	XmlDocument	Klassifikation von Webseiten mit Vektoren
automatKlassEingPost1Vektor	Bereich	XmlDocument	Klassifikation der Eingangspost zu einem Bereich mit Vektoren
automatKlassEingPost2Vektor		XmlDocument	Klassifikation der Eingangspost in verschiedene Bereiche mit Vektoren
KlassVektor	sText	XmlDocument	Klassifikation eines Textes mit Vektoren
automatKlassEingPost1Kosinus	Bereich	XmlDocument	Klassifikation der Eingangspost zu einem Bereich mit Kosinusmaß
Durchschnitt	Bereich	Dictionary	Durchschnitt eines Bereichs (z.B. von Rechnungen)
Anzahl_W	TempText	Integer	Anzahl der Wörter eines Strings
automatKlassError	Err, Zeile	Boolean	Verwaltung der Fehlermeldungen

Tab. 4.2: Funktionen des Webservice „Klassifikation“

Funktion automatKlassVektor

Diese Funktion klassifiziert analog zur der Funktion „automatKlass“ Webseiten. Der Unterschied zwischen den beiden Funktionen liegt bei der Berechnung der Relevanz von Teilseiten. In dieser Funktion wird die Relevanz mittels Vektoren berechnet. Analog zu den anderen Funktionen werden auch hier die Webseiten in ein Xml-Dokument zur Weiterverarbeitung geschrieben.

Funktion automatKlassEingPost1Kosinus

Bei dieser Funktion zur Klassifikation der Eingangspost mit Bereichsvorgabe wird das Kosinusmaß zur Relevanzberechnung verwendet.

4.4 Windows-Client

Um die Webservices **Thesaurus** und **Klassifikation** anwenden zu können, benötigte ich einen Client.

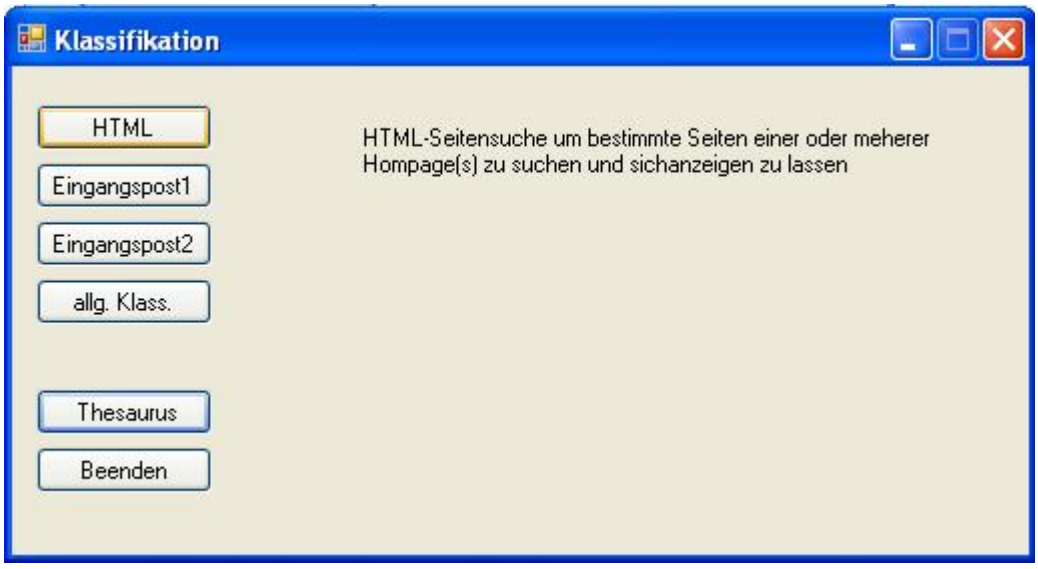

Abb. 4.2: Startbildschirm des Clients

Auf dem Startbildschirm des Clients ist es möglich, die verschiedenen Klassifikationen auszuwählen und anschließend zu benutzen. Es ist auch ein Button vorhanden, um den Thesaurus zu verwalten. Genauere Informationen über die verschiedenen Klassifikationsmethoden erscheinen beim Halten der Maus über den jeweiligen Button.

Beim Auswählen von „HTML“ erscheint ein neues Fenster, wie in **Abb. 4.3** zu sehen ist. Dort werden in der unteren Box die Webseiten verwaltet, die klassifiziert werden sollen. Die ausgewählten Webseiten werden in einer Xml-Datei gespeichert, so dass sie beim nächsten Start des Clients wieder geladen werden können. Es ist möglich, einzelne Seiten dynamisch hinzuzufügen sowie zu entfernen. In der oberen Box wird dann der Bereich ausgewählt, nach dem klassifiziert werden soll. Ist der Suchvorgang beendet, erscheint ein Explorer, wo auf der linken Seite die gefundenen Adressen aufgelistet werden, die sich mit einem Klick auf der rechten Seite öffnen. Diese Übersicht liefert alle Teilseiten der ausgewählten Webseiten, die dem Bereich zugeordnet wurden.

