Justus Liebig Universität Gießen
Fachbereich Germanistik im FB 05
Professor Dr. Gerd Fritz
Gießen, im Mai 2000

Studienarbeit zur Magisterprüfung

„Thematische Struktur in Hypertexten"

Methoden zur Visualisierung

Kai Wörner
Studiengang: MA Germanistik / Politikwissenschaft
Margaretenstraße 45
20357 Hamburg
Telefon: 040 9999 4995
E-Mail: woerner@coremedia.de

Hiermit versichere ich, dass ich diese Magisterarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Angabe der Quellen kenntlich gemacht.

Hamburg, im Mai 2000

Inhalt

1 Einleitung

„Pick up your pen, mouse or favorite pointing device and press it on a reference in this document – perhaps to the author's name, organisation or related work. Suppose you are directly presented with the background material – other papers, the authors coordinates, the organisation's address and its entire telephone directory. Suppose each of these documents has the same property of being linked to other original documents all over the world. You would have at your fingertips all you need to know about electronic publishing, high energy physics or for that matter asian culture. As you are reading this book on paper, you can only dream, but read on."[1]

Die Vision, die Tim Berners-Lee, der „Erfinder" des WWW, 1992 hatte, ist zumindestens für die Seite des puren Content, für die Informationen, die zur Verfügung stehen, inzwischen wahr geworden. Die ungeheure Popularität des Internet im allgemeinen und des World Wide Web (WWW) im speziellen hat zu einer wahren Explosion der in diesem Medium verfügbaren Information geführt. Trotzdem ist das WWW vom beschriebenen Szenario (Alle erdenklichen Informationen in „Klickweite") noch weit entfernt, da sich Informationen (z.B. alle Informationen über „high energy physics") oftmals nur unter extremen Anstrengungen überhaupt finden lassen.

Schon für die Vorstellung von Berners-Lee stellt sich bei näherer Betrachtung die Frage, über wie viele Verknüpfungen man wohl von seinem Dokument über die Möglichkeiten des WWW auf die Informationen über asiatische Kultur gelangen soll – und es ist nicht davon auszugehen, dass er sich 1992 vorstellen konnte, welche Ausmaße das WWW nur wenige Jahre später haben sollte. Diese Route durch das WWW nur über Hyperlinks in Dokumenten zu finden wäre heute wohl ein ziemlicher Glücksfall, in jedem Fall aber mit unzumutbarem Aufwand verbunden.[2]

---

[1] Tim Berners-Lee u.a. : „Introduction of ‚World-Wide Web: The Information Universe'" 1992 (reprint, S. 2)

[2] Eine Erhebung zu diesem Thema kommt zu dem Ergebnis, dass man von einer beliebigen Seite im WWW zu einer beliebigen anderen über durchschnittlich 19 Hyperlinks springen muss: *http://www.nd.edu/~networks/papers.html#paper1*

Suchmaschinen wie Altavista[3] oder Google[4] und Recherchetools wie alletra[5] versuchen, das Auffinden einzelner Dokumente zu bestimmten Suchbegriffen im WWW zu erleichtern. Diese Suchmaschinen leiden aber darunter, oftmals eine unüberschaubare Menge von Dokumenten zu liefern, deren Relevanz man nur sehr schwer abschätzen kann. Außerdem ist den einzelnen Suchergebnissen nicht anzusehen, wie ihr semantischer Zusammenhang untereinander ist – sofern vorhanden. Die WWW-Dokumente, die von diesen Suchmaschinen indiziert werden, werden in den allermeisten Fällen zwar kategorisiert, diese Kategorien bekommt der Suchende aber nicht zu sehen. Eine Suche nach dem Begriff „Kohärenz" kann durchaus vorwiegend Seiten liefern, in denen der naturwissenschaftliche Kohärenzbegriff vorkommt (und tut dies auch) – eine Einschränkung auf den Kohärenzbegriff, der in dieser Arbeit verwendet wird, ist mit derartigen Suchmaschinen nicht möglich.

In den Jahren seit ca. 1994 beschäftigten sich mehrere wissenschaftlichen Einrichtungen, privaten Forschungszentren und Suchmaschinenbetreibern mit der Fragestellung, wie man Suchende bei ihrer Suche unterstützen, Suchergebnisse im WWW anschaulicher darstellen und damit brauchbarer präsentieren kann.

In dieser Arbeit werde ich einige der in dieser Zeit angedachten Visualisierungs- und Navigationstechniken für Hypertexte vorstellen. Viele von ihnen haben den Weg zu Marktreife nicht geschafft, die dazugekommenen basieren zumeist auf ähnlichen Konzepten. Der Hauptgrund scheint für mich das fehlen verlässlicher Metadaten für die Hypertextdokumente im WWW zu sein, deshalb werde ich die wichtigsten Ansätze vorstellen, die sich des Problems angenommen haben – und zum Teil ebenso in der Versenkung verschwunden sind. Eines der noch existierenden Tools mit Visualisierungstechniken werde ich anhand eines kleinen Anwendertestes auf seine Tauglichkeit

---

Eine neuere Studie zum selben Thema kommt zu völlig anderen Ergebnissen: *http://www.spiegel.de/netzwelt/netzkultur/0,1518,76499,00.html*

[3] *http://www.altavista.com/*

[4] *http://www.google.com/*

[5] *http://www.alettra.de/tools.htm*

untersuchen, indem ich es der „puren" Navigation mit Hilfe eines Webbrowsers entgegenstelle.

In einem Seitenblick werde ich mich auf die Methoden des Text Retrieval beziehen und der Fragestellung nachgehen, ob diese Methoden eine Möglichkeit bieten, aus Hypertextdokumenten automatisch semantische Strukturen zu extrahieren, um daraus Metadaten zu generieren, mit denen sich die Visualisierungs- und Navigationshilfen für die Fälle effektiver machen ließen.

Mit der Frage, welche Rolle Metadaten für die Zukunft der Informationssuche im WWW spielen können, werde ich die Arbeit beschließen.

2 Kohärenzaspekte und thematische Struktur

2.1 Text vs. Hypertext

Für alle Fragen, die sich mit der Theorie von Hypertexten beschäftigen, ist es wichtig, eine klare Definition davon zu haben, was Hypertexte überhaupt sind. Angelika Storrer führt die beiden Punkte auf, die für die Definition des Begriffes *„Hypertext*" von zentraler Bedeutung sind[6]:

1. Nicht lineare Textorganisation
   Lineare Texte zeichnen sich durch einen „roten Faden", der den Lesepfad durch die Teiltexte bestimmt, aus. Nicht-linear organisierte Texte bestehen aus autonomen Textteilen, die – in Hypertexten – durch Hyperlinks miteinander verknüpft sind. Beachtenswert ist in diesem Zusammenhang die Tatsache, dass zunehmend auch in gedruckten Texten nicht-lineare Textformen Einzug halten, so z.B. in zerclusterten serviceorientierten Zeitschriftenformaten wie „Focus". Diese sind deshalb aber noch keine Hypertexte, denn es sind immer alle Textportionen gleichzeitig sichtbar, der Leser ist in der Wahl der zu lesenden Textportion durch das Fehlen von Hyperlinks zwar völlig frei, kann aber die Anordnung der Texte nicht verändern, und gedruckte Texte lassen keine Integration von multimedialen Elementen zu. Durch die Art, wie linear angelegte Texte rezipiert werden, können sogar diese zu nicht linearen Texten werden – gerade im Rahmen von wissenschaftlichem Arbeiten werden aus Büchern, die von vorne bis hinten als linearer Text angelegt sind, oftmals bestimmte Teile sehr selektiv gelesen. „In our view, throughout the recent literature of hypertext, paper has had a bad press! Printed texts are generally *not* linear, either in their semantic structure or in the way in which skilled readers use them." [7]Auf der anderen Seiten sind auch Hypertexte, die nur einen klar definierten Lesepfad zulassen, keine Hypertexte

[6] Storrer, Angelika: „Kohärenz in Text und Hypertext" in Lobin, Henning (Hg.) 1999.

[7] McKnight, Dillon, Richardson: „Hypertext In Context", Cambridge 1991, S. 140

im Sinne dieser Definition.[8]

2. Elektronische Publikationsform
   Hypertexte müssen elektronisch publiziert werden, um die Verlinkung der Textportionen zu ermöglichen. Das Lesen in Hypertexten wird über Software, sogenannte *Hypertextsysteme* realisiert. Moderne Hypertextsysteme können über reine Textinformation hinaus auch andere Medien integrieren, so. z.B. Video, Bild und Ton – man kann diese Integration auch unter dem Begriff *Hypermedia* zusammenfassen – Hypertext schließt Hypermedia aber nach der hier verwendeten Definition mit ein.

Da die größten Teile des WWW durch Hyperlinks miteinander verknüpft sind, könnte man es auch in diese Definition aufnehmen, ich werde mich aber der Nomenklatur von Storrer[9] anschließen und unter *Hypertext* thematisch abgeschlossene Hypertextsysteme (im WWW etwa „Webs") zusammenfassen. Das WWW wäre demnach ein *Hypertextnetz*, also mehrere, miteinander verknüpfte Hypertexte.

„*Hypertext,* a term coined by Theodor H. Nelson in the 1960s, refers also to a form of electronic text, a radically new information technology, and a mode of publication."[10]

2.2 Kohärenz und Hypertexte

Zur Frage der Kohärenz in Hypertexten sind bereits einige Abhandlungen geschrieben worden[11], die zum Teil zu recht unterschiedlichen Ergebnissen kommen – dies liegt vor allem in einem unterschiedlichen Verständnis des Begriffes Kohärenz.

Die Betrachtungen dieser Arbeit stützen sich auf eine prozessbezogene Perspektive, die auf einem handlungstheoretischen Ansatz fußt. Textkohärenz entsteht erst bei der Rezeption eines Textes und

[8] Siehe z.B.: *http://gutenberg.aol.de/goethe/faust1/faust001.htm* – Goethes Faust in einer verlinkten Version als Teil des *Projekt Gutenberg.*

[9] Storrer 1999, a.a.O. S. 40

[10] Landow: „Hypertext 2.0", Baltimore 1997, S. 3

[11] Neben den in dieser Arbeit erwähnten siehe auch: Kuhlen, Rainer: „Hypertext: Ein nicht lineares Medium zwischen Buch und Wissensbank". Berlin 1991; Wenz, Karin: „Raum, Raumsprache und Sprachräume". Tübingen 1997.

wohnt dem eigentlichen Textkörper nicht inne. Deshalb können Autoren aber sehr wohl Mittel ergreifen, um die Kohärenzbildung beim Rezipienten zu steuern.

Ein Produktbezogener Kohärenzbegriff, der die Kohärenz im Textkörper verortet, ist schon deshalb falsch, weil er auf Hypertexte nicht angewendet werden kann - Hypertexte sind demnach also per se nicht Kohärent und eine Betrachtung von Kohärenz in Hypertexten macht demnach keine Sinn.

Wenn man von dieser Prämisse ausgeht, muss man sich im Zusammenhang mit Hypertexten also die Frage stellen, wie die Kohärenzbildung beim Leser in Hypertexten gesteuert werden kann, so dass für diesen ein kohärenter Text entsteht. „The primary goal of both hypertexts and linear texts is to convey information in a coherent form to a reader."[12]

Suter schreibt: „Ein Hypertext setzt sich aus einer beliebigen Anzahl *Informationseinheiten (Knoten, 'nodes')* zusammen, welche durch ein Netz von *Verweisen ('links')* miteinander verbunden sind. Das Netz von Verweisen repräsentiert die semantischen Interpendenzen zwischen den Informationseinheiten und macht so die Gesamtkohärenz eines Hypertextes aus."[13]

Hier wird deutlich, wie wichtig die Unterscheidung zwischen den verschiedenen Arten von elektronischen Texten – in diesem Fall Hypertexten und Hypertextnetzen – tatsächlich ist. Die Sutersche Definition der Gesamtkohärenz lässt sich auf Hypertextnetze eben nicht ohne weiteres anwenden. Die größte und meistgenutzte Hypertextanwendung ist sicher das World Wide Web, und gerade für diese Anwendung kann diese Definition nicht befriedigen. Die Links, die sich an vielen Stellen des WWW finden, stellen eben nicht die semantischen Interpendenzen des Hypertextes dar – zumindestens nicht alle. Wenn Hyperlinks nicht gesondert gekennzeichnet und in einer stringenten, einheitlichen Art auf thematisch verwandte Dokumente verweisen, so kann der Rezipient niemals auch nur

[12] Foltz, Peter W: „Comprehension, Coherence, and Strategies in Hypertext an Linear Text" in Rouet u.a., 1996, S. 114

[13] Suter 1995, S. 7

annäherungsweise Wissen, um was für ein Dokument es sich handelt, das ihm mit dem Link angeboten wird. Es ist wohl davon auszugehen, dass eine thematische Gesamtkohärenz des Hypertextmediums WWW nicht existiert – das verbindende Element der unzähligen Hypertexte ist höchstens ihre technische Grundlage und Realisation.[14]

2.3 Hilfen bei der Kohärenzbildung in Hypertexten

Bei der Betrachtung der Struktur von Hypertexten wird deutlich, dass zur Kohärenzbildung vier Aspekte und Ebenen der Textstruktur von Bedeutung sind: Die thematische Struktur, die funktionale Struktur und die Elementstruktur von Texten sowie Wissensaufbau und Wissensvoraussetzungen.

Einige Hilfen zur Kohärenzbildung, die sich in linearen Texten anwenden lassen, lassen sich in Hypertexten nur in seltenen Fällen auch anwenden. Einige Beispiele dazu will ich hier nennen:

Eine Fokus-Nachführung, mit der ein Autor auf einen Themenwechsel referiert, mutet in einem Hypertext, in dem der Rezipient die Themenwechsel durch aktivieren von Hyperlinks selber auslöst, seltsam an. Zudem kann der Autor oft nicht wissen, von welcher Hypertext-Portion aus der Leser auf den aktuellen Text gesprungen ist.

Auch Hilfen zur Wissensstrukturierung, mit denen Teiltexte im Gesamttext verknüpft werden, sind aus den gleichen Gründen nur selten in Hypertexten zu gebrauchen. Kohäsionsmittel, die durch explizite und implizite Wiederaufnahmestrukturen thematische Verknüpfungen herstellen, lassen sich in Hypertexten nur innerhalb einzelner Knoten verwenden, da die Bezugspunkte möglicherweise noch nicht rezipiert wurden.

Diese Bemerkungen gelten freilich nicht für Hypertexte mit fest vorgegebenen, eindimensionalen Lesepfaden. Wie bereits angemerkt unterscheiden sich solche Hypertexte kaum von linearen, gedruckten

[14] Zu Problemen von Hypertextsystemen von Benutzer- und Autorenseite siehe auch: Feldkamp, Jürgen: „Kontextermittlung und –berücksichtigung in Hypertextinformationssystemen", Hamburg 1996, S. 21ff.

Texten (das klicken auf einen „nächste Seite"-Hyperlink entspricht in etwa dem tatsächlichen Blättern in einem Buch) und fallen deshalb aus dieser Betrachtung heraus.

Für die Unterstützung der Kohärenzbildung beim Rezipienten gibt es für den Hypertextproduzenten die Möglichkeit, vordefinierte Lesepfade anzulegen, auf denen die klassischen Muster der Kohärenzbildung Wirkung haben.

Das Anlegen von Lesepfaden allerdings scheint der Idee des Hypertextes ebenso direkt entgegenzustehen, da ja gerade die frei Wahl des Lesepfades ein – gewolltes – Hauptmerkmal von Hypertexten ist. In stark hierarchischen Hypertexten, die sich ohne Querverweise als Baumstruktur visualisieren lassen, können aber durchaus klare Lesepfade vorhanden sein, ohne dass dem Rezipienten die Möglichkeit der Auswahl genommen wird – es ist quasi die Auswahl eines von sehr vielen Lesepfaden, die alle am Ursprung des Hypertextes beginnen.

Der für die Betrachtung der Kohärenz interessante Fall, dass auf ein Hypertextelement von verschiedenen anderen Elementen verwiesen werden kann, kann bei dieser Form eines Hypertextes jedoch nicht vorkommen. Die Hypertextelemente haben nur jeweils ein hierarchisch übergeordnetes Element.

Hypertexte, die nur schwach hierarchisch strukturiert sind und sich eher in einer Netzstruktur visualisieren lassen kennen diesen Fall jedoch. Ein Element des Hypertextes kann in dieser Form nicht nur auf mehrere untergeordnete Teilelemente verweisen, es kann auch mehrer über- oder gleichgeordnete Elemente haben, die auf das Element selber verweisen. Für die Betrachtung der Kohärenz ergibt sich daraus der interessante Fall, das ein Element in unterschiedlichen Lesepfaden durchaus unterschiedliche Funktionen einnehmen kann.

Dieser Umstand lässt sich in der Sprachhandlungstheorie verankern. Wie in Dialogen kann man auch in diesem Fall über indem-Zusammenhänge explizieren, welche Funktion das Hypertext-Element in diesem Fall gerade einnimmt.

So hat ein Hypertextelement, das einen bestimmten Arbeitsablauf erklärt, für einen Rezipienten, der diesen Arbeitsablauf schon kennt, eine andere Funktion als für einen, der ihn noch nicht kennt. Für

ersteren *erklärt* das Element den Vorgang, *indem* es den Vorgang beschreibt, für den zweiten *erinnert* er ihn an diesen Vorgang, *indem* es den Vorgang beschreibt.[15]

Für den Hypertextproduzenten ist dieser Umstand wichtig, muss er doch bei der Textgestaltung darauf achten, dass solche Elemente nicht zu eindeutig in Richtung einer bestimmten Funktion formuliert sind.

Hilfreich bei der Kohärenzbildung beim Rezipienten ist, dass dieser einen kohärenten Text erwartet und sich meist der besonderen Umstände bewusst ist, die ihn in einem Hypertextsystem erwarten. So wird der Rezipient bei der Nutzung eines Hypertextes automatisch stärkere Aufmerksamkeit darauf legen, an welcher Stelle im Text er sich gerade befindet, wie er an diese Stelle gekommen ist und wohin er von diesem Punkt weitergehen kann. Eine möglichst klare Auszeichnung der Sprungmöglichkeiten, die aus verschiedenen Gründen in HTML nicht einfach zu realisieren ist, ist zur Hilfestellung besonders geeignet.

Vom Rezipienten werden Strategien verlangt, mit denen er aus den sich ihm bietenden Möglichkeiten des Verzweigens im Hypertext die für seinen Informationsbedarf richtigen auswählt. Gerade bei der Suche innerhalb eines Hypertextes, beim sogenannten *Browsing*, sind diese Strategien von Belang. In thematisch weniger abgeschlossenen Hypertexten und in Hypertextnetzen wie dem WWW führen solche Browsingstrategien mitunter zu Verschiebungen von Fokus Kohärenzkriterien und damit auch der thematischen Gesamtkohärenz des rezipierten Textes; die größte „Gefahr" beim Browsing in Hypertext und Hypertextnetzen besteht jedoch darin, dass der Rezipient die thematische und die örtliche Orientierung verliert, dass er „Lost in Hyperspace" bleibt. Kohärenz stellt sich in einem solchen Fall entweder nicht ein oder geht in diesem Moment verloren. „Erschwerend kommt hinzu, dass es unglaublich viele interessante Links gibt, die einen dazu verführen, das ursprüngliche Ziel aus den Augen zu verlieren. Es ist keine Seltenheit, wenn man auf der Suche nach einem Thema urplötzlich in gänzlich abwegige Gefilde abschweift und hinterher nicht

[15] vgl. Fritz, Gerd: „Coherence in Hypertext" in Bublitz u.a. (Hg.) 1999, S. 228

mehr weiß, wie man a) überhaupt hingekommen ist und b) von dort wieder zurückkommt."[16]

Das Wissensmanagement innerhalb eines Hypertextes stellt den Produzenten vor eine schwierige Aufgabe. Hat er in linearen Texten noch die Möglichkeit, auf bereits gelesene Abschnitte zu referieren, so kann er in einem Hypertext nicht davon ausgehen, dass ein bestimmter Abschnitt tatsächlich bereits gelesen wurde. Hier kann ein System von „Vermutungen" beim Rezipienten helfen, solche Lücken im Wissensmanagement zeitweise zu überbrücken. Textabschnitte sollten so formuliert sein, dass sie bestimmte Sachverhalte implizieren, ohne sie (an dieser Stelle) dezidiert darzulegen. In einem anderen Abschnitt, der vorher, nachher oder möglicherweise überhaupt nicht gelesen wird, kann dieser Sachverhalt dann bestätigt werden. Auch hier wird vom Rezipienten eine Vorleistung erwartet, die er beim Lesen eines linearen Textes normalerweise nicht erbringen müsste. Darüber hinaus gibt es natürlich andere Möglichkeiten der Sicherung des Wissensstandes, so z.B. eine einleitende, mit Links zu Textknoten versehene Liste mit Wissensvoraussetzungen, Glossarartige Querverweise etc.

Ohnehin ist die Gestaltung der einzelnen Informationsknoten von höchster Wichtigkeit – einerseits sollten diese Knoten thematisch, funktional und kohäsiv möglichst geschlossen sein, andererseits aber durch zu großen Umfang den Leser an der freien Wahl des Lesepfades hindern.

Hypertexte können also in der Art, wie Kohärenzbildung funktioniert, stark variieren. In Hypertexten, die linear aufgebaut sind oder sehr klare Lesepfade anbieten, funktioniert die Kohärenzbildung nach den gleichen Mustern wie in gedruckten, linearen Texten. In Abstufungen lassen sich diese Methoden immer weniger anwenden. Auf der Stufe, die von linearen, gedruckten Texten am weitesten entfernt ist, dem sog. „Browsing" auf der Suche nach bestimmten Informationen, können deshalb trotzdem für den Rezipienten vollständig kohärente Texte entstehen. Wie Fritz[17] bemerkt, könnte der

[16] Apitz, Rico: „Wissenschaftliches Arbeiten im World Wide Web" Bonn 1996, S. 241

[17] Fritz in Bublitz u.a. 1999, S. 230

Rezipient den Pfad, den er beim Browsen auf dem Weg zur gesuchten Information zurückgelegt hat, vollständig dokumentieren, und anschließen feststellen: „you will probably be able to justify each individual move as a relevant step and therefore you will classify the whole search path as coherent."[18]

Die Kohärenz in einem solchen Fall zu benennen kann durchaus schwierig sein, aber solange der Fall von „Lost in Hyperspace" nicht auftritt, ist die Kohärenz offenbar nicht verlorengegangen.

