Diplomarbeit

Semantische Anreicherung von Suchanfragen auf Basis von Topic Maps.

vorgelegt von
Andreas Christensen
am 3. Juni 2005

Studiengang Softwaretechnik

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Gunter Klemke

Fachbereich Elektrotechnik und Informatik
Department of Electrical Engineering and Computer Science

Semantische Anreicherung von Suchanfragen auf Basis von Topic Maps.

Stichwörter Information Retrieval, Semantic Web, Topic Maps, Suchmaschinen.

Zusammenfassung

In dieser Arbeit wird das System *TopicSEEK* entwickelt und vorgestellt, welches die semantische Anreicherung von Anfragen an Suchmaschinen unterstützt. Basis dafür sind Topic Maps, die zwar eine schwache, aber noch ausreichende semantische Ausdrucksstärke haben. Das System wurde mit einer angemessenen Fachbegrifflichkeit aus der Informatik ausgestattet. Das gesamte System besteht aus der Topic Map, einer Client-Applikation, sowie der Anwendungsschicht. Darüber hinaus wird eine geeignete Eingabe-Sprache angegeben. Als Suchmaschine wird Google an das System angebunden und für die Suche benutzt. Google liefert ein Java-API, weshalb die Implementierung von TopicSEEK ebenfalls in Java durchgeführt wurde. Als Engine, die die Topic Map abfragt, wurde *k42* ausgewählt, das ebenfalls ein Java-API zur Verfügung stellt.
Der Prototyp wurde erfolgreich realisiert. Die Testsergebnisse zeigen, dass das System dann funktioniert, wenn Begriffe innerhalb des semantischen Modells sinnvoll in Zusammenhang gebracht werden.

Semantic enrichment of search requests by using topic maps.

Keywords Information Retrieval, Semantic Web, Topic Maps, Search Engines.

Abstract

This paper introduces and documents the development of *TopicSEEK*, a system for semantic enrichment of requests for search engines, based on topic maps. Although they have weak semantic power, it is still sufficient. TopicSEEK has been supplied with appropriate expressions out of information technology. The complete system consists of a topic map, the client and the application layer. Furthermore there will be given an input grammar. As systems search engine, Google will be used for information retrieval. Google supplies a Java-API, wherefore TopicSEEK is developed in Java as well. Another Java-API is used to access *k42*, the chosen topic map engine.
The prototype has been realized successfully. As test results indicate, the system works well, if terms are related meaningfully within the semantic context.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1. Einleitung

Motivation

Will man im Internet mit Hilfe von Suchmaschinen an Informationen gelangen, wird man häufig enttäuscht, da man als Ergebnis nicht ausschließlich Seiten mit den gesuchten Inhalten erhält. Die relevanten Dokumente bleiben einem so mitunter verborgen. Beispielsweise bekommt man bei *Google* für das Stichwort *Java* diverse Seiten, die sich mit der Programmiersprache *Java* beschäftigen. Wollte man aber Informationen über *Java* als Urlaubsparadies, so sind viele der Antworten überflüssig und in diesem Kontext sogar störend.
Betrachtet man die Suchmaschinen inklusive ihrer Algorithmen als gegeben, so bleibt noch, die Eingabe zu untersuchen und, wenn möglich, zu verbessern. Benutzt man eine Konjunktion und ersetzt den Eingabestring durch *Java AND Urlaub*, so fallen viele unbrauchbare Ergebnisse weg und man erhält bereits ganz oben auf der Liste sinnvolle Antworten[1]. Der Einsatz von Topic Maps und einer entsprechenden Client-Anwendung kann solche Ersetzungen im Rahmen einer wohl definierten Fachlichkeit automatisieren.

Das Problem

Das allgemeine Problem könnte man durch folgende simple Frage formulieren: Wie findet man in der großen Menge von Informationen diejenigen heraus, die man haben möchte?
Das damit zusammenhängende informatische Problem ergibt sich aus der Kommunikationsfähigkeit von Mensch zu (stichwortbasierter) Suchmaschine und ihrer gleichzeitigen, natürlichen Inkompatibilität. Was beim Menschen am Ende des Denkprozesses steht und in aller Regel ausgedrückt wird, bewegt sich auf semantischer Ebene. Für die Maschine sind es aber nur syntaktische Konstrukte, die auf klar definierte Weise abgearbeitet werden.
Wie kann man also Semantik in maschinenverwertbare Form umsetzen?

[1] Zuletzt getestet am 10.03.2005.

Lösungsansätze

Ein klassischer Ansatz zur Lösung dieses Dilemmas besteht darin, Methoden zu formulieren, die die Relevanz der Dokumente bezüglich der jeweiligen Anfrage besser und genauer bestimmen. Das geschieht z.B., wenn man Texte annotiert. Dadurch erreicht man, dass Dokumente semantisch angereichert werden.
Ein anderer Lösungsansatz beschäftigt sich schon zuvor mit der Anfrage. Es ist leicht einsehbar, dass ein verbesserter Anfragestring eine genauere Antwortmenge bewirkt. Man versucht also, einen Teil der Semantik bereits vor der Anfrage an das Suchsystem syntaktisch darzustellen. Ob dieser Vorgang sinnvoll automatisiert werden kann, soll in diesem Papier untersucht werden.
Eine dritte Möglichkeit ist, die Menge der verfügbaren Daten gering zu halten. *Google Scholar* beispielsweise schränkt die Suchmenge rapide ein, so dass sich die Relevanz-Wahrscheinlichkeit für gefundene Objekte bereits im Vorwege erhöht.

Zielsetzung

Es ist u.a. die semantische Anreicherung von Suchanfragen auf Basis von Topic Maps zu untersuchen. Dabei soll eine Aussage über die Eignung von Topic Maps getroffen werden. Können Topic Maps die Qualität von Suchanfragen verbessern? Das impliziert natürlich auch die Frage, ob Topic Maps eine ausreichende, semantische Ausdrucksstärke haben.
Die Analyse wird den zugehörigen Kontext untersuchen und eine fundierte Basis schaffen, um ein prototypisches System namens *TopicSEEK* zu entwickeln. Zu diesem System gehören die folgenden, zu entwickelnden Komponenten: die Topic Map selbst, die Client-Anwendung und der Programmteil, der die Anwendungsschicht darstellt.
Dabei soll der Client dem Benutzer die Eingabe der Suchanfrage ermöglichen und diese dann zur Verarbeitung weiterleiten. Das Ergebnis muss angezeigt und durch den Benutzer verwertet werden können (siehe auch Abbildung 1.1).

Darüber hinaus kommt eine bereits existierende Topic Map Engine zum Einsatz. Die Topic Map soll mit einer angemessenen Fachbegrifflichkeit ausgestattet und für den Eingabestring eine geeignete Sprache gefunden werden.

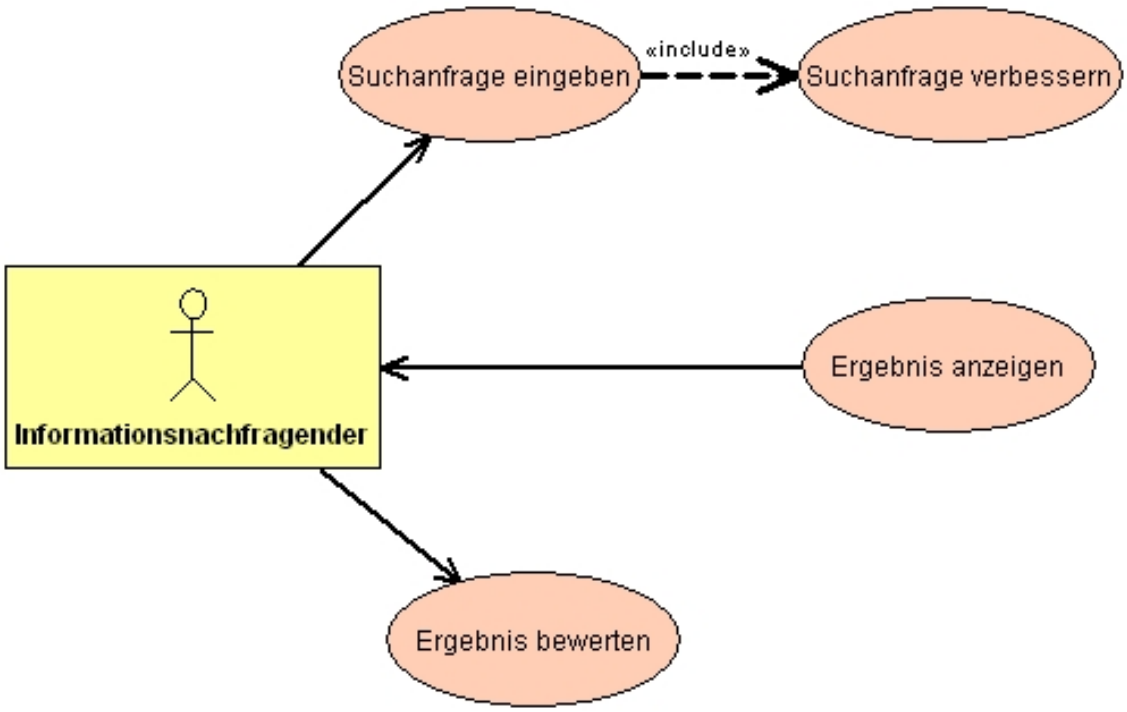

Abbildung 1.1.: Anwendungsfall Informationsnachfrage

Aufbau der Arbeit

Im ersten Teil der Arbeit werden zunächst Systeme untersucht, die auf das informatische Problem mit verschiedenen Lösungsansätzen reagieren. Neben WebFountain, OntoSeek und SHOE wird auch die bereits erwähnte Variante von *Google* namens *Google Scholar* analysiert.
Danach werden im Bereich *Information Retrieval vage Anfragen* und *unsicheres Wissen* vorgestellt. Darüber hinaus wird in die Evaluierung eingeführt, um die Pertinenz als Test-Methode auszuwählen, die in dieser Arbeit genutzt werden soll.
Darauf folgen Suchmaschinen, ihre interne Arbeitsweise, Metasuchmaschinen als Variante und Abfragesprachen. Umsetzungen mit zwei Fallstudien bedeutender Suchmaschinen beschließen dieses Kapitel.
Eine Untersuchung von Topic Maps, die zugrunde liegenden Konzepte, sowie die relevanten Aspekte beim Einsatz von Topic Maps, werden im darauf folgenden Kapitel aufgearbeitet. Dabei wird auch ein näherer Blick auf die Entwicklungsphasen geworfen, die im zweiten Teil der Arbeit relevant werden.
Das allgemeine Szenario, die daran beteiligten Akteure, Aktionen und Komponenten, sowie deren Zusammenwirken werden am Ende des ersten Teils vorgestellt.

Der zweite Teil umfasst die Synthese. Als erstes wird der Ausschnitt des allgemeinen Szenarios festgelegt, der durch das System abgebildet werden soll.
Danach folgen das Design und wichtige Anmerkungen zur Implementierung der Anwendung, die zum Gesamtsystem *TopicSEEK* gehört.
Ein Teil der Umsetzung umfasst die Grammatik, die festlegt, welche Eingaben als gültig akzeptiert werden sollen.
Nachdem die notwendigen Entwicklungsphasen der Topic Map, inklusive ihres Designs, aufgeführt wurden, beschließt die Darstellung der Tests und auch der entsprechenden Ergebnisse den zweiten Teil.

Der obligatorische Abschnitt *Zusammenfassung und Ausblick*, in dem auch eine kritische Einschätzung dieser Arbeit und das Fazit vorkommen, rundet dieses Papier ab.

2. Analyse

2.1. Vergleichbare Arbeiten

Bevor eine nähere Untersuchung der notwendigen Technologien und eine Erarbeitung anderer Grundlagen für diese Arbeit erfolgt, soll ein Blick auf weitere Projekte bzw. Papiere geworfen werden. Dabei soll von Interesse sein, ob und wie diese das vorliegende informatische Problem gelöst haben. Zum einen wird damit abgesichert, dass nicht bereits jemand anderes das Problem auf dieselbe Weise gelöst hat, zum anderen können eventuell auftretende Probleme bei der Lösung dieser Aufgabe rechtzeitig erkannt werden. Gegebenenfalls können auch interessante Ansätze in diese Arbeit einfließen.
Die Recherche führte u.a. zu folgenden Projekten:

- WebFountain
- Google Scholar
- OntoSeek
- SHOE

WebFountain

Im Internet und auch innerhalb großer Intranets stehen riesige Datenmengen zur Verfügung. Vieles ist dabei unstrukturiert[1]. Der semantische Ausdruck und somit der Nutzen dieser Daten kann durch eine Strukturmehrung deutlich erhöht werden. Das bezieht sich auf große Bereiche, wie dem Sammeln, dem Speichern und der Analyse von großen Datenbeständen. WebFountain ist ein Projekt des *IBM Almaden Research Centers*[2], das sich genau darum

[1] Der Grad der Unstrukturiertheit kann dabei natürlich abweichen.

[2]
> Over 300 researchers worldwide, about 200 patents, and four years of research have contributed to this new technology.

Dieses Zitat von der WebFountain Homepage verdeutlicht, welchen erheblichen Stellenwert das Projekt einnimmt.

bemüht. Basis dieses Projektes bildet eine offene, erweiterbare Plattform, die eine Sammlung von Forschungstechnologien zur Verfügung stellt. Dabei liegt der Fokus auf der Entdeckung von Beziehungen, Strukturen und Entwicklungen von Daten ([Cen05]). Nach eigenen Angaben der Projektbetreiber besteht der Kern im Wesentlichen aus drei Teilen:

- *Plattform*
  Sie vereinigt *Business Partner* Technologien mit *Third-Party* Inhalten [Cen05]. Dadurch kann man dem Endkunden fertige Lösungen anbieten. Es wird hierbei darauf geachtet, nur solche Programme einzusetzen, die sich an offene, skalierbare Standards halten.
- *Daten*
  Der Kunde von WebFountain hat darüber hinaus Zugriff auf mehrere Terabyte gespeicherter Daten, die sich aus verschiedensten Quellen zusammensetzen, wie z.B. alle mögliche Daten aus dem Internet, Foren, Zeitungen, Magazinen usw.
- *Multi-disziplinäre Text Analyse*
  Unterschiedliche Arten, Texte zu analysieren, wie natürlichsprachliche Verarbeitung, Statistiken, Mustererkennung u.a. werden durch eine integrierte Plattform unterstützt, die WebFountain ebenfalls zur Verfügung stellt.

Die Technologie im Bereich der Text Analyse, die auch bei WebFountain eingesetzt wird, basiert auf IBM's *UIMA*-Projekt. Da diese Analysen einen der Lösungswege des, dieser Arbeit zugrunde liegenden, informatischen Problems darstellen, soll UIMA ebenfalls Beachtung finden.

UIMA - The Unstructured Information Management Architecture Project Das Projekt findet seinen Ursprung in der Tatsache, dass die Suche in unstrukturierten Daten unergiebig ist. Zur Lösung sollen innerhalb der Daten, per Analyse, Dinge bestimmt werden, die von Interesse sind, damit Suchmaschinen dann relevante Resultate erzielen können. UIMA geht dabei von folgendem Selbstverständnis aus:

> The bridge from the unstructured world to the structured is analysis. IBM's Unstructured Information Management Architecture (UIMA) is an architecture and framework that helps you build that bridge. (Von der Projekthomepage: [Res05])

Wie UIMA sich den Weg vorstellt, der unstrukturierte Daten bis in die Hände des Suchenden führt, ist Abbildung 2.1 zu entnehmen[3]. Im Wesentlichen werden unstrukturierte Daten durch Hilfsmittel, wie Indizes, Taxonomien usw. strukturiert bzw. semantisch angereichert, um einen gezielten Zugriff zu ermöglichen.

[3] Entnommen aus [Res05].

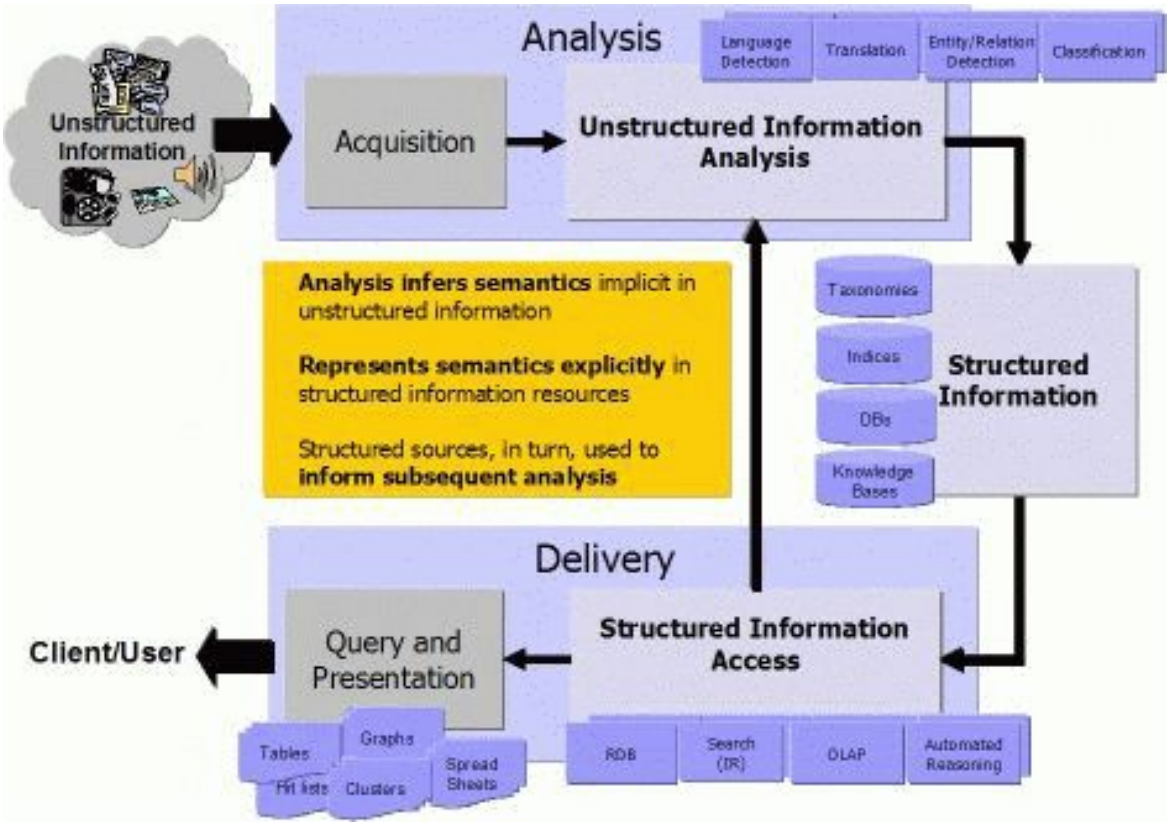

Abbildung 2.1.: UIMA Architekturbeschreibung

Google Scholar

Google Scholar ist eine weitere Suchmaschine aus dem Hause *Google*. Sie wird an dieser Stelle beschrieben und nicht in Kapitel 2.3, da für diese Arbeit insbesondere von Interesse ist, dass Google Scholar den Lösungsraum semantisch reduziert. Google Scholar[4] benutzt ein deutliches Selektionsverfahren, das strengen Kriterien unterliegt. Es werden also nicht alle Dokumente annotiert und aufgenommen[5]. Google Scholar stellt nur solche Dokumente und andere Antworten zur Verfügung, die im Wesentlichen folgenden Bereichen zuzuordnen sind:

- wissenschaftliche Literatur in Form von Artikeln, Büchern oder anderweitigen durchgesehenen Papieren
- abstrakte und technische Berichte aus allen Bereichen der Forschung

[4] URL: `www.scholar.google.com`.

[5] Mathematisch betrachtet könnte man allerdings unterstellen, dass die Relevanz nicht aufgenommener Elemente auf 0 gesetzt wird und somit auch nur ein Spezialfall des ersten Lösungsansatzes ist.

Auch ist dabei wichtig, dass sowohl online-, als auch offline verfügbare Dokumente berücksichtigt werden. Man erhält deshalb zusätzliche Treffer in Form von Verweisen auf Offline-Material, das man sich dann gegebenenfalls beschaffen kann.
Ansonsten stellt Google Scholar ähnliche Funktionalität zur Verfügung, wie auch Google selbst. Intern werden die Dokumente von Google Scholar ebenfalls annotiert und ihrer Relevanz entsprechend in eine sinnvolle Reihenfolge gebracht.
Was bei Google Scholar bislang noch fehlt, ist die Einschränkung auf bestimmte Fachlichkeiten. Zurzeit bekommt man sowohl Ergebnisse aus der Medizin, der Informatik, der BWL, Biologie und anderen Fachrichtungen[6], sofern sie der Anfrage entsprechen.

OntoSeek + SHOE

Zum Abschluss sollen noch kurz zwei weitere Projekte aufgeführt werden, die interessante Teilaspekte beinhalten. Beide setzen zur Lösung Ontologien ein[7].

OntoSeek Der Ansatz, bereits die Anfrage zu verbessern, wird von OntoSeek verfolgt. Es analysiert die Eingabe und versucht Konzepte zu finden, deren Bedeutung mit dem Eingabebegriff verwandt ist. Dann wird die Anfrage mittels Ersetzung präzisiert.
OntoSeek wurde von National Research Council, Ladseb-CNR u.a. in Gemeinschaftsarbeit entwickelt, um eine inhaltsbasierte Suche in *Produktkatalogen* und *Yellow Pages* zu ermöglichen (Näheres dazu bei [GMV05]).

SHOE Einen weiteren wichtigen Aspekt berücksichtigt das Projekt SHOE[8] von der Universität Maryland[9]. SHOE ist eine *knowledge representation language*, mit der man Webseiten mit Semantik anreichern kann [Hef05] (Daten können also bereits im Zuge der Erstellung besser formuliert werden). SHOE setzt dazu unterschiedliche Ontologien ein. Bei der Anreicherung arbeitet man mit spezifischen Metatags, mit denen Konzepte eines Dokumentes beschrieben werden. Ein SHOE-Webcrawler kann dann solche Dokumente in einer Wissensdatenbank ablegen. Über ein Anfragetool kann schließlich auf die Datenbank zugegriffen werden.

[6] Innerhalb von Google Scholar gibt es zusätzlich noch ein kleines Pilotprojekt, dass Zugriffs-Links namhafter Institute und Universitäten ausgibt, die normalerweise nicht öffentlich gemacht werden. Diese Links sind für Studenten, Professoren und andere legitimierte Personen gedacht, die Zugriffsrechte besitzen. Bis zu drei Institute können über Checkboxen ausgewählt werden. Näheres dazu unter [Sch05].

[7] Darüber hinaus gibt es mit *GETESS* und *SemanticMiner* zwei weitere, auf Ontologien basierende, Suchsysteme ([Nag03]), die für diese Arbeit jedoch keine weitere Bedeutung haben.

[8] SHOE (Simple HTML Ontology Extensions).

[9] Wie [HHLZ05] zu entnehmen ist, wird die SHOE Homepage nicht weiter aktualisiert. Die gesamte Arbeit der Universität Maryland im Bereich *web ontologies* findet man bei `http://www.mindswap.org/`.

Des Weiteren ist an SHOE interessant, dass das Projekt nicht nur, je nach Wissensgebiet, verschiedene Ontologien zur Verfügung stellt, sondern dem Benutzer darüber hinaus gestattet, diese auch zu erweitern.

Schlussbemerkungen

Nachfolgende Bemerkungen beschließen dieses Kapitel:

- Ein Projekt, das versucht, den Anfragestring mit Topic Maps semantisch ausdruckstärker zu formulieren, konnte nicht ermittelt werden[10]. Einen ganz ähnlichen Ansatz verfolgte allerdings die Diplomarbeit [Nag03], die anstatt Topic Maps Ontologien einsetzt[11]. Den gleichen Ansatz verfolgt auch OntoSeek, allerdings für Produktkataloge und Yellow Pages.

- Es existieren für alle drei Lösungsansätze, die oben beschrieben wurden, Projekte, die diese umsetzen. Wegen der Kosten, die in der Regel mit dem Aufbau von Ontologien verbunden sind, wäre es sinnvoll, nicht annotierte Dokumente nachträglich und automatisch, semantisch anzureichern. Projekte, wie WebFountain verfolgen diesen Ansatz.

- Google rückt von seinem Konzept ab, jegliches Informationsbedürfnis mit einer einzigen Suchmaschine befriedigen zu wollen. Die normale Suchmaschine könnte auch die gewünschten Informationen liefern. Google Scholar optimiert jedoch die potentielle Ergebnismenge bereits im Vorwege. Die Wissenschaft, Forschung und generell alle Akademiker haben jetzt ein, für ihre Belange funktionierendes, Werkzeug, um das Internet gezielter zu nutzen.