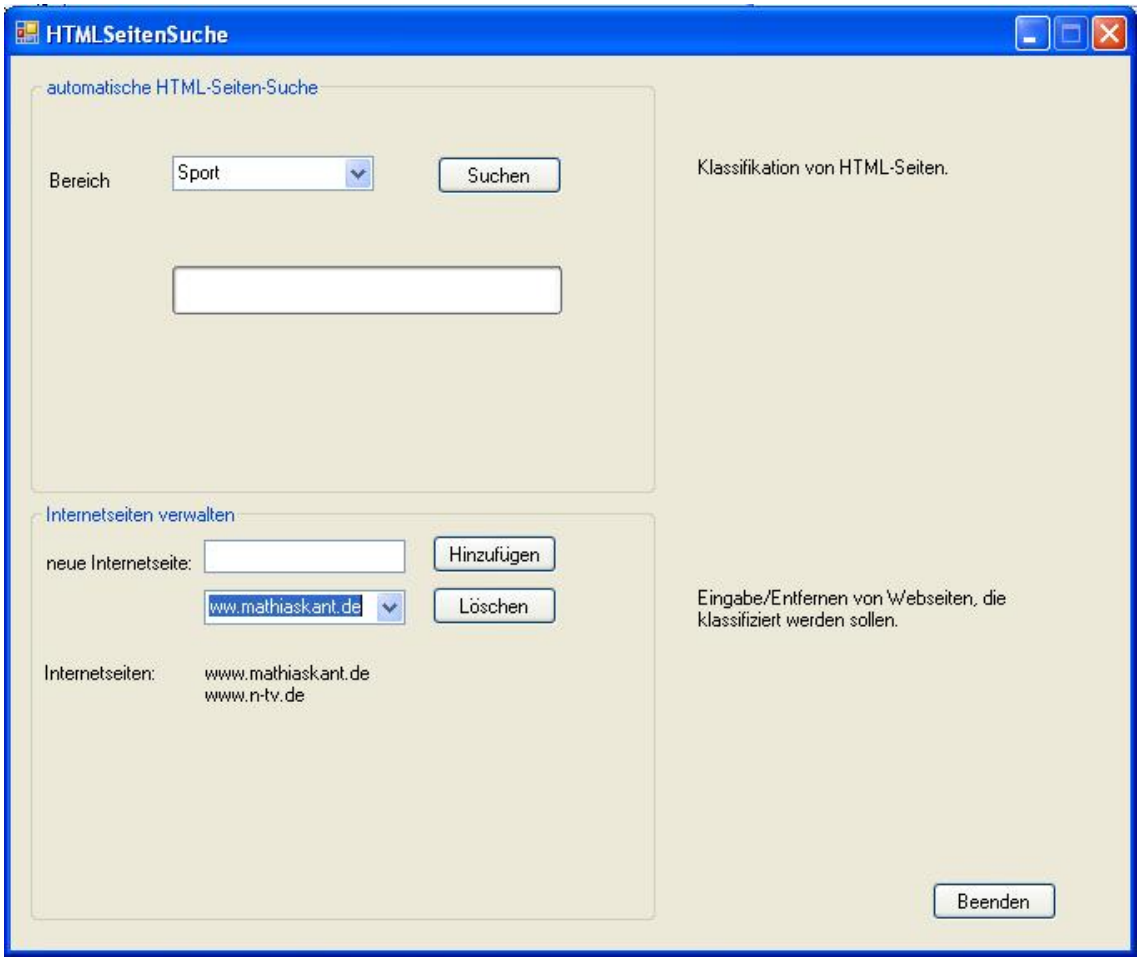

Abb. 4.3: HTML-Seitensuche

Wird „Eingangspost1“ im Startbildschirm ausgewählt, erscheint ein neues Fenster, in dem die Eingangspost nach ausgewählten Bereichen angezeigt werden kann. Hier kann dann ein Bereich bestimmt werden und anschließend die Klassifikation nach dem normalen Verfahren, dem Vektorverfahren oder dem Kosinusverfahren durchgeführt werden. Nach der Suche erscheint im unteren Feld eine Baumstruktur der Dokumente, die dem Bereich zugeordnet wurden. Mit einem Doppelklick auf den jeweiligen Eintrag

wird dieser geöffnet. Unter der Baumstruktur wird die Anzahl der gefundenen Dokumente angezeigt.

