

Navigation im World Wide Web
als Hypertextsystem

Diplomarbeit von Tobias Steinke

April 1999

Technische Universität Darmstadt

Fachbereich Informatik

Fachgebiet Programmiersprachen und Übersetzer

Betreuer: Prof. Dr. H.-J. Hoffmann

Betreuerin: Dipl.-Inform. E. Siemon

Inhaltsverzeichnis

Einleitung.....	3
1. Hypertextsysteme.....	5
1.1 Historische Entwicklung von Hypertextsystemen.....	5
1.1.1 Memex.....	5
1.1.2 NLS/Augment.....	5
1.1.3 Xanadu.....	6
1.1.4 NoteCards.....	6
1.1.5 HyperCard.....	7
1.1.6 Hypertextsysteme heute.....	7
1.2 Konzepte zur Navigation im Hypertext.....	8
1.2.1 Probleme bei der Navigation im Hypertext.....	8
1.2.2 Standardunterstützungen bei der Navigation.....	8
1.2.3 Geführte Touren.....	9
1.2.4 Metaphern und Metainformationen.....	9
1.2.5 Übersichtsknoten als Navigationshilfe.....	10
1.2.6 Stufenindex.....	11
1.3 Beispiele für Navigation in Hypertextsystemen.....	11
1.3.1 Windows-Hilfe.....	11
1.3.2 Infopedia.....	13
1.3.3 The Star Trek Encyclopedia.....	14
1.3.4 Cyborg.....	15
2. Das World Wide Web.....	17
2.1 Historische Entwicklung des WWW.....	17
2.1.1 Das Internet.....	17
2.1.2 Entstehung des World Wide Web.....	18
2.1.3 HTML und der „Browser-Krieg“.....	18
2.2 Navigationsunterstützung in WWW-Browsern.....	19
2.2.1 Standardelemente in beiden Browsern.....	20
2.2.2 Netscape Communicator.....	21
2.2.3 Microsoft Internet Explorer.....	22
2.3 Konzepte zur Navigation im World Wide Web.....	23
2.3.1 World Wide Web als Hypertextsystem.....	23
2.3.2 Nutzung der angebotenen Navigationsmittel.....	23

2.3.3 Verschiedene Modelle der History.....	25
2.3.4 Graphische Übersichten.....	26
2.3.5 Reverse Links als Navigationshilfe.....	27
2.3.6 Textbasierte Metaphern.....	28
2.3.7 Navigation in speziellen Anwendungen.....	29
2.3.8 Ideen für die Entwicklung des WWW.....	30
3. AlterNav.....	32
3.1 Idee.....	32
3.2 Funktionsweise.....	33
3.3 Implementierung.....	34
3.4 Einschränkungen.....	35
3.5 Mögliche Weiterentwicklungen.....	36
Zusammenfassung und Ausblick.....	37
Literaturverzeichnis.....	38
Anhang: Quellcode von AlterNav.....	40

Einleitung

Das World Wide Web ist die Anwendung, die dem Internet zum Durchbruch als Massenmedium verhalf. Es ist im Wesentlichen ein verteiltes Hypertextsystem, das unaufhörlich wächst und damit immer komplexer wird. Die Folge ist, daß die klassischen Probleme von Hypertextsystemen, Orientierungslosigkeit und mentale Überforderung, eine neue Dimension erhalten. Ansätze zur Navigation und Orientierung müssen dem offenen Charakter des World Wide Web Rechnung tragen.

Hypertextsysteme gab es schon lange vor dem World Wide Web. Ihr besonderer Vorteil, das aktive Bestimmen des Textfortgangs, bringt aber auch Probleme mit sich. Alle Hypertextsysteme der Vergangenheit fanden zum Teil unterschiedliche Lösungen, die dem Benutzer und der Benutzerin Hilfen zur Navigation und Orientierung bereitstellten. Dabei waren die Systeme aber immer in sich geschlossen und im Umfang überschaubar. Während in den achtziger Jahren Hypertextsysteme eher Gegenstand experimenteller Forschung waren, werden sie heute weit verbreitet eingesetzt und finden sich sowohl im Hilffsystem von Standardapplikationen als auch im unterhaltenen Informationsbereich („Edutainment“). Kapitel 1 beschreibt die Entwicklung von Hypertext und stellt verschiedene Ansätze zur Navigation und Orientierung darin vor. Dabei wird aber auch die jeweilige Eignung für das World Wide Web untersucht.

Das World Wide Web als Anwendung des Internet hat in kurzer Zeit nach der Entstehung eine stürmische Entwicklung erfahren. In wenigen Jahren wurde so aus dem Internet als weitgehend wissenschaftlich genutztem Medium ein populäres Massenmedium, dessen gesellschaftliche Auswirkungen noch gar nicht abzuschätzen sind. Das Angebot hat sich explosionsartig vergrößert und kommerzielle Interessen stehen gleichberechtigt neben privatem Idealismus. Auf technischer Seite hat sich mit der erfolgreichen Applikation der Firma *Netscape* und dem darauffolgendem Konkurrenzkampf mit der Firma *Microsoft* eine Eigendynamik jenseits von Standards entwickelt. Der damit verbundene schnelle Fortschritt bei der Benutzungsfreundlichkeit bringt aber auch den für ein offenes System verheerenden Nachteil der Inkompatibilität mit sich. Verglichen mit der Hypertextforschung bot das World Wide Web anfangs sehr wenig Unterstützung für die Navigation und Orientierung. Und auch jetzt kann vieles durch die offene Natur des Systems nur bedingt durchgeführt werden. Kapitel 2 zeigt die Entwicklung des World Wide Web und existierende Ansätze zur Unterstützung von Navigation und Orientierung.

In Kapitel 3 stelle ich das von mir entwickelte alternative Navigationswerkzeug *AlterNav* vor. Als wichtiges Mittel bei der Navigation in Hypertextsystemen wird die Liste

mit den besuchten Knoten, die sogenannte History, angesehen. *AlterNav* ist eine in HTML und JavaScript realisierte Anwendung, die das Verwalten mehrerer History-Listen innerhalb eines Seitenangebots ermöglicht. Auch bei späteren Besuchen können damit Pfade durch die Seiten nachvollzogen werden und der Kontext zu einzelnen Seiten bleibt erhalten.

1. Hypertextsysteme

Hypertext beschreibt eine Darstellung von Text in nichtlinearer Form. Verschiedene Knoten aus Text, die mit Verweisen („Links“¹) verbunden sind, bilden zusammen einen Hypertext. Ted Nelson definierte Hypertext als „a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ... of a non-linear text ... which cannot be printed conveniently on a conventional page.“ (zitiert nach Conklin, 1987). Darin kommt zum Ausdruck, daß diese Form der Textdarstellung, die dem menschlichen Denken angepaßt ist, eine Maschine wie den Computer zwingend voraussetzt. Erst dessen Fähigkeiten ermöglichen einen schnellen Wechsel zwischen den vernetzten Textknoten. Heute wird auch oft von Hypermedia gesprochen, da die Knoten auch andere Medien enthalten können (z.B. Bilder, Ton oder Video).

1.1 Historische Entwicklung von Hypertextsystemen

1.1.1 Memex

Als Vordenker zum Hypertext wird Vannevar Bush² angesehen. In seinem Artikel „As We May Think“ (Bush, 1945) stellte er sein Konzept von *Memex* vor, einer Wissensdatenbank, die einzelne Dateneinträge miteinander verknüpft. Er wollte die für das menschliche Denken als wesentlich erkannte Fähigkeit der Assoziation für den Zugriff auf maschinell gespeichertes Wissen übertragen. Die technische Grundlage sollten dabei Mikrofilme und mehrere parallele Projektoren sein. Mit Hilfe eines „Scanners“ hätte das System beliebig um persönliche Notizen, Bilder oder Artikel ergänzt werden können. Neben der individuellen Erschließung des gespeicherten Wissens, sah Bush auch die Möglichkeit des Aufbaus von „Pfadern“ durch das Datenmaterial vor. Dabei ging er so weit, die Entstehung des Berufs eines „Pfadsuchers“ vorherzusagen. *Memex* wurde nie realisiert.

1.1.2 NLS/Augment

Erst mit der fortschreitenden Entwicklung der Computern zeichnete sich eine technische Realisierbarkeit der Ideen von Bush ab. 1963 nahm Douglas Engelbart die Gedanken Bushs in „A Conceptual Framework for the Augmentation of Man's Intellect.“ auf und sah Computer als geeignete Werkzeuge dafür vor (vgl. Conklin, 1987). Daraus entstand

1 Wie im amerikanisch dominierten Bereich der Informationstechnologie häufig, hat sich auch hier der englische Begriff „Link“ gleichberechtigt mit dem deutschen „Verweis“ etabliert.

2 Vannevar Bush (1890-1974), wissenschaftlicher Berater des amerikanischen Präsidenten Roosevelt.

in den folgenden Jahren das System *NLS/Augment*, welches Dateien und Daten hierarchisch benennt und Links zwischen diesen ermöglicht. Als Eingabemedien dafür erfand Engelbart eine Reihe von Geräten, von denen die „Maus“ das bekannteste ist. *NLS/Augment* besteht aus einer Datenbank mit nichtlinearem Text, Sichtfiltern, die Informationen aus dem Datenbestand auswählen, und Sichten zur strukturierten Darstellung auf dem Bildschirm. Sichtfilter, die einzelne Hierarchieebenen oder Daten nach inhaltlichen Kriterien ausblenden können, sind durch den Benutzer bzw. die Benutzerin in einer eigenen Sprache selbst definierbar. Zudem ist die verteilte Mehrpersonennutzung vorgesehen.

1.1.3 Xanadu

Der Begriff „Hypertext“ wurde 1965 mit dem System *Xanadu* von Ted Nelson eingeführt (vgl. Nielsen, 1995). Dabei handelt es sich eher um ein Konzept, da *Xanadu* nie vollständig realisiert wurde. Nelsons Idee beschreibt die Sammlung aller geschriebenen Texte in einem großen Hypertext. Literaturverweise sind Links und Zitate werden immer mit dem Originaltext verbunden. Neuere Versionen des gleichen Artikels werden dem System hinzugefügt, ohne die alten zu löschen. Dadurch wird eine zeitliche Entwicklung deutlich, aber die Textmenge nimmt unaufhörlich zu. Für das entstehende Urheberrechtsproblem ist eine automatische Abrechnung beim Aufruf eines Artikels vorgesehen. Nelsons Konzept ist mit dem World Wide Web (WWW) in greifbare Nähe gerückt. Immer mehr wissenschaftlichen Arbeiten werden im WWW veröffentlicht. Allerdings ist die Textbasis nicht annähernd groß genug, um Literaturangaben nur auf Links zu stützen. Außerdem gibt es derzeit kein allgemeines System zur finanziellen Abrechnung von abgerufenen Texten (nur Individuallösungen), so daß kommerzielle Artikel kaum im WWW zu finden sind. Leider werden heute meistens ältere Version von WWW-Seiten durch neuere ersetzt. Hier wäre ein System zur Archivierung sicher sehr sinnvoll und teilweise auch dringend nötig. So bleibt *Xanadu* in Hinsicht auf das World Wide Web Vorbild und Vision zugleich.

1.1.4 NoteCards

Nach den ersten funktionstüchtigen Hypertextimplementierungen, *Hypertext Editing System* und *FRESS* von 1967 bzw. 1968 (vgl. Nielsen, 1995), gab es zunehmend mehr Hypertextsysteme, die meist aus Forschungsprojekten hervorgingen. Auch an einer der bedeutendsten Forschungsstätten, *Xerox PARC*, die zahlreiche Entwicklungen hervorbrachte (u.a. graphische Benutzungsoberflächen, Laserdrucker und Ethernet), befaßte man sich mit Hypertext (vgl. Conklin, 1987). Das dort entwickelte System *NoteCards* (1985) läuft auf der *PARC* eigenen D-Maschine, einer LISP-Workstation. Diese bietet einen hochauflösenden Bildschirm und eine fensterorientierte Benutzungsoberfläche.

Ziel von *NoteCards* war die Unterstützung bei der Erstellung analytischer Berichte. Um ein konzeptionelles Modell bilden zu können, werden Knoten mit Links verknüpft. Dazu gibt es vordefinierte Knotentypen für Text, Video und Animationen. Mit LISP können aber auch eigene Knotentypen geschaffen und sogar ganze Applikationen auf *NoteCards* aufbauend realisiert werden. Das System hat sich dabei im großen Einsatz mit vielen tausend Knoten und Links bewährt.

1.1.5 HyperCard

Trotz des erhöhten Forschungsinteresses fanden Hypertextsysteme nicht die große kommerzielle Verbreitung. Erst mit dem System *HyperCard* (1987) von *Apple*, das bis 1992 jedem *Macintosh*-Computer beigelegt wurde, änderte sich dies (vgl. Nielsen, 1995). Der *Macintosh*, als einer der ersten mit einer graphischen Benutzeroberfläche ausgestatteten PCs, eignet sich besonders für eine solche Anwendung. *HyperCard* ist eigentlich eher ein Autorenwerkzeug, d.h. ein Werkzeug zur Schaffung von Hypertextsystemen. Mit der leicht zu lernenden Sprache *HyperTalk* können graphische Anwendungen unterschiedlichster Art geschaffen werden. Diese basieren weitgehend auf der Karten-Metapher. Links können dynamisch zur Laufzeit berechnet werden. Viele Beispielsysteme auch im wissenschaftlichen Bereich basieren auf *HyperCard*, da es weit verbreitet war und flexibel ist.

1.1.6 Hypertextsysteme heute

Ende der achtziger Jahre setzten sich leistungsstarke PCs mit graphischer Benutzeroberfläche endgültig durch. Obwohl es auch leistungsfähige Hypertextsysteme auf reiner Textbasis gibt, schafften doch erst die fensterbasierten, mausgesteuerten Oberflächen eine breite Akzeptanz. Schnell etablierte sich die Präsentationsform als Hypertext für große Textmengen am Bildschirm. Besonders die sogenannte Online-Hilfe, d.h., die jederzeit verfügbare Bildschirmvariante des Benutzungshandbuchs, ist nur noch als Hypertext sinnvoll nutzbar. Standards für die dabei verwendeten Hypertextsysteme sind meist durch das Betriebssystem vorgegeben. Auch hier zeigt sich aber ein Trend zu HTML und den damit verbundenen Beschränkungen (siehe Kapitel 2).