- Der Ansatz, je nach Fachlichkeit eine unterschiedliche Persistenzschicht zu Grunde zu legen, wie vom Projekt SHOE durch unterschiedliche Ontologien realisiert, wird auch in diese Arbeit einfließen. Es soll dabei darauf geachtet werden, dass die Topic Map prinzipiell austauschbar bleibt.

[10] Weder die Standard Webseiten zu Topic Maps, die Suche mit Google/Google Scholar, noch Email Kontakte (Betreuer + Professoren) brachten solch ein Projekt zum Vorschein.

[11] Da leider die Qualität der Quelle nicht verifiziert werden konnte, fand das Projekt keine tragende Berücksichtigung.

2.2. Information Retrieval

Dieses Kapitel ist aus zweierlei Gründen relevant. Zum einen fällt das Problem, das dieser Arbeit zu Grunde liegt, direkt in den Bereich von *Information Retrieval*, zum anderen ist die Teildisziplin *Evaluierung* Basis der praktischen, halbstatistischen Erhebung, die zur Bewertung des angestrebten Prototypen hinzugenommen wird.
Nach einer kurzen Einführung wird die Herausforderung innerhalb des Information Retrieval im Abschnitt *Definition* erläutert. Danach werden die Evaluierung als Werkzeug zur Bewertung vorgestellt und die *Summative Evaluierung* als probates Mittel ausgewählt. Abschließend wird die Bedeutung des Begriffes *Pertinenz* im Kontext von Relevanz hervorgehoben, da sie im praktischen Teil dieser Diplomarbeit von Bedeutung ist.

Allgemeines

Schnelle Rechnernetze und große Speichermedien haben dazu geführt, dass heute riesige Datenmengen in elektronischer Form im Internet zur Verfügung stehen. Viele Information sind dabei frei und unentgeltlich verfügbar, so man sie denn findet. Das Internet in seiner heutigen Form ist relativ unübersichtlich, da jeder innerhalb seiner Internetpräsenz publizieren darf, was er für richtig hält. Verschiedene Technologien, sowohl auf der Client- als auch auf der Serverseite, dazu unterschiedlich aufgebaute Webseiten machen es nicht leicht, das Internet gut strukturiert zugänglich zu machen. Andererseits weckt das Internet aber auch Begehrlichkeiten:
So beschreibt [BYRN99], dass das Internet sich stetig zu einer weltweiten Lagerstätte menschlichen Wissens und Kultur entwickelt, welche es erlaubt, Ideen und Informationen in einer noch nie da gewesenen Größenordnung auszutauschen.
Auf so eine reichhaltige Quelle kann man natürlich schlecht verzichten. Das Problem definiert sich durch die Frage: Wie kann man beliebig strukturierten Informationsquellen mit unterschiedlicher semantischer Ausdrucksstärke Herr werden?
Der folgende Abschnitt orientiert sich größtenteils an [Fuh04]. Darüber hinaus finden sich Informationen zum Thema Information Retrieval u.a. bei [KT00], [Fuh99], [fIF05], [vR05] und [Fer03].

Definition

> Im Information Retrieval (IR) werden Informationssysteme in Bezug auf ihre Rolle im Prozess des Wissenstransfers vom menschlichen Wissensproduzenten zum Informations-Nachfragenden betrachtet. [Fuh04] (Seite 6)

Dabei konzentriert man sich auf den Umgang mit *vagen Anfragen* und *unsicherem Wissen*. Bei vagen Anfragen wird man sowohl mit uneindeutigen Antworten, als auch mit Antworten, die erst durch Reformulierung der Anfrage innerhalb der Antwortmenge zum Ziel führen, konfrontiert. Erschwerend findet man bereits bei der ursprünglichen Suche nicht alle relevanten Dokumente, die dann durch Reformulierung nicht zu erfassen sind (siehe Abbildung 2.2)
Die Darstellungsform von Wissen ist vielfältig. Die begrenzte Repräsentation einer allgemeinen Wissens-Semantik sorgt für Unsicherheit diesen Wissens. Darüber hinaus können die Daten selbst unvollständig oder unsicher sein.

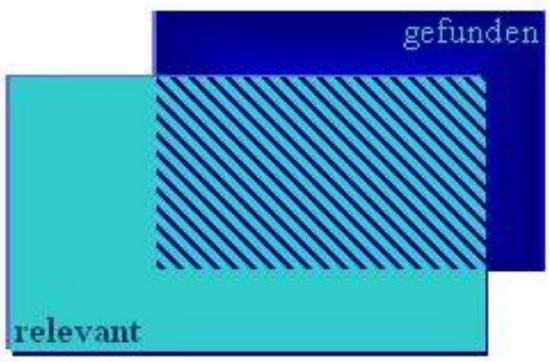

Abbildung 2.2.: Gefundene Dokumente

Hierbei muss man sich der begrifflichen Abgrenzung von Wissen zu Daten und Information bewusst sein. Dazu kann man die Erläuterungen aus der deutschen Informationswissenschaft wie in Abbildung 2.3 (entnommen aus [Fuh04]) heranziehen.

Evaluierung

Ein großes Kapitel von Information Retrieval beschäftigt sich mit der Evaluierung, bei der es darum geht, die Qualität eines Systems bewerten zu helfen. Dabei darf man nicht vergessen, dass es unterschiedliche Blickwinkel gibt, je nachdem, welches Interesse man verfolgt. Insgesamt ist das Ziel aber eindeutig, nämlich die Verbesserung des Systems.
Es gibt vier Arten von Evaluierung:

- Formative Evaluierung (zu Beginn)

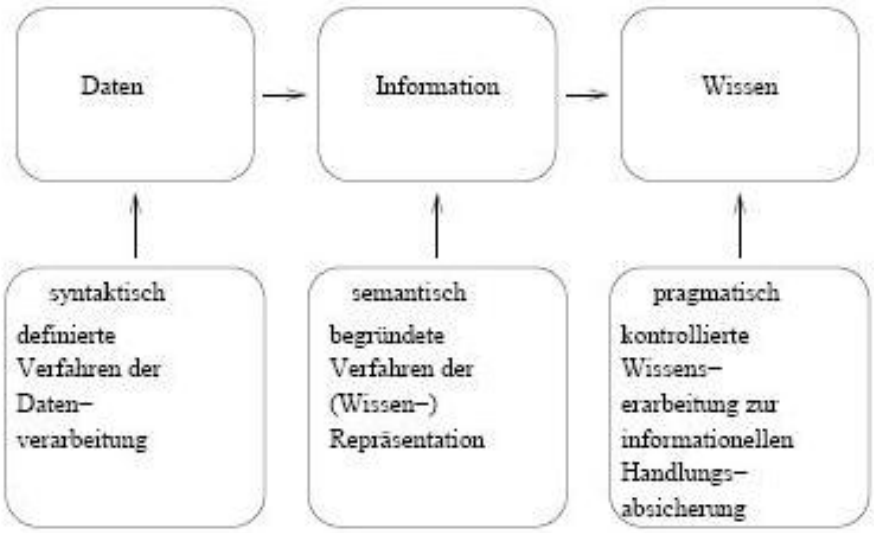

Abbildung 2.3.: Daten Information Wissen

- **Summative Evaluierung**
- Iterative Evaluierung (begleitend)
- Komparative Evaluierung (vergleichend)

Die summative Evaluierung wird am Projekt-Ende durchgeführt und vergleicht das realisierte System mit den ursprünglichen Projektzielen. Dieses wird in dieser Arbeit in leicht abgewandelter Form angewandt, denn die Erwartungshaltung ist nicht an ein positives Ergebnis gekoppelt. Vielmehr geht es darum, die Eignung von Topic Maps für das untersuchte informatische Problem zu hinterfragen, was auch mit einem negativen Resultat einhergehen kann.
Bei der Evaluierung sind viele Teilbereiche wichtig, wie Z.B. die Effizienz, Rangordnungen, Bewertungsmaße, Boolsches Retrieval usw. (siehe auch [Fer03]). Für diese Arbeit ist insbesondere der Begriff der *Relevanz* von Bedeutung, da, wie später ersichtlich wird, im Rahmen dieser Diplomarbeit nicht statistische, sondern halbstatistische Daten zugrunde liegen werden. Es geht nicht darum, möglichst große Datenmenge zu verarbeiten, sondern einen gewissen Grad an Plausibilität zu erreichen.

Relevanz

Es wird in vier verschiedene Arten von Relevanz unterschieden. Von

- situativer Relevanz
- **Pertinenz**
- objektiver Relevanz und
- Systemrelevanz

ist hier die Pertinenz wesentlich, die eine, für den Nutzer subjektive, Nützlichkeit des Dokumentes gegenüber seinem Informationswunsch, beschreibt. Eine solche Nützlichkeit ist schwerlich objektiv zu bewerten, daher muss die Beurteilung dieser Pertinenz mit Vorsicht vorgenommen werden.

Schlussbemerkungen

Es ist leicht einzusehen, dass *vage Anfragen* natürlich sind und auftreten, da der Benutzer häufig mehr mit Suchbegriffen verbindet, als innerhalb der Maschine ausgedrückt wird. Uneindeutige Antworten und Verfehlen relevanter Dokumente sind die daraus resultierenden Probleme. Sie haben wesentlichen Anteil daran, dass Anspruch und Wirklichkeit bei der Benutzung von Suchmaschinen teilweise sehr weit auseinander klaffen.
Mittels Pertinenz (als Teildisziplin der Relevanz) soll in dieser Arbeit ermittelt werden, ob eine Verbesserung der Suche dank des eingesetzten Prototyps und der angekoppelten Topic Map erzielt werden kann.

2.3. Suchmaschinen

Im Folgenden wird kurz in die Arbeitsweise von Suchmaschinen eingeführt und am Ende aufgezeigt, welcher Lösungsansatz (mindestens) mit ihnen realisiert wird. Zuvor wird dann die Metasuchmaschine als vorteilhafte Variante der einfachen Suchmaschinen vorgestellt. Vor den abschließenden Bemerkungen wird noch einmal an formale Sprachen und die *boolsche Suche* erinnert und ihre praktischen Umsetzungen diskutiert.
Der folgende Abschnitt orientiert sich an [Glo03].

Einleitung

Das WorldWideWeb (WWW) besteht aus einer Vielzahl von Hypertextdokumenten, die durch Hyperlinks miteinander verbunden sind. Dabei werden nach dem Client-Server-Prinzip durch den Benutzer angebotene Dienste des Servers konsumiert. Das Problem besteht darin, Benutzer und Dienste-Anbieter zusammenzuführen.
Eine einfache Lösung bilden dabei Suchkataloge[12] (z.B. yahoo). Dort sind viele Angebote des WWW eingetragen und können dadurch komfortabel abgerufen werden[13]. Wird ein Dienst jedoch nicht in einen Webkatalog eingetragen, ist er auch nicht verfügbar. Diese Lücke versuchen Suchmaschinen zu schließen.

Allgemeines

Das Ziel von Suchmaschinen besteht darin, das WWW möglichst vollständig abzubilden. Dabei sollen die Veränderungen an bereits gespeicherten Dokumenten möglichst zeitnah erfasst und gleichzeitig mit dem Wachstum im Internet schritt gehalten werden.
Viele Faktoren sorgen dafür, dass Informationen und Wissen im Internet nicht so einfach zu handhaben sind. Sie sind auf unterschiedliche Weise abgelegt, zum Teil nur temporär verfügbar und (noch) selten für automatisierte, semantische Verfahren vorbereitet.

> Grundsätzlich gilt die Regel, dass, je überschaubarer und eindeutig identifizierbar ein Datenbestand ist, je homogener die Eingangsdaten sind und je besser abgrenzbar die technischen Rahmenbedingungen sind, desto eher ist es möglich, Daten organisiert und kostengünstig in einen Bestand aufzunehmen, sowie über ihren Lebenszyklus hinweg periodisch zu aktualisieren. [Glo03] (Seite 25)

[12] Suchkataloge werden auch Webkataloge genannt.
[13] Weitere Informationen zu Suchkatalogen vs. Suchmaschinen in [Noa01].

Viele Internet-Angebote wirken da eher kontraproduktiv. Suchmaschinen bearbeiten deshalb Dokumente, bevor sie sie ablegen, um einen (syntaktischen) Standard zu erzielen und um eine semantische Ausdrucksstärke zu erreichen.

Aufbau von Suchmaschinen

Suchmaschinen bestehen in der Regel aus folgenden drei Teilen:

- Webrobots
- Information Retrieval System
- Query Processor

Webrobots

Webrobots (und Crawler bzw. Spider) werden vom Gesamtsystem eingesetzt, um den Datenbestand zu pflegen. Hierbei fungieren sie als Client, um vom Server Daten zu erhalten. Sie können bereits hier technisch fehlerhafte Inhalte übergehen, doppelte Inhalte erkennen und nicht erwünschte Daten ignorieren.
Webrobots finden aber nicht nur HTML-Dokumente, sondern auch andere Dokumenttypen - je nach Voreinstellung. Sinnvollerweise werden aber nur begrenzte Typen erfasst, um die Eingangsdaten homogen zu halten und dadurch einen höheren Effizienzgrad bei der Verarbeitung der Daten zu erreichen.
Webrobots setzen sich aus vier Elementen zusammen:

- Der Gatherer sammelt Dokumente im WWW ein.
- Um die Aufträge zu verwalten gibt es einen Loader.
- Die gesammelten URLs werden in einer URL-Datenbank abgelegt.
- Der Checker schließlich wendet, je nach Bedarf, Filter an.

Die Arbeitsweise von Webrobots kann durch den Gestalter einer Webseite gesteuert werden. Dazu kann man im Bereich der Meta-Informationen der Seite Kommandos erteilen, die den Webrobot beeinflussen. Beispielsweise kann er angewiesen werden, die Seite nicht zu indizieren oder Verlinkungen nicht zu beachten.
Eine große Schwäche von Webrobots liegt darin, die Aktualität einer Webseite festzustellen. Hat sich der Inhalt nicht geändert, so gibt es zwei Möglichkeiten. Entweder wurde die Seite nicht gepflegt oder der Inhalt ist bewusst gleich geblieben. Auch Datumsangaben lösen das Problem in keiner Weise, da sie dynamisch sein können.

Information Retrieval System

Die Hauptaufgabe des Information Retrieval System besteht darin, die gefundenen Daten aufzubereiten, damit sie effizient gespeichert und genutzt werden können.
Der erste Prozess ist die Datennormalisierung. Hier werden Dateien bzw. Inhalte in ein einheitliches Zielformat übersetzt, das eine Gleichbehandlung für weiterführende Prozesse ermöglicht.
Im Anschluss daran wird das Dokument inhaltlich analysiert. Dazu werden z.B. Zeichenfolgen semantisch erkannt. Man bezeichnet das auch als Wortidentifikation.
Danach wird die Sprache ermittelt und zudem Wortstämme gebildet, um mehr Treffer zu ermöglichen, gleichzeitig aber weniger Metadaten zu verwalten.
Ein Thema inhaltlich zu repräsentieren wird dann mittels Deskriptoren versucht. Hierbei ergibt sich allerdings die Schwierigkeit zu ermitteln, welche Zeichenketten als Deskriptoren für das jeweilige Dokument geeignet sind und welche nicht. Das Zipf'sche Gesetz besagt,

> dass es für den Verfasser einfacher ist, bestimmte Worte zu wiederholen, um ein Thema zu beschreiben, als ständig nach neuen Begriffen zu suchen. [Glo03] (Seite 55)

In der Praxis hat sich herausgestellt, dass sehr häufig verwendete Wörter jedoch genauso ungeeignet als Deskriptoren sind, wie sehr selten auftretende.
Wichtig für das Information Retrieval System ist auch noch die *black list*. Sie soll helfen, ungewünschte Inhalte aus dem Datenbestand fernzuhalten.
Die Indexierung wird dann überwiegend durch ein *invertiertes Dateisystem* realisiert. Alle Wörter der Dokumente werden in einem Index geführt. Jedes dieser Wörter verweist auf eine invertierte Datei, in der wiederum auf alle Dokumente verwiesen wird, die dieses Wort beinhalten.

Query Processor

Den Query Processor könnte man als *eigentliche* Suchmaschine bezeichnen. Er nutzt eine Funktion, um Dokumente mit einem *gewissen Ähnlichkeitsgrad*, innerhalb des vom Information Retrieval Systems verwalteten Datenpools, zu finden und danach in eine Reihenfolge zu bringen.[14] Durch verschiedene Gewichtungsmodelle wird der Relevanzgrad des Dokuments durch einen Algorithmus, bezogen auf eine Suchanfrage, berechnet. Dieser Algorithmus wird auch Retrievalfunktion genannt. [Glo03] Dabei kann die Retrievalfunktion verschiedene Verfahren kombinieren und unterschiedlich gewichten. Man unterscheidet Vektorraum basierte Modelle (binäres-, gewichtetes-) und Hypermedia basierte Modelle (relative Worthäufigkeit, inverse Dokumentenhäufigkeit).

[14] Näheres zu Rankings und Strategien finden sich zusätzlich bei [Fer03].

Relevant sind Dokumente, wenn sie ein zu definierendes Mindestmaß an Ähnlichkeit zur Suchanfrage besitzen. Beispielsweise liegt der Relevanzgrad für eine Zeichenkette *s* bei einer relationalen Datenbank genau bei 1, denn nur genau dann, wenn *s* gefunden wird, gibt es einen Treffer. Der Relevanzgrad innerhalb der Retrievalfunktion kann jedoch geringer sein.

Metasuchmaschinen

Unter Metasuchmaschinen versteht man Dienste, die die Suchanfrage an andere Suchmaschinen weiterleiten. Verschiedene Such- und Bewertungsverfahren, die durch Einbeziehung mehrerer Suchmaschinen ausgenutzt werden, sowie unterschiedlich abgedeckte Teilbereiche des WWW, sorgen für verbesserte Resultate. Es gibt aber auch zwei große Probleme zu lösen.
Zum einen muss berücksichtigt werden, dass Anfragen an die einzelnen Suchmaschinen angepasst werden müssen. Akzeptierte Wörter, Wortabstandsoperatoren, die integrierte *boolsche Suche* usw. unterscheiden sich mitunter erheblich[15].
Zum anderen müssen die erzielten Resultate in einer sinnvollen einheitlichen Weise zusammengefasst und dargestellt werden. Dabei muss auf verschiedene Dinge geachtet werden, wie z.B. der Umgang mit dem Auftreten von Webseiten, die durch unterschiedliche Suchmaschinen gefunden wurden und daher mehrfach auftreten können. [16]
Metasuchmaschinen existieren bereits und aufgrund ihres Vorteils der Mächtigkeit gegenüber einfachen Suchmaschinen ist ihr Einsatz daher grundsätzlich erstrebenswert.

Abfragesprachen

Wie in Abschnitt 1 ausgeführt wurde, erwartet eine (stichwortbasierte) Suchmaschine, ein syntaktisches Konstrukt. [CS03] führt aus, dass Syntax durch formale Sprachen beschrieben wird.

Formale Sprachen

Die folgenden beiden Definitionen sind [SSH95] entnommen[17]:

[15] Wobei es auch Metasuchmaschinen gibt, die die Anfrage ungefiltert an die einfachen Suchmaschinen verteilen (z.B. `http://www.search.com/`).
[16] Weiteres über die Realisierung von Metasuchmaschinen findet sich bei [Fer03].
[17] Weitere Informationen zu formalen Sprachen, kontextfreien Grammatiken und zur allgemeinen theoretischen Informatik finden sich in [HU94], [Sch99], [VSU99] und in [FB96].

2.3.1 DEFINITION (ALPHABET, WORT, WORTMONOID) Eine nicht leere, endliche Menge E von Zeichen nennen wir ein *Alphabet*, und jede endliche Folge über E nennen wir ein *Wort* über E. Die leere Folge heißt auch *leeres Wort* und wird mit λ bezeichnet. Die Menge E* aller Worte über E bildet bzgl. der Verknüpfung $\circ$ ein freies Monoid, das *Wortmonoid* über E.
$\diamond$

Mit Hilfe des Wortmonoids werden jetzt *formale Sprachen* definiert.

2.3.2 DEFINITION (FORMALE SPRACHE) Es sei E ein Alphabet (also eine endliche, nicht leere Menge von Zeichen), und es sei E* das Wortmonoid über E.
$L \subseteq E^*$ heißt eine *formale Sprache*, falls L durch ein endliches formales System vollständig beschreibbar ist.[18]
$\diamond$

Suchmaschinen, also Computerprogramme, die Anfragen entgegen nehmen, gehören zu solchen endlichen, formalen Systemen und beschreiben eine Menge L, die damit per Definition auch eine formale Sprache ist. Jedoch entsteht eine Diskrepanz aus der Tatsache, dass Suchmaschinen auf der einen Seite eine Vorstellung darüber haben, wie Benutzer ihre Dienste nutzen sollen, auf der anderen Seite aber nicht damit rechnen können, dass auch alle Benutzer dazu in der Lage sind. Suchmaschinen sind in der Regel für jeden gedacht, unabhängig von der Qualifikation. Um Nutzer nicht zu verärgern wird daher in der Regel alles akzeptiert, was eingegeben wird, egal wie semantisch abstrus die Eingabe ist.
Um eine Aussage über die Eignung von Topic Maps als Hilfsmittel zur Suche treffen zu können, sollen jedoch lediglich Eingaben verarbeitet werden, die relevant sind. Deshalb soll in dieser Arbeit ein Teil der Verantwortlichkeit auf den Benutzer übergehen. Das wird erreicht, indem eine genaue Beschreibung der zulässigen Eingaben angegeben wird. Das kann z.B. mittels der erweiterten Backus-Naur-Form (EBNF) geschehen, die dazu geeignet ist. Grundlage für die Aussagefähigkeit der EBNF bilden *Semi-Thue-Systeme* und die darauf aufbauende *Chomsky-Grammatik*, deren Definitionen hier nicht aufgeführt werden[19].

Operatoren

Theoretisch sollen Suchmaschinen, mit wenigen Operatoren ausgestattet, bereits sehr mächtig sein. Der *Schefferstrich* bildet zwar bereits alleine eine Junktorbasis, ist aber für

[18] Der Hinweis aus [SSH95], dass ein endliches, formales System nicht näher präzisiert wird, man sich aber alle möglichen Werkzeuge der diskreten Mathematik darunter vorstellen darf, soll hier ebenfalls Beachtung finden.
[19] Weiteres dazu bei [SSH95].

einen normalen Benutzer mehr als verwirrend. Suchmaschinen versuchen daher, mit natürlichen und alltäglichen Operatoren auszukommen. *UND* und *ODER* bilden zwar alleine auch kein Erzeugendensystem, nimmt man jedoch das *NOT* hinzu, erhält man die notwendige Mächtigkeit um Reformulierungen innerhalb der Zielmenge überflüssig zu machen. Diese Art der Suche nennt man auch *Boolsches Retrieval* ([Fer03], [Fuh04]).
Dazu gibt es im Umfeld der Suchmaschinen nach [Glo03] noch folgende Operatoren:

- *ADJ*, findet benachbarte Wörter.
- *NEAR*, findet Nachbarn der Entfernung 10-25 Wörter.
- *FAR*, findet Wörter, die mindestens einmal 25 Wörter auseinander stehen.
- *(...)*, Klammern sind notwendig, damit ein System erzeugt werden kann.
- *, fungiert als Platzhalter.
- *"..."*, dient zur Phrasensuche.

Umsetzungen

In der Praxis mit Suchmaschinen wird es leider sehr ungenau. Das, was Suchmaschinen behaupten leisten zu können, leisten sie auf eine teilweise unmathematische und proprietäre Art.

2.3.1 BEISPIEL (FALLSTUDIE GOOGLE) Marktführer *Google*, als bedeutender Vertreter einfacher Suchmaschinen, gibt auf seinen Seiten sinngemäß folgende Suchtipps an[20]:

- Eingabe von Suchbegriffen in das Suchfeld. Abschicken mittels *return*. Als Ergebnis soll eine Liste relevanter Ergebnisse erscheinen.
- Google soll Aussagen von Seiten über andere Seiten berücksichtigen. Zudem sollen Dokumente bevorzugt werden, in denen mehrere Suchbegriffe möglichst nah beieinander stehen.
- Eine automatische *Und-Suche* verknüpft Suchbegriffe und erspart das obligatorische *and*.
- Google arbeitet *mit Stop-Wörtern*, wie z.B. *http* und *.com*. Diese Begriffe werden nicht gefunden, es sei denn, man gibt mittels + an, diese Wörter exakt so suchen zu wollen. Vorher muss allerdings ein Leerzeichen stehen.
- In den Suchergebnissen werden kurze Auszüge aus dem Dokument zur Vorauswahl angezeigt.

[20] URL: `http://www.google.de/intl/de/help/basics.html`, zuletzt besucht am 17.03.2005.

- Die Suche ist nicht *case sensitive*. Umlaute werden nicht gesondert behandelt. Mit + kann man Google motivieren, auch danach exakt zu suchen.
- Worttrennungen und unterschiedliche Schreibweisen sollen irrelevant sein. *Musikanlage* und *Musik-Anlage* sollen daher ebenso identische Ergebnisse liefern wie *Grafik* und *Graphik*.