Die Suche nach Informationen und vor allem das Browsing soll auch die Art der Navigation innerhalb von Hypertexten sein, mit der sich der überwiegende Rest dieser Arbeit beschäftigt.

Wie Anwender vor allem großer Hypertextsysteme auf möglichst kohärentem (und also auch kurzem) Weg zu der von ihnen gesuchten Information gelangen ist eines der großen Probleme, denen sich Hypertextautoren gegenüber sehen. Auch die Betreiber von Suchmaschinen, die Hypertextsysteme und besonders das WWW indizieren, stehen vor dem Problem, thematische Strukturen in den riesigen Netzen besser explizieren zu wollen, um Navigationsprobleme in den Hypertexten entgegenzuwirken. Einige der typischen Navigations- und Suchprobleme will ich im nächsten Abschnitt beleuchten.

[18] Fritz in Bublitz u.a. 1999, S. 230

3 Typische Probleme bei Navigation und Suche im WWW

Eine typische und besonders nützliche Hilfestellung, die Autoren von Hypertexten den Rezipienten mitgeben können, ist das bereitstellen typisierter Links, also von Verweisen, die semantische oder funktionale Aspekte der Verknüpfung explizieren können. Gerade diese Möglichkeit bietet das dem WWW zugrundeliegende Hypertext Transfer Protocol (HTTP) nicht – und aus dieser Tatsache ergeben sich eine Vielzahl von Problemen bei der Navigation innerhalb dieses größten Hypertext-netzes.

3.1 Unklare Sprungziele

Im HTTP-Protokoll sind keine Mechanismen zur Typisierung von Hyperlinks vorgesehen und es werden – zumindestens was die immer noch primär eingesetzte Seitenbeschreibungssprache HTML angeht – keine dazukommen. Eine Typisierung von Links ist für den Webautoren natürlich möglich, schließlich hat er bei der Interfacegestaltung seiner Webseite die frei Wahl – es existiert aber kein Standard für eine solche gestalterische Lösung, so dass durch verschiedenartige grafische Gestaltungen eher Verwirrung aufkommen würde. Andere Hypertextsysteme wie *Intermedia, Storyspace* oder *Microcosm*[19] bieten typisierte Links beim Erstellen der Hypertexte an. „The advantages of typed links include that, when clearly labeled, they offer a generalized kind of previewing that aids reader comfort and helps navigating information space."[20] Immerhin hat sich auf einer Reihe größerer Webseiten eine grafisches Gestaltungsmittel für die Kennzeichnung von Links, die den aktuellen Hypertext (bzw. meist den aktuellen Server) verlassen, durchgesetzt. So wissen die Benutzer wenigstens, dass sie den Kontext der aktuellen Hypertextbasis verlassen. Die Art von Links, die im WWW eindeutig vorherrschen, sind die, die Landow

[19] Keines dieser Systeme war zum Zeitpunkt der Erstellung dieser Arbeit noch erhältlich, auch die Webseiten zu den Produkten sind aus dem WWW verschwunden.

[20] Landow 1997, a.a.O., S. 16

als *String (word or phrase) to Lexia-Links*[21] bezeichnet, also Links, die als Anker ein Wort oder eine Phrase und als Sprungziel ein neues Dokument des Hypertextes haben. Diese Form des Links hat den Vorteil, dass der Rezipient das aktuelle Dokument an zumeist mehreren Stellen verlassen kann und – durch Typisierung – mehrere Linkarten zulässt.

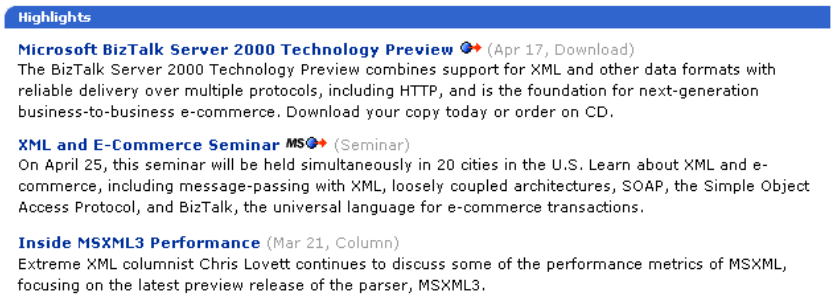

Abbildung 1: Kennzeichnung von Links, die den aktuellen Hypertextkontext (den Server) verlassen.
http://www.microsoft.com/xml/default.asp

Im Beispiel in Abb. 1 ist zwar gekennzeichnet, dass man mit dem klicken der Links die Microsoft.com-Seite verlässt, davon weiß der Benutzer aber noch lange nicht, wohin er mit diesem Link tatsächlich kommt. In diesem Beispiel liegen aber noch weitere Unstimmigkeiten vor: Der erste Link in der Liste der „Highlights" verlässt die Microsoft.com Webseite keineswegs, sondern verweist auf einen weiterführenden Artikel zum in der Überschrift genannten Thema – genau wie der dritte Link, der nicht als externer Link gekennzeichnet ist. Regelmäßige Besucher dieser Seite würden sich ohnehin darüber wundern, dass bereits in der Liste der „Teaser" externe Links auftauchen, findet man an dieser Stelle doch normalerweise *immer* erst einmal Links auf den eigentlichen, weiterführenden Artikel.

[21] Landow 1997, a.a.O., S. 12

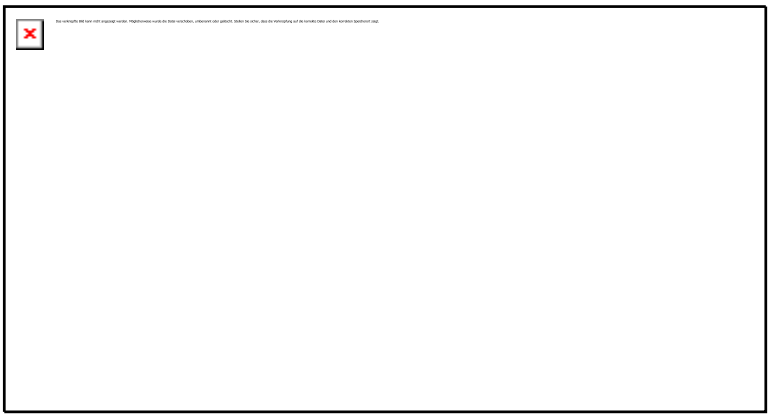

Abbildung 2: Uneinheitliche Verwendung von Hyperlinks bei *http://www.heise.de/news/*: Link auf einen älteren Artikel…

news

[<< Vorige] [Nächste >>]

Aktion: Knigge fürs Web

Foren, Gästebücher, Chats – sie sollen Raum für Meinungsaustausch und heiße Diskussionen bieten. Nicht selten gipfelt das geistige Hick-Hack jedoch in wüsten Beschimpfungen. Manche Homepage-Betreiber sehen sich in besonders hoffnungslosen Fällen sogar gezwungen, ihr Forum zu schließen und erst mal abzuwarten, bis sich die erhitzten Gemüter beruhigt haben.

Allen Homepage-Betreibern und Foren-Teilnehmern, die von derart "unterhaltsamen" Runden die Nase voll haben, will die Aktion "Keine Macht den WebIdioten" nützliche Tipps und Tricks gegen Störenfriede geben. Auf der Web-Site zur Aktion gibt's Anregungen zu Themen wie "Wie gehe ich mit WebIdioten um" oder "Mit welchen technischen Mitteln kann man WebIdioten entgegenwirken". Auch tiefe Einblicke in die Psyche der Foren-Aufmischer werden gewährt. Unterdessen scheiden sich im eigenen Forum die Geister über Sinn oder Unsinn der Aktion. Die Resonanz reicht von ungeteilter Unterstützung bis hin zu Aufrufen wie "Keine Macht dieser Site".

Abbildung 3: …Link auf eine externe Ressource.

Das Beispiel von der News-Seite des Heise Verlags[22], sorgt für noch größere Verwirrung – in diesem Fall sind die Hyperlinks überhaupt nicht gekennzeichnet und verweisen regelmäßig an unerwartete Orte. In diesem Beispiel verweist der Link in Abb. 2 auf einen anderen Artikel der News-Sektion, der eine Woche vorher erschienen war, der Link in Abb. 3 verweist auf die Webseite der Aktion „Keine Macht den WebIdioten" – wie man es wohl auch vermutet hätte. Dem geübten Anwender des WWW bietet sich immer noch die Möglichkeit, in der Statuszeile des benutzten Webbrowsers nachzuschauen, wohin der Link, auf dem der Mauszeiger gerade ruht, wohl führt – die neuen Technologien im Web – bis zu der Möglichkeit, die Statuszeile mit selbstgewählten Inhalten zu versehen – machen diese Hilfe aber zunehmend unwirksam.

[22] Interessant ist, dass der Heise Verlag regelmäßig in seinen Publikationen zu Themen wie Orientierung im WWW, XML und Standards Stellung bezieht und mit

3.2 Flüchtige Webseiten

Dezentral gelagerte Hypertexte wie die des WWW unterscheiden sich in einem ganz entscheidenden Punkt von gedruckten Texten – man kann sich nicht darauf verlassen, dass sie im nächsten Moment noch existieren. Der dezentrale Charakter des Internet und damit des WWW sorgt zwar einerseits für die unüberschaubare Informationsfülle, andererseits werden die Informationen aber auf Rechnern gelagert, die anfällig für Störungen sind, diese werden von Personen gepflegt, die von heute auf morgen entscheiden können, ein Angebot vom Server zu nehmen, Informationen können jederzeit mit minimalem Aufwand aktualisiert (und damit verändert) werden etc.. Exzellente Beispiele sind die Hypertext- und Visualisierungssysteme, die ich zum Teil in dieser Arbeit vorstelle: Produkte, die nicht mehr weitergeführt sind, sind schon nach kürzester Zeit auch nicht mehr im Web zu finden.

Navigationshilfen, die dem Benutzer von den Entwicklern der WWW-Browser an die Hand gegeben werden, hier besonders die Verwaltung von Bookmarks (Lesezeichen), die dem Benutzer ermöglichen sollen, über das Backtracking hinaus auf einzelne, bereits besuchte Webseiten direkt zugreifen zu können, verlieren durch diese Umstände an Wirksamkeit.

Neue Technologien, vor allem das dynamische Generieren von Webseiten auf dem Webserver mit Hilfe von Datenbanken, erschwert eine Verwaltung von Bookmarks noch zusätzlich. In diesem Fall ist es durchaus möglich, dass eine mit einer URL assoziierte Webseite nur das eine Mal existiert (als Dokument existiert sie im Fall einer datenbankgenerierten Seite ohnehin nicht), an dem man sie betrachtet. Fehlermeldungen wir „Diese URL ist nicht mehr gültig" werden so zum ständigen Begleiter des Websurfers, so sehr er sich auch mit der Verwaltung seiner Bookmarks Mühe gibt.

Für die Navigation innerhalb des Hypertextnetzes ist es allerdings viel bedeutender, dass durch die Unbeständigkeit von Webseiten oder

---

dem „Netzine" Telepolis (*http://www.heise.de/telepolis/*) eine Online-Publikation im Netz hat, die Wegweisende Ansätze zur Lösung solcher Probleme versucht.

ihrer URLs tote Hyperlinks in anderen Hypertext-Dokumenten entstehen können. Links, die auf nicht mehr existierende Dokumente verweisen, führen quasi sofort zu einer „Lost in Hyperspace"-Situation und bremsen den Benutzer bei der Navigation stark aus. Erschwerend kommt hinzu, dass Hypertextautoren solchen Fällen so gut wie nicht vorbeugen können.

3.3 Falsche Hyperlinks

Natürlich können Hyperlinks auch einfach auf ein falsches Dokument verweisen. Durch eine Unachtsamkeit beim Erstellen des Links kann es leicht passieren, dass beispielsweise statt auf ein in der Hierarchie tiefer liegendes Dokument auf einen Navigationsknoten auf gleicher oder höherer Ebene verzweigt wird. Zwar lässt sich ein solcher Autorenfehler vom Benutzer des Hypertextes leicht „ausbügeln", indem er einfach auf seinem Pfad durch den Hypertext einen schritt rückwärts geht – Vorraussetzung ist jedoch, dass er den Fehler sofort bemerkt.

Da Hypertextportionen möglichst einen in sich abgeschlossenen Charakter haben sollten kann es durchaus passieren, dass der Benutzer erst ein ganzes Stück des fälschlicherweise angewählten Textes lesen muss um den Fehler zu bemerken. Im schlimmeren Fall bemerkt er die falsche Verlinkung überhaupt nicht und findet sich auch hier in einer klassischen „Lost in Hyperspace"-Situation wieder – die Kohärenzbildung muss in einem solchen Fall scheitern.

3.4 Fehlende semantische Bezüge zwischen Ankern und Sprungzielen

Die geschilderten Probleme falscher und uneinheitlicher Linkauszeichnungen weisen auf ein Problem hin, das nicht auf Einzelfälle beschränkt ist: Links in HTML-Dokumenten und auch in allen anderen Hypertextsystemen, die primär von Autoren und nicht von automatischen Prozessen gewartet werden, müssen zu ihrem Sprungziel überhaupt keine semantische Beziehung haben. Wenn der Autor eines HTML-Dokumentes keine Metainformationen im Dokument

unterbringt (und die wenigsten HTML-Dokumente im WWW verfügen über aussagekräftige Metainformationen) und Links einsetzt, die auf thematisch unverwandte Dokumente verweisen, so ist die Idee von Hypertext ad absurdum geführt.

In Beispielen wie dem der Heise-Newsletter Seite sind Links zwar nicht stringent nach ihrer Funktion eingesetzt, der thematische und damit der semantische Bezug sind aber wenigstens vorhanden.

Das geschilderte Problem völligen Fehlens von thematischer Verbindung kommt in der Praxis des Internets zwar kaum vor, soll hier aber auf ein grundlegendes Problem hinweisen: Die manuelle Linkerstellung im WWW lässt Autoren alle Freiheiten nicht nur bei der grafischen, sondern vor allem bei der inhaltlichen Gestaltung. Diese Freiheiten können die Benutzer der Webseiten bei der Navigation vor ernsthafte Probleme stellen.

3.5 Mangelnde Orientierungshilfen

Für Benutzer von Hypertexten ist es wichtig, Hilfen an die Hand zu bekommen, um die aktuelle Position innerhalb des gesamten Hypertextes bestimmen zu können.[23] Die Browser, die zum Navigieren der Webseiten eingesetzt werden, bieten nur eine sehr rudimentäre Navigationshilfe: die History. Mit ihrer Hilfe kann man zwar in den meisten Fällen[24] den bereits zurückgelegten Weg wieder „rückwärts gehen" und bereits besuchte Webseiten wieder auffinden, die Position innerhalb des Hypertextangebotes jedoch lässt sich damit nicht bestimmen.

Einige Webseiten mit hierarchischer Struktur benutzen die Metapher des Dateisystems von Computern[25], um die Position innerhalb des „Navigationsbaumes" der Seite anzuzeigen. In der Tat erleichtert dies die Orientierung sehr und ist mit einfachsten Mitteln zu realisieren.

---

[23] Dies bezieht sich nicht auf die Gesamtheit des WWW. Die aktuelle Position innerhalb des WWW in textlicher Form darzustellen ist nicht möglich und wohl auch nicht hilfreich.

[24] Bei Datenbankbasierten Webangeboten funktioniert auch die Historyfunktion der Browser in vielen Fällen nicht mehr.

[25] Interessant ist dabei, das Dateisysteme von Computern dem Anwender ursprünglich mit einer Aktenschrank-Metapher nahegebracht werden mußten. Inzwischen ist wohl die Struktur von Dateisystemen mit Ordnern und Dateien so vertraut, dass sie selber als Metapher für etwas anderes herhalten kann.

Webseiten, die eine eher netzartige Struktur aufweisen, können aber auf solche Mittel nicht zurückgreifen.

Abbildung 4: Orientierungshilfe nach einer Dateisystem-Metapher auf der *Webmonkey*-Webseite *(http://www.webmonkey.com)*

4 Lösungsansätze

4.1 Suchmaschinen

Suchmaschinen sind das meistgenutzte Werkzeug von WWW-Anwendern, die auf der Suche nach bestimmten Informationen sind.

Bei Suchmaschinen kann man grundsätzlich zwischen drei verschiedenen Arten unterscheiden:

- Verzeichnisse: Hier wird das WWW oder ein Teil davon von einer Redaktion in einen Index überführt und kategorisiert. Dieser Ansatz bietet eine Reihe von Nachteilen. Eine Redaktion kann die Datenmenge des WWW niemals überschauen. Verzeichnisse wie yahoo[26] können dem Suchenden so nur einen winzigen Ausschnitt der Informationen bieten, die das WWW zu bieten hat. Der Vorteil, dass die WWW-Seiten, die solche Dienste anbieten, bereits von Menschen besucht wurden und dabei auf ihre Tauglichkeit für das Verzeichnis geprüft wurden, bedeutet zwar, dass man vermeintlich „bessere" Seiten zu sehen bekommt, leider bekommt man aber die allermeisten Seiten zum gesuchten Thema überhaupt nicht zu sehen. Auch ist es extrem schwierig, Verzeichnisse dieser Art aktuell zu halten. Ein Ansatz, diesen Problemen zu begegnen, bietet das *Open Directory Project*[27], an dem *Netscape* die Rechte hält. Dieses Verzeichnis wird – wie herkömmliche Verzeichnisse – zwar von Menschen gepflegt, im *Open Directory Project* kann allerdings jeder, der Interesse hat, „Redakteur" für eine Rubrik im Verzeichnis werden. Das Verzeichnis profitiert so vom Enthusiasmus und der Dynamik des Internet und ist damit schon sehr erfolgreich: Zum 1. Mai 2000 wurde das Verzeichnis von knapp 25.000 Redakteuren verwaltet und mehrere Seiten benutzen die Verzeichnisdaten des *ODP* auf

[26] *http://www.yahoo.com/; http://www.yahoo.de/*

[27] *http://www.dmoz.org/*

ihren eigenen Seiten.[28] Die Benutzung der Verzeichnisdaten ist – angelehnt an die Idee von *Open Source*[29], für jedermann frei. Darüber hinaus sind die Daten des *ODP* komplett in XML erfasst und über RDF kategorisiert. Wie diese Arbeit später zeigen wird, sind dies wichtige Voraussetzungen zur leichteren Navigation im WWW und zur Visualisierung von Datenräumen.

- Suchmaschinen. Suchmaschinen erledigen ihre Aufgaben – der Name deutet es bereits an – weitestgehend automatisch. Sogenannte *Crawler* oder *Spider,* automatische Prozesse der Suchmaschinen, durchforsten ununterbrochen das WWW und folgen allen Hyperlinks, auf die sie stoßen. Die Seiten, die sie besuchen, werden von einem *Parser* analysiert (dieser Vorgang unterscheidet sich bei den verschiedenen Suchmaschinen zum Teil beträchtlich) und die gewonnenen Daten anschließend in das Verzeichnis der Suchmaschine übernommen. Die Daten der untersuchten Seiten, die tatsächlich in die Datenbank der Suchmaschine kommen, sind je nach Suchmaschine unterschiedlich umfangreich. Alle Suchmaschinen übernehmen die Titel der HTML-Dokumente, die im <title>-Tag der Seite stehen, und die ersten Wörter oder Sätze, die auf der Seite stehen. Von den meisten Suchmaschinen werden auch die <meta>-Tags, die für HTML vorgesehenen Metainformationen, in die Datenbank aufgenommen. Einige Datenbanken gehen so weit, Seiten im Volltext in die Datenbank zu übernehmen, wenn sie häufig von Suchenden aufgerufen werden.
  Ein Suchender kann dann auf der WWW-Seite der Suchmaschine einen oder mehrere Begriffe und Operatoren zum Filtern der Suchanfrage eingeben, führt damit eine Volltextsuche in der Datenbank der Suchmaschine aus und bekommt die Ergebnisse in einer Liste präsentiert, die er bei einigen Suchmaschinen noch weiter durchsuchen oder filtern

[28] Unter anderem AOL, AltaVista, HotBot u.a.. Siehe *http://dmoz.org/Computers/Internet/WWW/Searching_the_Web/Directories/Open_Directory_Project/Sites_Using_ODP_Data/*

kann.

Auch Suchmaschinen erreichen mit ihren *Crawlern* nur einen Teil der im WWW verfügbaren Internetseiten, ihr Hauptproblem allerdings ist, dass sie meist eine unüberschaubare Anzahl von Suchtreffern liefern, die den Suchenden eher noch verwirren, als ihm eine gute Hilfe zu sein. Eine Suche mit der Suchmaschine *Northern Light*, die angeblich die meisten WWW-Seiten besucht und in der Datenbank abgelegt hat, nach dem Suchbegriff „+coherence +linguistics" (der Begriff „coherence" und der Begriff „linguistics" *müssen* im Dokument vorkommen) liefert alleine 10.202 WWW-Seiten als Treffer, von denen nur jeweils zehn auf einer Seite dargestellt sind. Die Kriterien, nach der die Seiten in Suchergebnissen weiter nach vorne gelangen, sind bei allen Suchmaschinen verschieden und werden von den Betreibern der Suchmaschinen sorgfältig geheim gehalten.