Mit der CD-ROM als neuem Speichermedium, das viel größere Datenmengen aufnehmen kann, war auch die Basis für interaktive Nachschlagewerke und Lernprogrammen geschaffen. Damit wurde auch Hypermedia, also die Integration verschiedener Medien, insbesondere Bild und Ton, realisierbar. Im breiten Angebot finden sich sowohl reine Unterhaltungstitel (viele Computerspiele nutzen Hypermedia) als auch seriöse Informationssammlungen. Für die neu entstandene Kategorie der unterhaltenden, interaktiven Wissensvermittlung hat sich die Begriffsschöpfung „Edutainment“ durchgesetzt. Als

technische Basis für diese Hypertextsysteme kommt häufig das Autorenwerkzeug *Macromedia* zum Einsatz. Eine eigene Literaturgattung hat sich mit Hypertext entwickelt, die sogenannte „Hyperfiction“. Hierbei ist besonders das Hypertextsystem *Storyspace* verbreitet.

Das Prinzip von Hypertext ist heute in alle Bereiche der Computernutzung eingeflossen und dabei für viele der Inbegriff der Computerbedienung geworden. Dazu hat entscheidend das World Wide Web beigetragen, worauf in Kapitel 2 eingegangen wird.

1.2 Konzepte zur Navigation im Hypertext

1.2.1 Probleme bei der Navigation im Hypertext

Hypertext präsentiert den Text nicht in einer vorher festgelegten Reihenfolge, sondern ermöglicht es dem Leser bzw. der Leserin, selbst den nächsten Abschnitt (Knoten) zu wählen. Dazu verweisen Links im aktuellen Knoten auf mehr oder weniger in Beziehung stehende andere Knoten. Einen Hypertext zu lesen heißt also, sich durch ein Netz von Knoten mit Textabschnitten zu „hangeln“. Zwei Hauptprobleme tauchen dabei auf. Zum einen verliert der Leser bzw. die Leserin schnell die Orientierung und fühlt sich „verloren“ in der scheinbar unüberschaubaren Knotenmenge. Zum anderen führen die „Exkursionen“ zu weniger relevanten Knoten zu einer mentalen Überforderung. Die freie Wahl des zu verfolgenden Pfades schafft eine Vielzahl eingeschlagener Pfadstücke, in denen die ursprüngliche Informationssuche in den Hintergrund gerät. Zur Lösung dieser Probleme bieten alle Hypertextsysteme Orientierungs- und Navigationshilfen an.

In Hinblick auf das World Wide Web muß aber schon hier gesagt werden, daß viele dieser Konzepte auf der Überschaubarkeit und Konsistenz von geschlossenen Systemen beruhen. Genau diese Eigenschaften hat jedoch das WWW nicht.

1.2.2 Standardunterstützungen bei der Navigation

Jakob Nielsen, der sich intensiv mit der Navigation im Hypertext beschäftigt hat, faßt die gängigen Lösungsansätze für diese Probleme zusammen (Nielsen, 1990 und 1995). Die wohl wichtigste Navigationsstrategie ist das *Backtracking*. Dabei ermöglicht ein „Zurück“-Knopf den erneuten Aufruf des vorher besuchten Knotens. Der Benutzer bzw. die Benutzerin ist so unbefangener beim Einschlagen eines Seitenpfades, kann er bzw. sie doch problemlos zurück zum Ausgangspunkt. Für das Backtracking muß zumindest intern eine Liste aller bisher besuchten Knoten mitgeführt werden. Da liegt es nahe, die-

se *History* dem Benutzer bzw. der Benutzerin zugänglich zu machen, so daß ein entfernter zurückliegender Knoten direkt angesprungen werden kann.

Zur Orientierung hilfreich sind graphische *Übersichtsdiagramme*, in denen besuchte Knoten, der aktuelle Knoten und von hier aus besuchbare Knoten mit allen Verknüpfungen dargestellt werden. Auch eine mögliche hierarchische Gliederung der Knoten kann in so einem Diagramm verdeutlicht werden.

Der Umgang mit der neuen Präsentationsform kann mit geeigneten *Metaphern*, z.B. Kartenstapel oder Buchseiten, vereinfacht werden. Dazu gehören auch *Lesezeichen*. Der aktuelle Knoten (bzw. ein Verweis darauf) wird in eine Merkliste aufgenommen und kann später sofort wieder aufgerufen werden. Schon besuchte Links werden zur Orientierung besonders markiert, Nielsen nennt das *Fußspuren*.

Eine Form der Personalisierung stellen *Anmerkungen* dar. Der Benutzer bzw. die Benutzerin kann dabei jedem Knoten persönliche Notizen anfügen.

1.2.3 Geführte Touren

Eigentlich dem Hypertextgedanken entgegenstehend, ist die Idee von *geführten Touren* (Nielsen, 1995). Dabei wird dem Benutzer bzw. der Benutzerin ein „Königsweg“ durch die Hypertextdaten angeboten, der eine bestimmte Informationspräsentation sicherstellt. Dies entspricht den „Pfad“ des *Memex* von Bush. Eine solche Tour gibt dem Leser bzw. der Leserin eine gewisse Sicherheit und Orientierung. Trotzdem sollte ihm bzw. ihr die Möglichkeit zu „Exkursionen“ abseits der Tour gegeben werden, sonst könnte gleich ein sequentieller Text als Medium gewählt werden.

Im WWW ist eine solche Tour insbesondere zur Übersicht eines themenbezogenen Seitenangebots denkbar. Seiten, die sich dem gleichen speziellen Thema widmen, schließen sich schon heute oft in einem sogenannten *Webring* zusammen. Dabei beinhaltet jedes eingeschlossene Angebot je zwei Links zu anderen Partnern im Ring („Vor“ und „Zurück“). Eine weitergehende Tour über Seiten von verschiedenen Anbietern setzt aber eine permanente enge Zusammenarbeit und Abstimmung voraus, die selten funktioniert.

1.2.4 Metaphern und Metainformationen

Dem Benutzer bzw. der Benutzerin eines Hypertextsystems müssen Hilfen zur Verfügung gestellt werden, um sich ein mentales Modell bilden zu können (Ventura, 1990). *Metaphern* können den Umgang mit der neuen Form Hypertext erleichtern, aber bei falscher Anwendung des Vergleichsmodells auch verwirren. Beliebt sind hierbei *Kartei-*

karten, Kartenstapel oder *Buch*. Bei zunehmender Verbreitung von Hypertextsystemen, wie es heute der Fall ist, sollten solche Hilfsmodelle jedoch in den Hintergrund treten. Wichtig ist vor allem, sich des Mediums Computer bzw. Bildschirm bewußt zu sein. Die Bildschirmauflösung ist geringer als die eines Buches und Lesen am Bildschirm dauert länger. Ventura empfiehlt daher wenig Worte und Graphik statt Text. Besonders letztere Empfehlung wird im World Wide Web mit längeren Lade- bzw. Antwortzeiten erkaufte, die eher schaden als nutzen.

Als *Metainformationen* werden die nicht zum eigentlichen Inhalt gehörenden Informationen, die die Handhabung erleichtern, bezeichnet. Dazu zählen Benutzungshinweise, also ein erläuternder Hilfstext. Dies ist besonders bei in sich abgeschlossenen Hypertextsystemen sehr sinnvoll. Da aber im World Wide Web möglichst nur Standardelemente Verwendung finden sollten, sind dort vorwiegend Informationen zur spezifischen Navigation auf bestimmten Seiten nötig. Weitere Metainformationen sind Inhalts- und Schlagwortverzeichnisse, im WWW kann man auch lokale Suchmaschinen dazu zählen. *Kontextinformationen* über die Einordnung der gerade angezeigten Informationen in das große Ganze sind gerade im WWW wichtig, dort aber nur lokal (für die eigenen Seiten) machbar.

1.2.5 Übersichtsknoten als Navigationshilfe

Bei der theoretischen Betrachtung von Hypertext ist die Trennung impliziter und expliziter Navigationshilfen denkbar (Purgathofer, 1993). Implizit meint hier die Hypertextverweise (Links) der Textknoten, explizit die speziellen *Übersichtsknoten* zur Verwaltung und Navigationsunterstützung. Übersichtsknoten in diesem Sinne sind gleichberechtigte Knoten, die ausschließlich aus Links zu anderen Knoten bestehen.

Übersichtsknoten können ein Inhaltsverzeichnis, die History, Lesezeichen oder geführte Touren enthalten. Je nach Anwendung werden sie dabei dynamisch erzeugt oder zusammen mit den Textknoten festgelegt. Suchanfragen liefern als Ergebnis ebenfalls Übersichtsknoten. Das Hypertextsystem kann eine besondere Darstellung dieser Übersichtsknoten vorsehen, die eine einheitliche Navigation sicherstellt. Aus den Knoten können mit Mengenoperationen neue Übersichtsknoten generiert werden. Ein Beispiel wäre eine Liste aller Knoten, die einer Suchanfrage entsprechen und am selben Tag schon besucht wurden (Verknüpfung von History und Suchergebnis).

Im World Wide Web hat sich ein spezieller Knotentyp, der ausschließlich aus Links besteht, nicht durchgesetzt. Obwohl die Software von *Netscape* intern eine solche Darstellung für seine Lesezeichenliste benutzt, gibt es keinen übergreifenden Standard, so daß auch logische Verknüpfungen nicht angeboten werden. Suchmaschinen im World Wide

Web liefern zwar auch als Ergebnis ein Liste mit Links, die aber ergänzt ist um Kommentare und weiteren Inhalt und damit keinen Übersichtsknoten darstellt. Ein Abgleich aller vorkommenden Links in Textknoten ist aber denkbar und wird von sogenannten Metasuchmaschinen¹ auch praktiziert.

1.2.6 Stufenindex

Dem *Stufenindex* liegt ein ähnliches Konzept zugrunde, wie den Übersichtsknoten (Herold, 1993). Dabei wird zu jedem Knoten ein weiterer Knoten geschaffen, in dem alle ausgehenden Links gespeichert werden, aber auch alle auf diesen Knoten verweisenden. Mit solchen Eltern-Kind-Listen kann einfach ein Hypertextgraph generiert werden, auf dem die verschiedenen Methoden der Graphentheorie Anwendung finden. Wie bei den Übersichtsknoten findet eine semantische Trennung von Navigation und Inhalt statt. Dadurch können graphische Übersichten problemlos erstellt werden und eine automatisierte Bearbeitung wird besser unterstützt. Denkbar sind auch weitergehende Attribute der Links, die im Stufenindex enthalten sind.

Im World Wide Web gibt es eine Trennung von Links und Inhalt noch nicht. Besonders Rückverweise auf Elternknoten sind bei dem sich ständig verändernden WWW nicht realisierbar. *Attributierte und typisierte Links* werden jedoch diskutiert.

1.3 Beispiele für Navigation in Hypertextsystemen

Hier sollen die Navigationsunterstützungen vier verschiedener Hypertextanwendungen vorgestellt werden. Dabei handelt es sich um das Hilffsystem² des weit verbreiteten Betriebssystems *Windows* von *Microsoft*, das multimediale Nachschlagewerk³ *Infopedia* vom *Tewi Verlag*, das Nachschlagewerk *The Star Trek Encyclopedia* von *Simon & Schuster Interactive* aus dem unterhaltenden Bereich und den seriösen Hypertext⁴ *Cyborg* von *Eastgate Systems*.

1.3.1 Windows-Hilfe

Die *Windows-Hilfe* stellt eine Online-Dokumentation für das Betriebssystem *Windows* dar. Insbesondere Beschreibungen zu bestimmten Funktionen und Problemen sollen

1 Metasuchmaschinen stellen Anfragen an mehrere Suchmaschinen und präsentieren dann alle Ergebnisse in einer Liste, was einer ODER-Verknüpfung entspricht.

2 Betrachtet wird die *Windows-Hilfe* des deutschsprachigen *Windows 95*.

3 Grundlage sind die englischsprachige und die deutschsprachige Version der *Infopedia 2.0*.

4 Betrachtet wird die amerikanische Version von *Cyborg* für *Windows*.

schnell zugreifbar sein. Zentrale der Navigation ist dafür eine aus drei Teilen bestehende Übersicht mit dem Namen „Hilfethemen“. „Inhalt“ bietet eine hierarchisch gegliederte Inhaltsübersicht nach Themen geordnet. „Index“ listet alphabetisch sortiert Schlagworte und Themengebiete auf. „Suchen“ schließlich ermöglicht eine Volltextsuche über die ganze Textbasis.

Innerhalb der Textfenster gibt es drei Typen von Links, die jeweils eine unterschiedliche Semantik aufweisen. Ein Kästchen vor einem Begriff öffnet bei Anklicken entweder direkt ein weiteres Textfenster oder zeigt eine Liste mit weiteren Links. Befindet sich ein Pfeil in solch einem Kästchen wird die beschriebene Funktion des Betriebssystems aufgerufen. Unterbrochen unterstrichene Begriffe öffnen bei Anklicken ein Pop-Up-Fenster, mit einer kurzen Erklärung des Wortes (Glossar). Diese eindeutige Zuordnung von Form und Funktionalität der Links stellt eine Typisierung dar.