Es fällt u.a. auf, dass Stop-Wörter, Worttrennungen und unterschiedliche Schreibweisen bereits Subjektivität provozieren. Welche Stop-Wörter nimmt das System auf? Wie wird mit Umlauten und 'ß' verfahren?
Der Umgang mit AND, OR und '-', das bei Google für das logische NOT steht, ist dazu mehr als fragwürdig. U.a. werden Klammern überlesen und OR vor AND ausgewertet. Eine einfache, maschinelle Ersetzung muss daher bedächtig eingesetzt werden. Das hat für diese Arbeit auch praktische Bedeutung.
Die Suchanfrage des Benutzers

```
Kaffee OR Java OR Smalltalk
```

könnte z.B. durch eine Maschine zu

```
Kaffee OR (Java AND Urlaub) OR Smalltalk
```

erweitert werden, würde aber aufgrund der Auswertungsreihenfolge von Google als

```
(Kaffee OR Java) AND (Urlaub OR Smalltalk)
```

verarbeitet.
Ein weiteres Problem ist, dass Google jede Eingabe akzeptiert und versucht, passende Ergebnisse zu liefern, ganz gleich, wie abstrus die Eingabe aussieht. Die Eingabe

```
the
```

führt zu ca. 3.570.000.000 Treffern.

```
java OR smalltalk
```

bewirkt 56.100.000 Treffer. Deshalb sollte

```
java OR the OR smalltalk
```

mindestens 3.570.000.000 Treffer liefern, führt aber zu 56.100.000 Treffern. Offensichtlich wurde *the* nicht berücksichtigt.
Als letztes Beispiel soll die Eingabe

```
+-Cos
```

untersucht werden. Laut Angabe der Google-Suchtipps müsste jetzt genau der Begriff '-Cos' ohne Variationen gesucht werden. In der Praxis scheint Google aber das - als Präfix, also als NOT zu verstehen. Dadurch wird die leere Ausgabe erzeugt.
Ein weiterer Nachteil von Google ist die Begrenzung auf zehn Suchbegriffe innerhalb einer Anfrage.
Abschließend soll noch erwähnt werden, dass Google für Entwickler ein API bietet[21], um unabhängig vom Thin-Client suchen lassen zu können. Die Klassen des API greifen über einen Web-Service auf die Suche von Google zu. Die Nutzung von Google über das API hat allerdings auch ein paar Einschränkungen, die bei normaler Benutzung zu beachten sind:

- Länge der Suchanfrage maximal 2048 Bytes.
- Auch über das API sind maximal zehn Suchbegriffe je Anfrage möglich.
- Bei expliziter Suche nach Webseiten ist nur eine Seite je Anfrage erlaubt.
- Eine starke Einschränkung gibt es bei der Größe der Ergebnismenge, die mit zehn deutlich kleiner ist, als bei der Suche mittels eines Browsers.
- Um das API einsetzen zu können, braucht man ein spezielles Konto bei Google. Dieses Konto ist kostenlos, begrenzt die Anzahl der Anfragen zurzeit jedoch auf 1000 je Tag.

◇

Aufgrund der zuvor beschriebenen Unsicherheiten beim Umgang mit Suchmaschinen, könnte man auch den Begriff des *Suchautomaten* einführen, der dann einen Standard definiert, an dem sich Suchmaschinen messen ließen. Der Begriff der *formalen Suchsprache* würde dann diesen theoretischen Ansatz abrunden.[22]
Leider sind die Algorithmen und Verfahren der am Markt befindlichen, großen Suchmaschinen nicht zugänglich. Zudem ist nicht geklärt, ob Suchmaschinenhersteller überhaupt ein Interesse an einer standardisierten Suchsprache hätten. Zwar würde die allgemeine Verwendung erleichtert, andererseits bindet man Nutzer durch proprietäre Ansätze auch an das

[21] Näheres dazu auf `http://www.google.com/apis/` (zuletzt getestet am 28.04.2005).
[22] Für den Fall, dass man Klammerungen zulassen möchte, müsste die *formale Suchsprache* dann aber nicht nur das genaue Aussehen des Suchstrings festlegen, sondern auch vorgeben, wie eine Suchmaschine den Abarbeitungsbaum aufzubauen hat.

eigene Unternehmen[23].
Ein weiteres Beispiel soll eine Alternative zu Google untersuchen.

2.3.2 BEISPIEL (FALLSTUDIE METAGER) *MetaGer*[24] ist eine Metasuchmaschine über deutsche Suchmaschinen. Der Service wird vom RRZN[25] der Universität Hannover angeboten. 1995 begann die Universität Hannover im Suchmaschinenlabor mit der Forschung und betreibt diesen Suchservice, trotz der großen Akzeptanz, auch zu Forschungszwecken. MetaGer selbst verweist dabei auf ein Interview mit dem ORF mit dem Titel *metager - eine suchmaschine als forschungsprojekt* [Tha01], in dem unter anderem von einer Zugriffszahl von 400.000-450.000 am Tag (Stand 28. Juli 2001) die Rede ist[26].
Im Vergleich zu anderen Meta-Suchmachinen macht MetaGer einen verantwortungsvollen Eindruck. MetaGer weiß um die möglichen Probleme im Umgang mit Suchmaschinen und zieht daraus einige Konsequenzen. Beispielsweise erwartet MetaGer von Suchmaschinen, dass sie den *sAND-Test* bestehen[27]. Außerdem kann man optional angeben, dass wissenschaftliche Quellen höher bewertet werden sollen, als nicht-wissenschaftliche Quellen. Man kann MetaGer auffordern, die Existenz der Ursprungsquelle zu überprüfen und nach Relevanz bzw. Aktualität zu sortieren. Darüber hinaus kann man die Suchmaschinen, die MetaGer benutzt, je nach Bedarf auswählen oder von der Suche ausschließen. Es kann die maximale Anzahl der Treffer, die maximale Suchzeit, die Anzahl anzuzeigender Dubletten, die alphabetische Clusterung der Ausgabe nach Webservern, sowie die Anzeige der Trefferzahlen der einzelnen Suchdienste eingestellt werden. Innovativ sind auch die ebenfalls optionalen phonetischen Suchvorschläge, falls die Suche zu wenig Treffern führte, sowie die Assoziatoren. Der Web-Assoziator und der neue Q-Assoziator suchen zu ein oder mehreren Begriffen semantisch verwandte Wörter heraus, mit denen man dann die ursprünglich erfolgsarme Suchanfrage verbessern kann.
MetaGer reicht nicht alle Anfragen ungefiltert an andere Suchmaschinen weiter. Außerdem akzeptiert MetaGer nicht jede Eingabe und stuft den Benutzer somit mündiger ein, als viele Mitbewerber dies tun. Auf eine Anfrage mit unzulässigem Sonderzeichen erhält man folgende Antwort:

> zulässig sind: alle Buchstaben und Ziffern, sowie Bindestrich - Punkt . Aus-

[23] Ein gutes Beispiel dafür ist Outlook Express von Microsoft, dessen benutztes Mailformat proprietär und somit schwer zu exportieren ist.
[24] Siehe auch [Uni05b].
[25] Siehe auch [Uni05a].
[26] Inzwischen gibt es laut eigenen Angaben ca. 35.000.000 http-Requests, 7.000.000 Seitenabrufe und 3.500.000 Netto-Abfragen im Monat. Ca. 250.000 externe Links auf MetaGer komplettieren diese Statistik. Damit ist MetaGer eine der am häufigsten abgefragten und verlinkten deutschen Suchmaschinen überhaupt.
[27] Der sAND-Test überprüft, ob die Suchmaschine korrekte Ergebnisse im Falle AND-verknüpfter Wörter liefert. Des Weiteren wird ermittelt, ob beim Stichwort *sand* auch Begriffe wie *Versand* gefunden werden. Wenn ja, dann fällt die Suchmaschine ebenfalls durch und wird nicht in die Suche integriert.

> rufezeichen(nur bei der Stop-Wort-Suche) ! Sternchen * und das Leerzeichen. Falls Sie mit „ (Anführungszeichen) eigentlich eine Stringsuche beabsichtigten: klicken Sie bitte auf der Startseite den Auswahlschalter „Alle Worte sollen im Dokument vorkommen“ an und schalten Sie um auf „Worte als String ...“

Der Stern steht dabei als Wildcat und alle Wörter, die nach dem Ausrufezeichen kommen, dürfen im Suchtext nicht vorkommen. Der Bindestrich soll umsichtig genutzt werden, weil Suchmaschinen, die MetaGer anfragt, unterschiedlich darauf reagieren. Dieser Hinweis ist wichtig, denn er macht klar, dass MetaGer die Verantwortung nicht voll übernimmt und die Anfrage offensichtlich nicht maximal an die Sprache der einzelnen Suchmaschinen anpasst. Ein weiterer Nachteil dieses Suchsystems ist die boolsche Suche. Mit einem Auswahlschalter kann man vier Voreinstellungen auswählen:

- Alle Wörter sollen im Dokument vorkommen
- Alles suchen, aber Wörter nach ! ausschließen
- Mindestens eines der Worte im Dokument
- Worte als String in Titel oder Kurzbeschreibung

Unglücklicherweise exkludiert diese Voraussetzung ein paar Möglichkeiten. Will man z.B. nach *Topic Maps* suchen und dabei den Begriff *RDF* ausschließen, so ist das nicht realisierbar. Dazu wäre eine Kombination von Stop-Wörtern und Strings notwendig. Im Übrigen bezieht sich die MetaGer Stringsuche ohnehin nur auf den Titel bzw. die Kurzbeschreibung des Dokumentes. Eine Erklärung für genau diese Einschränkung gibt MetaGer auf der eigenen Seite in der Hilfe:

> Eine echte String-Suche(also die Suche nach der exakten Wortfolge) über den gesamten Text aller Dokumente ist derzeit leider nicht möglich, da nur wenige der von MetaGer abgefragten Suchdienste diese Möglichkeit bieten.

Insgesamt ist MetaGer dennoch eine sehr innovative und empfehlenswerte Suchmaschine. Die Abfragesprache weist aber auch hier erhebliche Mängel auf.

◇

Schlussbemerkungen

Zum einen hat das Kapitel deutlich gemacht, dass Suchmaschinen mit den Komponenten, Information Retrieval System und Query Processor, darauf ausgelegt sind, auf Dokumenten-Seite Verfahren anzuwenden, die einen sinnvollen Zugriff unterstützen. Auch hier geht es nur um die Frage, wie man die Semantik, die der Benutzer impliziert, im Rechner abbilden

kann[28].
Zum anderen wurde die Grundlage geschaffen, in der anschließenden Synthese eine formale Sprache mittels EBNF anzugeben und somit die Verantwortlichkeit für den Benutzer festzulegen.
Die Auswahl einer geeigneten Suchmaschine fällt nicht leicht. Suchmaschinen sind auf allgemeine Benutzung ausgerichtet. Die Anfragesprachen sind daher nicht streng an der boolschen Algebra ausgerichtet. Welche Suchmaschine in dieser Arbeit zum Einsatz kommt, soll am Anfang der Synthese erläutert werden.

[28] Weitere Informationen zu Suchmaschinen finden sich auch bei [MW03].

2.4. Topic Maps

Die Persistenzschicht in diesem Projekt wird durch Topic Maps realisiert. Dazu muss eine Maschine eingesetzt werden, die die Topic Map ausliest und an das, zu erstellende, System angeschlossen wird. Der folgende Abschnitt soll die Grundlagen erarbeiten, um Topic Maps einsetzen zu können.
Es werden die Konzepte von Topic Maps erläutert und auf den Einsatz von Topic Maps in der Praxis hingewiesen. Dabei ist von Interesse, ob die semantische Ausdrucksstärke von Topic Maps überhaupt hinreichend ist.

Allgemeines

Erste Ansätze zur Entwicklung von Topic Maps gab es bereits 1991 durch die *Davenport Group*. Ihre Motivation bestand darin, eine Standard SGML DTD für Software Dokumentationen zu entwickeln. Das resultierte dann in der CApH

> This group quite quickly spun off an offshoot called CApH (Conventions for the Application of HyTime) , one of whose tasks was to design an application for computerized back-of-book indexes. ([Gar05], Kapitel 1.1.)

Die Idee dahinter war, die Fähigkeit zu erlangen, die unterschiedlichen Indexe zusammenfügen zu können. Das war die Geburtsstunde von Topic Maps. ISO's SGML Arbeitsgruppe akzeptierte Topic Maps 1996 und brauchten dann noch 4 Jahre, um Topic Maps zu standardisieren [Gar05]. Unter ISO 13250 kann man die genaue Spezifikation finden. Auf der Internetseite [HM03][29] findet sich folgender Abstract:

> ISO/IEC 13250:2003 (2nd edition) specifies two syntaxes for the interchange of Topic Maps. One of these syntaxes is based on the ISO/IEC 10744:1997 (HyTime) meta-DTD (meta Document Type Definition), and it is itself specified as a meta-DTD. The other, called XTM (XML Topic Maps), is specified as an eXtensible Markup Language (XML) DTD.

HyTime

HyTime hat das Aussehen einer Meta-DTD, deren *Generic Identifiers* frei vergeben werden können. Die Elementarten und Attributformen sind nicht vorgeschrieben und können kombiniert werden.

[29] Zuletzt besucht am 22.03.2005.

> Der Standard existiert seit 1992 und deckt eine Menge von *hypermedialen* Aspekten ab, wie etwa komplexes Hyperlinking, Einbindung jeglicher Art von Inhalt in Informationsobjekte (Grafik, Video, Text, etc.), virtuelle Zeit, Scheduling-Mechanismen, Synchronisation usw. Hypermedial meint dabei die Verwendung von Konzepten aus den Bereichen Hypertext und Multimedia. ([WM02], Seite 123)

Näheres zu HyTime, HyTime Moduln und über die Struktur von HyTime-Dokumenten finden sich in [WM02].

XTM

Die zweite Syntax wird heute überwiegend eingesetzt und nennt sich XTM [Gar05]. Der ISO-Standard, der auf HyTime und SGML beruhte, wurde praktisch nach XML portiert. Zusätzlich gibt es noch ein paar Erweiterungen, die [WM02] (im Kapitel 11) ausführt.
Der praktische Teil dieser Arbeit wird XTM einsetzen. Die genaue Syntax führt an dieser Stelle nicht weiter und ist bei Bedarf [ea03] (dort im Kapitel 6) zu entnehmen.

Konzepte

Es gibt fünf Schlüsselkonzepte[30]. Diese Konzepte tauchen in einer Topic Map auf und machen ihren Charakter aus:

- Topic
- Subject Descriptor
- Occurence
- Association
- Scope

[30] [Dac03] oder direkt bei `http://www.topicmaps.org/xtm/index.html`.

Topic

Hinter einem *Topic* kann sich alles verbergen. Ein Topic ist ein Stellvertreter mit einem Namen, der für alles stehen kann. Beispiele:

```
<topic id="HomepageHAW">
.
.
</topic>

<topic id="Dekan">
.
.
</topic>
```

Ein Topic kann das eigentliche Objekt sein oder auch ein Ding des täglichen Lebens, das nicht referenziert werden kann. Die Idee, die hinter dem Topic steckt, ist möglichst alles abbildbar zu machen, daher gibt es hier auch keine Einschränkungen.

Subject Descriptor

Ein *Subject Descriptor* dient zur Beschreibung von Topics und wird im Zusammenhang mit der Occurence eingesetzt. Er kann die eigentliche Ressource beschreiben, indem z.B. ein erklärender Text folgt oder eine verlinkbare Ressource angegeben wird:

```
<resourceRef xlink:href="http://www.haw-hamburg.de/">

<resourceData>
  Der Prüfungsausschussvorsitzende der HAW vertritt u.a. die Interessen,
  die durch die Prüfungsordnung der HAW ausgedrückt werden.
</resourceData>
```

Occurence

Bislang ist ein Topic ein Stellvertreter ohne Inhalt (bis auf den Namen, der vielleicht etwas über das Topic aussagen kann). Eine *Occurence* ändert das. Mittels einer Occurence und dem Subject Descriptor kann man nun ein Topic beschreiben:

```
<topic id="HomepageHAW">
  <occurence>
    <resourceRef xlink:href="http://www.haw-hamburg.de/">
  </occurence>
</topic>

<topic id="PrüfungsausschussvorsitzenderHAW">
  <occurence>
    <resourceData>
      Der Prüfungsausschussvorsitzende der HAW vertritt u.a. die Interessen,
      die durch die Prüfungsordnung der HAW ausgedrückt werden.
    </resourceData>
  </occurence>
</topic>
```

Association

Associations setzen Topics zueinander in Beziehung. Dazu werden lediglich mehrere Topics in derselben Association aufgeführt. Innerhalb einer Association können Topics eine bestimmte Rolle einnehmen, so dass die Beziehung eindeutig wird.

```
<association id="Vater_von">
  <instanceOf><topicRef xlink:href="#verwandt_mit" /></instanceOf>
  <member>
    <roleSpec><topicRef xlink:href="#Vater" /></roleSpec>
    <topicRef xlink:href="#Tom" />
  </member>
  <member>
    <roleSpec><topicRef xlink:href="#Tochter" /></roleSpec>
    <topicRef xlink:href="#Anna" />
  </member>
</association>
```

<instanceOf> sagt dabei aus, von welchem Typ die Association ist.
Vergleicht man z.B. relationale Datenbanken mit Topic Maps, so erkennt man durch das Konzept der Associations einen Unterschied in der semantischen Ausdrucksfähigkeit. Denn Elemente müsste man in einer relationalen Datenbank erst durch eine zusätzliche Tabelle in

Beziehung setzen und die Rolle wäre damit auch noch nicht ausgedrückt[31].
Dennoch sind auch XTM im Vergleich zu stark semantischen Sprachen, wie DAML+Oil, OWL usw. relativ schwach in ihrer semantischen Ausdrucksweise. Näheres dazu bei [Dac03], [MFHS02], [Fen02] und auch [BVL03].

Scope

Scopes sind mit *Namespaces* zu vergleichen. Namen innerhalb eines Scopes sind eindeutig. Ressourcen, die innerhalb eines Topic angegeben sind, haben den gleichen Scope wie das Topic [32]. Man kann z.B. Internationalisierungen relativ einfach durch Scopes lösen:

```
<topic id="en">
  <subjectIdentity>
    <subjectIndicatorRef
      xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#en" />
    </subjectIdentity>
    <baseName>
      <baseNameString>english</baseNameString>
    </baseName>
</topic>

<topic id="de">
  <subjectIdentity>
    <subjectIndicatorRef
      xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#de" />
  </subjectIdentity>
  <baseName>
    <baseNameString>german</baseNameString>
  </baseName>
</topic>
```

[31] Eine Sichtweise des *semantic web* beschreibt die Entwicklung der Eigenschaften von Daten, je nach ihrer Position innerhalb des ontologischen Spektrums, das von schwach semantisch bis stark semantisch gegliedert ist. Dabei beinhalten Daten aufgrund ihrer Darstellung desto mehr Informationen, je weiter sie in Richtung *Ontologien* platziert sind. [Dac03] spricht dabei von *smart data* und unterstreicht das durch:

> With the Web, Extensible Markup Language (XML), and now the emerging Semantic Web, the shift of power is moving from applications to data.[Dac03] (Seite 2)

Ontologien sind dabei wichtige Instrumente innerhalb von Wissensrepräsentationen, die das Wissen definieren, das in einer Applikations-Domäne existiert [Sow00].

[32] Topics, die den gleichen *base name* haben, sollten daher auch zusammengefasst werden [Dac03].

```
<!-- Sprachermittlung -->

<topic id="language">
  <baseName>
    <scope>
      <topicRef xlink:href="#de" />
    </scope>
    <baseNameString>german</baseNameString>
  </baseName>
  <baseName>
    <scope>
      <topicRef xlink:href="#en" />
    </scope>
    <baseNameString>english</baseNameString>
  </baseName>
</topic>
```

Und im Topic selbst:

```
<topic id="object">
  <baseName>
    <scope>
      <topicRef xlink:href="#de" />
    </scope>
    <baseNameString>objekt</baseNameString>
  </baseName>
  <baseName>
    <scope>
      <topicRef xlink:href="#en" />
    </scope>
    <baseNameString>object</baseNameString>
  </baseName>
</topic>
```

Topic Map Templates

Ein weiterer wichtiger Fakt ist die Existenz von *Topic Map Templates* [ea03], die einen wohl definierten Startpunkt verankern. Diese festgelegten PSIs (Published Subject Indicators) findet man u.a. unter `http://www.topicmaps.com`. Ein Beispiel dazu ist:

```
<topic id="company">
  <instanceOf>
    <subjectIndicatorRef
      xlink:href="http://www.topicmaps.com/xtm/1.0/template.xtm
      #association-role-class">
  <\instanceOf>
<\topic>
```

Topic Maps im Einsatz

Einsatzgebiete

Topic Maps sind vielseitig verwendbar, wie man dem nachfolgenden Zitat entnehmen kann:

> Topic maps have many applications, but one of their main applications is that of solving the findability problem of information, that is: how to find the information you are looking for in large body of information. Topic maps can also be used for knowledge management, for web portal development, content management, and enterprise application integration (EAI). Topic maps are also being described as an enabling technology for the semantic web. ([Gar05], Kapitel 1.1.)

In dieser Arbeit soll insbesondere die Fähigkeit von Topic Maps genutzt werden, Topics in Beziehung zu setzen, Topics (die Begriffe repräsentieren werden) aufgrund ihrer Beziehungen zu finden und auch, je nach Kontext, zu verwerfen. Näheres dazu im Synthese-Teil im Abschnitt 3.

B2B + B2C

Während der Einsatz von Topic Maps zwischen Firmen (B2B) unkompliziert ist, stellt sich das bei Firmen, die mit ihren Klienten kommunizieren (B2C), schon schwieriger dar. Grund dafür ist insbesondere die Fachbegrifflichkeit. Anders ist das bei Unternehmen untereinander. Z.B. können ein Fertigungsbetrieb und ein Zulieferer klar definieren, welche Wörter sie verwenden. Bei Klienten eines Unternehmens ist das anders. Sie kommunizieren durch natürliche Sprache und dementsprechend muss sich das Unternehmen darauf einstellen.
TalkBots[33] bieten eine interessante, auf Topic Maps basierende Lösung[34]. Der Kunde wird

[33] Näheres dazu auf der Webseite von [nA05].
[34] Leider ist die Umsetzung nicht veröffentlicht worden, so dass hier keine nähere Diskussion stattfinden kann.

dort mit natürlichsprachlicher Verarbeitung und Ausgabe sehr geschickt und ohne spürbare Sprünge, freundlich durch das Gespräch geleitet.

Entwicklungsphasen beim Einsatz von Topic Maps

[WM02] beschreibt sieben wesentliche Phasen der Entwicklung:

- Analyse: Abgrenzung bzw. Identifizierung von Dokumenten und Wissen. Sichtung des technischen Umfeldes.
- Design: Aufbau der Topic Map. [WM02] stellt eine Applikation zum Design von Topic Maps, sowie zur Mehrbenutzereditierung in den Mittelpunkt.
- Erstellung: Die Erstellung wird bei diesem Ansatz vom Design nicht sauber getrennt. Als Beispiele dienen die Zusammenführung von verschiedenen Teilen von Topic Maps, Benutzergruppen mit verschiedenen Rechten und Aufgaben, sowie regelbasierte Transformation von vorhandenen Daten in Topic Maps.
- Speicherung: Forderungen bei größeren Projekten: effiziente Speicherung, Einsatz von Cashing-Mechanismen, angepasste Sicherheitssysteme, sowie Versionsmanagement.
- Administration: Veränderung, Wartung und Bereitstellung von Topic Maps.
- Publikation: elektronische Bereitstellung und standardisierter Datenaustausch durch z.B. SGML oder XML.
- Verwendung: Navigation und Abfrage der Topic Map.

Engines, Abfrage von Topic Maps

XML Topic Maps sind XML Dateien. Um ihre besonderen Eigenschaften zu nutzen, bedarf es Software, die diese XML Dateien als Topic Maps, möglichst effizient, ausliest. Diese Anwendungen nennt man auch *Topic Map Engines*. Auf der Internetseite [Woo05] finden sich einige Engines, u.a. *k42*, *TM4J* und *tmproc*.
Damit nicht jede Engine mit einer proprietären Schnittstelle ausgeliefert werden muss, gibt es inzwischen einen API Standard, der sich *TMAPI* nennt. Dadurch können Entwickler ausschließlich mit einem API arbeiten, ohne sich auf eine Engine festlegen zu müssen. Zudem wird dadurch die Wiederverwendbarkeit des Programmcodes unterstützt. Engines, die TMAPI unterstützen sind u.a. *TM4J*, *tinyTM* und *XTM4XMLDB*.
Um an die Daten innerhalb der Topic Map zu gelangen, wird die angebundene Topic Map

Engine beauftragt, der entsprechenden Topic Map die gewünschten Informationen zu entnehmen. Dafür kann auch eine Abfragesprache wie *Tolog*[35] eingesetzt werden.
Für relationale Datenbanken gibt es mit SQL einen bekannten Abfragestandard. Analog wird eine Transaktionssprache mit Namen *TMQL* entwickelt. Sie soll ein ISO Standard für die Abfrage von Topic Maps werden[36].

