

Jochen Schneider

# Transactions

**Entwurf eines wissenschaftlichen  
Publikationssystems**

Diplomarbeit

Studiengang Informatik  
Fachbereich 3  
Universität Bremen

Ostern 1997

1	Einleitung .....	3
2	Fachinformation und wissenschaftliche Kommunikation.....	6
2.1	Formen der Fachinformation.....	6
	Zeitschriftenartikel.....	6
	Andere Formen wissenschaftlicher Kommunikation.....	7
	Fachinformation und Elektronisches Publizieren .....	8
2.2	Schreibmodelle.....	10
	Phasenmodell von RADA.....	11
	Prozeßmodell nach HAYES U. FLOWER .....	11
2.3	Phasen bei der Erstellung eines wissenschaftlichen Textes.....	13
	Literaturübersicht und Themenwahl.....	14
	Literaturbeschaffung und -auswertung.....	16
	Ordnung.....	17
	Schreiben.....	18
	Endredaktion.....	19
	Veröffentlichung.....	19
2.4	Schreiben mit dem Computer .....	20
	Textverarbeitung und Persönliche Informationssysteme .....	20
	Hypertextsysteme .....	21
2.5	Computerunterstützung des wissenschaftlichen Schreibens .....	23
	Beschreibung des Ist-Zustandes .....	24
	Grobanforderungen an ein technisches System zum Schreiben .....	27
3	Verteilte Hypertextsysteme.....	30
3.1	Prägung des Begriffes „Hypertext“ durch NELSON .....	30
3.2	Das World-Wide Web .....	32
	Geschichte und Anforderungen.....	32
	Implementierung des WWW.....	33
	WWW-Server.....	39
	WWW-Editoren.....	42
	Möglichkeiten zur Erweiterung des WWW.....	44
3.3	HyperWave (Hyper-G).....	45
	Anforderungen.....	45
	Entwurf und Implementierung.....	45
3.4	Xanadu.....	52
	Geschichte von Xanadu .....	53
	Xanadu Version 1988.....	57
3.5	Vergleich von Xanadu, WWW und HyperWave.....	88
4	Ein wissenschaftliches Hypertext-Publikationssystem .....	94
4.1	Ein Anwendungsszenario.....	94
4.2	Anforderungen.....	96
	Weiterentwicklung der Anforderungen.....	96
	Spezifikation von Transactions.....	101
4.3	Entwurf von Transactions.....	112
	Objektentwurf.....	112
	Funktionaler Entwurf.....	114
4.4	Weiterentwicklung des WWW .....	115
	Umsetzungen der Anforderungen durch ein erweitertes WWW.....	115
	Technische Umsetzung .....	124
4.5	Zusammenfassung und Schlußbemerkung .....	131
5	Literatur.....	134

# 1 Einleitung

Der gedruckte Bogen überwindet Raum und Zeit. Der gedruckte Bogen, die Unendlichkeit der Bücher, muß überwunden werden. DIE ELEKTROBIBLIOTHEK.

El Lissitzky, „Topographie der Typographie“, 1923

Das World Wide Web (WWW) hat der Benutzung des Internets zum Durchbruch verholfen. Es gibt zur Zeit viele Bestrebungen, es zu verbessern: Style sheets und die Programmierung mittels Java (Sun) und ActiveX (Microsoft) stellen nur zwei Beispiele dar. Sie zielen auf die Präsentation von Information im WWW. Es gibt auch Entwicklungen im Bereich der Informationsorganisation, z.B. die Verbesserung von Indizes und Suchmaschinen. Wieder andere Erweiterungen zielen auf Abrechnung, ihre Sicherheit und die Sicherheit von Daten ab, z.B. das Secure Hypertext Transfer Protocoll (SHTTP) und digitales Geld wie DigiCash. Relativ neu sind Kombinationen aus WWW-Browsern und -Editoren, z.B. Netscape Atlas Gold, AOL-Press und Adobe PageMill.

Zusammengefaßt dienen aktuelle WWW-Erweiterungen dazu, Inhalte attraktiver darzustellen, Geschäfte über das Internet zu ermöglichen und es Benutzerinnen zu erlauben, selbst Inhalte im Netz zu verbreiten. Benutzerinnen sollen also insgesamt mehr ‚surfen‘ und konsumieren und dürfen sich auch selbst ein wenig präsentieren. Wo bleibt da die vielgenannte Information? Sie fällt unter den Bereich ‚Inhalt‘.

Das WWW stellt ein Hypertextsystem dar. Unter dem Namen ‚Hypertext‘ existiert ein Konzept in der Informatik, welches sich mit der Verwaltung und Benutzbarkeit von Textstücken beschäftigt, zwischen denen abstrakte, jedoch explizite Verbindungen, ‚Links‘ bestehen. Hypertexte unterscheiden sich in der Benutzung erheblich von traditionellen Textträgern wie z.B. Büchern, lassen sich aber, das ist zumindest das große Versprechen hinter dem Hypertextkonzept, flexibler Lesen und Schreiben.

Die Frage nach dem speziellen Einsatzzweck eines Hypertextsystems wird in einigen Jahren durch das Aufkommen immer allgemeinerer Systeme vielleicht nicht mehr relevant sein, so wie sie es z.B. bei Textverarbeitungssystemen kaum noch ist. Derzeit bestimmt sie aber noch die Entwicklung. Dabei steht v.a. das Erschließen eines neuen Marktes, die Sicherung von Marktanteilen und Experimente mit neuen Techniken und Inhalten im Vordergrund.

Das WWW war jedoch ursprünglich für die Erstellung und Verbreitung von technischen und wissenschaftlichen Texten entworfen worden, ähnlich wie seine konzeptionellen Vorgänger Memex und Xanadu. Für das Publizieren im klassischen Sinn ist es jedoch kaum geeignet: Text kann nicht im System selbst erstellt werden, es gibt nur eine Art von Dokumenten bzw. Strukturelementen (Links) zwischen ihnen, und der Schutz von Urheberrechten wird nicht vom System unterstützt.

In dieser Arbeit wird ein Hypertextsystem für das wissenschaftliche Arbeiten spezifiziert und auf Basis des WWW entworfen. Dabei steht die Erstellung von wissenschaftlichen Texten im Vordergrund, nicht Fragen der Publikation und Abrechnung. Zunächst werden die Hypertextsysteme WWW, HyperWave (früher HyperG) und Xanadu verglichen. Das WWW wurde wegen seines großen, weltweiten Erfolges gewählt, HyperWave, weil es verspricht, eine Weiterentwicklung des WWW zu sein und Xanadu schließlich als nie umgesetztes Vorbild der anderen beiden Systeme. Der Gesichtspunkt, der für den Vergleich im Vordergrund steht, ist die Eignung des jeweiligen Systems für die Erstellung und Veröffentlichung von wissenschaftlichen Texten. Dabei handelt es sich um ein relativ stark abgegrenztes und reglementiertes Gebiet des Schreibens, dessen Regeln u.a. in Anleitungen zum Erstellen wissenschaftlicher Arbeiten und zum Einreichen von Artikeln an wissenschaftliche Zeitschriften festgelegt sind. Autorinnen und Leserinnen von wissenschaftlichen Texten stammen beide aus demselben Milieu, dem akademischen. In ihm können neue Publikati-

onsmittel exemplarisch getestet werden, wobei kommerzielle Funktionen wie Abrechnungen zunächst außer acht gelassen werden können.

Das Schreiben von wissenschaftlichen Texten stellt ein Teil des Handwerks von Wissenschaftlerinnen dar. Es wird nicht nur durch das Lernen von Meisterinnen auf diesem Gebiet, sondern auch durch Anleitungen zum ‚wissenschaftlichen Arbeiten‘ weitergegeben. Einige dieser Anleitungen werden in der vorliegenden Arbeit ausgewertet. In ihnen wird teilweise noch die Benutzung von Karteikarten und maschinengeschriebenen Manuskripten vorgeschlagen, wo viele Wissenschaftlerinnen schon mit Textverarbeitungsprogrammen arbeiten, mit denen sie photosatzfähige Druckvorlagen erstellen. Es stellt sich die Frage, ob das Schreiben von wissenschaftlichen Texten noch weiter durch den Computer unterstützt werden kann. Mit dem Aufkommen von sogenannten Preprint-Servern und v.a. dem phänomenalen Erfolg des weltweiten Hypertextsystemes WWW bietet es sich an, den ganzen Prozeß vom ersten Informieren über ein Thema bis zum Publizieren im Computer zu erledigen.

Das Interesse an einer solchen Maschinisierung und Rationalisierung des Schreib- und Publikationsprozesses korrespondiert mit dem Zwang zu Einsparungen in Bibliotheken (und macht sie teilweise – wie oft in solchen Fällen – erst möglich). Auf diesen Sparzwang wird zuerst mit Abbestellungen von wissenschaftlichen Zeitschriften reagiert. Daß diese Abbestellungen und Umstellungen von Zeitschriften auf digitale Form Preiserhöhungen für die restlichen Abonentinnen bewirken, die wieder zu neuen Abbestellungen führen, ist bedauerlich. Wie wir noch sehen werden, sind Wissenschaftlerinnen jedoch vielfach selbst an einer Änderung des zeitschriftenbasierten Publikationsprozeß interessiert, weil sie mit langen Begutachtungszeiten und geringem Verbreitungsgrad ihrer Artikel unzufrieden sind. Andererseits sind sie auf begutachtete Zeitschriften angewiesen, um Ergebnisse zu veröffentlichen und sich damit selbst bekannt zu machen.

Preprints als alternativ verbreitete Zeitschriftenartikel werden oft als Postscriptdateien verbreitet, die mit klassischen Textverarbeitungssystemen erzeugt wurden. Diese Dateien werden dazu auf FTP-Server im Internet gelegt. Sie können quasi nur ausgedruckt gelesen werden, weil die Auflösung von Computerbildschirmen durchschnittlich viel geringer als die von Druckern ist. Bei der Erstellung eines Textes gehen die benutzten Strukturen wie Abschnitte, die auch in einer anderen Reihenfolge ein Dokument ergeben könnten, verloren. Hypertexte werden Strukturen genannt, die Textabschnitte auf mehr als eine Weise verbinden und daher mehrere Lesereihenfolgen anbieten. Bevor drei verteilte Hypertextsysteme beschrieben werden, wird zunächst im nächsten Kapitel das Anwendungsgebiet dieser Arbeit, das wissenschaftliche Schreiben, untersucht und im Hinblick auf eine Umsetzung mit dem Computer modelliert.

Für die digitale Umsetzung von wissenschaftlichen Publikationen gibt es weltweit Initiativen, z. B. das ‚Digital Library Project‘ der amerikanischen National Science Foundation (NSF) und das Programm ‚Information als Rohstoff für Innovation‘ der deutschen Regierung. Diese Vorhaben verfolgen evolutionäre Ansätze, möchten also die zugrundeliegende Ordnung der Erstellung und Verbreitung von wissenschaftlicher Information zunächst nicht antasten, sondern nur verbessern.

Die vorliegende Arbeit zeigt, daß auch NELSON mit dem Hypertextsystem Xanadu die Absicht verfolgt hat, die Erstellung, Verbreitung und Nutzung von wissenschaftlicher Literatur zu verbessern, aber auf radikale Weise: Von der Erstellung bis zur Benutzung soll alles in einem System ablaufen, welches daher Zettelkasten, Textverarbeitung, Druckerpresse, Buchhandel und Bücherei in einem ist.

Heutige Ansätze für elektronisches Publizieren basieren auf dem WWW, was aufgrund der schon geschilderten Popularität nicht verwundert. Wenn v.a. wissenschaftliches Publizieren mittels einer Erweiterung dieses Systems erfolgen soll, so kann das WWW, so die Ausgangsthese dieser Arbeit, in einigen Punkten von Xanadu lernen. Um diese These zu prüfen, wird ein Publikationssystem spezifiziert, das von Xanadu inspiriert ist. Es wird zunächst geschildert, worum es sich beim

Problem des wissenschaftlichen Publizierens in traditioneller Form handelt. Der Publikationsprozeß wird dazu analysiert und modelliert. Danach werden die drei Hypertextsysteme WWW, HyperWave und Xanadu beschrieben und verglichen. Es wird ein Hypertext-Publikationssystem auf Basis der Analyse spezifiziert, das ‚Transactions‘ genannt wird. Schließlich wird beschrieben, wie das WWW erweitert werden kann, um wissenschaftliches Publizieren zu ermöglichen.

## 2 Fachinformation und wissenschaftliche Kommunikation

Diese Arbeit untersucht, wie Fachinformation mit Hilfe von speziellen Computersystemen erstellt und genutzt werden kann. Fachinformation ist „die aktuelle fachliche Information zum eigenen Arbeitsgebiet“ für Wissenschaft und Wirtschaft (BRÜGGEMANN-KLEIN et al. 95, 26). Sie ist ein Teil der wissenschaftlichen Kommunikation, die WALKER U. HURT folgendermaßen definieren:

Scientific/technical communication is one means of sharing knowledge or information in the form of ideas, research findings, interpretations, or observations by scientists and technologists. ... It differs from more general communication in that it makes reference to a well-defined and codified corpus of knowledge and formats of publication. (90, X)

(Auf den Unterschied zwischen wissenschaftlichen und technischen Texten wird in Abschnitt 2.1.2 eingegangen.) Das Veröffentlichen eines Textes ist in den Wissenschaften die übliche Art, Arbeitsergebnisse vorzustellen und Qualifikationen nachzuweisen. Wissenschaftliche Kommunikation wird hier daher als schriftliche analysiert. Diese schriftliche Kommunikation wird von zwei Seiten beschrieben: von der Seite der Nutzung in Form von Artikeln u. ä. und der Seite der Erzeugung durch die einzelne Schreiberin.

### 2.1 Formen der Fachinformation

Um die schriftliche wissenschaftliche Kommunikation in den späteren Kapiteln auf den Computer abbilden zu können, wird hier zunächst ein Modell von ihr aufgestellt. Dazu ist es notwendig, ihre verschiedenen Formen zu analysieren und voneinander abzugrenzen.

#### 2.1.1 Zeitschriftenartikel

Zeitschriften sind die wichtigsten aktuellen Kommunikationsmittel. “The journal – sometimes known as the archival or primary journal – is a collection of original contributions intended to communicate new research-findings, articulate new theories or concepts, report observations of events or describe new phenomena” (ebd., 1). Eine Zeitschrift enthält als „original contributions“ speziell für sie verfaßte Artikel, Ausschnitte oder Kurzfassungen von schriftlichen oder mündlichen Darstellungen und Nachrichten von wissenschaftlichen Gesellschaften (ebd.). Die britische Royal Society nennt als wichtigste Funktionen von wissenschaftlichen oder technischen Zeitschriften:

1. Verbreitung von Forschungsergebnissen an Forscherinnen, Technikerinnen und andere (Hauptfunktion)
2. Archivierung von Forschung und Beobachtungen
3. Anmeldung einer Idee, Theorie oder Entdeckung als intellektuellen Besitz eines Individuums oder einer Gruppe
4. Aufrechterhalten von Qualitätsstandards (ebd., 18)

Daneben dienen diese Zeitschriften dem Aufbau der Reputation einer Wissenschaftlerin, dem Verbreiten von Informationen über Produkte und anderen Ankündigungen und dem Veröffentlichen von Buchbesprechungen (ebd.). Ursprünglich wurden Forschungsergebnisse in Zeitschriften nur veröffentlicht, um sie schnell zu verbreiten, nicht, damit sie mit ihnen archiviert würden. Die Archivfunktion wurde nur Büchern zugestanden, die die wichtigsten Punkte aus Zeitschriftenartikeln zusammenfassen (ebd., 40).

Artikel werden in Zeitschriften gewöhnlich nur nach Begutachtung („Refereeing“) veröffentlicht. Dabei bewertet eine Spezialistin aus demselben Gebiet wie die Autorin deren Artikel und gibt Empfehlungen an die Herausgeberinnen, die dann immer noch selbst über eine Veröffentlichung ent-

scheiden. Die Begutachterin bleibt meist anonym. Eine Begutachtung führt oft zur Überarbeitung von Artikeln durch ihre Autorinnen. Sie soll weiterhin sicherstellen, daß ein Artikel dem Profil einer Zeitschrift entspricht und von hoher Qualität ist (ebd., 25ff). Das Veröffentlichen in einer Zeitschrift ist nicht immer kostenlos. Mit ‚Page charges‘ wird eine Gebühr pro Seite bezeichnet, die Geldgeberinnen von Forschung bezahlen sollen, damit entsprechende Ergebnisse veröffentlicht werden. Page charges sind für manche Zeitschriften jedoch obligatorisch (ebd., 31).

In der Wissenschaftsgemeinde werden neben den schon beschriebenen Vorteilen auch Nachteile der Kommunikation über Zeitschriften diskutiert. WALKER U. HURT nennen die folgenden: zu große Verzögerungen beim Veröffentlichen (oft ein Jahr oder mehr), Verstümmelung durch Kürzungen, zu starke Streuung von Information in Folge von Spezialisierung, zu hoher Preis und schließlich zu zeitaufwendige Erstellung (47f). Zeitschriften sind aber dennoch unter Forscherinnen und Technikerinnen hoch angesehen, weil v. a. erstere ihnen ihren Ruf verdanken (ebd., 69).

Bei der Herstellung einer Zeitschrift lassen sich aus Sicht ihrer Macherinnen folgende Schritte unterscheiden:

1. Herausgabe: Bewertung von eingehenden Manuskripten, Auswahl von Gutachterinnen, Korrespondenz mit ihnen und mit Autorinnen
2. Redaktion: Überprüfung und Korrektur von Rechtschreibung, Formulierung und Literaturverweisen; typographische Auszeichnung
3. Korrektur und Satz
4. Verwaltung: Buchhaltung, Abonnementverwaltung, Vertrieb.
5. Werbung und PR (ebd., S. 38f, nach HESLOP-HARRISON 78)

Redaktion, Korrektur und Satz werden heute allerdings oft Autorinnen von Artikeln aufgebürdet.

### 2.1.2 Andere Formen wissenschaftlicher Kommunikation

Zu wissenschaftlichen Zeitschriften und den in ihnen veröffentlichten Artikeln wurden verschiedene Alternativen entwickelt, die jeweils unterschiedliche Probleme ansprechen. Sie lassen sich in ‚Weiße‘ und ‚Graue Literatur‘ einteilen. Die ‚Graue Literatur‘ steht zwischen der ‚Weißen‘, die in bestimmten Bezügen vorrätig und v. a. bibliographisch erfaßt ist, und der ‚Schwarzen‘, für die dies überhaupt nicht gilt.

‚Graue Literatur‘ enthält nicht unbedingt neue wissenschaftliche Erkenntnisse, wird nicht über das Buchvertriebssystem transportiert und ist schwer zu lokalisieren. Beispiele sind die meisten Tagungsbände, technische Berichte, Abschlußarbeiten und Patente (WALKER U. HURT 90, xxi). Die bereits beschriebenen wissenschaftlichen Zeitschriften hingegen gehören zur Weißen Literatur.

POENICKE unterscheidet weiterhin selbständig und unselbständig erschienene Literatur. Selbständig erschienene umfaßt in sich abgeschlossene Veröffentlichungen mit eigenem Titelblatt und eigener Seitenzählung und meist auch eigener Angabe von Verlag, Erscheinungsort und -jahr in einem Impressum. Hauptvertreter dieser Art sind Monographien, Anthologien, Nachschlagewerke und Zeitschriften. Als unselbständige Literatur werden Teile von selbständiger bezeichnet, z. B. Zeitschriftenaufsätze, Gedichte, Beiträge zu Anthologien, Eintragungen in Nachschlagewerken, einzelne Buchkapitel und ungedruckte Texte wie Dissertationen (88, 150). Unselbständige Literatur entspricht also nicht Grauer Literatur .

Technische Zeitschriften (Technical journals) gehören zur Weißen Literatur, unterscheiden sich von wissenschaftlichen Zeitschriften jedoch in ihrer Anwendung. Der Inhalt von technischen Zeitschriften wurde meist schon in einer wissenschaftlichen Zeitschrift, einem Patent oder einem technischen Bericht (s. unten) veröffentlicht. Technische Zeitschriften sind meist leichter lesbar als archivarische Zeitschriften, ihre Typographie ist ansprechender, ihre Artikel enthalten viele farbige

Grafiken, Diagramme und Fotografien, aber auch mehr Werbung. Sie dienen vornehmlich der Verbreitung von Erkenntnissen aus der Forschung, damit diese in Wissenschaft und Technik angewendet werden können (WALKER U. HURT 90, 21f).

Ein ‚Seperate‘ ist ein Zeitschriftenartikel, der als eigenständige bibliographische Einheit veröffentlicht wird und damit ebenfalls zur Weißen Literatur gehört. Er wird vor dem Abdruck begutachtet und u. U. nach einer weiteren und strengeren Begutachtung schließlich auch in einer wissenschaftlichen Zeitung abgedruckt (ebd., 58). Mit ‚Preprints‘ werden Schriftstücke bezeichnet, die zwar für den Druck bestimmt sind, aber noch nicht gedruckt wurden. Sie gehören zur Grauen Literatur. Beispiele für Preprints sind wissenschaftliche Artikel, die ihre Verfasserinnen an Kolleginnen verteilen und sie gleichzeitig an Zeitschriften zur Veröffentlichung schicken.

Preprints sind deshalb aktueller, weil sie noch nicht begutachtet wurden. Daher enthalten sie möglicherweise kontroverse Inhalte. Es gibt Versuche, die Verteilung von Preprints zu organisieren, die jedoch teilweise gescheitert sind. Nachteile von Preprints liegen darin, daß sie nicht indiziert oder bibliographisch erfaßt werden. Sie sind daher eigentlich nicht zitierbar. Teilweise werden Preprints nicht einmal Korrektur gelesen (ebd.).

Gerade der technische Bericht gilt als wichtiger Gegenpart zu wissenschaftlichen Artikeln. Er wendet sich an eine Leserin, die eine bestimmte Information oder einen bestimmten Rat braucht, soll also v. a. nützlich sein (ebd., 114). Technische Berichte werden entweder in Forschungs- und Entwicklungsabteilungen privater Unternehmen oder im Rahmen staatlich finanzierter Forschung als Abschlußberichte geschrieben. Wenn technische Berichte in Unternehmen entstehen, werden sie meist nicht öffentlich zugänglich gemacht und sind dann der Grauen Literatur zuzurechnen (ebd., 107f). MEADOWS nennt daher auch als wichtigsten Unterschied zwischen einem Artikel in einer wissenschaftlichen Zeitschrift und einem technischen Bericht, daß hinter letzterem meist eine Sponsorin steht, die sein Erscheinen finanziert und über seine Verbreitung entscheidet (74, 116).

Eine weitere Besonderheit von technischen Berichten im Vergleich zu Zeitschriftenartikeln liegt darin, daß erstere der Technikwelt entstammen (WALKER U. HURT 90, 115f). Sie stellen oft nur einen Zwischenschritt dar, sind informeller, zitieren weniger und werden weniger zitiert, sind teilweise geheim, haben eine kürzere Veröffentlichungszeit, werden nur über Projektgeldgeberinnen ‚begutachtet‘ und haben einen eher anwendungsorientierten Inhalt (ebd., 115f u. 119ff). Zeitschriftenartikel hingegen sind kürzer und formeller und dienen u. a., wie schon erwähnt, der Mehrung wissenschaftlichen Ruhmes (ebd., 121f).

### 2.1.3 Fachinformation und Elektronisches Publizieren

Mit ‚Elektronischem Publizieren‘ wird die Veröffentlichung von Texten in elektronischer Form bezeichnet, in welcher die Texte entweder direkt gelesen oder von der aus sie gedruckt werden (BRÜGGEMANN-KLEIN 89, 150). Etwas ausführlicher:

Unter elektronischem Publizieren versteht man einen Prozeß, in dem Dokumente die Informationskette von der Erstellung durch die Autoren über die Begutachtung und Aufbereitung im Verlag, die Bereitstellung und schließlich die Nutzung durch die Leser ohne Medienbrüche in elektronischer Form durchläuft. (ENDRES 95, 15)

Als Beispiele für Elektronisches Publizieren nennen BRÜGGEMANN-KLEIN et al. digitale Bibliotheken, verteilte multimediale Informationssysteme und elektronische Zeitschriften (95, 27). Digitale Bibliotheken helfen (aus Bibliothekssicht) gegen Platz- und Finanzprobleme und (aus Leserinnensicht) gegen fehlende Aktualität und Probleme beim Finden von Literatur. Sie werfen jedoch noch wirtschaftliche und rechtliche Probleme auf, außerdem ihre langfristige Archivierung ungeklärt (ebd.).



Ein multimediales Informationssystem nimmt von der einzelnen Nutzerin Angaben über für sie interessante Informationsarten an, worauf es „alle dafür relevanten Quellen erschließt, die gefundenen Informationen bewertet und selektiv und in variabler Darstellung anbietet“ (ebd.). Es kommt also darauf an, daß das System die Eingaben der Benutzerin richtig verarbeiten kann und daß genügend Informationen zu einem Thema elektronisch vorliegen.

Für die Verlegerin liegen die Vorteile elektronischer Zeitschriften in den geringeren Material- und Druckkosten, für die Leserin „in der schnellen Verfügbarkeit“ und „in den verbesserten Möglichkeiten der Recherche und der Weiterbearbeitung“ (27). Nach WALKER U. HURT beträgt die Zeitersparnis bei der Erstellung elektronischer Zeitschriften im Vergleich zu herkömmlichen bis zu 80% (90, 69f). Sie nennen außerdem das Wegfallen einer Längenbeschränkung für Artikel, die direkte Verbindung eines Artikels mit Kommentaren, Zusätzen, Verbesserungen, Kritiken und Bestätigungen und die Lieferung nur von interessanten Artikeln an jeden beliebigen Ort (mit einem Computer am Netz) als Vorteile (ebd.). Sie nennen aber auch Nachteile: Die Abhängigkeit von Hard- und Software, die schlechte Unterstützung von nichttextlichen Inhalten und die Starrheit des alten Zeitschriftensystems gegenüber Veränderungen (ebd., 72).