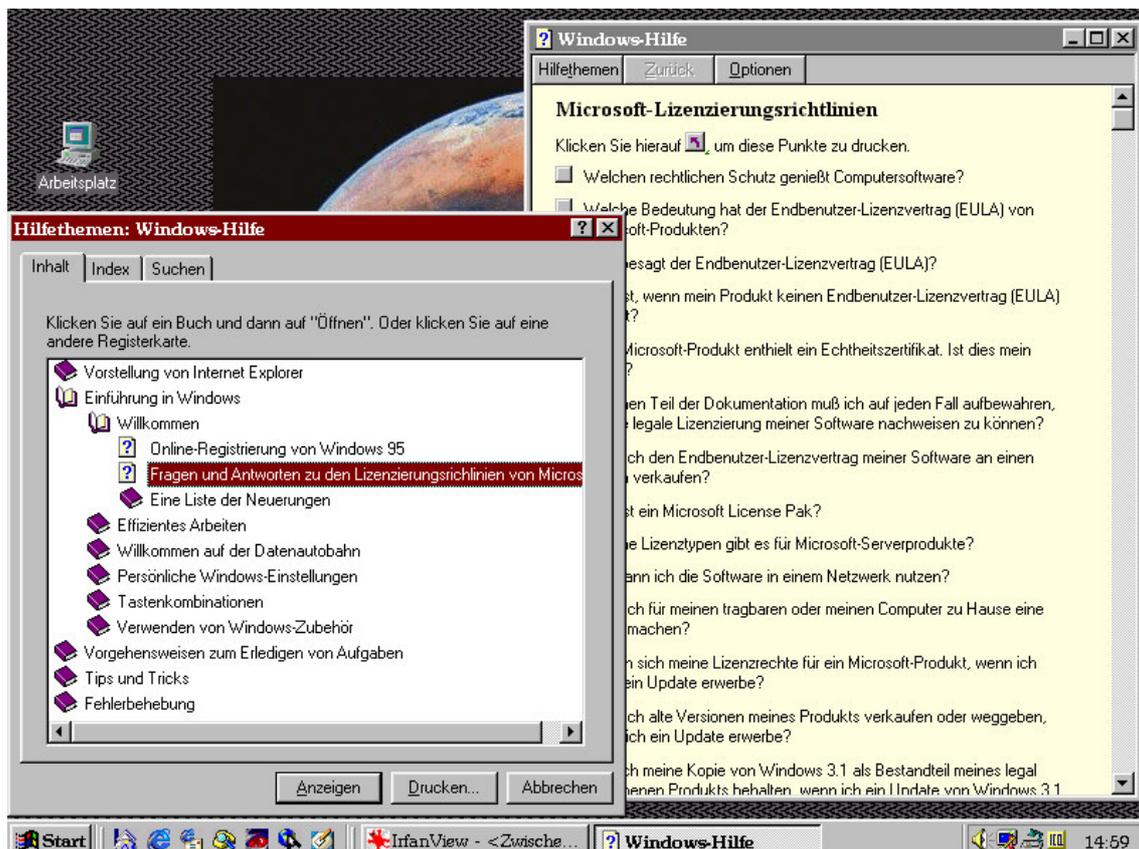


Abbildung 1: Windows-Hilfe

Im Textfenster befindet sich ein „Zurück“-Knopf. Manche Beschreibungen gehen über mehrere Seiten, wozu dann die Knöpfe „<<“ und „>>“ angeboten werden. Zu jedem Textfenster können persönliche Anmerkungen geschrieben und abgespeichert werden. Die Benutzung von History und Lesezeichen ist nur in einigen Fenstern möglich. Be-

suchte Links werden nicht besonders gekennzeichnet. Es gibt einen Hilfstext zur Benutzung des Hypertextsystems.

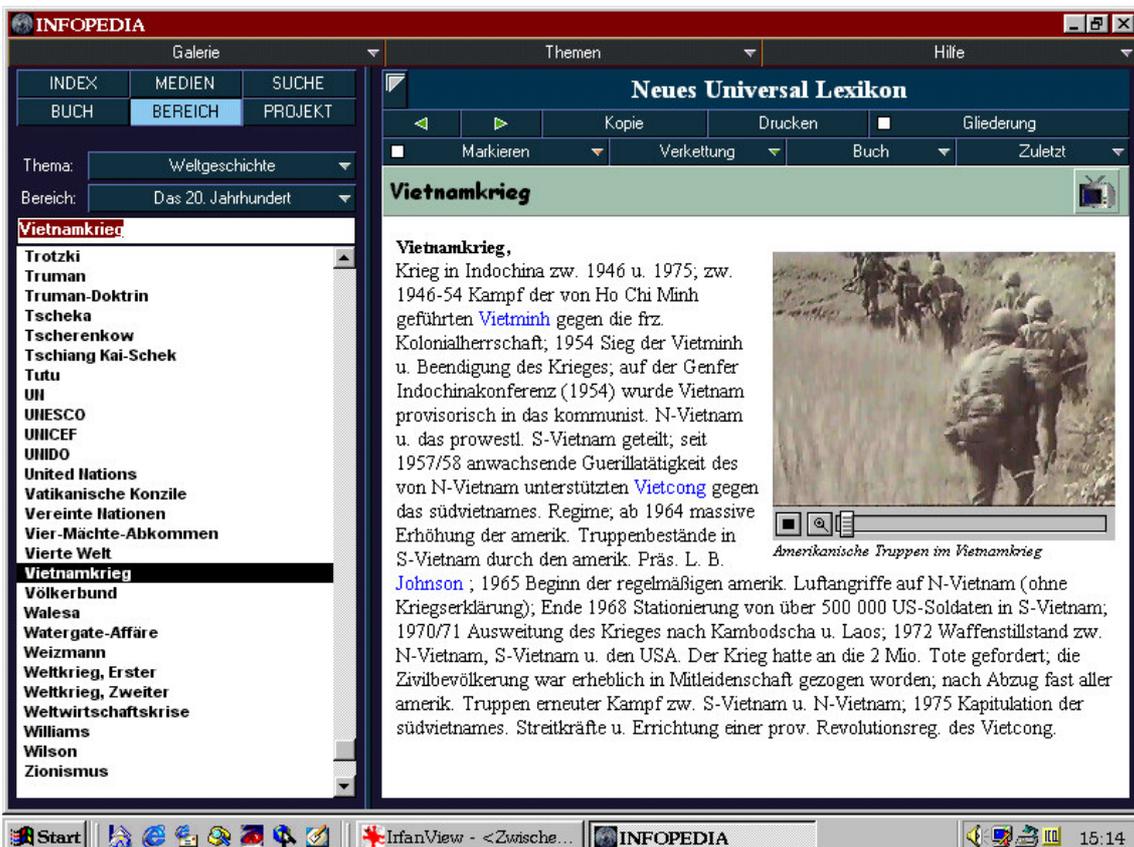


Abbildung 2: Infopedia

1.3.2 Infopedia

Die *Infopedia* ist eine Sammlung von Nachschlagewerken, Lexikon und Wörterbücher, die unter einer Oberfläche vereint sind. Es gibt eine deutschsprachige und eine englischsprachige Version davon, die sich in Umfang und Auswahl zwar stark unterscheiden, deren zugrunde liegendes Hypertextsystem aber das gleiche ist. Die Darstellung gliedert sich in eine Liste links und ein Fenster rechts für Text, Graphik, Ton oder Video.

Die Liste enthält einen alphabetischen Index der Schlagwörter aller enthaltenen Bücher. Sie kann nach bestimmten Kriterien gefiltert werden und zwar nach den verwendeten Büchern¹, nach Themenkategorien und nach den im Knoten enthaltenen Medien. Weiterhin ist eine Volltextsuche über alle oder einzelne Bücher möglich, deren Ergebnis

¹ Hierbei ist weniger eine Metapher gemeint, als vielmehr die zugrunde liegende Textbasis aus verschiedenen Nachschlagewerken, die ursprünglich auch als gedruckte Bücher erhältlich sind.

dann in der Liste gezeigt wird. Der in der Liste aktivierte Eintrag wird im rechten Fenster gezeigt. Gibt es zu einem Schlagwort mehrere Knoten in verschiedenen Büchern, kann mit einem Pop-Up-Menü der gewünschte ausgewählt werden. Im Text werden Links zu anderen Knoten durch blaue Textfarbe gekennzeichnet. Ein Pull-Down-Menü listet außerdem alle im Text enthaltene Links auf. Ein Doppelklick auf ein nicht hervorgehobenes Wort des Textes ruft den dazu gehörigen Eintrag (soweit vorhanden) im Buch „Rechtschreibung“ auf.

Es gibt einen „Zurück“-Knopf (hier im Deutschen „Zuletzt“ genannt) und eine History. Die Knöpfe „<“ und „>“ rufen den vorhergehenden bzw. nachfolgenden Knoten in der Liste auf. Lesezeichen sind möglich, werden aber nicht dauerhaft abgespeichert. Es können keine eigenen Anmerkungen gemacht werden. Besuchte Links werden nicht besonders gekennzeichnet. Es gibt einen Hilfstext zur Benutzung des Hypertextsystems.

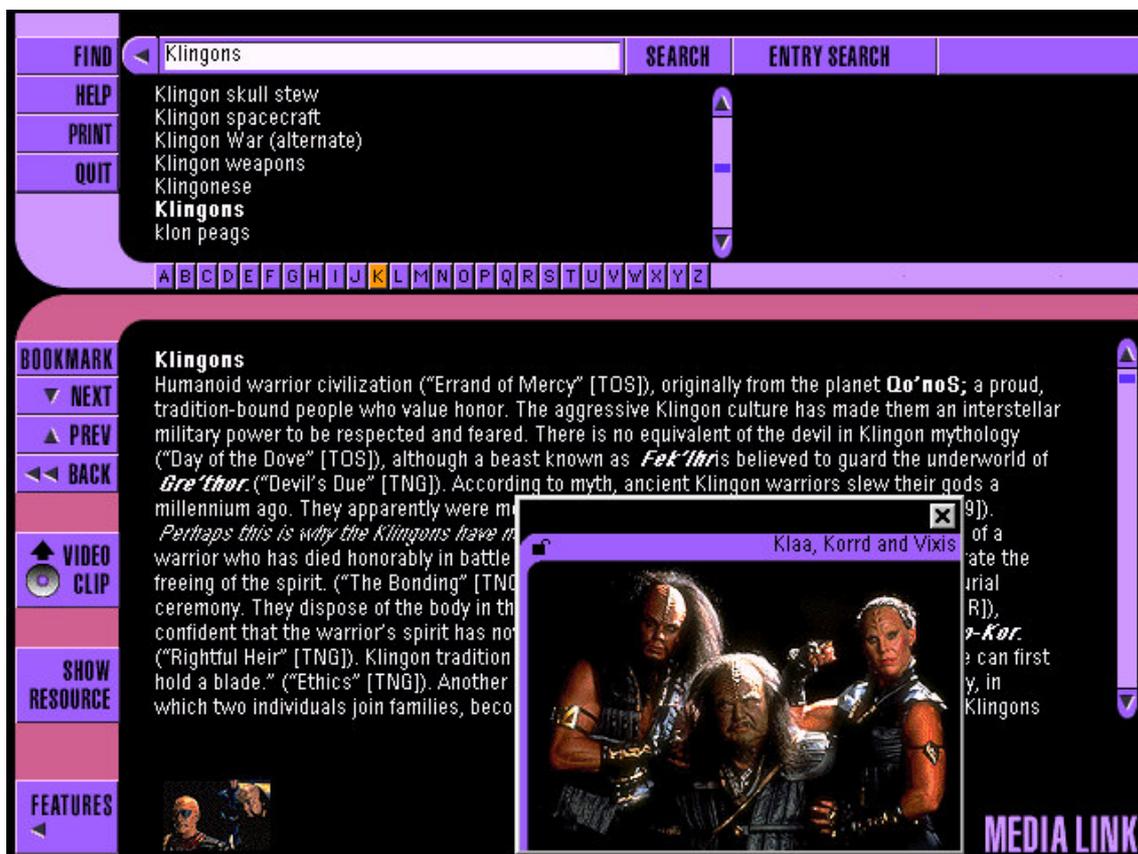


Abbildung 3: The Star Trek Encyclopedia

1.3.3 The Star Trek Encyclopedia

The Star Trek Encyclopedia ist genauso wie die *Infopedia* eine multimediale Umsetzung von gedruckten (englischsprachigen) Büchern. Es handelt sich dabei um eine Sammlung von Ereignissen aus einer Science Fiction Fernseh- und Kinoserie. Grundlage ist ein Le-

xikon mit Namen und Begriffen aus der Serie, die Episodenbeschreibungen aller Folgen und eine Chronologie der fiktiven Ereignisse. Das verwendete Hypertextsystem wurde mit dem Autorenwerkzeug *Macromedia* entwickelt, das besonders im Edutainment-Bereich häufig verwendet wird. Die drei verwendeten Bücher werden grundsätzlich getrennt ausgewählt, enthalten aber Links untereinander.

Die Anzeige gestaltet sich als eine alphabetische Liste im oberen Teil und einem Darstellungsbereich im unteren. Die Liste zeigt entweder Schlagworte des Lexikons, Episodentitel oder Jahreszahlen an. Sie kann nach inhaltlichen Aspekten der Serie gefiltert werden. Die Suchfunktion kann über allen drei Indizes laufen, aber auch eine Volltextsuche durchführen. Im Anzeigebereich wird der in der Liste aktivierte Text dargestellt. Zugehörige Bilder und Videos werden in Pop-Up-Fenstern gezeigt. Fettgedruckte Worte sind Links zu anderen Knoten. Ein Klick auf normalen Text startet eine Volltextsuche.

Es gibt einen „Zurück“-Knopf, aber keine History. Mit „<“ und „>“ wird der vorhergehende bzw. nachfolgende Knoten in der Liste aufgerufen. Lesezeichen sind möglich und können auch abgespeichert werden. Es können zwar keine eigenen Anmerkungen zu vorhandenen gemacht, jedoch völlig neue Knoten erstellt werden. Besuchte Links werden nicht besonders gekennzeichnet. Es gibt einen Hilfstext zur Benutzung des Hypertextsystems. Eine ungewöhnliche Funktion ist die zufällige Präsentation von Knoten, was dem willkürlichen Aufschlagen eines Buches entspricht.

1.3.4 Cyborg

Cyborg von Diane Greco ist die Hypertext-Umsetzung eines Sachtextes, der anhand der sogenannten „Cyberpunk“-Literatur die menschliche Körperlichkeit diskutiert. Das zugrundeliegende Hypertextsystem ist *Storyspace*, welches besonders bei kommerziellen Veröffentlichungen von Hyperfiction¹ verbreitet ist. Der *Storyspace reader* ist ein eigenständiges Programm, das den Hypertext *Cyborg* als Datendatei lädt.

Storyspace stellt einen eigenen Desktop mit Menüleiste. Es gibt zwei Arten von Fenstern, Textfenster und graphische Übersichten. Im Textfenster wird der Inhalt der Knoten des anzuzeigenden Hypertextes (in dem Fall *Cyborg*) dargestellt. Graphische Übersichtsfenster bieten verschiedene Darstellungsformen aller Knoten und Links des Hypertextes, etwa als Baum oder Tabelle. Links im Text werden normalerweise nicht gekennzeichnet (nur nach Drücken einer Taste). Bei Klick auf ein nicht als Link definiertes

¹ *Hyperfiction* bezeichnet eine Literaturgattung von Erzählungen, die speziell für das Medium Hypertext verfaßt wurden.