Bearbeitung

Die Topic Map muss nach ihrem Design hergestellt werden. Man kann eine XML Datei in einem Texteditor erstellen, ein Programm schreiben, das den Erstellungsprozess automatisiert oder Tools einsetzen, mit deren Hilfe man die Topic Map in vorgefertigten Masken erfasst. Ein kleines, nützliches Tool findet man z.B. bei [Hec05] und weitere bei [Woo05]. Aufgrund sich verändernder Informationen und Anforderungen reicht es natürlich auch nicht aus, Topic Maps einzusetzen ohne sie regelmäßig zu warten. Auch dafür kann man bei [Woo05] Tools finden.
Bei der Suche nach geeigneten Tools muss man allerdings der Tatsache Rechnung tragen, dass Topic Maps und ihre Spezifizierung noch sehr jung sind und daher sowohl Literatur als auch Software nicht in dem Maße erhältlich sind bzw. funktionieren, wie bei älteren, etablierteren Technologien. So wird zum Beispiel *Protégé* mit dem Plug-In *TMTab* von [Woo05] zu einem geeigneten Werkzeug für Topic Maps erklärt, was jedoch innerhalb dieser Arbeit leider nicht nachgewiesen werden konnte.

Schlussbemerkungen

Eine Frage, die sich bezüglich des Semantic Web aufdrängt, soll hier kurz erläutert werden. Denn könnte man nicht alle schwach semantischen Technologien austauschen und sie durch starke Ontologien ersetzen? In diesem Zuge stünde dann auch die Existenzberechtigung von Topic Maps zur Debatte, die [Dac03] im mittleren Bereich des ontologischen Spektrums ansiedelt. Das folgende Zitat macht deutlich, wofür Ontologien u.a. geschaffen werden.

> The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web-a web of data that can be processed directly or indirectly by machines. (Tim Berners-Lee, *Weaving the Web*, Harper San Francisco, 1999)[37]

[35] Näheres bei [Gar01].
[36] Zum Zeitpunkt dieser Arbeit jedoch noch in der Entwicklung.
[37] Entnommen aus [Dac03], Seite 1.

Ontologien und weitere verwandte Technologien haben allerdings nicht nur Vorteile. Sie sind auch teurer als weniger semantisch ausdrucksstarke Technologien. Je weiter man auf der semantischen Skala nach oben gelangt, desto teurer werden die Technologien in der Regel. Das gilt sowohl für Kosten bei der Herstellung als auch beim Einsatz. Man muss sich also stets fragen, welche Anforderungen an eine Technologie zu stellen sind, bevor man eine nähere Auswahl trifft.
Abschließend soll erwähnt werden, dass die oben aufgeführten Konzepte (nach [Dac03]) auch anders gewichtet werden können. [Wei03] beispielsweise spricht nur von drei elementaren Komponenten bei Topic Maps, nämlich den Topics, Occurrences und Associations. Außerdem wird dort eine Unterteilung in Topic-Schicht (zur Navigation) und Dokumenten-Schicht vorgenommen, wie man sie in Abbildung 2.4 (entnommen aus [Wei03]) erkennen kann. Dabei handelt es sich jedoch um eine begrenzte, kontextabhängige Sichtweise, denn es setzt voraus, dass Topic Maps genutzt werden, um Dokumente zu repräsentieren.
Durch Associations erreichen Topic Maps ein semantisches Niveau, auf dem man Bezie-

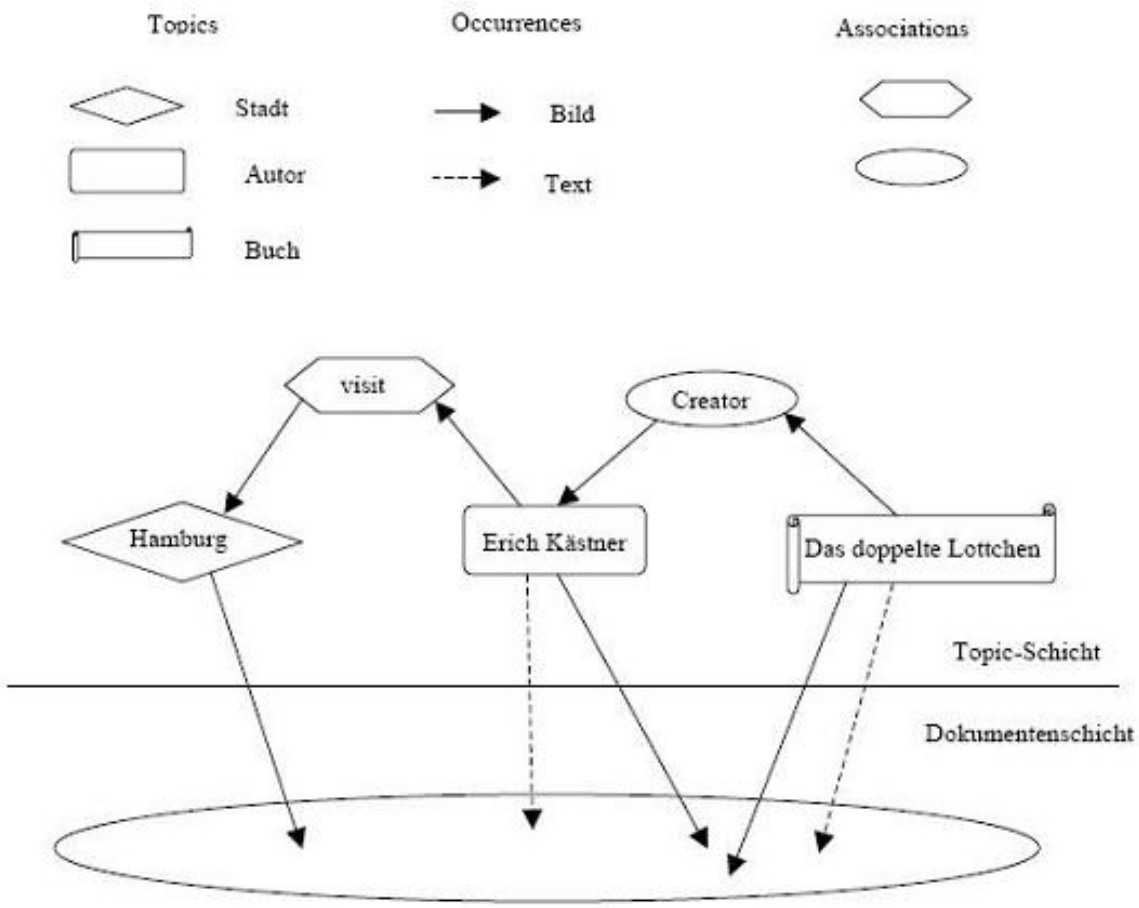

Abbildung 2.4.: Topic Map Schichten

hungen ausdrücken kann. Sollen also einfache Ersetzungen vorgenommen werden, wie z.B. *java –> java AND urlaub*, so ist eine Association zwischen *java* und *urlaub* dafür natürlich hinreichend. Abschließend sei noch erwähnt, dass die *Entwicklungsphasen beim Einsatz von Topic Maps* zur Orientierung bei der praktischen Umsetzung innerhalb dieser Arbeit dienen sollen, so weit sie für die Problemstellung angemessen erscheinen.

2.5. Das allgemeine Szenario

Ontologien und andere semantische Technologien sind, wie weiter oben ausgeführt wurde, kostenintensiv (vgl. Abschnitt 2.4). Daher muss zunächst ein weiterer Akteur hinzugefügt werden, der das semantische Modell betreut und vor allem das semantische Wissen pflegt. Das ist der Abbildung 2.5 zu entnehmen, in der das Diagramm aus Abbildung 1.1 im Abschnitt 1 entsprechend erweitert wurde.

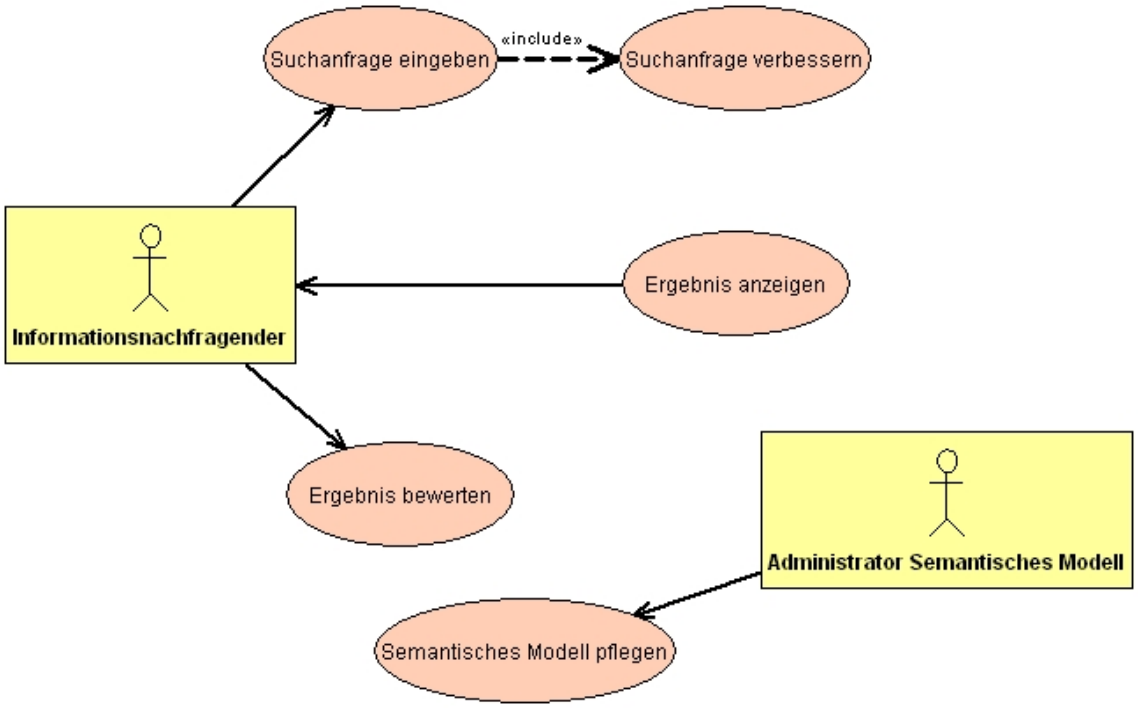

Abbildung 2.5.: Allgemeine Anwendungsfälle

Aufgrund der bisherigen Analyse können nun die Subsysteme des allgemeinen Szenarios bestimmt werden. Wie in Abbildung 2.6 zu sehen ist, kann ein System aus folgenden Teilen bestehen[38]:

- Anfrageermittlung
- Anreicherungssystem
- Semantisches Modell
- Anfrageanpassung

[38] Dabei wird von zeichenkettenbasierten Anfragen ausgegangen.

- Suchsystem
- Auswertung

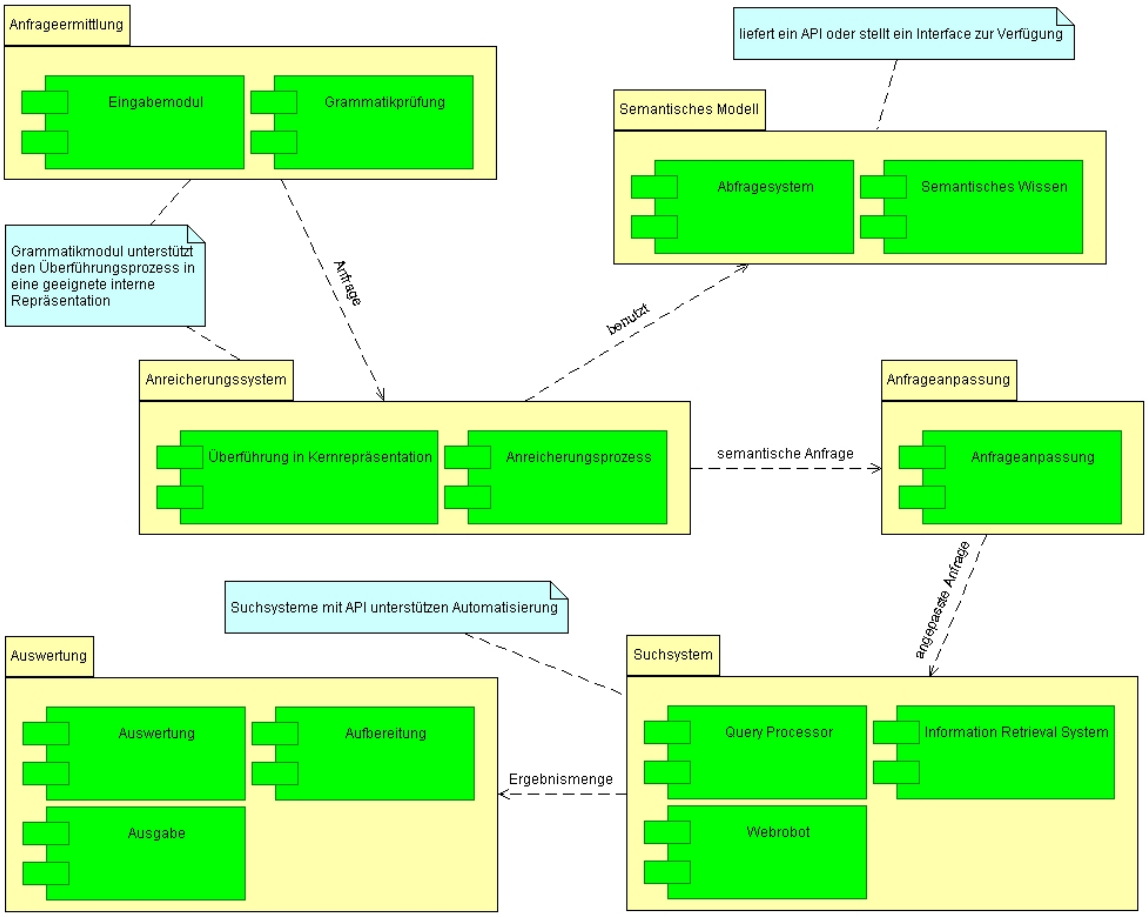

Abbildung 2.6.: Subsysteme im allgemeinen Szenario

Anfrageermittlung Der Informationsbedarf steht sequentiell am Anfang des Systems und wird durch eine Eingabekomponente ausgedrückt[39]. Optional kann die Eingabe einer festgelegten Grammatik gehorchen und darauf geprüft werden. Dadurch ergeben sich innerhalb des Subsystems zwei Komponenten:

- Eingabemodul
- Grammatikprüfung

[39] Es muss sich dabei nicht notwendigerweise um menschliche Eingaben über einen Client handeln.

Anreicherungssystem Das Anreicherungssystem wird die Anfrage zunächst durch eine Umsetzungskomponente in eine systeminterne Kernrepräsentation überführen. Diese Darstellung ermöglicht dann die semantische Anreicherung. Dazu kommuniziert der verantwortliche Anreicherungsprozess mit dem semantischen Modell. Die semantische Ausdruckskraft der ursprünglichen Anfrage wird an dieser Stelle verändert bzw. verbessert. Auch hier lassen sich zwei Komponenten identifizieren.

- Überführung in Kernrepräsentation
- Anreicherungsprozess

Semantisches Modell Das semantische Modell kann vielfältig umgesetzt werden. Topic Maps und Ontologien sind nur zwei der möglichen Modellierungen[40]. Ein Interface ermöglicht den Zugriff auf das semantische Modell. Dieses Subsystem kann ebenfalls in Komponenten unterteilt werden. Dazu bieten sich an:

- Abfragesystem
- Semantisches Wissen

Anfrageanpassung Die Anfrageanpassung muss die angereicherte Anfrage an das jeweilige Suchsystem angleichen. Dabei ist zu berücksichtigen, dass sich Suchsysteme sowohl in der Anfragegrammatik, als auch in der semantischen Ausdruckskraft unterscheiden können (siehe auch Kapitel 2.3). Der Prozess der Anpassung kann automatisiert werden, hängt aber auch jeweils von dem API ab, das das Suchsystem gegebenenfalls zur Verfügung stellt. Eine Unterteilung in verschiedene Komponenten ist nicht unbedingt notwendig.

Suchsystem Das Suchsystem kann z.B. eine der üblichen Suchmaschinen im Internet sein. Hilfreich ist es, wenn sie ein API anbietet. Die Beschreibung der einzelnen Komponenten der Suchsysteme sind dem Abschnitt 2.3 zu entnehmen[41]. Der Vollständigkeit halber hier noch einmal die einzelnen Komponenten:

- Webrobots
- Information Retrieval System
- Query Processor

[40] Theoretisch kann sich die Semantik auch hart codiert im Anreicherungsprozess befinden, so dass die Komponente bei der Umsetzung nicht explizit sichtbar ist.

[41] Im Übrigen sind Suchmaschinen innerhalb des Systems nicht zu beeinflussen, es sei denn, man entwickelt ein eigenes Suchsystem.

Auswertung Gegebenfalls muss die Rückgabe des Suchsystems aufbereitet werden, bevor sie dem Benutzer zur Verfügung gestellt werden kann. Das hängt davon ab, was die Suchsysteme in welcher Form zurückgeben.
Die Auswertung muss nicht maschinell vorgenommen werden. Handelt es sich beim System um ein Programm, das einen Benutzer bei seiner Internetrecherche unterstützt, so wird die Auswertung durch den Benutzer selbst vorgenommen.

Schlussbemerkungen

Es sei noch einmal darauf hingewiesen, dass es sich in diesem Abschnitt um Subsysteme und Komponenten handelt. Diese unterliegen einem Denkmodell, das nicht unbedingt in identischer Weise umgesetzt werden muss[42]. Vor allem der Datenfluss orientiert sich an den Fähigkeiten der einzelnen Moduln und kann real anders gesteuert sein.[43]

[42] Beispiel dafür war das, möglicherweise hart codierte, semantische Modell

[43] Welcher Funktionalität eine eigene Komponente oder gar Subsystem zugesprochen wird, hängt stark von Faktoren, wie Wartbarkeit, Wiederverwendbarkeit usw. ab (siehe auch [Szd01]). Die Granularität kann dabei durchaus variabel sein.

3. Design und Implementierung

3.1. Umsetzung des allgemeinen Szenarios

Ziel des zu erstellenden Systems ist die Verbesserung vager Anfragen (vgl. auch mit Abschnitt 2.2), Dadurch soll die Ergebnismenge optimiert werden. Im Zuge dieser Optimierung wird in Kauf genommen, dass relevante Dokumente entfallen, weil zusätzliche Begriffe gesucht werden, die dort möglicherweise nicht auftauchen[1].
Das semantische Modell wird durch Topic Maps realisiert, die dann auch gepflegt werden müssen, wie dem speziellen Anwendungsfall in Abbildung 3.1 zu entnehmen ist. Ansonsten gelten die Anwendungsfälle aus Abbildung 2.5).

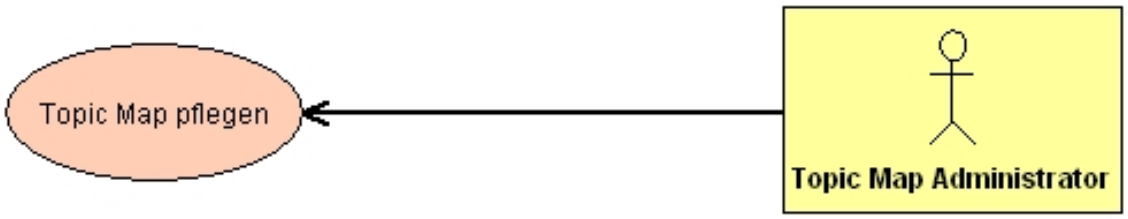

Abbildung 3.1.: Anwendungsfälle des Topic Map Systems

Ausgehend vom allgemeinen Szenario aus Abschnitt 2.5 werden folgende Subsysteme umgesetzt, wie auch Abbildung 3.2 zu entnehmen ist:

- Anfrageermittlung TM
- Anreicherungssystem TM
- Semantisches Modell TM
- Anfrageanpassung TM
- Suchsystem TM
- Auswertung TM

[1] Vgl. auch mit Abbildung 2.2.

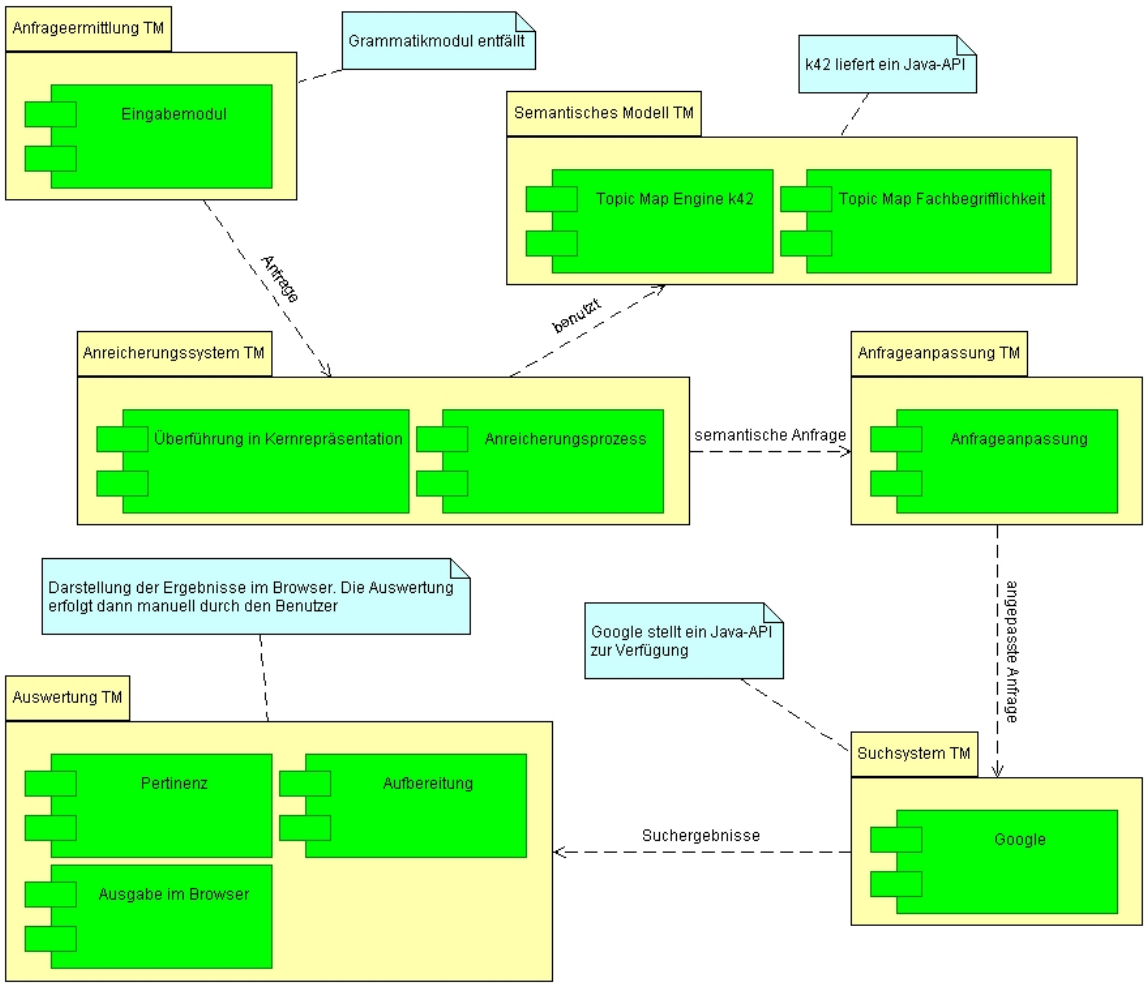

Abbildung 3.2.: Spezielles System

Semantisches Modell TM Da in diesem Projekt die Eignung von Topic Maps den Hintergrund bilden, musste zunächst eine geeignete Topic Map Engine gefunden werden. Die Wahl fiel schließlich auf *k42*. Kriterien[2] für die Auswahl der Engine für den speziellen Einsatz in diesem Projekt waren:

- Verfügbarkeit
- Umsetzbarkeit: Die Topic Map Engine spielt in dieser Arbeit nur eine untergeordnete Rolle. Darum sollte sie nach Möglichkeit einfach anzubinden sein. Das Java API, das k42 mitliefert, erfüllt diese Kriterien. Des Weiteren liegt eine musterhafte Anbindung in Form der Diplomarbeit von Roland Herrmann vor (vgl. mit [Her04]).