Der technische Bericht als weniger starre Publikationsform als der Zeitschriftenartikel legt daher eine Elektronifizierung näher als klassische Artikel (WALKER U. HURT 90, 127f). Bei ihm wurde sie schon öfter als bei allen anderen Literaturformen der Fachinformation versucht (ebd., 124). Hinzu kommt, daß technische Berichte wie Onlinedienste staatlich subventioniert sind (was man immer noch über das Internet sagen kann) und es weniger bibliographische Dienste wie die Erstellung von Indizes und Kurzfassungen für sie gibt (ebd., 124f). WALKER U. HURT legen sogar BUSHS Artikel ‚As We May Think‘ (45) so aus, daß er ein „full-text online system for technical reports“ beschreibe (ebd., 126). BUSHS Aufsatz wird in dieser Arbeit noch öfter erwähnt, weil sich viele Hypertextpioniere auf ihn berufen, z.B. TED NELSON (s. Abschnitt 3.1).

Verlage müssen in einem elektronischen Netzwerk nicht überflüssig werden. Sie können auch dort Dokumente<sup>1</sup> an ihren Standards messen und sie veröffentlichen (DREXLER 86, 222f). BRÜGGEMANN-KLEIN et al. sprechen von „Gutachter- und Editierprozessen“, die sogar eine höhere Qualität als heute sicherstellen könnten (95, 27). Ob elektronische Abbilder von wissenschaftlichen Zeitschriften auf Dauer existieren werden, ist hingegen fraglich. Elektronische Artikel können nämlich aus einem solchen Verbund gelöst werden, was auch neue Vergütungsformen erfordert (ebd.). BRÜGGEMANN-KLEIN et al. drücken insgesamt eine Skepsis gegen die allzu nahe Modellierung der neuen Medien der wissenschaftlichen Kommunikation nach den alten aus:

Bei elektronischen Zeitschriften und digitalen Bibliotheken geht es eigentlich darum, in der Vergangenheit bewährte Formen und Einrichtungen der Fachinformation auf eine neue technische Basis zu überführen. Das erinnert etwas an die Kutschen, die man im vorigen Jahrhundert mit Benzinmotoren ausrüstete. (ebd.)

GINSPARG sagt noch aus anderen Gründen voraus, daß elektronische Dienste Zeitschriftenverlage in der jetzigen Form ersetzen werden. Wenn Wissenschaftlerinnen nämlich veröffentlichen, um Erkenntnisse zu verbreiten und ihren eigenen Ruhm zu mehren, haben sie es eben nicht auf Geld abgesehen (96, Abschn. 3). Sie zahlen sogar manchmal für eine Veröffentlichung, was als ‚Page charges‘ schon beschrieben wurde (s. S. 7). Für Wissenschaftlerinnen ist es also schwer einzusehen, daß irgend jemand über die Grunddienstleistung hinaus Geld mit dem Publizieren ihrer Artikel verdienen sollte (ebd.).

HARNAD unterscheidet ebenfalls zwischen kommerzieller und wissenschaftlicher Veröffentlichung. Bei der ersten Form ist eine Verlegerin nötig, um die Autorin von den Handelsaspekten ihrer

1 Unter Dokumenten werden in dieser Arbeit allgemein Schriftstücke verstanden, die eine gewisse Selbständigkeit besitzen und eher am Ende des schriftlichen wissenschaftlichen Produktionsprozesses stehen.

Arbeit zu entlasten, das Interesse von Autorin und Verlegerin zielen in eine Richtung. Bei der wissenschaftlichen Veröffentlichung ist das jedoch nicht der Fall. Eine Wissenschaftlerin ist v.a. an der Verbreitung ihrer Arbeit interessiert (so lange sie ihr zugeschrieben wird), ob diese über offizielle Kanäle (Zeitschriftenabonnements, Nachdrucke) geschieht oder nicht (Fotokopien). Sie kann jedoch einer Verbreitung, bei der ihre Verlegerin nichts verdient, letztendlich nicht zustimmen, weil eine solche die Existenz des Mediums, über das sie ihre Arbeit verbreitet, gefährdet (HARNAD 95, Abschn. ‚Trade Publishing vs. Scholarly Publishing‘).

Auf einem Dateiserver im Internet (einem sogenannten FTP-Server) kann eine Autorin sowohl ein Preprint (s. S. 8) als auch die endgültige Form eines Dokumentes veröffentlichen. Diese Art von Service stellt eine noch größere Gefahr für Verlage dar als die Verbreitung über Fotokopien. Bei einer weiteren Loslösung vom Zeitschriftenmodell dürfte Elektronisches Publizieren jedoch 70-80% billiger sein als traditionelles Publizieren und daher notfalls von Wissenschaftlerinnen und den sie unterstützenden Institutionen allein finanzierbar sein. Wenn also Verlage nicht bereit sind, sich auf diese Art der Veröffentlichung einzustellen, könnten sie sie doch nicht verhindern (ebd., Abschn. ‚The Subversive Potential of Anonymous FTP‘).

Auch ohne Verlag müßten, so HARNAD, ihre klassischen Dienste wie Herausgabe, Redaktion und Satz<sup>2</sup> erhalten bleiben, v.a. jedoch die Qualitätskontrolle. Diese wird jedoch heute schon von Wissenschaftlern selbst ausgeübt, wie der neben ‚Refereeing‘ andere englische Name dafür, ‚Peer review‘<sup>3</sup>, ausdrückt (ebd.). Diese Qualitätskontrolle ist also nicht abhängig vom Medium der wissenschaftlichen Veröffentlichung (ebd., Abschn. ‚The Anarchic Initial Conditions on the Net‘).

HARNAD nennt ein Beispiel für die Umsetzung neuer Ideen der Qualitätskontrolle im Internet: PSYCOLOQUY ist eine internationale, interdisziplinäre elektronische Zeitschrift für Verhaltens- und Kognitionswissenschaften, welche von der American Psychological Association unterstützt wird. In ihr wird nur nach Begutachtung veröffentlicht. Sie bietet auch Kommentare zu Artikeln und Antworten von Autorinnen darauf. Diese Interaktivität nennt HARNAD das eigentlich revolutionäre an dieser Form der wissenschaftlichen Veröffentlichung, sieht aber auch bei ihr die Notwendigkeit der Begutachtung. Die Schaffung einer „Prestigehierarchie“ für elektronische Zeitschriften ist seiner Meinung nach nämlich die Voraussetzung dafür, daß Wissenschaftlerinnen und wissenschaftliche Institute diese Zeitschriften annehmen (ebd., Abschn. ‚Imposing Order Through Peer Review‘).

HARNAD nennt folgende Vorteile der Übertragung des Begutachtungsprozesses auf ein elektronisches Medium: Begutachterinnen können sich freiwillig zur Verfügung stellen, die Kommunikation zwischen ihnen und einer Herausgeberin geschieht schneller und Artikel können elektronisch kommentiert werden. Diese Vorteile gelten auch für die Begutachtung von Artikeln für klassische Zeitschriften im Netz, weil auch diese meist mit dem Computer geschrieben werden (ebd.). In einem Computernetzwerk ist es außerdem möglich, neben begutachteten Veröffentlichungen auch unbegutachtete zuzulassen. Begutachtung ist nämlich manchmal eine zu große Hürde für gewisse Ideen, die dennoch ihren Wert haben. Die Begutachtung könnte durch unmoderierte Diskussionen ergänzt werden, auch in der Rolle von Qualitätssicherung (ebd., Abschn. ‚Creative Chaos‘).

## 2.2 Schreibmodelle

In diesem Abschnitt wird ein Modell des wissenschaftlichen Schreibens vorbereitet, das die Spezifikation von Software ermöglicht, die dieses Schreiben unterstützen kann. Im ersten Unterabschnitt

2 Der jedoch häufig von Wissenschaftlerinnen selbst erledigt wird.

3 Eine ‚Review‘ ist (im Umfeld der ACM) eine Bewertung eines Artikels durch Expertinnen für eine Redakteurin im Hinblick auf Klarheit, Fundiertheit, Neuigkeit, bisherige Veröffentlichung, richtige Zitate usw. ‚Refereeing‘ bedeutet einen gründlichen Review v.a. nach den ersten beiden Kriterien ( DENNING U. ROUS 95, 107f).

wird der Zusammenhang zwischen den wichtigsten Sichtweisen auf den Schreibprozeß, das kognitive/linguistische, das praktisch/technische und das logische Modell, dargestellt und durch eine Graphik verdeutlicht. Danach wird kurz die erste genannte Sichtweise erläutert, ebenfalls mit einem graphischen Modell. Aus der praktisch/technischen und der logischen Sichtweise werden am Ende dieses Kapitels eine Ist-Analyse durchgeführt und Anforderungen an ein Computer-Schreibsystem formuliert.

### 2.2.1 Phasenmodell von RADA

Als Übersicht für die theoretische Herangehensweise an das Schreiben soll ein Modell aus der Informatik dienen, die drei Phasen und drei Ebenen des Schreibens nach RADA (91, s. Abbildung 1). Die oberste Ebene stellt die kognitiven Prozesse dar, die bei der Erstellung eines Textes durchlaufen werden. Die mittlere Ebene nennt die entsprechenden physischen Strukturen auf Papier. Die untere Ebene schließlich gibt die dahinter liegenden logischen Strukturen an, die bei der Dokumentenerstellung mit einem Computer explizit gemacht werden können (ebd., 6). Die Phasen in den entsprechenden Ebenen werden im folgenden ausführlich beschrieben, wenn auch nicht immer unter den hier angegebenen Namen.

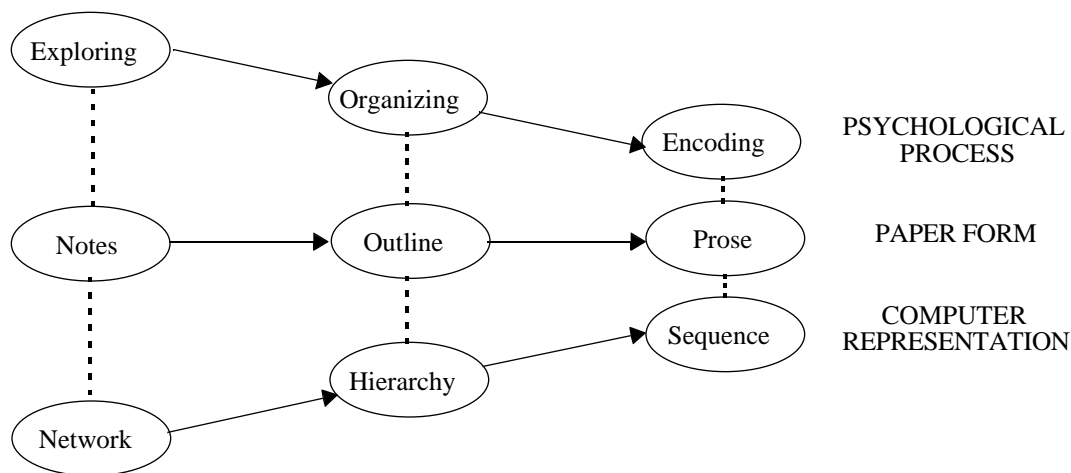


Abb. 1. Die drei Phasen des Schreibens, nach RADA 91, 6

Für diese Arbeit interessieren v.a. die unteren beiden Ebenen, wobei die mittlere die ist, die durch die untere auf dem Computer abgebildet werden soll. Sie ist auch die, auf die in praktischen Anleitungen zum Schreiben vornehmlich abgehoben wird (s. Abschnitt 2.3 auf S. 13). Mit der oberen Ebene beschäftigen sich v.a. Linguistinnen und Psychologinnen. Sie sollen in den nächsten Absätzen kurz zu Wort kommen.

### 2.2.2 Prozeßmodell nach HAYES U. FLOWER

Im folgenden geht es um einen vielzitierten Ansatz, der „die kognitive Schreibforschung bis heute wesentlich beeinflusst“ (JECHLE 92, 6). Dabei handelt es sich um das Modell von HAYES U. FLOWER (s. Abbildung 2). Es modelliert den Schreibprozeß und sein Zusammenspiel mit dem Arbeitsumfeld („Task Environment“) einerseits und dem (Langzeit-)Gedächtnis der Schreiberin („Writer’s Long-Term Memory“) andererseits. Das Arbeitsumfeld umfaßt sowohl die Anforderungen an die Schreiberin als auch den von ihr bereits erstellten Text, allerdings keine Notizen. Das Langzeitgedächtnis unterstützt nach dem Modell mit dem Wissen über das Thema, die Leserinnenschaft und über die Abläufe des Schreibens selbst das Generieren von Ideen und Fragmenten (RIEHM et al. 92, 17f).

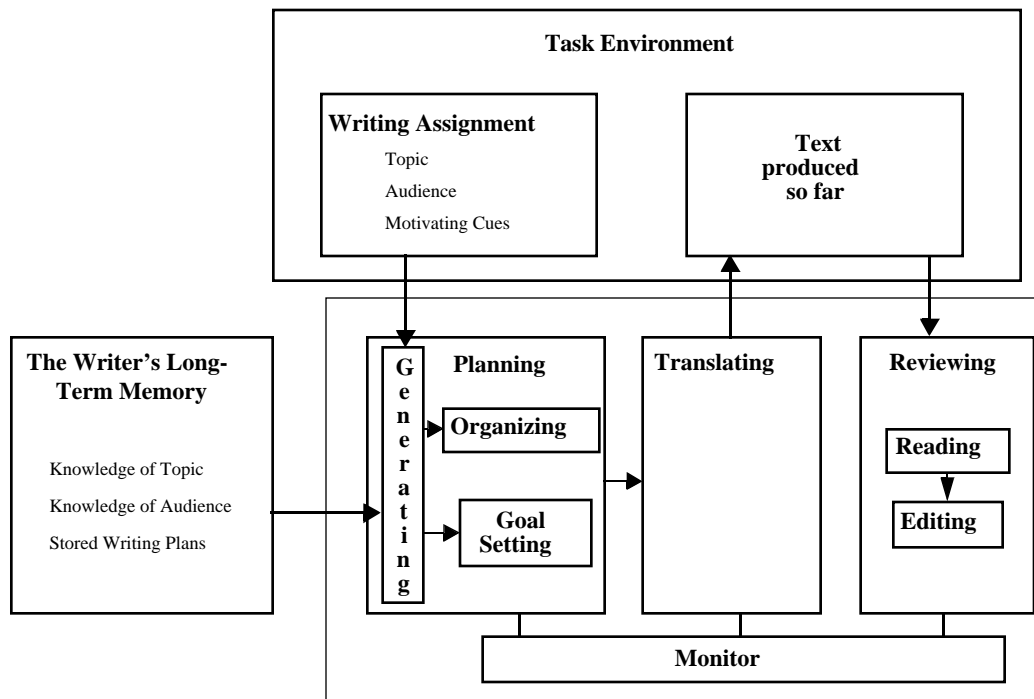


Abb. 2. Schreibmodell nach HAYES U. FLOWER 80, 11, abgebildet in RIEHM et al. 92, 18.

Ältere Modelle des Schreibprozesses gehen von einem sequentiellen Ablauf der Phasen Planen, Formulieren und Überarbeiten aus (s. z.B. RAU 94, 12f zur Übersicht). Im Modell von HAYES U. FLOWER dagegen wird berücksichtigt, daß Schreiberinnen schon geschriebene Textteile beim erneuten Lesen umschreiben und in neue Zusammenhänge stellen, sogar völlig neue Inhalte erzeugen. Sie nennen den Schreibprozeß daher rekursiv, aber immer noch sukzessiv, weil zu einem bestimmten Zeitpunkt nur ein Teilprozeß aktiv ist (80, nach JECHLE 92, 12f). Der Prozeß ‚Monitor‘ überwacht und steuert diese Prozesse. Er ermöglicht ein Wechseln zwischen ihnen, stellt aber auch die Zielgerichtetheit der Gesamtarbeit sicher. Außerdem wird durch den Monitor die individuelle Vorgehensweise der Schreiberin abgebildet (RIEHM et al. 92, ebd.).

JECHLE unterscheidet als wichtigste der Teilprozesse des Schreibens:

1. Repräsentation der Schreibaufgabe und Zielbildung (Goal setting)
2. Bereitstellen von Information (Generating)
3. Anordnen von Information (Organizing)
4. Einzelsprachliche Realisierung (Translating)
5. Durchsehen und Überarbeiten (Reviewing bzw. Revising) (92, 20)

Die Repräsentation der Schreibaufgabe beinhaltet die zugrundeliegende Intention und Motivation der Autorin, das Schreibthema, einen Schreibplan und die Kenntnis um die Leserinnen (ebd., 20ff). Beim Bereitstellen von Informationen greift die Schreiberin auf ihr Langzeitgedächtnis und auf externe Quellen zurück. Die Schreibforschung setzt sich v.a. mit dem Gedächtnis auseinander (ebd., 24).

Nach der Erstellung liegen Inhalte in einer Reihenfolge vor, die dem Schreibziel nicht angemessen ist. Sie unterscheiden sich außerdem darin, ob sie eher zusammenfassen oder sich auf Details beziehen (ebd., 27).

Allgemein besteht die Funktion der mit dem Anordnen von Inhalten verbundenen Prozesse darin, zwischen der Repräsentation bereitgestellter Inhalte, die sich aus der Repräsentation von Wissen im Gedächtnis ableitet, und einer linearisierten Repräsentation von Inhalten, wie

sie für eine Umsetzung in Sprache erforderlich ist, zu vermitteln. (ebd., 29f)

Die einfachsten Ordnungsformen von Inhalten sind die räumliche, die chronologische und die hierarchische Ordnung. Letztere, welche sich v.a. für die Darstellung abstrakter Sachverhalte anbietet, muß nicht eindeutig sein, sie kann sich z.B. nach Adressatinnen unterscheiden (ebd. 28).

Die einzelsprachliche Realisierung übersetzt die linearisierten Ideen und äußert sie graphomotorisch (ebd., 30). Sie enthält sowohl inhaltliche, sprachliche als auch kommunikative Aspekte (ebd., 35). ‚Sprachlich‘ bezieht sich allgemein auf eine natürliche Sprache mit ihren Vorgaben (ebd., 30ff), die ‚kommunikativen Aspekte‘ auf die konkrete Umsetzung von Ideen in eine sprachliche Form (ebd., 10).

Das Überarbeiten kann sich auf Gedächtnisinhalte, auf gerade erst formulierte Textstücke (so im Modell in HAYES U. FLOWER 80) oder auf eine Rohfassung beziehen (JECHLE 92, 36f). Nach HAYES U. FLOWER wird sich dabei an sprachlichen Konventionen, der intendierten Bedeutung, der Verständlichkeit für Leserinnen und der Akzeptanz durch sie orientiert (80, nach JECHLE 92, 38).

### 2.3 Phasen bei der Erstellung eines wissenschaftlichen Textes

In diesem Kapitel soll das Schreiben eines wissenschaftlichen Textes v. a. nach technischen (im Sinne von ‚handwerklichen‘) und medialen (im Sinne von ‚wissensvermittelnden‘) Aspekten untersucht werden, mit dem Hintergrund, diese durch den Computer zu unterstützen. Kreative Aspekte werden dabei nur indirekt berücksichtigt (s. dazu jedoch v. WERDER 93).

Nach MEADOWS ist es für Wissenschaftlerinnen wichtiger, Zeitschriftenartikel zu lesen als zu schreiben (74, 111). Er zitiert eine Untersuchung über amerikanische Psychologinnen, nach der nur die Hälfte der Artikel, die in 25 Psychologiezeitschriften kurz vor der Untersuchung veröffentlicht worden waren, von mehr als einem Prozent der Befragten gelesen worden waren. Nur 11 von 429 Artikeln waren von mehr als zehn Prozent der Befragten gelesen worden. Weniger als ein Drittel der Befragten hatten überhaupt einen Artikel aus den betrachteten Zeitschriften gelesen (ebd., 105, aus APA 63/65/69). Das Schreiben soll daher in diesem Abschnitt genauer als das Lesen behandelt werden.

Ein wissenschaftlicher Text behandelt eine wissenschaftliche Untersuchung. Diese muß nach ECO folgende Kriterien erfüllen:

1. Sie beschäftigt sich mit einem (konkreten oder abstrakten) genau benannten Gegenstand.
2. Sie deckt neue Aspekte dieses Gegenstandes auf oder beleuchtet schon bekannte Aspekte neu.
3. Sie kann von anderen weiterverwendet werden.
4. Sie ist nachprüfbar. (92, 40ff)

Gängige Anleitungen zum wissenschaftlichen Schreiben wie die von ECO (92) und die von POENICKE (88) verfolgen meist keine analytischen, sondern eher pädagogische Absichten. Sie wenden sich vorwiegend an Studienanfängerinnen oder an Studentinnen, die vor dem Verfassen ihrer Abschlußarbeit stehen. Das aus ihnen gewonnene, hier dargestellte Prozeßmodell des Schreibens stellt also einen idealen Ablauf dar, den in der Praxis kaum eine Wissenschaftlerin so ausführen wird. Dieser Ablauf wird hier dennoch bis ins Detail geschildert, weil davon ausgegangen wird, daß die teilweise pedantischen Anleitungen reale Probleme ansprechen, die in einem Computersystem flexibler gelöst werden können. Wie wissenschaftliches Schreiben kreativ angegangen werden kann, beschreibt v. WERDER in seinem bereits zitierten Buch (93).

Das wissenschaftliche Schreiben wird in der entsprechenden Literatur meist als Abfolge von Phasen beschrieben. (Für eine Übersicht siehe v. WERDER 93, 52.) Beispielsweise nennt ECO die

Phasen „Die Wahl des Themas“, „Die Materialsuche“, „Der Arbeitsplan und die Anlage der Kartei“, „Das Schreiben“, „Die Schlußredaktion“ (92, XVff). POENICKE gliedert seine Anleitung in „Das Sammeln und Ordnen von wissenschaftlichem Material“, „Entwurf und Gliederung des Manuskripts“, „Das Erstellen des Manuskripts“, „Die Literaturangabe“, „Das Literatur- oder Quellenverzeichnis (Bibliographie)“ und „Vorbereitung zur Veröffentlichung, Korrekturlesen und Registermachen“ (88, 9ff). Bei v. WERDER lautet die Abfolge „Entwicklung eines groben Schreibkonzeptes“, „Schaffung von Schreibstimuli am Material“, „Schaffung eines differenzierten Schreibkonzeptes“ und „Schreiben und Überarbeiten“ (93, 403). THEISEN schließlich unterscheidet „Planung“, „Vorarbeiten“, „Materialübersicht und Themenabgrenzung“, „Materialauswahl“, „Materialauswertung“, „Manuskript“, „Ergebnisgestaltung und Typoskript“ und „Veröffentlichung und Druck“ (93, XI).

Auch von Informatikerinnen, die wissenschaftliches Schreiben explizit mit dem Computer (in diesem Fall sogar mit einem Hypertextprogramm, dem Programm ‚Notes‘ für das *Andrew*-System der Carnegie-Mellon-Universität, s. Abschnitt 2.4.2 auf S. 21) unterstützen wollen, gibt es zugrundeliegende Phasenmodelle: Aufnahme von Wissen, Analyse des Wissens, Ordnung des Wissens und schließlich Linearisieren und Ordnen in Schrift (NEUWIRTH et al. 88, S. 122). TRIGG U. IRISH beschreiben den Einsatz des Hypertextsystems NoteCards am Xerox PARC zur Unterstützung des Verfassens wissenschaftlicher Texte. Sie sehen es als Hilfsmittel folgender Einzelaufgaben: Notizen machen, Organisieren und Strukturieren, Gliedern und Verwalten von Literaturverweisen und -listen (88, 89).

Das erwähnte Modell des Schreibens von JECHLE (s. S. 12) läßt sich mit den geschilderten Aufteilungen in Übereinstimmung bringen, wenn man JECHLES Phase 2, ‚Bereitstellen von Information‘, v.a. auf externe Quellen bezieht. Außerdem werden in den Anleitungen zum Schreiben noch weitere Punkte berücksichtigt, die auf Phase 5, ‚Durchsehen und Überarbeiten‘ folgen, wobei für unsere Betrachtung besonders das Veröffentlichende wichtig ist.

Hier nun die Phasen aus den Schreibanleitungen, die für diese Untersuchung zunächst unterschieden werden sollen, mit den entsprechenden Phasen aus der Schreibforschung in Klammern, ergänzt durch den Punkt ‚Veröffentlichung‘:

1. Literaturübersicht und Themenwahl  
(Repräsentation der Schreibaufgabe und Zielbildung)
2. Literaturbeschaffung und -auswertung  
(Bereitstellen von Information)
3. Ordnung (Anordnen von Information)
4. Schreiben (Einzelsprachliche Realisierung)
5. Endredaktion (Durchsehen und Überarbeiten)
6. Veröffentlichung

Diese Phasen werden nun im einzelnen aus Sicht der Schreibpraxis erläutert und nötigenfalls ergänzt. Dabei werden die jeweils benutzten Texteinheiten und ihre Benutzung beschrieben. Es sollte jedoch nicht vergessen werden, daß Schreiberinnen jeweils individuelle Schreibstrategien verfolgen (s. dazu MOLITOR 85).

### 2.3.1 Literaturübersicht und Themenwahl

Am Anfang einer wissenschaftlichen Arbeit steht ihr Gegenstand, das Thema. Wenn das Thema nicht vorgegeben ist, muß die Autorin schon vor Beginn ihrer eigentlichen Arbeit Literatur studieren. Dabei sieht sie, welche Themen in ihrem wissenschaftlichen Gebiet gerade aktuell oder schon behandelt sind und zu welchen es überhaupt Literatur gibt. Auch wenn ein Thema vorgegeben ist,

steht am Anfang einer wissenschaftlichen Arbeit eine Sichtungsphase. Diese beginnt mit der Untersuchung vorhandener selbstgeschriebener und fremder Materialien.<sup>4</sup> Sie gelingt am leichtesten, wenn die Autorin eine allgemeine Schriftgutablage pflegt (POENICKE 88, 87).