- Metasuchmaschinen. Diese Suchmaschinen haben überhaupt keinen eigenen Index. Sie leiten Suchanfragen an eine Reihe anderer Suchmaschinen weiter und kümmern sich nur um das „Ranking", also um die Reihenfolge, in der die Suchergebnisse der anderen Suchmaschinen aufgelistet werden und darum, dass keine Seiten doppelt aufgeführt werden. Tendenziell liefern Metasuchmaschinen auf diese Weise natürlich noch mehr Treffer als Suchmaschinen, einige Betreiber wie z.B. *Metacrawler*[30] sind allerdings dazu übergegangen, nur noch einen winzigen Teil der tatsächlichen Suchergebnisse auszugeben, um den Suchenden nicht unnötig zu frustrieren. Die Möglichkeit, dass dabei ausgerechnet die Seite nicht bei den Suchergebnissen ist, die man sehen möchte, ist hier allerdings groß. Zwei Vorteile von Metasuchmaschinen stechen allerdings heraus: Bei sehr speziellen Suchanfragen steigt die Wahrscheinlichkeit, überhaupt Treffer für die Suchanfrage zu bekommen, an und durch die verschiedenen Methoden, nach denen die von der

[29] Siehe *http://www.opensource.org/*

[30] *http://www.metacrawler.com/*

Metasuchmaschine abgefragten Suchdienste ihre Suchergebnisse gewichten, kann man davon ausgehen, dass Seiten, die von mehreren Suchmaschinen weit vorne platziert werden, tatsächlich besonders relevant für die Suche sind.

Zusammenfassend kann man festhalten: Das vorwiegende Problem der echten Suchmaschinen ist die Unübersichtlichkeit und schlechte Navigierbarkeit der Suchergebnisse. Bei Verzeichnissen sind die Ergebnisse bereits in Struktur gebracht. In Verzeichnissen wird auch eher die Technik des Browsens anstelle einer Suche angewandt. Dies erkauft man sich mit dem m.E. nicht akzeptablen Kompromiss, nur einen winzigen Bruchteil der möglichen Seiten überhaupt angeboten zu bekommen. Eine Ideale Lösung wäre eine Suchmaschine vom Umfang der existierenden Suchmaschinen mit der Möglichkeit, die Einträge der Datenbank mittels Browsing zu durchsuchen, am besten in einer optischen Aufbereitung, die der menschlichen Wahrnehmung entgegen kommt.

„Das Kernproblem [von Suchmaschinen, K.W.]: Alle Treffer werden einzeln ausgegeben. Semantische Bezüge zwischen den gefundenen Dokumenten erkennt die Suchmaschine nicht. (...) Besser ist es, die Suchergebnisse in Gestalt einer grafischen Karte auszugeben. Thematisch benachbarte Dokumente stehen dann auch nahe beieinander, wie in einer wirklichen Bibliothek mit systematischer Aufstellung, wo man ja auch rechts und links neben einem Titel Publikationen zu demselben Themengebiet findet."[31] Diese Möglichkeiten bieten wollen vor allem die Betreiber der Suchmaschinen in Zukunft anbieten können. Dies ist der Grund, weshalb die stärksten Impulse für Lösungen zur Visualisierung von Hypertextstrukturen aus den Entwicklungsabteilungen der Suchmaschinenbetreiber kommen.

Mit den Ansätzen, die zur Visualisierung von thematischen Strukturen in Hypertexten in den letzten Jahren unternommen wurden, will ich mich im nächsten Kapitel befassen.

[31] Vogt, Petra: „Datenlandkarten" in: c't Magazin für Computertechnik, 5/1998, S. 204

4.2 Visualisierung

Die Anforderungen, die an die Visualisierung von Hypertextstrukturen zu stellen sind, lassen sich gut aus den Problemen ableiten, die bei der Betrachtung der Suchmaschinen ins Auge gefallen sind. „A great search algorithm isn't terribly useful if it's interface isn't designed for its users or if it isn't sensitive to the peculiarities of the documents being retrieved. Similarly, a wonderful interface is useless unless it can be used to retrieve documents that people need.“[32]

Das Problem, dass zu einem Suchbegriff zu viele Treffer ausgegeben werden, lässt sich mit Hilfe von Visualisierungstechniken nur schwer beheben, solange von Suchmaschinen immer – auch – im Volltext gesucht wird. Eine Suche in Konzepten oder thematischen Bezügen wäre aber nur mit Hilfe von Metadaten möglich, die im WWW (noch) nicht in ausreichendem Masse eingesetzt werden.

Die meisten existierenden Methoden der Visualisierung, die ich hier vorstellen werde und die sich auf Inhalte im WWW beziehen, extrahieren ihre Bezüge daher nur aus tatsächlichen, im Hypertext existierenden Links.

Systeme wie *Apples Hotsauce* wurden zwar auch vor allem zur Visualisierung von Angeboten im WWW genutzt und verließen sich nicht auf die Verknüpfung mittels Links in Hypertexten, in diesem Fall wurde die Visualisierungsinformation allerdings aus einer Datei mit Metadaten bezogen, die mit dem eigentlichen Hypertext in keinem – technischen – Zusammenhang steht, d.h. die Visualisierung würde auch immer noch funktionieren, wenn der eigentliche Hypertext gar nicht mehr existiert und vor allem gibt es ohne diese – manuell anzulegenden – Metadaten-Datei auch keine Visualisierung.

Fowler spricht einen weiteren wichtigen Punkt bei der technischen Umsetzung an: „(...) visualization of semantic information spaces must provide: 1) spatialization of the abstract data. which may entail both a) data organization and b) derivation of a visual spatial repesentation

[32] Golovchinsky, Gene: „Reaction to SIGIR 99 Panel on User Interface Issues“, Palo Alto 1999.

of data, 2) presentation of the spatial representation in a display space for user interaction and viewing. such as Euclidian two- or three-dimensional space or a distorted space, and 3) techniques and tools for user interaction with the visual representation."[33]

Neben der eigentlichen, räumlichen Darstellung der Informations-einheiten und ihrer Verknüpfungen müssen auch Möglichkeiten geschaffen werden, sich innerhalb dieser räumlichen Darstellung zu bewegen und mit der Darstellung zu interagieren, also quasi im Informationsraum zu *Navigieren*, um die Weltraum-Metapher aufzugreifen, die einige der Systeme bemühen.

Konkret müssen Benutzer sich innerhalb der Darstellung von Knoten zu Knoten bewegen können, entweder nur über Verbindungs-linien – sofern vorhanden – oder aber völlig frei. Es sollte die Möglichkeit geben, an den Ursprungsknoten im Raum zurückzu-springen, wünschenswert wäre, den zurückgelegten Pfad speichern zu können (ähnlich wie dies in Browsern bereits geschieht), um bereits besuchte Punkte leichter auffinden zu können. Methoden zur Fokussierung der Bereiche, in denen man sich gerade aufhält, sind oft das primäre Unterscheidungsmerkmal der verschiedenen Produkte. Schließlich sollte die Navigation ermöglichen, nicht nur die Titel der Knoten anzuschauen, sondern das tatsächliche Dokument aufzurufen, um es studieren zu können.

Bevor ich auf einzelne Produkte zur Visualisierung eingehe, will ich noch zwei Ansätze zur Darstellung von Datenstrukturen und –Räumen erwähnen, die meines Wissens in kein konkretes Produkt zur Visualisierung von Hypertexten gemündet sind:

***Perspective Wall*:** Diese Methode der Visualisierung eignet sich vor allem für die Darstellung linearer Informationsstrukturen.

[33] Fowler, Richard H. u.a. 1998 a.a.O. (online)

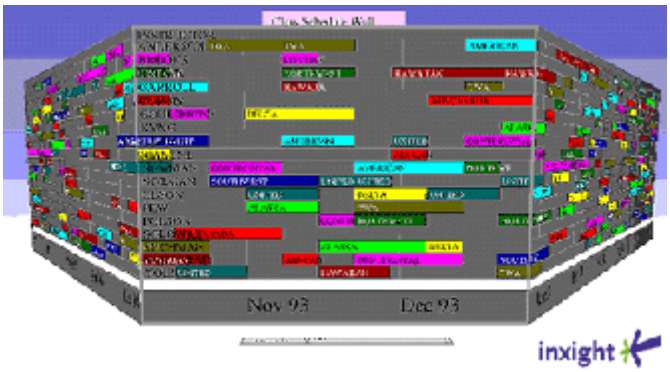

Abbildung 5: Ein Dateimanager mit *Perspective Wall* Darstellung.

Die *Perspective Wall* ist – wie in der Abbildung zu erkennen – eine nach hinten in den dreidimensionalen Raum gefaltete Wand. Der Informationsfokus liegt auf der dem Benutzer zugewandten Seite, zur Navigation kann er von rechts oder links Informationen auf diese Seite holen oder den Winkel, in dem die Wände gefaltet sind, ändern. Im Beispiel werden Dateien nach der Zeit, in der sie bearbeitet wurden, sortiert ausgegeben. *Perspective Wall* war ursprünglich ein Teil von *Inxights* Visualisierungsmodulen *VizControls*, die Entwicklung und der Support wurden allerdings eingestellt – bei *Inxight* wurden alle Anstrengungen auf die Weiterentwicklung der *Hyperbolic Tree* Technik, von der später noch die Rede sein wird, verlegt.

***Cone Trees*.** Bei dieser Visualisierungsmethode, die auch Teil der *VizControls* der Firma *Inxight* waren, wurde die dritte Dimension stärker zur tatsächlichen Darstellung von Information genutzt. Die Darstellung entspricht etwa der um die dritte Dimension erweiterten Baumstruktur, mit der sich hierarchische Dokumentstrukturen gut darstellen lassen – sie kommt damit dem eigentliche Baum sehr nahe. Von jeweils einem Knoten zweigen die Äste zu untergeordneten Knoten ab, die Kegelförmig (daher *Cone* Trees) unter bzw. rechts vom Ausgangsknoten angeordnet sind. In der Darstellung lassen sie alle *Cones* frei drehen, so dass man jedes Element in den Vordergrund

befördern kann.

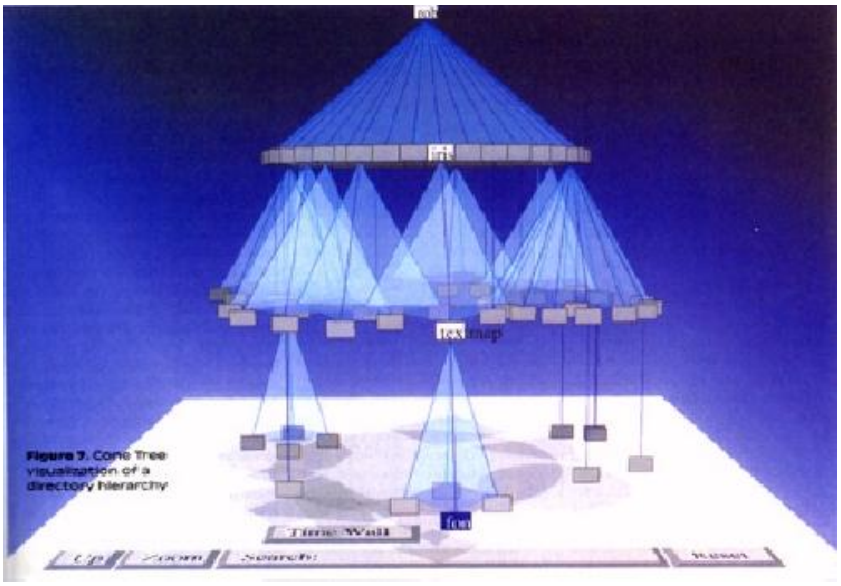

Abbildung 6: *Cone Tree* Darstellung einer Verzeichnisstruktur. Quelle: *http://www2.iicm.edu/0x811bc92d_0x0002e396*

Zur Darstellung hierarchischer Datenstrukturen eignen sich *Cone Trees* schon sehr gut; durch das Fehlen einer Entsprechung der sog. *Fisheye-Technik*, wie sie z.B. bei der *Hyperbolic Tree* Visualisierung eingesetzt wird, eignet sie sich allerdings nicht für unendlich große Datenbestände. Auch die Navigation durch drehen der einzelnen Kegel kann zum Teil recht lange dauern.

Im folgenden will ich die anderen Darstellungs- und Navigations-methoden, die sich bei der Visualisierung von Hypertextstrunkturen einsetzen lassen, anhand von Produkten vorstellen, die allerdings wie eingangs erwähnt zu einem großen Teil schon nicht mehr existieren.

4.2.1 Eingestellte Systeme

Im Beitrag „Datenlandkarten" vom Mai 1998[34] gab Petra Vogt einen Überblick über Systeme, die sich mit der Visualisierung von Strukturen innerhalb des WWW und von Hypertexten auseinander setzten. Der Überblick vermittelte den Eindruck, als stünde die breite Durchsetzung von Visualisierungstechniken kurz vor dem Durchbruch: zwölf Systeme mit unterschiedlichen Ansätzen aus renommierten Forschungsanstalten wurden in dem Beitrag beschrieben, von denen die meisten bereits in experimentellem Einsatz waren.

Heute stellt sich die Situation anders dar: Die Entwicklung von neun der zwölf vorgestellten Systeme wird nicht mehr weitergeführt, zum Teil sind kaum noch Hinweise auf deren Existenz im WWW selber zu finden. Die Systeme, die „überlebt" haben oder danach noch dazu gekommen sind, verfolgen im großen und ganzen den selben Ansatz.

Auch beim *„InfoViz*"-Projekt der FH Potsdam, das sich mit allen Arten der wissenschaftlichen Visualisierung beschäftigt, funktionieren auf der Übersichtsseite, die einen hervorragenden Überblick über Projekte zum Thema geben könnte, ein großer Teil der Hyperlinks nicht mehr.[35]

Ich will exemplarisch einige der eingestellten Systeme trotzdem vorstellen um eine Erklärung zu versuchen, warum die Entwicklung nicht sinnvoll weitergeführt werden konnte.

4.2.1.1 Hotsauce

Hotsauce[36] wurde von *Apple Computer*[37] 1996 eingeführt. Das Tool diente vor allem zur halbautomatischen Erstellung von Sitemaps aus einer reinen Textdatei, ähnlich wie beim Internet-3D Standard VRML[38]. Um die *Hotsauce*-Sitemaps ansehen zu können, benötigt man ein Plugin für den benutzen Internet-Browser.

Die eigentliche Visualisierung eignet sich besser für hierarchisch gegliederte Informationsangebote als für stark netzähnliche Strukturen. Von einem Startpunkt aus bewegt man sich mittels der Maus in einem pseudo-dreidimensionalen Raum. Je tiefer man in den Raum eindringt, desto tiefer kommt man in der Struktur des Verzeichnisses nach unten. Dabei kann man mit der Maus die Richtung bestimmen, in die man sich bewegen will. In dem abgebildeten Beispiel (Abb. 5) kann man sich so beispielsweise mit der Maus in Richtung des

---

[34] Vogt 1998

[35] *http://fabdp.fh-potsdam.de/infoviz/repository.html*

[36] Die offiziellen WWW-Seiten zu Apples *Hotsauce* sind inzwischen nahezu vollständig verschwunden. Eine (nicht mehr aktualisierte) Seite mit Beispielen und einer Downloadmöglichkeit für das Plugin findet sich unter *http://www.xspace.net/hotsauce/*

[37] *http://www.apple.com/*

[38] Nähere Informationen zu VRML gibt es u.a. beim Web 3D-Consortium unter *http://www.vrml.org/*

Knotens *Buildings* bewegen, während man darauf zu fliegt. Wenn man sich einem bestimmten Punkt nähert, erscheinen weitere, diesem Punkt untergeordnete Punkte im virtuellen Raum, bis man auf der untersten Hierarchieebene angekommen ist.

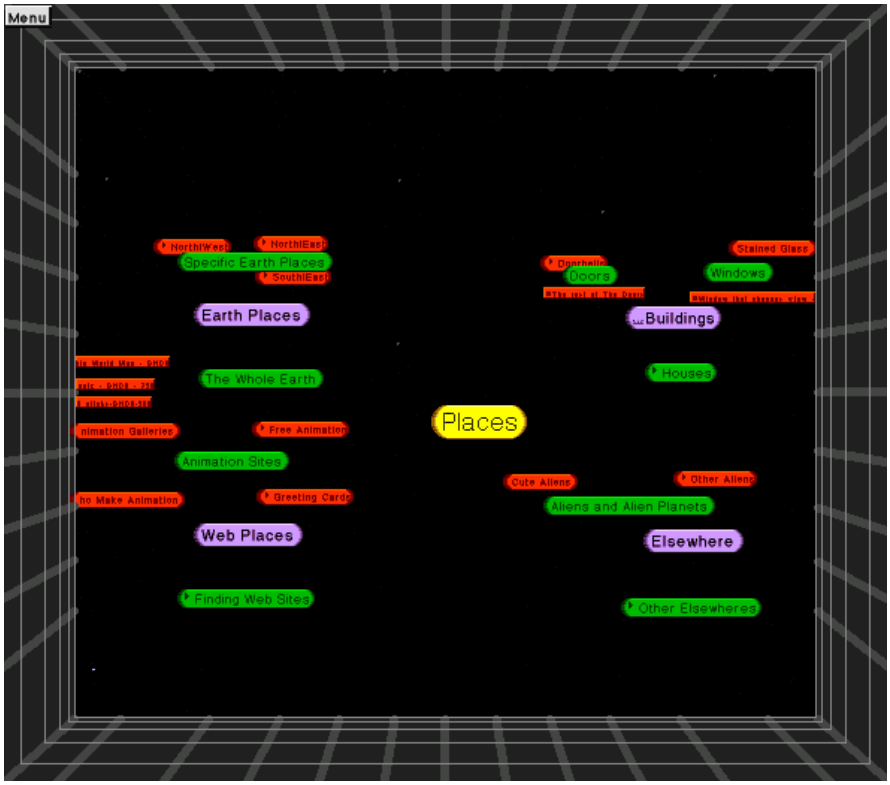

Abbildung 7: Mittels *Hotsauce MCF* generierte Map. *http://www.teleport.com/~cooler/MMMM/meta/MMB.mcf* (Hotsauce-Plugin wird benötigt)

Wie erwähnt werden die Darstellungen aus einer Textdatei generiert, die im MCF (Meta Content Format) geschrieben ist. Dieses Format wurde von *Apple* extra für diesen Zweck entwickelt und ist der interessanteste Aspekt der *Hotsauce*-Technologie. In dem reinen Textformat wird jeder Knoten einzeln beschrieben. Es gibt für jeden Knoten einen Eintrag, der den Namen des Knotens, ein evtl. dazugehörendes Dokument (eine URL) und seine Position innerhalb des *X-Spaces* (so heißen die virtuellen Räume, die durch eine *MCF*-Datei beschrieben werden) beschreibt. Diese Position setzt sich aus dem übergeordneten Knoten und einer zweidimensionalen Koordinate in Raum zusammen (die Tiefe des Knotens wird durch die Hierarchieebene bestimmt). Ein Ausschnitt aus einer solchen MCF-Datei ist zum besseren Verständnis hier abgebildet:

```
begin-headers:
MCFVersion: 0.95
name: "Animals"
end-headers:

unit: "Dogs.mcf"
name: "Dogs"
; no parent slot required since this is a child of the node
corresponding to this file.

unit: "Cats.mcf"
name: "Cats"

unit: "Pets.mcf"
name: "Pets"

unit: "FamousDogs.html"
name: "Famous Dogs Page"
parent: #"Dogs.mcf"
```

Zur Erstellung solcher Dateien gab es einige Autorensysteme, die zwar eine Arbeitserleichterung bedeuteten, dem Autoren die manuelle Erstellung der Dateien aber nicht abnehmen konnten. Einige Webseiten, die *MCF* zur Anzeige ihrer Sitemaps einsetzten, schrieben eigene Skripte auf dem Webserver, die automatisch die *MCF*-Dateien aktualisierten, wenn sich in der Struktur der Seite etwas änderte.

Noch 1997 war das Interesse an *Hotsauce* groß, vor allem *Netscape* wollte Unterstützung für *MCF* in seine Software einbauen und beteiligte sich an der Weiterentwicklung. Auch beim WWW-Konsortium (W3C) war man seit längerem an einer Standardisierung für Metadaten im WWW interessiert. Als Apple in starke finanzielle Schwierigkeiten geriet, wurde die komplette Spezifikation des *MCF* dem W3C als Vorschlag übergeben und die Weiterentwicklung und der Support bei *Apple* und *Netscape* komplett eingestellt. Beide Firmen beteiligten sich jedoch an der W3C-Gruppe zur Erarbeitung eines Metadaten-Standards auf der Basis von XML, der jetzt als verabschiedeter Standard unter dem Namen RDF (Resource Description Framework) vorliegt (Siehe Kapitel 4.3.3). Einige Konzepte von MCF wurden für RDF übernommen, so dass Apples Vorstoß in diese Richtung sicherlich ein Wegweisender Schritt war. Es wurden auch einige Tools entwickelt, um MCF-Dateien automatisch in RDF umzuwandeln. Aus heutiger Sicht ist die Einstellung der Weiterentwicklung nicht als besonders tragisch zu bewerten, da RDF als Metadaten-Format flexibler und leistungsfähiger ist und das Visualisierungsmodul von *Hotsauce* die Möglichkeit missen lässt, netzartige Strukturen und die tatsächliche Verlinkung von

Dokumenten abzubilden. Ein klarer Vorteil des Systems – die Tatsache, dass die Visualisierung nicht auf Links, sondern auf thematischen Strukturen beruht – wird durch die manuelle Erstellung der MCF-Dateien erkauft. Mit Tools zur Extrahierung von semantischen Bezügen wie die später beschriebenen von *Autonomy* oder *Inxight* ließen sich mit wenig Aufwand durchaus automatisch brauchbare MCF-Dateien erzeugen.

4.2.1.2 Perspecta SmartContent

Das *Perspecta SmartContent* System wurde von einer Gruppe von ehemaligen M.I.T. Mitarbeitern entwickelt und wurde Mitte des Jahres 1997 vorgestellt. Die Firma *Perspecta,* obwohl finanziell von M.I.T.-Direktor Nicholas Negroponte und dem Datenbankhersteller Informix unterstützt, existiert inzwischen nicht mehr. Mit der Webseite der Firma sind auch so gut wie alle technischen Unterlagen verschwunden. Alles, was es zu dem Produkt und seinen Verfahren zu sagen gibt, lässt sich nur noch aus Tests der Software rekonstruieren, die damals erschienen.[39]

Perspecta verwaltet Daten und Webinhalte in einer Datenbank mittels Metainformationen. Dabei verwendete es zunächst ein proprietäres Metadatenformat, später (1998) wurde die Unterstützung von XML zumindest angekündigt. Mit Hilfe dieser Metadatenbank konnten Informationsportionen in verschiedene Rubriken und unter verschiedene Themen *(Topics)* eingeteilt werden, so dass in der Gesamtarchitektur ein Dokument an verschiedenen Stellen auftauchen kann. So sollte es möglich sein, sich Informationen von verschiedenen Stellen zu nähern und von dort auf vielfältige Weise weiter zu navigieren.