[2] Vgl. mit Abschnitt 2.4.

- Die Abfragemöglichkeiten sind ebenfalls durch das API gewährleistet.
- Schließlich sollte die Engine XTM als Topic Map Standard benutzen, was k42 ebenfalls leistet.

Ob k42 für größere Projekte, die real eingesetzt werden sollen, tauglich ist, wurde nicht untersucht. Es ist auch nicht weiter relevant, da die Entwicklung von k42 inzwischen eingestellt wurde. Für dieses Projekt ist es dennoch eine gute und vor allem unkomplizierte Wahl, da der Autor bereits Erfahrungen mit der Engine hat[3].
Durch den Einsatz von k42 stellt sich auch die Frage der einzusetzenden Programmiersprache nicht mehr. Java ist modern und aktuell und die Entscheidung, sie einzusetzen, ist akzeptabel.
Das semantische Wissen wird folglich durch den Einsatz einer Topic Map erreicht. Die Fachlichkeit wird, wie beim Projekt SHOE, eingeschränkt. Außerdem soll beim Design der Topic Map darauf geachtet werden, dass sie austauschbar bleibt und die Persistenzschicht dadurch insgesamt flexibel ist.

Anfrageermittlung TM Innerhalb des Subsystems Anfrageermittlung wird die Komponente *Eingabemodul* in Form eines Java-Clients realisiert. Er nimmt die Anfrage des Akteurs *Informationsnachfragender* entgegen und leitet sie zur Verarbeitung an das Anreicherungssystem weiter.
Die Komponente Grammatikprüfung entfällt, da dem Benutzer für dieses Projekt eine Grammatik zur Verfügung gestellt wird, an der er sich orientieren kann[4].

Anreicherungssystem TM Das Anreicherungssystem nimmt Anfragen entgegen. Die Vorbedingung dafür ist, dass sie grammatikalisch in Ordnung sind. Die jeweilige Anfrage wird dann durch einen Scanner in Token zerlegt, die schließlich, unter zu Hilfenahme des semantischen Modells, durch einfache Ersetzungen semantisch angereichert werden.

Anfrageanpassung TM Google stellt ein API zur Verfügung, das benutzt wird, um einen Suchauftrag außerhalb der typischen Thin-Client Suchumgebung abzusetzen. Die vom Benutzer eingegebene Anfrage wird daher mittels API-Funktionen an Google zur Auswertung weitergeleitet. Dafür ist eine, auf Google spezialisierte, Suchklasse innerhalb des Modells zuständig, die die Anfrage an Google anpasst.

[3] Für Systeme, die nicht prototypisch sind, ist wahrscheinlich *TM4J* zu bevorzugen. Hier wird auch ein TMAPI mitgeliefert.

[4] Es sollte stets bedacht werden, dass dieses prototypische System nicht für jedermann zur Verfügung stehen soll und daher die Verantwortung an den Benutzer delegiert werden kann.

Suchsystem TM Nach gründlicher Überprüfung möglicher Suchsysteme soll Google zum Einsatz kommen[5]. Die Wahl fiel nicht auf eine der Metasuchmaschinen, da diese bis auf wenige, die Anfrage ungefiltert weiterleiten. Da nicht deutlich wird, wie die jeweilige Suchmaschine mit der Anfrage umgeht, sind die Antworten sehr schwer einzuschätzen. Auch MetaGer, als innovative Suchmaschine ist zurzeit noch nicht in der Lage eine vollständige Anpassung an eingesetzte Suchsysteme zu bieten. Dazu wäre eine Umsetzung des Suchwortes an alle Sprachen angefragter einfacher Suchsysteme notwendig. Aufgrund der Betriebsgeheimnisse, grade im Bereich der Algorithmen bei Suchsystemen, scheint das im Moment nicht möglich zu sein. Auf den Vorteil einer Metasuchmaschine wird daher bewusst verzichtet, zumal es für die Ziele dieser Arbeit nicht absolut notwendig ist.
Alle im Vorwege getesteten, einfachen Suchmaschinen hatten erhebliche Mängel bei der Umsetzung einer hinreichenden boolschen Suche. Insbesondere der Einsatz von Klammern ist nur selten möglich.
Google hat, wie die Fallstudie zeigte, ebenfalls Schwächen[6]. Beispielsweise war der Einsatz des + erklärungsbedürftig und die Auswertungsreihenfolge der boolschen Operatoren folgt nicht der mathematischen Vorgehensweise.
Dennoch kann eine Grammatik angegeben werden, die von Google verstanden wird, auch wenn sie die Sprache von Google nicht vollständig erfasst. Durch diese Einschränkung wird es möglich, Google kontrolliert einzusetzen. Diese Vorgehensweise soll hier angewendet werden.
Des Weiteren spricht für Google das bereits aufgeführte Java-API, das trotz seiner Einschränkungen für den praktischen Einsatz brauchbar ist.

Auswertung TM Das Ergebnis, das Google zurückliefert, wird aufbereitet und als HTML-Datei auf einem Datenträger hinterlegt. Anschließend wird aus dem Java-Programm heraus ein Browser mit der zuvor gespeicherten HTML-Datei geöffnet und dem Benutzer präsentiert. Nun kann das Ergebnis, ähnlich der normalen Benutzung von Google, verwendet und ausgewertet werden. Pertinenz fällt nicht in den Bereich der automatisierten Bearbeitung, wurde aber der Vollständigkeit halber in das System übernommen.

Sequentielle Interaktion Wie die einzelnen Subsysteme in der Anwendung sequentiell zusammenarbeiten, kann Abbildung 3.3 entnommen werden. Die Steuerung des Systems ist eher eine Implementierungsentscheidung, hätte aber auch als Subsystem formuliert werden können. Aus Gründen der Lesbarkeit findet sie sich ebenfalls in der Abbildung wieder.

[5] Zunächst war geplant, MetaGer einzusetzen [Uni05b].
[6] Jedoch ist Google die marktführende Suchmaschine und man sollte gute Gründe haben, um sie zu ignorieren.

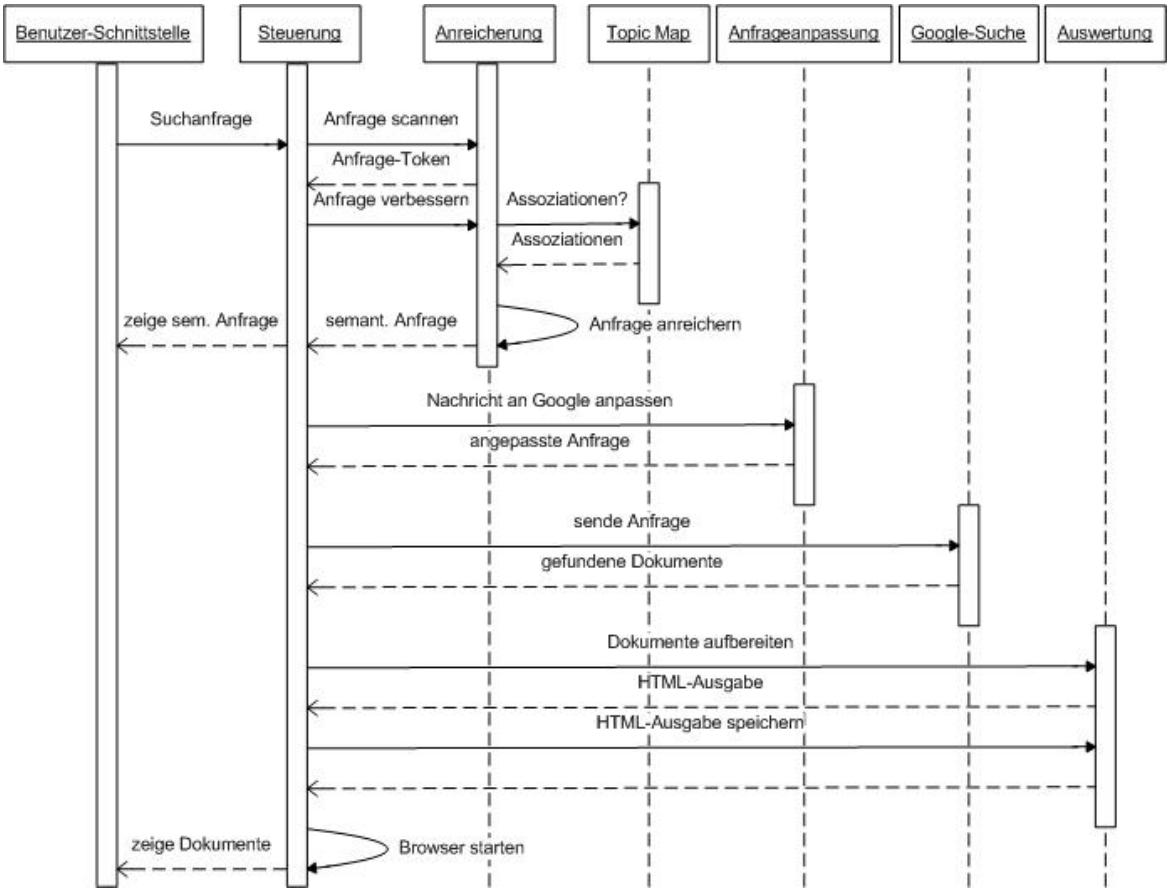

Abbildung 3.3.: Sequentieller Ablauf

Darüber hinaus wurde wegen der Übersichtlichkeit innerhalb des Sequenz-Diagramms bewusst auf die Unterteilung in Komponenten verzichtet, zumal es den Wissenstransfer ohnehin nicht entscheidend angehoben hätte.

3.2. Entwicklung der Anwendung

Die Anwendung, die in diesem Abschnitt beschrieben wird, erfüllt mehrere Funktionen:

- Sie stellt einen Client zur Verfügung, über den der Benutzer seine Eingaben absetzen kann (vgl. mit Abbildung 3.7).
- Ein Scanner zerlegt die Eingabe in Token, damit die einzelnen Suchbegriffe mit Hilfe der Topic Map bearbeitet werden können.
- Das Resultat der Bearbeitung wird zur Kontrolle im Client ausgegeben.
- Eine Anreicherungsroutine pflegt die Rückgaben in die ursprüngliche Eingabe ein.
- Eine weiterer Teil sorgt für die Anpassung einer, von Google erwarteten, API-konformen Anfrage.
- Die Antworten werden dann als strukturierte HTML-Seiten persistent gemacht.
- Schließlich wird ein Browser mit der hergestellten HTML-Seite aufgerufen.

Design

Da die Eingaben des Benutzers über eine Client-Applikation erfasst werden, bietet es sich an, dem Anwendungsdesign eine klassische Model-View-Controller-Architektur ([Oes98]) mit Observer-Funktionalität zu Grunde zu legen.
Der Client besteht aus zwei Komponenten zur Ein- und Ausgabe der ursprünglichen und der bearbeiteten Anfrage, sowie aus einem Button zur Aktivierung des Bearbeitungsvorgangs.
Das Modell besteht aus mehreren Teilen. Zunächst gibt es eine Workflow-Klasse, die als Mediator ([GHJV96]) fungiert, also den Programmablauf koordiniert und dazu geeignete Methoden aufruft.
Das Anreicherungssystem wird durch eine Klasse umgesetzt, die die notwendige Funktionalität zur Verfügung stellt, wie z.B. den Scanner und den Anreicherungsprozess selbst.
Für den Zugriff auf die Topic Map Engine k42, als auch für die Anbindung des Google-API werden Adapter-Klassen ([GHJV96]) eingesetzt. Auf dieser Ebene können bei Bedarf analog andere Topic Map Engines bzw. Suchmaschinen-APIs angebunden werden.
Für die Ausgabe gibt es eine Klasse, die lediglich die Rückgabe von der Suchmaschine aufbereitet und persistent macht. Sie arbeitet nach dem Prinzip des Strategy-Pattern [GHJV96][7], das auch für beliebige andere Suchmaschinen implementiert werden kann. Die aufbereitete Ausgabe kann danach aus der Applikation heraus aufgerufen und dem Nutzer präsentiert werden.

[7] Weitere Informationen zu Mustern, Werkzeugen, Kopplung usw. finden sich bei [Rie97].

Implementierung

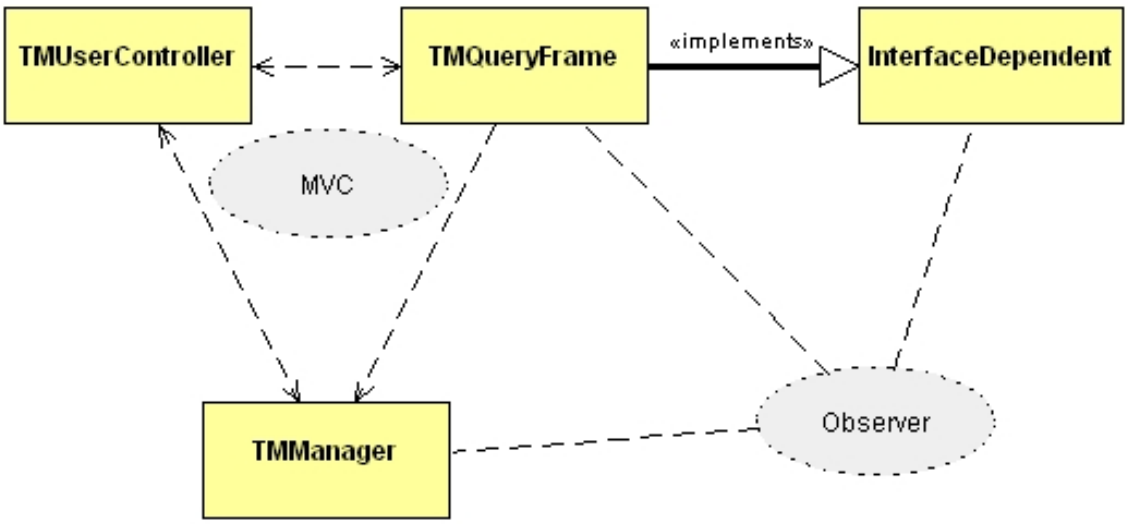

Abbildung 3.4.: Model-View-Controller

In der Umsetzung wird die Model-View-Controller-Architektur durch ein einfaches Interface mit Namen *InterfaceDependent* unterstützt, wie man auch in Abbildung 3.4 erkennen kann. Mittels der Methode *update* wird dann das Observer-Pattern realisiert. Die Kopplung zwischen dem Model (TMManager) und dem View (TMQueryFrame) ist dadurch relativ locker und es können bei Bedarf einfach weitere Views hinzugefügt werden.
Das Model-View-Controller-Konzept wird von den Java-Swing-Klassen leider nicht unterstützt, weshalb der Controller (TMUserController) innerhalb der Anwendung bis auf wenige Basis-Methoden redundant ist. Der Controller wurde jedoch dennoch aufgenommen, falls von Swing im Nachhinein abgewichen wird.

Das Model der Anwendung besteht, wie oben erwähnt, nicht nur aus dem TMManager. In Abbildung 3.5 erkennt man alle Klassen, die am Model beteiligt sind.
Der TMQueryImprover sorgt lediglich für einen sauberen Startpunkt, der das Programm aktiviert und die Kontrolle dann an die Workflow-Klasse TMManager abgibt. TMManager fungiert als Mediator-Klasse.
TMk42Access ist der Adapter, der das Java-API von k42 an das Model ankoppelt. Es wird jedoch nicht nur die Funktionalität adaptiert, sondern es werden darüber hinaus ein paar Funktionen hinzugefügt, die die Abfrage der hinterlegten Topic Map erleichtern.
TMQueryEnrichment ist die Klasse, die für die Anreicherung zuständig ist. Sie beinhaltet den Scanner, der die Anfrage in Token zerlegt. TMQueryEnrichment hat darüber hinaus Zugriff auf TMk42Access und fragt darüber für jedes Token die Topic Map ab. Abschließend wird dann die Rückgabe in die Anfrage eingebaut.

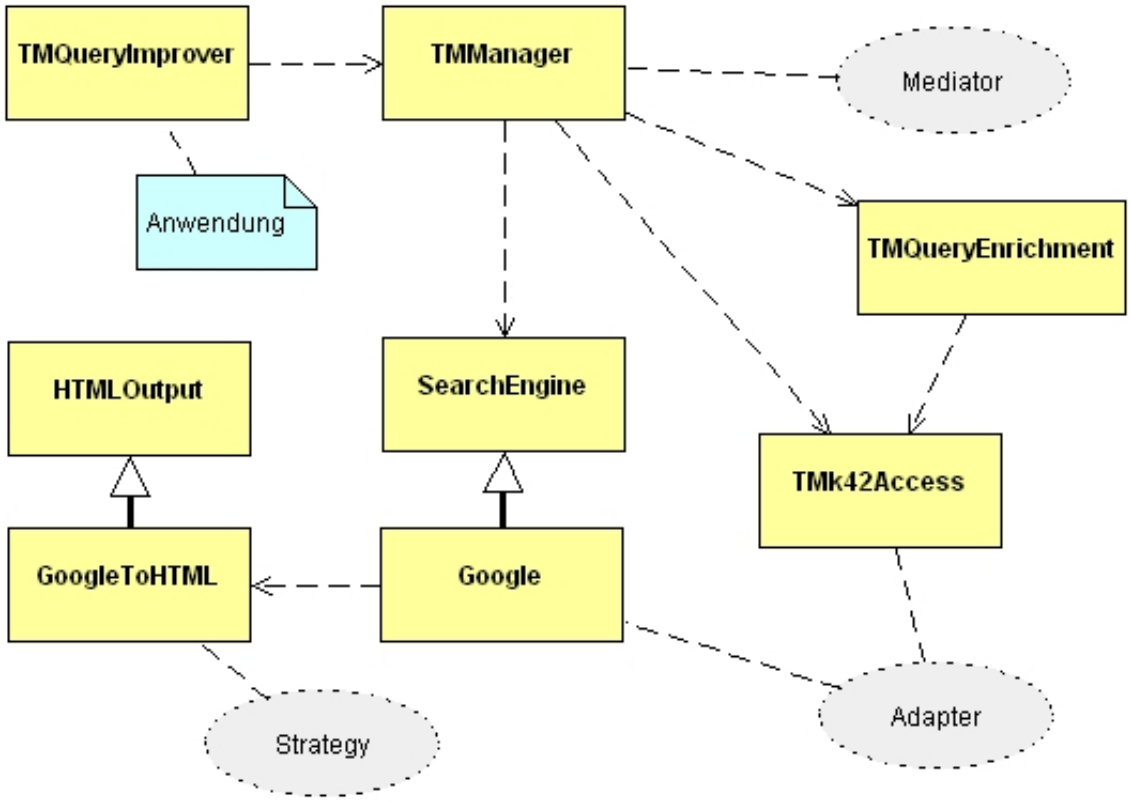

Abbildung 3.5.: Model

SearchEngine ist die abstrakte Oberklasse und somit die Klasse, von der konkrete Suchmaschinenklassen abgeleitet werden können, um das System an bestimmte Suchmaschinen anzupassen. Diese Suchmaschinenklassen, wie hier Google, passen die Anfragen an die, von der Suchmaschine erwartete, Sprache an. Des Weiteren werden, bei vorhandener API, Funktionen formuliert, die das Absetzen der Anfrage von innerhalb der Anwendung gestatten.
GoogleToHTML bearbeitet die Rückgabe von Google und hinterlegt eine HTML-Datei auf der Festplatte. TMManager präsentiert dem Benutzer dann die generierte HTML-Seite, indem es einen Browser aufruft und die HTML-Datei als Parameter liefert.

Angaben zur Implementierung Das System wurde erfolgreich praktisch umgesetzt. Die Implementierung wurde dabei unter folgenden Bedingungen realisiert:

- Standard-Rechner mit Windows XP Professional
- Eclipse[8], Version 3.0.1 (Code-Entwicklung)
- Java, JRE System Library, Version 1.5.0
- Topic Map Engine: k42(tm) v1.1.1 by empolis UK Ltd.
- Google API, Beta 2 - August, 2002
- OTW 2.4, Objekttechnologie-Werkbank (Dokumentation, Metrik-Analyse)
- Microsoft Office Visio 2003 (Dokumentation)
- Javadoc (Dokumentation)
- Advanced Installer 2.6.4 (Deployment)[9]

[8] URL: `http://www.eclipse.org/platform/`.
[9] Testversion.

3.3. Grammatik

Wie im Kapitel 2.3 herausgearbeitet wurde, kann Verantwortlichkeit an den Benutzer des Systems delegiert werden, indem die einzugebende Sprache vorgeschrieben wird. In erweiterter BNF wird deshalb die zu benutzende Grammatik für *TopicSeek* angegeben, die dann von Google sinnvoll verarbeitet wird:

```
<Zeichen> ::= 'a' | ... | 'z' | 'A' | ... | 'Z' | '0' | '1' | ... | '9'
<Trennzeichen> ::= ' '
<Stringbegrenzer> ::= '"'
<NOT> ::= '-'
<BinärerOp> ::= 'AND' | 'OR' | '|'
<Nichts> ::= ''
<AlleZeichen> ::= <Trennzeichen> | <Zeichen>
<UnärerOp> ::= <NOT>
<BinärerOpUndTrennung> ::= <BinärerOp><Trennung> | <Nichts>
<Zeichenfolge> ::= <Zeichen><Zeichenfolge> | <Zeichen>
<BeliebigeZeichenfolge> ::= <AlleZeichen><BeliebigeZeichenfolge> |
<AlleZeichen>
<String> ::= <Stringbegrenzer><BeliebigeZeichenfolge><Stringbegrenzer>
<Trennung> ::= <Trennzeichen><Trennung> | <Trennzeichen>
<UnärerAusdruck> ::= <Zeichenfolge> | <UnärerOp><Zeichenfolge> | <String> |
<UnärerOp><String>
<Ausdruck> ::= <UnärerAusdruck><Trennung><BinärerOpUndTrennung><Ausdruck> |
<UnärerAusdruck>
```

◇

1 BEMERKUNG *UnärerAusdruck* der obigen Grammatik darf dabei weder genau *AND* noch genau *OR* ergeben.
Wird der binäre Operator weggelassen, so ergänzt Google die Lücke automatisch mit dem logischen *AND*.
Das Zeichen - steht bei Google für das logische *NOT*. Das Zeichen | ist bei Google äquivalent zum logischen *OR*.
◇

Aus obiger Grammatik können nun einfache Anfragen abgeleitet werden:

3.3.1 BEISPIELE (EINFACHE ANFRAGEN)

Java

```
Java | Fortran
Java AND Fortran
Java Fortran
Java Smalltalk OR Fortran
Java -Fortran
Java OR -Fortran AND Smalltalk
```

◇

Mit dieser Grammatik steht nun eine Sprache zur Verfügung, mit der man praktischen Nutzen erzielen kann. Es galt, möglichst abstruse Konstruktionen auszuschließen, die Google nur deshalb akzeptiert, um dem Benutzer den Umgang mit Google zu erleichtern. Durch obige Grammatik liegt für den Prototyp eine Vorbedingung fest, so dass auf eine Grammatikprüfung innerhalb der Anwendung verzichtet werden kann. Sinn des Prototyps ist nicht der praktische Einsatz, sondern der Nachweis der Machbarkeit. Darüber hinaus unterstützt er die Tests und damit die Einschätzung der Leistung der zugrunde liegenden Topic Map.

3.4. Entwicklung der Topic Map

Wie in Abschnitt 2.4 zu sehen ist, kann die Entwicklung und der Einsatz von Topic Maps in 7 Phasen aufgeteilt werden. Diese Phasen sind zum Teil für große Projekte gedacht. Die einzelnen Phasen und die Notwendigkeit der Umsetzung innerhalb des Prototyps im Rahmen dieser Arbeit sind Tabelle 3.1 zu entnehmen. Dabei ist wieder zu beachten, dass es sich um eine Lösung handelt, die den Nachweis der Machbarkeit führt. Die Anforderungen an die Topic Map und das Umfeld sind deshalb relativ gering.

Phase	Umsetzung
Analyse	benötigt (siehe unten)
Design	benötigt (siehe unten)
Erstellung	partiell benötigt (siehe unten)
Speicherung	entfällt
Administration	entfällt
Publikation	entfällt
Verwendung	benötigt (siehe unten)

Tabelle 3.1.: Entwicklungsphasen der Topic Map im Rahmen der prototypischen Umsetzung

Analyse

Das technische Umfeld ist relativ spartanisch und nicht von größerer Bedeutung. Als Topic Map Engine wird k42 eingesetzt. Die Engine akzeptiert XTM. Das zugehörige Java-API erlaubt den Einsatz aller gängigen Konzepte von Topic Maps (vgl. mit Abschnitt 2.4).
Die Fachbegrifflichkeit wird eingeschränkt. Prinzipiell sollte sie frei wählbar sein. Somit ist das System flexibel einsetzbar. Für diese Arbeit werden beispielhaft Ausschnitte aus der Informatik ausgewählt. Geeignete Begriffe findet man z.B. bei *ACM*, in Form des 'ACM Computing Classification System 1998'. Aus dieser Liste werden auszugsweise Wörter entnommen und innerhalb der Topic Map zueinander in Beziehung gesetzt.