Aus Gründen der Aktualität wird eine Autorin für die Themenfindung v. a. auf Artikel aus wissenschaftlichen Zeitschriften zurückgreifen (s. Abschnitt 2.1.1 auf S. 6). Sie kann dabei auch allgemeine, ‚unwissenschaftliche‘ Literatur benutzen, z. B. Konversationslexika, Zeitungen und populäre Zeitschriften (THEISEN 93, 39ff), die sie in ihrer späteren Arbeit vielleicht nicht mehr verwenden wird. (Das gezielte Erschließen wissenschaftlicher Literatur wird in den nächsten Abschnitten behandelt.) Die Erkenntnis wächst bei der Auseinandersetzung mit dem Gegenstand der Untersuchung und der entsprechenden Literatur. Dadurch kann sich die Arbeitshypothese (als Aussage über den Untersuchungsgegenstand) verändern. Themenfestlegung und -bearbeitung erfolgen also nicht streng hintereinander (POENICKE 88, 84f).

Für die Literatursuche werden Verzeichnisse benutzt, die generell in Bibliographien und Kataloge unterschieden werden können. Bibliographien geben alle veröffentlichte Literatur einer Publikationsform, einer Herkunft oder eines Themas an. Es gibt rückblickende und periodische Bibliographien, selbständige und unselbständige, wobei letztere Teile von Sammelwerken, Zeitschriften und Monographien sind.<sup>5</sup> Gedruckte Kataloge von Bibliotheken können ebenfalls als Bibliographien angesehen werden. Weitere Beispiele sind das *Verzeichnis lieferbarer Bücher (VLB)* bzw. *Books in Print* in den USA, Verzeichnisse von Hochschulschriften und Zeitschriftenbibliographien (ebd., 63ff).

Kataloge nennen alle an einem bestimmten Ort, meist in einer Bibliothek, vorhandene Literatur. Bibliotheken lassen sich unterscheiden in wissenschaftliche und öffentliche, zwischen denen die Übergänge jedoch fließend sind. Wissenschaftliche Bibliotheken dienen Studium, Forschung und Lehre und arbeiten archivarisch, sortieren also keine veraltete Literatur aus. Zu ihnen gehören Universal- und Fachbibliotheken. Öffentliche Bibliotheken sortieren hingegen veraltete oder zerlesene Bücher aus (ebd., 51f).

In Bibliotheken kann eine Wissenschaftlerin nach Literatur suchen, zu der sie nähere Angaben hat oder die sich mit einem bestimmten Thema beschäftigt. Dabei helfen ihr formal/mechanische und inhaltliche Kataloge. Formal/mechanische Kataloge ordnen nach formalen Gesichtspunkten. Diese Merkmale sind jeweils offensichtlich und eindeutig. Inhaltliche Kataloge ordnen nach sachlicher, systematischer oder historischer Zusammengehörigkeit. Diese Merkmale sind uneindeutig und müssen aus dem Inhalt der jeweiligen Publikation erschlossen werden (ebd., 33f).

Der alphabetische Katalog ist der wichtigste formale, er enthält Namen von Personen (Verfasserinnen und Herausgeberinnen) oder Körperschaften, die zu einem Werk gehören. Er verweist auf den entsprechenden Standort. Dem systematischen Katalog liegt eine hierarchische Ordnung von Wissensgebieten zugrunde. Die Systematik unterscheidet im Gegensatz zur klassischen Wissenschaftssystematik auch nach formalen Gesichtspunkten, z. B. nach Literaturarten wie Zeitschriften und Bücher. Sie ist an den jeweiligen Bestand einer Bibliothek bezüglich Schwerpunkten und Lücken angepaßt (ebd., 55f u. 37).

Der systematische Katalog führt teilweise auch formal zusammengehörige Literatur gemeinsam auf. Er eignet sich für die thematische Suche. Das gilt noch mehr für den Schlagwortkatalog. Seine Einträge, die Schlagwörter, sind zwar alphabetisch, also formal geordnet, sie wurden aber aus dem Inhalt von Schriften extrahiert. Der Schlagwortkatalog enthält viele Querverweise, weil er nicht hierarchisch aufgebaut ist. Er verweist meist auf Einträge aus dem alphabetischen oder dem systematischen Katalog (ebd., 55f).

4 Der Begriff des ‚Materials‘ soll in dieser Arbeit ähnlich unterbestimmt wie der des ‚Dokuments‘ bleiben. ‚Material‘ umfaßt hier allgemein Schriftstücke, die für die Produktion von Dokumenten benötigt werden.

5 Zusätzlich existieren auch Bibliographien von Bibliographien.

Eine 1970 veröffentlichte Untersuchung des AMERICAN INSTITUTE OF PHYSICS (AIP) beschreibt, wie Wissenschaftlerinnen, in diesem Fall Ärztinnen und Chemikerinnen, auf Literatur aufmerksam werden (s. Tabelle 1). Sie benutzen also v. a. aktuelle Artikel. In der Tabelle tauchen aber auch Hilfsmittel für die Literaturrecherche auf, die besonders Anfängerinnen im wissenschaftlichen Metier empfohlen werden, nämlich thematisch geordnete Bibliographien und Bibliothekskataloge. POENICKE nennt weiterhin Handbücher und terminologische Glossare des entsprechenden Faches (88, 49).

Method	Ranking order	
	Physicists	Chemists
Following up citations in relevant papers	1	1
Keeping up by reading current publications	2	2
References from conversations with colleagues	3	5
Unpublished materials obtained from colleagues	4	10
Use of abstract journals and indexes	5	3
References from correspondence with colleagues	6	9
Use of personal index or record	7	4
Use of a textbook	8	6
Use of a subject bibliography	9	7
Use of a library card-index	10	11
Consulting internal reports of one's own organisation	11	8
Asking librarian or information officer	12	12

*Tabelle 1.* Wie Ärztinnen und Chemikerinnen auf Literatur aufmerksam werden (MEADOWS 74, 94, nach AIP 70).

### 2.3.2 Literaturbeschaffung und -auswertung

Nach oberflächlicher Sichtung von Literatur und Formulierung einer Arbeitshypothese wird diese präzisiert und das Material stärker eingegrenzt und intensiver bearbeitet. Auch in dieser zweiten Phase kann eine Autorin auf Material stoßen, welches die der Arbeit zugrundeliegende These umwirft (POENICKE 88, 86f). In der schon zitierten Untersuchung des AIP haben Physikerinnen die folgende Wichtigkeit von Informationsquellen angegeben: archivierte Literatur (Bücher und Zeitschriftenartikel) an erster Stelle, für die daher die relative Wichtigkeit 1,0 festgelegt wurde. Danach folgen persönliche Kontakte mit der relativen Wichtigkeit 0,8, Vorträge auf Treffen mit 0,6, nicht-archivierte (Graue) Literatur mit ebenfalls 0,6, Abstract journals (Zeitschriften mit Zusammenfassungen von Aufsätzen aus anderen Zeitschriften) mit 0,4 und die restlichen Informationsquellen mit einer relativen Wichtigkeit von 0,1 (MEADOWS 74, 94, nach AIP 70).

Nachdem Literatur identifiziert und beschafft wurde, muß sie ausgewertet werden. Um zu entscheiden, ob es sich lohnt, ein Buch zu lesen, können Verzeichnisse und andere spezielle Teile wie Inhalts-, Namen- und Sachverzeichnisse, die Einleitung und Zusammenfassungen studiert werden (POENICKE 88, 20). Eigene oder kopierte Literatur kann durch Markieren von Schlüsselbegriffen, Definitionen und Zusammenfassungen und Einfügen von Randnotizen erschlossen werden (ebd.).

Der erste Schritt vom Lesen zum Schreiben ist das Exzerpieren auf z. B. Karteikarten. (Weitere Papierformate, die für das Exzerpieren geeignet sind, werden im nächsten Abschnitt beschrieben). Das Exzerpieren beinhaltet nicht nur das Abschreiben von Ausschnitten aus Quellen als Zitate, sondern auch die Auswertung, Kommentierung und Zuordnung von Gedanken (ebd., 18f). Auf eine



Exzerptkarte zu einer Quelle werden neben dem Haupttext ein Titel als Überschrift, (meist an den Rändern) Seitenzahlen, Stichwörter und Kurzkomentare notiert (ebd., 31).

Nach ihrem Inhalt lassen sich weiter folgende Aufzeichnungen unterscheiden: Literaturkarten identifizieren Quellen, nennen gegebenenfalls ihren Standort in der lokalen Bibliothek und enthalten einen Kommentar. Karten mit Exzerpten aus Sekundärquellen sind zu den schon beschriebenen aus Primärquellen analog, werden aber unter der jeweiligen Primärquelle eingeordnet, auf die sie sich beziehen. Alphabetisch abgelegte Schlagwörter definieren einen Begriff unter Angabe der Quelle (ebd., 45ff). Außerdem gibt es Ideenzettel, die dem Festhalten von spontanen Gedanken dienen, die später verwertet werden können (ebd., 88).

### 2.3.3 Ordnung

Neben der im Zusammenhang mit der Themenfindung schon erwähnten allgemeinen Schriftgutablage ist für die Erstellung eines speziellen Schriftstückes eine Projektablage nötig. Der Arbeitspunkt ‚Ordnen‘ umfaßt nämlich die Ablage von Material während des Lesens und die Gliederung von solchem Material als Vorbereitung für das Schreiben. Notizen aus und über Literatur fallen in einer Reihenfolge an, die hauptsächlich arbeitsorganisatorisch bestimmt ist.

Da die Reihenfolge ihrer späteren Verwendung am Anfang noch nicht bekannt ist, sollten Notizen zunächst in eine Ablage eingeordnet werden, die nicht nur für die Arbeit an einem Thema, sondern für ein ganzes Gebiet genutzt werden kann. MOON nennt eine solche Ablage ‚Persönliches Informationssystem‘ und beschreibt ein solches System so (mit HEEKS 86, 51) so:

In a personal information system, the user both stores and retrieves the information, in contrast to the multi-user systems in libraries. This means that personal information systems are idiosyncratic in design, the stored information is biased towards the user's interests, and subjective notes and comments are often found. The user is also an information amateur. (MOON 88, 266)

MOON betont außerdem, daß ein solches System nicht nur die Benutzung von Literatur unterstützen, sondern auch dabei helfen soll, Informationen aus Texten in persönlichen Strukturen zu ordnen (ebd., 271).

POENICKE nennt folgende Anforderungen an ein Ablagesystem: Es soll dynamisch und zukunftssicher sein, also auch neue Erkenntnisse sich in der Ordnung niederschlagen lassen. Es sollte Unterteilungen besitzen. Einzelinformationen müssen in großen Zusammenhängen auffindbar sein und in neue einsortiert werden können. Materialien wie Kopien, Zeitungs- und Zeitschriftenausschnitte müssen ebenfalls einsortiert werden können (ebd., 21f). Als Ablagen scheiden daher fest geheftete oder gebundene Schreibmaterialien aus wie Schreibhefte und Notizbücher (ebd., 22).

Eine Ablage von Karteikarten in -kästen hat hingegen die Vorzüge, daß sie übersichtlich und zugänglich ist, wobei diese Aspekte in einem Spannungsverhältnis zueinander stehen. Übersichtlichkeit verlangt z.B. wenige Unterteilungen, wodurch einzelne Karten schwerer gefunden werden können. Die Benutzung von Karteikarten und -kästen hat den Nachteil, daß Kasteneinteilungen und ihre Reihenfolge vor der Benutzung festgelegt werden müssen, wenn die Ordnung länger gültig sein soll. Außerdem sind Karten schlecht kopierbar oder maschinell beschreibbar. Schließlich benötigt eine Kartenablage relativ viel Platz (ebd., 24f).

Ähnlich wie Bibliothekskataloge (s. S. 15) lassen sich auch persönliche Ablagen entweder formal/mechanisch oder inhaltlich ordnen. Die alphabetische Ablage erfolgt zunächst nach den Namen der Verfasserin des Werkes, auf das man sich bezieht, dann nach dem Titel. Hinter die Werke einer Autorin kann die dazugehörige Sekundärliteratur eingestellt werden. Diese Art der Ablage ist einfach durchzuführen, bietet jedoch keine inhaltliche Erschließung und zerstört Zusammenhänge.

Diesem Nachteil kann durch eine zusätzliche systematische oder Stichwortkartei abgeholfen werden (ebd., 35f).

Eine ausschließlich systematische private Ablage ist für das wissenschaftliche Arbeiten nicht geeignet, weil sie generell komplex und starr ist und eine zumindest grundlegende Kenntnis des gesamten Arbeitsgebietes voraussetzt. Sie muß auf den genauen Zweck zugeschnitten sein, z. B. bezüglich ihrer ‚Auflösung‘ als Anzahl von Notizen in den kleinsten Ordnungseinheiten. Nicht nur bei zu starker Untergliederung ist die Zuordnung von Karten zu Einteilungen oft uneindeutig (ebd., 37ff). Dem kann jedoch mit Verweiszetteln abgeholfen werden, die aus einem Systemfach in ein anderes verweisen (ebd., 88).

Eine weitere Möglichkeit der Ablage ist die Kennzeichnung von Materialeinheiten durch Schlagwörter und deren anschließende alphabetische Sortierung. Diese Ablageform verbindet eine inhaltliche Erschließung mit leichtem Zugriff. Allerdings ist die Festlegung von Schlagwörtern schwierig, weil die Fachterminologie bekannt sein und auf Synonyme geachtet werden muß (ebd., 41f). Auch bei Schlagwörtern gibt es das ‚Auflösungsproblem‘: Der zu wählende Grad der Konkretheit eines Schlagwortes bereitet zunächst Schwierigkeiten .

#### 2.3.4 Schreiben

Nach POENICKE stellt das Formulieren eine der schwierigsten Phasen beim Erstellen einer wissenschaftlichen Arbeit dar, welche auch die letzte Eingrenzung des Themas umfaßt. Es sollten daher so früh wie möglich Prototypen der Einleitung und Rohentwürfe für einzelne Kapitel geschrieben werden. Diese Texte geben einen konkreten Rahmen vor. Sie müssen mit dem Fortschritt der Arbeit immer wieder überarbeitet werden (88, 83).

Außer der Einleitung besteht eine wissenschaftliche Arbeit (im Maximalfall) noch aus Titelblatt, Widmung, Geleitwort und Motto, dem Vorwort, Inhaltsverzeichnis, Disposition (Aufriß der Denk- und Arbeitsschritte), dem Literaturverzeichnis, dem Verzeichnis von Abkürzungen und Formelzeichen, nach der Einleitung dem Durchführungsteil und Schluß, schließlich aus ergänzenden Materialien (Statistiken, Tabelle und Tafeln) und dem Namen- und Sachverzeichnis (ebd., 107).

Die klassischen Manuskriptformen einer wissenschaftlichen Arbeit sind ‚Zettelmanuskript, Rohmanuskript, Vorentwurf zur Reinschrift und eigentliche Reinschrift‘ (ebd., 117). Ohne Textverarbeitung müssen diese Manuskripte jeweils vollständig neu geschrieben werden. Sie sollten viel Rand für Korrekturen und Anmerkungen einer Beurteilerin enthalten (ebd., 121). Das Zettelmanuskript entsteht, wenn Materialien für einen Text in einer Gliederung geordnet werden, wie dies im letzten Abschnitt vorgeschlagen wurde, und in die Gliederung jeweils Übergangstexte von einem Punkt zum nächsten einfügt werden (ebd.).

Das Rohmanuskript ist der erste durchgängige Text. Zitate und Quellen werden nur soweit eingefügt, wie es für das Wiederfinden durch die Autorin selbst nötig ist (ebd., 118). Wenn in das Rohmanuskript Zitate, Zitatbelege und Fußnoten eingefügt werden, entsteht der Vorentwurf zur Reinschrift (ebd.). Zitate werden aus den folgenden Gründen in ein Manuskript eingefügt (in absteigender Reihenfolge): Aktualität, Hinweis für die Leserin, Überzeugungskraft, Anerkennung, soziale Übereinkunft, benutzte Methoden, Techniken und Daten und schließlich (negative) Kritik (BROOKS 85, nach MOON 88, 271).

Zum wissenschaftlichen Umgang mit Quellen gehört, daß Übernahmen aus ihnen und eigene Gedanken streng getrennt werden: ‚Unterlassung einer Literaturangabe kann den Verdacht des Plagiats nach sich ziehen‘ (POENICKE 88, 141). Zitate dürfen zwar geringfügig modifiziert werden, wenn dies ihren Sinn nicht entstellt, die Modifikationen müssen aber für die Leserin offensichtlich sein. Einfügungen werden z. B. durch eckige Klammern, Auslassungen durch drei Punkte kenntlich gemacht (ebd., 43). Fußnoten verweisen dabei auf die entsprechenden Quellen. Fußnoten werden

außerdem eingesetzt, um auf andere Teile desselben Dokuments oder auf Informationen zu verweisen, die von der Hauptargumentation abweichen (ebd., 135).

Quellenverweise können auch als Kurzbelege ausgeführt werden, die auf einen Eintrag im Literaturverzeichnis verweisen. Sie bestehen aus dem Namen der Autorin(nen), bei mehreren zitierten Werken derselben Autorin(nen) zusätzlich aus dem abgekürzten Titel oder (wie in dieser Arbeit) dem Erscheinungsjahr (ebd., 141). Nach dem *MLA Style Manual* erfolgt der Kurzbeleg von Quellen mit feststehender Gliederung (z.B. sakraler Texte und Dramen) durch Nennung von optional der Verfasserin, dem Kurztitel und der entsprechenden Gliederungsteile in arabischen Zahlen, z.B. „*Ilias* 9.19“ (ACHTERT U. GIBALDI 85, 178). Generell ist die Angabe von Quellenverweisen und die Gestaltung eines Literaturverzeichnisses von Disziplin zu Disziplin verschieden.

### 2.3.5 Endredaktion

Man erhält eine Reinschrift, wenn man dem Vorentwurf eine Literaturliste und eventuell ein Register mit den entsprechenden Seitenzahlen anfügt (ebd., 120). Gewöhnlich enthält ein Literaturverzeichnis nur die zur Erstellung einer wissenschaftlichen Arbeit benutzten Quellen. Es kann auch darüber hinausgehen: „Angesichts der wachsenden Informationsüberflutung“ sieht POENICKE in der Erstellung einer Bibliographie für ein bestimmtes Thema schon eine eigenständige wissenschaftliche Leistung. Nichtbenutzte Quellen sollten darin entsprechend gekennzeichnet werden. Eine besondere Form des Literaturverzeichnisses ist das annotierte, welches zu jedem aufgelisteten Werk Anmerkungen enthält (88, 146).

Gewöhnlich werden die Einträge in einem Literaturverzeichnis zunächst alphabetisch nach Namen von Autorin(nen), Herausgeberin(nen) oder Körperschaften sortiert, bei jeweils mehreren Werken nach Erscheinungsjahr oder Titel (ebd., 147f). Die Einträge enthalten im einfachsten Fall die schon erwähnten Verantwortlichen, den Titel, den Verlag, Erscheinungsort und -jahr, bei unselbständig erschienener Literatur (s. S. 7) die bibliographischen Angaben zur dazugehörigen selbständigen Veröffentlichung und die Seitenzahl. Das Register bietet eine andere Art des Zugriffs auf den Text als das lineare Inhaltsverzeichnis. Es gibt Personen- und Sachregister, die oft zusammengefaßt werden. Ein Register kann gewöhnlich nur die Verfasserin eines Textes selbst erstellen (POENICKE 88, 190ff).

### 2.3.6 Veröffentlichung

Bei der traditionellen Publikation wird ein Manuskript einer Autorin beim Verlag neu gesetzt, auch wenn es mit einem Computer erstellt wurde. Die typographische Gestaltung des Manuskriptes ist daher überflüssig (BRÜGGEMANN-KLEIN 89, 148). Bei Publikationen, bei denen es auf Aktualität und wegen geringer Auflage auf geringe Kosten ankommt, wird die Vorlage einer Autorin unverändert für den Druck übernommen, als ‚Camera ready copy‘. Dafür bietet sich die Erstellung mit dem Computer an, um seine typographischen Möglichkeiten wie mathematische Formeln, verschiedene Schriften etc. benutzen zu können. Durch die Übertragung des Satzes auf die Autorin entsteht ihr jedoch mehr Arbeit.

Das Format der Druckvorlage wird jeweils von den Herausgeberinnen vorgegeben (ebd.). Um Text, der bereits von der Autorin elektronisch verfaßt wurde, aber trotzdem von professionellen Setzerinnen gesetzt werden soll, nicht neu erfassen zu müssen, wird er in einen Satzcomputer übernommen und dort neu gesetzt. Das führt u. a. zu einer einheitlichen Gestaltung im Gegensatz zum Druck nach Camera ready copies (ebd., 149).

Nach dem Satz wird ein Text mittels eines bestimmten Prozederes, an dem Autorin und Verlag beteiligt sind, korrigiert. Teilweise werden noch im Verlag durch die sogenannte Hauskorrektur die größten Satzfehler entfernt. Der gesetzte Text wird dann fortlaufend ohne Umbruch auf sogenann-

ten Fahnen an die Verfasserin zurückgeschickt. Sie korrigiert auf den Fahnen mittels festgelegter Korrekturzeichen und gibt sie wieder an den Verlag. Danach erhält sie entweder einen neuen Fahnenabzug oder einen Umbruchbogen mit der endgültigen Seitenaufteilung, auf dem sie Verweise mit Seitenzahlen einträgt. Sie gibt den Umbruchbogen schließlich durch einen Vermerk (das ‚Impri-matur‘) frei (ebd., 183f). Es handelt sich also insgesamt um einen aufwendigen Prozeß mit einigen Iterationen.

## 2.4 Schreiben mit dem Computer

### 2.4.1 Textverarbeitung und Persönliche Informationssysteme

Textverarbeitung auf dem Computer setzt oft schon bei der Formulierung von Ideen ein. Das gaben bei einer Untersuchung von RIEHM et al. 33,9% der befragten Autorinnen an (s. Abbildung 3). Bei Benutzung eines Computers wird die auf S. 18 erwähnte, von POENICKE vorgenommene Unterscheidung in Roh- und Reinschrift überflüssig, wie er selbst zugibt (88, 118).<sup>6</sup> Durch den Computer werden damit Veränderungen in Gang gesetzt, an deren Ende stehen könnte, die Übergänge zwischen *allen* Formen eines Dokumentes fließend zu gestalten.<sup>7</sup>

Vorformen einer Publikation stellen in einigen Hypertextsystemen bloße Versionen eines Dokumentes dar, wie in der noch folgenden Beschreibung von solchen Systemen deutlich werden wird. Klassische Textverarbeitungen präsentieren die jeweils letzte Version eines Textes im Computer hingegen als die einzige: Im Gegensatz zum Schreiben auf Papier, bei dem Korrekturen zwischen und über altem Text angebracht werden, sind diese auf dem Bildschirm nicht sichtbar (ebd.). Versionen können meist nur durch Speichern von datierten Dateien erhalten bleiben (RIEHM et al. 92, 28), automatisch gesichert wird höchstens die letzte Version als ‚Backup‘.

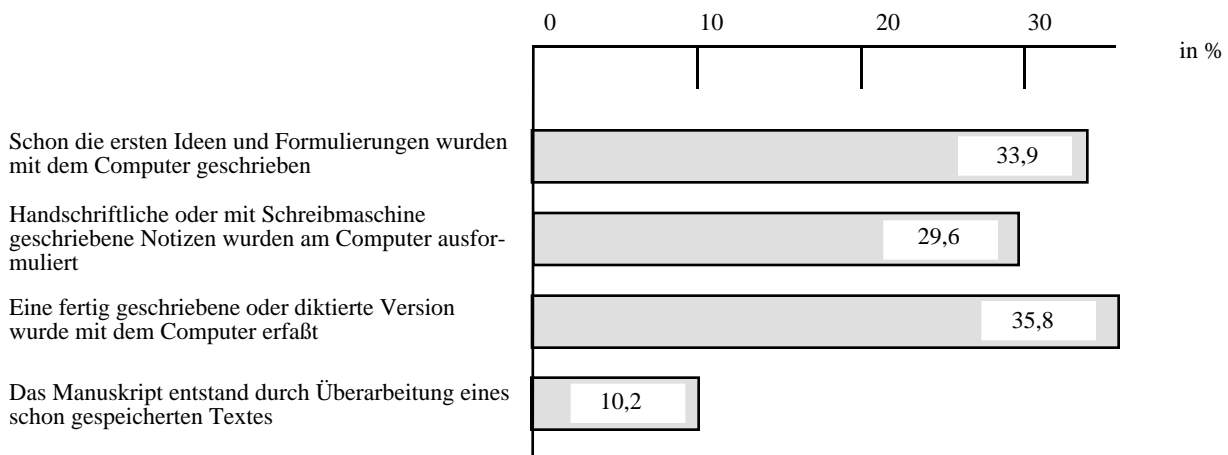


Abb. 3. Umfang der Computerunterstützung, n = 276 elektronische Manuskripte ( RIEHM et al. 92, 55)

Neben dem Namen kann nur nach dem Inhalt unterschieden werden, ob man eine Version eines alten oder ein neues Dokument vor sich hat. LEVY macht dies am Beispiel eines Dokumentes deutlich, welches man ändert und unter einem neuen Namen speichert. Es könnte sich dann um ein eigenständiges Dokument handeln, wenn es ein anderes Thema oder eine andere Zielgruppe als das Ursprungsdokument hat (94, 26).

<sup>6</sup> Populäre Textverarbeitungsprogramme wie Microsoft Word bieten stattdessen mehrere Sichten auf ein Dokument, z.B. eine ‚Normal-‘ und eine ‚Layoutsicht‘.

<sup>7</sup> Diese Arbeit beschreibt Hypertextsysteme, die genau das erreichen können.

Textverarbeitung wird von Autoren weniger wegen der größeren typographischen Möglichkeiten, sondern v.a. wegen des einfacheren Umgangs mit Dokumenten benutzt, selbst wenn diese nicht veröffentlicht werden sollen (BRÜGGEMANN-KLEIN 89, 152). Die Verwaltung von Materialien als ‚Rohstoffe‘ für Texte sind die Hauptaufgaben von Persönlichen Informationssystemen (s. S. 17). Mit diesen Systemen können weiterhin Bibliographien aus vorhandenen Daten erstellt, Nachdrucke angefordert und angefragte und erhaltene Artikeln aufgezeichnet werden. Außerdem kann auf externe elektronische Datenquellen zugegriffen werden. Erweiterte Systeme könnten darüber hinaus noch Notizen und Exzerpte verwalten (MOON 88, 271).