Wort wird ein vorgegebener Link aktiviert. Das aktuelle Textfenster wird nach der Link-Aktivierung geschlossen und ein neues geöffnet, das in seiner Darstellung dem Knoteninhalt angepaßt ist. Anderes als bei den anderen Beispielen, ist bei *Cyborg* nicht das Auffinden bestimmter Informationen das Ziel. Vielmehr soll sich der Leser oder die Leserin den Text erschließen. Trotzdem gibt es auch eine Volltextsuche.

In jedem Textfenster (es können auch mehrere Textfenster geöffnet sein) gibt es einen „Zurück“-Knopf und eine History. Ebenso stehen Lesezeichen zur Verfügung. Diese können genauso wie die ganze Lesesitzung abgespeichert werden. Man kann eigene Anmerkungen und Schlüsselworte zu einzelnen Knoten erstellen. Jedem Link können eigene Namen gegeben werden. Zwischen schon besuchten und nicht besuchten Knoten wird optisch kein Unterschied gemacht. Die graphischen Übersichten erlauben die Darstellung der Knoten und Links in verschiedenen Detailstufen und eigener Anordnung. Leider gibt es zu den teilweise komplexen Funktionen von *Storyspace* keinerlei Hilfstext.

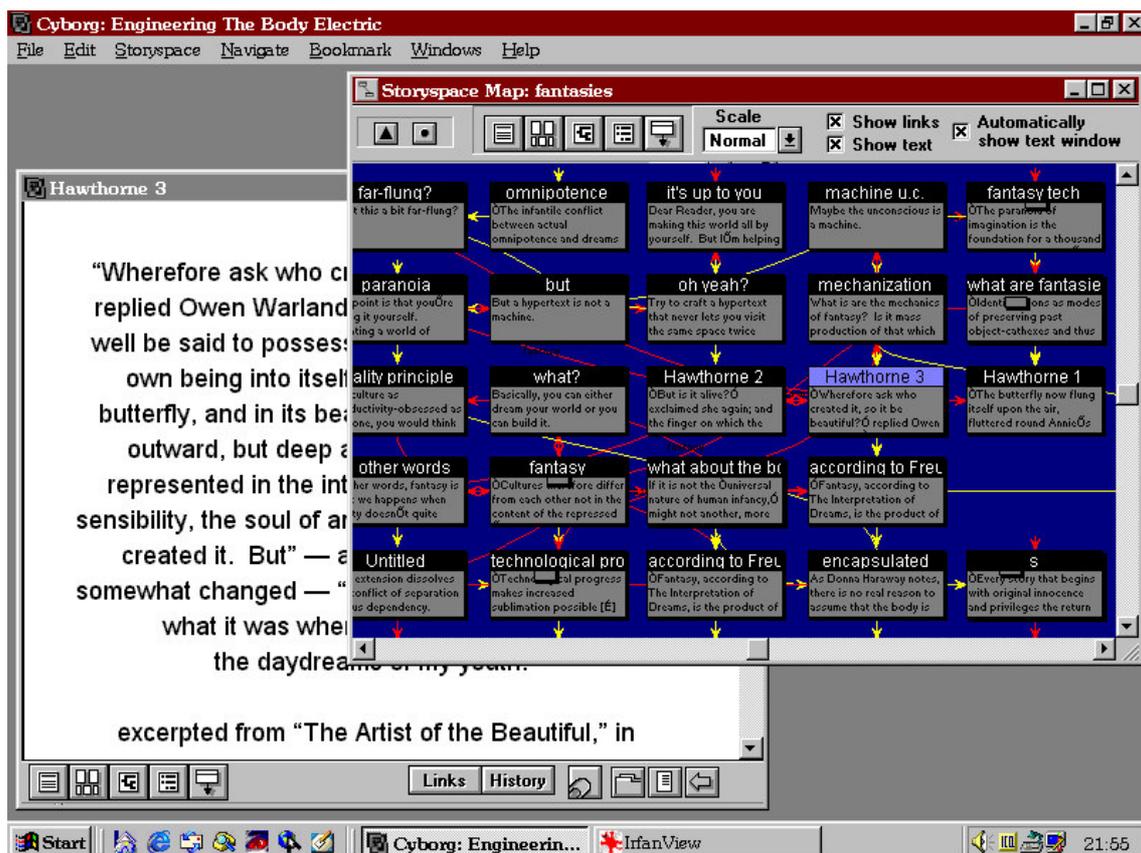


Abbildung 4: Cyborg

2. Das World Wide Web

Der Begriff *World Wide Web* (WWW) beschreibt eine Anwendung des Internet. Aufbauend auf dem dort verwendeten *Transmission Control Protocol / Internet Protocol* (TCP/IP) wird das *HyperText Transfer Protocol* (HTTP) zur Übertragung von Daten vom Server zum Client benutzt, welche normalerweise Dokumente in der Beschreibungssprache *HyperText Markup Language* (HTML) sind. HTML-Dateien enthalten Text und Verweise (Links) zu anderen HTML-Dokumenten oder beliebigen Dateien (Bilder, Töne, Videos). Ein Link im WWW ist eine hierarchisch strukturierte, eindeutige Adresse, der *Uniform Resource Locator* (URL). Auf der Client-Seite stellt eine spezielle Applikation, der *Browser*¹, das von der übertragenen HTML-Datei beschriebene Dokument (inklusive verknüpfter Graphiken und anderer Medien) dar. Durch Anklicken eines Links im Browser wird die mit der zugrunde liegenden URL benannte Datei vom Server übertragen.

2.1 Historische Entwicklung des WWW

2.1.1 Das Internet

Das *Internet* bezeichnet einen Zusammenschluß vieler lokaler Netze (Münz, 1998). Hervorgegangen ist es aus dem Projekt *ARPA-Net* des amerikanischen Militärs (1969). Ziel dabei war eine Vernetzung von selbständigen Einheiten, die auch bei Unterbrechung einzelner Verbindungen noch weiter funktioniert. Jede Einheit ist sowohl Sender als auch Empfänger. Aus dem militärischen Netz wurde bald ein wissenschaftlich genutztes zum Austausch von Forschungsergebnissen. Der militärisch genutzte Teil spaltete sich Anfang der Achtziger unter dem Namen *Milnet* ab, der Rest des *ARPA-Net* wurde bald nur noch *Internet* genannt.

Da im ständig wachsenden Netz aus Netzen verschiedenste Rechner integriert waren, mußte ein maschinenunabhängiger Standard für das Verbindungsprotokoll geschaffen werden. Dieses bildete das mehrschichtige Protokollsystem *Transmission Control Protocol / Internet Protocol* (TCP/IP). In Europa entstand auch ein Rechnerverbund, der nach anfänglichen Eigenentwicklungen auch TCP/IP unterstützte. Die in verschiedenen Kontinenten entstandenen Netzverbände wurden zusammen schließlich das globale Internet auf Basis von TCP/IP. Jeder Rechner darin hat eine eindeutige, 32-Bit-lange

¹ Als deutschsprachige Bezeichnung für Browser wird von Herrn Prof. Hoffmann u.a. „Stöberer“ vorgeschlagen. Inzwischen hat sich jedoch in der deutschen Umgangssprache der Begriff „Browser“ genauso etabliert wie z.B. „Computer“. Um deshalb Verwirrung beim Leser zu vermeiden, wird hier die übliche Bezeichnung „Browser“ verwendet.

Adresse (IP-Adresse). Um diese Zahlenadressen besser verständlich zu machen, werden sie einem hierarchisch strukturiertem Namensschema zugeordnet. Durch Punkte getrennt enthält ein solcher Adreßname verschiedene *Domänen*, die auf den Ort schließen lassen (z.B. kennzeichnet „.de“ deutsche Adressen und „.edu“ amerikanische Lehr- und Forschungseinrichtungen). Es entstanden verschieden Dienste und Anwendungen basierend auf TCP/IP. Die bekannteste ist *E-Mail*, das Übermitteln elektronischer Mitteilungen. Weitere Dienste sind *Usenet* (eine Art „Mitteilungsbrett“), *File Transfer Protocol* (FTP) (die Übertragung beliebiger Dateien) und *Telnet* (die „Fernsteuerung“ eines anderen Rechners im Netz).

2.1.2 Entstehung des World Wide Web

Tim Berners-Lee, ein Wissenschaftler der europäischen Forschungsstätte *CERN*, verfaßte 1989 eine Papier über die Idee eines Hypertextsystems im Internet (CERN, 1997). Zusammen mit Robert Cailliau arbeitete er diese Idee im folgenden Jahr weiter aus und erstellte Prototyp-Software für das *World Wide Web* (WWW) getaufte Projekt. 1991 wurde ein graphischer Browser für NeXT-Systeme verfügbar gemacht, kurz darauf dann eine einfache Textversion für andere Plattformen. Anfang 1993 gab es etwa 50 Server, die HTTP unterstützten. Am *National Center for Supercomputing Applications* (NCSA) der University of Illinois entwickelte Marc Andreessen im selben Jahr den Browser *Mosaic* für X Windows. *Mosaic*, das bald darauf auch auf andere Plattformen portiert wurde, verhalf mit seiner einfach zu bedienenden graphischen Oberfläche dem WWW zu großer Beliebtheit besonders im wissenschaftlichen Raum. Ende 1993 gab es bereits 500 HTTP-Server. 1993 wurde auch vom *CERN* entschieden, die WWW-Technik lizenzfrei zur Verfügung zu stellen (Cailliau, 1995). 1994 fand die erste World Wide Web Konferenz statt, die unerwartet hohen Zuspruch hatte. Das *CERN* gab die Weiterentwicklung an das unabhängige *World Wide Web Consortium* (W3C) ab. Marc Andreessen verließ NCSA und gründete die Firma *Netscape*, deren Browser *Navigator* den endgültigen Durchbruch für das WWW brachte.

2.1.3 HTML und der „Browser-Krieg“

Ab Mitte der Neunziger „explodierte“ das World Wide Web und damit auch das Internet förmlich. Die einfache, mausgesteuerte Bedienung des *Netscape*-Browsers machte das Internet nun auch für computerunerfahrene Privatnutzer interessant. Das WWW wurde plötzlich zur Hauptanwendung im Internet (neben E-Mail). Einfaches Klicken auf Links zum schnellen Wechsel zwischen Seiten auf verschiedenen Servern der ganzen Welt wurde unter dem Begriff „Surfen“ ein populäres Thema. Durch die Möglichkeit, als Benutzer oder Benutzerin recht einfach auch ein eigenes Angebot zu präsentieren, die sogenannte „Homepage“, wuchs auch die inhaltliche Menge sehr schnell an.

Grundlage für alle WWW-Seiten ist die Beschreibungssprache *HyperText Markup Language* (HTML), eine Ausprägung der wesentlich komplexeren *Standard Generalized Markup Language* (SGML). HTML bot in der Grundform nur sehr begrenzte Möglichkeiten zur Seitengestaltung. Für die Weiterentwicklung von HTML sollte eigentlich das W3C zuständig sein. Die stürmische Entwicklung und das Quasi-Monopol von *Netscape* führten jedoch dazu, daß jede neue Version des *Navigators* auch neue Erweiterungen von HTML brachte. Das W3C konnte erst nachträglich einige dieser Änderungen zum Standard erklären, welcher dann längere Zeit als HTML 3.2 bekannt war.

Ende 1995 erkannte die Firma *Microsoft* die Bedeutung des WWW und entwickelte einen eigenen Browser, den *Internet Explorer* (IE). Um auch im Internet-Bereich die gleiche Vorrangstellung wie bei anderer Software zu erhalten, wurde dabei ein erheblicher Aufwand getrieben. Während *Netscape* seine Software nur für Bildungseinrichtungen kostenlos anbot, wurde der IE für jedermann unentgeltlich zur Verfügung gestellt. Auch *Microsoft* entwickelte eigene HTML-Erweiterungen, was zu der unschönen Situation führte, daß das Aussehen vieler Seiten vom verwendeten Browser abhängt. Der IE konnte einen erheblichen Marktanteil¹ neben dem *Navigator* erzielen, nicht zuletzt durch die Integration in das verbreitete Betriebssystem *Windows* von *Microsoft*.

Die Browser haben sich inzwischen zu umfassenden Internet-Programmen entwickelt, die auch E-Mail, Usenet und andere Dienste anbieten. Daher nennt sich das nun auch kostenlos erhältliche Programm von *Netscape* jetzt *Communicator*. Um die Entwicklung nicht vollständig auseinander laufen zu lassen, stellte das W3C HTML 4.0 vor, das derzeit weder vom *Communicator* noch vom IE voll unterstützt wird. Trotzdem scheint sich HTML 4.0 mit seinen zahlreichen Erweiterungen als Standard durchzusetzen. Das Internet, dabei hauptsächlich das WWW, hat sich als globales Medium mit all seinen Möglichkeiten und Risiken im öffentlichen Bewußtsein festgesetzt².

2.2 Navigationsunterstützung in WWW-Browsern

Im folgenden werden die aktuellen Versionen der beiden meist verwendeten WWW-Browser *Netscape Communicator* und *Microsoft Internet Explorer* bezüglich ihrer Navigationsunterstützung verglichen. Für den *Communicator* ist dies Version 4.5, für den IE

1 Browser-Marktanteile in Deutschland im Oktober/November 1998: *Netscape* 52%, *Microsoft* 47% (Quelle: *Focus* 49/1998).

2 In Deutschland verfügen Ende 1998 fünf Millionen Haushalte über einen Online-Anschluß (Quelle: *Mainzer Allgemeine Zeitung* vom 26.10.98).

die Version 5.0 Beta 2, die nach Angaben von *Microsoft* im Funktionsumfang der endgültigen Version entsprechen soll.

2.2.1 Standardelemente in beiden Browsern

Die folgenden Navigationselemente gab es zum Teil schon in *Mosaic*. Dazu gehört ein „Zurück“-Knopf, aber auch ein „Vor“-Knopf, der eine Benutzung des „Zurück“-Knopfes rückgängig macht. Weiterhin zählen zu den Standardknöpfen „Aktualisieren“ zum Neuladen der gleichen Seite, „Abbrechen“ zur vorzeitigen Unterbrechung einer Seitenübertragung und „Homepage“ zum schnellen Aufruf einer definierten Seite (meist die Hauptseite des Browser-Herstellers).