Design

Das Design der Topic Map wird entscheidend durch das Konzept der Associations bestimmt. Ziel ist es, die Anzahl der Rückgabewerte der Suchmaschine einzuschränken. Dafür sind die Operatoren *AND* und - (logisches NOT) wichtig. Mit ihrer Hilfe kann die Lösungsmenge verkleinert werden.

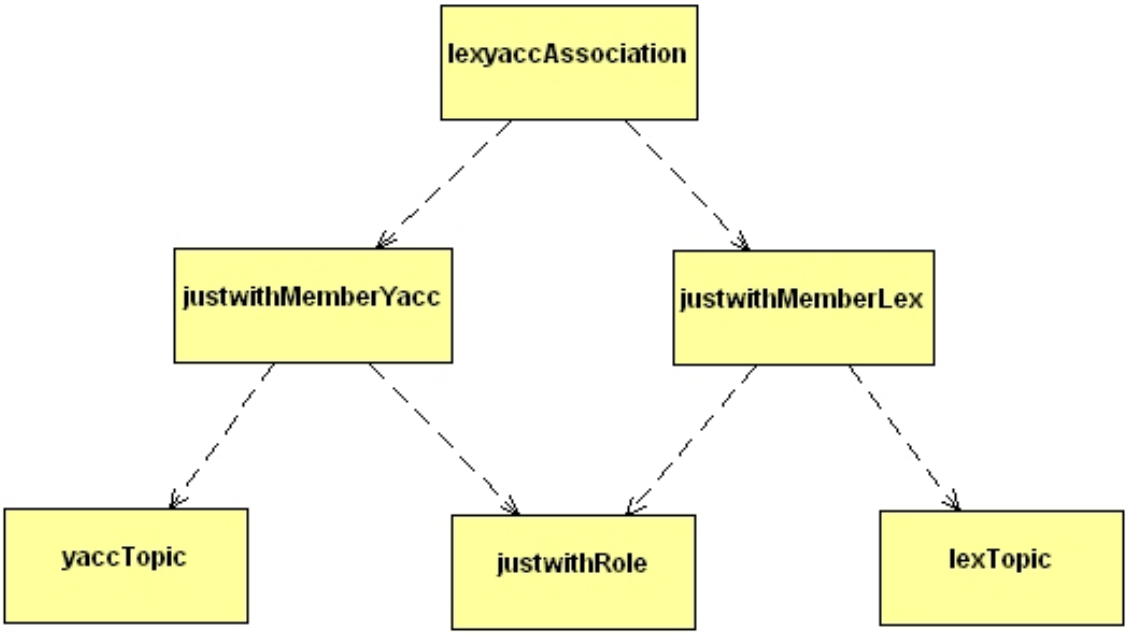

Abbildung 3.6.: Topic Map Diagramm

Das Diagramm 3.6 zeigt am Beispiel der *lexyaccAssociation*, auf welche Weise Assoziationen, Topics und ihre Rollen in dieser Topic Map zusammenarbeiten.
Dabei sollen die Begriffe *lex* und *yacc* so zueinander in Beziehung gesetzt werden, dass eine Anfrage, die nur einen der beiden Begriffe beinhaltet, durch den anderen ergänzt wird (z.B. wird *lex* zu *lex AND yacc*). Dafür werden mit *lexTopic* und *yaccTopic* zwei neue Topics erstellt. Diese werden mit der Rolle *justwithRole* zu Membern zusammengefasst (die allerdings im Gegensatz zum Diagramm 3.6 keine Bezeichnung haben) und dann der *lexyaccAssociation* zugeordnet. *lexyaccAssociation* ist dabei vom Typ *justwithAssociation*. Alternativ kann man die Umsetzung dem folgenden Code-Fragment entnehmen:

```
<association id="lexyaccAssociation">
 <instanceOf><topicRef xlink:href="#justwithAssociation" /></instanceOf>
 <member>
  <roleSpec><topicRef xlink:href="#justwithRole" /></roleSpec>
  <topicRef xlink:href="#lex" />
 </member>
 <member>
  <roleSpec><topicRef xlink:href="#justwithRole" /></roleSpec>
  <topicRef xlink:href="#yacc" />
 </member>
```

```
</association>
```

Diese Art der Umsetzung kann jetzt für beliebige Wörter, die in Zusammenhang stehen sollen, wiederholt werden. Für einen anderen logischen Operator werden entsprechend auch die Assoziationen und Rollen ausgetauscht.
Fachliteratur wird häufig mehrsprachig abgefasst, was natürlich nicht nur für die Informatik gilt[10]. Daher wurde ein weiteres Konzept von Topic Maps, namens *Scope*, eingesetzt. Mit Hilfe von Scopes können nun Begriffe in verschiedenen Sprachen abgelegt werden. Wie das umgesetzt wird, wurde bereits in Kapitel 2.4 beschrieben und soll hier nicht wiederholt werden.

Erstellung

Dieses Projekt wird neu erstellt. Es gibt daher keinerlei Problemstellungen, wie dem Zusammenführen von Wissen oder der Transformation von vorhandenen Daten. Da das Projekt nicht für den praktischen Einsatz geplant ist, sind auch Rechte und Aufgaben von Benutzergruppen redundant.
Die Fachbegriffe, die ausgewählt werden, können bei diesem Prototypen ganz einfach manuell in die Topic Map eingepflegt werden.

Verwendung

Die Navigation und Abfrage der Topic Map geschieht mittels der Topic Map Engine k42, die als Java API vorliegt. Die Verwendung kann in mehrere Schritte aufgeteilt werden:

- Soll mit Internationalisierung gearbeitet werden, so wird zunächst der Scope ermittelt, innerhalb dem ein Begriff in der Topic Map gefunden wurde. Damit steht dann auch fest, in welcher Sprache die Antwort erfolgen soll.
- Innerhalb des Prototyps kann die Sprache bereits im Client ausgewählt werden. Intern wird dann ein entsprechender Scope hergestellt, um dann die Topic Map adäquat abzufragen.
- Danach werden zu einem Begriff x (Topic) diejenigen Begriffe gefunden, die zu x in Beziehung stehen und die, dem jeweiligen Scope, entsprechenden Begriffe zurückgegeben.
- Schließlich wird innerhalb der Applikation, je nach Art der logischen Verknüpfung, die Anfrage für die gefundenen Begriffe angereichert (Näheres dazu in Abschnitt 3.2).

[10] Beispielsweise muss man in der Ägyptologie häufig auf französischsprachige Literatur zurückgreifen.

3.5. Test

Rahmenbedingungen

Wie bereits in der Analyse im Abschnitt 2.2 erwähnt wurde, wird hier eine summative Evaluierung mit Hilfe von Pertinenz durchgeführt. Dazu wurde die Topic Map mit Begriffen gefüllt, die dem 'ACM Computing Classification System 1998' entnommen sind[11]. Tabelle 3.2 zeigt, welche Begriffe, die getestet werden, wie in Beziehung gesetzt wurden.

logische Operation	Begriffe
AND	Parallelism Concurrency
AND	Alternation Nondeterminism
AND	Reducibility Completeness
AND	EDI Electronic Data Interchange
AND	DDL Data Description Language
AND	Controller MVC
NOT	Controller Channel

Tabelle 3.2.: Begriffe und ihre Beziehung

2 BEMERKUNG Bei der Entscheidung, welche Begriffe zusammengehören und welche nicht, muss man mit unerwarteten Dingen rechnen. Beispielsweise gibt es einen *16-port Asynchronous Communications Multiplexer*, der mit *The Macrolink MVC 16-port Communications Multiplexer* bezeichnet ist. Die entsprechende Webseite beinhaltet sowohl den Begriff *Controller*, als auch *MVC*. Deswegen wird hier exemplarisch *channel* von *Controller* getrennt, um diese Webseite ebenfalls zu filtern.
Ob das notwendig ist, sei dahingestellt, da in diesem Fall grade mal ein Dokument ausgeschlossen wird. Allerdings kann es auch Fälle geben, bei denen diese Vorgehensweise deutlich mehr Dokumente ausschließen würde.

[11] MVC ist dabei die einzige Ausnahme.

Als Suchmaschine wird Google eingesetzt. Von Interesse soll sein, wie viele Treffer Google zum jeweiligen Thema innerhalb der ersten zehn Suchergebnisse liefert. Dabei werden die Antworten in jene mit Themenbezug und solche ohne Themenbezug unterschieden.

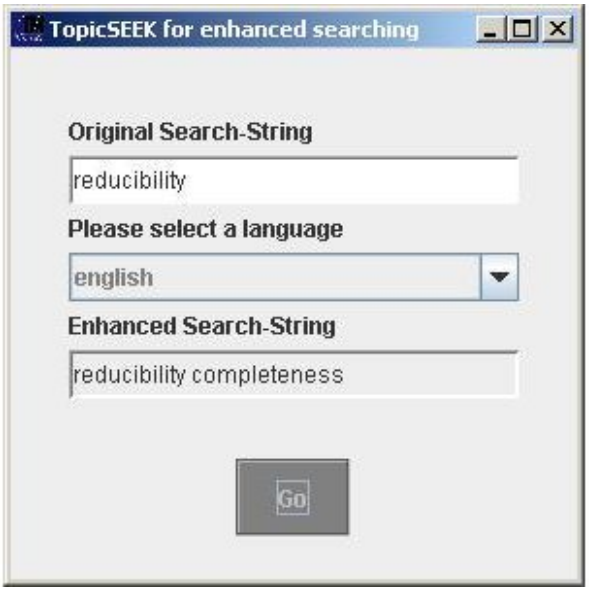

Abbildung 3.7.: Client Anwendung

Der Test wurde mit *TopicSEEK* für die angereicherte Suchanfrage durchgeführt (siehe auch Abbildung 3.7), danach ausgewertet und schließlich den ausgewerteten Ergebnissen der normalen Google-Suche mit nicht angereicherten Suchanfragen gegenübergestellt.

Testergebnisse

Tabelle 3.3 zeigt die Ergebnisse der einzelnen Tests.

Die Testergebnisse aus Tabelle 3.3 zeigen, dass sich durch Anreicherung des Strings prinzipiell Erfolge erzielen lassen. Unabhängig davon, wie viele themenverwandte Treffer in der normalen Variante erzielt wurden, konnte bei den Testbeispielen durchgehend eine Steigerung in die Nähe von 100 Prozent erreicht werden.
Auffällig ist dabei der Begriff *Reducibility*, der bereits in der unangereicherten Variante neun Treffer mit Themenbezug ergab. Offensichtlicht gilt das für all jene Begriffe, die fast ausschließlich im Zusammenhang mit ihrer Fachlichkeit auftauchen. *Reducibility* ist ein gutes Beispiel dafür.
Anders sieht das für *Parallelism* (siehe auch Abbildung 3.8) und *Alternation* aus, deren thematischer Bezug erst mit Hilfe von *Concurrency* (siehe auch Abbildung 3.9) bzw. *Nondeterminism* hergestellt wird.

Anfrage	Status	mit Bezug
“parallelism“	normal	0
“parallelism concurrency“	angereichert	10
“alternation“	normal	0
“alternation nondeterminism“	angereichert	10
“reducibility“	normal	9
“reducibility completeness“	angereichert	10
“edi“	normal	4
“edi electronic data interchange“	angereichert	9
“ddl“	normal	1
“ddl data description language“	angereichert	8
“controller“	normal	0
“controller mvc -channel“	angereichert	9

Tabelle 3.3.: Treffer mit Themenbezug für die ersten 10 Google-Resultate vom 20 Mai 2005.

Ebenfalls wurde das System erfolgreich für Abkürzungen getestet. So ist bei bloßer Eingabe von *EDI* unklar, ob sich die Anfrage nicht an das *Eidgenössische Departement des Innern* richtet. Und bei DDL könnte es sich auch um *Digital Direct for Linux* handeln. Eine Ergänzung mit dem ausgeschriebenem Inhalt könnte also dem Benutzer viele themenfremde Treffer ersparen.
Begriffe, sinnvoll mit dem logischen *AND* verknüpft, bilden also in einigen Fällen semantisch hochwertigere Einheiten, als die einzelnen Begriffe für sich genommen. Andererseits darf man auch nicht zu viele Begriffe gleichzeitig verknüpfen, damit nicht die Gefahr besteht, eine hohe Anzahl von relevanten Dokumenten zu verfehlen. Stattdessen kann man mit der logischen Verknüpfung *NOT* einen Begriff ausschließen, der bei irrelevanten Themen mit hoher Wahrscheinlichkeit auftaucht. Dadurch kann man ebenfalls gute Ergebnisse erzielen.
Aufgrund der Tests kann man das System als brauchbar bewerten. Die Fähigkeiten des Systems hängen jedoch sehr stark von der Güte der Semantik ab, die die in Beziehung gesetzten Begriffe auszudrücken vermag. Das bedeutet, dass die Herstellung der Topic Map sehr sorgfältig geschehen muss.

parallelism - [Diese Seite übersetzen]
... **parallelism** of words: She tried to make her pastry fluffy, sweet, and delicate. **parallelism** of phrases: Singing a song or writing a poem is joyous. **...**
humanities.byu.edu/rhetoric/Figures/P/**parallelism**.htm - 5k - Im Cache - Ähnliche Seiten

Parallel Form - [Diese Seite übersetzen]
... Faulty **Parallelism**, Corrected Version. Formerly, science was taught by the **...** Faulty **Parallelism**, Corrected Version. My income is smaller than my wife. **...**
webster.commnet.edu/grammar/**parallelism**.htm - 8k - Im Cache - Ähnliche Seiten

Parallelism - [Diese Seite übersetzen]
... If you would like to review **Parallelism** before taking this quiz (or at any time during it), click HERE. 1. Which of the following sentences is **...**
webster.commnet.edu/grammar/quizzes/niu/niu10.htm - 10k - Im Cache - Ähnliche Seiten
[Weitere Ergebnisse von webster.commnet.edu]

Parallel Structures - [Diese Seite übersetzen]
... Changing to another pattern or changing the voice of the verb (from active to passive or vice versa) will break the **parallelism**. **...**
owl.english.purdue.edu/handouts/grammar/g_parallel.html - 16k - Im Cache - Ähnliche Seiten

LEO: **Parallelism** - [Diese Seite übersetzen]
... Simple **Parallelism**. **Parallelism** using Common Connectors **...** A slightly different **parallelism** involves the common connectors-- either/or, neither/nor, **...**
leo.stcloudstate.edu/grammar/**parallelism**.html - 6k - Im Cache - Ähnliche Seiten

The Journal of Instruction-Level **Parallelism** - [Diese Seite übersetzen]
... journal dedicated to soliciting, thoroughly reviewing, and publishing state-of-the-art papers in all areas of instruction-level **parallelism** (ILP.) **...**
www.jilp.org/ - 12k - Im Cache - Ähnliche Seiten

CATHOLIC ENCYCLOPEDIA: **Parallelism** - [Diese Seite übersetzen]
Visit New Advent for the Summa Theologica, Church Fathers, Catholic Encyclopedia and more.
www.newadvent.org/cathen/11473a.htm - 17k - Im Cache - Ähnliche Seiten

Skeletal **Parallelism** - [Diese Seite übersetzen]
The Skeletal **Parallelism** homepage has moved to. http://homepages.inf.ed.ac.uk/mic/Skeletons/
www.dcs.ed.ac.uk/home/mic/skeletons.html - 1k - Im Cache - Ähnliche Seiten

The Writing Center at the University of Colorado at Colorado **...** - [Diese Seite übersetze
... The following examples illustrate faulty and correct **parallelism**: **...** In each of the examples of faulty **parallelism**, one of the elements doesn't follow **...**
www.uccs.edu/~wrtgcntr/handouts/**parallelism**.html - 4k - Im Cache - Ähnliche Seiten

PARALLELISM - [Diese Seite übersetzen]
... buy stuff - shop now open U-Sound, Vol. 1 double CD (PAR 010-2) now available. an X-Ray Vision web site All content ©2004 **Parallelism**.
www.**parallelism**.com/ - 6k - Im Cache - Ähnliche Seiten

Abbildung 3.8.: Test-Ergebnis für *parallelism* vom 20 Mai 2005

Parallel Scientific Computing
... **Parallelism, Concurrency** and Dependability. 9. Monte Carlo and p. Area = p.
2. 2. **Parallelism, Concurrency** and Dependability ...
http://www.cs.haverford.edu/courses/CMSC100/parallel.ppt

Ch. 1 Introduction **Concurrency, parallelism Concurrency** and ...
... **Concurrency, parallelism**. " Sequential programs. □ one thread of control.
" Concurrent programs. □ multiple threads of control. □ communication ...
http://www.cs.helsinki.fi/u/alanko/rio/S02/kalvokopiot/ch1_p6.pdf

Parallelism/Concurrency specification within UML
... **Parallelism/Concurrency** specification within UML. Sébastien Gérard (CEA-LIST: Sebastien.Gerard@cea.fr). Ileana Ober (Telelogic:Ileana. ...
http://wooddes.intranet.gr/uml2001/WhitePaper/WhitePaperOnParallelism.pdf

Bill Clementson's Blog
... Wikipedia definitions for **parallelism, concurrency** and distribution that ...
Parallelism and **concurrency** are intimately related, but it is frequently ...
http://home.comcast.net/~bc19191/blog/050125.html 24k

Profiling Grid Data Transfer Protocols and Servers
... This study shows that increased **parallelism** or **concurrency** ... The users should select the correct **parallelism** or **concurrency** level ...
http://www.cs.wisc.edu/condor/stork/papers/profiling-europar2004.pdf

Profiling Grid Data Transfer Protocols and Servers George Kola ...
... This study shows that increased **parallelism** or **concurrency** level does not necessarily ... The users should select the correct **parallelism** or **concurrency** ...
http://www.cs.wisc.edu/condor/stork/papers/profiling-europar2004.ps

Date: Mon, 15 Jul 1996 15:58:35 -0700 From: George Forman <forman ...
... Identifiers: schedule activations. user-level management. **parallelism**. **concurrency**. parallel programming. operating system kernel. user-level library ...
http://www2.cs.washington.edu/uns/apps/inspec2bibtex.txt 6k

Algorithmics, 3rd ed.
... **Parallelism, concurrency** and alternative models ... was previously titled "**Parallelism** and **Concurrency**" and is now called "**Parallelism, Concurrency**, ...
http://www1.idc.ac.il/yishai/algo3.htm 22k

1. What system requirements do you have - eg Unix/Linux, CORBA ...
... What levels and "styles" of **parallelism/concurrency** or serialization are ...
No restriction on **concurrency/parallelism** with other components. ...
http://www.esmf.ucar.edu/esmf_presentations/pres_0305_wkshpmichalakes.ppt

Some **Concurrency** & **Parallelism** Operators (CIAO)
... Some **Concurrency** & **Parallelism** Operators (CIAO). Objective: express (independent) And-**parallelism, concurrency** (dependent And-**Parallelism**), and fairness ...
http://clip.dia.fi.upm.es/lpnet/distrib_imperial/node10.html 4k

Abbildung 3.9.: Test-Ergebnis für *parallelism concurrency* vom 20 Mai 2005

4. Zusammenfassung und Ausblick

4.1. Zusammenfassung

Ziel dieser Arbeit war es, ein System zur semantischen Anreicherung von Anfragen an Suchmaschinen zu entwickeln. Basis dafür sollten Topic Maps sein, falls mit ihnen eine ausreichende semantische Ausdruckstärke erreicht werden könnte. Das System sollte aus einer, mit angemessener Fachbegrifflichkeit ausgestatteten, Topic Map, einer Client-Anwendung, sowie der Anwendungsschicht bestehen. Darüber hinaus sollte eine geeignete Eingabe-Sprache angegeben werden. Alle Ziele wurden erreicht.
Nachdem das informatische Problem identifiziert wurde, konnten mögliche Lösungsansätze aufgezeigt werden. Projekte, die sich in diesem Kontext bewegen, wurden näher betrachtet und in Zusammenhang mit der Aufgabenstellung ausgewertet.
Eine Analyse der entsprechenden informatischen Themen, sowie der beteiligten Technologien, schufen die Grundlage für die Entwicklung eines neuen Systems. Dabei wurde im Abschnitt *Information Retrieval* 2.2 mit *Pertinenz* ein Verfahren gefunden, das nicht auf Basis statistischer Daten abläuft. Dieses wurde für die Auswertung innerhalb dieser Arbeit benutzt.
Das Kapitel, das sich mit Suchmaschinen im Allgemeinen und Google und MetaGer als Fallstudien im Speziellen beschäftigt, untersuchte den Bereich der Suchsysteme, der für diese Arbeit eine wichtige Rolle spielt, obwohl er praktisch als *black box* behandelt werden muss.
Mit Topic Maps wurde für diese Arbeit eine Technologie gewählt, die eine schwache semantische Ausdrucksstärke hat. Das Kapitel 2.4 zeigt jedoch, dass Topic Maps Konzepte bieten, die gut genug sind, um einfache Beziehungen auszudrücken. Außerdem wurde untersucht, wie Topic Maps entwickelt werden können.
Nach Darstellung des allgemeinen Szenarios, wurde der zu realisierende Ausschnitt ausgewählt und anschließend das Fundament für das System *TopicSEEK* gelegt. Dazu wurden die benutzen Komponenten in einen objekt-orientierten Ansatz überführt.
Die Topic Map wurde den zur Verfügung stehenden Konzepten entsprechend aufgebaut und mit Begriffen gefüllt, die für die Informatik relevant sind.
Nach Angabe der Grammatik waren schließlich alle Voraussetzungen realisiert, die für den praktischen Einsatz des Systems notwendig waren. Die Testphase lieferte dann auch die erhofften, positiven Resultate, die die Antworten mit Themenbezug durch semantisch sinnvolle Verknüpfungen teilweise auf zehn Treffer von zehn Antworten steigern konnte.

4.2. Fazit

Diese Arbeit hat gezeigt, dass der Ansatz, die Anfrage semantisch anzureichern, erfolgreich umgesetzt werden kann. Die Anzahl relevanter Dokumente als Ergebnis einer Suche kann dadurch erheblich gesteigert werden.
Zur Realisierung des semantischen Modells sind Topic Maps ausreichend. Sie können Begriffe in Beziehung setzen und somit einen Teil der Anfrage, der aus genau einem Begriff besteht, gegen einen komplexeren Anfrage-Teil ersetzen, dessen Begriffe logisch in Beziehung stehen. Die notwendigen logischen Operatoren sind *AND* und *NOT*. Wie die Begriffe aus der Topic Map syntaktisch zusammenhängen, entscheidet die Applikation, die auf die Topic Map zugreift.
Google, als führende Suchmaschine, ist für Anfragen geeignet, auch wenn nicht alle logischen Kombinationen darstellbar sind. Andererseits ist das für den Normalfall auch nicht notwendig. Außerdem eignet sich Google für eine weitergehende Automatisierung, weil es ein API zur Verfügung stellt und aus der Anwendung heraus abgefragt werden kann.
Die Herstellung semantischer Modelle ist kostenaufwendig. Topic Maps bilden da keine Ausnahme. Das Verhältnis der Kosten zum entsprechenden Nutzen muss ausreichen. Für die Verbesserung von Suchanfragen und damit der Förderung der Suche scheint das gegeben, vor allem, wenn man die große Nachfrage bei Suchmaschinen bedenkt.
Unerwartet hat sich während der Erstellung der Topic Map und der Durchführung der Tests herausgestellt, dass das System auch insbesondere geeignet ist, um fachspezifische Abkürzungen zu vervollständigen.

4.3. Kritischer Rückblick

Zum Design der Anwendung

Das Design der Anwendung ist insgesamt gut strukturiert. MVC und der Workflow bilden eine solide und ausgereifte Grundlage des Gesamtsystems. Klassen wurden nach Möglichkeit lose gekoppelt, um deren Wiederverwendung zu unterstützen. Gebräuchliche Entwurfsmuster wurden eingesetzt, soweit sie von Nutzen waren, ohne dabei jedoch auf theoretische Vollkommenheit zu bestehen. Es wurde weiterhin ein vernünftiger Kompromiss zwischen Abstraktion (insbesondere Vererbungstiefe) und momentaner Anforderung an das System geschlossen. So ist z.B. eine abstrakte Oberklasse für Suchmaschinen sinnvoll, falls mehrere Suchmaschinen angebunden werden sollen oder sich die Bedingungen bei der Benutzung von Maschinen ändern. Andererseits hielt die Überlegung eines generischen Ansatzes für Topic Map Engines nicht stand. Hier wird genau eine Engine benötigt, die auch nur lokalen Einflüssen unterliegt[1].