#### 2.4.2 Hypertextsysteme

Mit Textverarbeitungsprogrammen können Materialien und Dokumente zwar flexibel erstellt und gehandhabt werden, sie dienen aber immer noch der Erstellung von ‚Papers‘, wie Artikel nach ihrem materiellen Träger im angelsächsischen Sprachraum auch genannt werden. Die Unterstützung der Erstellung von Texten ist ein Anwendungsfeld allgemeiner und speziell dafür entwickelter Hypertextsysteme. Wie wir gesehen haben, bestehen Dokumente nicht nur aus Text, sondern auch aus Verweisen auf andere Dokumente, Verzeichnisse etc. Hypertextsysteme erlauben die Explizierung solcher Verweise als ‚Links‘, benötigen jedoch auch eigens dafür und im besten Fall damit erstellte Dokumente, die dann nur noch mit dem System gelesen werden können. (Für eine Definition von ‚Hypertext‘ siehe S. 30.)

Im folgenden werden zwei Beispiele der Anwendung von Hypertextsystemen für das Verfassen wissenschaftlicher Dokumente beschrieben, ‚NoteCards‘ vom Xerox PARC und die Erweiterung ‚Notes‘ für das Lehr- und Forschungssystem ‚Andrew‘ der Carnegie-Mellon-Universität. Sie können als Vorstufen für ein noch zu entwickelndes globales Hypertextsystem für die Unterstützung von Autorinnen dienen.

Die Entwicklerinnen von ‚Notes‘ sind davon ausgegangen, daß beim Schreiben ein „komplexes Netzwerk von Ideen“, welches sich teilweise erst beim Formulieren entwickelt, in eine lineare Form gebracht wird (NEUWIRTH et al. 88, 121). Schreiberinnen überführen aber nicht nur Ideen in ausformulierte Texte:

Writing actual prose represents a bottom-up planning procedure. As in other complex tasks, writers engage in a combination of top-down and bottom-up planning. (ebd., 132)

Je weniger eine Schreiberin über ein Gebiet weiß, über das sie schreibt, desto öfter muß sie diese Ideen reorganisieren (ebd., 121).

Für das Notieren von Ideen wurden für den Entwurf von Notes Karteikarten angenommen. Sie dienen dazu, eigenes Wissen, Quellen und Reaktionen auf fremde Texte aufzuschreiben und diese Notizen in eine gewünschte Ordnung zu bringen. In Notes wird eine digitale Version der amerikanischen Standardnotizkarten der Größe 3" x 5" benutzt (ebd., 123f), die also etwa kleiner als die hier gebräuchlichen DIN A6-Karten sind. NEUWIRTH et al. nennen als Probleme dieser Notizkarten, daß darauf notierte Ideen aus dem Zusammenhang gerissen sind, daß das Exzerpieren auf sie leicht zu Fehlern führt und sie immer nur in jeweils einer Ordnung abgelegt werden können (88, 124), wenn man keine Verweiszettel benutzt.

Notes verwaltet den Ursprungstext, über den Notizen erstellt werden sollen, und die dazugehörigen Notizen. Die Notizen werden in einem vernetzten System gespeichert. Für eine Bearbeitung muß also eine Netzverbindung existieren (ebd., 126). Wenn ein Text in Notes vorliegt, kann ein Link zwischen einem Textausschnitt und einer Notiz gesetzt werden (ebd., 124f). Dies ist gleichzeitig der einzig mögliche Link (ebd., 130). Nach dem Erstellen erscheint im Ursprungstext ein Icon ähnlich einem Fußnotenzeichen (ebd., 126f). Da bei der Entwicklung von Notes davon ausgegangen wurde, daß zunächst nur wenige Texte elektronisch verfügbar sind, wurde es eher für das Orga-

nisieren von Notizen als zum Verweisen auf Quellen entworfen. Notizen können auch gedruckt werden (NEUWIRTH et al. 88, 125).

Notizen erhalten in Notes von ihrer Verfasserin immer einen Titel. Von geschriebenen Notizen einer Benutzerin wird automatisch eine Liste erstellt. Es gibt außerdem eine systemweite Liste, die alphabetisch zuerst nach Autorinnennamen des entsprechenden Ursprungstextes und dann nach Notiznamen sortiert ist. Notizen können zusätzlich in mehrere Hierarchien eingeordnet werden (ebd., 126 u. 130). Zum Ansehen werden sie aus einem Verzeichnis ausgewählt und können auch zwischen Verzeichnissen verschoben werden (ebd., 127). Notizen können außerdem zu Klassen zusammengefaßt werden. Bei diesen Klassen handelt es sich eigentlich um Attribute, die Karten zugewiesen werden (ebd., 128). Notizen können nach Inhalt, Klasse, Autorin des Ursprungstextes, Titel des Ursprungstextes, Erstell- und letzter Änderungszeit gesucht werden (ebd.).

Das Design von Notes löst nach Ansicht der Entwicklerinnen die von ihnen beschriebenen Probleme mit Karteikarten folgendermaßen: Keine Zitate, sondern nur eigene Umformulierungen müssen aufgeschrieben werden, weil Quellen per Link verfügbar sind. Notizen von eigenen Ideen zu Teilen von Primärliteratur können kurz gehalten werden, weil der jeweilige Kontext ebenfalls schnell verfügbar ist (ebd., 124f). Die Entwicklerinnen sind jedoch mit der für Notes benutzten Datenstruktur unzufrieden und geben an, daß sie sie zu einer allgemeineren Hypertextdatenbank erweitern wollen (ebd., 131).

„NoteCards“ ist ein Hypertextsystem, welches am Xerox PARC entwickelt wurde (s. HALASZ 88). Es bietet den Benutzerinnen wiederum Karteikarten, von denen es auch seinen Namen hat, als Analogie zu den schon beschriebenen 3" x 5"-Pappkarten. Karten bestehen in NoteCards aus einem Titel und dem Inhalt ihrer Hauptfläche. Die Hauptfläche kann einen Text, eine Zeichnung oder einen Graphen enthalten. Es können auch andere Kartenarten definiert werden. Karten werden auf dem Bildschirm als Fenster dargestellt, von denen sich auch mehrere gleichzeitig auf dem Schirm befinden können. Als zweites Grundobjekt bietet NoteCards Links als Verbindung zwischen Karten. Benutzerinnen können Links beschriften und damit Linktypen schaffen (TRIGG U. IRISH 88, 91).

Im System bereits vorhanden sind die speziellen Karten Karteikasten (Filebox), Browser und Dokument. Ein Karteikasten enthält andere Karten (also auch Karteikästen) in einer hierarchischen Ordnung, welche aus einem gerichteten, azyklischen Graphen besteht. Für jede Karte gibt es mindestens einen Karteikasten, der sie enthält, auch für Karteikasten selbst. Jede Karte kann in mehr als einem Karteikasten abgelegt werden (ebd., 92).

Ein Browser stellt Abhängigkeiten zwischen Karten, die von der Benutzerin als Links eingefügt wurden, graphisch als Hierarchie dar. In dem Hierarchiegraph bilden die Karten die Knoten und die Links die Kanten. Im Browser können Karten erzeugt und bewegt werden. Dokumentkarten dienen dazu, mehrere Karten halbautomatisch in eine bestimmte Reihenfolge zu bringen. Für die Linearisierung wählt eine Benutzerin aus dem Netzwerk ihrer Karten eine Startkarte, Typen von Links, die von der Startkarte aus verfolgt werden sollen und Formatierungsoptionen für den entstehenden Text. NoteCards kopiert dann den Inhalt aller besuchten Karten in die Dokumentenkarte, wobei die Verbindung des jeweiligen Textes zu den ursprünglichen Karten verlorengeht (ebd.).

NoteCards wurde am Xerox PARC von Wissenschaftlerinnen verschiedener Gebiete benutzt: von Anthropologinnen, Physikerinnen, Linguistinnen und Informatikerinnen. TRIGG U. IRISH beschreiben eine Untersuchung über zwanzig dieser Forscherinnen, bei der sie über ihre Erfahrungen mit dem System für das Schreiben von kurzen Aufsätzen, Präsentationen, Doktorarbeiten etc. befragt wurden. Mit neun der Forscherinnen wurden eingehende Interviews geführt, wobei sie ihre NoteCards-Umgebung jeweils selbst vorgestellt haben. Die Untersucherinnen konnten außerdem auf die Umgebungen aller Wissenschaftlerinnen zugreifen (ebd., 90).

Aus den Aussagen der von TRIGG U. IRISH befragten Wissenschaftlerinnen ergaben sich folgende Kriterien für die Benutzung von NoteCards:

1. *Komplexität der Struktur eines Dokumentes*: Je höher diese ist, desto wahrscheinlicher ist der Einsatz von NoteCards im Vergleich zu anderen Schreibhilfen.
2. *Möglichkeit der Verwendung von Dokumenten, die sich schon in NoteCards befinden*: Ihr Vorhandensein macht den Einsatz wahrscheinlich.
3. *Aufwand der Bedienung von NoteCards*: Dieser ist größer als beim Einsatz eines Texteditors (ebd., 93f).

Da alle Karten in die Hierarchie eingebunden werden müssen, legen Autorinnen oft Fileboxen an, in der sie Karten ablegen, von denen sie noch nicht wissen, wo sie sie einordnen sollen (ebd., 96). Für größere Schreibprojekte werden Karten oft in mehreren Hierarchien gehalten (ebd., 100f). NoteCards wird auch zum Brainstorming benutzt. Dafür werden Gedanken entweder in einer einzigen Karte niedergeschrieben oder als Titel von Karten, die im Browser erzeugt werden (ebd., 97). Autorinnen erzeugen gewöhnlich nicht in einem Schritt aus einer alten Gliederung in einer Filebox oder einem Browser eine neue, um die alte sofort zu löschen, sondern gestalten den Übergang langsam. Daher müssen gleichzeitig verschiedene Sichten auf dieselbe Hypertextstruktur erlaubt sein (ebd., 101 und 105f).

An der Dokumentenkarte kritisieren die Benutzerinnen, daß die Verbindung des linearisierten Textes zu seinen Ursprungskarten zu inflexibel ist: Um den Inhalt einer Dokumentenkarte zu ändern, müssen die Ursprungskarten geändert und die Karte neu erzeugt werden. Es ist umgekehrt nicht möglich, über Änderungen im Dokument die Ursprungskarten zu ändern, von denen der geänderte Text stammt (ebd., 104f). (TRIGG U. IRISH geben an, daß dafür eine Outlinerfunktion nötig wäre. Eigentlich müßte aber nur die Verbindung von den Ursprungskarten zur Dokumentenkarte aufrecht erhalten werden, s. S. 71.)

HALASZ beschreibt die Ablage von Ideen in NoteCards als dreistufig: Ideen müssen auf eine Karte geschrieben, diese Karte muß mit einem Namen versehen und in eine Filebox abgelegt werden. Dabei tritt das, wie der sagt, „Problem der vorschnellen Organisation“ auf. Zunächst sind nämlich weder der Umfang einzelner Ideen, günstige Namen noch die Abhängigkeit zwischen Ideen bekannt (88, 845). Als Lösung schlägt er ‚Virtual composites‘ vor. Das sind Hypertextknoten, die erst zur Zeit ihrer Anforderung aus beliebigen anderen Knoten zusammengesetzt werden. Sie entsprechen also Datenbankabfragen (ebd., 846) bzw. Verweisen auf Daten, die noch nicht existieren müssen.

## 2.5 Computerunterstützung des wissenschaftlichen Schreibens

Der oben beschriebene Ablauf des wissenschaftlichen Schreibens hat sich bewährt und ist eingespielt. Auf ihm ruht ein Großteil der wissenschaftlichen Kommunikation als Mittel wissenschaftlichen Fortschritts und Ruhms für einzelne Wissenschaftlerinnen. Er ist allerdings auch so zeitaufwendig, daß er für einige Dokumententypen nicht durchlaufen werden kann, z.B. für technische Berichte. Er ist relativ teuer und trennt stark zwischen den einzelnen Abläufen und den Institutionen und Gruppen, die sie ausführen, z.B. zwischen Autorinnen und Leserinnen, obwohl erstere immer auch die andere Rolle einnehmen müssen. Er geht vom linearen Druck auf Papier aus, auch wenn von Anfang an Computer eingesetzt werden.

In dieser Arbeit wird ein System für das Erstellen, Organisieren und Publizieren von wissenschaftlichen Texten konzipiert. Organisieren umfaßt dabei das Bereithalten von altem und die Gliederung von neuem Text. Es steht hier zusammen mit dem Erstellen im Vordergrund. Funktionen und Institutionen des Publizierens wie Vergütung, Verlage und Zeitschriften werden also nur soweit

mitmodelliert, wie es für das Verständnis des Systems nötig ist. Als Nutzerinnen des Systems werden Wissenschaftlerinnen angenommen, weil deren Schreibtätigkeit relativ gut dokumentiert ist und ansatzweise auch schon über das Internet erfolgt. Das System soll verteilt sein, damit nicht nur Autorinnen, sondern auch Gutachterinnen und Leserinnen auf Texte zugreifen können. Die Vernetzung soll auf dem Internet basieren, der Standardnetzinfrastruktur für den wissenschaftlichen Bereich.

Zusammengefaßt ist Ziel dieser Arbeit ein Konzept eines verteilten Textsystemes im Internet für die billigere und schnellere Erstellung von wissenschaftlichen Texten durch Wissenschaftlerinnen. Um es umzusetzen, muß zunächst der ‚Ist-Zustand‘ festgehalten werden, also der an Papier orientierte Prozeß.

### 2.5.1 Beschreibung des Ist-Zustandes

Nach den theoretischen Modellen und den praktischen Beschreibungen des Schreibvorganges soll nun das wissenschaftliche Schreiben konkreter dargestellt werden. Dazu wird ein Objekt- und ein Prozeßmodell nach der ‚Object Modeling Technique‘ (OMT) von RUMBAUGH et al. (91) aufgestellt. Diese Methode wurde gewählt, weil ihre Diagrammarten aus klassischen Methoden des Software-Engineering entwickelt wurden: Das Objektdiagramm der OMT ist ein erweitertes Entity-Relationship-Diagramm (RUMBAUGH et al. 91, 271), und das Datenflußdiagramm wurde von der Structured Analysis-Methode von DEMARCO übernommen (RUMBAUGH et al. 91, 123).

Zunächst wird ein Modell der beim herkömmlichen Schreiben beteiligten Klassen von Objekten und Verbindungen zwischen ihnen aufgestellt, das Abbildung 4 zeigt. Das Modell enthält nur Klassen, keine Instanzen. Die Vererbung wird in einem OMT-Objektmodell durch Linien mit einem gleichschenkligen Dreieck dargestellt, dessen gleiche Schenkel sich in der Ecke treffen, die zur Oberklasse hinzeigt. Die anderen Linien zwischen Klassen stellen Assoziationen dar, die Relationen im relationalen Datenbankmodell entsprechen. Einfache Verbindungsenden bedeuten das genau einmalige Auftreten der entsprechenden Klasse in einer Relation. Ein ausgefüllter Kreis bedeutet hingegen, daß mehrere Objekte der so verbundenen Klasse mehrfach in der Assoziation vorkommen können. Rauten stellen Aggregationen dar, also ‚Besteht-aus‘-Beziehungen.

Im Schreibmodell in Abbildung 4 sind Übernahmen von Text von einem Dokument (einer Quelle) in ein anderes als benannte Assoziation modelliert. Positionen in einem Text, z.B. die Fundstellenangabe zu einer Quelle, werden hingegen als unbenannte Assoziationen dargestellt, die durch Schleifen an der entsprechenden Verbindungslinie befestigt sind. Das Attribut dieser Assoziationen, die jeweilige Position in einem Dokument, läßt sich weder dem Quelltext an dem einen noch dem Zielttext an dem anderen Ende zuordnen und ist daher an der Assoziation selbst befestigt.

Es folgt das Datenverzeichnis (Data dictionary), welches die einzelnen Objekte des Modelles in Stichworten beschreibt:

*Absatz:* Enthält den eigentlichen Text von Objekten der Klasse ‚Manuskript‘. Ein Absatz ist in Manuskripten jeweils einem Gliederungspunkt untergeordnet. Quellenangaben und Fußnoten beziehen sich jeweils auf einen Absatz.

*Bibliographie oder Katalog:* Unterklasse von ‚Verzeichnis‘. Verzeichnis von Literatur eines bestimmten Bereiches (z.B. Landes oder Faches) oder an einem bestimmten Ort (z.B. in einer Bibliothek). Eine Bibliographie oder Katalog besteht aus bibliographischen Angaben.

*Bibliographische Angabe:* Unterklasse von ‚Eintrag‘. Kennzeichnung eines Stückes Literatur nach Autorin(nen), Titel, Verlag, Ort und Jahr. Befindet sich in einer Bibliographie oder einem Katalog, geordnet nach der Autorin der Quelle. Zu jeder bibliographischen Angabe gehört ein Buch.



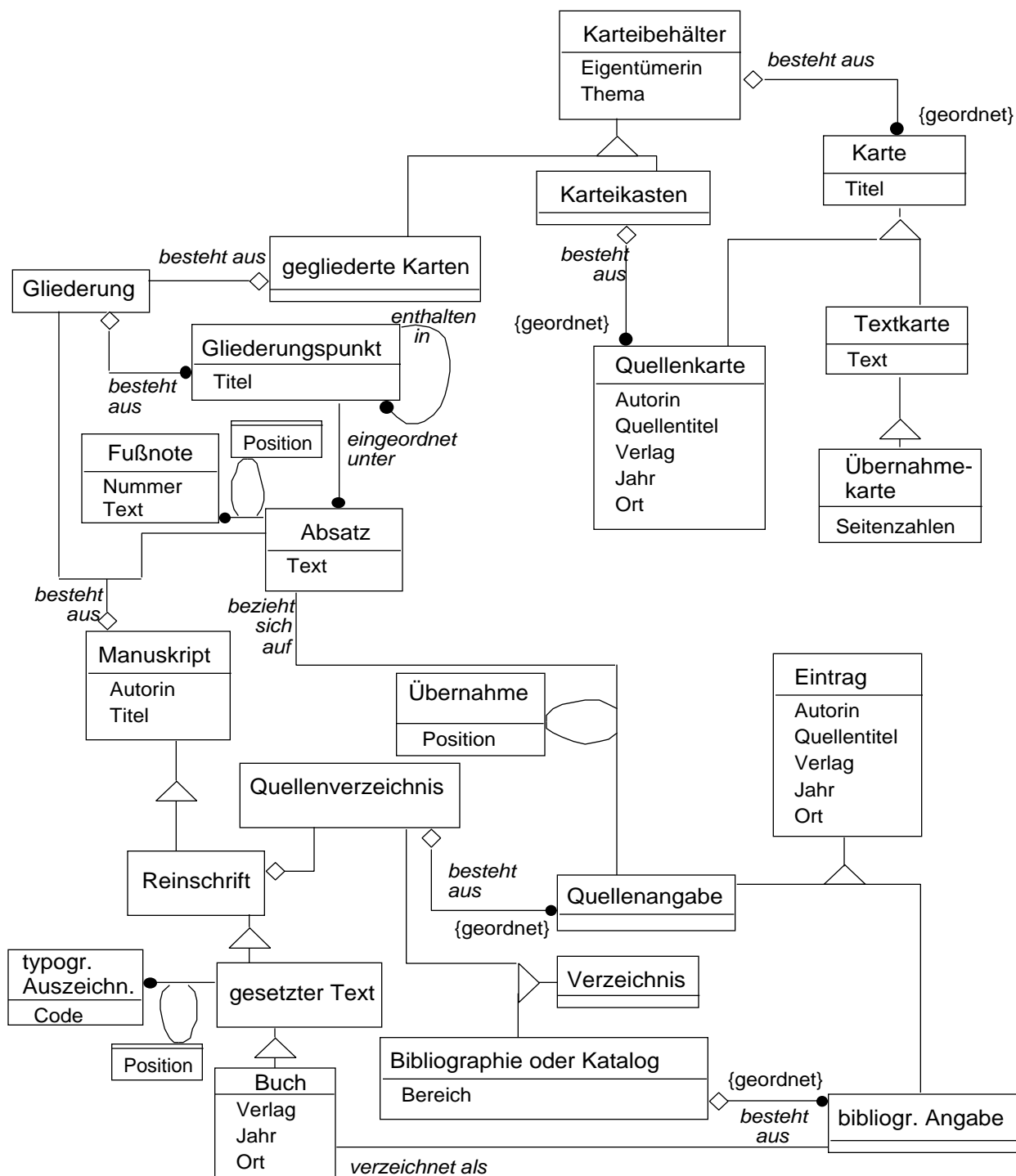


Abb. 4. Objekt-Diagramm zur Veröffentlichung eines wissenschaftlichen Dokumentes, modelliert nach OMT.

**Buch:** Unterklasse von ‚gesetzter Text‘. Veröffentlichtes Schriftstück (worunter auch Zeitschriftenartikel fallen). Identifiziert durch Autorin(nen), Titel, Verlag, Ort und Jahr. Zu jedem Buch gehört eine bibliographische Angabe.

**Eintrag:** Enthält Autorin, Quellentitel, Verlag, Erscheinungsort und -jahr einer Quelle.

**Fußnote:** Kurzer Text, auf den in einem Absatz verwiesen wird. Besteht aus dem Text und einer laufenden Nummer.

*Gegliederte Karten:* Unterklasse von ‚Karteibehälter‘. Zusammenfassung von Karten, die in eine bestimmte Reihenfolge gebracht wurden, die durch eine Gliederung festgelegt ist.

*Gesetzter Text:* Text unmittelbar vor der Veröffentlichung. Besteht aus dem Text einer Reinschrift und typografischen Auszeichnungen.

*Gliederung:* Ordnung eines Textes. Besteht nur aus Gliederungspunkten in einer festgelegten Reihenfolge.

*Gliederungspunkt:* Bestandteil einer Gliederung. Kann auf weitere Objekte derselben Klasse und auf Absätze verweisen.

*Karte:* Enthält einen Titel. Befindet sich in einem Karteibehälter.

*Karteibehälter:* Enthält geordnete Objekte der Klasse ‚Karte‘. Hat eine Eigentümerin und bezieht sich auf ein bestimmtes Thema.

*Karteikasten:* Unterklasse von ‚Karteibehälter‘. Enthält geordnete Objekte der Klasse ‚Quellenkarte‘.

*Quellenangabe:* Unterklasse von ‚Eintrag‘. Teil eines Quellenverzeichnisses.

*Quellenkarte:* Unterklasse von ‚Karte‘. Enthält Autorin, Quellentitel, Verlag, Erscheinungsort und -jahr einer Quelle.

*Quellenverzeichnis:* Unterklasse von ‚Verzeichnis‘. Besteht aus nach Autorinnen geordneten Quellenangaben. Gehört zu einer Reinschrift.

*Reinschrift:* Manuskript eines Textes, welcher noch redigiert und gesetzt werden muß. Enthält eine Gliederung.

*Textkarte:* Unterklasse von ‚Karte‘. Enthält Text.

*Typographische Auszeichnung:* Layoutangabe in einem gesetzten Text, besteht aus einem typografischen Code (der z.B. eine Kursivdarstellung oder einen Umbruch bewirkt) und befindet sich jeweils an einer bestimmten Position.

*Übernahme:* Verbindung von einem Quellenverweis an einer bestimmten Position in einem Absatz zu einer Quellenangabe.

*Übernahmekarte:* Unterklasse von ‚Textkarte‘. Wörtliche oder inhaltliche Übernahme aus einer Quelle, zu der es eine Quellenkarte gibt. Besitzt als neues Attribut die Nummern der Seiten, aus denen der Text stammt.

*Verzeichnis:* Quellenverzeichnis, Bibliographie oder Katalog. Besteht aus Einträgen von Quellen, entweder zu einer Reinschrift gehörend oder zu einem bestimmten Bereich (wie einer bestimmten Bibliothek).

Die funktionalen Aspekte eines Systems werden in OMT durch Datenflußdiagramme nach DEMARCO modelliert (RUMBAUGH et al. 91, S. 123). In unserem Fall werden die Prozesse (durch Ellipsen gekennzeichnet) von der Autorin ausgeführt (s. Abbildung 5). Zwischen ihnen werden Produkte ausgetauscht, die im Modellgraphen als Kantenbezeichnungen dargestellt sind. In Datenspeichern, die im Graphen aus jeweils zwei horizontalen Linien bestehen, sind die privaten Angaben über Literatur in einer Literaturliste und die Exzerpte und Ideen in einem Karteikasten gespeichert.

Die den privaten Speichern entsprechenden öffentlichen Einrichtungen in Form von Bibliographien und Katalogen auf der einen und dem Buchhandel (Verlage, der Groß- und Einzelhandel) und den Bibliotheken auf der anderen Seite sind als Datenquellen bzw. -senken (mit Kästen) gekennzeichnet. Sie stehen damit zunächst außerhalb des Modells. Sie in das Modell miteinzubeziehen ist aber gerade der Ansatz, der beim Elektronischen Publizieren verfolgt wird (s. Abschnitt 2.1.3 auf S. 8). Dies gilt es für die weitere Analyse zu berücksichtigen.