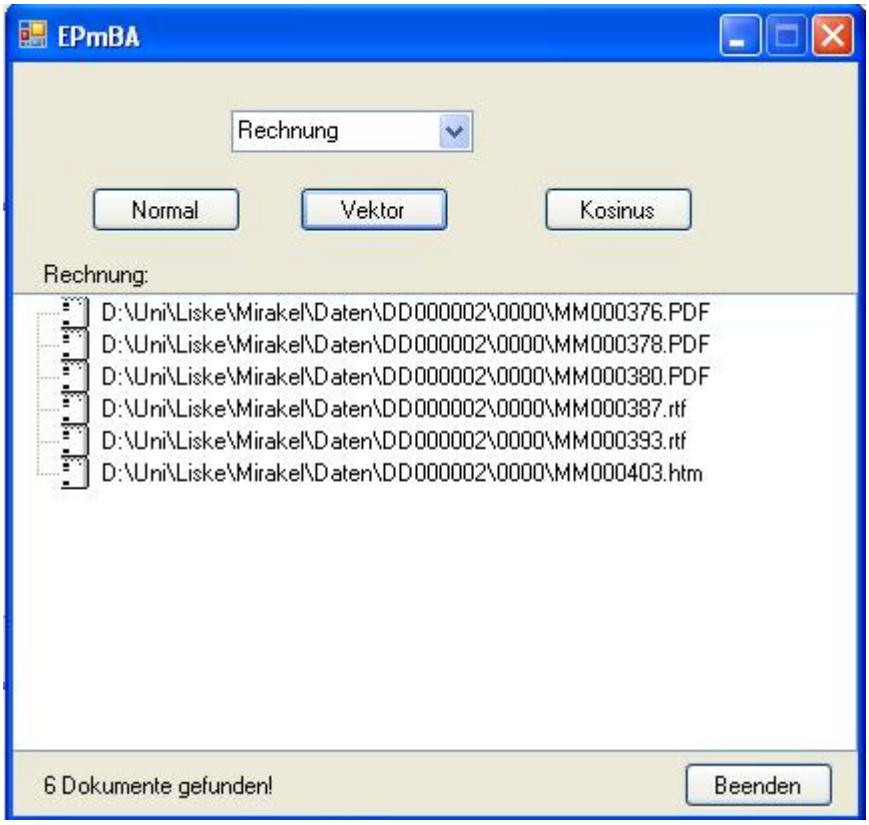

Abb. 4.4: Eingangspostsuche mit Bereichsangabe

In dem Fenster zu „Eingangspost2“ gibt es die Möglichkeit die gesamte Eingangspost zu klassifizieren. Dies kann durch das normale oder das vektorbasierte Verfahren geschehen. Nach dem Vorgang erscheint eine Baumstruktur der Bereiche (z.B. „Rechnungen“ oder „Aufträge“), in denen Dokumente, welche eine Hierarchiestufe tiefer angezeigt werden, gefunden wurden.