In einer Zeile steht die URL der angezeigten Seite, die manuell geändert werden kann, um eine neue Adresse aufzurufen. Alle selbstgetippten URLs werden dabei in einer aufrufbaren Liste gespeichert. Daneben ist die History verfügbar und zwar zum einen als Pull-Down-Menü am „Zurück“ bzw. „Vor“-Button, zum anderen als eigene, sortierbare Liste. Diese Liste wird auch abgespeichert und kann später erneut benutzt werden.

Sehr weiterentwickelt hat sich seit *Mosaic* die Verwaltung der Lesezeichen. Sie können hierarchisch in Ordnern angeordnet werden und sind in einem Menü jederzeit verfügbar. Außerdem gibt es eine eigene Leiste, in der bestimmte Links noch schneller aufgerufen werden können. Text auf der angezeigten Seite kann mit einer Suchfunktion gefunden werden. Zu beiden Browsern gibt es Hilfstexte mit Benutzungshinweisen.

Bei beiden Browsern gibt es einen „Suchen“-Knopf, der aber nur eine Seite aufruft, die wiederum andere Suchmaschinen benutzt. Suchmaschinen durchsuchen das World Wide Web nach Begriffen, allerdings nur in ihnen bekannten Seiten. Jede Suchmaschine liefert also unter Umständen verschiedene Ergebnisse. Suchmaschinen sind heute mit das wichtigste Navigationswerkzeug im WWW, gehören aber eigentlich nicht zum Browser selbst. Sie werden, genauso wie andere Seiten, per URL vom Browser aufgerufen.

Text-Links im angezeigten HTML-Dokument werden normalerweise unterstrichen und blau dargestellt. Dies kann jedoch sowohl vom Autor bzw. von der Autorin des Dokuments als auch vom Benutzer bzw. von der Benutzerin des Browsers geändert werden. Schon besuchte Links werden mit einer anderen Farbe gekennzeichnet, in der Regel lila. Normalerweise findet die Navigation im selben Fenster statt. Es kann aber mit jedem Link auch ein neues Browser-Fenster geöffnet werden, das dann eigene Bedienelemente enthält.

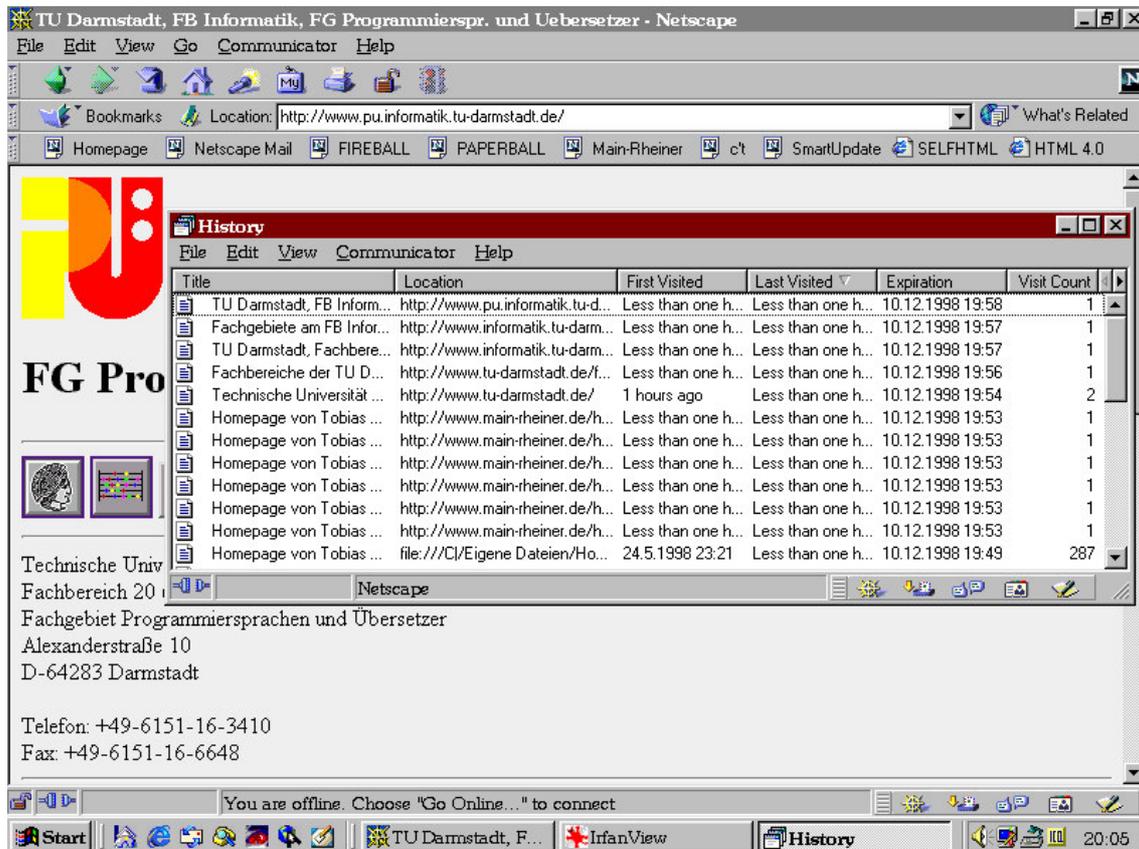


Abbildung 5: Netscape Communicator mit History-Liste

2.2.2 Netscape Communicator

Der Browser des *Netscape Communicator* bietet neben den im vorigen Abschnitt genannten Elemente einen „What's Related“-Button. Dieser zeigt ein Menü mit zur aktuellen Seite themenverwandten Links an, was allerdings nur mit einer Datenbankabfrage bei *Netscape* funktioniert und oft keine brauchbaren Ergebnisse liefert. Ein weiterer Dienst von *Netscape* ist eine personalisierbare Startseite, die auch über einen eigenen Button als Navigationselement verfügbar ist. Auf so einer Startseite werden nach persönlichen Kriterien aktuelle Meldungen und Links präsentiert.

Die History-Liste zeigt für jede besuchte Seite eine Zeile in einer Tabelle an. Eine solche Zeile enthält den Titel der Seite, die URL, das Datum und die Uhrzeit des ersten Aufrufs, die Zeit seit dem letzten Aufruf, das Verfallsdatum (nach der einstellbaren Dauer der Aufbewahrung eines Links in der History) und die Anzahl der Aufrufe. Nach allen diesen Kriterien kann die Liste sortiert werden. Es können Suchanfragen zu den Daten in der Tabelle gestellt werden, deren Ergebnis wie ein Filter wirkt.



Abbildung 6: Microsoft Internet Explorer mit History-Liste

2.2.3 Microsoft Internet Explorer

Neben den Standardelementen zur Navigation kann der *Internet Explorer* eine Liste mit allen Links in der angezeigten Seite erzeugen (nur mit der Erweiterung „PowerToys“). Zwar können auch mit dem IE keine eigenen Anmerkungen zu den Seiten erstellt werden, aber es können Textstellen mit einem virtuellen Textmarker gekennzeichnet werden, was allerdings nicht abgespeichert wird. Durch die Integration in das Betriebssystem wird kein Unterschied zwischen lokalen und anderen Dateien im Internet gemacht, wodurch die Navigation vereinheitlicht wird. Der Browser mit seinen Elementen dient auch zur Bewegung im lokalen Dateisystem. Andererseits können sogenannte „Webordner“ angelegt werden, die einen transparenten Dateizugriff auf Webserver ermöglichen (erfordert die Unterstützung des jeweiligen Servers).

Die History-Liste kann in verschiedenen Arten dargestellt werden. In der Ansicht „Nach Site“ werden die einzelnen Seiten unter ihren Titeln in Ordnern zusammengefaßt, die nach dem Server-Namen in der URL benannt sind (z.B. würden alle besuchten Seiten, deren Adressen mit „http://www.tu-darmstadt.de/“ beginnen, in einem Ordner „tu-darmstadt“ zusammengefaßt). Bei der Ansicht „Nach Datum“ befinden sich diese Ordner

wiederum in Tagesordnern („Montag“, „Heute“, „Vorige Woche“). Reine Auflistungen aller besuchten Seitentitel zeigen die Ansichten „Nach der Anzahl der Zugriffe“ und „In der Zugriffsreihenfolge von heute“. Zu einzelnen Einträgen sind der Titel, die URL, das Datum des letzten Zugriffs und die Anzahl aller Zugriffe verfügbar. Eine Suchanfrage an die History sucht einen Begriff in den Titeln, den URLs und in den Inhalten der Seiten, allerdings nur wenn diese noch im Cache¹ vorhanden sind.

2.3 Konzepte zur Navigation im World Wide Web

2.3.1 World Wide Web als Hypertextsystem

Die schnelle Entwicklung des World Wide Web von einer Idee am *CERN* zum globalen Massenmedium ließ das eigentliche Hypertextsystem WWW sehr einfach ausfallen. Zwei Faktoren bestimmen dabei die Navigations- und Orientierungsunterstützung: Die Beschreibungssprache HTML und der Funktionsumfang des benutzten Browsers. Die Möglichkeiten von HTML bedingen die Gestaltung der Seiten im World Wide Web. Darüber hinaus gibt es aber keine verbindlichen Richtlinien für den Aufbau und die Bedienungselemente von Seiten. Auch verwendete Metaphern hängen von der jeweiligen Seite ab. Eine einheitliche Gestaltung und Bedienung, wie bei klassischen Hypertextsystemen meist gegeben, ist so nicht vorhanden. Die einzigen immer gleichen Navigationsmittel sind die vom Browser angebotenen. Da es aber verschiedene Browser gibt, können sich Seitenautoren und -autorinnen nicht auf deren Spezialfunktionen verlassen.

Das verteilte System ohne zentrale Verwaltung und mit ständigem Wachstum ermöglicht auch keine komplette Übersicht und kein vollständiges Inhaltsverzeichnis. Drei Suchebenen stehen jedoch trotzdem zur Verfügung. Sogenannte „Suchmaschinen“ erstellen eigene Indizes, die allerdings immer unvollständig sind. Darin kann eine thematische oder Wortsuche erfolgen. Weiterhin bieten die Autoren und Autorinnen von Seiten häufig eigene Suchmöglichkeiten und Übersichten ihrer Seiten an. Schließlich können die Browser Begriffe auf bereits übertragenen Seiten finden.

2.3.2 Nutzung der angebotenen Navigationsmittel

Schon in den ersten Jahren des WWW gab es Untersuchungen über die Benutzung von Browsern (Nielsen, 1994; Catledge, 1995; Jones, 1996; Tauscher, 1997). Grundlage für die Untersuchungen war der Browser *Mosaic*. Das Verhalten der Benutzer und Benutzerinnen wurde in drei Kategorien eingeteilt: Die direkte Suche nach einer bestimmten In-

¹ Der Cache ist ein temporärer Zwischenspeicher, in dem die Daten aller aufgerufenen Seiten geladen werden, so daß beim Backtracking die Seiten nicht noch mal vom Server übertragen werden müssen.

formation, eine themenbezogene Suche und das ziellose „Umherstreifen“ („Surfen“). Diese unterschiedlichen Strategien wurden aus der Häufigkeit der Zugriffe auf eine Seite und an den Pfadlängen, d.h., die Anzahl der Seiten bis zur gesuchten Information, ersichtlich. Als Schlußfolgerung für die Designer und Designerinnen der Seiten ergibt sich, daß das Angebot für die verschiedenen Strategien aufbereitet werden muß. Insbesondere sollte es möglichst wenig Zwischenschritte bis zur eigentlichen Information geben und jederzeit eine Übersicht des eigenen Angebots verfügbar sein.

Nielsens Studie von 1994 konzentrierte sich auf die Gestaltung der Seiten. Suchfunktionen innerhalb eines Seitenangebots sind danach bei den Benutzern und Benutzerinnen zwar beliebt, reichen aber nicht zur Navigation aus. Vorsichtig müsse mit Metaphern umgegangen werden, da sie in einem offenen System aus verschiedenen Seiten mißverstanden werden können. Dem offenen Charakter sollte eher durch externe Links zu anderen Seiten Rechnung getragen werden.

Bei der Untersuchung von 1995 nutzten die Testpersonen kaum die Lesezeichen und die History. Allerdings war beides in *Mosaic* sehr umständlich zu bedienen und existierte nur mit begrenztem Funktionsumfang. Das beliebteste Navigationsmittel der Browser (auch bei anderen Studien) ist der „Zurück“-Knopf, der eine gewisse Sicherheit beim Ausprobieren von Links gibt. Die eigentlichen Links im Dokument werden aber letztlich am häufigsten verwendet, weshalb einer sinnvollen Navigationsstruktur der Seiten eine hohe Bedeutung zukommt. Als weiteres Verhalten zeigte sich, daß bei zu langen Übertragungszeiten schnell der „Abbruch“-Knopf zur Anwendung kam. Daraus kann die Forderung nach sparsamem Einsatz umfangreicher Datenmengen auf einer Seite (z.B. große Bilder und Java-Applets) abgeleitet werden.

In der Studie von 1996 werden die Navigationsmöglichkeiten in Laden (über Link oder mit manueller URL-Eingabe), erneutes Aufrufen („Zurück“, History) und Neuladen („Aktualisieren“) klassifiziert. Diese Browser-Elemente harmonisieren oft nicht mit den eigenen Navigationselementen der Seiten. Ein „Zurück“-Knopf auf der Seite muß z.B. nicht die gleiche semantische Bedeutung wie der „Zurück“-Knopf des Browsers haben. Da WWW-Seiten nicht in einem geschlossenen System sind, fühlt sich die benutzende Person verwirrt, wenn ihr ständig verschiedene, völlig eigene Benutzungsoberflächen begegnen.

Die Studie von 1997 legte das Augenmerk mehr auf die Nutzung der History. Dabei zeigte sich, daß viele Seiten mehrfach besucht werden. Andererseits werden nur wenige Seiten regelmäßig aufgesucht. Die meisten erneut besuchten Seiten befinden sich unter den letzten zehn in der History (ohne Duplikate). Stark bedingt ist die Nutzung der Hi-

story durch den einfachen und schnellen Zugriff darauf. Problematisch scheint die Verwendung des „Stapel“-Modells in der History zu sein, bei der zuletzt aufgesuchte Seiten „unerwartet“ fehlen (mehr dazu im nächsten Abschnitt).