Im hier vorgestellten Datenflußmodell stehen die im Abschnitt 2.3 beschriebenen Prozesse im Vordergrund, wie sie es bei dem Modell aus der Schreibforschung (s. Abbildung 2 auf S. 12) auch

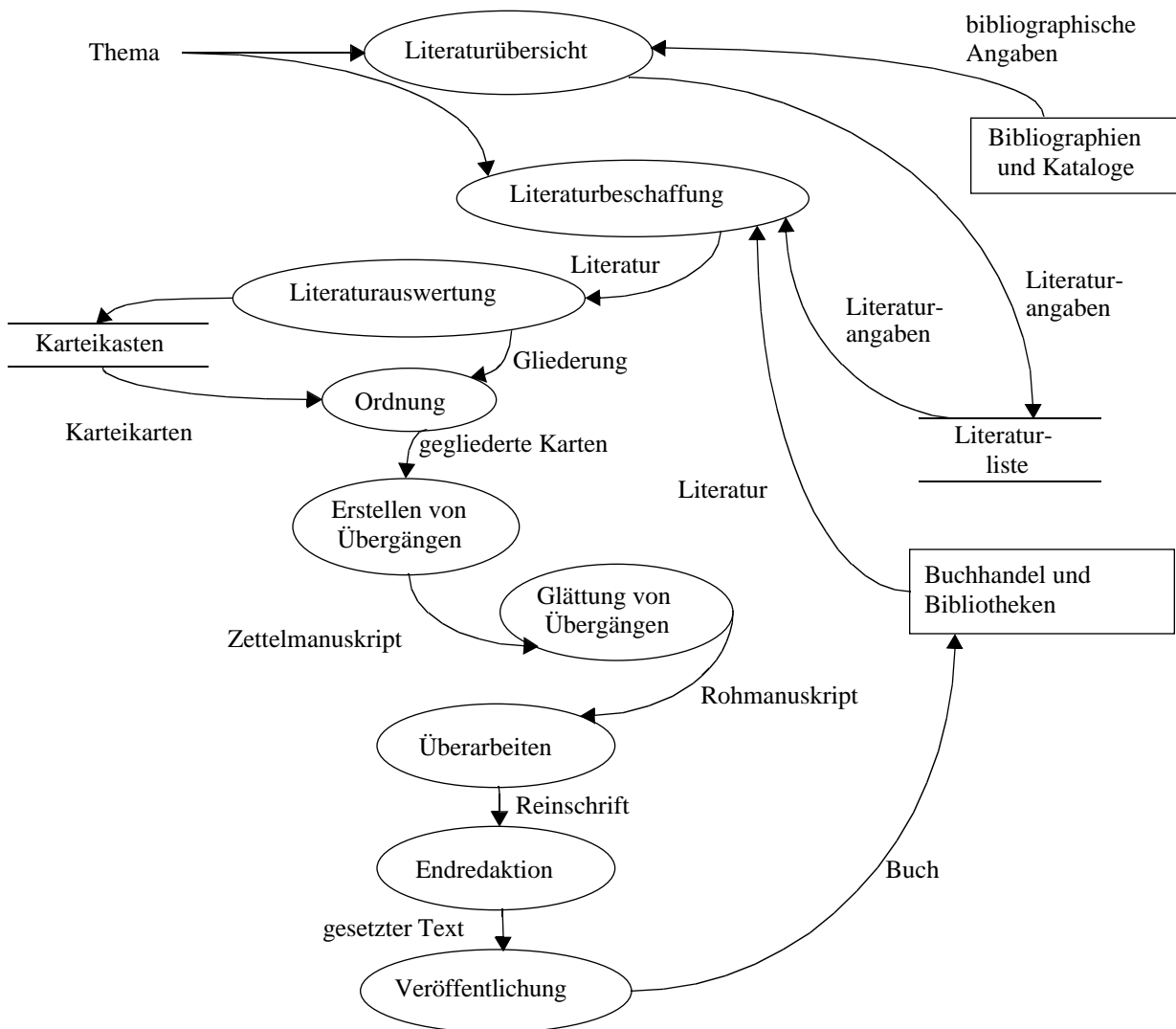


Abb. 5. Vorgang der Veröffentlichung eines wissenschaftlichen Dokumentes, modelliert nach SA.

schon taten. Diese Prozesse sind nicht automatisierbar, sondern werden jeweils von einem Menschen gemeinsam mit einem Computer ausgeführt, wobei der Computer Daten darstellt und verwaltet. Im Gegensatz zu dem linguistischen Modell sind in dem hier vorgestellten die zwischen den Prozessen fließenden Daten berücksichtigt.

### 2.5.2 Grobanforderungen an ein technisches System zum Schreiben

Nach der im letzten Abschnitt vorgenommenen Problemanalyse folgt nun die Formulierung erster Anforderungen an ein computergestütztes, verteiltes Textsystem für wissenschaftliche Schriftstücke, die sich aus den obigen Erkenntnissen über das wissenschaftliche Arbeiten ergeben:

1. Eine Wissenschaftlerin muß in dem System alle Phasen des Schreibens (s. S. 14) bruchlos erledigen können:

- 1.1. Autorinnen müssen mit dem System bestehende digitale Texte lesen, auf Textteile verweisen und neue Texte schreiben können. Unter Textverweisen werden Quellenverweise und von der Hauptargumentationslinie des Textes abweichende Abschnitte verstanden. Da der Übergang vom Lesen zum Schreiben fließend ist, muß es im System selbst möglich sein, Text zu schreiben.
- 1.2. Um möglichst vielen Netzbenutzerinnen den Zugriff auf Texte im System zu ermöglichen, müssen sie von verbreiteten, bestehenden Systemen aus zumindest gelesen werden können.
- 1.3. Aus allen Texten müssen Autorinnen Auszüge erstellen und in eine Ablage einsortieren können, ebenso eigene Texte und Teile davon.
- 1.4. Das System muß das Erstellen von Gliederungen unterstützen.
- 1.5. Autorinnen müssen Textteile zu Texten montieren können, indem sie in Gliederungen Auszüge aus fremden Dokumente und eigene Texte einfügen. Texte müssen optisch attraktiv gestaltet und immer wieder überarbeitet werden können.
- 1.6. Arbeitsmaterialien und Texte müssen schon vor einer Veröffentlichung des endgültigen Dokumentes anderen zugänglich gemacht werden können. Umgekehrt muß es Autorinnen auch möglich sein, anderen den Zugriff auf private Daten zu verwehren.
- 1.7. Für Pflicht- (z.B. Dissertationen) oder nicht bibliographisch erfaßte Veröffentlichungen (z.B. technische Berichte) muß die jeweilige Autorin den letzten Schritt der Publikation als Bereitstellen für die Öffentlichkeit und Aufnehmen in öffentliche Verzeichnisse selbst vornehmen können.
- 1.8. Es muß ersichtlich sein, wer ein Dokument geschrieben und publiziert hat. Gerade in einem verteilten digitalen Medium besteht nämlich die Gefahr, daß die Verbindung von Dokumenten zu ihrer jeweiligen Autorin verlorengeht und daß sich der Name und die Möglichkeiten des Zugriffs auf ein Dokument ständig ändern. Es muß auch außerhalb des Systems möglich sein, einen eindeutigen Bezeichner eines Dokumentes zu notieren.
- 1.9. Die inhaltliche Qualität von Dokumenten muß im System gesichert werden können, z.B. durch eine Begutachtungshierarchie, die v.a. auf sozialen Vereinbarungen basiert. Um Begutachtungen zu ermöglichen und das Verstehen von fremden Texten zu erleichtern, müssen beliebige Benutzerinnen an eigenen und Texten von anderen Anmerkungen anbringen können, für sich und für andere Leserinnen.
- 1.10. Selektierte Texte müssen (in Bibliographien etc.) öffentlich verzeichnet werden können. Es müssen jedoch Verzeichnisse für alle Arten von Dokumenten erstellt werden können. Außerdem muß es allen Benutzerinnen möglich sein, Verzeichnisse von beliebigen Dokumenten zu erstellen.
- 1.11. Die Archivierung ist zwar ebenfalls zunächst ein technisches, aber v.a. wegen der Flüchtigkeit digitaler Daten auch ein soziales Problem, weil nicht nur Archivierungsserver eingerichtet werden müssen, sondern auch festgelegt werden muß, wann ein Dokument auf ihnen durch ein anderes ersetzt oder gelöscht werden darf.
- 1.12. Die Abrechnung von Zugriffen auf Dokumente stellt keine Systemanforderung dar, weil sie für die Verbreitung wissenschaftlicher Informationen eher hinderlich ist (s. S. 9). Der Zugriff auf bestimmte Dokumente kann jedoch durch Paßwörter gesperrt werden, die verkauft werden können.

2. Das System soll auf die vorhandene Infrastruktur in z.B. Universitäten zurückgreifen, sowohl bezüglich der technischen Gegebenheiten als auch ihrer Verwaltung.
3. Das System soll arbeitstäglich benutzt werden können. Daher muß es einerseits eine angenehme Bedienung erlauben und ein relativ zügiges Antwortverhalten aufweisen. Andererseits ist den Benutzerinnen eine gewisse Einarbeitungszeit zuzumuten, weil das System differenzierte Prozesse unterstützen muß, ähnlich wie das klassische wissenschaftliche Schreib-, System‘.

Ein elektronisches Ablage-, Schreib- und Publikationssystem ist nur dann umsetzbar, wenn gemeinsame Objekte und Funktionen der Einzelkomponenten ausfaktoriert und als Basis bereitgestellt werden. Weiterhin soll es mehr bieten als die Digitalisierung bestehender Prozesse. Das bei der Beschreibung von Notes und NoteCards kurz angeklungene Hypertextkonzept scheint den richtigen Ansatz für solch ein Publikationssystem zu liefern. Bevor es in Abschnitt 4 spezifiziert wird, werden im folgenden Kapitel die verteilten Hypertextsysteme World Wide Web, HyperWave und Xanadu vorgestellt und verglichen.

### 3 Verteilte Hypertextsysteme

Das Objektmodell für das klassische wissenschaftliche Publizieren aus Abschnitt 2.5.1 (s. Abbildung 4 auf S. 25) enthält sich in der Praxis stark unterscheidende Klassen von Schreibprodukten, die im Modell jedoch ähnlich sind und auseinander abgeleitet werden. Es treten beispielsweise verschiedene Dokumentarten auf, wie ‚Gliederung‘ und ‚Manuskript‘, die zwar durch Schreiben erzeugt, deren Bestandteile aber jeweils unterschiedlich organisiert werden.

Weiterhin spielen sowohl bei der Erstellung als auch der fertigen Publikation implizite Verbindungen zwischen Textstücken (als Reihenfolge von Abschnitten) und explizite Verbindungen (als Zitate) eine große Rolle. Technisch bestehen alle Objekte des genannten Modelles einfach aus Text verschiedenen Umfangs oder verbinden andere Objekte, z.B. ein Karteikasten die entsprechenden Karten als Behälter.

Textblöcke mit expliziten Verbindungen zwischen ihnen werden in der Informatik ‚Hypertext‘ genannt. Das Hypertextkonzept erlaubt die einheitliche Behandlung vieler verschiedener Text- und Verbindungsarten. In dieser Arbeit soll daher das Objektmodell aus dem letzten Kapitel nicht direkt, sondern als Hypertextsystem umgesetzt werden. Im folgenden wird zunächst der Begriff des Hypertextes näher bestimmt, bevor drei vernetzte Hypertextsysteme beschrieben werden, das WWW, HyperWave und Xanadu.

#### 3.1 Prägung des Begriffes „Hypertext“ durch NELSON

Auch wenn selbst nach den Angaben NELSONS auf ihn weder die dahinterliegende Idee (sondern auf BUSH, s. NELSON 65) noch der Link als eines der Grundelemente (sondern auf ENGELBART, s. NELSON 93) zurückgehen, so ist er doch unbestritten Hypertextpionier. Das gilt nicht nur, weil der Begriff ‚Hypertext‘ von ihm stammt.<sup>1</sup> Wenn sich auch NELSONS Vorstellung hinter dem Begriff geändert hat, so hat sich seine Definition nicht gewandelt.<sup>2</sup> Sie entspricht immer noch der von 1965: „... a body of written or pictorial material interconnected in such a complex way that it could not conveniently be printed or represented on paper” (96). So hebt auch seine zwei Jahre später veröffentlichte Definition an:

... “hypertext” is the generic term for any text which cannot be printed (or printed conveniently) on a conventional page, or used conveniently when bound between conventional covers. “Non-linear text” might be a fair approximation. Hypertext may differ from ordinary text in its sequencing (it may branch into trees and networks), its organization (it may have multiple levels of summary and detail), its mode of presentation (it may contain moving or manipulable illustrations, moving or flashing typography), and so on. (67, 195)

Wenn NELSON von ‚nichtlinearem Text‘ spricht, meint er auch die Titelseiten von Zeitungen und Zeitschriftenseiten mit Illustrationen (93, 1/17), also das, was MCLUHAN in seinem berühmten Buch zur Medientheorie (94) als ‚Mosaik‘ bezeichnet hat.

Klassisch besteht ein Hypertext aus Textknoten, die nicht (wie Buchseiten) hintereinandergereiht sind, sondern zwischen denen durch explizite Verbindungselemente gesprungen werden kann. Hypertexte können zunächst nach der Art ihrer Knoten und Links unterschieden werden: Wenn Knoten nicht nur Text, sondern auch Bilder, Töne usw. enthalten, muß das System eigentlich

1 NELSON hat auch den Begriff ‚Hypermedia‘ zum ersten Mal benutzt (wenn auch nicht geprägt, weil er sich dessen lange Zeit nicht mehr bewußt war, s. NELSON 93, Erratum, und er ihn nicht im heute gängigen Sinn verstanden hat): “Films, sound recordings, and video recordings are also linear strings, ... The hyperfilm – a browsable or varissequenced movie – is only one of the possible hypermedia” (NELSON 65, 96).

2 Dieses Festhalten an der einfachen ersten Definition ist eigentlich erstaunlich. Sie wurde jedoch meistens durch System- oder andere Beschreibungen begleitet, so daß sie NELSONS Zwecken, der sich in den letzten Jahrzehnten nicht mehr an ein akademisches Publikum gewandt hat, genügt haben mag.

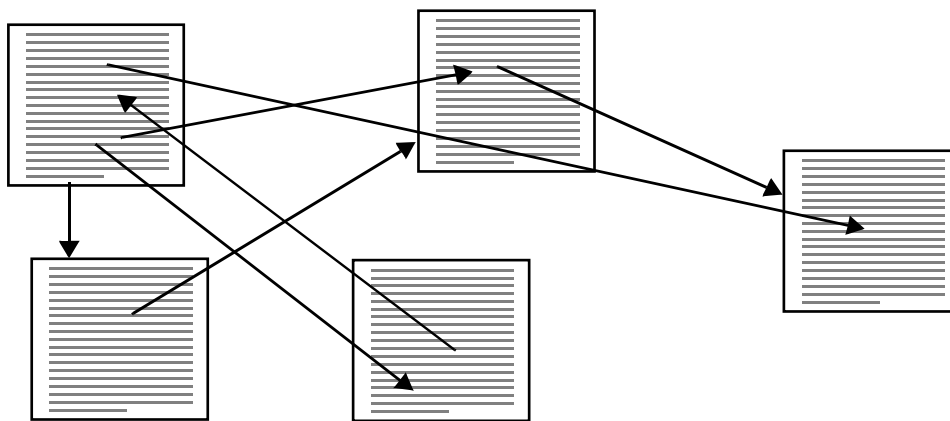


Abb. 6. Eine ‚Chunk-Style‘-Hypertextstruktur mit fünf Knoten und sieben Links.

Hypermedium genannt werden. Wenn in dieser Arbeit von Hypertext die Rede ist, ist auch Hypermedia gemeint. Links können ungerichtet oder gerichtet sein. Ihre Enden, bzw. ‚Anker‘, können sich jeweils auf einen ganzen Knoten beziehen oder auf einen Teilknoten. Die eben beschriebene Art Hypertext heißt nach den Textblöcken, die verbunden sind, ‚Chunk-Style-Hypertext‘. Sie ist die populärste. Wie wir noch sehen werden, gibt es jedoch auch andere Arten, die z.B. die *Einfügung* von Textausschnitten in einen anderen Block erlauben.

In der Hypertextgemeinde ist nicht einmal die obige, vage Definition von Hypertext unumstritten. NIELSEN nennt daher ein System Hypertextsystem, wenn es einen gewissen „Look and Feel“ aufweist und z.B. den schnellen Wechsel zwischen Texten erlaubt. Dieser schnelle Wechsel ist besonders schwierig in verteilten Hypertextsystemen umzusetzen, bei denen die Haltung und der Zugriff auf Daten von verschiedenen Computern aus erfolgen kann, die über ein WAN (Wide Area Network) verbunden sind. Sie sind technisch und von der Bedienung her wesentlich aufwendiger als lokale Hypertextsysteme, bei denen die Daten auf dem Computer abgespeichert sind, von dem aus auf sie zugegriffen wird.

Das gebräuchlichste WAN ist derzeit das Internet, welches erst durch das Hypertextsystem WWW zu Bekanntheit außerhalb technischer und akademischer Kreise gelangt ist, in populärer Vorstellung sogar mit diesem übereinstimmt. Aus diesem Grund soll das WWW als erstes von drei Hypertextsystemen besprochen werden. Es ist außerdem das einfachste der hier beschriebenen Systeme und weist daher einige Schwächen auf, welche bei der Entwicklung von HyperWave (früher HyperG) teilweise vermieden wurden. Die beiden Systeme sind implementiert und im Einsatz, das WWW in weitaus größerem Umfang als HyperWave. Das ist ein weiterer Grund, ersteres in dieser Arbeit als Referenzsystem zu betrachten. Xanadu, das System des Hypertextpioniers NELSON, ist bisher nicht implementiert, und es ist unklar, ob es je dazu kommen wird. Es soll jedoch im Anschluß an das WWW und HyperWave beschrieben werden, weil es als Publikationssystem entworfen wurde, das zumindest auf dem Papier die Anforderungen aus dem letzten Kapitel erfüllt, wenn man sie in Hypertext ‚übersetzt‘. Diese Übertragung soll im nächsten Kapitel geschehen, nachdem die drei Hypertextsysteme in diesem zusammenfassend verglichen wurden.

## 3.2 Das World-Wide Web

### 3.2.1 Geschichte und Anforderungen

Die Entwicklung des World Wide Web (WWW) wurde 1989 von Tim Berners-Lee am CERN (Centre Européen de Recherches Nucléaires) in Genf initiiert (BERNERS-LEE 96), jedoch noch nicht unter diesem Namen. Das WWW sollte das Problem der Dokumentation am CERN lösen. Dort werden ständig neue Geräte und Programme entwickelt, die dokumentiert werden müssen. Da außerdem die am CERN arbeitenden Wissenschaftlerinnen ständig wechseln, müssen Informationen über ihre Arbeit auch für andere, besonders für neue Mitarbeiterinnen, zur Verfügung stehen (ebd., Abschn. „Losing Information at CERN“).

BERNERS-LEE hatte 1980 ein Hypertextsystem für die Dokumentation von Software geschrieben. Es konnte kleine Texte verwalten, zwischen denen über Links gesprungen werden konnte. BERNERS-LEE hatte darin Informationen über Mitarbeiterinnen und Programm-Module gespeichert. Das Programm lief auf einem Mehrbenutzerinnensystem, auf dem viele Benutzerinnen auf dieselben Daten zugreifen konnten. Eine Version unter VMS wurde auch von anderen Beschäftigten des CERN benutzt (ebd., Abschn. „Personal Experience with Hypertext“).

Da die zu speichernde Informationsstruktur des neuen Systems mit ihrem Inhalt wachsen und sich ändern sollte, wurde ein Gewebe („Web“) vorgeschlagen, weil eine Hierarchie dafür zu restriktiv wäre. Jeder Knoten des Gewebes sollte aus einem kleinen Text oder einer kleinen Grafik bestehen, welche einen Menschen, Gruppen von Menschen, Projekte, Konzepte, ein Softwaremodul, Hardwaretypen oder eine bestimmte Hardwareinheit beschreibt. Die Knoten sollten durch Links verbunden sein, die jeweils das Verhältnis „hängt ab von“, „ist Teil von“, „hat ... gemacht“, „bezieht sich auf“, „benutzt“ oder „ist ein Beispiel von“ ausdrückt. Das System sollte ursprünglich diese Typen kennen und jeweils speziell behandeln, v.a. darstellen (ebd., Abschn. „Linked Information Systems“), was im heutigen WWW jedoch nicht möglich ist.<sup>3</sup>

BERNERS-LEE hat 1989 folgende technische Anforderungen für ein Hypertextsystem des CERN formuliert: Es muß von externen Computern über das Internet benutzbar sein, wobei verschiedene Plattformen gleichzeitig unterstützt werden müssen (VM/CMS, Mac, VAX/VMS und Unix). Bereits bestehende Datenbanken müssen in das Hypertextsystem eingebaut werden können. Es soll dynamische Dokumente erlauben, die erst bei der Abfrage erstellt werden. Das Hypertextnetz muß eine dezentrale Struktur aufweisen, welche mittels Software analysiert werden kann (um z.B. ‚Löcher‘ und undokumentierte Programme aufzuspüren). Schließlich sollen private Links zwischen eigenen und öffentlichen Dokumenten und private Anmerkungen von Links und Knoten eingefügt werden (ebd., Abschn. „CERN requirements“).

Als unwichtig für ein Dokumentationssystem am CERN hat BERNERS-LEE die Anforderungen Authentifizierung, Abrechnung, Durchsetzung von Urheberrechten und Datenschutz („information exchange is still more important than secrecy“) genannt (ebd.). Die Unterstützung von Multimedia-Datenformaten stand ebenfalls nicht im Vordergrund. Daten sollten entweder weltweit ohne Einschränkung oder bei Zugriff über ein Dateisystem je nach gesetzten Zugriffsrechten möglich sein (BERNERS-LEE U. CAILLIAU 90).

Auffallend ist, daß BERNERS-LEE die Sicherstellung der referentiellen Integrität<sup>4</sup> von Links nicht einmal erwähnt. Sie spielt in einem kleinen Hypertextsystem jedoch auch keine große Rolle

<sup>3</sup> In HTML 3.2 können Links Typen zugewiesen werden, außerdem ein Name, der am Fuße des Browserfensters erscheint, wenn die Maus sich über einem Link befindet (s. RAGGETT 96b).

<sup>4</sup> Dieser Begriff stammt aus dem Umfeld der Theorie Relationaler Datenbanken und bezeichnet eine Bedingung, die dann erfüllt ist, wenn sich alle Tupel einer Relation, die auf andere Relationen verweisen, nur auf *existierende* Tupel in dieser anderen Relation beziehen (ELMASRI U. NAVATHE 94, 147). Für Hypertextsysteme liegt referentielle Integrität dann vor, wenn alle Links nur Objekte verbinden, die auch tatsächlich vorhanden sind.



und kann innerhalb eines Instituts notfalls durch einen Telefonanruf sichergestellt werden. Außerdem ist es technisch aufwendig, sie umzusetzen, wie wir bei den noch folgenden Beschreibungen der Hypertextsysteme HyperWave und Xanadu sehen werden.

Für die Entwicklung des Hypertextsystems am CERN sollten zunächst kommerzielle Programme auf ihre Brauchbarkeit untersucht werden. Als Kriterien sollten die Portabilität oder die Verfügbarkeit auf vielen Plattformen im Vordergrund stehen, sowie die Erweiterbarkeit auf neue Datenformate (ebd., Abschn. „A Practical Project“). Aus Gründen der Flexibilität und Heterogenität sollte das Hypertextsystem nach dem Client-Server-Prinzip implementiert werden, bei dem Datenpräsentation und -haltung zwischen Clientprogrammen für Benutzerinnen und zentralen Servern aufgeteilt sind. Wichtig für diese Aufteilung ist die Definition der Schnittstelle zwischen Client und Server, die ein entsprechendes System stark prägt. Danach können verschiedene Client- und Serverprogramme getrennt entwickelt werden (BERNERS-LEE 96, Abschn. „What will the system look like?“).

Der Client, ‚Browser‘ genannt, sollte auf verschiedenen Plattformen angeboten werden und die jeweilige Oberfläche unterstützen. Seine wichtigste Funktion neben der Darstellung von Hypertextknoten sollte die Unterstützung bei der Verfolgung von Links sein. Die Unterstützung beinhaltet das Merken von besuchten Knoten und die Möglichkeit des Zurückgehens zu diesen.

Server sollten nicht speichern, welche Knoten die Benutzerin besucht hat. Sie sollten nur das Netz von Hypertextknoten auf der jeweiligen Maschine verwalten und u.U. Daten zwischen verschiedenen Formaten konvertieren. Als besonderes Merkmal des WWW war nämlich die Verhandlung des Datenformats zwischen Client und Server geplant. Ein Browser mit ASCII-Darstellung würde z.B. Knoten nur im ASCII-Format erhalten (BERNERS-LEE U. CAILLIAU 90, Abschn. „Building Block“).

Damit Benutzerinnen des Hypertextsystemes von BERNERS-LEE nicht auf speziell dafür geschriebenen Hypertext warten müßten, sollte das System den Zugriff auf bestehende Datenbanken erlauben. Dazu sollte ein Hypertextserver als ‚Gateway‘ zwischen den Client-Programmen und der Datenbank vermitteln, so daß deren Daten nicht in das Hypertextformat konvertiert werden müßten. Auf manche Daten könnten Normalbenutzerinnen nur lesend zugreifen. Nur eine kleine Gruppe von Leuten hätte dann Schreibzugriff auf die Datenbanken und könnte Informationen zur Verfügung stellen.

In eine bestehende Datenbank könnten Hypertextinformationen abgespeichert werden, die ein Gateway-Server für die Darstellung aufbereiten könnte (ebd., Abschn. „Accessing Existing Data“). Als Beispiele für solche Datenbanken nennt BERNERS-LEE Usenet-News, das ähnliche VAX/Notes, das Dokumentenverwaltungssystem CERNDoc, Dateisysteme, ein Telefonverzeichnis, Unix-Handbücher (Man-Pages) und kommerzielle Datenbanken (BERNERS-LEE 96, Abschn. „What will the system look like?“).

### 3.2.2 Implementierung des WWW

Der Vorschlag von BERNERS-LEE für die Entwicklung eines Hypertextsystemes wurde 1990 unter dem Namen ‚World-Wide Web‘ (WWW) umgesetzt (W3C o.J.). Das System stellt ein „globales Informationsuniversum“ dar, welches bestehende Technik benutzt (BERNERS-LEE et al. 92, 52). Es verbindet Hypertext mit bestehenden Informationsdiensten wie Alex, Gopher, Prospero und WAIS (ebd., 52f).