Perspecta verfügte, wie die aktuellen in dieser Arbeit besprochenen Systeme von *Autonomy* und *Inxight* über Tools zur automatischen Extrahierung von thematischen Strukturen, die Metadaten aus

[39] siehe u.a.
http://webreview.com/pub/p/Perspecta_SmartContent(tm)_System
http://www.xml.com/pub/SeyboldReport/ip020528.html
http://www.zdnet.com/pcweek/news/0630/30persp.html

bestehenden Texten generierten. Welche Verfahren dazu eingesetzt wurden, war heute nicht mehr festzustellen.

Die *Perspecta*-Software setzte sich aus dem *SmartContent* System, das auf einem Webserver aufsetzte, und einer in *Java* geschriebenen Clientsoftware, die im Internetbrowser lief, zusammen. Die Clientsoftware besorgte die Visualisierung der thematischen Strukturen in einer ähnlichen Weise, wie sie auch von *Hotsauce* angeboten wurde, allerdings wurden in *Perspecta* auch Verknüpfungen, sowohl reelle Hypertext-Links als auch thematische Verwandtschaften in der Datenbank, mit Hilfe von Linien zu anderen Knoten angezeigt.

Über den Grund, warum *Perspecta* scheiterte, lässt sich nur mutmaßen. Ein Grund war sicherlich, dass die Software auf Webserver beschränkt war, die unter dem Betriebssystem *Solaris* liefen und mit Preisen ab $30.000 nicht gerade ein Sonderangebot war. Auch die Unterstützung von etablierten Standards wie XML kam für den Markt vermutlich zu spät. Einige der Entwickler, die an *Perspecta* mitgearbeitet hatten, arbeiten heute beim damaligen Konkurrenten *Inxight,* der mit seiner Software ähnliche Wege geht.

Die Ansätze der Software, die thematische Struktur der verwalteten Dokumente mittels Metadaten abzubilden und räumlich zu visualisieren, waren auf jeden Fall zukunftsweisend.

4.2.1.3 Bookmark Exploring Dabbler (BED)

Der *Bookmark Exploring Dabbler*, ein Projekt des *Swiss Federal Institute of Technology,* das 1996 vorgestellt und seit 1997 nicht mehr weiterentwickelt wird, ist am ehesten wegen seiner Darstellungstechnik interessant. Das Programm diente einzig dazu, ein Verzeichnis von Netscape-Bookmarks (die in Netscape als HTML-Datei abgespeichert werden) in eine VRML-Datei zu überführen. VRML ist ein Standard zur Darstellung von 3D-Grafiken im WWW.

Bookmarks werden in Netscape von Benutzerseite aus immer in eine Ordnerstruktur sortiert, so dass die Bookmarks in einer hierarchischen Ordnung vorliegen. Diese Ordner werden in der VRML-Darstellung als *Planeten* dargestellt, die in einem Ring um einen

sternförmigen Mittelpunkt angeordnet sind. Von diesen Planeten gehen jeweils Linien zu weiteren, kleinen *Planeten (Monden)* ab, die die Unterordner abbilden. Innerhalb dieser *Monde* befinden sich kastenförmigen Elemente, die für die eigentlichen Bookmarks stehen – man muss also in das Objekt hinein manövrieren um an die Bookmarks zu gelangen. Diese Bookmarks können angeklickt werden, um zur eigentlichen Seite zu gelangen.

Für die Navigation in diesem Informationsraum bieten sich die Navigationsmittel des Programms an, das jeweils zur Darstellung von VRLM-Dateien *(Worlds)* genutzt wird – und davon gibt es viele. VRML ist aber eher für Rundgänge oder –flüge in virtuellen Räumen angelegt als für die Navigation in Datenräumen; so bietet VRML keine Möglichkeiten, das 3D-Modell zu ändern, Elemente aus- oder einzublenden etc. während man sich in dem Raum bewegt. Bei sehr großen Räumen kann die Übersichtlichkeit so schnell verloren gehen.

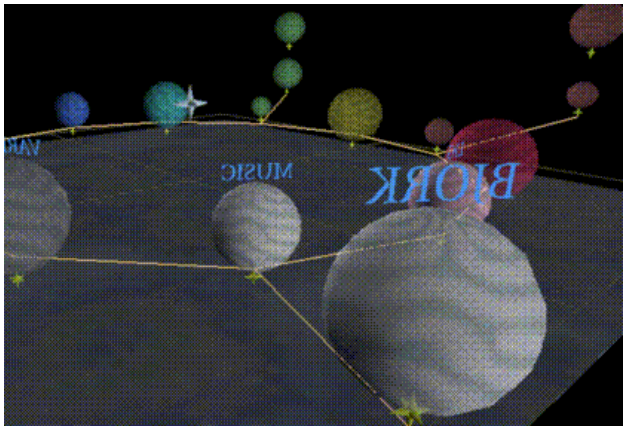

Abbildung 8: Eine Netscape-Bookmark-Datei, die mit *BED* in VRML umgewandelt wurde. Die eigentlichen Dokumentenverweise sind noch nicht zu sehen.

Der Ansatz von *BED* ist aber trotzdem interessant, weil er einen sehr einfachen Einstieg in die Welt der Visualisierung von Datenstrukturen mit bereits bestehenden, standardisierten Techniken bot. Zum Projekt existiert eine Webseite *(http://ligwww.epfl.ch/-~rezzoni/VG/tform.html)*[40], auf der man seine eigene Netscape-

[40] Zum Zeitpunkt der Anfertigung der Arbeit war die Funktion außer Betrieb, unter der selben Adresse lassen sich allerdings Beispiele von bereits generierten *Worlds* ansehen.

Bookmarkdatei in eine VRML-Datei umwandeln kann, um sich einen Eindruck vom Projekt zu verschaffen.

4.2.1.4 AltaVista Cow9

Über AltaVista Cow9 lassen sich heute kaum noch Informationen im WWW oder an anderen Stellen finden. Die Technologie, mit der *AltaVista*, die zu diesem Zeitpunkt (*Cow9* wurde 1997 vorgestellt und bereits 1998 von der Webseite genommen) erfolgreichste Suchmaschine im Internet, die Suche im Internet revolutionieren wollte, scheiterte auch an verlässlichen Mechanismen zur Kategorisierung von Webseiten.

Das Prinzip war, über eine grafische Darstellung von Kategorien, die zu einer Suchanfrage (in einem normalen *AltaVista* Suchfeld) passen, Suchergebnisse auf relevante Inhalte beschränken zu können. In der Abbildung lässt sich dieses Prinzip gut nachvollziehen.

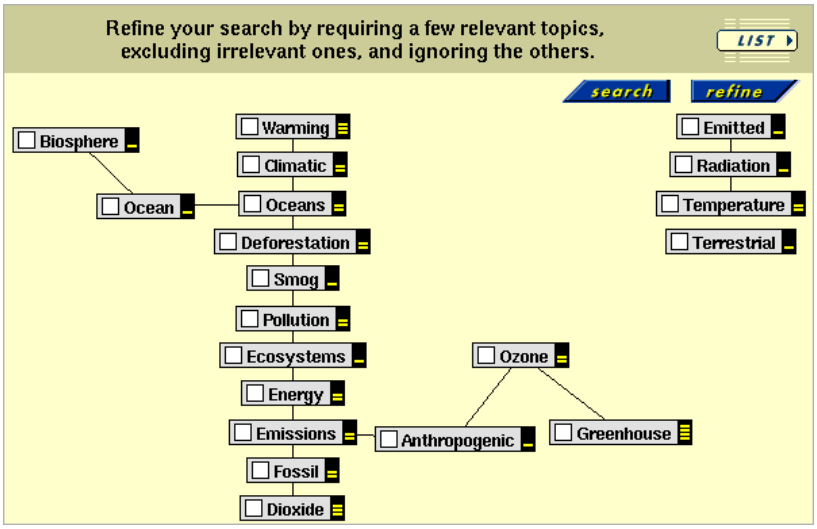

Abbildung 9: *AltaVista Cow9*-Darstellung von Themen, die für die Suchanfrage „Greenhouse Effect" relevant sind. Die Balken neben den Schlagwörtern geben die Relevanz im Bezug auf die Suchanfrage an.[41]

Die Basis dieses Prinzips war natürlich eine Kategorisierung der bei *AltaVista* verzeichneten Seiten in Kombination mit einer Datenbank an Schlagwörtern, die miteinander verknüpft waren. Dies wurde durch eine Datenbank von Metadaten versucht zu erreichen, die in einem

[41] Quelle: *http://www.cma.ensmp.fr/Francois.Bourdoncle/ina.html*

eigenständigen Format jede Webseite bestimmten Schlagworten zuordnen konnte. Diese Kategorisierung fand zum Teil mit Systemen statt, die denen von *Autonomy* (siehe dort) ähnelten, zum anderen wurde sie aber auch manuell vorgenommen.

Wie man an dem Beispiel des Suchbegriffes „Greenhouse Effect" in der obigen Abbildung sehen kann, war das System durchaus in der Lage, für bestimmte Begriffe relevante Kategorien zu bestimmen und diese in einer Art Karte von Beziehungen auszugeben. Das eigentlich neue an *Cow9* war aber genau diese Visualisierung – das System der Kategorien existierte bereits vorher: über einen Textlink, der mit „refine search" bezeichnet war, konnte man die Kategorien in einer einfachen Listenform ausgeben lassen und seine Suchergebnisse durch klicken auf eine dieser Kategorien auf Seiten beschränken, die den Suchbegriff enthalten *und* dieser Kategorie zugeordnet sind. Kritiker[42] merkten an, das die Art der Visualisierung der unbestrittenen Nützlichkeit der „refine search"-Funktion nichts hinzufügen konnte. Dies kann als Grund gewertet werden, warum AltaVista das Projekt fallen lies; über den Grund, warum sie die „refine search" Funktion gleich mit abschafften, kann man nur spekulieren.

Auch hier liegt die Annahme nahe, dass es an leistungsfähigen Werkzeugen zur automatischen Kategorisierung der Webseiten mangelte. Die Kooperationen, die *AltaVista* inzwischen eingegangen ist, und die Initiative, die sie mit *Discovery* (siehe dort) zeigen, deuten aber darauf hin, das weder die Idee der Kategorisierung noch der Visualisierung fallen gelassen wurden. Auch *AltaVista* bedient sich in seinem Webkatalog (*Directory)* – also nicht in der Suchmaschine *(Index)* – der Daten des *Open Directory Project* und es ist davon auszugehen, dass zumindest dieser Teil des Dienstes mit Metadaten im offenen RDF-Standard abgelegt sind. Mittelfristig strebt AltaVista an, 90% der mittels *crawling* besuchten Seiten in Kategorien im Verzeichnisteil des Dienstes einzusortieren[43] – ein ehrgeiziges Ziel, das aber sicher die meisten Suchmaschinenbetreiber verfolgen.

[42] siehe u.a. Vogt 1999, S. 207f.

[43] siehe *http://doc.altavista.com/company_info/about_av/background.shtml*

4.2.2 Aktuelle Systeme

Einige der vorgestellten Konzepte zur Visualisierung von Hypertext-strukturen und Datenräumen wurden von den Entwicklern nicht weitergeführt, weil das manuelle Aufarbeiten der thematischen Strukturen in den Hypertexten mit zu viel Arbeit behaftet war und automatische Verfahren entweder keine zufriedenstellenden Ergebnisse lieferten oder nicht zur Verfügung standen.

Von den Systemen zur Visualisierung, die noch existieren und die noch weiter entwickelt werden, will ich exemplarisch einige vorstellen, auf die auch Endanwender ein Auge werfen können, ohne gleich astronomische Summen für die Software ausgeben zu müssen.

4.2.2.1 Manuell generierte Sitemaps

Eine der populärsten Methoden, die Struktur einer Webseite darzustellen um dem Benutzer eine Orientierungshilfe an die Hand zu geben, ist das Erstellen sog. „Sitemaps" – kaum ein größeres, kommerzielles Internetangebot verzichtet heute noch darauf.

Da Sitemaps von den Webseiten-Entwicklern angelegt und auch frei gestaltet werden, kann man Aussehen und Funktion nicht sinnvoll beschreiben. Das Beispiel der Sitemap der Firma *Dynamic Diagrams* in Abb. 9 gibt ein gutes Beispiel für eine typische Sitemap: Die Struktur der Seite mit ihren verschiedenen Bereichen ist gut zu erkennen und passt (in der Abbildung nicht zu sehen) auch noch zum Erscheinungs-bild der restlichen Webseite. Durch das Anklicken der Dokumente in der Sitemap kann der Benutzer direkt zum gewählten Dokument springen, ohne die sonst vorgesehene Navigation in Anspruch nehmen zu müssen.

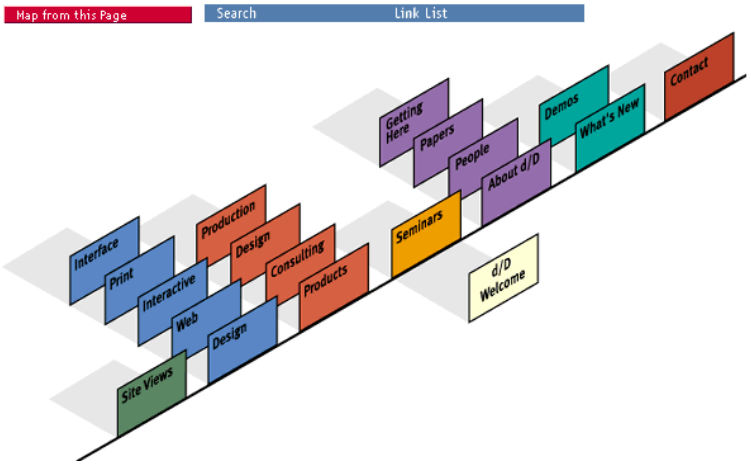

Abbildung 10: Manuell generierte Sitemap[44]

Der größte Nachteil solcher Sitemaps liegt auf der Hand: Sie müssen manuell erstellt werden. Je nach grafischem Aufwand ist das erstmalige Erstellen einer solchen Sitemap bereits recht zeitraubend, vor allem aber müssen die Sitemaps auch manuell geändert werden, wenn sich in der Struktur der Seite etwas ändert. Größere Webangebote lassen sich zudem nicht mit Sitemaps darstellen. Die in Abb. 10 dargestellte Sitemap der *Holtzbrink Verlagsgruppe* beispielsweise lässt sich im WWW überhaupt nicht mehr darstellen – die hier abgebildete Sitemap ist vielmehr ein Poster mit einer Kantenlänge von immerhin 2 Metern.

[44] Sitemap von Dynamic Diagrams *http://www.dynamicdiagrams.com/*

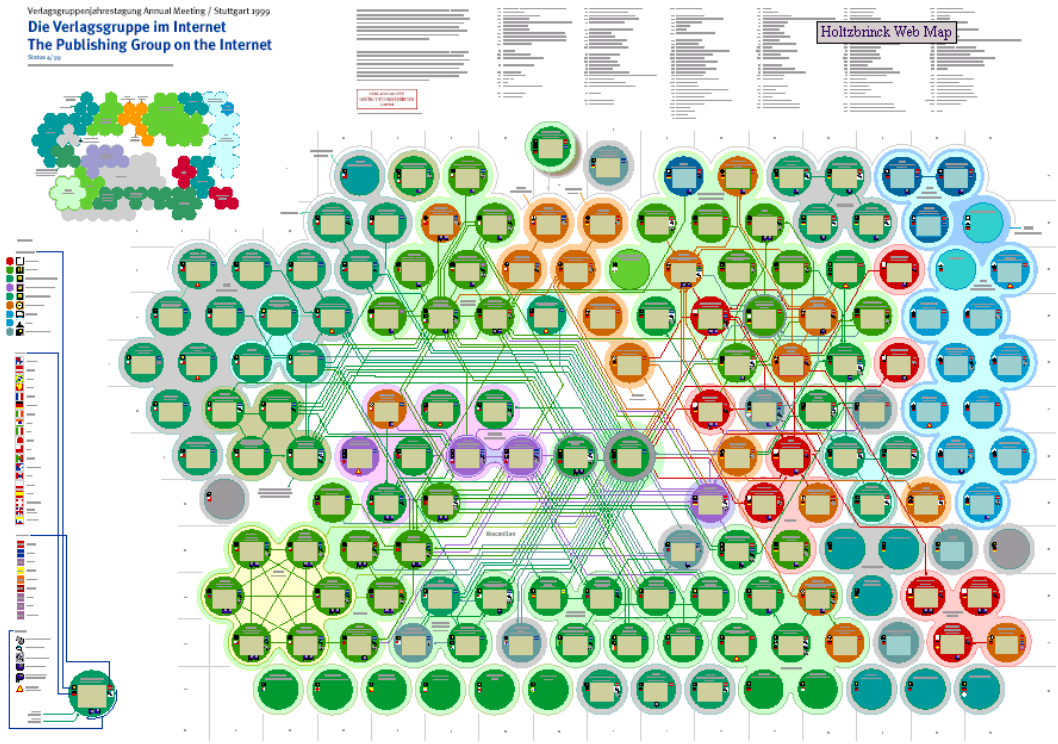

Abbildung 11: Manuell gestaltete Sitemap für mehrere Sites der Holtzbrinck Verlagsgruppe (existiert nur als Plakat, nicht online).[45]

Manuell generierte Sitemaps können auch mittels Skriptsprachen oder anderer Animationstechniken eine dynamische Darstellung ermöglichen: Die in Abb. 11 gezeigte Sitemap der *Daimler Chrysler AG,* die übrigens nicht bis auf die Stufe der einzelnen Dokumente herunterreicht, um die Übersichtlichkeit nicht zu gefährden, lässt die untergeordneten Knoten erst erscheinen, wenn man den übergeordneten Knoten anklickt. Das ändert allerdings nichts daran, dass auch diese Sitemap manuell gewartet werden muss, wenn sich an der Seitenstruktur etwas ändert.

[45] Sitemap erstellt von dynamic Diagrams (*http://www.dynamicdiagrams.com*).

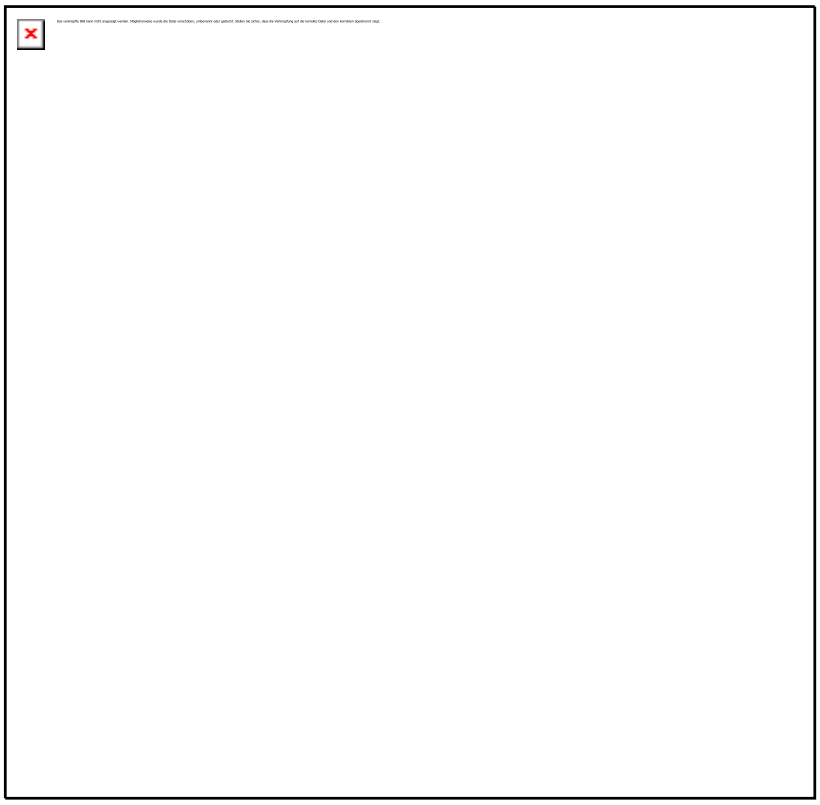

Abbildung 12: Manuell erstellte, interaktive Sitemap der Daimler-Chrysler AG (*www.mercedes-benz.de*).

4.2.2.2 Automatisch generierte Sitemaps

Einen Ausweg aus diesem Dilemma können Programme bringen, die Sitemaps selbstständig erstellen können. In den meisten Fällen sind dies Autorenwerkzeuge zur Erstellung von Webseiten. Diese können, wie *Visual Interdev* von *Microsoft* in Abb. 12, Sitemaps aus den in HTML angelegten Linkstruktur extrahieren. Da diese Sitemaps aber immer auch alle Querverweise mit anzeigen, kann die hierarchische Ordnung der meisten Webseiten auf diese Weise oft nicht sinnvoll dargestellt werden.

Andere Programme wie *Allaire Fusion* oder *Microsoft Frontpage* bieten dem Benutzer eine Sitemap-Übersicht, in der er neue Webseiten anlegt und selber auf die Hierarchieebene achten muss. Querverweise werden in diesen Darstellungen nicht angezeigt, Änderungen aber automatisch übernommen.