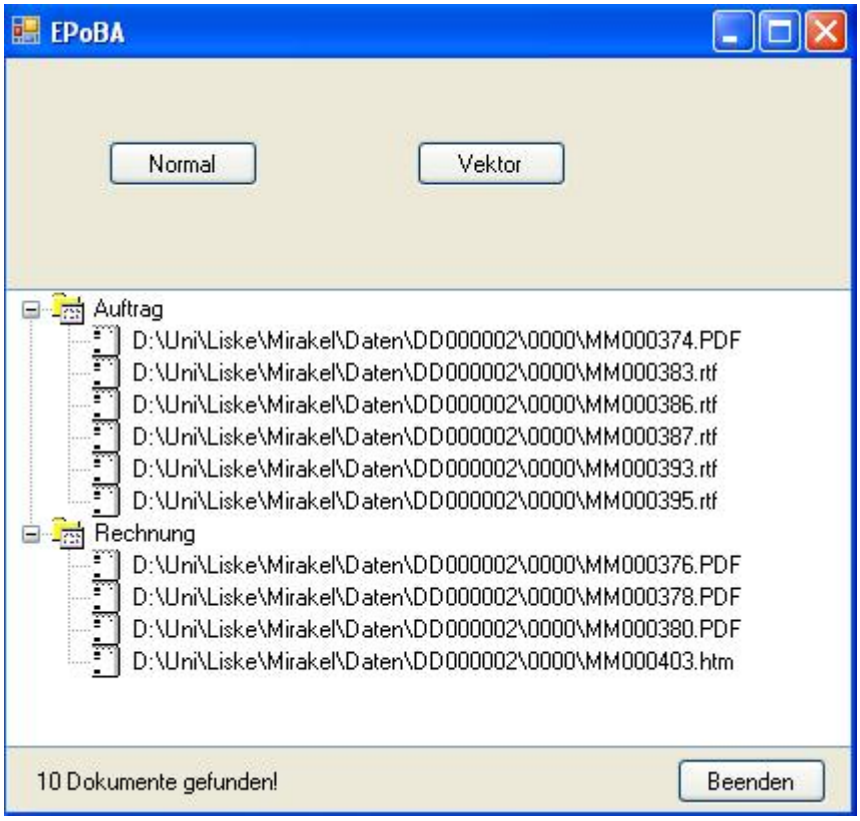

Abb. 4.5: Eingangspostsuche ohne Bereichsvorgabe

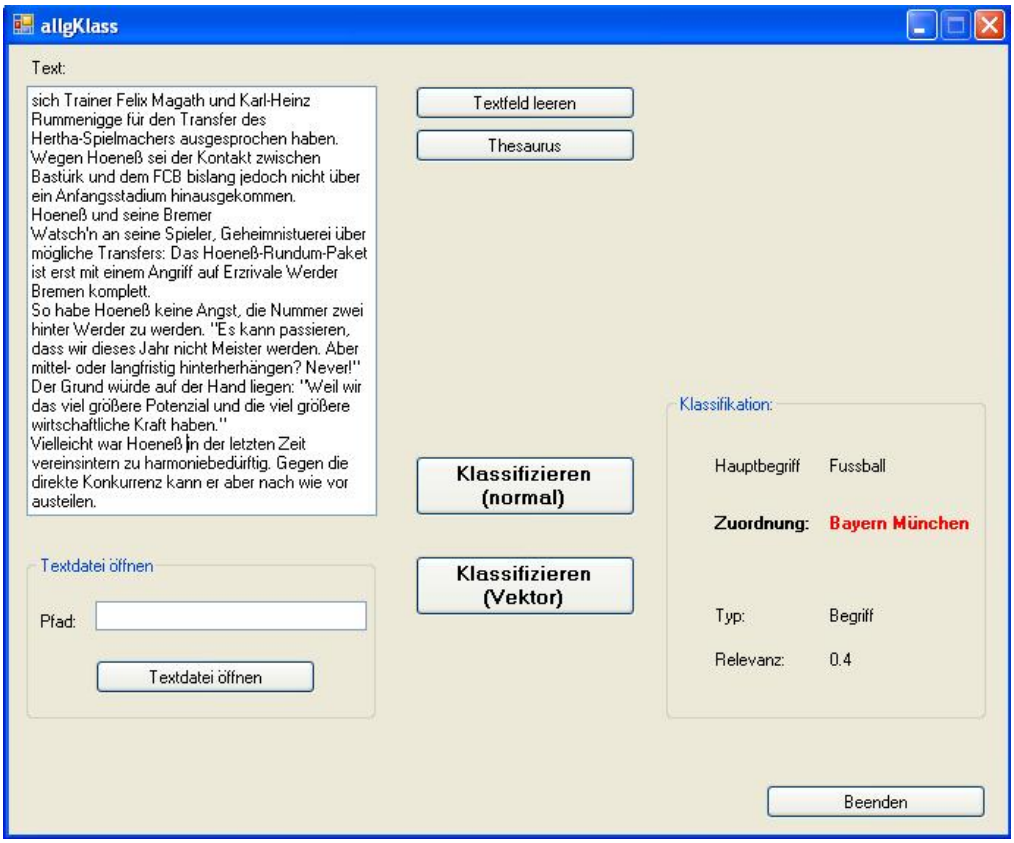

Abb. 4.6: Allgemeine Klassifikation eines Textes

Wird „allg. Klass.“ im Client ausgewählt, so besteht die Möglichkeit, einen eingegebenen Text oder einen Text aus einer Text-Datei zu klassifizieren. Hierbei wird nach der „normalen“ Methode klassifiziert. Bei der Relevanzberechnung ist es möglich, diese normal (Funktion Klass) oder mit Hilfe von Vektoren (Funktion KlassVektor) durchführen zu lassen. Auf der rechten Seite wird anschließend das Ergebnis der Klassifikation angezeigt. Der Nutzer kann durch Klicken auf den Button „Textfeld leeren“ selbiges tun.

Bei dem Fenster „Thesaurus“ wird dieser verwaltet. Wie in **Abb. 4.7** zu sehen ist, wird es ermöglicht, die Struktur des Thesaurus anzuschauen. Dabei können neue Haupt-, Ober- und Unterbegriffe eingefügt sowie editiert und gelöscht werden. Wenn z.B. „Formel 1“ gelöscht wird, dann werden auch automatisch alle Unterbegriffe wie z.B. „Ferrari“ oder „Schumacher“ zu diesem Oberbegriff gelöscht.