2.3.3 Verschiedene Modelle der History

Die History ist eine Verlaufsliste der zuletzt besuchten Knoten in einem Hypertextsystem. Dabei gibt es verschiedene Möglichkeiten, wie diese erstellt und präsentiert wird (Tauscher, 1997). *Mosaic* und die ersten Versionen der Browser von *Netscape* und *Microsoft* benutzten ein „Stapel“-Modell. Die URL der gerade besuchten Seite wird an eine Liste angefügt, auch wenn sie in dieser schon vorhanden ist. Ein Eintrag auf dieser so entstandenen Liste kann direkt angewählt werden, womit aber die Liste auch an jener Stelle weitergeführt wird. Dadurch verschwinden alle ursprünglich nachfolgenden Einträge. Eine Studie (Jones, 1996) verdeutlicht die Probleme des „Stapel“-Modells. Es wird von einer Seite A eine Seite B über einen Link aufgerufen. Mit dem „Zurück“-Knopf wird erneut die Seite A geladen, von der eine Seite C über eine Link aufgerufen wird. Aufgrund des „Stapel“-Modells kann nun mit dem „Zurück“-Knopf nicht mehr Seite B erreicht werden, was nicht dem mentalen Modell der meisten Anwender und Anwenderinnen entspricht.

Neuere Browser verwenden das „Stapel“-Modell noch immer für die schnelle Zugriffsliste, die für den „Vor“- und den „Zurück“-Knopf benutzt wird, speichern aber alle URLs mit Zahl der Besuche zusätzlich ab, was in einer sortierbaren Liste abgerufen werden kann. Als Sortierung ist die sequentielle Auflistung der zuletzt besuchten Knoten denkbar. Dabei kann noch unterschieden werden, ob mehrfache Vorkommen im ersten oder letzten Erscheinen gebündelt werden oder jeweils eigenständig aufgeführt sind. Die History kann auch nach der Häufigkeit der Besuche geordnet werden. Eine komplexere Sortierung stellt die hierarchische dar. Seiten, die durch Links miteinander verbunden sind, werden als zusammengehörig gebündelt. Auch nach dem Server bzw. der URL kann sortiert werden. Für einen sinnvollen Aufbau einer History muß die Liste auch über verschiedene Sitzungen, d.h., mehrere unterbrochene Benutzungen, weitergeführt werden.

Ein weiterer Aspekt der History ist die Wiedererkennbarkeit der jeweiligen Einträge. Ein solcher besteht in der Regel aus dem Titel und/oder der URL. Der Titel wird im HTML-Dokument definiert und sollte möglichst so eindeutig sein, daß davon schnell auf den Inhalt der Seite geschlossen werden kann. Modelle mit graphischen Repräsentationen der Seiten (verkleinert) in der History sind meist wenig effektiv, da sich viele Seiten optisch nicht signifikant unterscheiden.

Häufig werden verschiedene Seiten über Links von einer bestimmten Seite aufgerufen. Dazu muß diese Links-Liste schnell wieder erreichbar sein, doch der Aufwand diese Seite in der History zu finden ist oft zu hoch. Wird das Backtracking über den „Zurück“-Knopf genutzt, kann die fragliche Seite durch das „Stapel“-Modell schon gar nicht mehr erreichbar sein.

2.3.4 Graphische Übersichten

Aufgrund seiner Größe und des offenen Charakters läßt sich vom ganzen Hypertextsystem World Wide Web keine graphische Übersicht schaffen. Es gibt aber Überlegungen, zumindest die besuchten Knoten, wie sie etwa in der History gespeichert sind, oder die Strukturen von einzelnen Seitenangeboten zu visualisieren.

WebMap ist ein separates Programm, das sich vom Browser (in dem Fall *Mosaic*) den aktuellen Knoten holt (Dömel, 1994). In der Visualisierung werden der aktuelle Knoten und alle (aus den Links ersichtlichen) Kinderknoten mit verbundenen Ovalen gezeigt. Verschiedene Farben der Verbindungen weisen auf unterschiedliche Semantiken hin (z.B. Serverwechsel). Jeder Knoten erscheint auch bei mehrfachem Aufruf nur einmal im Diagramm. Aus dem entstehenden Graph wird ein aufspannender Baum gebildet. Eine graphische Gruppierung einzelner Knoten wäre über zusätzliche Metainformationen im HTML-Code denkbar (z.B. mit der schon existierenden „LINK“-Anweisung).

Einen anderen Ansatz wählt die *Graphische History* (Ayers, 1995). Als Erweiterung der Browser-Funktionen (ein modifiziertes *Mosaic*) wird eine angelegte History visualisiert. Auch hier wird ein Baum erstellt, zu dem aber nur der aktuelle Knoten hinzugefügt wird. Dabei ist jeder Knoten als verkleinertes Bild der Seite dargestellt. Zur Übersicht können Teile des Baums zu Symbolen verkleinert werden. Ein Knoten kann mehrfach auftauchen. Der so entstandene History-Baum kann abgespeichert und später wieder eingeladen werden.

Völlig eigenständige Wege beschreitet *CyberMap* (Gloor, 1997). Dabei soll aus einer Textmenge (nicht notwendigerweise Hypertext) eine graphische Darstellung generiert werden, die auf den Gewohnheiten und Wünschen der benutzenden Person beruht. Der Text wird nach inhaltlichen Aspekten in Bereiche unterteilt (hier „Hyperdrawer“ genannt). Innerhalb dieser Bereiche werden Knoten gebildet, die nach semantischen Gesichtspunkten über Links verknüpft sind, die grundsätzlich nichts mit schon vorhandenen Links zu tun haben. Die generierte Aufteilung der Bereiche und Knoten erfolgt mit externen Informationen über die Thematik, Benutzungsprofilen und Analyse der ausgewählten Links. Die WWW-Version von *CyberMap* basiert auf Java für die Darstellung

und die Indexierung. Das eigentliche Diagramm kann in zahlreichen Varianten präsentiert werden (u.a. mit 3D-Objekten).

Die Darstellung aller Kinderknoten bei *WebMap* dürfte das Diagramm schnell unübersichtlich machen. Häufig stehen auf aktuellen Seiten zahlreiche Verweise zu anderen, was auch mit der unüberschaubar gewordenen Größe des WWW zu tun hat. Zudem ist die Form des aufspannenden Baums nicht unbedingt intuitiv. Die verkleinerten Bilder der Seiten von der *Graphischen History* sind in der Praxis oft wenig aussagekräftig. Allerdings kann ein Bild als optische Markierung für eine interessante Links-Sammlung diese in der History schneller auffindbar werden lassen. Fraglich bleibt, ob eine solche Baumdarstellung tatsächlich Vorteile gegenüber einer sortierten History-Liste hat. Am ambitioniertesten ist sicherlich *CyberMap*. Für ein korrektes Funktionieren müssen aber entsprechende Metainformationen der einzelnen Seiten zur Verfügung stehen und eine sinnvolle Indexierung benötigt fortgeschrittene Methoden der künstlichen Intelligenz.

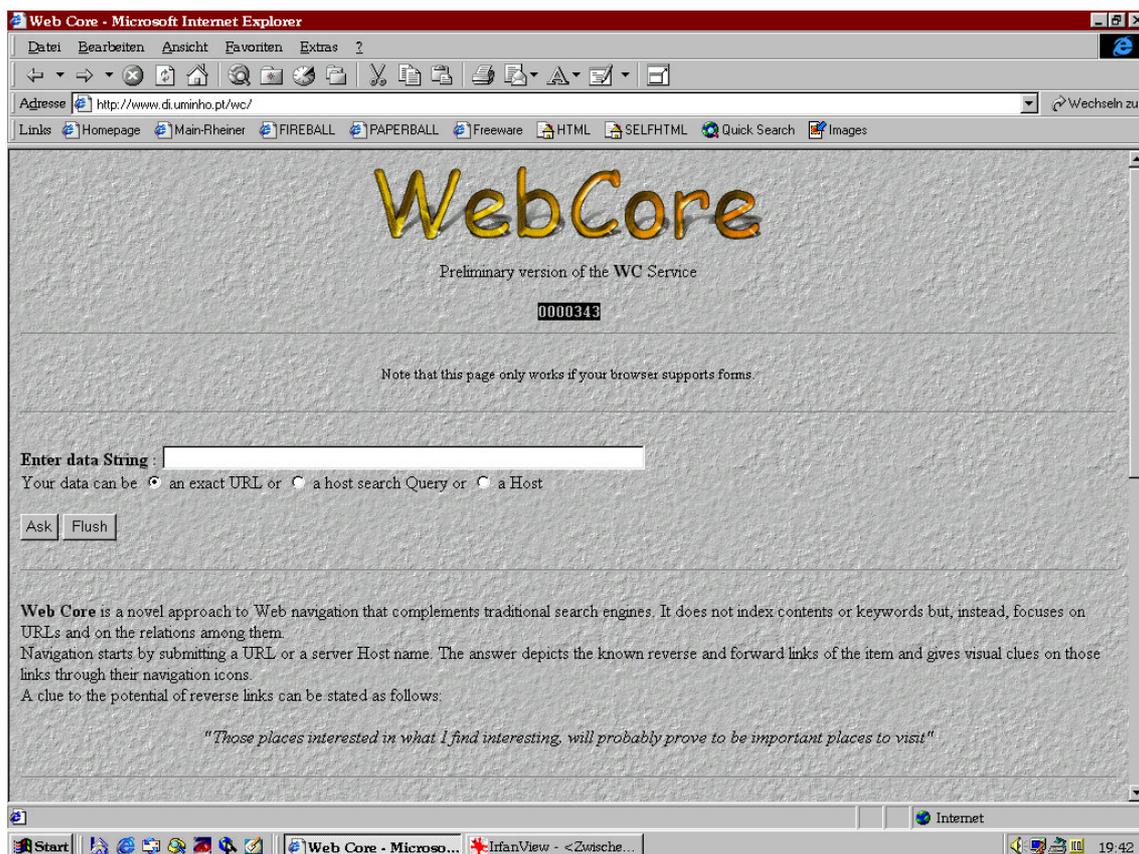


Abbildung 7: Web Core

2.3.5 Reverse Links als Navigationshilfe

Jedes neue Dokument im WWW kann Links auf beliebige schon bestehende Dokumente enthalten. Für die anderen Dokumente ist ein solcher Link nicht bekannt, da es keine

zentrale Verwaltung der Verweisstruktur des WWW gibt. *Reverse Links*, also Verweise auf andere Seiten, die Links zur aktuellen Seite enthalten, können deshalb weder vom Browser noch von der Seite selbst angeboten werden. Trotzdem wären Reverse Links bei der Navigation sehr hilfreich.

Eine Nutzung Reverser Links ermöglicht das Projekt *Web Core* (Baquero, 1996). *Web Core* ist eine eigene Suchmaschine, die zum Aufbau des Datenbestands genauso wie herkömmliche Suchmaschinen das Web mit sogenannten „Web Bots“ (automatische Programme, die nach neuen Seiten im WWW fahnden) durchsucht. Dabei werden auch alle Links auf den jeweiligen Seiten ermittelt, so daß das System daraus Reverse Links erstellen kann (natürlich nur innerhalb des Datenbestands). Eine Anfrage an *Web Core* liefert zu jeder URL eine Liste aller Links und Reversen Links. Gleichzeitig werden die Links mit farbigen Symbolen nach Erreichbarkeit und Typ (HTML-Dokument oder nicht) klassifiziert. Zu zwei Servern können alle Links auf deren Seiten zwischen beiden angezeigt werden.

Wie bei allen Suchmaschinen hängt auch bei *Web Core* der Nutzen stark vom Datenbestand ab. Gerade bei Reversen Links wäre eine annähernd komplette Erfassung aller Seiten nötig. Daher sollte das Konzept von *Web Core* besser in eine der bestehenden, umfangreichen Suchmaschinen (z.B. *Alta Vista* oder *Fireball*) übernommen werden.

2.3.6 Textbasierte Metaphern

Ein sehr eigenwilliges Konzept verfolgt das System *Juggler* (Dieberger, 1996). Dieses nutzt MUDs (Multi-User-Dungeons) bzw. MOOs (MUD objektorientiert) als Navigationshilfe im WWW.

Ein MUD ist ein Netzwerkspiel, bei dem durch Text beschriebene virtuelle Welten erforscht werden, was wiederum über textuelle Kommandos erfolgt (z.B. „look at book“). In dieser Welt kann auch mit anderen MUD-Spielern und -Spielerinnen im Netz kommuniziert werden (z.B. „say to Merlin 'nice collection“). Die virtuelle Welt ist in Text-Knoten unterteilt, zwischen denen man sich mittels Kommandos einer Lauf-Metapher bewegt (z.B. „n“ für „gehe nach Norden“).

Die Idee von *Juggler* ist, die MUD-Knoten mit URLs zu koppeln, deren zugehörigen Seiten im separaten Browser aufgerufen werden. Das MUD-System ermöglicht die Verwendung jeder mit Text beschreibbaren Metapher. Im wesentlichen lassen sich so interaktiv geführte Touren realisieren, die das Hypertextkonzept hinter dem MUD-Konzept zurückstellen. Zudem läßt sich Kommunikation mit andern Netzteilnehmern und -neh-

merinnen über bestimmte WWW-Seiten führen, indem eine URL zum andern Rechner übermittelt wird.

Zwar kann *Juggler* einen anderen Zugang zum WWW vermitteln und für bestimmte Anwendungen sinnvolle Metaphern ermöglichen, aber es wird sich wohl kaum als allgemeines Navigationsmittel durchsetzen. Es erfordert einen erheblichen Aufwand, eine entsprechende Umgebung zu verfassen. Hauptproblem ist aber, daß heute graphische Oberflächen Standard sind und eine reine Texteingabe wohl nicht mehr akzeptiert wird.

2.3.7 Navigation in speziellen Anwendungen

Das World Wide Web bietet inzwischen nicht nur Informationsseiten und Texte, sondern auch interaktive Anwendungen. Hierbei besteht ein Unterschied der Gestaltung zu klassischen Hypertextsystemen. Besonders sogenannte *Shopping-Systeme* zum Einkaufen über das WWW erfreuen sich zunehmender Beliebtheit (Shubin, 1997).