Zur Implementierung der Anwendung

Inwieweit das Anwendungsdesign und die Umsetzung in den Java-Quellcode benutzt werden kann, muss vor einer eventuellen Weiterentwicklung hinterfragt werden. Dazu kann man Quellcode durch Metriken analysieren lassen. Metriken werden entwickelt, um große Systeme bewerten zu können und beruhen auf Einschätzungen. Zudem sind die Kriterien von Metriken auch nicht immer eindeutig. So deutet ein hoher NOC-Wert zwar auf eine bessere Wiederverwendbarkeit hin, andererseits bedeutet er auch wesentlich höheren Testaufwand [Rei02]. Deshalb wird ein optimaler Code wohl unmöglich in allen Bereichen maximal gut abschneiden können.
Metriken sind keine Allzweckwaffen innerhalb der Programmierung. Sie können z.B. nicht prinzipiell entscheiden, ob Programme terminieren (Halteproblem) oder auch wie die Komplexität von Algorithmen einzuschätzen ist. Möglicherweise würde ein Algorithmus besser bewertet als ein anderer, obwohl er eine schlechtere Komplexität aufweist. Dennoch sind Metriken Indikatoren und können mögliche Schwierigkeiten aufzeigen[2].
In diesem Fall wurde der Quellcode des Prototyps mittels *OTW*[3] auf Basis von Chidamber und Kemerers Metrikensuite[4] untersucht. Daraus resultieren die Werte aus Tabelle 4.1, in der Abkürzungen mit folgender Bedeutung erscheinen:

[1] Vorausgesetzt die Lizenzbedingungen ändern sich nicht.
[2] Näheres zu Metriken z.B. bei [SC94], [Rei02] und [Fäh02].
[3] OTW 2.4 Objekttechnologie-Werkbank Build 53.
[4] Metriken nach Sharble/Cohen bzw. Lorenz/Kidd erbrachten ähnliche Resultate. Chidamber/Kemerer erhielten aufgrund ihrer Popularität den Vorzug.

- <WMC> Weighted Methods per Class
- <DIT> Depth of Inheritance Tree
- <NOC> Number of Children
- <CBO> Coupling between Object Classes
- <RFC> Response for a Class

Aufgrund der geringen Größe des Systems ist es eher unwahrscheinlich, dass eine Qualitätsprüfung der Implementierung von TopicSEEK mit Metriken erforderlich ist. Andererseits sind die sehr unterschiedlichen Werte (z.B. für *DIT*) ein Indikator dafür, dass man den Code strukturell noch verbessern kann. Dieser Hinweis sollte zumindest für die Weiterentwicklung hilfreich sein, wenn zu entscheiden ist, auf welcher Basis weitergearbeitet werden soll. Ob die angegebene Werte aus Tabelle 4.1 insgesamt gut genug sind, muss dann je nach Zielsetzung entschieden werden.

Metrik	WMC	DIT	NOC	CBO	RFC
Wartbarkeit	84%	76%	-	-	88%
Wiederverwendbarkeit	84%	24%	7%	-	88%
Erweiterbarkeit	84%	24%	7%	-	-
Testbarkeit	84%	-	93%	-	88%
Zuverlässigkeit	-	-	-	-	-

<-> konnte laut OTW aufgrund fehlender Daten nicht berechnet werden

Tabelle 4.1.: Code-Analyse mit OTW auf Basis von Chidamber und Kemerers Metrikensuite

Der Einsatz von *Swing* innerhalb von Java ist nicht die beste Wahl, da es kein MVC-Design unterstützt. Im Rahmen der prototypischen Realisierung ging es jedoch nicht in erster Linie um die Java-Applikation, sondern vielmehr um den Einsatz von Topic Maps, weshalb eine einfache, schnelle Lösung durchaus ausreichend war. Für eine saubere Umsetzung von MVC findet man z.B. Unterstützung durch neue Technologien, wie *Java Server Faces*[5] oder auch *Jakarta Struts*[6].

Zur Auswahl von Google

Sollte *TopicSEEK* zum Einsatz kommen, so ist zu beachten, dass die Benutzung von Google aus dem Programm heraus begrenzt ist. Der Zugriff über das Google-API ist daher auch mit

[5] URL: `http://java.sun.com/j2ee/javaserverfaces/index.jsp`
[6] URL: `http://struts.apache.org/`

Nachteilen verbunden. Da allerdings die Qualität der Antworten verbessert wird, kommt man möglicherweise mit zehn Antworten aus. Für den Fall, dass man mehr Antworten braucht, muss man gegebenenfalls die verbesserte Anfrage manuell über Google absetzen.

Zur semantischen Ausdrucksstärke von Topic Maps

Eine Frage dieser Arbeit war, ob Topic Maps semantisch ausdrucksstark genug sind. Vielleicht ist die Fragestellung jedoch nicht präzise genug. Denn das, was eigentlich damit gemeint war, ist: Wie ausdrucksstark sind Topic Maps ohne Applikation? Falls Topic Maps allerdings Defizite haben, könnten diese möglicherweise von der Applikation kompensiert werden. Deswegen hätte die Ausgangsfrage auch die Kombination von Topic Maps und Applikationen berücksichtigen müssen.
Mit zusätzlichen Tabellen kann man in einer relationalen Datenbank ebenfalls Beziehungen darstellen, die durch geeignete Applikation ausgewertet werden. In Datenbanken gespeicherte Informationen sind allerdings in höherem Maße von der auswertenden Applikation abhängig. Semantische Modelle in Anwendungen sind mit abnehmendem Anteil von Meta-Informationen zunehmend proprietär. Der Herstellungsaufwand ist dabei die Kehrseite der Medaille.
Es ist also möglicherweise nicht alleine interessant, wie ausdrucksstark einsetzbare Technologien sind, sondern in welchem Verhältnis der semantische Nutzen einer Technologie zum entsprechenden Aufwand steht.

4.4. Ausblick

Folgendes sollte bei Weiterentwicklung des Systems *TopicSeek* Beachtung finden:

- Je nach eingesetzter Suchmaschine und Art des Benutzers (Anfänger, Experte usw.), macht es Sinn, Grammatikmoduln in das System zu integrieren.
- Um ein vollständiges System zu liefern, müssen zusätzlich noch geeignete Tools ermittelt oder erstellt werden, die eine angenehme Pflege der Topic Map gewährleisten.
- Ziele von Weiterentwicklungen müssen nicht ausschließlich Systemverbesserungen sein. Auch ein Angebot hochwertiger Topic Maps zu erstellen, könnte zum Erfolg des gesamten Produkts und einer allgemeinen Akzeptanz beitragen. Der Vorteil liegt auf der Hand. Teurer, aber einmaliger Arbeit bei Erstellung, steht eine vereinfachte und hochwertigere Benutzung gegenüber. Dabei sollten auch nicht zuletzt fachspezifische Abkürzungen berücksichtigt werden.
- Das Erstellen und die Pflege von Topic Maps ist teuer. Eine automatisierte oder teilweise automatisierte Pflege von Topic Maps wäre erstrebenswert. Dazu wäre es möglich, das Suchverhalten der Benutzer zu protokollieren. Das Mitwirken ausgewählter Benutzer, die z.B. mit einem Haken anmerken, wie gut die aktuellen Suchergebnisse sind, könnte allen Benutzern zur Verfügung gestellt werden. Dazu ließen sich Topic Maps auf zentralen Servern realisieren, die dort angepasst und angeboten werden.
  Individuell könnte ein weiteres Modul der Client-Anwendung die Daten der Topic Map für den einzelnen Benutzer anpassen. Dazu sollte man das Projekt SHOE (siehe Kapitel 2.1) näher untersuchen. Dort können Benutzer Ontologien auch erweitern.
- Bei Weiterentwicklung des Systems sollte eine Datenklasse eingeführt werden. Sie sorgt für weitergehende Entkopplung der beteiligten Klassen. Außerdem könnten dort Pfade, die noch zum Teil fest in das System codiert sind, dynamisch behandelt werden.
- Für ein System, das nicht mehr nur Prototyp ist, sollte man *TM4J*[7] bevorzugen. TM4J liefert ein standardisiertes TMAPI aus. Der Vorteil ist, dass sich das System dann näher an offenen Standards positioniert. Dazu müsste gegebenenfalls die Anbindung der TM Engine angepasst werden.
- Google unterstützt maximal 1000 Anfragen je Tag und Konto. Der Google-Account-Lizenz-String sollte daher ausgelagert und je Benutzer spezifiziert werden. Jeder, der das Tool dann benutzt, müsste sich ein eigenes Benutzer-Konto bei Google anlegen. Der String könnte z.B. in einer Datei abgelegt und von der Applikation abgefragt werden[8].

[7] URL: `http://www.tmapi.org/apiDocs/index.html`
[8] Über die Zulässigkeit solcher Maßnahmen, geben die Lizenz-Vereinbarungen von Google Auskunft.

A. Screenshots der Tests

Bei den folgenden Screenshots handelt es sich um die grafische Darstellung der Ergebnisse, die durch die Tests aus Kapitel 3.5 entstanden. Dabei werden zunächst das Resultat der ursprünglichen Anfrage und dann das Resultat der angereicherten Anfrage verwendet. Um welche Anfrage es sich handelt, ist der Tabellenbeschreibung zu entnehmen[1]

[1] Da sowohl die originale Ausgabe von Google, als auch die von TopicSEEK generierte, für dieses Dokument etwas zu groß sind, wurden lediglich die Treffer ohne Kopf- und Fußbereich dargestellt. Auch wurden die Ausgaben teilweise auf der rechten Seite beschnitten, ohne dabei wesentliche Informationen zu unterschlagen.

Home of **Alternation** and guide to the Cambridge Indie/Alternative ... - [Diese Seite üb
Cambridges biggest and best alternative music club. Great site with a resource of many gigs and clubs in and around Cambridge.
www.**alternation**music.com/ - 2k - Im Cache - Ähnliche Seiten

Alternation home page - [Diese Seite übersetzen]
... Prior to publication, each publication in **Alternation** is refereed by at **...** **Alternation** is indexed in The Index to South African Periodicals (ISAP) and **...**
singh.reshma.tripod.com/**alternation**/ - 9k - Im Cache - Ähnliche Seiten

AlterNation
Click the image to go to our site. Please click Here To Enter the MoH: Desert Storm website.
alternation.mohfiles.com/ - 2k - Im Cache - Ähnliche Seiten

Welcome to MoH: Desert Storm! - [Diese Seite übersetzen]
Click to enter. Click here for the website! Nedstat Basic - Free web site statistics Personal homepage website counter Free counter.
alternation.mohfiles.com/mohds/ - 2k - Im Cache - Ähnliche Seiten

Alternation.pl
Kliknij aby wejść.
www.**alternation**.pl/ - 2k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

ALTERNATION 2119 SQUAT A PARIS ARTS ET CULTURE THEATRE DANCE E
... écrire : association@alternation2119.com et nous sommes sur MSN Messenger : alternation2119@hotmail.com. **ALTERNATION** - 19, rue Pierre Bourdan Paris 12 **...**
web2006.free.fr/ - 9k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Alternation of Generations - [Diese Seite übersetzen]
Alternation of Generations. Sexual reproduction involves the two alternating processes of meiosis and fertilization. In meiosis, the chromosome number is **...**
users.rcn.com/jkimball.ma.ultranet/ BiologyPages/A/Alternation.html - 4k - Im Cache - Ähnliche Seite

alternation - [Diese Seite übersetzen]
Definition of **alternation**, possibly with links to more information and implementations.
www.nist.gov/dads/HTML/**alternation**.html - 3k - Im Cache - Ähnliche Seiten

Regex Tutorial - **Alternation** with The Vertical Bar - [Diese Seite übersetzen]
In a regular expression, the vertical bar or pipe symbol tells the regex engine to match any of two or more options.
www.regular-expressions.info/**alternation**.html - 10k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Alternation
... Eine **Alternation** zweier Ausdruckseinheiten kann verschiedenartigen Status im **...** Es kann eine rein phonetische **Alternation** sein, die grammatisch und **...**
www.uni-erfurt.de/sprachwissenschaft/ personal/lehmann/Termini/Alternation.html - 3k - Im Cache - Ä

Abbildung A.1.: Test-Ergebnis für *alternation* vom 20 Mai 2005

(Gupta S.) Determinism, **Nondeterminism**, **Alternation**, and Counting
... Determinism, **Nondeterminism**, **Alternation**, and Counting. Sanjay Gupta (Department of Computer Science, Virginia Polytechnic Institute and State ...
http://www.jucs.org/jucs_7_9/determinism_nondeterminism_alternation_and

SARP Results | **Alternation** and **Nondeterminism**
Publication of Software Research Results produced by NASA's Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP).
http://sarpresults.ivv.nasa.gov/ViewCategory/512.jsp 5k

Alternation and the power of **nondeterminism**
... Subjects: **Alternation** and **nondeterminism**. Additional Classification: F. Theory of Computation F.1 COMPUTATION BY ABSTRACT DEVICES ...
http://portal.acm.org/citation.cfm?id=800061.808764

Welcome to IEEE Xplore 2.0: Bounded **nondeterminism** and **alternation** ...
... Bounded **nondeterminism** and **alternation** in parameterized complexity theory. Chen, Y. Flum, J. Grohe, M. Abt. fur Mathematische Logik, ...
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1214407

Bounded **nondeterminism** and **alternation** in parameterized complexity ...
... FPT + parameter-bounded alternating **nondeterminism**. of bounded **alternation** depth (Theorem 17(3)). Figure 2. Machine descriptions of parameterized ...
http://ieeexplore.ieee.org/iel5/8614/27296/01214407.pdf?arnumber=1214407

Bounded **Nondeterminism** and **Alternation** in Parameterized Complexity ...
... p. 13 Bounded **Nondeterminism** and **Alternation** in Parameterized Complexity Theory ... whose use of **nondeterminism** is bounded in terms of the parameter. ...
http://csdl.computer.org/comp/proceedings/complexity/2003/1879/00/18790013abs.htm

Bounded **Nondeterminism** and **Alternation** in Parameterized Complexity ...
Title: Bounded **Nondeterminism** and **Alternation** in Parameterized Complexity Theory. Authors: Yijia Chen, Jörg Flum, and Martin Grohe ...
http://www.informatik.hu-berlin.de/~grohe/pub/ccc03-abs.html 3k

Bounded **Nondeterminism** and **Alternation** in Parameterized Complexity ...
... FPT + parameter-bounded alternating **nondeterminism**. of bounded **alternation** depth (Theorem 17(3)). C. Our machine model is based on the standard random ...
http://www.informatik.hu-berlin.de/~yijia/papers/uwp.pdf

Reviews.com
... 1-10 of 56 Reviews about "**Alternation** And **Nondeterminism** (F.1.2...)":, Date Reviewed. NFA reduction algorithms by means of regular inequalities ...
http://www.reviews.com/browse/browse_topics4.cfm?ccs_id=1122 116k

Verification = Logic + Algorithmics \Lambda Moshe Y. Vardi Rice ...
... **Nondeterminism** vs. **Alternation** **Nondeterminism**:. ffl (s; a) 7! fs ... **Alternation** vs. **Nondeterminism** Theorem: [Miyano + Hayashi, 1984] Automaton has a ...
http://www.cs.rice.edu/CS/Logic/Games/task5.ps

Abbildung A.2.: Test-Ergebnis für *alternation nondeterminism* vom 20 Mai 2005

NC many-one **reducibility** - [Diese Seite übersetzen]
Definition of NC many-one **reducibility**, possibly with links to more information and implementations.
www.nist.gov/dads/HTML/nCmanyone.html - 3k - Im Cache - Ähnliche Seiten

[PDF] **Reducibility** and unsolvability
Dateiformat: PDF/Adobe Acrobat - HTML-Version
... reducibility. It has traditionally been a very powerful and widely used tool in **...** The formal definition of **reducibility** between sets appears below. **...**
cs.engr.uky.edu/~lewis/texts/ theory/unsolvability/reduce.pdf - Ähnliche Seiten

Computational **Reducibility** -- from MathWorld - [Diese Seite übersetzen]
... "Computational **Reducibility**." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/ComputationalReducibility.html **...**
mathworld.wolfram.com/ComputationalReducibility.html - 19k - Im Cache - Ähnliche Seiten

SAT self-**reducibility** - [Diese Seite übersetzen]
SAT self-**reducibility**. **...** SAT self-**reducibility**. Our definition of problem is really a decision problem namely is tex2html_wrap_inline1269 . **...**
www.cs.jcu.edu.au/ftp/web/teaching/ Subjects/cp3050/1998/Notes/lectures/node3.html - 4k - Im Ca

Logic and Language Links - **reducibility** - [Diese Seite übersetzen]
... reducibility. This concept has currently no gloss. **reducibility** is a: subtopic of recursion theory. **reducibility** has currently no subtopics. **...**
staff.science.uva.nl/~caterina/LoLaLi/Pages/325.html - 8k - Im Cache - Ähnliche Seiten

[PDF] On the **Reducibility** of Sets Inside NP to Sets with Low Information **...**
Dateiformat: PDF/Adobe Acrobat - HTML-Version
... in terms of logspace self-**reducibility** [8]. It is a strong counterpart to **... Reducibility** to sparse sets has been studied for a long time and the **...**
tal.cs.tu-berlin.de/tantau/ publications/OgiharaT2001.pdf - Ähnliche Seiten

Axiom of **Reducibility** - [Diese Seite übersetzen]
Axiom of **reducibility** in philosophy. **...** axiom of **reducibility**. Axiom introduced by English philosopher and mathematician Bertrand Russell (1872-1970) in **...**
www.philosophyprofessor.com/ philosophies/**reducibility**-axiom.php - 41k - Im Cache - Ähnliche Se

ENVIRONMENTS, POLICIES, AND **REDUCIBILITY** - [Diese Seite übersetzen]
ENVIRONMENTS, POLICIES, AND **REDUCIBILITY**. **...** ENVIRONMENTS, POLICIES, AND **REDUCIBILITY**. In this section, we will introduce our formalism. **...**
www-2.cs.cmu.edu/afs/cs/project/ jair/pub/volume6/agre97a-html/node5.html - 9k - Im Cache - Ähnli

Reducibility, randomness, and intractibility (Abstract) - [Diese Seite übersetzen]
... 4 Richard E. Ladner, On the Structure of Polynomial Time **Reducibility**, **...** On &ggr;-**reducibility** versus polynomial time many-one **reducibility**(Extended **...**
portal.acm.org/citation.cfm?id=803405 - Ähnliche Seiten

Reducibility game download - [Diese Seite übersetzen]
... Freeware games; Flash games; ZX Spectrum games. **reducibility** game pages 1 **...** //**reducibility** game. by Apus Software ..level X your appropriate element **...**
www.games4win.com/game/**reducibility**/ - 9k - Im Cache - Ähnliche Seiten

Abbildung A.3.: Test-Ergebnis für *reducibility* vom 20 Mai 2005

NC-**Reducibility** and P-**Completeness**
NC-**Reducibility** and P-**Completeness**. ... Obviously, the NC-**reducibility** is an extension of the NC tex2html_wrap_inline3418 -**reducibility**. ...
http://www.i.kyushu-u.ac.jp/~shoudai/P-complete/all/node2.html 7k

Reducibility and **Completeness** in Private Computations
... We define the notions of **reducibility** and **completeness** in (two-party and multiparty) ... Key words. private computation, **reducibility**, **completeness**, ...
http://epubs.siam.org/sam-bin/dbq/article/32174

Completeness in approximation classes
... In this section we define a natural approximation preserving **reducibility** and introduce the notion of **completeness** both in NPO and in APX. Definition 8 ...
http://www.nada.kth.se/~viggo/wwwcompendium/node5.html 12k

Course Descriptions--66 Computer Science--Rensselaer Catalog 97|98
... computational complexity, **reducibility**, **completeness**, Cook's theorem. ... parallel vs. sequential complexity, **reducibility**, **completeness**, ...
http://www.rpi.edu/dept/catalog/97-98/Courses/66.html 32k

Computational Complexity: Favorite Theorems: NP-**Completeness**
... Tautology and Subgraph Isomorphism are hard for NP under P-**reducibility**. ... The issue with NP-**completeness** is a relativistic notion of simultaneity. ...
http://weblog.fortnow.com/2005/04/favorite-theorems-np-completeness.html 36k

Contents of the first edition
... Polynomial time **reducibility** - Definition of NP-**completeness** - The Cook-Levin Theorem; EXAMPLES OF NP-COMPLETE PROBLEMS ...
http://www-math.mit.edu/~sipser/itoc-1.html 7k

Reviews.com
... 1-10 of 106 Reviews about "**Reducibility** And **Completeness** (F.1.3...)", Date Reviewed. Reductions between Disjoint NP-Pairs Glaser C., Selman A., ...
http://www.reviews.com/browse/browse_topics4.cfm?ccs_id=1131 63k

Reductions and (Non-)Approximability, by L. Trevisan
... and we prove several **completeness** results using the AP-**reducibility**. By making use of the PTAS-**reducibility** (a generalization of the AP-**reducibility**) we ...
http://eccc.uni-trier.de/eccc-local/ECCC-Theses/trevisan.html 6k

SARP Results | **Reducibility** and **Completeness**
Publication of Software Research Results produced by NASA's Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP).
http://sarpresults.ivv.nasa.gov/ViewCategory/519.jsp 5k

CS 4341 A97 - Syllabus
... Tu, Nov 18, Quiz III, **Reducibility** via Computation Histories, 5.1 ... Tu, Dec 09, Quiz IV, NP-**completeness** - **Reducibility**, 7.4 ...
http://www.cs.wpi.edu/~ruiz/Courses/cs4123_B97/schedule_cs4123_b97.html 4k

Abbildung A.4.: Test-Ergebnis für *reducibility completeness* vom 20 Mai 2005

Eidgenössisches Departement des Innern - **EDI**
Bietet Informationen zu Publikationen und aktuellen Themen.
www.**edi**.admin.ch/ - 15k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Edi - ECIN - Electronic Commerce Info Net
EDI/Electronic Data Interchange ist nicht tot: Web-**EDI** und Internet-**EDI** öffnen neue Wege.
www.ecin.de/**edi**/ - 57k - Im Cache - Ähnliche Seiten

What is **EDI**? - A Word Definition From the Webopedia Computer ... - [Diese Seite übe
This page describes the term **EDI** and lists other pages on the Web where you can find additional information.
www.webop**edi**a.com/TERM/E/EDI.html - 46k - Im Cache - Ähnliche Seiten

Welcome to XML/**EDI** Group's Home Page - [Diese Seite übersetzen]
XML/**EDI** Group promotes **EDI** as an XML application; combining structured presentation with structured data allows for document-centric tools to be used ...
www.geocities.com/WallStreet/Floor/5815/ - 24k - Im Cache - Ähnliche Seiten

BAA - Edinburgh Airport official site - [Diese Seite übersetzen]
Official Edinburgh Airport website - live flight arrivals, timetable, travel information, services and company information.
www.baa.com/main/airports/**edi**nburgh/ - 40k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Nielsen **EDI** - The Worldwide Box Office Authority - [Diese Seite übersetzen]
... Nielsen **EDI** proudly introduces FLASH (Film Location And Sales Heartbeat), a Web based system that allows our clients to instantly access real-time film ...
www.entdata.com/ - 11k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

EDI Organisation KEG
EDI Organisation KEG - Ihr Spezialist in Internet und Softwarefragen.
www.**edi**.org/ - 5k - Im Cache - Ähnliche Seiten

Washington Publishing Company - **EDI** - HIPAA - XML - [Diese Seite übersetzen]
,,Washington Publishing Company **EDI** HIPAA and XML.
www.wpc-**edi**.com/ - 27k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Dienststellen Polizei Baden-Württemberg - Frameset
www.polizei-bw.de/**edi**/ - 2k - Im Cache - Ähnliche Seiten

GLOSSAR.de: EDIFACT, **EDI**, Electronic Data Interchange For ...
EDIFACT, **EDI**-FACT, **EDI**, Electronic Data Interchange For Administration Commerce and Transport, Edifact, Edifakt.
www.glossar.de/glossar/z_**edi**.htm - 14k - Im Cache - Ähnliche Seiten

Abbildung A.5.: Test-Ergebnis für *edi* vom 20 Mai 2005

What is **EDI**? - A Word Definition From the Webopedia Computer ...
This page describes the term **EDI** and lists other pages on the Web where you can find additional information.
http://www.webopedia.com/TERM/E/EDI.html 46k

Workgroup for **Electronic Data Interchange** Home Page
Workgroup for **Electronic Data Interchange**. Improving healthcare through **Electronic** Commerce.
http://www.wedi.org/ 59k

Medicare **Electronic Data Interchange**
This is the Medicare **EDI** website. ... If so, go to **EDI** List. Last Modified on Monday, May 16, 2005. Department of Health and Human Services Logo ...
http://cms.hhs.gov/providers/edi/default.asp 19k

6-6.com - paper - Practical **EDI** - Toronto Based E-Commerce/**EDI** ...
offers tips on **EDI** deployment, lists various **EDI** related costs and discusses role of Internet and XML in the future of **EDI**. The paper helps decision makers ...
http://www.6-6.com/66/paper/edi.html 9k