Bereits 1990 hatte BERNERS-LEE einen WYSIWYG-Browser/Editor für den NeXT-Computer entwickelt, bei dem interaktiv Links erstellt werden konnten. Außerdem war am CERN ein zeilenorientierter Browser programmiert worden. Im darauffolgenden Jahr wurden WWW-Programme für gängige Computer und ein anonymer Telnet-Zugang zum WWW der Öffentlichkeit zur Verfü-

gung gestellt (W3C o.J.). Im Jahr 1992 wurden verschiedene graphische Browser für das graphische Fenstersystem X unter dem Betriebssystem UNIX vorgestellt.

Im Februar 1993 wurde von dem amerikanischen ‚National Center for Supercomputing Applications‘ (NCSA) ein Browser für das WWW unter dem Namen ‚Mosaic‘ veröffentlicht. Er war unter anderem von MARC ANDREESSEN entwickelt worden (ebd.). Dieser wird 1994 Mitbegründer von ‚Mosaic Communications‘. Diese Firma nennt sich später in ‚Netscape Communications‘ um und produziert den erfolgreichen WWW-Browser Netscape (ebd.).

Im WWW hat jedes Datenobjekt eine Adresse, über die es erreicht werden kann (URI)<sup>5</sup>. Weiterhin gibt es ein Netzwerkübertragungsprotokoll für diese Objekte (HTTP) und ein Darstellungsformat für Informationen (HTML). Schließlich erlaubt das WWW den Zugriff auf im Internet bereits vorhandene Informationen, die über die genannten Elemente adressiert und dargestellt werden können (BERNERS-LEE et al. 94, 78).

Die Objektnamen im WWW heißen ‚Universal Resource Identifiers‘ (URIs) und werden durch Zeichenketten dargestellt. Sie bestehen aus einem Präfix für das Protokoll und der Adresse eines Objektes auf einem Server, z.B.:

`http://www.w3.org/pub/WWW/TR/PR-html32-961105.html`

Gewöhnlich dienen Schrägstriche als hierarchische Trenner im Namen. Ein Fragezeichen dient dazu, um zwischen einem Abfrageobjekt und der speziellen Abfrage zu unterscheiden (BERNERS-LEE et al. 94, 78). Der erste Teil der Adresse, ‚www.w3.org‘, bezeichnet einen Computer im Internet. Eine Netzadresse enthält also einen Ort in diesem Netz. Noch nicht verwirklicht, aber geplant ist ein Namensschema, bei dem Dokumente unabhängig von ihrem Ort adressiert werden können (ebd., 81). Wenn ein unvollständiger URI an einen WWW-Server geschickt wird, der in einem Verzeichnisnamen endet, sucht der Server zuerst nach Dateien mit den Namen `home.html` und `index.html` in dem entsprechenden Verzeichnis und zeigt die erste gefundene Datei an. Wenn keine solchen Dateien vorhanden sind, zeigen manche Server den Inhalt des entsprechenden Verzeichnisses an. Verzeichnisse werden dann also nicht nur als Namenspräfixe benutzt, sondern auch als eigenständige Namen von Objekten.

Das Übertragungsprotokoll des WWW heißt ‚Hypertext Transfer Protocol‘ (HTTP). Es dient jedoch nicht nur der Übertragung von Hypertext, sondern von allen möglichen Daten, auch in binärer Form. Die Übertragung erfolgt schnell genug, um ‚Sprünge‘ zwischen Dokumenten zu ermöglichen (ebd., 78). HTTP basiert auf dem Internetstandardprotokoll TCP/IP und ist im Gegensatz zu dem Dateiübertragungsprotokoll FTP (File Transfer Protocol) und dem Usenet-News-Protokoll NNTP (Network News Transport Protocol) zustandslos<sup>6</sup>, eine TCP-Verbindung wird also nur für jeweils eine Anfrage offen gehalten. Als Grund dafür geben BERNERS-LEE et al. an, daß ein Link sowohl auf ein Objekt auf dem letzten zugegriffenen Server als auch auf einem ganz anderen Server zeigen kann. Ein URI sollte eine (möglichst) eindeutige Bedeutung haben, das Ergebnis seiner Anfrage also nicht von vorherigen Anfragen abhängen (ebd., 78).

Client und Server kommunizieren über Abfragen (Requests), die der Client an den Server schickt. Für diese Abfragen gibt es in HTTP/1.0 drei verschiedene Methoden: ‚GET‘ und ‚HEAD‘

5 ‚URI‘, ‚Universal Resource Identifier‘, ist im Umfeld des WWW der Name für alle kurzen Strings, die sich auf Objekte beziehen. Derzeit im WWW benutzte Adressen, die mit ‚ftp:‘ oder ‚http:‘ oder ‚file:‘ (für lokale Dateien im Dateisystem des Browsers) anfangen, sind dagegen ‚URLs‘, ‚Uniform Resource Locators‘. ‚URNs‘, ‚Uniform Resource Names‘ schließlich heißen alle URIs, die keine URLs sind (und von denen es z. Z. noch keine gibt) (W3C 96).

6 Dieser Begriff bezieht sich nach TANNENBAUM eigentlich auf Dateiserver. Der korrekte Begriff lautet ‚verbindungslos‘, weil Daten in HTTP übertragen werden, ohne daß dafür auf HTTP-Ebene eine Verbindung auf- und wieder abgebaut wird (92, 27ff und 661ff).

dienen der Anforderung von Daten durch den Clienten vom Server, ‚POST‘ dient der Lieferung von Daten vom Client an den Server. Diese Methoden haben jeweils u.a. eine URI, also eine WWW-Adresse als Parameter. ‚GET‘ liefert die Daten, auf die sich die URI bezieht. Wenn diese URI auf ein Programm verweist (z.B. ein Skript nach dem weiter unten beschriebenen CGI-Standard) wird nicht dieses Programm, sondern die Daten, die es erzeugt, geliefert (s. S. 40).

Es gibt eine Variante der GET-Methode, bei der ein Datum angegeben wird und die nur dann zu einer Übertragung führt, wenn das angesprochene Objekt seit diesem Datum verändert wurde. Dadurch können Daten, die in einem Cache abgelegt wurden, einfach erneuert werden (BERNERS-LEE et al. 95, Abschn. 8, ‚Method Definitions‘). Bei einem Cache handelt es sich um einen Speicher in einem Client oder Server, der angeforderte Daten aufbewahrt, um sie beim späteren Auftreten derselben Anforderung vorrätig zu haben. Der Cache spart dadurch Datenübertragungen, was sich in schnelleren Antwortzeiten bemerkbar macht (ebd., Abschn. 1.2, ‚Terminology‘).

Die HEAD-Methode arbeitet ähnlich wie die GET-Methode. Sie liefert jedoch nicht das ganze angesprochene Objekt, sondern nur die zu ihm gehörenden Metainformationen. Außerdem teilt sie einem Browser Informationen über den Server mit (s. die Beschreibung des ‚<HEAD>‘-Abschnittes von HTML-Dokumenten weiter unten). Es gibt keine Variante wie bei GET, die nur nach einem bestimmten Datum geänderte Daten überträgt (ebd., Abschn. 8, ‚Method Definitions‘).

Die POST-Methode enthält außer einer URI noch Daten, die auf dem Server ‚unterhalb‘<sup>7</sup> des durch den URI adressierten Objektes abgelegt werden sollen. Als Beispiel für die Anwendung dieses Befehles geben die Autoren an: Annotieren von Objekten, Einfügen eines Artikels in z.B. eine News-Gruppe, Übertragung von Daten, die etwa in einem HTML-Formular eingegeben wurden und schließlich Ablegen von Daten in einer Datenbank. Unter diesen und anderen Funktionen unterscheidet der Server, meistens in Abhängigkeit von dem angegebenen URI. Der POST-Befehl muß nicht zur Erzeugung eines Objektes auf dem Server führen, auf welches später über eine URI zugegriffen werden könnte. Wenn ein solches Objekt jedoch gespeichert wird, wird dies dem Client durch den Server mitgeteilt (ebd.).

In der ersten Version von HTTP gab es neben POST noch PUT zur reinen Änderung eines Dokumentes. BERNERS-LEE et al. schreiben 1994, daß die Schreibbefehle POST und PUT aus Mangel an Hypertexteditoren noch wenig verbreitet sind, im Gegensatz zu GET (und HEAD), welches von allen Browsern erzeugt werden kann (78). Metainformationen zu Objekten werden im Standard-MIME-Format übertragen (ebd.).

Das Datenformat zur Darstellung von Hypertext im WWW heißt ‚Hypertext Mark-Up Language‘ (HTML). Es kann sowohl von Menschen in einem Texteditor geschrieben als auch von Programmen erzeugt werden. Weil HTML aus ASCII-Zeichen besteht (und sich standardmäßig auf 7-Bit-ASCII beschränkt), kann es nicht nur über HTTP, sondern auch SMTP (Usenet-Mail) übertragen werden (ebd., 79).

Die letzte Version von HTML, die bisher nur als ‚Working Draft‘ existiert, trägt die Versionsnummer 3.2 (RAGGETT 96b, Abschn. ‚Introduction to HTML 3.2‘). HTML 3.2 wurde vom WWW-Konsortium (W3C) zusammen mit IBM, Microsoft, Netscape, Novell, SoftQuad, Spyglass, and Sun Microsystems entwickelt. Es ist abwärtskompatibel zu HTML 2.0 und enthält Elemente, deren Benutzung zwar weitverbreitet ist, die aber noch nicht standardisiert sind, z.B. Tabellen, Applets und Text, der Bilder umfließt (ebd.).

HTML basiert auf dem Standardtextformat SGML (Structured General Markup Language, s. GOLDFARB 90) und ist daher über eine SGML-Deklaration und eine SGML-Document Type Definition (DTD) festgelegt. HTML-Elemente (‚Befehle‘) sind in spitzen Klammern eingeschlossen.

7 Dieses ‚unterhalb‘ soll nach Aussage der Autoren ein Verhältnis ausdrücken wie es eine Datei zu einem Verzeichnis, ein Artikel zu einer News-Gruppe oder ein Datensatz zu einer Datenbank hat (ebd.).