Ein Shopping-System stellt ein in sich geschlossenes System dar, das dementsprechend einheitlich und konsistent gestaltet sein muß. Externe Links auf andere Seiten sollten nur in wohldefinierten Bereichen auftreten und nicht auf Formularseiten. Das steht deutlich im Gegensatz zu Informationstexten, die gerade durch die transparente Verknüpfung mit anderen Texten an Wert gewinnen. Bei speziellen Anwendungen tritt das Problem der Diskrepanz von Browser-Navigation und Seitennavigation besonders hervor. Da der Kontext nur durch die angezeigte Seite gegeben ist, muß jede Seite entsprechende Informationen enthalten, auch Fehlermeldungen. Oft kommen bei Shopping-Systemen Metaphern zum Einsatz (Warenkorb), aber eine sinnvolle Nutzung der WWW-Möglichkeiten kann dadurch auch erschwert werden.

Aus Sicherheitsgründen entfernen sich viele WWW-Anwendungen von der HTML-Oberfläche und stellen eine eigene zu Verfügung, die als Java-Applet realisiert ist. Ein Java-Applet ist ein Programm, das erst zum Client übertragen und dann dort gestartet wird. Der Browser wird hierbei nur noch als Rahmen für das Programm benutzt. Verbunden damit sind allerdings hohe Ladezeiten, da das Applet und die Daten übertragen werden müssen. Fraglich bleibt, ob solche „halben“ Programme (Java-Applets haben aus Sicherheitsgründen nicht die gleichen Möglichkeiten wie eigenständige Programme) wirklich der beste Weg sind. Vom Browser unabhängige Internet-Programme könnten für manche Anwendungen die bessere Lösung sein (Nielsen, 1998).

2.3.8 Ideen für die Entwicklung des WWW

Viele Konzepte und Ansätze zur Navigation in Hypertextsystemen könnten in zukünftigen Weiterentwicklungen des World Wide Web umgesetzt werden (Bieber, 1997). Dabei könnte es eine „evolutionäre“ Entwicklung der Knoten zu abstrakten Datentypen geben, vergleichbar zum objektorientierten Paradigma in der Programmentwicklung.

Typisierte Knoten und Links ermöglichen eine Kategorisierung. Link-Attribute könnten für eine strukturbasierte Suche Schlüsselwörter und Zeitstempel enthalten. Teile des Knoteninhalts, die aus anderen Knoten stammen, könnten automatisch oder als Hinweis aktualisiert werden. Anmerkungen und Links würden als persönlich oder öffentlich gekennzeichnet und mit entsprechenden Sicherheitsmechanismen versehen. Mit externen Link-Datenbanken könnte die Aktualisierung einfacher vorgenommen werden. Aus dem persönlichen Nutzungsverhalten könnten Link-Datenbanken nach den eigenen Vorlieben generiert werden.

Manche dieser Ideen sind mit den Metainformationen, die in HTML angegeben werden können, realisierbar. Allerdings werden diese momentan von den Browsern kaum interpretiert. Beschreibungen zu einem Link können in HTML 4.0 mit dem „TITEL“-Attribut festgelegt werden. Diese werden dann vom Browser beim Überfahren des Links mit der Maus als kleiner Kasten eingeblendet („Tool-Tip“). Die wesentlich universellere *Extensible Markup Language* (XML) bietet typisierte Links und vieles mehr (Münz, 1998). XML ist eine Beschreibungssprache, die weit mächtiger als HTML ist. Zukünftig wird sie im WWW Bedeutung gewinnen, zumal der *Internet Explorer* Version 5.0 sie unterstützt, aber wohl HTML nicht gänzlich ablösen.

Zu den klassischen Konzepten zur Navigation in Hypertextsystemen gehören lokale und globale Übersichten, Pfade und geführte Touren und verschiedene Backtracking-Strategien (auch aufgabenorientiert). Einiges davon wurde oben untersucht. Lokale Übersichten und Pfade bzw. Touren (empfohlene oder erzwungene Wege) können vom Autor bzw. der Autorin einzelner Seiten mit den zur Verfügung stehenden Mitteln realisiert werden. Die sehr offene Struktur des WWW ermöglicht aber dafür keine verbindlichen Richtlinien.

Eine Stärke des WWW ist die Möglichkeit für jeden Benutzer und jede Benutzerin, auch selber Anbieter bzw. Anbieterin von Seiten zu werden. Andererseits läßt sich der sinnvolle Einsatz der komplexer werdenden Möglichkeiten von HTML von „Amateuren“ kaum noch bewältigen. Außerdem wird so die Entwicklung von konsistenter und übersichtlicher Navigation erschwert. Daher wird es wohl zukünftig eine stärkere Trennung

von Benutzern und Benutzerinnen zu professionellen Autoren und Autorinnen geben, wie sie auch bei der Programmentwicklung stattfand.

3. AlterNav

AlterNav ist ein alternatives Navigationswerkzeug, das das Verwalten mehrerer Histories für ein Seitenangebot ermöglicht. Es ist in HTML und JavaScript implementiert und kann ohne Anpassung anderen HTML-Seiten hinzugefügt werden.



Abbildung 8: AlterNav im Internet Explorer

3.1 Idee

Im wesentlichen unterstützen bei aktuellen Browsern zwei Konzepte die Navigation, die History und die Lesezeichen. Die History spiegelt den Weg zur aktuellen Seite wider, aber nur einen Pfad. Lesezeichen speichern einzelne Seiten (bzw. deren Adresse) ab, ohne dabei den Kontext und den Weg dorthin festzuhalten.

AlterNav versucht, eine Brücke zwischen History und Lesezeichen zu schlagen. Es wird eine vom Browser unabhängige History geführt, die jederzeit in Speicherplätze abgelegt werden kann. Diese stellen somit eine Art Lesezeichen mit erhaltenem Kontext dar. Dabei soll es übersichtlich bleiben und möglichst direkt zugreifbar. Die Länge der einzelnen Histories ist auf zehn Einträge beschränkt und die Anzahl der Speicherplätze beträgt

vier. Das hängt einerseits mit der Realisierung der Speicherplätze als Cookies¹ und den damit einhergehenden Größenbeschränkungen zusammen. Andererseits hat eine Untersuchung (Tauscher, 1997) gezeigt, daß die weitaus meisten History-Zugriffe auf die letzten zehn Einträge erfolgen.

AlterNav ist nicht als Zusatzprogramm zum Browser konzipiert, sondern als universelle Ergänzung (durch Implementierung mit HTML und JavaScript) zu einem bestehenden Seitenangebot, wo es die interne Navigation unterstützt.

Bei der optischen Gestaltung stand die Übersichtlichkeit und die Ausnutzung des beschränkten Platzes im Vordergrund. Einerseits sollen die Elemente von *AlterNav* komplett zu sehen sein, andererseits steht die aktuelle WWW-Seite natürlich im Mittelpunkt der Anzeige. Um Platz zu sparen, mußten für die Knöpfe zur Bedienung Abkürzungen gewählt werden. Diese orientieren sich an der Benutzung von Taschenrechnern (Wert in Speicherplatz ablegen oder löschen).

3.2 Funktionsweise

Nach dem Aufruf von *AlterNav* (über die Datei `altindex.htm`) wird der Anzeigebereich im Fenster des verwendeten Browsers mit Frames² in zwei Hälften unterteilt. Im oberen Teil befindet sich der Anzeigebereich von *AlterNav*, im unteren die Startseite des betrachteten Seitenangebots (häufig `index.htm`).

Auf der linken Seite von *AlterNav* ist eine Auswahlliste mit zehn Einträgen, von denen der erste den Titel der Startseite anzeigt und die restlichen leer sind. Dies ist die History von *AlterNav*. Wird nun ein Link auf der Startseite aktiviert, wird diese History automatisch um den Titel des neuen Dokuments im unteren Frame ergänzt. Mit dem „C“-Knopf (für engl. „clear“, „löschen“) links daneben kann die History jederzeit wieder gelöscht werden. Der „U“-Knopf (oder nach Aktivierung der „T“-Knopf) schaltet die Anzeige der History-Einträge von den Titeln auf die zugehörigen URLs (Adressen) um. Es gibt keine doppelten Einträge in der History, jede Seite wird nur mit dem letzten Aufruf eingetragen.

1 Cookies ermöglichen es einer HTML-Seite (mit JavaScript), eine Zeichenkette beim Anwender abzuspeichern, die später von dieser Seite (und nur von dieser) wieder abgefragt werden kann.

2 Frames sind eine Möglichkeit von HTML 4.0, mehrere HTML-Dokumente in einem Fenster darzustellen. Die Browser von *Netscape* kennen diese schon seit Version 2.

Rechts wird die History des aktuellen Speicherplatzes angezeigt. Mit den Knöpfen „1“ bis „4“ kann ein anderer Speicherplatz aufgerufen werden. Der Knopf „S“ (für engl. „store“, „speichern“) legt die History von *AlterNav* in diesem Speicher ab. Im Feld darüber kann ein individueller Name für die History vergeben werden. Der aktuelle Speicherplatz wird mit dem mittleren „C“-Knopf gelöscht.

Ein Klick auf einen Eintrag in der History von *AlterNav* oder in der History des aktuellen Speicherplatzes führt zum Aufruf der dazugehörigen Seite. Die vier Histories in den Speicherplätzen werden zusammen mit den vergebenen Namen sofort als Cookie abgespeichert. Damit stehen sie auch bei einem späteren Aufruf noch zur Verfügung.

3.3 Implementierung

AlterNav ist mit HTML und JavaScript implementiert. Dabei wurde es mit den aktuellen Versionen der Browser von *Netscape* und *Microsoft* getestet. Durch den Verzicht auf spezifische Befehle von HTML 4.0, funktioniert es auch mit der veralteten Version 3 des Browsers von *Netscape*. Eine JavaScript-Implementierung ermöglicht geringe Übertragungszeiten und erhöht die Transparenz für den Benutzer bzw. die Benutzerin, da jederzeit der Quellcode eingesehen werden kann. Java bietet diese Vorteile nicht, kann aber flexiblere Funktionen ermöglichen. Weitere Alternativen zur Implementierung wären ein externes Programm, das mit dem benutzten Browser kommuniziert, und komplett neue Browser-Entwicklungen auf Basis von vorhandenem Quellcode (*Netscape*) oder mit Programmintegration (*Internet Explorer*). Das brächte allerdings Plattform- und Browser-Abhängigkeit mit sich. Zudem ist *AlterNav* als Funktionsergänzung und nicht als eigenständige Navigationseinheit konzipiert.

Zu *AlterNav* gehören die Dateien `altindex.htm`, `alternav.htm` (siehe Anhang), `alternavbig.htm` und `altman.htm`. Die Datei `altindex.htm` baut den Frameset auf und lädt in den oberen Frame `alternav.htm` und in den unteren die gewünschte Startseite (vorgegeben ist `index.htm` im selben Verzeichnis). Bei entsprechend hoher Bildschirmauflösung (und einem Browser ab der Version 4) ersetzt `alternavbig.htm` automatisch `alternav.htm`. Außer der besseren Ausnutzung einer Bildschirmbreite von 1024 Bildpunkten und mehr, ist `alternavbig.htm` mit `alternav.htm` identisch. Ein Klick auf den Namen „AlterNav“ ruft eine kurze Bedienungsanleitung auf (`altman.htm`).

Zur internen Speicherung der Histories werden zweidimensionale Felder für die Titel und die URLs angelegt. Ein weiteres Feld nimmt die selbstvergebenen Namen der Hi-

story-Speicher auf. Als Hilfsfunktion zur Zeichenersetzung in Zeichenketten muß `MyReplace()` definiert werden, da eine entsprechende Standard-Funktion erst ab HTML 4.0 existiert.

Nach dem Aufbau als HTML-Tabelle und der JavaScript-Initialisierung wird mit `setTimeout` jede Sekunde einmal die Funktion `TestChange()` aufgerufen. Diese testet, ob sich der Titel oder die URL des Dokuments im unteren Frame geändert hat. In diesem Fall wird der neue Titel in die History von *AlterNav* aufgenommen. Existiert der Titel darin schon, verändert sich nur die Reihenfolge. Die globale Variable `store` bezeichnet die Nummer des aktuellen Speichers und `display` gibt an, ob die Anzeige der Einträge als Titel oder URL erfolgt.

Die Funktion `StoreCookie()` wird immer dann aufgerufen, wenn die gespeicherten Histories oder deren Namen verändert wurden. Als Cookie wird eine Zeichenkette aus allen Einträgen der Histories (Titel und URL) und den Namen gebildet. Trennzeichen dabei ist `§%`. Da eine Semikolon das Ende eines Cookies definiert, wird dieses Zeichen intern durch `§&` ersetzt. Das Cookie-Verfallsdatum wird auf das aktuelle Datum plus ein Jahr gesetzt.

3.4 Einschränkungen

Die Implementierung mit HTML und JavaScript bringt leider einige Einschränkungen mit sich. Da aus Sicherheitsgründen mit JavaScript nicht auf die Daten von anderen Dokumenten in einem anderen Frame zugegriffen werden darf (z.B. Lesen von Titel und URL), kann *AlterNav* nur innerhalb einer Domäne funktionieren. Es bietet sich dadurch als Ergänzung für die Navigation in einem umfangreichen Seitenangebot oder für die Nutzung in einem Intranet an.

Einem Seitenangebot kann die Funktionalität von *AlterNav* problemlos durch Kopieren der entsprechenden Dateien auf den Server hinzugefügt werden. Allerdings müssen die Seiten, die mit *AlterNav* benutzt werden sollen, bestimmte Kriterien erfüllen. Externe Links führen zu den beschriebenen Sicherheitsproblemen und erzeugen deshalb Fehlermeldungen. Das Problem kann dadurch umgangen werden, daß alle externen Links in einem separaten Fenster angezeigt werden. Mit dem HTML-Attribut „`target='name'`“ wird ein solches Fenster geöffnet. Ist dabei „`name`“ immer gleich, wird auch immer dasselbe Fenster benutzt. Die Darstellung in einem Frame bedingt, daß als „`target`“ für Links niemals „`_top`“ angegeben werden darf, wodurch der Frame nämlich „gesprengt“ würde. Bestehen die Seiten selber aus Frames, bekommt *AlterNav* Änderungen in Un-

terframes nicht mit. Deshalb sollten in dem Fall für jede Unterseite der Frameset mit anderem Titel neu aufgebaut werden. Als Nebeneffekt können damit auch wieder problemlos normale Lesezeichen auf solch eine Seite gesetzt werden.

Aufgrund eines Programmfehlers der *UNIX*-Versionen des *Netscape*-Browsers funktioniert *AlterNav* damit nicht korrekt. Dort wird nämlich für Dokumente in Frames immer eine leere Zeichenkette als Titel geliefert.