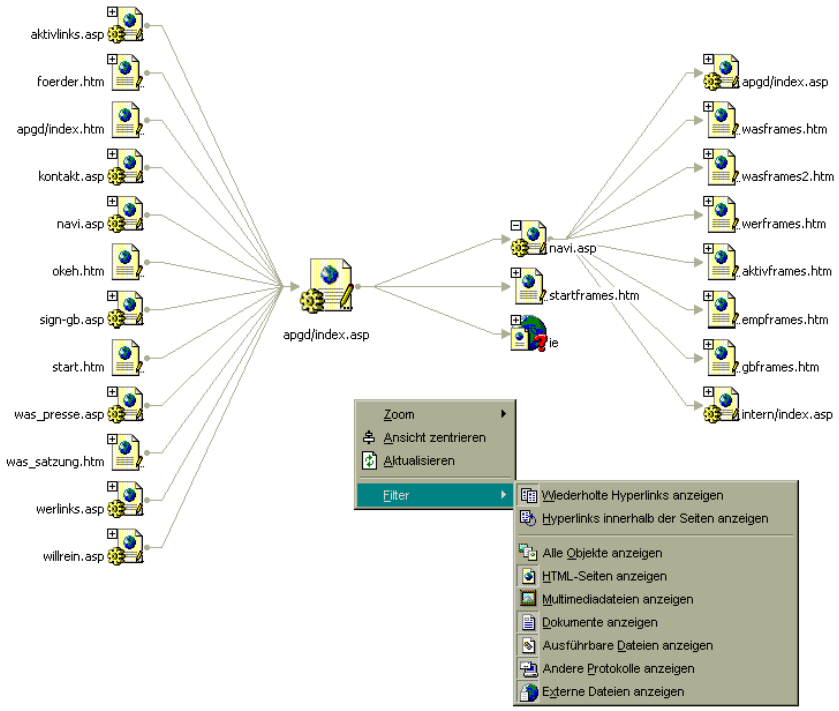

Abbildung 13: Automatisch generierte Sitemap aus einem Webentwicklungs-Werkzeug[46]

Für einige Spezialfälle können die Sitemaps, die auf die angegeben Weise automatisch gewonnen werden können, direkt auf die Webseite übernommen werden – in den meisten Fällen eignen sich die Ergebnisse aber eher als Vorlagen für manuell zu erstellende Sitemaps, vor allem, wenn Webangebote von verschiedenen Personen mit unterschiedlichen Werkzeugen gepflegt werden; hier kommen ohnehin nur Programme in Frage, die Hyperlinks als Grundlage ihrer Sitemap benutzen.

4.2.2.3 Kontextbasiertes Vorschlagsystem: Autonomy Kenjin

Der Softwarehersteller *Autonomy Software* hat sich auf Systeme zur automatischen Textkategorisierung und Indizierung spezialisiert. Im kommerziellen Umfeld dient die von Autonomy hergestellte Software zum verwalten großer Unternehmenswebsites, bei denen sich (noch) keine komplexes Dokumentenmanagement oder ein Redaktionssystem

[46] Screenshot aus einem Projekt in *Microsoft Visual InterDev*.

um die Verwaltung der Inter- und Intranetseiten kümmert. Sie bedient sich dabei einer Vielzahl von Mechanismen, von denen in dieser Arbeit bereits die Rede war. Das Kernprodukt, das diese Technologien zur Verfügung stellt, der sog. *Knowledge Server*, wird in lizenzierter Form auch von anderen Herstellern, u.a. Inxight Software oder Pixelpark Deutschland, angeboten und für eigene Produkte verwendet.

Der *Knowledge Server* indiziert automatisch vom Administrator des Systems vorgegebene Datenquellen, darunter nicht nur HTML-Dokumente sondern auch Dokumente in Textverarbeitungsformaten, Postscript-Dokumente, E-Mails, Datenbankverwaltete Dokumente und weitere mehr. Nach Methoden, die durch *Autonomy* aus verständlichen Gründen nicht offengelegt werden (schließlich handelt es sich um ein kommerzielles, geschlossenes System), werden diese Informations-quellen mit Hilfe von Textretrieval-Methoden automatisch indiziert, kategorisiert und mit Abstracts versehen. Dabei werden Dokumente auch nach Relevanz in ihrer Haupt- und in anderen Kategorien gewichtet und verlinkt[47]. Diese Verlinkung kann nicht nur innerhalb des *Knowledge Servers*, sondern auch tatsächlich, also innerhalb der indizierten Dokumente stattfinden, d.h. das Programm fügt beispielsweise in HTML-Dokumenten, die es durchsucht, automatisch Links auf verwandte Dokumente oder Kategorien innerhalb des *Knowledge Server* ein.

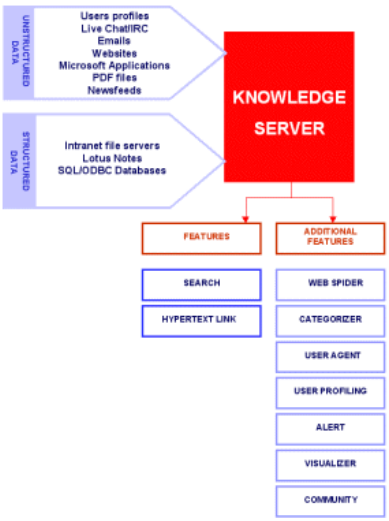

Abbildung 14: Architektur des *Autonomy Knowledge Server*.

[47] Zu Verfahren und Methoden vgl. Lenders, Wilfried; Willée, Gerd: „Linguistische Datenverarbeitung", Opladen/Wiesbaden 1998

In den auf *Knowledge Server* aufsetzenden Produkten werden aus diesen Daten automatisch Portalseiten für die Benutzung vor allem in Intranets erstellt. Bei der Recherche innerhalb dieser Daten stehen dem Benutzer fortschrittliche Methoden zur Verfügung. Neben dem Browsen der Kategorien und der Volltext- bzw. Topicsuche im Datenbestand werden auch Visualisierungstechniken eingesetzt. Diese basieren auf Topic-Maps, die primär hierarchische Strukturen und bei der Fokussierung auf einen bestimmten Teilbereich des Netzes auch Querverweise darstellen können.

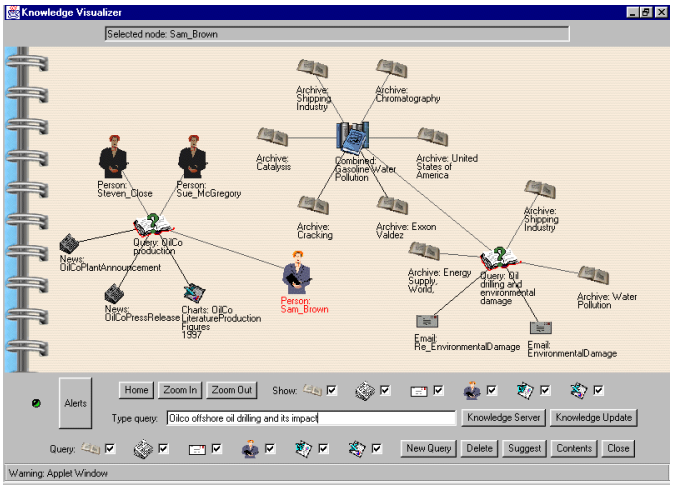

Abbildung 15: Visualisierung einer Suchanfrage mit *Autonomy Knowledge Server.*

Eine Fish-Eye Perspektive bietet die Software nicht an, dafür aber eine Form von Datenlandkarte, die mit sog. „Topic Islands" arbeitet. Bestimmte Themenbereiche werden auf einer dreidimensionalen Landkarte als Inseln dargestellt, die auch miteinander verbunden sein können. Dokumente, die auf den „Bergen" dieser Inseln liegen, sind für das Thema besonders relevant.

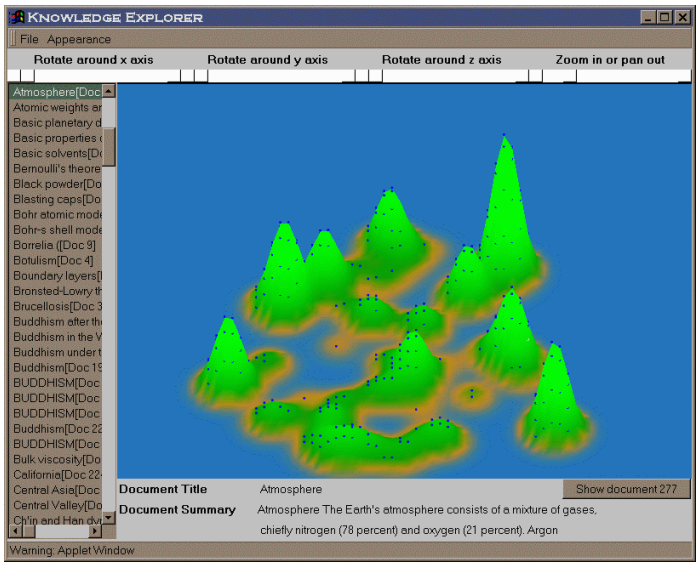

Abbildung 16: „Clustering" in *Autonomy Knowledge Server:* Dokumente werden anhand ihrer Relevanz für ein bestimmtes Thema auf sog. *Topic Islands* angeordnet.

Um die Übersichtlichkeit nicht zu stark einzuschränken, sind die Dokumententitel in einer solchen Datenlandkarte nicht zu sehen, sie erscheinen erst beim Berühren mit der Maus.

Die Vorteile des *Knowledge Server*-Ansatzes liegen auf der Hand: unabhängig vom Format können Dokumente automatisch indiziert und komfortabel gebrowsed werden. Der Nachteil ist ebenso offensichtlich: Das Produkt kann die Administratoren großer Webseiten durchaus dazu verleiten, keine stringente Struktur in die zu verwaltenden Daten zu bringen. Könnte sich ein System wie *Knowledge Server* auf einen verlässlichen Bestand von Metadaten verlassen, so könnte man auf die Retrievaltechniken, die sicher den größten Teil der Software ausmachen, in großen Teilen verzichten. Auch bietet die Software (noch) keine Möglichkeit, aus den beim Retrieval gesammelten Daten automatisch Metadaten für die indizierten Dokumente zu generieren – ein Feature, das aber sicherlich schon angedacht ist.

Um einen Eindruck von der Leistungsfähigkeit der „großen Lösung" zu bekommen, bietet die Firma *Autonomy* ein kostenloses Consumer-Product namens „Kenjin" an. Dieses Produkt indiziert Dateien, die der Benutzer lokal gespeichert hat und erstellt daraus einen Index, der sich durchsuchen lässt, verzichtet aber auf die aufwendigen Visualisierungstechniken der kommerziellen Programme. Außerdem

verfügt es über ein Modul, dass den Benutzer bei der Arbeit beobachtet und ihm im Kontext, in dem er Texteingaben macht, Sachverwandte Links ins WWW vorschlägt, die über eine bei *Autonomy* gepflegten Katalog von Webseiten (der wohl mit *Knowledge Server erstellt wird)* generiert werden. Die Vorschläge, die das Programm beispielsweise beim Schreiben dieser Arbeit machen konnte, waren erstaunlich brauchbar (siehe folgende Abb.).

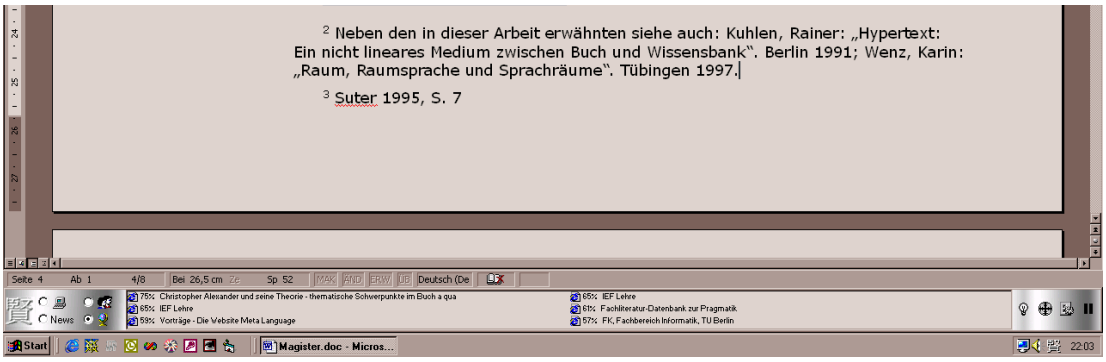

Abbildung 17: Autonomy Kenjin schlägt verwandte Webseiten vor, während in Word ein Dokument editiert wird.

4.2.2.4 Alexa

Alexa arbeitet nach einem ähnlichen Prinzip wie *Autonomy Kenjin:* Auf einem Internetserver bei *Alexa* wird eine Datenbank vorgehalten, die ein kategorisiertes Verzeichnis von Internetseiten beinhaltet. Wenn man sich mit einem Webbrowser, für den man sich ein Alexa-Plug-In[48] installiert hat, bestimmte Webseiten besucht, kann *Alexa* andere Webseiten vorschlagen, die sich mit einem ähnlichen Thema befassen. Große Teile des Verzeichnisses, auf das *Alexa* aufbaut, werden manuell verwaltet. Die einigen automatischen Algorithmen, die *Alexa* einsetzt, überprüfen die Relevanz von angebotenen Links danach, ob sie auch

[48] „Plug-Ins" sind Programmmodule, mit denen man die Funktionalität von Programmen (in diesem Fall Webbrowsern) erweitern kann. Dadurch, dass sie als Plug-In ausgeführt sind, integrieren sie sich in die Benutzeroberfläche des eigentlichen Programms. Populäre Plug-Ins für Webbrowser dienen meistens dazu, bestimmte proprietäre Dateiformate im Browser anzeigen zu können, so z.B. Macromedia Flash (Animationen) oder Adobe Acrobat (eine erweiterte Postscript-Implementierung).

angeklickt werden. Damit ist das Produkt genauso sinnvoll bzw. sinnlos wie manuell erstellte Suchkataloge wie z.B. Yahoo: Sie können mit der Informationsflut im Internet nicht mithalten, die wenigen Daten, die sie vorhalten, sind meist veraltet etc. Für deutschsprachige Seiten bot *Alexa* im Verlauf meiner Versuche zu dieser Arbeit in den meisten Fällen keine verwandten Links an, bei englischsprachigen Seiten zumeist nur Links auf sehr große und ohnehin populäre Webseiten. Das die *Alexa*-Technologie sowohl in der aktuellen Version des *Microsoft Internet Explorer* als auch des *Netscape Communicator* bereits eingebaut ist verwundert angesichts der stark eingeschränkten Möglichkeiten des Produktes etwas. Man kann aber davon ausgehen, dass die Daten über die verfolgten, vorgeschlagenen Links bei Alexa für die Erstellung von Nutzerprofilen genutzt werden – unter diesen Umständen verwundert es dann doch nicht.

Obwohl bei *Alexa* in der Datenbank offensichtlich Daten zur Relevanz von Webseiten für bestimmte Themen vorgehalten werden, bietet auch diese Lösung – genau wie *Kenjin* – keine Möglichkeiten der Visualisierung, verschenkt damit wertvolles Potential und soll hier auch nicht weiter betrachtet werden.

4.2.2.5 Hyperbolic Tree

Im *Palo Alto Research Center (PARC)*[49] der Firma Xerox, aus dem bereits viele wegweisende Entwicklungen der Computerindustrie gekommen sind, wurden seit 1988 mehrere Verfahren zur Visualisierung großer Datenbestände entwickelt. Die bereits beschriebenen *Cone Tree* und *Perspective Wall* stammen aus dieser Forschungsarbeit. 1996 gründete die Firma eine Tochter namens *Inxight*, die für die Weiterentwicklung der Verfahren zuständig ist. Das fortgeschrittenste Produkt, auf das sich der Fokus der Firma im Bereich der Visualisierung auch zunehmend legt, ist allerdings der *Hyperbolic Tree.*

Mit diesem Namen bezeichnet *Inxight* eine Form der Darstellung und Navigation großer, hierarchischer Datenbestände und Hypertext-

[49] *http://www.parc.xerox.com/parc-go.html*

systeme. Die Struktur der Darstellung entspricht einer zweidimensionalen Baumstruktur, die auf eine Kugel projiziert ist, um eine bessere Übersichtlichkeit zu erhalten. Durch diese Projektion auf eine Kugel lässt sich immer ein sehr großer Bereich des Netzes im Auge behalten und trotzdem ist die Lesbarkeit des fokussierten Bereiches nicht gefährdet.

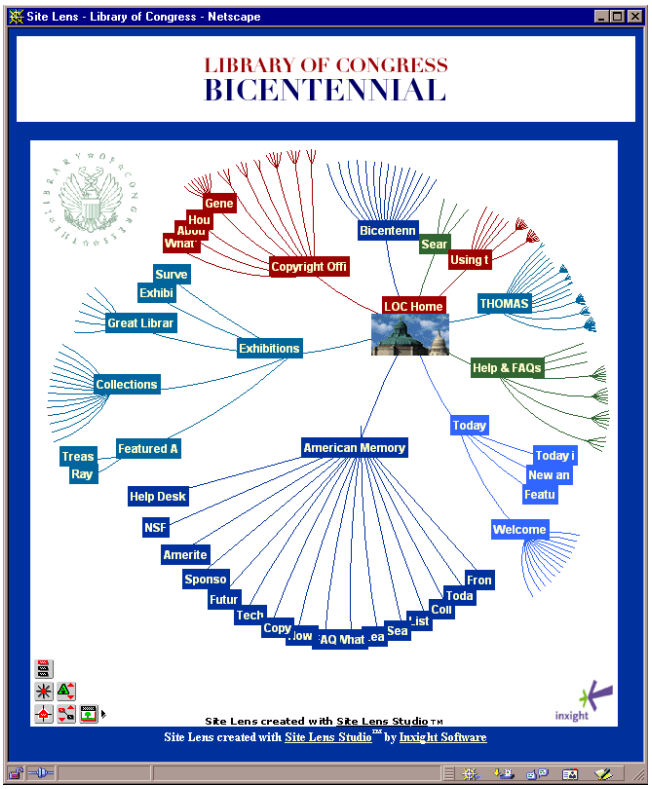

Abbildung 18: Mit *Inxight Site Lens* erstellte Sitemap der Library of Congress (*http://www.loc.gov/bicentennial/*).

Die Technologie des *Hyperbolic Tree* findet sich in Produkten der Firma *Inxight* wieder, wird aber auch im Paket mit anderen Visualisierungstools als *VizControls* lizensiert, so dass andere Firmen die Methoden in eigenen Programmen verwenden können. In Form der *VizControls* bietet *Inxight* tatsächlich nur die Werkzeuge zur Visualisierung und Navigation mit Hilfe eines Datenmodells, das nur für dieses Produkt gültig ist.

Ein Programm, das Daten für die *VizControls* automatisch generiert, ist z.B. *Site Lens* aus dem gleichen Haus. Dieses Programm nimmt ein bestehendes Webangebot und extrahiert aus den existierenden Hyperlinks eine Datenbasis für die Visualisierung. Der Benutzer erhält

quasi eine aktive Sitemap seiner Seite, in der er sich frei bewegen kann. Die Darstellung wird mit Hilfe eines Java-Applets bewerkstelligt, findet also auch im Browser des Anwenders statt. Die von *Site Lens* generierten Sitemaps bieten über die Navigation (verschieben des Fokus, anklicken der Knoten zum Aufrufen der Dokumente) noch weitere Möglichkeiten, so z.B. Zoomen, Anzeige der Knoten nach HTML-Titeln oder nach Dateinamen, farbliche Markierung von Dokumenten, die sich in letzter Zeit geändert haben etc.

Ein Problem der Darstellung des Navigationsraumes schränkt es allerdings sehr ein: Die Struktur, die der *Hyperbolic Tree* darzustellen vermag, ist immer streng hierarchisch, d.h. jeder Knoten bis auf den Ausgangsknoten hat immer nur einen übergeordneten und mehrere untergeordnete Knoten. Diese Einteilung wird vielen Hypertexten und auch einigen Datenbanken nicht gerecht. Querverweise auf Dokumente, die in der Hierarchie eines Hypertextes mehrere Ebenen über oder unter dem aktuellen Dokument liegen oder in einem völlig anderen Strang tragen zu den Möglichkeiten, die Hypertexte bieten, bei, werden aber von diesem System nicht berücksichtigt.

Die *HP Laboratorys* von *Hewlett Packard* beschäftigen sich mit einer Lösung genau dieses Problems. Basierend auf der *Hyperbolic Tree* Technologie von *Inxight* entwickelten Sie das Produkt weiter und führten das System der *Hidden Links* in die Technik ein.[50] Mit jedem Knoten wird in diesem System eine Liste mit Links geführt, die an andere Stellen der Hypertextbasis verweisen, aber keine unter- oder übergeordneten Knoten sind. Bei der Navigation funktioniert das System der *Hidden Links* zunächst wie ein *Hyperbolic Tree*. Wenn der Anwender jedoch zu einem Knoten kommt, der über eine Liste mit *Hidden Links* verfügt, so werden in der Darstellung diese Links in einer anderen grafischen Form dargestellt, wie in der ersten Abbildung zu erkennen. Vom Knoten „sendmail" gehen drei *Hidden Links* zu anderen Knoten des Hypertextes aus, die sich von anderen Knoten grafisch absetzen. Wenn der Benutzer nun einen dieser Links anklickt, gerät er

[50] vgl. Hao, Hsu, Dayal, Krug: „Visual Mining Large Web-based Hyperbolic Space Using Hidden Links", Palo Alto 1999; dies.: „Navigating Large Hierarchical Space Using Invisible Links", Palo Alto 2000; dies.: „Web-Based Visualization of Large Hierarchical Graphs Using Invisible Links in a Hyperbolic Space", Palo Alto 2000.

zu dem entsprechenden Knoten an einer anderen Stelle (in der Abbildung durch einen Pfeil gekennzeichnet.

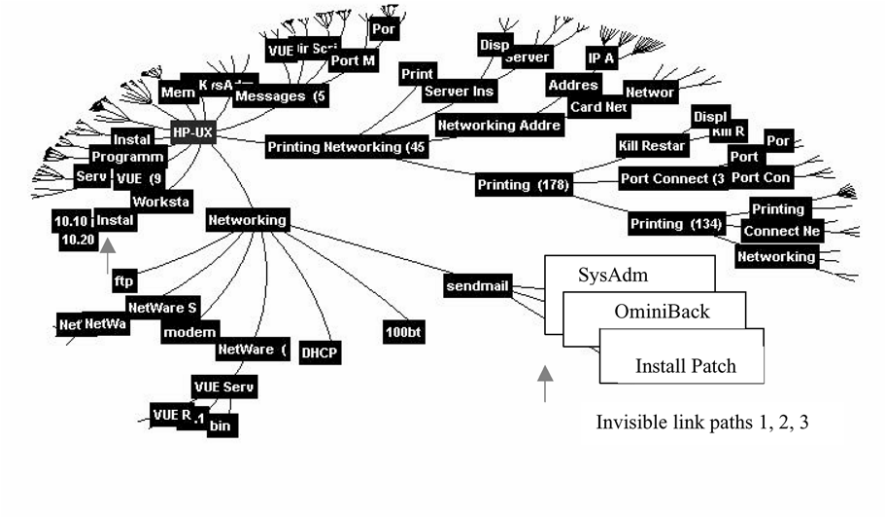

Abbildung 19: *Hidden Link*-Technologie in einer *Hyperbolic Tree*-Darstellung: Vom Knoten „sendmail" führen drei Links zu anderen Stellen im Hypertext, ohne dass direkte Linien die Übersichtlichkeit beeinflussen.