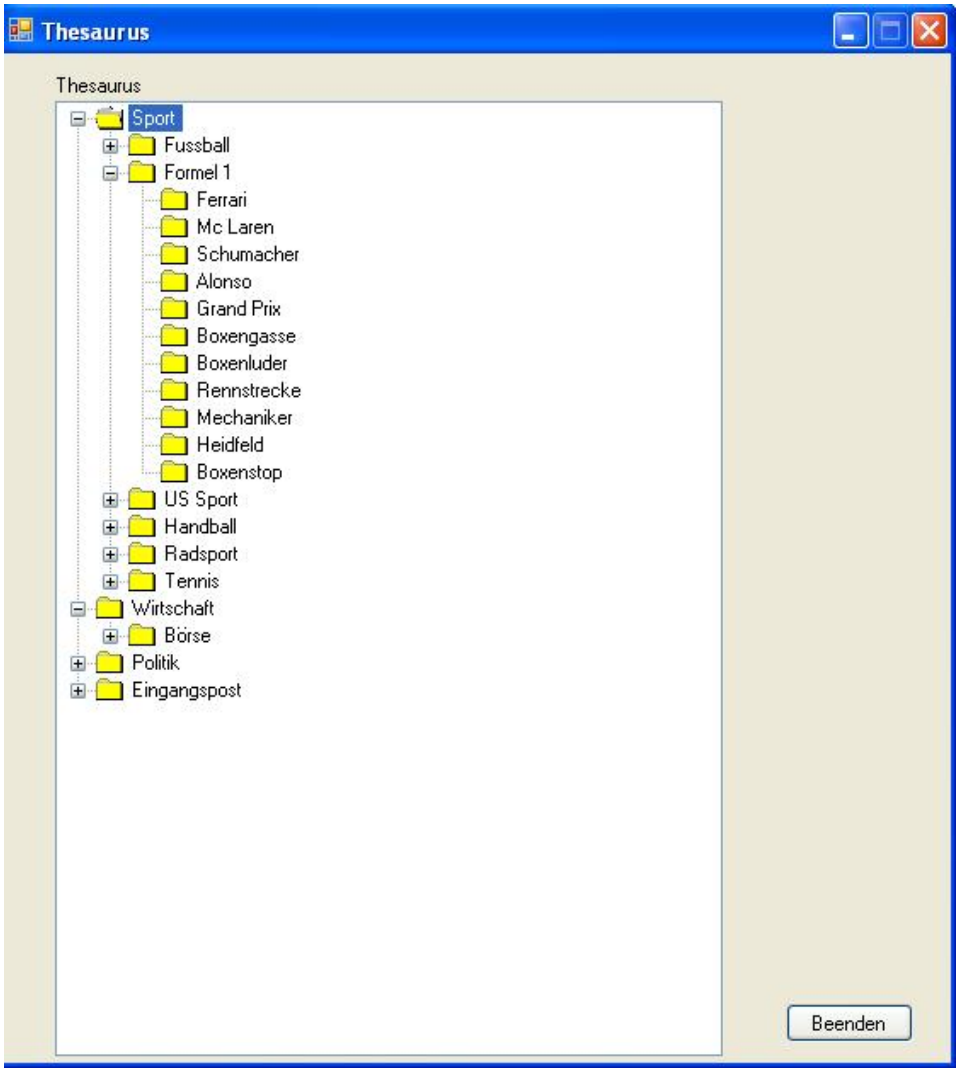

Abb. 4.7: Thesaurus

5 Anwendungsbeispiel – HTML-Seitensuche

Um die Funktionsweise meines Clients ein wenig anschaulicher zu machen, folgt als Anwendungsbeispiel die HTML-Seitensuche.

5.1 Auswahl der Webseiten

Am Anfang der Klassifikation der Webseiten muss der Anwender die Webseiten wählen, in denen er suchen will. Zur Veranschaulichung habe ich die Webseiten www.n-tv.de, www.mdr.de und www.n24.de, wie in **Abb. 5.1** zu sehen ist, gewählt.