3.5 Mögliche Weiterentwicklungen

Die Haupteinschränkung liegt in der ausschließlichen Anwendung innerhalb einer Domäne. Der Grund dafür ist die Implementierung mit JavaScript. Eine Version von *AlterNav* in Java könnte diesen Nachteil umgehen.

Die Anzahl der Histories und deren Einträge wurden wegen der Übersichtlichkeit und Implementierung mit Cookies beschränkt. Eine mögliche Java-Implementierung ohne Cookies könnte diese Beschränkung aufheben, was dann allerdings über die Anwendung als Lesezeichen mit Kontext hinausginge und Fragen der praktischen Benutzbarkeit aufwürfe.

Höhere Bildschirmauflösungen können eine überschaubarere Darstellung und selbst-erklärende Buttons ermöglichen.

Letztlich wäre aber eine Integration der Funktionalität von *AlterNav* in die bestehenden Browser-Konzepte am wünschenswertesten.

Zusammenfassung und Ausblick

Das World Wide Web ist zum bedeutendsten Hypertextsystem geworden. Bekannte Probleme bei der Navigation und Orientierung in Hypertextsystemen bekommen im World Wide Web neue Dimensionen. Manche der klassischen Lösungsansätze sind deshalb nicht einfach im WWW umsetzbar. Obwohl die Browser als WWW-Werkzeuge immer bessere Unterstützung bei der Navigation bieten, sind viele Möglichkeiten noch nicht realisiert.

Mit *AlterNav* wird eine solche Möglichkeit vorgestellt. Abspeicherbare Histories bewahren den Kontext einer besuchten Seite und ermöglichen das spätere Nachvollziehen verschiedener Pfade. Die Implementierung mit HTML und JavaScript ermöglicht einen universellen Einsatz innerhalb eines Seitenangebots.

Manches, wie typisierte Links, erfordert eine grundlegende Änderung von HTML. Verbesserte Lesezeichen und vereinfachte Histories können aber auch in neueren Versionen der Browser realisiert werden. Fortschritte bei HTML sind durch den Gegensatz des WWW-Konsortiums auf der einen Seite und dem Wettstreit um die Führungsstellung bei Browsern zwischen *Netscape* und *Microsoft* auf der anderen Seite bedingt. Die Hypertextforschung darf sich nicht von der WWW-Entwicklung abkoppeln, zumal sich HTML auch bei Hypertextsystemen, die unabhängig vom WWW sind, durchzusetzen scheint. Damit entsteht ein Hypertext-Standard, der die Benutzbarkeit von Hypertext-Datenbeständen auf absehbare Zeit sicherstellen könnte.

Literaturverzeichnis

- Ayers, E. Z., J. T. Stasko (1995): Using Graphic History in Browsing the World Wide Web. **Fourth International World Wide Web Conference Proceedings, World Wide Web Journal**, S. 451-459.
- Baquero, C., J. P. Andrade (1996): Web Core - Forward and Backward Navigation on a Multi-grain Web Representation. **Poster Proceedings: Fifth International World Wide Web Conference**, S. 59-69.
- Bieber, M., F. Vitali, H. Ashman, V. Balasubramanian, H. Oinas-Kukkonen (1997): Fourth generation hypermedia: some missing links for the World Wide Web. **Int. J. Human-Computer Studies**, **47**, S. 31-65.
- Bush, V. (1945): **As We May Think**.
<http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm>
- Cailliau, R. (1995): **A Little History of the World Wide Web**.
<http://www.w3.org/History.html>.
- Catledge, L., J. Pitkow (1995): Characterizing browsing strategies in the World Wide Web. **Computer Networks and ISDN Systems**, **27**, S. 1065-1073.
- CERN (1997): **History and growth**.
<http://www.cern.ch/Public/ACHIEVEMENTS/WEB/history.html>
- Conklin, J. (1987): Hypertext: An Introduction and Survey. **IEEE Computer**, **20**, 9, S. 17-41
- Dieberger, A. (1996): Browsing the WWW by interacting with a textual virtual environment – A framework for experimenting with navigational metaphors. **ACM Hypertext '96**, S. 170-179.
- Dömel, P. (1994): Webmap - A Graphical Hypertext Navigation Tool. **The Second International WWW Conference Fall '94 in Chicago**, S. 785-798.
- Gloor, P. A., S. B. C. Dynes (1997): Cybermap - Visually Navigating the Web. **Journal of Visual Languages and Computing**, **9**, S. 319-336.
- Herold, W., M. Müller (1993): Ein Stufenindex als Navigationshilfe für Hypertextdokumente. **Hypermedia, Proceedings der internationalen Hypermedia '93 Konferenz**, Springer-Verlag, Berlin, S. 150-159.
- Jones, S., A. Cockburn (1996): A Study of Navigational Support Provided by Two World Wide Web Browsing Applications. **ACM Hypertext '96**, S. 161-169.

- Münz, S. (1998): **SELFHTML: Version 7.0**. <http://www.teamone.de/selfhtml/>.
- Nielsen, J. (1990): The Art of Navigating through Hypertext. **Communications of the ACM**, **33**, 3, S. 296-310.
- Nielsen, J. (1994): **Report From a 1994 Web Usability Study**.
http://www.useit.com/papers/1994_web_usability_report.html
- Nielsen, J. (1995): **Multimedia & Hypertext: the Internet and Beyond**.
Academic Press Professional, Boston.
- Nielsen, J. (1998): Does Internet = Web?. **Alertbox for September 20, 1998**,
<http://www.useit.com/alertbox/980920.html>
- Purgathofer, P., T. Grechenig (1993): Navigation in hypertext by browsing in survey objects. **Hypermedia, Proceedings der internationalen Hypermedia '93 Konferenz**, Springer-Verlag, Berlin, S. 116-129.
- Shubin, H., M. M. Meehan (1997): Navigation in Web Applications.
ACM interactions, **11/1997**, S. 13-17.
- Tauscher, L., S. Greenberg (1997): How people revisit web pages: empirical findings and implications for the design of the history systems.
Int. J. Human-Computer Studies, **47**, S. 97-137.
- Ventura, A. (1990): Benutzerorientierter Entwurf von Hypertexten.
Hypertext und Hypermedia, Informatik-Fachberichte 249,
Springer-Verlag, Berlin, S. 220-229.

Anhang: Quellcode von AlterNav

altindex.htm

```
<html>
<head>
<!-- AlterNav by Tobias Steinke, 1999 -->
<!-- Frameset for alternav.htm -->
<meta name="author" content="Tobias Steinke">
<meta name="created" content="1999">
<meta name="keywords" content="AlterNav, navigation, history">
<title>AlterNav</title>
</head>
<frameset rows="80,*">
  <frame src="alternav.htm" name="AlterNavUp" marginwidth=5 marginheight=5>
  <frame src="index.htm" name="AlterNavDown">
  <noframes>
    <p>Frames only!</p>
  </noframes>
</frameset>
</html>
```

alternav.htm

```
<html>
<head>
<!-- AlterNav by Tobias Steinke, 1999 -->
<!-- Called by altindex.htm -->
<title>AlterNav</title>
<script language="JavaScript">
<!--
if(navigator.appVersion.substring(0,1)>="4")
  if(screen.availWidth>=1024)
    location.replace("alternavbig.htm");
var i, j, store=1, display=0;
var titles=new Array(5);
var urls=new Array(5);
var names=new Array(5);
for(i=0;i<5;i++)
{ titles[i]=new Array(11);
  urls[i]=new Array(11);
  names[i]="";
}
for(i=0;i<5;i++)
  for(j=0;j<11;j++)
    { titles[i][j]=" ";
      urls[i][j]=" ";
    }
for(i=1;i<5;i++)
{ titles[i][0]="History "+i;
  urls[i][0]="History "+i;
}

function InitAlterNav()
{ var i=0, j, k, c;
  if(document.cookie)
  { c=document.cookie.split("$%");
    if(c[i++]=="AlterNav=")
    { for(j=1;j<5;j++)
      { names[j]=MyReplace(c[i++], "$&", ",");
```

```

        for(k=1;k<11;k++)
        { titles[j][k]=MyReplace(c[i++],"$&","");
          urls[j][k]=MyReplace(c[i++],"$&","");
        }
    }
    document.history.myname.value=names[1];
    for(j=1;j<11;j++)
        document.history.historystore.options[j].text=titles[1][j];
}
}
document.history.historystore.options[0].text=titles[1][0];
TestChange();
}

function TestChange()
{ var i,j;
  var
t=MyReplace(MyReplace(parent.AlterNavDown.document.title,"$%",""),"$&","");
  var
u=MyReplace(MyReplace(parent.AlterNavDown.location.href,"$%",""),"$&","");
  if(t!=titles[0][0] || u!=urls[0][0])
  { for(i=10;i>0;i--)
    { if(i<10)
      { if(display==0)
        document.history.list.options[i].text=titles[0][i-1];
        else
        document.history.list.options[i].text=urls[0][i-1];
      }
      titles[0][i]=titles[0][i-1];
      urls[0][i]=urls[0][i-1];
      if(titles[0][i]==t && urls[0][i]==u)
      { for(j=i;j<10;j++)
        { if(display==0)
          document.history.list.options[j].text=titles[0][j+1];
          else
          document.history.list.options[j].text=urls[0][j+1];
          titles[0][j]=titles[0][j+1];
          urls[0][j]=urls[0][j+1];
        }
      }
    }
    if(display==0)
        document.history.list.options[0].text=t;
    else
        document.history.list.options[0].text=u;
    titles[0][0]=t;
    urls[0][0]=u;
  }
  setTimeout("TestChange()",1000);
}

function Go(store0,index)
{ if(urls[store0][index]!=" " && index>0)
  parent.AlterNavDown.location.href=urls[store0][index];
  if(store0==0)
    document.history.list.selectedIndex=0;
  else
    document.history.historystore.selectedIndex=0;
}

function ClearHistory(index)
{ var i;
  for(i=1;i<11;i++)
  { titles[index][i]=" ";
    urls[index][i]=" ";
    if(index==0 && i<10)
        document.history.list.options[i].text=" ";
    if(index>0)

```

```

        document.history.historystore.options[i].text=" ";
    }
    if(index>0)
        StoreCookie();
}

function StoreCookie()
{
    var now=new Date();
    var inayear=new Date(now.getTime()+1000*60*60*24*365);
    var i,j,c="AlterNav=$%";
    for(i=1;i<5;i++)
    {
        c+=names[i]+"$%";
        for(j=1;j<11;j++)
        {
            c+=titles[i][j]+"$%";
            c+=urls[i][j]+"$%";
        }
    }
    document.cookie=MyReplace(c,";", "$&")+"; expires="+inayear.toGMTString()+" ";
}

function StoreName()
{
    names[store]=MyReplace(MyReplace(document.history.myname.value,"$%", ""), "$&","");
    document.history.myname.value=names[store];
    StoreCookie();
}

function Store()
{
    var i;
    for(i=0;i<10;i++)
    {
        titles[store][i+1]=titles[0][i];
        urls[store][i+1]=urls[0][i];
        if(display==0)
            document.history.historystore.options[i+1].text=titles[0][i];
        else
            document.history.historystore.options[i+1].text=urls[0][i];
    }
    StoreCookie();
}

function Call(index)
{
    var i;
    store=index;
    document.history.myname.value=names[store];
    for(i=0;i<11;i++)
    {
        if(display==0)
            document.history.historystore.options[i].text=titles[store][i];
        else
            document.history.historystore.options[i].text=urls[store][i];
    }
}

function ChangeDisplay()
{
    var i;
    display=!display;
    if(display==0)
    {
        document.history.tubutton.value=" U ";
        if(document.all)
            document.all.tutitle.title=" Show URLs ";
        for(i=0;i<10;i++)
        {
            document.history.list.options[i].text=titles[0][i];
            document.history.historystore.options[i+1].text=titles[store][i+1];
        }
    }
    else
    {
        document.history.tubutton.value=" T ";
    }
}

```

```

        if(document.all)
            document.all.tutitle.title=" Show titles ";
        for(i=0;i<10;i++)
        { document.history.list.options[i].text=urls[0][i];
          document.history.historystore.options[i+1].text=urls[store][i+1];
        }
    }
}

function MyReplace(s,c1,c2)
{ var r="",a,i;
  a=s.split(c1);
  for(i=0;i<a.length;i++)
  { r+=a[i];
    if(i<a.length-1)
      r+=c2;
  }
  return r;
}
//-->
</script>
</head>

<body bgcolor=white text=black onload="InitAlterNav()">
<form name="history">
<table bgcolor=#eeeeee border=0 cellspacing=0 cellpadding=3>
  <tr>
    <td>
      <input type=button name="tubutton" value=" U " onclick="ChangeDisplay()" style="width:25px;" id="tutitle" title=" Show URL's ">
    </td>
    <td>
      <font face="Arial, Helvetica" size=+1><b>
        <a href="altman.htm" title=" Manual " target="AlterNavDown">
          AlterNav</a>&nbsp;&nbsp;&nbsp;</b>
        <font size=-1>&copy;&nbsp;&nbsp;&nbsp;1999&nbsp;&nbsp;&nbsp;Tobias&nbsp;&nbsp;&nbsp;Steinke
        </font></font>
      </td>
    <td width=3>&nbsp;&nbsp;&nbsp;</td>
    <td>
      <input type=button value=" 1 " onclick="Call(1)" style="width:25px;"
      title=" Call history 1 ">
    </td>
    <td>
      <input type=button value=" 2 " onclick="Call(2)" style="width:25px;"
      title=" Call history 2 ">
    </td>
    <td width=3>&nbsp;&nbsp;&nbsp;</td>
    <td>
      <input type=button value=" S " onclick="Store()" style="width:25px;"
      title=" Store history ">
    </td>
    <td>
      <input type=text name="myname" onchange="StoreName()" size=37 style="width:305px;" maxlength=60 title=" Name of this history ">
    </td>
  </tr>
  <tr>
    <td>
      <input type=button value=" C " onclick="ClearHistory(0)"
      style="width:25px;" title=" Clear AlterNav's history ">
    </td>
    <td>
      <select size=1 name="list" onchange="Go(0,document.history.list.options.selectedIndex)" style="width:305px;" width=305>
        <option selected>AlterNav does not function without JavaScript.
        <option>&nbsp;&nbsp;&nbsp;
        <option>&nbsp;&nbsp;&nbsp;
      </select>
    </td>
  </tr>
</table>

```