Von dieser Stelle kann er in der *Hyperbolic Tree* Darstellung beliebig weiter verzweigen (siehe zweite Abbildung). Sobald er sich allerdings von dem Knoten entfernt, den er über den *Hidden Link* angesprungen hat, erscheint in der Darstellung ein zusätzlicher *End*-Button, der ihn wieder an die Stelle des Links zurückführt. Damit ist ein zusätzliches Merkmal geschaffen, um „Lost in Hyperspace"-Situationen zu entschärfen – der Benutzer hat wenigstens die Möglichkeit, an einen wichtigen Punkt seines Weges durch das Netz zurückzukehren.

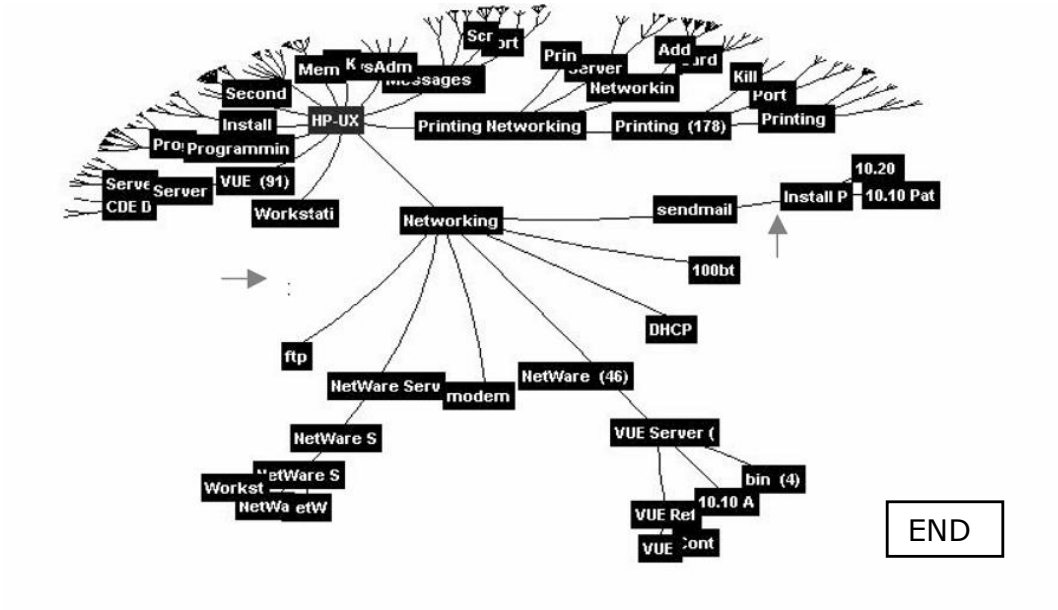

Abbildung 20: Wenn der Benutzer zu einen *Hidden Link* gesprungen ist, erscheint ein „END"-Button, der ihn jederzeit wieder zur Position des Links zurückführt.

Die *Hidden Link* Technologie stellt eine echte Bereicherung der *Hyperbolic Tree* Visualisierung dar und wird meines Erachtens in die Produkte der Firma *Inxight* Einzug halten, bügelt sie doch eine der größten Schwachstellen des Programms aus.

Mit oder ohne *Hidden Links*, auch bei *Hyperbolic Tree* müssen die Daten, die visualisiert werden, irgendwo herkommen. Die Struktur des Hypertextes legt das Programm in Dateien, die einem eigenen Standard folgen, ab. *Inxight* hat in seiner Produktpalette allerdings bereits die Werkzeuge, die sowohl aus bestehenden Hypertexten als auch aus eventuell vorhandenen Metadaten automatisch die zur Visualisierung benötigten Daten generieren. Produkte wie *SiteLens* und auch *AltaVista Discovery* zeigen, dass sich auch ohne Metadaten mit dem Programm brauchbare Datenlandkarten erstellen lassen.

4.2.2.6 AltaVista Discovery

Das einzige frei und kostenlos erhältliche System, das mehrere der hier vorgestellten Techniken einsetzt, ist *Discovery*[51] des Internet-Suchdienstes *AltaVista*. Das Programm wird clientseitig installiert, d.h.

es handelt sich nicht um einen Dienst, der auf einem Webserver läuft, und beinhaltet Komponenten für das Crawling, Highlighting (die Hervorhebung von gesuchten Wörtern in einem Text wie mit einem Textmarker), die Indexerstellung und Zusammenfassung von Webseiten, bietet Visualisierung von sog. „Search Spaces" mit Hilfe von *Hyperbolic Tree*-Darstellung, kann wie *Kenjin* oder *Alexa* Vorschläge für verwandte Webseiten machen und bietet darüber hinaus noch eine Schnittstelle zur Suchmaschine AltaVista mit all deren Möglichkeiten wie Rückwärtssuche etc..

Diese Vielfalt an Features setzt das Programm weit von bereits vorgestellten Lösungen ab und wurde nur durch den Zukauf bereits ausgereifter Technologien von Drittanbietern möglich. Nach einer kurzen Beschreibung der Arbeitsweise von *Discovery* werde ich kurz auf die einzelnen Module des Programms eingehen.

Abbildung 21: *Discovery* kombiniert Module von Drittanbietern zu einem integrierten Produkt.

Discovery wird als Programm gestartet und arbeitet dann – wie *Kenjin*, immer im Hintergrund. Wenn man den Webbrowser (oder ein anderes Programm) aktiviert, heftet sich ein *Discovery Bar* an die Oberseite des Programmfensters und ermöglicht den Zugriff auf die Optionen des Programms.

Abbildung 22: Der *Discovery Bar* heftet sich an alle Programmfenster.

[51] Informationen zum Programm und kostenloser Download unter *http://discovery.altavista.com/*

Discovery bietet die meisten Funktionen nicht nur für Inhalte des WWW, sondern auch für Dokumente, die der Benutzer auf seiner Festplatte lagert, an. Ich will mich auf die Funktionalität im Bezug auf Webinhalte beschränken.

Wenn man bei aktiviertem *Discovery* eine Webseite mit dem Browser aufsucht, kann man, indem man einen Button mit einer symbolischen Pinwand-Nadel klickt, die komplette Webseite (oder einen Teil bis zu einer bestimmten Tiefe) zu den *Search Spaces* hinzufügen. In diesem Moment werden zwei Module aktiv:

- Ein Crawler (eine abgespeckte Version des Crawlers, der auch bei der *AltaVista* Suchmaschine Verwendung findet) beginnt, die miteinander verlinkten Seiten auf der ausgewählten Domain zu besuchen.
- Die besuchten Webseiten werden jetzt mit Hilfe von Text Retrieval und Sprachanalysesystemen der Firma *Teragram*[52] analysiert – dies passiert auch in der *AltaVista* Suchmaschine; welche Module in *Discovery* tatsächlich zum Einsatz kommen, ist nicht dokumentiert, es wird aber – schon aus Performancegründen – nicht das komplette Set angewendet werden können. Die *Language Tools* von *Teragram* bieten u.a. folgende Funktionen: *„Normalisierung*" von Ausdrücken zur Kategorisierung, so werden z.B. *the third of February*, *February 3rd* und *2/3* in die selbe Kategorie „Datum" eingeordnet; *„Part of Speech tagging*" ordnet – englische – Wörter nach Möglichkeit grammatischen Kategorien zu, um Mehrdeutigkeiten zu vermeiden; *„Morphological Stemming*" führt gebeugte Wortformen in ihre ungebeugten Formen zurück; *„Derivational Stemming*" führt abgeleitete Wortformen auf ihre Ausgangsform zurück; *Text-Indexierung* u.a.m.
  Die Module, die von *Teragram* zur Verfügung gestellt werden, bieten komplexe linguistische Analysemethoden, die zur automatischen Indexerstellung von großem Vorteil sind. Mit Hilfe einiger dieser Module erstellt *Discovery* einen Index, der

[52] *http://www.teragram.com/w3/oem.htm*

dem in der Suchmaschine vorgehaltenen Index ähnelt – kombiniert mit einem Set von Metadaten. In diesem Index werden auch Teile des Volltextes, evtl. vorhandene Metadaten, Links zu anderen Seiten und Links von anderen Seiten gespeichert. Zusätzlich lassen sich Zeitintervalle bestimmen, in denen die Webseiten erneut vom Crawler besucht werden sollen.

Auf diesen Index bauen die anderen Module von *Discovery* auf. Innerhalb des lokalen Index lässt sich nun eine Volltextsuche durchführen, lassen sich Zusammenfassungen von Dokumenten generieren etc.. Wörter, die man innerhalb des Index sucht, werden mit Hilfe einer Highlighting-Software[53] im gefundenen Dokument hervorgehoben. Alle Ergebnisse, die *Discovery* liefert, werden im Internet-Browser als HTML-Seiten dargestellt. Das Programm benutzt dazu einen eigenen, lokalen Webserver (*Inside Out* der Firma *Inso*[54]), der darauf spezialisiert ist, Dokumente verschiedenster Formate als HTML-Dateien darzustellen.

All diese Möglichkeiten bieten dem Informationssuchenden bereits nützliche Ansatzpunkte, lassen sich aber mit Hilfe anderer Produkte – und wenn es eben nur die bekannten Webbasierten Suchmaschinen sind, auch erreichen. Das Feature, das *Discovery* für meine Betrachtungen besonders interessant macht, ist das *Hyperbolic Tree*-Modul der Firma *Inxight*, das zur Visualisierung der angelegten *Search Spaces* dienen kann.

[53] Highlighting-Software markiert Textstellen in elektronischen Texten so, als wären sie mit einem Textmarker angestrichen, ähnlich wie die entsprechende Funktion in *Microsoft Word.*

[54] *http://www.ie.inso.com/*

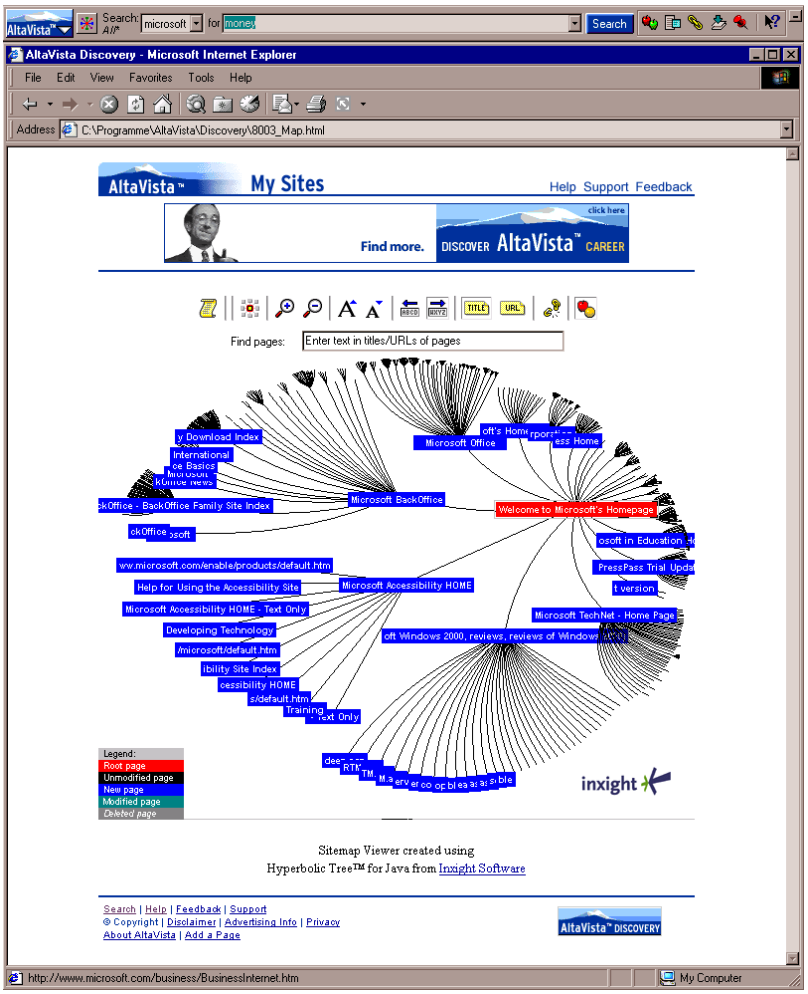

Abbildung 23: Eine *Hyperbolic Tree*-Darstellung der von *Discovery* durchsuchten *Microsoft.com*-Webseite.

Hier werden erstmals die Möglichkeiten der Visualisierung für eine breitere Öffentlichkeit. Anhand dieses Programms führte ich einen Stichprobenartigen Anwendertest durch, um die bereits unter *Hyperbolic* Tree beschriebenen Vor- und Nachteile in einer Alltagssituation zu verifizieren.

Bei den Betrachtungen viel *Discovery* von den angebotenen Programmen eindeutig aus dem Rahmen. Die Idee, fortschrittliche Module des Text-Retrieval und der Visualisierung in einem *relativ* einfach zu bedienenden Programm zu vereinen hat sowohl vor- als auch Nachteile. Vorteilhaft ist eindeutig, dass das Programm bei der Suche innerhalb lokaler Dokumente helfen kann und beim Surfen im WWW die Möglichkeiten der *AltaVista* Suchmaschine und Fetaures zur Textzusammenfassung immer in Reichweite hält. Nachteilig – und im Rahmen dieser Arbeit von höchster Bedeutung – ist, dass man in

Discovery klar die Grenzen solcher Systeme vor Augen geführt bekommt:

- Wenn Webangebote nicht im Hinblick auf Suchmaschinen und Visualisierungstools optimiert sind oder dynamisch generiert sind, scheitern einfachere Indexmechanismen oft;
- Ohne einen verzeichnisartigen Index lassen sich weder fortschrittliche Suchen noch Visualisierungstechniken realisieren;
- Das Erstellen von Indexen auch kleinerer Webseiten dauert auf durchschnittlich performanten Rechnern viel zu lange und verschlingt immense Mengen an Speicherplatz, um „schnell einmal" einen *Hyperbolic Tree* zur Hilfe bei der Suche heranziehen zu können – vor allem das eigentliche *crawling* der Seiten kann mitunter mehrere Stunden dauern;
- Die Techniken zur Visualisierung sind – wie bei *Hyperbolic Tree* bereits erwähnt – komplexen Seitenstrukturen nicht gewachsen und machen die Strukturen eher unübersichtlicher.

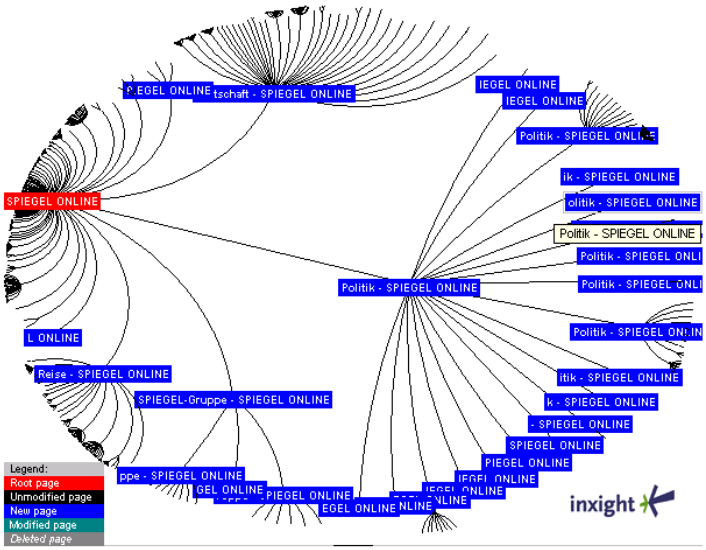

Abbildung 24: Ein von *Discovery* erstellter *Hyperbolic Tree* eines Teils der *spiegel.de*-Webseite – diese Darstellung verwirrt eher, als das sie bei der Orientierung weiterhelfen kann.

4.2.3 Aussicht

Von den aktuellen Systemen zur Visualisierung, die ich hier vorgestellt habe, basiert eine auf extensiver manueller Arbeit (manuelle Sitemaps), bieten zwei eigentlich gar keine Visualisierung (*Kenjin* und *Alexa*) und von den dreien, die automatische Visualisierung anbieten, kann eines nur Linkstrukturen abbilden und die beiden anderen basieren auf dem selben Produkt, der Visualisierungssoftware von *Inxight*. Es könnte der Eindruck entstehen, es gäbe nur eine vernünftige Software zur Darstellung von semantischen Strukturen in Hypertextsystemen.

Tatsächlich schien bei meinen Untersuchungen die Software von *Inxight* diejenige zu sein, die am meisten in funktionierenden Hypertextumgebungen eingesetzt wird. Die anderen Systeme, die ich vorgestellt habe, gehören aber trotzdem in diesen Kontext, und zwar aus folgenden Gründen:

Sitemaps, ob automatisch oder manuell generiert, sind die Visualisierungsmethode, auf die man im WWW am allermeisten trifft. Das liegt nicht nur daran, dass man keine spezielle Software braucht, um eine Sitemap manuell zu generieren und kein Plugin, um sie anzusehen - offensichtlich haben sich bei Sitemaps auch gewisse grafische Standards herausgebildet, in denen sich Anwender besonders gut zurechtfinden. Durchdacht gestaltete Sitemaps führen suchende Besucher von Webseiten in tausenden von Fällen sicher ans Ziel – für die Entwickler von Visualisierungssoftware ist dies allemal ein Zeichen, wohin die Reise gehen muss. Gerade die häufig eingesetzte Darstellungstechnik von *Inxights VizTools* lässt an Lesbarkeit stark zu wünschen übrig und ist zudem grafisch nicht besonders ansprechend. Solange solche Probleme nicht gelöst werden, wird es immer ein Anwendungsgebiet für manuell generierte Sitemaps geben.

Vorschlagsysteme wie *Kenjin* und *Alexa* setzen an der Stelle der automatischen Extrahierung von semantischen Inhalten an, der für die Visualisierung so immens wichtig ist.

"Visualizations of document connectivity based on existing, author-created links provide an important component in faciliating orientation

and navigation in the large WWW information space. However, for users it is typically documents' content relationships which are of most interest, rather than link based relationships. and such semantic relationships are only partly reflected by link structure."[55]

Um thematische Strukturen visualisieren zu können, müssen diese Strukturen erst einmal offen liegen. Metadaten werden in Zukunft wohl die entscheidende Rolle in diesem Zusammenhang spielen, aber solange diese Metadaten nicht vorliegen, müssen sie auf irgend eine Art und Weise gewonnen werden. Die Systeme, die diese Arbeit innerhalb von *Kenjin* oder auch *AltaVista Discovery* leisten sind Abbilder ihrer großen Brüder, die dies in Suchmaschinen und *Language Tools* in großem Maßstab machen.

Die *VizTools,* die Grundlage von *HyperbolicTree* und des Darstellungsteils von *AltaVista Discovery* sind und ihre Erweiterung durch die *Hidden Link*-Technologie von *HP* zeigen in diesem Rahmen am ehesten, wohin die Reise in der Darstellung zu gehen scheint. Für die Kombination von netzartigen und hierarchischen Strukturen, wie sie in Hypertexten vorherrschen, bietet sich eine Darstellungsform wie diese stark an. Durch die Verwaltung der Darstellung mittels Metadaten ist das System auch nicht auf die Darstellung von Linkstrukturen beschränkt, sondern kann, die richtigen Daten vorausgesetzt, auch semantische Bezüge gut darstellen.

4.3 Metadaten

4.3.1 Nicht fortgesetzte Formate

An Metadaten-Formaten, die heute nur noch geringe oder keine Bedeutung mehr haben, gibt es sehr viele. Neben Formaten, die eine weitere Verbreitung gefunden haben, aber aufgrund ihrer mangelnden Flexibilität nicht weiterentwickelt wurden, wie das bereits besprochene *MCF*-Format aus *Apple*s *Hotsauce* Technologie gibt es eine Reihe von Metadaten-Formaten, die sich nur in einzelnen Software-Projekten wiederfinden.

[55] Fowler, Richard H. u.a.: „Document Explorer Visualizations of WWW Document and Term Spaces". 1998 (online)

So verfügt jede im WWW verfügbare Suchmaschine und vor allem die Webverzeichnisse über einen Bestand an Metadaten, die in jeweils eigenständigen Formaten vorliegen – Teils in Form von Hypertexten, Teils in Datenbankstrukturen, die technisch völlig von der Hypertext-idee abgelöst sind. Die Technologien und Datenformate, die hinter den populären Internet-Suchdiensten stecken, sind wohlgehütete Geheimnisse. Ein Verzeichnis wie das *Open Directory Project,* das alle Mechanismen offen legt und zur Diskussion stellt und die Daten darüber hinaus in einem standardisierten, offenen Format ablegt, ist die Ausnahme, wenn nicht ein Einzelfall.

Die Betreiber von Suchmaschinen vertreiben die zur Suche eingesetzten Softwares allerdings oftmals auch an Firmen, die diese dann innerhalb ihrer Firmennetzwerke einsetzen. Von dieser Seite werden die Produzenten zunehmend gezwungen, auf offene und transparente Standards zu setzen. Wenigstens der Import von XML-Daten und RDF-Metadaten gehört inzwischen zum unverzichtbaren Fetaureset von Suchsoftware für den kommerziellen Einsatz.