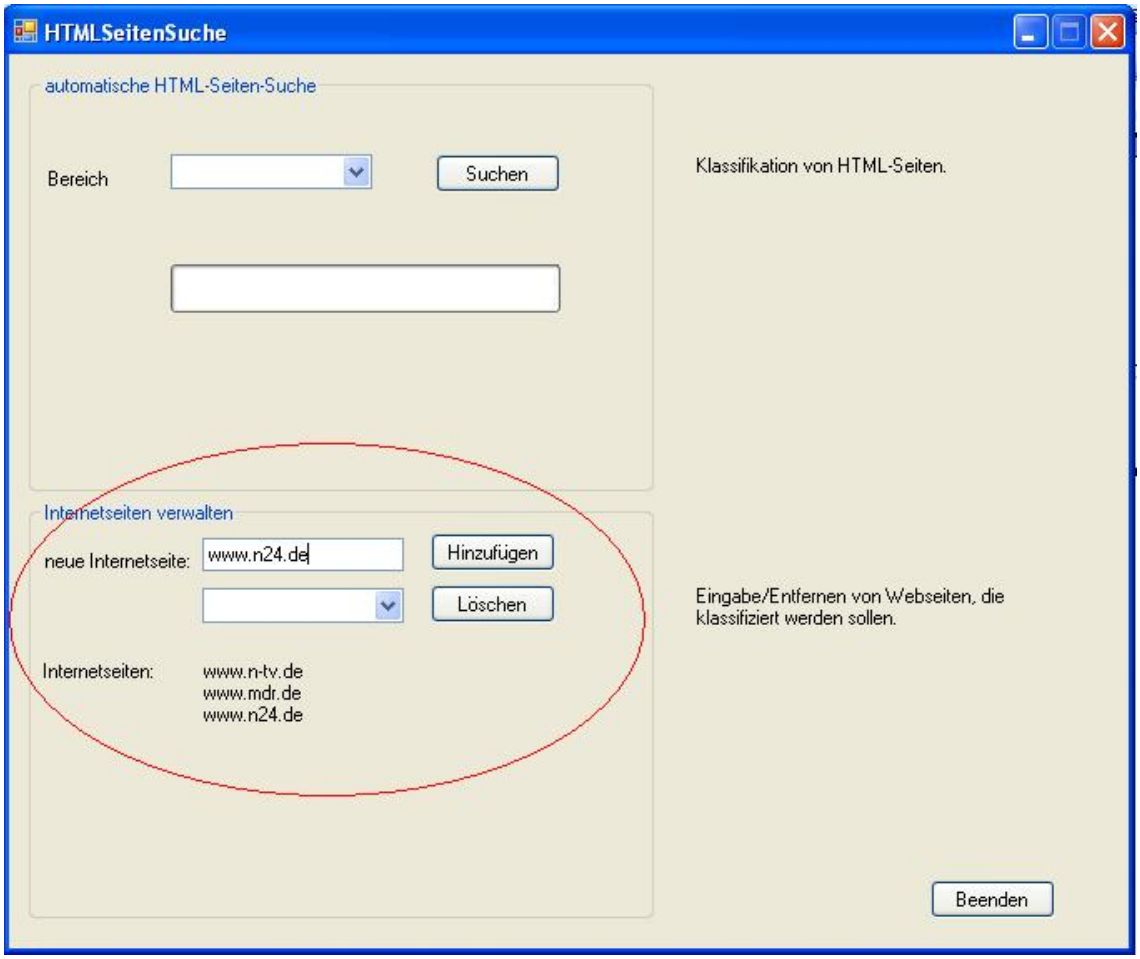

Abb. 5.1: Auswahl der Webseiten

5.2 Auswahl des Bereichs und Anzeige des Ergebnisses

Anschließend muss der Bereich ausgewählt werden, nach dem klassifiziert werden soll. In meinem Beispiel habe ich mich für den Bereich „Sport“ entschieden, was bedeutet, dass der Client alle Teilseiten der gewählten Webseiten sucht, die er dem Bereich zuordnet. Durch Klicken auf „Suchen“ wird der Vorgang gestartet, wobei der Balken anzeigt, dass die Suche noch läuft.