Electronic Data Interchange (EDI) Information at Business.com
Providers of **EDI** information resources, products and services. ... Provider of B2B, **EDI** (**electronic data interchange**) and ecommerce solutions and services. ...
http://www.business.com/directory/internet_and_online/ecommerce/electronic_data_interchange_

EDI Translator / **EDI** Software for Mapping HIPAA, X12 & EDIFACT **EDI**
Complete **EDI** translation software supporting all X12 and EDIFACT versions and transaction sets for Windows, Unix and Linux. Supports HIPAA **EDI** standards.
http://www.proedi.com/ 22k

EDI - a Whatis.com definition - see also: **Electronic Data Interchange**
News, trends and advice for CIOs and IT executives.
http://searchcio.techtarget.com/sDefinition/0,,sid19_gci213925,00.html 30k

DISA - About
... the **Data Interchange** Standards Association helps individuals and the business ... HEDNA promotes **electronic** distribution and the use of technology to ...
http://www.disa.org/ 34k

http://www.edi.wales.org/

Electronic Data Interchange-Internet Integration (ediint) Charter
... **Electronic Data Interchange** (**EDI**) is a set of protocols for conducting ... method for packaging the **EDI** X12 and UN/EDIFACT transactions sets in a ...
http://www.ietf.org/html.charters/ediint-charter.html 9k

Abbildung A.6.: Test-Ergebnis für *edi electronic data interchange* vom 20 Mai 2005

Digital Direkt for Linux
DDL Logo. **DDL** - Digital Direct for Linux Multiprotokoll-Controler und Steuerungssoftware für digitale Modelleisenbahnen ...
www.vogt-it.com/OpenSource/DDL/ - 6k - Im Cache - Ähnliche Seiten

DDL Deutsche Dermatologische Lasergesellschaft Hautarzt ...
Hautärztinnen und Hautärzte mit Spezialisierung auf Laseranwendungen bemühen sich um die modernste und erfolgreichste Lasertherapie der Haut und informieren ...
www.**ddl**.de/ - 10k - Im Cache - Ähnliche Seiten

PhazeDDL.com - Full Games, Movies, Apps, Warez Downloads for Free - [Diese Seit
Warez Free Appz Gamez Mp3z Hacking Serialz Crackz Ftpz Romz.
www.phaze**ddl**.com/ - 16k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

MPEG-7 **DDL** Home Page - [Diese Seite übersetzen]
... MPEG-7 **DDL** Working Draft 4.0. w3575.doc (Beijing, July 2000) ... **DDL** FAQ. Or send **DDL** Queries to: mpeg7-**ddl**@darmstadt.gmd.de ...
archive.dstc.edu.au/mpeg7-**ddl**/ - 7k - Im Cache - Ähnliche Seiten

AntoSoft **DDL** - FULL Downloads - [Diese Seite übersetzen]
... Friends/Partners PhazeDDL.com FULL XXX MOVIES Katz.ws DIRECT DOWNLOADS FULL GAMES XXX DivX Movies SatanWarez.com Beasty's Portal Illegal WAREZ ...
www.anto**ddl**.com/ - 47k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Das Dosierte Leben - Jahreszeitenschrift für Sinn und Unsinn | Home
... Die aktuelle Ausgabe | Das Dosierte Leben No. 34 - Die POLYESTERFRAGMENTE des Gunter Wessalowski Die aktuelle Ausgabe: Das Dosierte Leben No. 34 - ...
www.das-dosierte-leben.de/ - 36k - Im Cache - Ähnliche Seiten

Den danske Landinspektørforening - Søgeside
Interesseorganisation for landinspektører i Danmark. Er paraplyorganisation for Praktiserende Landinspektørers Forening (PLF), ...
www.**ddl**.org/ - 12k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

DDL OMNI Engineering - [Diese Seite übersetzen]
DDL OMNI Engineering Corp. performs a wide spectrum of Engineering and Technical Services devoted to the acquisition, design, testing, logistics, ...
www.**ddl**omni.com/ - 15k - Im Cache - Ähnliche Seiten

DDL Package Testing Services - Shelf Life, Product & Material Testing - [Diese Seite
DDL Testing Services provides shelf-life, package, product & material testing services and validation including shock, vibration, tensile, leak, ...
www.testedandproven.com/ - 19k - Im Cache - Ähnliche Seiten

DDL Main Page - [Diese Seite übersetzen]
DDL is the drug design laboratory of School of Pharmacy at Milan University.
users.unimi.it/~**ddl**/ - 2k - Im Cache - Ähnliche Seiten

Abbildung A.7.: Test-Ergebnis für *ddl* vom 20 Mai 2005

What is a **DDL**?
... The **DDL** is a dictionary of definitions which describes a **language** for specifying
... **DDL** provides the framework on which this dictionary is organized by ...
http://www.iucr.org/iucr-top/cif/mmcif/ndb/ddl/ddl/node3.html 4k

Macromolecular **DDL**
... **DDL** Definitions **Describing** Dictionaries and **Data** Blocks. DATABLOCK;
DATABLOCK_METHODS; DICTIONARY; DICTIONARY_HISTORY. References; Abou
http://www.iucr.org/iucr-top/cif/mmcif/ndb/ddl/ddl/ddl.html 7k

Spark's Pensieve - **DDL**
... your **ddl** script (which defines the **data** format) of the actual **data** file.
Then you tell it □give me the value of left-aileron-angle□ and the **DDL** engine ...
http://pensieve.thinkingms.com/CategoryView,category,DDL.aspx 66k

System.**DDL** Homepage
... contains the **language** specifications and introduces you to programming the **DDL**
... **DDL** engine in your own applications and the API that the **DDL** exposes. ...
http://pensieve.thinkingms.com/sparksite/work/System.DDL/ 11k

ddl : Definition from the Online Dictionary at Datasegment.com
Definition of **ddl** from several databases powering the datasegment.com online ...
2003) : **DDL Data** Definition **Language DDL** Document **Description Language** ...
http://onlinedictionary.datasegment.com/word/ddl 4k

define database(**ddl**) define database(**ddl**) define database □create ...
... define index(**ddl**). DIAGNOSTICS. See Chapter 3 for a discussion of errors and
... define relation(**ddl**). The following example defines a field and then ...
http://www.ibphoenix.com/downloads/ddl_syntax.pdf

Definition: **data description language**
data description language (**DDL**). **data description language** (**DDL**): Synonym **data**
definition **language**. These definitions were prepared by ATIS Committee T1A1. ...
http://www.atis.org/tg2k/_data_description_language.html 2k

ddl2java: Java **Data Description** Tool
... Due to the semantics of **DDL**, ddl2java must handle occurring fields and ...
If a **DDL** field is an occurring field, the corresponding instance variable ...
http://nonstop.compaq.com/nsswdocs/nsj/nsj_1_6/toolsref/ddl2java.htm 16k

* **DDL** - (GIS): Definition
DDL Online Encyclopedia. ... **DDL Data** definition **language**. SQL statements that
can be used either interactively or within programming **language** source code ...
http://en.mimi.hu/gis/ddl.html 10k

Programming **Languages** (**Data Description Languages**, Basic Assembly ...
... This is also known as **Data Description Languages**, Basic Assembly **Language**,
DDL, Computer **Languages**, Development **Languages**, Computer Programming ...
http://www.bitpipe.com/tlist/Programming-Languages.html 36k

Abbildung A.8.: Test-Ergebnis für *ddl data description language* vom 20 Mai 2005

Start D
... Willkommen auf der Website des Internationalen **Controller** Vereins in Deutschland! Der Internationale **Controller** Verein eV hat über 4.000 Mitglieder ...
www.**controller**verein.de/ - 21k - Im Cache - Ähnliche Seiten

CM - **Controller** Magazin online
Homepage des **Controller** Magazin. ... Arbeitsergebnisse aus der **Controller** Praxis * Controlling-Anwendungen im Management * ISSN 1616-0495 - im 30. Jahrgang ...
www.**controller**magazin.de/ - 5k - Im Cache - Ähnliche Seiten

Controller Akademie
CA **Controller** Akademie - privates Institut fuer Unternehmensplanung und Rechnungswesen AG.
www.**controller**akademie.de/ - 7k - Im Cache - Ähnliche Seiten

Controlling
... Controlling, **Controller**, Unternehmensplanung, Rechnungswesen und Finanzen ... zu finden:**Controller**-Kollegen-Verzeichnis (C's Directory) (ca. ...
www.my-controlling.de/ - 31k - Im Cache - Ähnliche Seiten

Aircraft For Sale at **Controller**.com: Airplanes For Sale, Aircraft ... - [Diese Seite übe
Piper Cubs at **Controller**.com. Your source for airplanes for sale, used aircraft, used airplanes.
www.**controller**.com/ - 62k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Österreichisches **Controller**-Institut
Österreichisches **Controller**-Institut - Wissen für **Controller** und Manager - Controlling, Accounting, Finance.
www.oeci.at/ - 2k - Im Cache - Ähnliche Seiten

California State **Controller's** Office - [Diese Seite übersetzen]
Controller's profile, introduction to office, and resources to assist the people of California.
www.sco.ca.gov/ - 33k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

EuroDicAutom - Search - [Diese Seite übersetzen]
European Union terminology. Available on most European Union languages.
europa.eu.int/eurodicautom/login.jsp - 19k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

BVBC: Intro
skip intro.
www.bvbc.de/ - 4k - Im Cache - Ähnliche Seiten

Controller Magazine - [Diese Seite übersetzen]
Includes tables of contents and indices to back issues, copies of special reports, and links to resources for the financial manager.
www.**controller**mag.com/ - 6k - 18. Mai 2005 - Im Cache - Ähnliche Seiten

Abbildung A.9.: Test-Ergebnis für *controller* vom 20 Mai 2005

(ootips) Model-View-**Controller**
The Model-View-**Controller** (**MVC**) is a commonly used and powerful architecture for GUIs. How does it work?
http://ootips.org/mvc-pattern.html 8k

Designing Enterprise Applications with the J2EE Platform, Second ...
... Model-View-**Controller** ("**MVC**") is the BluePrints recommended architectural ...
A Web-tier **MVC** **controller** maps incoming requests to operations on the ...
http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5

Java BluePrints - J2EE Patterns
... By applying the Model-View-**Controller** (**MVC**) architecture to a JavaTM 2 Platform, Enterprise Edition (J2EETM) application, you separate core business ...
http://java.sun.com/blueprints/patterns/MVC-detailed.html 9k

Model View **Controller**
... original **MVC** had four objects: not only the model, view, and **controller**, ...
Action as **controller** and models as JavaBean models to have a **MVC** pattern. ...
http://c2.com/cgi/wiki?ModelViewController 20k

trinket : model view **controller** pattern
... the model-view-**controller** (**MVC**) design pattern. here advantages example app models controllers views todos refs. **MVC** introduction ...
http://www.cs.indiana.edu/~cbaray/projects/mvc.html 9k

http://en.wikipedia.org/wiki/Model_view_controller

php.MVC The Model View **Controller** (MVC) Framework for PHP Web ...
A framework for PHP web applications that implements the **MVC** design pattern. Also include SleeK Action Wizard tool.[Open source, LGPL]
http://www.phpmvc.net/ 14k

Model-View-**Controller** Pattern
... Model-View-**Controller** (**MVC**) is a classic design pattern often used by applications that need the ability to maintain multiple views of the same data. ...
http://www.enode.com/x/markup/tutorial/mvc.html 8k

Application Architecture: The Model-View-**Controller** Design Pattern
... View Objects Present Information to the User **Controller** Objects Tie the Model to the View Why Is **MVC** Important? Model Objects Represent Data and Basic ...
http://developer.apple.com/documentation/Cocoa/Conceptual/AppArchitecture/Concepts/MVC.

:: phpPatterns() - Model View **Controller** Pattern
... 2) In the **MVC** design pattern, one view always has one **controller**, ... 3) The purpose of a **Controller** in **MVC** in a desktop application is to react to user ...
http://www.phppatterns.com/index.php/article/articleview/11/ 37k

Abbildung A.10.: Test-Ergebnis für *controller mvc -channel* vom 20 Mai 2005

B. Inhalt der CD

Auf der CD finden sich folgende Daten:

- Eine installierbare Datei, um den Prototypen zu testen,
- der Quellcode der Anwendung,
- die Javadoc-Dokumentation,
- die benutzte Topic Map,
- dieses Dokument als PDF-Datei,
- das API von k42,
- das API von Google,
- die UML-Dateien aus OTW und MS Visio,
- der Bibtex-Eintrag für dieses Dokument.

C. Danksagung

Als erstes möchte ich herzlich meiner Oma und meinen Eltern für die liebe Unterstützung danken, die mir sehr geholfen hat.
Meinen Freunden Hayat und Roland danke ich ebenfalls für ihre freundliche Hilfe und die Tatsache, dass ich mich auf sie verlassen konnte und weiterhin verlassen kann.
Mein Freund und Leidensgenosse Mirko hat es stets verstanden, mich im richtigen Moment bei Laune zu halten und soll hier ebenfalls einen Ehrenplatz erhalten. :)
Lars, Olli, Georg, Thorsten und Irene teilten das Schicksal der letzten Prüfungsleistung mit mir und dürfen hier auch auf keinen Fall vergessen werden. Es hat viel Spaß gemacht.
Wie ein Glücksfall ist Carsten, der selber an seiner Diplomarbeit sitzt, im richtigen Moment aufgetaucht, um seine Erfahrung im Java-Deployment-Umfeld mit mir zu teilen. Danke schön.
Ein ganz besonderes Lob möchte ich meinem Erstbetreuer Kai von Luck aussprechen, der es auf hochwertig eloquente Weise geschafft hat, mir beizubringen, wie man eine wissenschaftliche Arbeit anfertigt. Dabei vermochte er es, mir trotz permanent akuter Zeitnot, auch noch zwischen Tür und Angel eine entscheidende Hilfestellung zu geben. Ohne ihn läge diese Arbeit qualitativ auf einem anderen Niveau und würde nicht einmal die Kriterien einer Diplomarbeit erfüllen.
Danken möchte ich auch Professor Klauck, der es mühevoll aber leidenschaftlich erreicht hat, mich wieder auf den richtigen Pfad zu führen. Als ich drohte, etwas kaum zu Formalisierendes zu formalisieren und den Bezug zum eigentlichen Thema zu verlieren, hat er mein Navigationssystem neu geeicht.
Professor Böhm, der für mich fachlich während des ganzen Studiums ein Vorbild war, hat mir, mit CiteSeer und Google Scholar, zwei exzellente Werkzeuge empfohlen, ohne die ich kaum ausgekommen wäre. Darüber hinaus habe ich ihm wichtige konzeptionelle Fertigkeiten zu verdanken und möchte ihm dafür das gebührende Lob aussprechen.
Die Zweitbegutachtung meiner Arbeit hat Professor Klemke vorgenommen. Obwohl er zur Zeit der Erstellung dieses Papiers bereits extrem viele Studenten betreute, hat er sich ohne zu zögern dazu bereit erklärt. Mit dem Hinweis, das Thema interessiere ihn ohnehin, hat er mir zu guter Letzt auch noch mein schlechtes Gewissen genommen, ihm diese Arbeit aufzubürden. Vielen Dank dafür.
Last but not least, gebührt mein Dank Birgit Wendholt. Sie hatte das Pech, ihr Praktikum direkt neben meinem Raum zu betreuen. Deshalb war es ihr auch nicht möglich, zu entfliehen, wenn ich u.a. Schwierigkeiten mit Latex hatte. ;-)

Literaturverzeichnis

[BVL03] Sean Bechhofer, Raphael Volz, and Phillip Lord. Cooking the Semantic Web with the OWL API, 2003. University of Manchester, UK, Institute AIFB, University of Karlsruhe, Germany, http://www.cs.man.ac.uk/~phillord/download/publications/cooking03.pdf.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, 1999. ISBN 0-201-39829-X.

[Cen05] IBM Almaden Research Center. WebFountain, March 2005. http://www.almaden.ibm.com/webfountain/.

[CS03] Volker Claus and Andreas Schwill. *Duden Informatik, 3. Auflage*. Dudenverlag, Mannheim, 2003. ISBN 3-411-10023-0.

[Dac03] Michael C. Daconta. *The Semantic Web*. Wiley Publishing, Indiana, 2003. ISBN 0-471-43257-1.

[ea03] Jack Park et al. *XML Topic Maps, Creating and Using Topic Maps for the Web*. Addison-Wesley, Boston, 2003. ISBN 0-201-74960-2.

[FB96] Robert Floyd and Richard Beigel. *Die Sprache der Maschinen, 1. Auflage*. International Thomson Publishing, Bonn, 1996. ISBN 3-8266-0216-1.

[Fen02] Dieter Fensel. *Language Standardization for the Semantic Web: The Long Way from OIL to OWL*. Springer-Verlag, London, UK, 2002. DCW '02: Revised Papers from the 4th International Workshop on Distributed Communities on the Web, ISBN 3-540-00301-0.

[Fer03] Reginald Ferber. *Information Retrieval - Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Dpunkt Verlag, Heidelberg, web version edition, October 2003. ISBN 3-89864-213-5, http://information-retrieval.de/irb/ir.html.

[Fäh02] Prof. Dr. Klaus-Peter Fähnrich. *Software-Qualitäts-Management*. Kernfachvorlesung: Praktische informatik, Universität Leipzig, Leipzig, 2002. http://ais.informatik.uni-leipzig.de/download/2002s_v_sqm/2002s_sqm_v_08.pdf.

[fIF05] Gesellschaft fuer Informatik Fachgruppe 2.5.4. Information Retrieval, March 2005. http://www.uni-hildesheim.de/~fgir/.

[Fuh99] N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, 17(3):229–249, 1999.

[Fuh04] Norbert Fuhr. Information Retrieval. *Skriptum zur Vorlesung 2004*, June 2004. http://www.is.informatik.uni-duisburg.de/courses/ir_ss04/folien/irskall.pdf.

[Gar01] Lars Marius Garshol. tolog: A topic map query language, XML Europe 2001, Berlin, 2001. http://www.ontopia.net/topicmaps/materials/tolog.html.

[Gar05] Lars Marius Garshol. Topic maps, RDF, DAML, OIL, A comparison, 2005. http://www.ontopia.net/topicmaps/materials/tmrdfoildaml.html.

[GHJV96] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Entwurfsmuster, Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley-Longman, Bonn, 1996. ISBN 3-89319-950-0.

[Glo03] Michael Gloeggler. *Suchmaschinen im Internet*. xpert.press. Springer, Heidelberg, 2003. ISBN 3-540-00212-x.

[GMV05] Nicola Guarino, Claudio Masolo, and Guido Vetere. OntoSeek: Content-BasedAccess to the Web, April 2005. http://scholar.google.com/scholar?hl=en&lr=&q=cache:2YXjYIrNyF8J:www.ladseb.pd.cnr.it/infor/ontology/Papers/OntoSeek.pdf+.

[Hec05] Ronald Heckel, 2005. http://www.topicmap-design.com/.

[Hef05] Jeff Heflin. SHOE, Parallel Understanding Systems Group, Department of Computer Science, University of Maryland at College Park, April 2005. http://www.kbs.uni-hannover.de/ki/ki2kb/aij-shoe.ps.

[Her04] Roland Herrmann. Natürlichsprachliches interface für wissensbasierte systeme. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, Hamburg, 2004.

[HHLZ05] Jeff Heflin, James Hendler, Sean Luke, and Qin Zhendong. SHOE: A Knowledge Representation Language for Internet Applications, April 2005. http://www.cs.umd.edu/projects/plus/SHOE/.

[HM03] Mr. G. Ken Holman and Dr. James David Mason. Information technology – SGML applications – Topic maps, March 2003. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38068&ICS1=35&ICS2=240&ICS3=30.

[HU94] John E. Hopcroft and Jeffrey D. Ullman. *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 3. Auflage*. Addison-Wesley, Bonn, 1994. ISBN 3-89319-744-3.

[KT00] Mei Kobayashi and Koichi Takeda. Information retrieval on the web. *ACM Computing Surveys*, 32(2), 2000.

[Kur04] Dominik Kuropka. *Modelle zur Repräsentation natürlichsprachlicher Dokumente. Ontologie-basiertes Information-Filtering und -Retrieval mit relationalen Datenbanken*. Logos, Berlin, 2004. ISBN 3-8325-0514-8.

[MFHS02] Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. Daml+oil: An ontology language for the semantic web. *IEEE Intelligent Systems*, 17(5):72–80, 2002. IEEE Educational Activities Department, Piscataway, NJ, USA, http://dx.doi.org/10.1109/MIS.2002.1039835.

[MW03] Marcel Machill and Carsten Welp. *Wegweiser im Netz, Qualitaet und Nutzung von Suchmaschinen*. Verlag Bertelsmann Stiftung, Guetersloh, 2003. ISBN 3-89204-714-6.

[nA05] novomind AG. IQ interactive, March 2005. http://novomind.de/.

[Nag03] Karin Nagel. Ontologien in der stichwortbasierten suche. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg, Hamburg, 2003.

[Noa01] Wilhelm Noack. Information retrieval - suchmodelle und data-mining-verfahren für textsammlungen und das web. *Hannover*, 2001. Hannover, Uni RRZN Regionales Rechenzentrum fuer Niedersachsen, RRZN-Klassifizierungsschluessel Net.All 10, Paper Version, http://www.rrzn.uni-hannover.de.

[Oes98] Bernd Oestereich. *Objektorientierte Softwareentwicklung, Analyse und Design mit der Unified Modeling Language*. Oldenbourg Verlag, Wien, 1998. ISBN 3-486-24787-5.

[Rei02] Ralf Reißing. *Bewertung der Qualität objektorientierter Entwürfe*. Dissertation, Institut für Informatik der Universität Stuttgart, Stuttgart, 2002. www.worte-projekt.de/reissing/Dissertation_Reissing.pdf.

[Res05] IBM Research. The Unstructured Information Management Architecture Project, March 2005. http://www.research.ibm.com/UIMA/.

[Rie97] Dirk Riehle. *Entwurfsmuster für Softwarewerkzeuge*. Addison Wesley Longman, Bonn, 1997. ISBN 3-8273-1147-0.

[SC94] Chidamber S.R. and Kemerer C.F. A Metrics Suite for Object-Oriented Design, June 1994. IEEE Transactionson Software-Engineering.

[Sch99] Uwe Schoening. *Theoretische Informatik - kurzgefaßt, 3. Auflage*. Spektrum Akademischer Verlag, Ulm, 1999. ISBN 3-8274-0250-6.

[Sch05] Google Scholar. Preferences, March 2005. `http://scholar.google.com/scholar_preferences?prev=/`.

[Sow00] John F. Sowa. *Knowledge Representation, Logical, Philosophical, and Computational Foundations*. Brooks/Cole, Thomson Learning, 2000. ISBN 0-534-94965-7.

[SSH95] Peter Sander, Wolffried Stucky, and Rudolf Herschel. *Grundkurs Angewandte Informatik IV, Automaten Sprachen Berechenbarkeit, 2. Auflage*. B.G. Teubner Stuttgart, Stuttgart, 1995. ISBN 3-519-12937-X.

[Szd01] Andre Szdzuy. Aspekte der Granularität von Komponenten, 2001. `http://swt.cs.tu-berlin.de/lehre/seminar/ws00/folien/Granu.pdf`.

[Tha01] Thomas Thaler. metager - eine suchmaschine als forschungsprojekt, 2001. `http://matrix.orf.at/bkissue/011028_1.htm`.

[Uni05a] Hannover Uni. RRZN Regionales Rechenzentrum fuer Niedersachsen, 2005. `http://www.rrzn.uni-hannover.de`.

[Uni05b] RRZN Hannover Uni. MetaGer, die MetaSuchmaschine, 2005. `http://www.metager.de/`.

[vR05] Keith van Rijsbergen. The Information Retrieval Group, March 2005. `http://ir.dcs.gla.ac.uk/`.

[VSU99] Alfred V.Aho, Ravi Sethi, and Jeffrey D. Ullmann. *Compilerbau Teil 1, 2. Auflage*. Oldenbourg Verlag München Wien, Wien, 1999. ISBN 3-486-25294-1.

[Wei03] Gerd Weiß. Ausarbeitung zum Oberseminar: Zu digitalen Bibliotheken, Indexierung und Volltextsuche, Visualisierung und Ranking, Julius-Maximilians-Universität Würzburg, 2003. `http://www2.informatik.uni-wuerzburg.de/mitarbeiter/ebner/teaching/seminar/seminarSS2003/weiss.pdf`.

[WM02] Richard Widhalm and Thomas Mueck. *Topic Maps, 1. Auflage*. xpert.press. Springer, Berlin, 2002. ISBN 3-540-41719-2.

[Woo05] Murray Woodman. Hand-crafted Machine-generated Knowledge Interchange, March 2005. `http://www.topicmap.com/topicmap/tools.html`.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 3. Juni 2005	
Ort, Datum	Unterschrift