4.3.2 META-Tags in HTML

Schon im HTML-Modell 3.2[56] gab es ein verbindliches Modell für Metadaten innerhalb von HTML-Dateien.[57] Diese Metadaten befinden sich im Kopf (<HEAD>)-Teil der HTML-Dateien in Form von *Meta-Tags*, die jeweils aus einem Name – Wert/Wertliste-Paar bestehen. So lassen sich für HTML-Dokumente, unabhängig vom eigentlichen Dokument-inhalt, Informationen über den Inhalt des Dokuments ablegen. Informationen über den Erstellungszeitpunkt und Autor eines HTML-Dokuments können dann zum Beispiel so aussen:

```
<HEAD>
<META NAME="Author" CONTENT="Kai W&ouml;rner">
<META NAME="Date" CONTENT="Tue, 14 Mar 2000 14:25:27 GMT">
</HEAD>
```

[56] Die HTML 3.2-Spezifikation wurde verabschiedet, nachdem man sich beim W3-Konsortium nicht auf eine Spezifikation 3.0 einigen konnte.

[57] *http://www.w3.org/TR/REC-html32.html#meta*

Darüber hinaus haben bestimmte *Meta-Tags* auch Funktionen innerhalb des *HTTP*[58]*,* die direkt vom Browser ausgewertet werden können, so zum Beispiel über das „Verfallsdatum" einer HTML-Seite.

Meta-Tags haben heute insbesondere deshalb Bedeutung, weil alle großen Suchmaschinen im Internet bevorzugt nach bestimmten Namen in diesen Tags der Webseiten suchen. Dies sind vor allem die *Description* und die *Keywords*-Namen, die eine Kurzbeschreibung der Seiteninhalte in einem Satz und eine Liste von Stichwörtern, die für die Seite exemplarisch sind, enthalten sollten. Eine präzise und sorgfältige Auswahl der hier eingesetzten Ausdrücke entscheidet oftmals darüber, ob eine Webseite bei der Suche nach einem bestimmten Stichwort weit vorne erscheint. Mit dem Thema, seine Webseite mittels Metatags in Suchmaschinen gut zu positionieren, beschäftigen sich eine riesige Zahl an Webseiten[59] – und damit wird ein grundsätzliches Problem der Tags deutlich.

Viele Besucher auf werbefinanzierten Webseiten bedeuten bares Geld für die Betreiber der Webseite – so ist ihnen oftmals jedes Mittel recht, Besucher auf die Seite zu locken, die dort möglicherweise gar nichts verloren haben. Metatags bieten hier eine völlig kostenlose Möglichkeit, die Besucherzahlen zu erhöhen – auf Kosten der Verlässlichkeit der Informationen. Das führt so weit, dass die Betreiber der populären Internet-Suchmaschine *Excite*[60] dazu übergegangen sind, Metatags bei der Suche mittels Crawlern nicht mehr zu berücksichtigen: „We believe our decision protects our users from unreliable information and ensures that Web publishers, if they choose, can have an active role in the representation of their content and services to the online consumer."[61]

Diese Entwicklungen haben die eigentlich positiven Möglichkeiten, die Metatags für HTML bieten, stark eingeschränkt. Immerhin nutzen alle Suchmaschinen im WWW die *Description* und *Keyword*-Metatags dazu, eine Beschreibung der Seite anzuzeigen: wenn ein *Description*-Metatag vorliegt, so nutzen die Suchmaschinen bei entsprechender

[58] Hypertext Transfer Protocol, das dem WWW zugrundeliegende Protokoll.

[59] Einen guten überblich verschafft *http://www.searchenginewatch.com/*

[60] *http://www.excite.com/*

Länge meist diesen, um eine Kurzbeschreibung der Webseite anzuzeigen. Auch die speziellen *Robots*-Metatags, die zur Steuerung von Suchmaschinen eingeführt wurden, werden beachtet. So bedeutet der Metatag

```
<meta name="ROBOTS" content="INDEX, NOFOLLOW">
```

beispielsweise, dass ein Crawler, der die Seite besucht, diese zwar in seinen Index aufnimmt (*INDEX*), Links auf dieser Seite aber nicht folgt (*NOFOLLOW).* So lassen sich Teile von Webangeboten vom Besuch durch Suchmaschinen ausschließen.

Metatags in HTML haben durchaus ihre Berechtigung und erweitern HTML, das eigentlich nur Auszeichnungen vorsieht, die der Gestaltung des Textes in einer bestimmten Umgebung dienen, zumindestens um eine rudimentäre Möglichkeit der Metaauszeichnung. Das W3C bemüht sich auch um eine Standardisierung der möglichen META-Namen, die Entwicklung im WWW geht aber eindeutig in Richtung XML und für die Metadaten zu RDF.

4.3.3 RDF (Resource Description Framework)

Eine Vielzahl der Probleme, die sich bei der Navigation im WWW auftun, das hat auch diese Arbeit bislang gezeigt, resultieren daraus, dass die riesige Menge an Informationen zwar auf Rechnern gelagert und also in maschinenlesbarer Form vorliegt, nicht aber maschinen*verständlich* ist, d.h. dass die Semantik der Daten und also ihre thematische Struktur, das Thema dieser Arbeit, nicht von Maschinen und automatischen Prozessen erfasst werden kann. Dies ist ganz konkret ein Problem von Metadaten, und dieses Problem wurde von vielen Seiten bereits angegangen. Die Entwicklungen in diesem Bereich laufen ganz eindeutig auf das *Resource Description Framework (RDF)* heraus, einer Initiative des *W3-Konsortiums.*

Zunächst einmal basiert RDF auf XML, d.h. RDF-Dateien werden in XML-Syntax formatiert. XML wiederum ist eine Anwendung von

[61] *http://www.excite.com/Info/listing8.html*

SGML[62]. XML wurde entwickelt, um die Flexibilität von SGML im Web zugänglich zu machen, ohne einen zu großen technischen Overhead zu produzieren und damit die Beschränkungen von HTML, vor allem die fehlende Auszeichnungsmöglichkeit von Strukturen innerhalb von Texten.[63] Wie auch SGML bietet XML keine vorgegebene Tag-Syntax, auch in XML werden DTDs zur Definition der Elemente, die in einem XML-Dokument vorkommen können, verwendet.

RDF basiert auf mehreren Technologien, unter anderem auf der *Hotsauce*-Technologie von *Apple*, von der in dieser Arbeit bereits die Rede war, aber auch auf der *Platform for Internet Content Selection (PICS)*[64]. Die Entwicklung von *PICS* wurde auch maßgeblich vom W3C beeinflusst und diente als System, Inhalte im WWW nach bestimmte Kriterien zu bewerten. Die häufigste Anwendung für *PICS* war das Rating bestimmter Webseiten unter Gesichtspunkten des Jugendschutzes. Angaben zu *PICS* fanden auch in den bereits beschriebenen Metatags in HTML ihren Platz; eine entsprechende Beschreibung einer Webseite unter Jugendschutzgesichtspunkten sieht z.B so aus:

```
<meta http-equiv="PICS-Label" content='(PICS-1.1
"http://www.rsac.org/ratingsv01.html" l gen true comment
"RSACi North America Server" for "http://www.delta-bike.de"
on "1998.07.19T17:26-0800" r (n 0 s 0 v 0 l 0))'>
```

Die eigentlichen Ratings befinden sich in Klammern hinter dem *r* und besagen, dass die Seite in allen Kategorien (n=nudity, s=sex, v=violence, l=language) eine Wertung von 0 (Null) hat. Dieses Ratingsystem fand eine so starke Verbreitung, weil es sowohl im *Netscape Navigator* als auch im *Internet Explorer* eingebaut war und so Eltern eine Möglichkeit gab, Inhalte aus dem WWW zu filtern, sofern die Autoren der Webseiten an diese Möglichkeit gedacht hatten. Auch *PICS* bot so bereits eine Form von Metadaten, die zudem noch einer einheitlichen Syntax unterlagen und so einfach auszuwerten waren.

---

[62] vgl. *http://www.arbortext.com/Think_Tank/SGML_Resources/Getting_Started_with_SGML/getting_started_with_sgml.html*

[63] vgl. *http://www.xml.com/pub/98/10/guide1.html*

[64] vgl. *http://www.w3.org/PICS/* und *http://www.icra.org/*

RDF soll die Rolle, die *PICS* bislang spielte, neben anderen, in Zukunft mit übernehmen. Über diese Funktion hinaus soll RDF aber noch folgende Möglichkeiten bieten:

- Thesauri und andere Webverzeichnisse beschreiben; RDF bietet *Schemata*, die genau solche Strukturen und andere bibliothekarische Verzeichnisse abbilden kann;
- Sitemaps – in RDF lassen sich Abhängigkeiten und hierarchische Strukturen von Dokumenten untereinander abbilden;
- Inhaltsbeschreibung von Webseiten – zu diesem Zweck wurde das „Dublin Core Schema"[65] für RDF angepasst, das standardisierte Beschreibungen zulässt und bereits in Metatags in HTML Verwendung fand;
- Beschreibungen darüber, wie Webseiten mit privaten Daten umgehen – auch hier soll eine einheitliche Syntax in Form eines Schemas gefunden werden, um dem Anwender transparenter zu machen, wie mit seinen Daten umgegangen wird;
- Beschreibungen der Darstellungsmöglichkeiten von Endgeräten wie z.B. Bildschirmauflösung, Farben, aktive Inhalte;
- Metadaten über Metadaten – auch Metadaten sind schließlich Daten, die ihrerseits wieder bestimmte Attribute haben können.

Die allem zugrunde liegende Leistung, die RDF erbringen soll, beschreibt das W3C so:

> *„The possible uses of the Web seem endless, but there the technology is missing a crucial piece. Missing is a part of the Web which contains information about information - labeling, cataloging and descriptive information structured in such a way that allows Web pages to be properly searched and processed in particular by*

[65] siehe *http://www.oclc.org/~emiller/dc/documents/wd-dc-schema.html*

computer. In other words, what is now very much needed on the Web is metadata."[66]

RDF beschreibt Ressourcen durch ihre Eigenschaften. Ein sehr einfaches Beispiel für eine RDF-Datei könnte beispielsweise so aussehen:

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:DC="http://purl.org/dc/elements/1.0/">
<Description about="http://www.w3.org/folio.html">
<DC:title>The W3C Folio 1999</DC:title>
<DC:creator>W3C Communications Team</DC:creator>
<DC:date>1999-03-10</DC:date>
<DC:subject>Web development, World Wide Web
Consortium, Interoperability of the Web</DC:subject>
</Description>
</RDF>
```

Diese RDF-Datei lehnt sich an das *Dublin Core Schema* zur Beschreibung von Webseiten an; *Schemas* dienen in RDF dazu, Charakteristiken der zu beschreibenden Eigenschaften zu definieren. Für jede Eigenschaft, die in der RDF-Datei beschrieben wird, kann in einem Schema festgelegt werden, was die Eigenschaft für eine Bedeutung hat, welche Werte sie annehmen kann etc.; sie ist sozusagen eine DTD für eine RDF-Datei. Diese Schemas werden in einer *Schema Definition Language* erzeugt, die wiederum dem XML Standard folgt.

Man kann hier eine Hierarchie von Metadaten erkennen: Die *Schema Definition Language* beschreibt das *Schema*, das *Schema* beschreibt die *Eigenschaft* und die *Eigenschaft* beschreibt die eigentliche Ressource.

Im Bezug auf die Probleme, die in dieser Arbeit aufgezeigt wurden, bietet das RDF einen klaren Weg aus der Misere. Das große Problem, mit dem sich alle Navigationshilfen für das WWW herumschlagen müssen ist die Tatsache, das die Inhalte im Web nicht maschinen-verständlich sind. Sollte sich RDF durchsetzen, wäre dieses Problem nahezu gelöst. Die Arbeit, sinnvolle RDF-Beschreibungen zu den eigentlichen Ressourcen schreiben zu müssen, würde den Autoren

[66] Swick, Ralf u.a.: W3C Activity Statement *http://www.w3.org/Metadata/Activity.html*

nicht abgenommen. Zunehmend erscheinen aber Werkzeuge, um Autoren diese Arbeit so einfach wie möglich zu machen.

Mittels RDF kann Software die Semantik von Daten erfassen und darauf basierend Entscheidungen treffen, Visualisierungshilfen geben u.a.m.. Benutzer können dann Entscheidungen darüber treffen, welche Informationen sie überhaupt sehen wollen – mittels RDF kann auch Benutzermodellierung eine viel größere Ausbreitung erfahren.

Ein Nachteil, den RDF aufweist, ist eher technischer Natur: Die XML-Syntax ist aufgrund ihrer Flexibilität relativ komplex und erzeugt große Dateien, die von den entsprechenden Softwareprodukten natürlich aufwendig geparsed werden müssen. Datenbankprodukte sind in solchen Szenarien wesentlich performanter. Doch auch hier zeichnen sich Lösungen ab: so stellte *Intel* stellte kürzlich eine Serie von Geräten vor, die XML-Transaktionen mit Hilfe von Hardware beschleunigen können.[67]

Der Erfolg von RDF steht und fällt mit der Unterstützung, die das Format in Industrie, Wirtschaft und bei den Anwendern findet, aber die Chancen stehen gut: XML ist auf dem besten Weg, sich zum de facto Standard im Web-Commerce-Bereich zu etablieren und große Spieler im Web-Geschäft, besonders *IBM* und *Microsoft* gehen der XML-Bewegung mit Werkzeugen und Ressourcen voran. Auch das *Open Directory Project*, das seinen gesamten Datenbestand für alle zugänglich in Form von RDF-Daten ablegt ist ein gutes Beispiel für die Akzeptanz, die das Format bereits jetzt besitzt.

4.3.4 Automatische Erstellung von Metadaten mittels Text Retrieval und Aussichten

So schön man sich das durch Metadaten kategorisierte WWW vorstellen mag – bis alle Informationen im Web mit Metadaten beschrieben sind, wird wohl noch einige Zeit ins Land gehen, und es ist ohnehin nicht damit zu rechnen, dass dieses Ziel komplett erreicht werden kann. Damit die Werkzeuge, die im Kontext von Metadaten,

[67] *http://www.intel.com/netstructure/products/xml_accelerators.htm*

XML und vor allem RDF jetzt und in Zukunft entwickelt werden, auch greifen können, wenn diese Metadaten nicht vorliegen, müssen Wege gefunden werden, Ressourcen nachträglich mit Metadaten zu versehen.

Die Tatsache, dass schon *PICS* und auch RDF nicht mehr darauf angewiesen sind, dass die Metadaten mit dem Dokument gespeichert werden (so wie es noch bei Metatags innerhalb von HTML der Fall war), erleichtert diese Aufgabe.

Einige der Aufgaben, die Text Retrieval-Werkzeuge mit Hypertexten bewältigen können, habe ich bereits bei der Vorstellung von *Autonomy Kenjin* und *AltaVista Discovery* vorgestellt. Die Ergebnisse, die Text Retrieval beim Parsen von Volltext liefern kann, hängen natürlich von den verwendeten Methoden, aber auch von den Texten ab, die von den Programmen geparsed werden. In den meisten Fällen wird eine automatische Indizierung vorgenommen, die nach der Häufigkeit bestimmter Wörter in einer Hypertextbasis sucht. Nachdem der „Noise" (und, der, aber etc. und Füllwörter) aus den Texten gefiltert wird, wird die Worthäufigkeit in den Dokumenten einer Hypertextbasis gesucht. Wenn ein Ausdruck beispielsweise in einem Set von Dokumenten besonders selten vorkommt, in einem einzelnen Dokument des Sets aber besonders häufig, so wird davon ausgegangen, dass dieses Wort besonders relevant für den Index ist. Mit Normalisierungs- und Stemmingmethoden werden dann diese Wörter für den Index „getrimmt", so dass sie auch gefunden werden, wenn nach anderen Formen des Wortes gesucht wird und die Wörter werden in einen Index übernommen. Suchmaschinen nutzen meist ein solches oder ähnliche Verfahren.

An dieser Stelle böte sich jetzt die Möglichkeit, Metadaten für die untersuchten Dokumente zu erstellen, und einige Suchmaschinen machen auch genau das: wenn keine Metadaten zur Beschreibung einer Webseite vorliegen, dann legen sie aus den beim Parsen der Seiten gewonnen Daten eigene Metadaten an und legen sie in der eigenen Datenbank ab.

Diese Methode könnte sicherlich helfen, Metadaten für Ressourcen zu generieren, bei denen nicht abzusehen ist, dass in naher Zukunft

auf anderem Wege Metadaten erzeugt werden. Diese Methoden können aber, wegen der fehlenden Maschinenverständlichkeit der Ressourcen, bestimmte Aspekte, die sich in RDF abbilden lassen, nicht oder nur unzureichend erfassen. So sind beispielsweise die Beziehungen von Dokumenten untereinander mit Hilfe des Text Retrieval im Grunde nicht herauszubekommen; der einzige Ausweg, die Linkstrukturen als Grundlage dieser Verknüpfungen zu benutzen, ist keiner – dies wurde an anderer Stelle bereits gezeigt.

Diese Ressourcen werden durch die automatische Indizierung aber immer noch leichter auffindbar, weshalb sich der Aufwand durchaus lohnen könnte. Alle Möglichkeiten, die ein komplettes Set von Metadaten für eine Ressource zur Verfügung stellen kann, wird man auch bei manuell angefertigten RDF-Dateien in Zukunft wohl nicht unbedingt erwarten können.

4.4 Visualiserung in der Praxis: Ein kleiner Anwendertest

4.4.1 Versuchsbeschreibung

Um herauszubekommen, was Visualisierungstechniken für Hypertexte in der Praxis tatsächlich bringen, führte ich einen Anwendertest in einem kleine Rahmen (10 Teilnehmer) durch.

Die Aufgabe bestand jeweils darin, von einer bestimmten Ressource im Internet eine bestimmte andere Ressource aufzufinden, und zwar einmal mit Hilfe der *HyperbolicTree* Darstellung aus *AltaVista Discovery* und einmal ohne, d.h. nur mit Hilfe eines Internetbrowsers.

Sechs der Versuchsteilnehmer stuften sich im Umgang mit dem WWW als „erfahren", vier als „ wenig erfahren" ein (Skala: keine Erfahrung; wenig Erfahrung; erfahren; Profi), alle hatten schon einmal Webseiten im Internet besucht und konnten mit dem Begriff „browsen" bzw. „surfen" (im Zusammenhang mit dem WWW) etwas anfangen. Jeweils drei „erfahrene" und zwei „wenig erfahrene" Personen wurden in einer Gruppe zusammengefasst und erhielten ihre Aufgaben.

Die Teilnehmer der ersten Gruppe sollten zunächst mit Hilfe von *Discovery* auf der deutschen *Microsoft*-Webseite nach der Informationsseite von *Microsoft Word* suchen. Dazu wurde mit *Discovery* ein *Search Space* der *Microsoft*-Webseite angelegt und visualisiert. Der Ausgangsbildschirm, den die Teilnehmer zu sehen bekamen, sah folgendermaßen aus:

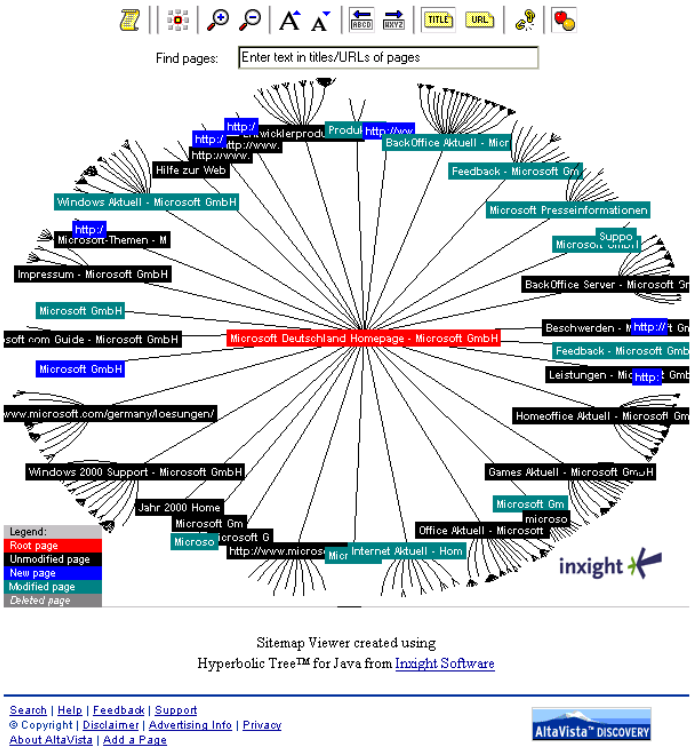

Abbildung 25: Die Startsituation für die Teilnehmer der Gruppe, die die *Microsoft Word*-Seite mit *AltaVista Discovery* finden sollten.

Es wurde ein Zeitlimit von 5 Minuten festgelegt, um den Versuch nicht ins Endlose auszudehnen; nach dieser Zeit hätte jeder Websurfer längst aufgegeben. Die Teilnehmer bekamen vorher anhand einer anderen Visualisierung die Navigation innerhalb des *Hyperbolic Tree* gezeigt.

Anschließend bekam die erste Gruppe die Aufgabe, auf der Seite des *Spiegel*[68] eine bestimmte, aktuelle Nachricht zu finden („*Metallica, diese reichen Affen*" über eine Animation im WWW, die die Rock-Gruppe Metallica parodiert), diesmal einfach mit Hilfe des Web-browsers.

Die zweite Versuchsgruppe hatte das umgekehrte Szenario zu bewältigen: Sie sollten innerhalb von 5 Minuten in einer *Hyperbolic Tree*-Darstellung der *Spiegel*-Webseite die entsprechende Nachricht finden und anschließend nur mit dem Webbrowser auf der deutschen *Microsoft*-Seite das Dokument über *Microsoft Word* finden. In allen

[68] *http://www.spiegel.de/*

Fällen wurde die Zeit gemessen, die von den Probanden für die Bewältigung der Aufgabe benötigt wurde.

Im optimalen Fall war der Artikel auf der *Spiegel*-Webseite *(http://www.spiegel.de/netzwelt/netzkultur/nf/0,1518,76464,00.html)* mit 2 Klicks zu erreichen (zuerst auf *Netzwelt*, dann auf die Überschrift des Artikels), und zwar sowohl im Browser als auch in *Discovery*.

Auch die Webseite von *Microsoft Word (http://www.microsoft.com/germany/office/word/)* lässt sich mit zwei Klicks erreichen, gleichfalls mit beiden Navigationsmitteln. Beachtenswert ist an dieser Stelle, dass die URL der *Spiegel*-Seite keine Aussage über den zu findenden Inhalt macht, während man, wenn man mit die Syntax der *Microsoft*-URLs vertraut ist, bestimmte Microsoft-Seiten auch durch einfache Eingabe in das Adressfeld des Browsers finden kann; besonders die englische Version der *Microsoft*-Seite kann hier als vorbildlich gelten – hier kommt man mittels Eingabe der URL *www.microsoft.com/,* gefolgt vom Produktnamen, fast immer ans Ziel.

Sowohl *Discovery* als auch die HTML-Dokumente von *Spiegel*-Online und *Microsoft* bieten eine Suchfunktion an. Zur Benutzung dieses Hilfsmittels wurde vor dem Versuch keine Aussage gemacht, um herauszufinden, ob diese Hilfsmittel selbstständig genutzt werden.

4.4.2 Ergebnis

Die folgende Tabelle zeigt das Ergebnis des Versuchs: Jeweils ein Teilnehmer aus der ersten und zweiten Gruppe sind in einer Spalte zusammengefasst, da die Teilnehmer einer Gruppe jeweils nur zwei Aufgaben zu lösen hatten.

	Teilnehmer 1/6	Teilnehmer 2/7	Teilnehmer 3/8	Teilnehmer 4/9	Teilnehmer 5/10
Spiegel Online Artikel über Webseite	2,20	∞	1,20	2,10	4,20
Spiegel Online Artikel über *Discovery*	∞	∞	∞ (mit Suche)	∞	∞
Microsoft Word über Webseite	1,10	2,00	∞ (mit Suche)	3,20	3,45
Microsoft Word über *Discovery*	4,10	∞	3,20	∞	3,35

In den Zellen, in denen ein ∞-Zeichen zu finden ist, wurde der Versuch nach 5 Minuten abgebrochen.

Ein Teilnehmer benutzte in beiden Fällen die angebotenen Suchfunktionen und scheiterte auf der Microsoft-Webseite daran, dass er gleich den ersten angebotenen Link anklickte und damit nicht auf die gewünschte Seite gelangte. Statt andere Suchergebnisse auszuprobieren, versuchte er es wieder von der Startseite aus, worauf die Zeit nicht mehr ausreichte.

Die Suche konnte ihm auch in *Discovery* nicht weiterhelfen. Gesucht werden kann in *Discovery* nur nach Text innerhalb des Seitentitels und der URL – wie ich bereits feststellte, enthalten die URLs beim Spiegel keinen Hinweis auf das Dokument, das sie darstellen, und auch die Seitentitel lauten beim *Spiegel* allesamt nur „*Spiegel Online*"*.*

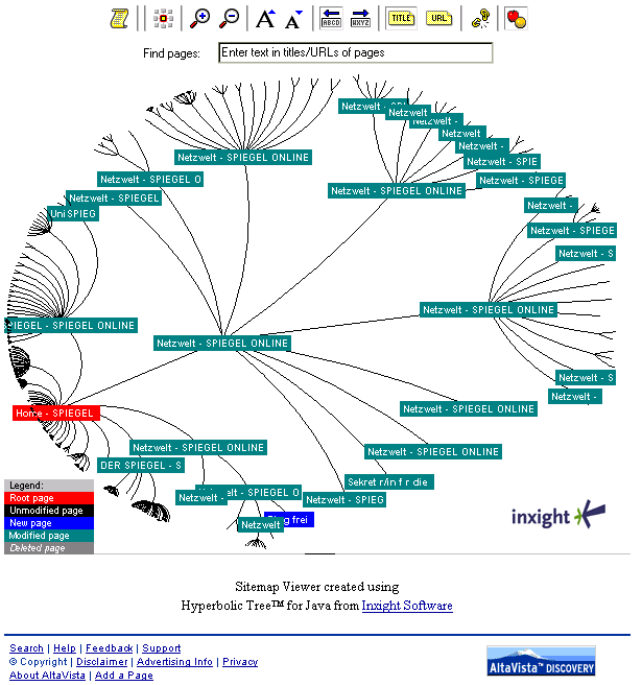

Abbildung 26: Mit diesen Seitenbenennungen hilft auch die Visualisierung nicht weiter: alle Seiten haben den gleichen Namen. Auch die Anzeige der URLs würde in diesem Fall nicht weiterhelfen.

4.4.3 Schlussfolgerung

Um allgemeingültige Schlüsse ziehen zu können war die erhobene Datenbasis sicherlich nicht groß genug. Dennoch fallen einige Punkte ins Auge:

- Die *Visualisierung* mittels Discovery brachte auf keinen Fall eine Beschleunigung des Suchvorganges. Vor allem aber gab es im Fall der *Spiegel-Online*-Seite aufgrund der nichtssagenden Seiten- und URL-Titel zu viele „Totalausfälle".
- Das Auffinden der Seiten über die Navigationsangebote der Webseite funktionierte in den meisten Fällen.
- Die Benutzung der Suchoption brachte in diesem Versuch keine Verbesserung, daraus lässt sich aber keine Regel ableiten.

Besonderer Wert muss darauf gelegt werden, dass der durchgeführte Versuch keinen Schluss über die Brauchbarkeit der *Hyperbolic Tree* Darstellung oder einer anderen Visualisierungstechnik zulässt. Weder die *Microsoft* noch die *Spiegel* Webseite waren auf eine

Darstellung durch diese Technik besonders vorbereitet, die Daten, die der Visualisierung zugrunde lagen, wurden aus der Linkstruktur von *Discovery* selbstständig extrahiert. Mit speziell für diesen Zweck aufbereiteten Metadaten, mit der Absicht einer Visualisierung mit diesem Produkt im Hinterkopf, ließen sich sicherlich glücklichere Darstellungen erreichen. Ein Beispiel für eine solche Darstellung findet sich u.a. unter *http://www.xerox.com/go/xrx/search/sitemap.jsp.*

Als in jeder Hinsicht unglücklich muss allerdings die Benennung der Seiten und der URLs bei *Spiegel Online* gewertet werden. Zwar werden die Seiten dieses Online-Angebotes offensichtlich aus einer Datenbank generiert, trotzdem ließen sich mit minimalem Aufwand wenigstens aussagekräftige Seitentitel generieren – dies wäre auch für die Verwaltung von Bookmarks im Browser wünschenswert. Für den Umstand, dass jede Seite einfach „*Spiegel Online*" heißt, gibt es jedenfalls keine Entschuldigung.

Obwohl die Teilnehmer mit grundlegenden Navigationskonzepten im Browser vertraut waren, fanden immerhin zwei von ihnen auf diesem Wege die gewünschte Webseite nicht – und das innerhalb einer Zeitspanne, die wohl die wenigsten Webnutzer überhaupt aufbringen würden. Die Navigationshilfen der beiden Webangebote sind also scheinbar noch optimierungsbedürftig. Unverständlich ist auch, weshalb bei der Suche nach „*word*" auf der Microsoft-Seite die zentrale *Microsoft Word*-Seite erst als dritter Treffer erscheint – auch dies führte dazu, dass einer der Teilnehmer die Seite nicht in der angegebenen Zeit finden konnte. Wäre er bei der anderen Gruppe gewesen, hätte er vermutlich mehr Glück gehabt: Bei einer Suche nach dem Begriff *„metallica*" bei *Spiegel Online* wäre gleich der erste Treffer der richtige gewesen – so hätte man die Seite in kürzester Zeit auffinden können.

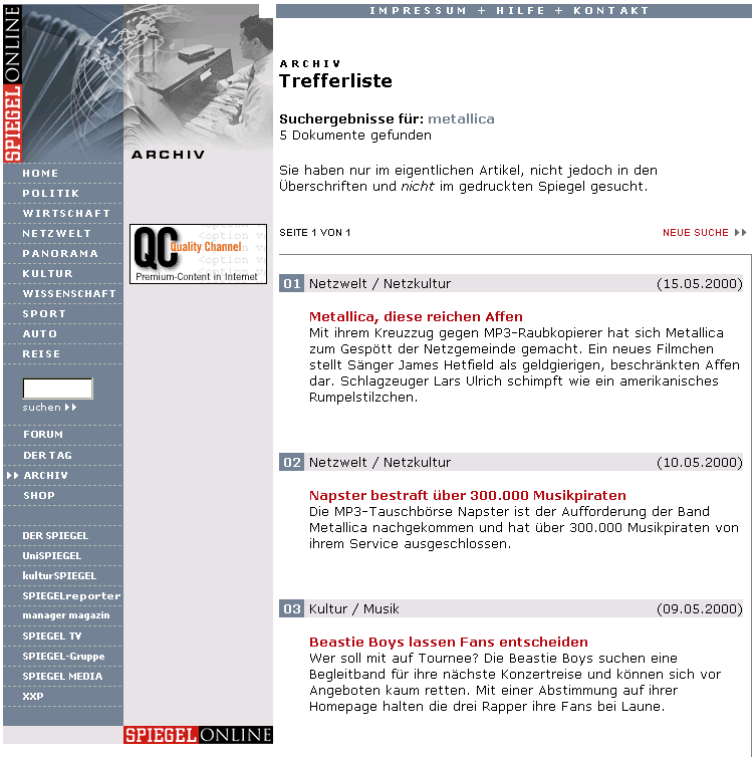

Abbildung 27: Die gesuchte Seite steht bei *Spiegel Online* als erste in der Liste der Suchergebnisse, wenn man nach *„metallica*" sucht.

5 Zusammenfassung und Fazit

Kohärenz ist keine Eigenschaft, die Texten innewohnt. Kohärenz entsteht beim Lesen von Texten – oder sie entsteht nicht. Dies gilt für alle Texte, für lineare, gedruckte Texte, für Hypertexte und für alle Formen, die dazwischen existieren. Die Unterschiede entstehen, wenn Autoren daran gehen, die Kohärenzbildung beim Leser zu steuern bzw. sicherstellen zu wollen. Das Thema und somit die thematische Struktur ist es, was einen Text in den meisten Fällen zusammenhält, und deshalb spielt diese thematische Struktur eine zentrale Rolle wenn es darum geht, Kohärenz in Hypertexten zu sichern.

Größere Hypertextnetze und mit ihnen das größte, das WWW, werden nicht durch ein bestimmtes Thema zusammengehalten. Es teilt sich in unendlich viele thematische Inseln auf, die sich mit bestimmten Themen beschäftigen, wobei die verschiedenen Hypertexte oft nicht einmal voneinander wissen. Wer Informationen in diesem größten elektronischen Wissensspeicher finden will, wird von der Fülle der Informationen fast erschlagen und muss schnell feststellen, dass es keine wirklich brauchbaren Werkzeuge gibt, ihm die ganze Informationsfülle des WWW für seine spezielle Anforderung zu erschließen.

In dieser Arbeit habe ich versucht herauszuarbeiten, dass einer der hervorragenden Gründe für diesen Zustand das Fehlen von brauchbaren Metadaten ist, die in der Lage sind, die Inhalte im WWW zu beschreiben, und zwar in einer Form, die sie für Maschinen (also für Software) verstehbar macht. Mit Hilfe dieser Metadaten ließen sich thematische Strukturen und Beziehungen zwischen Hypertexten, die nicht miteinander verlinkt sind, aufzeigen und navigierbar machen. Hier könnten Programme zur Visualisierung solcher Strukturen ansetzen.

Die Systeme zur Visualisierung, die ich in dieser Arbeit vorgestellt habe, krankten zum großen Teil daran, dass ihnen eben diese Metadaten fehlen. Sofern sie sich auf solche Daten stützen, müssen diese oft manuell erstellt werden oder von Software aus Volltexten extrahiert werden.

Thematische Strukturen und die Semantik von Hypertexten zu extrahieren ist aber keine triviale Aufgabe für Software – wenn es möglich wäre, alle Aspekte per Text Retrieval aus den Volltexten zu extrahieren, bestünde kein Bedarf an Metadaten.

Mit dem *Resource Description Framework* RDF hat in den letzten Jahren eine Technologie die Bühne der Metadaten betreten, die das Zeug hat, die technischen Beschränkungen zu überwinden. Es ist hinreichend flexibel, es basiert auf existierenden und akzeptierten Standards und es zeichnet sich eine breite Akzeptanz der Industrie ab. Auch die Werkzeuge, welche die Erstellung von RDF-Daten für Autoren von Hypertexten und anderen Ressourcen nehmen langsam Gestalt an. Mit diesem wichtigen Schritt ergeben sich auch neue Möglichkeiten zur Navigation und zur Informationssuche im WWW.

Die Suche nach Informationen *kann* relevante Daten liefern wenn mit Hilfe von Benutzermodellierung und relevanten Metadaten Ressourcen gleich ausgeschlossen werden können, die für den Suchenden nicht nützlich sind. [69] Das anfangs beschriebene Szenario mit der Suche nach „Kohärenz" ließe sich so einfach auf den gewünschten Kohärenzbegriff einschränken.

Visualisierung *kann* funktionieren und sie *kann* signifikanten Zeitgewinn bei der Suche nach Informationen bringen, wenn sie auf brauchbaren Daten beruht. Eine Kombination aus Internetsuche und Visualisierung kann mit einer *guten* Basis an Metadaten völlig neue Recherchetools und Softwareagenten ermöglichen. Die mittelfristige Ablösung des eingeschränkten HTML-Standards durch XML ermöglicht darüber hinaus eine ganz andere Verwertbarkeit der damit gefundenen Dokumente.

Von der technischen Seite ist der erste Schritt in die richtige Richtung gemacht. Jetzt liegt es an den Autoren, die neuen Möglichkeiten auch zu nutzen. Für RDF müssen Schemas geschrieben werden, die verschiedenen Anforderungen genügen. Das *Dublin Core* Schema hat sich als Beschreibung für Web- und Bibliotheksressourcen bewährt, aber es wollen auch völlig andere Inhalte mit Metadaten versehen werden. In Zukunft werden sicher die gängigen Werkzeuge

[69] siehe auch Feldkamp 1996, S. 118ff.

die Möglichkeiten bieten, Metadaten für Dokumente zu erzeugen (auch die Kommentare, die man beispielsweise mit Microsoft Word-Dokumenten ablegen kann, sind ja bereits Metadaten) und in standardisierten Formaten zu speichern. Bis dahin sollten Autoren, die ihre Publikationen im WWW veröffentlichen wollen, sollten sich rechtzeitig daran machen, Metadaten zu ihren Publikationen zu erstellen, wenn sie von diesen Entwicklungen profitieren wollen.

6 Literatur

- Appelrath, H.J.: „Ein systematisches Verzeichnis des deutschen WWW: GERHARD". Uni Oldenburg 1997 (Online). *http://www.offis.uni-oldenburg.de/jahresbericht/jb97/p9_3.htm*
- Berners-Lee, Tim: „Metadata Architecture". World Wide Web Consortium Personal View, 1997. *http://www.w3.org/DesignIssues/Metadata*
- Berners-Lee, Tim u.a.: „World-Wide Web: The Information Universe" , Electronic Networking: Research, Applications and Policy, Vol 1 (1992). Online Version von: *http://sunsite.org.uk/media/literary/collections/Online-Book-Initiative/Networking/WWW/ENRAP_9202.ps*
- Berners-Lee, Tim: „Semantic Web Road Map". World Wide Web Consortium Draft, 1998. *http://www.w3.org/DesignIssues/Semantic*
- Böhme, Rainer: „Resource Description Framework". Leipzig 1999. *http://www.informatik.uni-leipzig.de/db/seminararbeiten/semSS99/arbeit5/Rdf.html*
- Kalina Bontcheva, Yorick Wilks: „Combining Language Generation and Belief Modelling into a Flexible Hypertext System". Sheffield 1997. *http://www.dcs.shef.ac.uk/~kalina/papers/flexht/flexht1.html*
- Bublitz, Wolfram;Lenk, Uta; Ventola, Eija(Hg.): „Coherence in spoken and written discourse", Amsterdam 1999
- Bünte, Oliver: „XML auf dem Vormarsch" in c't Magazin für Computertechnik 10/2000, Hannover 2000.
- Czap, Hans; Ohly, H. Peter; Pribbenow, Simone (Hg.): „Herausforderungen an die Wissensorganisation: Visualisierung, multimediale Dokumente, Internetstrukturen". Würzburg 1998.
- Feldkamp, Jürgen: „Kontextermittlung und –berücksichtigung in Hypertextinformationssystemen". Hamburg 1996.
- Ferber, Dr. R.: „Data Mining und Information Retrieval". Skript zur Vorlesung am FB Informatik der TU Darmstadt WS

1999/2000.
http:// http://www.darmstadt.gmd.de/~ferber/dm-ir/

- Fowler, Richard H.; Fowler, Wendy A.L.; Williams, Jorge L.: „Document Explorer Visualizations of WWW Document and Term Spaces". University of Texas - Pan American Edinburg 1998.
  http://www.cs.panam.edu/info_vis/de_rep.html
- Fritz, Gerd: „Kohärenz. Grundfragen der linguistischen Kommunikationsanalyse". Tübingen 1982.
- Fritz, Gerd; Hundsnurscher, Franz: „Handbuch der Dialoganalyse". Tübingen 1994.
- Gloor, Peter: „Elements of Hypermedia Design : Techniques for Navigation & Visualization in Cyberspace", Boston 1996 (als Online-Dokument unter *http://www.birkauser.com/hypermedia*[70])
- Golovchinsky, Gene: „Reaction to SIGIR 99 Panel on User Interface Issues". FX Palo Alto Laboratory, Palo Alto 2000.
- Hao, Ming; Hsu, Meichun; Dayal, Umesh; Krug, Adrian: „Visual Mining Large Web-based Hyperbolic Space Using Hidden Links". HP Laboratories Technical Report HPL-1999-20, Palo Alto 1999
- Ianella, Renato: „An Idiot's Guide to the Resource Description Framework", Queensland, AUS, 1999.
  http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/
- Landow, George: „Hypertext 2.0 – The Convergence of Contemporary Critical Theory and Technologiy". Baltimore 1997.
- Lassila, Ora; Swick, Ralph R.:"Resource Description Framework (RDF) Model and Syntax". World Wide Web Consortium Recommendation, 1999.
  http://www.w3.org/TR/REC-rdf-syntax/
- Lenders, Winfried; Willée, Gerd: „Linguistische Datenverarbeitung – Ein Lehrbuch", 2. Auflage, Opladen/Wiesbaden 1998

[70] Wegen der „Flüchtigkeit" WWW-basierter Texte habe ich die wichtigsten WWW-basierten Quellen und diejenigen, aus denen ich zitiert habe, als Ausdrucke in einem gesonderten Materialienordner vorliegen.

- Lobin, Henning (Hg.): „Text im digitalen Medium". Opladen/Wiesbaden 1999
- McKnight, Cliff; Dillon, Andrew; Richardson, John: „Hypertext in Context". Cambridge 1991.
- Pott, Oliver; Wielage, Gunter: „XML Praxis und Referenz", München 2000.
- Suter, Bettina Ansel: „Hyperlinguistics - Hypertext-Lernumgebungen im akademischen Kontext: Eine Fallstudie". Zentralstelle der Studentenschaft, Zürich 1995
- Swick, Ralf u.a.: „W3C Activity Statement on RDF" *http://www.w3.org/Metadata/Activity.html*
- Vogt, Petra: „Datenlandkarten" in c't Magazin für Computertechnik 5/1988. Hannover (Heise) 1998.
- Zwisler, Rainer: „Navigation in Hypertextsystemen" Regensburg 1988 (online) *http://www.zwisler.de/scripts/hyper_nav/hyper_nav.html*

URLs, die bei der Erstellung der Arbeit hilfreich waren:

- *http://msdn.microsoft.com/xml/default.asp*
  Microsoft Developer Network XML-Startseite
- *http://www.w3.org/TR*
  W3-Konsortium Technische Spezifikationen
- *http://www.w3.org/Metadata*
  W3-Konsortium Metadaten-Seite
- *http://www.acm.org*
  Association for Computing Machinery, vor allem die Special Interest Groups: *http://www.acm.org/sigweb/* für Hypertext, Hypermedia und WWW
- *http://www.gerhard.de/*
  GERman Harvest Automated Retrieval and Directory
  Verzeichnis wissenschaftlich relevanter, deutscher WWW-Seiten, geordnet nach einer Variante der universellen Dezimalklassifikation (UDK).

- *http://www.eastgate.com/HypertextNow/*
  Hypertext Now, Hinweise und Tipps für Hypertextautoren und Links auf brauchbare Resourcen von *Eastgate Systems*, den Herstellern des Bookmark-Visualisierungstools *Web Squirrel* und der Hypertext-Autorensoftware *Storyspace*.
- *http://www.public.iastate.edu/~CYBERSTACKS/BigPic.htm*
  The Big Picture, Überischt über Visualisierungstools für das WWW und Datenbanken (wird offenbar nicht mehr aktualisiert).
- http://metadata.net.dstc/
  Am *Distributed Systems Technology Centre* wird ein Metadaten-Editor angeboten, der verschiedenste Metdatan-Formate, u.a. auch RDF, erstellen kann.
- *http://multimedia.pnl.gov:2080/infoviz/*
  Das *Pacific Northwest National Laboratory* untersteht dem US-Energieministerium und entwickelt in eigener Regie Systeme zur Visualisierung und zum Information Retrieval (Projekt *SPIRE)*. Ein System zur Anwendung auf Inhalte des WWW ist in Arbeit.
- *http://fabdp.fh-potsdam.de/infoviz/*
  Infoviz – Visualisation of Dataspaces. Riesige Übersicht über alle Arten der Visualisierung.
- *http://www.kom.e-technik.tu-darmstadt.de/~ht99/*
  Webseite zur *ACM Hypertext '99* Konferenz in Darmstadt.
- *http://xdev.datachannel.com/default.asp*
  XML Resource Center von DataChannel, u.a. mit dem „XML 101", einer XML-Einführung.
- *http://www.xml.org/*
  Das XML „Industrie-Portal" mit umfangreicher Linksammlung.
- *http://www.xml.com/*
  Sehr umfangreiche XML-Resourcenseite des O'Reilly Verlages.

- *http://user.fachdid.fu-berlin.de/Docs/HTXT/htxt.htm*
  Einführung in Hypertexte fon Stefan Münz, den Autor von *SelfHTML.*