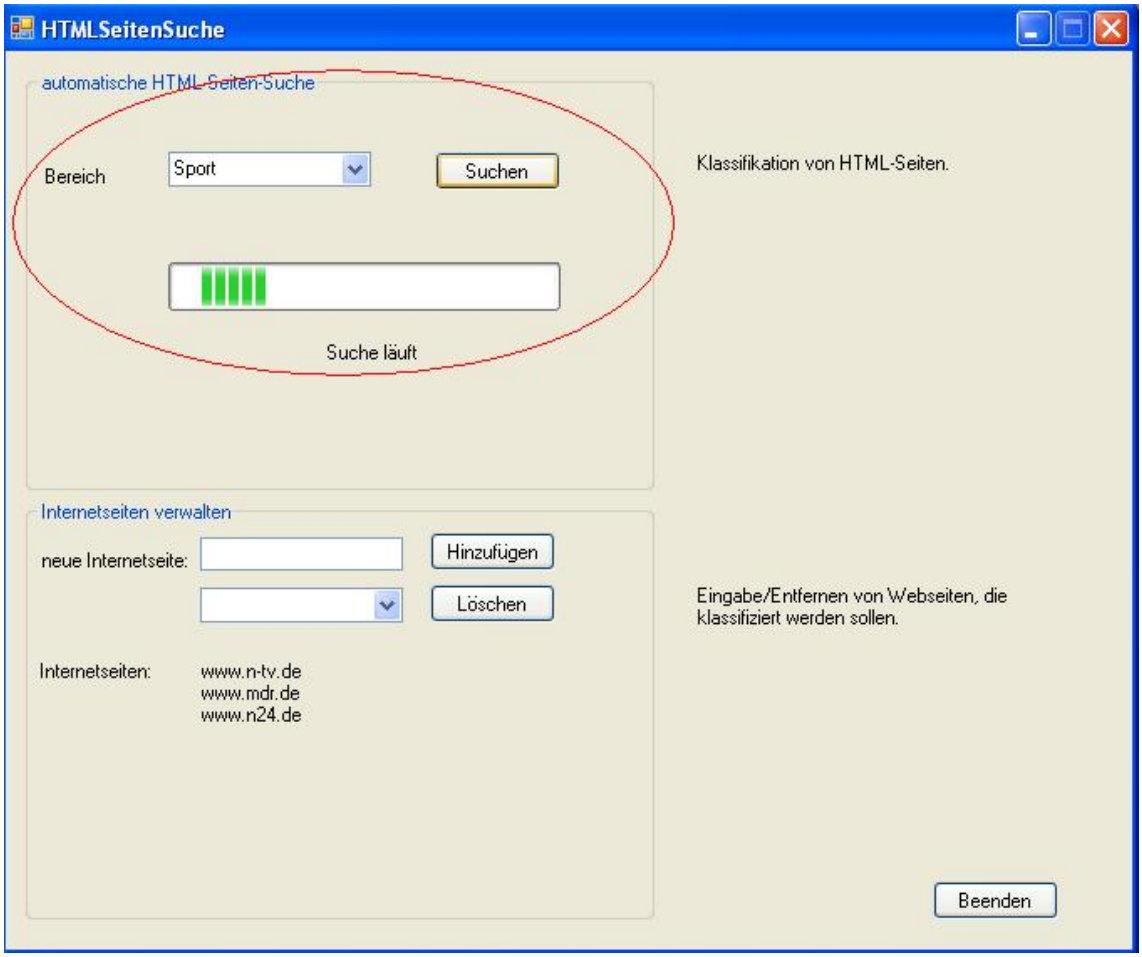

Abb. 5.2: Bereichsauswahl & Start der Klassifikation

Nun werden die Webseiten in Mirakel gespeichert und auf dieser Grundlage werden dann die Teilseiten klassifiziert. Ist die Klassifikation abgeschlossen, erscheint ein Button „Ergebnis“, wodurch sich ein Explorer öffnen lässt (**Abb. 5.3**). Dort sind auf der linken Seite die Teilseiten aufgelistet, die dem Bereich zugeordnet wurden, und durch einen Klick erscheinen diese auf der rechten Seite. Der Anwender braucht sich nun nicht durch alle drei Webseiten klicken, sondern hat eine kompakte Darstellung aller ihn interessierenden Seiten.

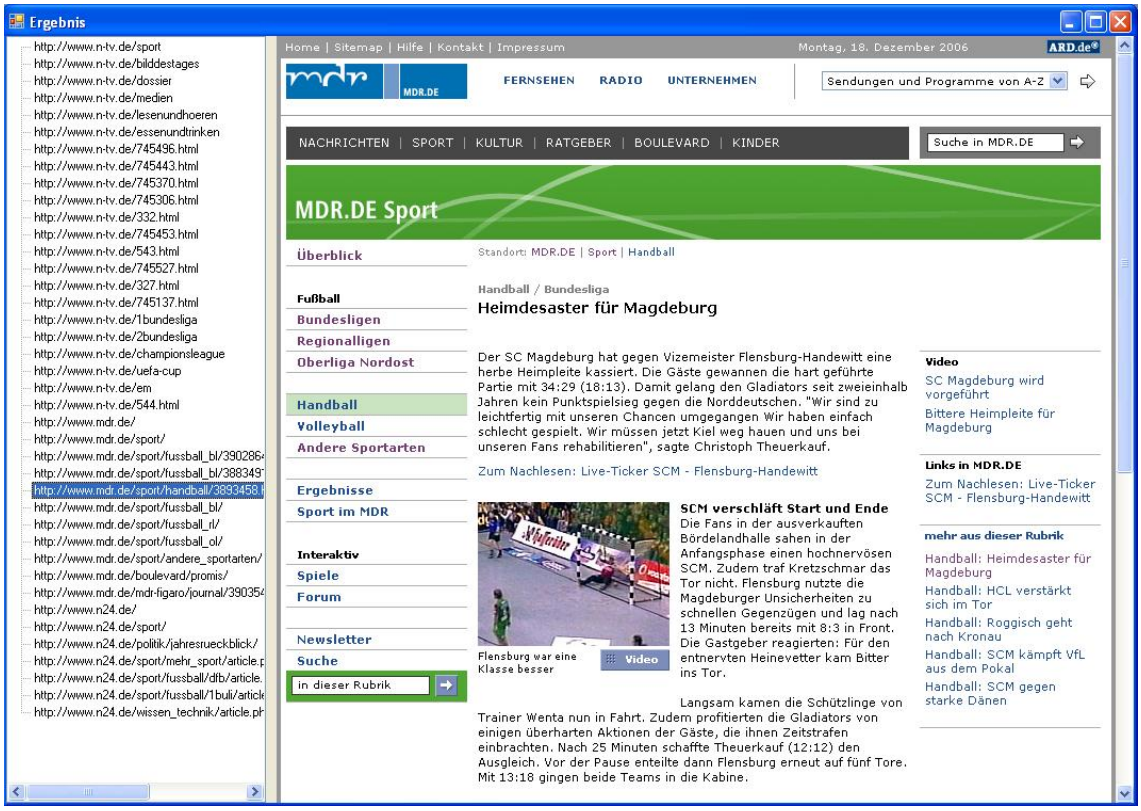

Abb. 5.3: Ergebnisanzeige

6 Zusammenfassung und Ausblick

In meiner Studienarbeit hatte ich das Ziel, einen Webservice zur Klassifikation und einen dazugehörenden Client zu entwerfen und zu implementieren. Im Laufe der Arbeit fand ich verschiedene Formeln wie das Kosinus-Maß oder die TF-IDF-Formel um Dokumente zu klassifizieren. Auf der Basis dieser implementierte ich im Folgenden verschiedene Funktionen zur Klassifikation. Anschließend schrieb ich ein Client, mit dem es möglich ist, alle Methoden benutzen und zusätzlich den Thesaurus betrachten und bearbeiten zu können. Da die Funktionen nur von außen auf Mirakel zugreifen, wäre es ein Ziel für die Zukunft, dass eine der Funktionen in das Produkt Mirakel integriert wird und die bestehenden Dokumente in Ordner automatisch klassifiziert werden.

Da das Speichern der Webseiten in Mirakel ziemlich zeitaufwändig ist, sollte diese Funktion in der Nacht ausgeführt werden, damit der Anwender am nächsten Morgen die Webseiten anschauen kann, die ihn interessieren könnten. Durch mehrmaliges Auftreten der Begriffe in den Menüs ist es schwierig, Webseiten zu klassifizieren. Der Begriff „Sport“ ist beispielsweise auf jeder Teilseite vorhanden, da er häufig im Menü steht.

In meiner Implementierung der Klassifikation der Eingangspost werden auch noch Dokumente in Bereiche eingeordnet, die nicht dorthin gehören. Aus diesem Grund sehe ich in meiner Implementierung noch Verbesserungspotential, damit die Quote der richtigen Ergebnisse erhöht wird.

Bei der allgemeinen Klassifikation ist bis jetzt nur eine einfache Berechnung der Relevanz vorhanden. Sie geht momentan nur von den beiden Größen „Anzahl des Auftretens“ und „Wörteranzahl im Text“ aus. Zur Verbesserung könnte noch die Wertigkeit der Thesaurusbegriffe, der als Attribut dem Begriff angehängt ist, zur Berechnung hinzugenommen werden.

Literaturverzeichnis

Brückner, T.; Dambeck, H. (2003): Sortierautomaten Grundlagen der Textklassifizierung, In: c´t, Heft 19, S. 192-196.

Freeman, A.; Jones, A.(2003): Microsoft .NET XML Webdienste Schritt für Schritt. Unterschleißheim, übersetzt von Kronast, L; Haselier, R. G.; Fahnenstich, K.

Heuer, A.; Saake, G.(2000): Datenbanken: Konzepte und Sprachen. 2. Aufl., Bonn.

Hoffmann, R. (2002): Entwicklung einer benutzerunterstützten automatisierten Klassifikation von Web-Dokumenten, Graz.

o.V.: SQL. http://de.wikipedia.org/wiki/SQL. 5.September 2006.

o.V.: Mirakel, http://www.liske.de, 5.September 2006.

o.V.: Grundbegriffe Thesaurus nach DIN 1463, http://www.bui.haw-hamburg.de/pers/ulrike.spree/grundthes/grundthes.htm, 6.Januar 2007

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Studienarbeit selbstständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solch kenntlich gemacht.

Magdeburg, den