Wenn sie sich auf einen Textbereich beziehen, wird dieser von zwei Elementen eingerahmt, wobei das letzte einen Schrägschlag vor dem Namen trägt. Elemente können Attribute enthalten, die ihre Parameter angeben. HTML-Dokumente beginnen mit der Angabe der HTML-Version in einem speziellen SGML-Format und schließen die restlichen Daten in ein HTML-Element ein. Ein typisches HTML-Dokument könnte so aussehen:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Draft//EN">
<HTML>
<HEAD>
<TITLE>Der Titel des HTML-Dokumentes</TITLE>
<INDEX>
<LINK REL=Contents HREF=Inhalt.html>
<LINK REL=Previous HREF=Kapitel7.html>
<LINK REL=Next HREF=Kapitel9.html>
<META NAME="Author" CONTENT="Christine Pauli">
</HEAD>
<BODY>
<P>Im Körper steht der eigentliche Text des Dokumentes. Er ist im allgemeinen wesentlich länger als der Kopf, z.B. bei
<A HREF="http://www.w3.org/pub/WWW/TR/WD-html32.html">
diesem Dokument</A>.</P>
<P>Das genannte Dokument hat auch eine
<A HREF="http://www.w3.org/pub/WWW/TR/WD-html32.html#intro">
Einleitung</A>.</P>
</BODY>
</HTML>
```

HTML-Dokumente beginnen mit einem ‚HEAD‘-Teil. Dieser enthält Metainformationen zu einem Dokument, wie den Titel, Angaben über die Autorin und Erstellungs- und Ablaufdaten. Optional werden im Kopf auch Daten über das Verhältnis des vorliegenden Dokumentes zu anderen Dokumenten gespeichert (ebd., Abschn. „The Head Element“).

Im Kopfabschnitt können auch Angaben zu Style-Sheets (Formatvorlagen) und Skriptprogrammen gemacht werden, die aber in HTML 3.2 noch nicht näher spezifiziert sind, obwohl sie in kommerziellen Browsern schon Verwendung finden (ebd.). Im Kopf kann ein HTML-Dokument weiterhin als ‚INDEX‘ deklariert werden, was bedeutet, daß Browser bei seiner Darstellung automatisch ein Eingabefeld generieren und eingegebene Daten an eine bestimmte WWW-Adresse schicken. Auf dem entsprechenden Server muß ein Programm laufen, das die Anfragen auswertet und mit einem WWW-Dokument beantwortet (ebd.). Bei einer Anfrage kann es sich z.B. um einen String handeln, der vom Server in seinen Dokumenten gesucht wird. Als Antwort generiert ein Server eine Liste mit allen Dokumenten, die den String enthalten.

Über das Element ‚LINK‘ im Dokumentenkopf können inhaltliche Verbindungen zwischen HTML-Dokumenten und anderen WWW-Objekten hergestellt werden. (Dieses Element sollte nicht mit dem zur Erzeugung von Sprunglinks im Textkörper eines HTML-Dokumentes verwechselt werden!) Im Beispiel werden die Adressen eines Inhaltsverzeichnisses, eines Vorgänger- und eines Nachfolgerdokumentes angegeben. Das Link-Element findet sich schon in den ersten HTML-Standards, wird jedoch nur von wenigen Browsern ausgewertet, z.B. dem weiter unten beschriebenen AOLPress.

Link-Elemente sollen entweder als Buttons in einem WWW-Browser dargestellt werden (z.B. um einen Sprung zum Inhaltsverzeichnis auslösen zu können) oder eine Dokumentenreihenfolge für den Ausdruck eines Baumes von HTML-Dokumenten auf Papier angeben. Link-Elemente ent-

halten als Attribut die Adresse des anderen Objektes, das Verhältnis zu diesem und seinen Titel. Als Verhältnis zwischen Objekten kann man u.a. angeben, daß eines ein Inhaltsverzeichnis, Index, Glossar, Urheberrechtshinweis, das nächste oder vorherige in einer Folge, ein Hilfedokument oder ein ‚Lesezeichen‘ des anderen darstellt.<sup>8</sup> Es kann auch eine ‚Home Page‘ oder E-Mail-Adresse der Autorin genannt werden (ebd.), was für einen wissenschaftlichen Text im WWW wichtig wäre, aber eben nicht von Browsern ausgewertet wird. Leserinnen können die Angabe aber im Zweifelsfall immer noch sehen, indem sie den HTML-Sourcecode des Dokumentes betrachten (um z.B. die Autorin eines vorliegenden Dokumentes festzustellen). Die Autorin eines Dokumentes kann auch über ein ‚META‘-Element im Kopf angegeben werden (ebd.). Es hat mit dem Link-Element gemein, daß es in den wenigsten Fällen von einem Browser ausgewertet wird.

Der eigentliche Inhalt eines HTML-Dokumentes wird im ‚Körper‘ angegeben. Dieser wird vom ‚BODY‘-Element eingeschlossen (ebd., „The BODY element“). Er kann Textelemente wie Überschriften, Tabellen, Listen und Paragraphen enthalten, außerdem die Angabe der Adresse der Autorin, die vom Browser auf eine besondere Weise dargestellt wird (ebd.). Ein weiteres Body-Element ist das Formular, welches durch ‚FORM‘ angegeben wird. Es dient der Eingabe von Daten durch die Nutzerin einer WWW-Seite. Es kann daher Felder enthalten, die aus graphischen Benutzungsoberflächen bekannt sind wie einzeilige und mehrzeilige Texteingabeboxen und Auswahlfelder (ebd.).

Über ein Formular kann auch eine Datei vom Rechner, auf dem der Browser läuft, an einen Server geschickt werden. Dazu kann entweder der Dateiname in einem Textfeld des Formulars eingegeben, oder in einem Dateiauswahldialog, wie er aus anderen Programmen für graphische Benutzungsoberflächen bekannt ist, ausgewählt werden. Die Auswahl einer Datei in einem Formular und ihre Übertragung an einen Server wird ‚Form-based File Upload‘ genannt. Auf diese Weise können auch mehrere Dateien gleichzeitig übertragen werden (NEBEL U. MASINTER 95).

Ausgangs- und Zielpunkte für Hypertextsprünge werden in HTML Anker genannt und über das HTML-Element ‚A‘ angegeben. Dieses Ankerelement kann nicht geschachtelt werden und schließt zwischen Anfangs- und Endteil den HTML-Text ein, der als Quelle oder Ziel dienen soll (ebd.). Ausgangsanker enthalten den URL des Sprungzieles, Zielanker einen Namen, über den sie angesprochen werden können (ebd.).

Zielanker dienen dazu, zu einem bestimmten Bereich in einem HTML-Dokument zu springen, den ein Anker umgibt. Ein Bereich in einem Dokument wird adressiert, indem sein Name hinter dem Zeichen ‚#‘ an die URL des Dokumentes angehängt wird, wie man am zweiten Link im obigen Beispiel sehen kann. Wenn ein solcher URL aufgerufen wird, lädt ein Browser das ganze Dokument und scrollt es so, daß der Anfang des betreffenden Bereiches am Anfang des Fensters steht. Der Bereich wird jedoch nicht (z.B. farbig) hervorgehoben.

Über das Element ‚<IMG>‘ ist es möglich, Bilder in ein HTML-Dokument einzubinden. Es enthält die Adresse der entsprechenden Bitmap-Datei als URL, die auch auf einem anderen Server als der entsprechende Text liegen kann. (Bilder werden also im WWW ‚transkludiert‘, wie eine Einbindung in Xanadu genannt wird; s. S. 71.) In HTML-Dateien eingebundene Graphiken können als Auswahlmenüs dienen. Ein Mausklick auf eine dafür eingerichtete Graphik führt dazu, daß die Koordinate des Bildpunktes unter dem Mauszeiger an einen Server übertragen wird. Der Server muß diese Koordinate über ein CGI-Skript auswerten. Er kann z.B. die ganze Graphik als Menü und bestimmte Koordinatenbereiche als einzelne Menüpunkte interpretieren. Der Aufruf eines bestimmten Menüpunktes führt dann dazu, daß ein für ihn festgelegtes Dokument an den Browser zurückgeliefert wird.

<sup>8</sup> Ein Link-Element gibt tatsächlich nur das Verhältnis zwischen zwei Objekten an. Man kann zwischen zwei Attributen wählen, um festzulegen, welches von den beiden Objekten z. B. das Inhaltsverzeichnis des anderen ist.

Um graphische Menüs erstellen zu können, ohne daß auf einem Server Koordinaten in zu Bereichen gehörende URLs umgerechnet werden müssen, können Bilder in neuen HTML-Versionen als sogenannte ‚Client-Side-Maps‘ definiert werden. Bei ihnen wird die Abbildung von Koordinaten in Aktionen im entsprechenden Browser durchgeführt. Zu jeder Client-Side-Map werden Linkanker in Form von geometrischen Figuren angegeben, denen URLs zugeordnet sind. Wenn eine Figur durch Mausklick aktiviert wird, wird die entsprechende URL geladen (ebd.).

Verschiedene Browser erlauben das Einbinden von kleinen Programmen als Skripte in HTML-Dokumenten. Der Code dieser Skripte liegt immer in Textform vor und wird interpretiert. Skripte können sowohl direkt in dem entsprechenden HTML-Dokument vorliegen als auch in einer eigenen Datei gespeichert werden, deren URL in dem HTML-Dokument angegeben ist. Die bekanntesten Beispiele für Skriptsprachen für das WWW sind JavaScript von Netscape Communications und VBScript von Microsoft.

Obwohl die Ausführung der Skripte von den Fähigkeiten des jeweiligen Browsers abhängt, gibt es einen Vorschlag des W3C für die Standardisierung zumindest der Syntax des Einbindens (RAGGETT 96a). Skripte können dazu dienen, einfache interaktive HTML-Dokumente zu erstellen und Eingaben in HTML-Formularen auf Korrektheit zu überprüfen (ebd., Abschn. „Introduction“).

Weil sich das WWW-Konzept als allgemein genug erwiesen hat, um den Zugriff auf Netzdaten in alten Formaten zu erlauben, wird es nach Voraussicht seiner Entwickler wahrscheinlich auch den Übergang zu noch nicht definierten Formaten erlauben (BERNERS-LEE et al. 94, 80). Schon implementierte und noch geplante Eigenschaften des WWW sollen es ermöglichen, daß seine Datenformate leicht weiterentwickelt werden können. So werden unbekannte HTTP-Daten von Browsern und Servern ignoriert, wodurch neue Formate das bestehende Netz nicht stören (ebd., 78). Über die geplante ‚Content-Negotiation‘ können eigene Formate im WWW benutzt und getestet werden. Dabei handelt es sich um ein Protokoll zum Aushandeln von Datenformaten und Kompressionsverfahren für eine HTTP-Übertragung und der Sprache eines Hypertextes zwischen Browser und Server. Der Browser soll dabei eine gewichtete Liste von Formaten an den Server schicken, worauf dieser das ‚gehaltvollste‘ vom Browser verstandene auswählt (ebd.).

WWW-Browser können mit Servern kommunizieren, die ganz verschiedene Dienste anbieten. Die Browser stellen alle Daten in einem einheitlichen Format dar und sind daher ein Fortschritt gegenüber bisherigen ‚User Agent‘-Programmen für Internetdienste, die jeweils nur eine Art von Daten verarbeiten und darstellen können (ebd., 76). WWW-Browser haben von Anfang an den Zugriff auf lokal verfügbare Dateisysteme erlaubt und die Netzwerkprotokolle FTP, NNTP, das Gopher-Protokoll und das eigene Protokoll HTTP verstanden. Der schon erwähnte Präfix der WWW-Adresse wird benutzt, um die Protokolle zu unterscheiden (BERNERS-LEE 92, 55f). Gewöhnliche WWW-Dokumente liegen als HTML-Dateien auf Servern. Es gibt jedoch auch dynamische (virtuelle) Dokumente, die von einem Server erst auf Anforderung erzeugt werden. Sie bieten sich für die Abfrage von Datenbanken und die Darstellung von sich ständig ändernden Daten an (ebd., 53).

Daten beliebigen Formats werden im WWW veröffentlicht, indem ein kleines Serverprogramm darauf Zugriff gewährt. Sie müssen dafür nicht geändert werden (ebd., 54). Da es zunächst nur wenige HTML-Editoren gab, wurden die meisten Informationen im WWW von Systemverwalterinnen, Herausgeberinnen o.ä. veröffentlicht (BERNERS-LEE 94, 82). Mit Hypertexteditoren sollen in Zukunft nicht nur Dokumente erstellt und geändert und Links auf andere Dokumente gelegt, sondern auch Anmerkungen verfaßt werden können (ebd., 81). Das als Zukunftsvision beschriebene gleichzeitige Ändern von Dokumenten im WWW durch mehrere Benutzerinnen, das Abhalten von Telekonferenzen und die Unterstützung von Virtual Reality (ebd., 82) hat sich ebenfalls noch nicht auf breiter Front durchgesetzt. Es gibt jedoch die versprochenen kleinen WWW-Server für kleine Computer (ebd., 82), damit Individuen und kleine Gruppen selbst im Netz veröffentlichen können.

Traditionelle Hypertextforschung (mit Ausnahme des explizit genannten Xanadu) hat sich nach der Aussage von BERNERS-LEE et al. auf die Benutzungsschnittstelle und das Erzeugen von Hypertextdokumenten konzentriert. Vernachlässigt wurden ihrer Meinung nach jedoch die Verteilung des Hypertextsystems und der Zugriff darauf von verschiedenen Computern aus. Diese werden im WWW durch einen Client-Server-Ansatz ermöglicht (ebd., 54), der im letzten Abschnitt bereits beschrieben wurde.

Da Daten im WWW über Links verbunden sind, kann sich ihre Struktur ständig ändern und sich änderndem Wissen anpassen (BERNERS-LEE et al. 92, 53). Im WWW beziehen sich Links nämlich auf Dateinamen. Wenn diese Namen beibehalten werden, können sich Dateinhalte beliebig ändern, ohne daß sich die Struktur des Hypertextes ändert. In kleinem Umfang erstellte Daten, z.B. von verschiedenen Projekten, können im WWW zu einem größeren Netz verbunden werden (BERNERS-LEE et al. 94, 76).

Menüs und Verzeichnisse lassen sich durch WWW-Dokumente darstellen, welche hauptsächlich aus Links auf andere Dokumente bestehen. Anfragen an Datenbanken werden im WWW durch URLs dargestellt, die auch Optionen enthalten können (BERNERS-LEE et al. 92, 53f). Anfrageergebnisse besitzen eigene Adressen, die jeweils aus der Adresse einer ‚Suchmaschine‘ und der eigentlichen Anfrage bestehen. Man kann also Links auf die Ergebnisse einer Datenbankanfrage legen, bei deren Verfolgung die Anfrage erneut ausgeführt wird (BERNERS-LEE et al. 94, 76).

Auch wenn es im WWW Indizes geben soll, ist dafür kein eigener Dokumententyp vorgesehen. Sie werden als normale HTML-Dokumente abgelegt, auf die mit Links verwiesen wird und die beim Suchen in einem Server gefunden werden können (BERNERS-LEE et al. 92, ebd.). Für hierarchische Verzeichnisse gibt es ebenfalls keinen eigenen Datentyp (BERNERS-LEE et al. 94, 78).

Schon früh wurde von den Entwicklerinnen des WWW die Notwendigkeit gesehen, daß unabhängige Erstellerinnen von Diensten wie Bewertungen, Indizes und Übersichten auf Daten zurückgreifen können müssen, die ihnen nicht gehören. Diese Dienste sollen den Nutzerinnen des Netzwerkes zur Verfügung stehen und die ‚Informationsexplosion‘ eindämmen helfen (BERNERS-LEE et al. 92, 56).

In Hypertextsystemen sind Namen von Links allgemein nicht direkt sichtbar, sondern durch die Browser versteckt. Manchmal müssen sie jedoch über E-Mail verschickt bzw. auf Papier notiert werden. Daher sollten Namen aus druckbaren Zeichen bestehen und möglichst kurz sein (BERNERS-LEE et al. 92, 54). Damit der Speicherort eines Hypertextdokumentes nachträglich verändert werden kann, erhält ein Dokument im WWW einen logischen Namen, der aus dem Namen der veröffentlichenden Organisation und der Bezeichnung des Dokumentes innerhalb der Organisation besteht. Die Adresse enthält weiterhin keine sich ändernden Eigenschaften eines Dokumentes wie sein Format oder seine Länge (ebd., 55).

### 3.2.3 WWW-Server

Beim ersten Blick auf ihre Beschreibung wirken WWW-Server wie einfachere FTP-Server: WWW-Server dienen dazu, Dateien auf Anfrage zu übertragen. Ihre Leistungsfähigkeit ist jedoch wesentlich größer, als dieser Vergleich andeutet. WWW-Server unterstützen nämlich die Verwaltung von Benutzerinnen und Paßwörtern, die Auswahl unter Dokumenten in verschiedenen Sprachen, das Protokollieren von Zugriffen und die Erstellung von dynamischen Dokumenten.

‚Apache‘ ist der Name eines public-domain WWW-Serverprogrammes unter UNIX, das im Dezember 1996 nach einer Untersuchung von NETCRAFT auf fast 250.000 Servern eingesetzt wurde. Dies bedeutet für die Untersuchung einen Marktanteil von 41% und macht Apache damit zu dem bei weitem erfolgreichsten Server (96). Er soll hier stellvertretend für andere Server besprochen werden.

Weil das WWW nur ein Übertragungsprotokoll (HTTP) und kein verteiltes Dateisystem benutzt (wie z.B. NFS von Sun), welches Zugriffsrechte im Netz verwaltet, muß ein WWW-Server ein eigenes Schutzsystem beinhalten.<sup>9</sup> In Apache können bestimmte Verzeichnisse gesperrt bzw. freigegeben werden, indem für sie festgelegt wird, welche Benutzerinnen (bzw. -gruppen) auf sie mit welchen HTTP-Methoden zugreifen können. Zu einem Benutzerinnennamen kann ein Paßwort vereinbart werden, welches die Benutzerin eingeben muß, um Zugang zu erhalten (ebd., Abschn. „AccessFileName directive“ u. „require directive“). Es ist außerdem möglich, den Zugriff in Abhängigkeit von dem Rechner, von dem aus auf den Server zugegriffen wird, freizugeben oder zu sperren. Dies ist nützlich, um intern in einer Institution den Zugriff auf bestimmte Seiten zu erlauben, auf die externe Benutzerinnen keinen Zugriff haben sollen (APACHE o.J.d).

Verschiedene (menschliche) Sprachen von Dokumenten werden im WWW über Dateiendungen im MIME-Standard (z.B. ‚en‘ für Englisch und ‚de‘ für Deutsch) angegeben<sup>10</sup> (APACHE o.J.a, Abschn. „AddLanguage“). In der Dokumentation zu Apache steht, daß diese Angabe weniger dazu dient, dem Browser die aktuelle Sprache mitzuteilen, als mit ihm eine Sprache auszuhandeln, im einfachsten Fall also auf den Sprachwunsch des Browsers mit dem Dokument in der gewünschten Sprache zu reagieren (ebd.). Um Leserinnen mitzuteilen, daß das von ihnen gewünschte Dokument auch in anderen Sprachen vorliegt, muß seine Autorin jedoch explizit Links zu den anderen Dokumenten setzen.

Oft wäre es für die Leserin eines Dokumentes interessant, von einem Dokument zu anderen zu springen, die *auf* es zeigen und nicht nur zu solchen, auf die Links vom aktuellen Dokument aus zeigen. Das Rückwärtsverfolgen einer Linkkette könnte z.B. zu Sekundärliteratur zum aktuellen Dokument führen oder zu einem Verzeichnis ähnlicher Dokumente wie dem vorliegenden. Es wäre auch für die Autorin interessant, weil sie dann die Autorinnen der Dokumente, die auf ihres zeigen, benachrichtigen könnte, wenn sie ihr Dokument ändert oder löscht. Apache kann immerhin protokollieren, auf welches Dokument ein WWW-Browser zuletzt zugegriffen hat, bevor er ein neues von Apache anfordert (APACHE o.J.c).

Der Zugriff auf bestehende Datenbanken erfolgt im WWW über virtuelle bzw. dynamische Dokumente, die ein WWW-Server erst dann aus Datenbankdaten erzeugt, wenn sie abgefragt werden. Mittels dynamischer Dokumente können Datenbanken zunächst nur abgefragt, nicht geändert werden, obwohl das Ändern über die schon besprochenen HTML-Formulare auch möglich ist (s. S. 37).

Dynamische Dokumente können beliebige Zusammenstellungen und Berechnungen enthalten, z.B. Suchanfragen an einen der zentralen Suchserver des WWW. Weil dynamische Dokumente normale WWW-Adressen haben und aus Standard-HTML bestehen, können sie von quasi allen WWW-Browsern dargestellt werden, auch solchen, die nicht die neuesten Techniken beinhalten. Daher können sie auch dazu dienen, WWW-Funktionen auf möglichst kompatible Art zu erweitern (s. Abschnitt 4.4 auf S. 115).

Apache ist in der Lage, dynamische Dokumente nach dem ‚Common Gateway Interface‘ (CGI) zu erstellen. CGI-Dokumente haben eine URL, wie andere HTML-Dokumente auch. Es handelt sich aber bei ihnen nicht um statische Dateien, die auf dem jeweiligen WWW-Server liegen und nur übertragen werden müssen, sondern um die dynamischen Ausgaben von Programmen, die ein Server erst nach der Anfrage durch Ausführen der Programme erstellt. Ihre Standardanwendung liegt in der schon erwähnten Verbindung eines WWW-Servers mit einer Datenbank, wobei Daten von dieser mittels eines CGI-Programmes an einen Browser weitergeleitet werden (NCSA o.J.a).

9 Sun und Microsoft bieten mittlerweile verteilte Dateisysteme für das WWW unter den Namen ‚WebNFS‘ bzw. ‚CIFS‘ an.

10 Weil Dateitypen wie HTML auch über Dateiendungen festgelegt werden, häufen sich diese am Ende eines Dateinamens, z. B. ‚hilfe.de.html‘ für ein deutsches HTML-Dokument.



Im Prinzip können beliebige Daten durch CGI-Programme aufbereitet werden. Da es sich bei CGI-Skripten um Programme handelt, die auf den Servern ausgeführt werden, stellen sie ein erhebliches Sicherheitsrisiko dar. Daher ist es gewöhnlichen Benutzerinnen und WWW-Autorinnen meist nicht gestattet, sie einzusetzen, was wiederum eine starke Einschränkung ihrer Möglichkeiten der Gestaltung von WWW-Anwendungen bedeutet (ebd.).

CGI-Programme können in einer beliebigen, auf dem System des Servers vorhandenen Programmiersprache geschrieben werden. Gewöhnlich werden sie als Skripte in einer Sprache wie Perl, TCL oder AppleScript formuliert (ebd.). Wenn ein CGI-Programm durch einen Server nach der Anfrage eines Clients aufgerufen wird, erhält es (unter UNIX) Daten in System-, also Umgebungsvariablen. Bei diesen Daten handelt es sich um Angaben über den Server und den Computer, von dem aus das Skript aufgerufen wurde. Wenn die Benutzerin sich einloggen mußte (s. oben), erhält ein Skript auch Angaben über diese (NCSA o.J.b).

Neben den statischen und den dynamischen HTML-Dokumenten gibt es ‚Server-parsed HTML documents‘ (SPML) als Zwischenform, auch ‚Server-side includes‘ (SSI) genannt. Als statische Dokumente existieren sie schon vor dem Abruf. Wie die dynamischen werden sie jedoch erst bei der Abfrage vervollständigt. In die statische Vorlage sind bei einem SPML spezielle Kommandos eingebaut, die es dem Server erlauben, spezielle Variablen, die Ausgabe von CGI-Programmen und UNIX-Shell-Skripten, eine Dateigröße, das Datum der letzten Änderung oder eine andere HTML-Datei einzufügen. Es ist sogar möglich, Ausdrücke anzugeben, die die Ausgabe wie den Kontrollfluß einer Programmiersprache steuern. Dazu gibt es die Befehle ‚if‘, ‚elif‘, ‚else‘ und ‚endif‘ (APACHE o.J.d). Mittels SPMLs können nicht nur Standardlayoutelemente und Datumsangaben automatisch auf dem laufenden gehalten, sondern auch Dokumente in Abhängigkeit von Browsereinstellungen modifiziert werden. Diese Einstellungen manifestieren sich in Variablenbelegungen auf der Serverseite.

Dokumentencaches (s. S. 35) finden sich immer häufiger nicht nur in Browsern, sondern auch in Servern. Dedizierte Server dienen als sogenannte Proxy<sup>11</sup>-Caches, die gegenüber anderen Servern als Browser auftreten und Daten, die ein Browser über sie anfordert, zwischenspeichern und an ihn weiterleiten. Browser müssen auf die Benutzung eines Cache-Servers vorbereitet sein, was aber bei den neueren der Fall ist. Die Adresse einer zu benutzenden Proxy kann bei ihnen als Parameter gewählt werden. Proxy-Caches werden häufig in eigenen Netzen zusammengeschlossen, so daß die Beantwortung einer Anfrage dem Cache überlassen wird, der gerade am wenigsten beansprucht ist.

Durch eine Proxy kann auch die Funktionalität des WWW *zwischen* Client und Server erweitert werden. So stellen BROOKS et al. mit dem Konzept der ‚OreOs‘<sup>12</sup> Filter vor, die sowohl auf Abteilungsservern eingesetzt werden können als auch dem jeweils lokalen Rechner einer Benutzerin, auf dem sie ihren Browser laufen läßt (95, 541f). Dadurch können für eine Erweiterung Standard-Browser benutzt werden, solange sie proxyfähig sind. Weiterhin werden Benutzerinnen nicht dazu gezwungen, eine zentrale Erweiterung außerhalb der Kontrolle ihres Rechners zu benutzen. Der erste OreO diente dazu, URLs zu überprüfen und Fehler direkt an einen Browser zurückzumelden, Datentransferraten zu messen und die von einer Gruppe aufgerufenen WWW-Seiten anzuzeigen. Ein anderer OreO konnte Anmerkungen zu WWW-Dokumenten innerhalb einer Benutzerinnen-gruppe verarbeiten (ebd., 542f), den BROOKS et al. jedoch nicht weiter vorstellen .

11 ‚Proxy‘ ist das englische Wort für ‚Stellvertreterin‘.

12 benannt nach der amerikanischen Entsprechung der ‚Prinzenrollen‘-Kekse, weil sie eine Server- und eine Browserseite besitzen, die durch eine funktionale ‚Füllung‘ zusammengehalten wird (BROOKS et al. 95).

### 3.2.4 WWW-Editoren

Wie beschrieben, war der erste WWW-Browser, den BERNERS-LEE entwickelt hatte, gleichzeitig ein Editor, mit dem HTML-Dokumente erstellt werden konnten. Über seine Funktionalität ist nichts bekannt. Heute gängige Browser dienen nur der Darstellung von Dokumenten. Das ist unbefriedigend für die Erstellung wissenschaftlicher Texte und anderer WWW-Seiten, weil HTML-Codes dann mühsam in einem Texteditor erstellt und Links als Adressen eingefügt werden müssen, anstatt z.B. durch ‚Drag-and-Drop‘<sup>13</sup>. Die Erstellung einiger HTML-Befehle durch Eingabe ihres Codes ist jedoch oft nicht zu vermeiden, weil das von gängigen Browsern darstellbare HTML-Idiom sich ständig ändert, und zwar schneller, als neue Versionen von HTML-Editoren erstellt werden.

Für Hypertexte gilt andererseits NELSONS Kritik am WYSIWYG-Konzept (S. 59), daß durch die Forderung nach nur einer einzigen Sicht auf ein Dokument viel verschenkt wird. Das Konzept schon von WYSIWYG-Paradebeispielen wie populären Textverarbeitungsprogrammen unter Fenstersystemen nicht durchgehalten, wenn sie eine Gliederungssicht auf ein Dokument erlauben, in welcher Überschriften verschoben und Texte ausgeblendet werden können. Weiterhin bezieht sich die WYSIWYG-Forderung hauptsächlich darauf, daß Daten so auf dem Bildschirm erscheinen sollen, wie sie gedruckt werden. Sie gilt also für Hypertexte nur bedingt, die ja gerade nicht daraufhin optimiert sind, gedruckt zu werden, teilweise gerade über ihre Undruckbarkeit definiert sind (s. S. 30).

Für das wissenschaftliche Arbeiten wäre es günstig, die technische Grenze zwischen Lesen, Ordnen und Schreiben aufzuheben. Wie wir im ersten Kapitel gesehen haben, gehört zum Schreiben auch das Lesen von existierenden Dokumenten und zum aktiven Lesen auch das Notieren von Zusammenfassungen, was bei der Verwendung verschiedener Werkzeuge für die einzelnen Aufgaben nicht möglich ist.

Es gibt zwar inzwischen Textverarbeitungsprogramme, die zu WWW-Browsern/Editoren erweitert sind (z.B. ‚Word‘ von Microsoft und ‚StarWriter‘ von Star Division, s. BAGER et al. 96, 194ff), allerdings sind sie durch ihre papierorientierten Funktionen zu schwerfällig für flüssiges Arbeiten. Außerdem ist ihr Entwicklungszyklus wesentlich länger als der von WWW-Browsern, so daß sie die jeweils letzten HTML-Erweiterungen selten unterstützen. (Diese Erweiterungen sind zwar für wissenschaftliche Texte nicht immer nötig, der Wechsel zwischen verschiedenen Browsern je nach Anwendungszweck ist Benutzerinnen aber auch nicht zuzumuten.)

Es sollen nun kurz einige gängige Browser-Editor-Programme für das WWW vorgestellt werden, zusätzlich ein experimenteller des W3C. Die bekanntesten Editoren sind Frontpage von Microsoft, AOLPress (früher GNNPress) von AOL und PageMill von Adobe (BAGER et al. 96, 190ff). Netscape bietet eine erweiterte Version seines Browsers mit dem Namenszusatz ‚Gold‘ an, der ebenfalls Editierfunktionen eingebaut hat (ZIVADINOVIC 96, 112f). Der Editor des W3C heißt Amaya und soll ausdrücklich dem Testen von WWW-Erweiterungen dienen.

Frontpage ist ein WYSIWYG-Editor. Das geht soweit, daß es in ihm nicht möglich ist, HTML in Rohform zu editieren. Das Programmpaket beinhaltet nicht nur einen Editor, sondern auch einen Server einschließlich eines speziellen Verwaltungsprogrammes für Dokumente auf einem Server (BAGER et al. 96, 194). Frontpage erlaubt nicht nur das Erstellen von Dokumenten, sondern auch das Pflegen einer eigenen ‚Site‘. Dabei handelt es sich um einen Rechner, der mit dem Internet verbunden ist und WWW-Seiten bereithält. Frontpage ist auch von Anfängerinnen bedienbar.

Die relativ schwierige Erstellung von CGI-Programmen für dynamische Dokumente wie Suchseiten wird durch vorgefertigte ‚Web-Bots‘ erleichtert. Dabei handelt es sich um auf dem Server

<sup>13</sup> Darunter versteht man bei graphischen Benutzungsoberflächen allgemein die Anwendung einer Funktion auf ein Objekt, indem man das Objekt mit der Maus auf ein Funktionssymbol oder ein anderes Objekt zieht und dort ‚fallenläßt‘. Bei der Erstellung eines Links wäre ein Symbol für das Zieldokument des Links das Objekt, das man auf der Position im Ausgangsdokument, an der der Link eingefügt werden soll, fallenläßt.

lauffähige Programme, also eine spezielle Art der schon beschriebenen CGI-Skripte. Das Server-Verwaltungsprogramm des Frontpagepaketes ist in der Lage, alle Seiten auf einem Frontpage-Server graphisch als Liste oder Netz darzustellen. Auf einem eigenen Server kann eine Autorin damit eigenes Material ordnen, wenn auch nur auf Dateiebene (ebd.).

Bei AOLPress handelt es sich ebenfalls um einen WYSIWYG-Editor, allerdings auch mit Browser-Funktionalität. In der ersten Version war er jedoch nicht in der Lage, die neueren HTML-Erweiterungen darzustellen. Sie können immerhin in einem HTML-Textmodus eingegeben werden. Ähnlich wie Frontpage, aber ohne mitgeliefertem Server (der von AOL ebenfalls erhältlich ist und mit dem es besonders gut zusammenarbeitet) erlaubt AOLPress das gemeinsame Verwalten mehrerer Seiten in einem sogenannten ‚MiniWeb‘. Es können die Links zwischen den Dokumenten in einem MiniWeb überprüft und in einer Netzansicht dargestellt werden (ebd., 191). Dokumente in einem MiniWeb können auch gemeinsam gedruckt werden. AOLPress ruft sie dafür nacheinander auf und druckt sie über den Standarddialog des entsprechenden Betriebssystems. Er ist jedoch nicht in der Lage, Links zwischen gedruckten Dokumenten durch Querverweise mit Seitennummern zu ersetzen.

PageMill von Adobe war der erste erfolgreiche WYSIWYG-Editor für das WWW. Bilder, Links und Dialogelemente (Checkboxes etc.) können per Drag-and-Drop eingefügt und über eingebaute Werkzeuge geändert werden. HTML kann nicht direkt bearbeitet werden. Bei SiteMill handelt es sich um eine erweiterte Version von PageMill. SiteMill erlaubt es, ganze Server zu verwalten. Es enthält alle Funktionen von PageMill und zusätzlich solche, um mehrere HTML-Dateien und ihre Links auf einmal zu ändern. SiteMill kann Dokumente in Listen darstellen und angeben, wenn sie nicht von anderen Dokumenten benutzt werden oder falsche Links enthalten. Wenn man eine Datei verschiebt, werden lokale Links automatisch angepaßt (EHRMANN 96, 80ff).

Beim Netscape Navigator Gold handelt es sich um den erfolgreichen Browser von Netscape, der um WYSIWYG-Editierfunktionen erweitert ist. Erstellte Seiten können per FTP oder HTTP an einen WWW-Server übertragen werden. Es ist möglich, HTML in Textform einzugeben. Die komfortable Eingabe per Maus ist allerdings auch nicht für alle vom Browser unterstützten HTML-Elemente möglich (ZIVADINOVIC 96, 112f).

Amaya ist ein WWW-Browser und -Editor des W3C. Seine Autorinnen bezeichnen ihn als WWW-Programm der vierten Generation. Als erste Generation sehen sie reine Browser, als zweite Programme zur Unterstützung der Erstellung von WWW-Seiten, als dritte HTML-Editoren. Der nächste Schritt ist ihrer Meinung nach die Einbeziehung von Strukturen und Erweiterungen wie Tabellen, (Java-)Applets und Style Sheets (Formatvorlagen). Amaya soll v.a. dem Testen neuer Erweiterungen von WWW auf der Clientseite dienen. Erweiterbarkeit stand daher bei seiner Entwicklung im Vordergrund und wird durch ein Application Programming Interface (API) unterstützt (s. QUINT 97).

Amaya ist aber nicht nur ein Testprogramm, sondern eines zum täglichen Arbeiten, welches sich durch Benutzungsfreundlichkeit auszeichnet. Das Bearbeiten von Dokumenten im HTML-Format erfolgt z.B. in WYSIWYG-Manier. Das Veröffentlichen auf WWW-Servern außerhalb des lokalen Netzes wird explizit unterstützt. Es wird erwartet, daß CSCW-Unterstützung eingebaut werden wird. Zusätzlich soll Amaya auf verschiedenen Plattformen zur Verfügung gestellt werden (zunächst auf SUN Solaris und Linux, später auch MS Windows) und in akzeptabler Geschwindigkeit laufen. Das Programm wird zusammen mit dem Quellcode veröffentlicht (QUINT 96).

Insgesamt bieten die besprochenen Editoren zuwenig Textverarbeitungsfunktionalität und zuwenige Möglichkeiten, Dokumente zu organisieren. Wenn diese letzte Aufgabe überhaupt unterstützt wird, dann meist in Form von Server-Verwaltungswerkzeugen für Systemadministratorinnen, nicht zum schnellen Ordnen privater Daten.

### 3.2.5 Möglichkeiten zur Erweiterung des WWW

WWW-Browser können durch kleine Programme in Form von ‚Plug-Ins‘ und ‚Java-Applets‘ erweitert werden. Dadurch können Browser neue Datenformate darstellen und eingeben lassen, zusammen mit CGI-Skripten auf Servern also das WWW erweitern. Die Browser-Erweiterungsmöglichkeiten werden hier besprochen, weil im nächsten Kapitel nach der Spezifikation eines Hypertextsystems für das wissenschaftliche Schreiben seine Umsetzung als WWW-Erweiterung vorgestellt wird.

Plug-Ins, die von Netscape für ihren Browser ‚Navigator‘ eingeführt und von Microsoft für ihren Browser übernommen wurden, bestehen aus Maschinencode der jeweiligen Plattform. Sie dienen dazu, bestehende Programme, die auch schon plattformabhängig geschrieben wurden, in einen Browser einzubinden (NETSCAPE o.J.d). Plug-Ins sind relativ klein und können mit gängigen Entwicklungswerkzeugen in C oder C++ geschrieben werden. Sie können ein Browser-Fenster für Ein- und Ausgabe benutzen, auf Tastatur- und Mauseingaben reagieren und WWW-Daten anfordern und verschicken (ebd.). Für ein Plug-In werden ein oder mehrere Datentypen (nach dem Internet-MIME-Standard) definiert, auf die sie reagieren, z.B. neue Grafikformate. Sie können entweder in HTML-Seiten eingebundene Daten in einem rechteckigen Bereich (ähnlich wie Bilder) darstellen oder das ganze Browserfenster für völlig eigenständige Datentypen benutzen (z.B. Dateien nach dem PDF-Standard von Adobe, einer PostScript-Weiterentwicklung). Plug-Ins müssen installiert sein, bevor eine Benutzerin Daten anfordert, die sie darstellen sollen. Sie werden nicht zusammen mit den jeweiligen Daten übertragen (ebd.).

Java wurde von Sun entwickelt. Es handelt sich dabei um eine Programmiersprache mit dazugehöriger Laufzeitumgebung. Es entstammt einem Projekt, welches ein Betriebssystem für eingebettete Systeme hervorbringen sollte. Später wurde klar, daß Java für die Entwicklung von verteilten Anwendungen im WWW eingesetzt werden kann (GOSLING U. MCGILTON 95, 12). Java-Programme können auf eigene Standardobjektbibliotheken zurückgreifen, u.a. für graphische Benutzungsoberflächen und Netzwerkfunktionen (ebd., 13).

Java-Programme werden in einen speziellen ‚Bytecode‘ kompiliert, einen Pseudomaschinencode, der auf verschiedenen Hardware-Plattformen ausgeführt werden kann. Dafür muß jeweils eine ‚Virtuelle Maschine‘ implementiert sein, die die eigentliche Ausführung übernimmt. Bei der Virtuellen Maschine handelt es sich um eine Implementation einer abstrakten Computerarchitektur (ebd., 14f).

Java-Programme sind ‚hardwareunabhängig‘ für alle Hardware-Betriebssystemplattformen, für die ein solcher Java-Pseudoprozessor vorliegt. Das erste große Java-Programm für Endbenutzerinnen ist der WWW-Browser ‚HotJava‘ (ebd., 72). Er stellt außerdem eine Laufzeitumgebung für kleine Java-Programme, sogenannte ‚Applets‘, dar. Sie werden gleichzeitig mit den entsprechenden Daten in den Browser geladen (ebd., 75) und müssen daher nicht vor Benutzung installiert werden. Java-Applets können z.B. Animationen und Ton abspielen und interaktive Elemente bereitstellen. Sie benutzen entweder einen rechteckigen Ausschnitt einer dargestellten HTML-Datei (ähnlich wie Plugin-Daten) oder öffnen ein eigenes Fenster. In den neuen WWW-Browsern von Netscape und Microsoft können ebenfalls Applets eingesetzt werden.

### 3.3 HyperWave (Hyper-G)

#### 3.3.1 Anforderungen

„HyperWave“<sup>14</sup> ist der Name eines Hypermediasystems im Einsatz, das am ‚Institut für Grundlagen der Informationsverarbeitung und Computergestützte Neue Medien‘ (IICM) an der TU Graz entwickelt und etwa 1990 implementiert wurde (s. MAURER et al. 90). Seine Entwicklerinnen sind auch für das österreichische BTX-System verantwortlich (KAPPE 93, 1). HyperWave wurde in Anlehnung an die Ideen NELSONS als Publikations- und Kommunikationssystem konzipiert (KAPPE U. PANI o.J., 3).

Die Grundanforderung an HyperWave ist weit gefaßt: „The system should allow access to all kinds of information one can think of“ (KAPPE 91, 12). Dabei steht der *Zugriff* auf Informationen im Vordergrund, nicht das *Erstellen* von ihnen. Diese Informationen können in der Form von Text, Zeichnungen, Rasterbildern, Ton, Film, Animationen, Karten, Beiträgen in ‚Schwarzen Brettern‘ und Konferenzsystemen, E-Mail und Bildschirmdialogen erscheinen (ebd.). HyperWave sollte die Verwaltung sehr vieler Daten unterstützen und leicht erweiterbar sein (bezüglich seiner Benutzungsschnittstelle und der verwaltbaren Datentypen) (ebd., 13). Das System sollte sowohl von Expertinnen als auch Nichtexpertinnen benutzt werden können (ebd.). Es sollte mehrere Sprachen unterstützen, sowohl auf seiten des Interfaces als auch der verwalteten Daten, außerdem mehrere Navigationshilfen (ebd., 14).

Das System sollte weiterhin Zugriffsrechte verwalten (ebd., 16), jedoch auch anonym benutzt werden können, mit verschiedenen Abstufungen der Anonymität (s. weiter unten). Auch anonyme Benutzerinnen sollten Dokumente und Links verändern können und dabei vom System durch Editierwerkzeuge unterstützt werden (ebd., 14f). Manche Dienste sollten kostenpflichtig sein und automatisch abgerechnet werden (ebd., 15). Auf HyperWave sollte über Terminals mit unterschiedlichen Fähigkeiten zugegriffen werden können (ebd., 15f). Benutzerinnen können im heutigen HyperWave-Netz auf mehrere Arten zugreifen: Sie können eine Suchanfrage formulieren, die sich entweder nur auf Objektattribute oder auf Volltext bezieht, sie können die Hierarchie von Collections oder eine Tour verfolgen (die beiden Begriffe werden weiter unten erklärt) und schließlich von ihnen selbst oder von anderen erstellte Links aktivieren (KAPPE 93, 76f).

Ursprünglich war HyperWave für die Benutzung in einem Local Area Network (LAN) konzipiert worden (KAPPE 93, 1). Seine Entwicklerinnen bezeichnen es als „massively distributed hypermedia system“. Es soll über das ganze Internet verteilt werden können. Dies sehen sie als Verbesserung gegenüber ebenfalls verteilten Hypermedia-Systemen wie ‚Intermedia‘, ‚Hyperties‘ und (das in dieser Arbeit schon beschriebene) ‚NoteCards‘, weil auf diese höchstens mittels verteilter Dateisysteme zugegriffen werden kann, deren Reichweite beschränkt ist. Außerdem unterstützen sie nur wenige Plattformen (KAPPE U. PANI, ebd.). Das Internet charakterisieren KAPPE U. PANI dabei als „real-world environment of fast LANs connected by slower WANs“ (ebd., 4). Die Auswirkungen dieser Sicht auf das Design ihres Systems wird später noch deutlich.

#### 3.3.2 Entwurf und Implementierung

HyperWave ist ein Client-Server-System. KAPPE U. PANI nennen folgende Vorteile des Client-Server-Konzeptes für verteilte Hypermediasysteme: Die Verarbeitung von Ein- und Ausgaben erfolgt schneller, Verzögerungen über Satellitenleitungen fallen weniger ins Gewicht (weil Daten in größeren Blöcken übertragen werden), Benutzerinnen können das System über ihnen vertraute, z.B. grafische Oberflächen von verschiedenen Plattformen aus bedienen und schließlich können Client-Pro-

14 HyperWave wurde unter dem Namen ‚Hyper -G‘ erstellt und zunächst auch verbreitet. Das ‚G‘ stand dabei für den Entwicklungsort Graz (MAURER et al. 90, 125).

gramme auf verschiedene Server mit verschiedenen Protokollen zugreifen und die gelieferten Daten einheitlich darstellen (ebd., 5f).

Benutzerinnen von HyperWave sind dem System in vier Abstufungen bekannt: anonym, anonym identifiziert, halbidentifiziert oder identifiziert. Anonyme Benutzerinnen (die sich z.B. über ein öffentliches Terminal einloggen) können nur lesend auf das System zugreifen, eventuell von ihnen vorgenommene Einstellungen werden auch nicht gespeichert. Einige Verwaltungsdaten werden jedoch trotzdem erfaßt (KAPPE 91, 33). Diese Daten dienen u.a. dazu, daß eine Benutzerin in einen vorherigen Zustand gelangen, also ein ‚Undo‘ ausführen kann (ebd., 45). Den anonym identifizierten Modus erreicht man, indem man das System ein Pseudonym und ein Paßwort für eine oder mehrere Sitzungen generieren läßt. Das System kann diese nicht mit einem Klarnamen einer Benutzerin in Verbindung bringen. Das Sichern in privaten Dokumenten und das Speichern von Einstellungen sind in diesem Modus möglich, jedoch nicht das Ändern von öffentlichen Dokumenten (KAPPE 91, 33).

Im halbidentifizierten Modus ist eine Benutzerin dem System mit ihrem Klarnamen bekannt und darf auf alle Dokumente schreibend zugreifen (in Abhängigkeit von ihren Zugriffsrechten). Andere Benutzerinnen erfahren jedoch nur ein Pseudonym, von dem auch mehrere benutzt werden können. Im identifizierten Modus schließlich erfahren auch andere Benutzerinnen den Klarnamen einer Benutzerin, wenn sie z.B. E-Mail von ihr erhalten (ebd.).

Alle Objekte in HyperWave wie Dokumente, Links und Verzeichnisse (Collections) besitzen Attribute, die über ihre Autorin, ihr Erstellung-, Zugriffs-, Änderungs- und Ablaufdatum, ihre Sprache, ihre Zugriffsrechte und ihren Preis Auskunft geben (ebd., 35). Der Preis eines Objektes wird von seiner Autorin festgelegt. Wenn er größer als Null ist, wird er vor Anforderung des Objektes angezeigt. Er kann auch negativ sein. Anonyme Benutzerinnen können Objekte mit Preisen an bestimmten Terminals nach Einschieben einer Chipkarte abrufen, bei allen anderen Benutzerinnen werden die Kosten ihrer Zugriffe über Konten abgerechnet (ebd., 46).

HyperWave verwaltet Hypertext in Dokumenten. Zugriff darauf erlaubt ein sogenannter Dokumentenserver (Document server), eine objektorientierte Datenbank, die speziell für HyperWave entwickelt wurde (ebd., 50). An den Dokumentenserver schicken Browser Dokumentenbezeichner und erhalten das jeweilige Dokument als Antwort (KAPPE U. PANI o.J., 6). Dokumente verwandten Inhalts können durch spezielle Links zu ‚Clustern‘ gruppiert werden. In solchen Clustern können Dokumente von verschiedenen Unterprogrammen dargestellt werden, was die Erweiterbarkeit vereinfacht. Cluster können außerdem Texte in verschiedenen Sprachen enthalten (KAPPE 91, 19ff). Es hat den Anschein, als ob es in HyperG zunächst nicht möglich war, eine Graphik direkt in einen Text einzubinden. Cluster sollten daher zumindest ermöglichen, daß z.B. ein Text- und ein Graphikdokument, die zusammengehören, gebündelt werden konnten.

Dokumente sind in ‚Collections‘ verzeichnet, die selbst wieder Collections enthalten können und Dateiverzeichnissen in herkömmlichen Betriebssystemen entsprechen. Da ein Dokument oder eine Collection sich in mehreren Collections befinden kann, ergibt sich ein gerichteter azyklischer Graph als Struktur (ebd., 28). Jedes Dokument und jede Collection befindet sich in mindestens einer Collection. Die oberste heißt Wurzelcollection (ebd., 41f). Eine Folge von Dokumenten kann zu einer ‚Tour‘ aneinandergereiht werden. Eine Tour stellt eine Lektion oder einen ‚Associative Trail‘ im Sinne BUSHS (91) dar (ebd., 29).

Dokumentenbezeichner dienen in verteilten Hypertextsystemen generell sowohl dazu, ein Dokument eindeutig zu identifizieren als auch seine Lieferung von einem Server anzufordern. Diese beiden Zwecke stehen in einem Spannungsverhältnis zueinander. Der erste legt nahe, identischen Dokumenten auch gleiche Bezeichner zu geben. Für die zweite Anforderung ist es günstiger, Dokumente nach Servern als Speicherorten zu benennen. Dadurch erhalten jedoch identische Dokumente auf zwei verschiedenen Servern unterschiedliche Bezeichner (KAPPE U. PANI o.J., ebd.).

In HyperWave werden Objekte wie Dokumente, Links und Anker durch eine 32-Bit-Zahl benannt (KAPPE U. PANI o.J., 10f). Sie bezieht sich immer auf den Server, auf dem das entsprechende Objekt gespeichert ist, ist also lokal. Eine globale Objektzahl wird daraus, wenn ihr eine weitere 32-Bit-Zahl als Serverbezeichner vorangestellt wird. Es können also etwa vier Milliarden Server mit jeweils vier Milliarden Dokumenten adressiert werden. Servernummern stimmen aus Flexibilitätsgründen nicht mit IP-Adressen überein, daher müssen HyperWave-Server registriert werden, wobei ihre Nummer festgelegt und ihrer IP-Adresse zugeordnet wird (ebd., 16).

Im Gegensatz zum WWW werden Links in HyperWave nicht direkt in den Dokumenten abgelegt, auf die sie sich beziehen. Das hat den Vorteil, daß Links auch in Dokumente eingefügt werden können, die selbst nicht änderbar sind, z.B. weil sie von der Autorin geschützt wurden oder auf einer CD-ROM liegen. Die externe Speicherung ermöglicht außerdem das Einfügen privater Links durch verschiedene Benutzerinnen in ein beliebiges Dokument. Schließlich können Links in zwei Richtungen verfolgt werden (,Bifollowability‘). Das ist für Leserinnen interessant, weil sie alle Links angezeigt bekommen können, die auf das aktuelle Dokument verweisen. Außerdem kann das System dadurch Links anpassen, wenn das Dokument, auf welches sie zeigen, modifiziert oder gelöscht wird (ebd., 8).

Links beziehen sich nicht direkt auf Dokumente, sondern sind über sogenannte ,Anker‘ an ihnen befestigt. Diese besitzen neben Informationen über die Autorin und die Zugriffsrechte Positionsdaten, die sich auf das Dokument beziehen, zu dem sie gehören (ebd., 10f). Anker und damit Links können an allen unterstützten Dokumententypen befestigt werden, also auch an Bildern, Grafiken, Tondokumenten und Animationen (KAPPE 91, 25). ,Aktive Anker‘ führen jeweils ein kleines Programm aus, wenn sie angewählt werden. Ein solches Programm kann z.B. eine Auswahl aus verschiedenen Dokumenten erlauben (ebd., 26).

Links werden in sogenannten Link-Servern verwaltet. Sie liegen daher getrennt von Dokumenten. Ein Link-Server speichert außerdem andere HyperWave-Daten wie Dokumentenbeschreibungen, Linkanker, Collections, Touren etc. und ist als objektorientierte Datenbank implementiert. Daher kann die Attributenliste eines Objektes leicht erweitert werden (KAPPE U. PANI o.J., 9). Der Link-Server ist für die Vergabe von Objektzahlen verantwortlich. Er stellt sicher, daß eine Nummer nur einmal (und nach Löschen des entsprechenden Objektes nicht erneut) vergeben wird und vergibt an modifizierte Objekte eine neue Nummer (ebd.). Nur im Link-Server werden Objektzahlen auf die eigentlichen Objekte abgebildet. Deshalb müssen Änderungen wie z.B. das Verschieben eines Objektes von einem Server auf einen anderen nur in den beteiligten Link-Servern bekannt gemacht werden (ebd., 10).

Im ursprünglichen Design von HyperG sind neben den üblichen statischen auch dynamische Links vorgesehen. Diese werden erst bei ihrer Aktivierung vom System generiert. Sie können z.B. dazu dienen, zu jedem Wort den Sprung zu seinem Eintrag in einem Wörterbuch oder in einem Lexikon zu erlauben (KAPPE 91, 22). Der Link-Server speichert zu einem Dokument neben seiner Identifikationsnummer, seinem Typ, seiner Autorin und seinem Titel auch Zugriffsinformationen. Dazu gehört der Rechner, auf dem es sich befindet und das Protokoll, über das es abgerufen werden kann. Das Link-Objekt enthält neben den üblichen Attributen einen Typ, nach dem Links gesucht und gefiltert werden können. Allerdings kann dieser Typ von einer Benutzerin nicht geändert werden (ebd., 10f).

Obwohl der Link-Server objektorientiert ist und Daten in Objekten ablegt, enthalten diese Objekte keine Referenzen auf andere Objekte. Verbindungen zwischen Objekten sind vielmehr durch Relationen gespeichert, die in beliebige Richtungen abgefragt werden können. In der zweistelligen Dokument-Anker-Relation werden Anker und die Dokumente gespeichert, in denen sie sich befinden. Die Relation kann nicht nur dazu dienen, alle Anker zu einem bestimmten Dokument abzufragen, sondern auch, zu welchem Dokument ein Anker gehört (ebd., 11). Die dreistellige

Link-Anker-Relation verbindet zwei Anker mit einem Link zwischen ihnen. Die Relation ist gerichtet, ein Anker wird als Quelle und der andere als Ziel aufgefaßt. Es kann aber auch vom Ziel zur Quelle gesprungen werden (ebd., 11f).

Wenn ein Browser ein bestimmtes Dokument darstellen soll, fordert er es zunächst an, indem er eine Anfrage mit dem entsprechenden Bezeichner an den Link-Server schickt. Dieser liefert ihm das zu dem Bezeichner gehörende Dokument und die Anker dazu, wenn die Benutzerin die nötigen Zugriffsrechte hat. Der Browser benutzt die Zugriffsinformationen in dem Dokumentenobjekt zur Anforderung des entsprechenden Inhaltes, z.B. Textes. Es stellt diesen zusammen mit den Ankern an den entsprechenden Positionen dar. Wenn die Benutzerin einen bestimmten Anker in dem Dokument aktiviert, schickt der Browser dessen Nummer wieder an den Link-Server. Über die Link-Anker-Relation findet dieser den entsprechenden Link und Zielanker und schließlich das dazugehörige Dokument. Dieses schickt der Server zurück an den Browser (ebd., 12).

KAPPE U. PANI nennen auch Nachteile ihres Konzeptes eines eigenen Linkservers: Es benötigt komplexere Client- und Serverprogramme als das einfache Konzept, und es entstehen zusätzliche Daten zwischen Client und Link-Server (ebd.). Es wird von der Beobachtung aus anderen verteilten Informationssystemen ausgegangen, daß sich zwischen Dokumenten, die auf demselben Server gespeichert sind, viele Links befinden, zwischen solchen auf verschiedenen Servern jedoch relativ wenige (ebd., 15). Es gibt daher in HyperWave einen Link-Server für jedes LAN, was für einen schnellen Zugriff auf Links sorgt.

Link-Server sind über Relationen verbunden, in denen Objekte vorkommen können, die von mehr als einem Link-Server verwaltet werden. So kann z.B. über einen lokalen Anker, der mit einem entfernten Dokument verbunden ist, ein Link auf dieses gesetzt werden. Relationen werden von allen beteiligten Servern geführt. Ein Link auf ein entferntes Dokument wird dadurch auch auf dem Server gespeichert, auf dem sich das Dokument befindet (ebd., 16). Anfragen, die sich auf Anker oder Links beziehen, müssen immer an den Link-Server gerichtet werden, in dem das jeweilige Objekt gespeichert ist (ebd., 16f).

Da Relationen sich auf nicht mehr als zwei verschiedene Server beziehen (es gibt also höchstens ein entfernt liegendes Objekt in einer Relation), muß bei einer Änderung auch nur höchstens ein anderer Server informiert werden. Dies gilt für eine beliebige Anzahl von Servern im Netz (ebd., 17). Wenn jedoch viele Server einen Link auf ein bestimmtes Dokument enthalten, müssen alle diese nacheinander informiert werden. Das hat die Übertragung vieler Daten zur Folge und führt mit nur geringer Wahrscheinlichkeit zum Erfolg, weil nicht alle Server jederzeit erreichbar sind (KAPPE o.J., 5). Der Abgleich von Daten zwischen Servern wird auf S. 75 diskutiert. Hier soll zunächst nur der Ansatz in neueren Implementierungen von HyperWave beschrieben werden.

Dort wird ein Algorithmus namens ‚P-Flood‘ benutzt. Dieser geht auf den ‚Flood-D‘-Algorithmus des ‚Harvest‘-Systems von der Universität Colorado in Boulder zurück. Das Grundprinzip von P-Flood ist, daß Daten nicht von einem Server nacheinander an alle betroffenen Server geschickt werden, sondern nur an eine bestimmte Anzahl  $k$  von ihnen. Diese schicken sie an weitere  $k$  Server. Daten werden dadurch über das Netzwerk ‚geflutet‘, und es wird insgesamt die Netzwerk- und Serverlast auf das Netzwerk verteilt (ebd., 6).

P-Flood garantiert allerdings nur eine schwache Konsistenz, bei dem das Netzwerk also für eine gewisse Zeit inkonsistent sein kann, jedoch sicher gegen einen konsistenten Zustand konvergiert. Updates werden immer noch von einem einzigen Server eingeleitet, der jedoch von der Art des geänderten Objektes abhängt: Für ein Dokument ist es der Server, auf dem es sich befindet, für einen Link der, auf dem sich das Dokument befindet, von dem er ausgeht. Diese Server müssen während des Updates immer noch am Netz sein, was sie aber auch sein müssen, um zu überprüfen, ob eine Benutzerin überhaupt genug Zugriffsrechte hat, das Dokument zu ändern (ebd., 8). Als



Anforderungen, die bei der Entwicklung des ‚P-Flood-Algorithmus‘ im Vordergrund standen, nennt KAPPE:

- Geschwindigkeit: Die Dauer der Inkonsistenz soll so gering wie möglich gehalten werden.
- Robustheit: Alle Meldungen müssen nach und nach bei allen Servern ankommen, auch wenn einige davon zunächst nicht erreichbar sein sollten.
- Skalierbarkeit: Die Laufzeit und der Umfang der Daten soll nicht zu stark von der Anzahl der Server abhängen.
- Automatisierung: Kanäle für die ‚Flut‘ sollen automatisch konfiguriert werden.
- Priorisierung: Die Verbreitungszeit und der Ressourcenverbrauch einer Mitteilung sollen wählbar sein (ebd., 8f).

Diese Anforderungen werden in P-Flood durch einen probabilistischen Ansatz erfüllt. Der Algorithmus besitzt einen Parameter  $p$ , der angibt, an wieviele andere ein Server eine Liste von Update-Aufforderungen weiterleitet. Diese Weiterleitungen werden in jedem Zeittakt von allen Servern durchgeführt, jedoch nicht genau synchron. Solange ein Server nicht erreichbar ist, werden die Mitteilungen für ihn aufbewahrt (ebd., 9f). KAPPE hat Simulationen durchgeführt, um das Verhalten des Algorithmus‘ zu testen und einen günstigen Wert für den Parameter  $p$  zu bestimmen. Es hat sich dabei gezeigt, daß der Algorithmus tatsächlich skalierbar und schnell ist und mit dem Ausfall von Servern zurecht kommt. Der von ihm verursachte Netzverkehr ist im Vergleich zum Gesamtaufkommen vernachlässigbar (ebd., 10ff).

PITKOW U. JONES bezweifeln jedoch, daß der P-Flood-Algorithmus mit der schnellen Änderung des Internets mithalten kann. (Sie liefern jedoch keine empirischen Hinweise für ihre Vermutung.) In ihrem eigenen Entwurf eines Hypertextsystems auf Basis des WWW, in dem Links extern zu Dokumenten verwaltet werden, sehen sie daher nur Benachrichtigungen zwischen den beiden Servern vor, zwischen denen sich jeweils ein Link geändert hat (96, Abschn. ‚Scalability of Notification‘). Diese Lösung wurde in HyperG zunächst auch eingesetzt und wird von KAPPE dahingehend kritisiert, daß sie bei vielen Links auf ein und dasselbe Dokument nicht effizient genug ist (o.J., S. 5). PITKOW U. JONES gehen für diesen Fall davon aus, daß eine Autorin wählen kann, ob sie Autorinnen anderer Seiten benachrichtigen möchte, die auf eines ihrer Dokumente verweisen (ebd., Abschn. ‚Publishers, Subscribers, and Policies‘).

Objekte des Link-Servers werden in Caches zwischengespeichert (KAPPE U. PANI o.J., 17). Üblicherweise tritt bei Caches das Problem auf, daß sie eine alte Version eines Objektes vorliegen haben und weiterleiten, obwohl sich am ursprünglichen Ort schon eine neue befindet. Weil neue Versionen von Objekten in HyperWave eine neue Objekt Nummer erhalten, tritt dieses Problem nicht auf: Nachdem eine neue Version erstellt wurde, beziehen sich nach und nach alle Referenzen und Zugriffe nur noch auf sie. Sie wird daher in den Cache eingelagert und die alte Version entfernt, wenn auf sie längere Zeit nicht mehr zugegriffen wurde (ebd., 14).

Um das Problem des langsamen Zugriffes auf entfernte Dokumente zu lösen, greifen HyperWave-Browser nicht direkt auf Dokumentenserver, sondern ebenfalls über einen Cache auf sie zu. Dieser entfernt bei vollem Speicher Dokumente nach dem Least-Recently-Used (LRU)-Prinzip. Es soll sich mindestens ein Dokument-Cache in jedem LAN befinden (ebd., 13). Weil das Update-Problem nicht auftritt, können es jedoch auch beliebig viele sein (ebd., 14).

Der Cache-Server wird auch zur Protokoll- und Formatkonversion eingesetzt. Über ihn können HyperWave-Clients z.B. auf WAIS-, Gopher- und WWW-Server zugreifen, deren Dokumente dann ebenfalls zwischengespeichert werden (ebd., 14). Umgekehrt kann in dem Dialog des HyperWave-Viewers, über den ein HyperWave-Objekt angefordert werden kann, auch ein WWW-Objekt durch Angabe seines URLs ausgewählt werden (HYPERWAVE 96c). WWW-Browser können schließlich

auf ein HyperWave-Dokument zugreifen, indem aus einem HyperWave-Servernamen, einer Nummer eines Servers und einer Dokumentennummer eine URL erstellt wird.<sup>15</sup> Datenbankabfragen in HyperWave beziehen sich immer auf eine bestimmte Collection auf (gewöhnlich) einem einzigen Server, was ihre effiziente Beantwortung ermöglicht (KAPPE U. PANI o.J., 18). Die Volltextsuche wird von einer bestimmten Komponente des Link-Servers durchgeführt (KAPPE 93, 6).

Das Protokoll für die Datenübertragung zwischen Clients und Linkservern heißt in HyperWave ‚Client-Server-Protokoll‘ (CSP). Daneben gibt es noch ein Server-Server- (für das Abgleichen von Datenbanken zwischen Servern) und ein Client-Client-Protokoll (für die Kommunikation zwischen Clients über einen Server) (KAPPE U. PANI 94, 3). Das Protokoll basiert wie HTTP auf TCP/IP, ist jedoch im Gegensatz zu diesem verbindungsorientiert (ebd., 6).

In Version 7.05 des CSP gibt es 36 Kommandos. Einige dienen dem Einfügen, Holen, Verschieben, Löschen und Kopieren von ganzen Dokumenten. Andere ermöglichen die Verwaltung von Objekten, Collections und den dazugehörigen Attributen. Wieder andere CSP-Kommandos dienen der Abfrage von Anker und damit verbundenen Dokumenten, der Suche in Attributen und Text von Objekten und der Systemsteuerung. Es gibt auch solche Kommandos, die den Zugriff auf Daten von Nicht-HyperWave-Servern wie Gopher- und WWW-Servern erlauben (ebd., S. 9ff).

HyperWave benutzt ein Textformat namens ‚HTF‘, welches nicht nur dem Namen nach HTML ähnelt. HTF ist ebenso wie HTML als ein SGML-Dokumententyp definiert (KAPPE U. GAISBAUER 94, 5). Ähnlich wie HTML stellt HTF weniger ein semantisches Textformat dar, wie es dem Geist von SGML entspräche, sondern ist als typographisches Format eher auf einfache Erstellung und Darstellung ausgelegt (ebd., 7). Informationen über ein Dokument wie die Collection, in die es eingefügt werden soll, seine Sprache und die Zeit, zu der es ungültig wird, werden in HTF als SGML-Kommentare nach einer bestimmten Konvention angegeben (ebd., 8).

HTF und HTML können weitgehend automatisch ineinander überführt werden. Abgesehen von Unterschieden, die nur das Layout von Texten betreffen, ergeben sich auch welche aus der verschiedenen Behandlung von Links in HyperWave und dem WWW. Anker dürfen in HTF nicht nur geschachtelt sein, was in HTML bereits nicht möglich ist, sondern sich auch überlappen. Sie werden daher nicht über ein Anfangs- und Endelement definiert, sondern über zwei Elemente ‚<AS>‘ und ‚<AE>‘ (für ‚Ankerstart‘ und ‚-ende‘), zu denen jeweils kein Endelement existiert.

Beide Elemente besitzen einen obligatorischen Bezeichner als Attribut, über den jeweils ein Anfangs- und ein Endelement miteinander verbunden sind. Das ‚<AS>‘-Element besitzt bei Zielankern ein Attribut ‚<KEY>‘, welches als Titel des Anker dient, nach dem auch gesucht werden kann. Ein solcher Titel enthält als Präfix die Abkürzung der Sprache, in der er verfaßt ist, z.B. ‚en‘ für Englisch (ebd., 12f). Ein Ausschnitt aus einem HTF-Dokument mit Zielankern könnte also so aussehen:

```
<AS ID="0x00000001" KEY="en:TheWholeList">
<LI>
<IT>Item 1
<AS ID="0x00000002" KEY="en:Items2-4">
<IT>Item 2
<AS ID="0x00000003" KEY="en:Items3-5">
<IT>Item 3
<IT>Item 4
```

15 Dazu werden vor die hexadezimale Darstellung der beiden 32-Bit-Nummern für Server und Dokumente jeweils die Zeichen ‚0x‘ gestellt, sie werden durch einen Unterstrich zusammengestellt und hinter den WWW-Protokollbezeichner ‚http:‘ und den Namen eines beliebigen HyperWave-Server gesetzt. Aus der Id ‚82496a1f‘ für den Server und ‚00136ed2‘ für das Dokument wird also die URL ‚http://<Server-Name>/0x82496a1f\_0x00136ed2‘ (HYPERWAVE 96a).

```

<AE ID="0x00000002">
<IT>Item 5
<AE ID="0x00000003">
</LI>
<AE ID="0x00000001">

```

In dem Beispiel überlappen sich die Anker 2 und 3. (Über die Elemente ‚LI‘ und ‚IT‘ wird eine Aufzählung angegeben.)

Anker werden in HyperWave gewöhnlich interaktiv in einem Browser erstellt. Das Einfügen von Anker in HTF empfiehlt sich nur bei der Konvertierung von großen Datenbeständen. Diese Konvertierung wird dadurch erleichtert, daß Zielanker beim Einfügen eines Dokumentes noch nicht existieren müssen. Der eigentliche Link wird erst erzeugt, wenn später ein Dokument mit einem Anker mit der entsprechenden ID eingefügt wird. Anker-Elemente werden außerdem nicht im Dokumentenserver gespeichert. Wie schon beschrieben wurde, werden sie im Linkserver abgelegt und dafür beim Einbringen von Dokumenten in einen HyperWave-Server aus dem Text entfernt und bei der späteren Abfrage durch neugenerierte Elemente mit neuen IDs ersetzt (ebd., 12).

Ein spezielles HTF-Element erlaubt die Angabe von Schlüsselwörtern, die bei Einfügen des Dokumentes in den Server extrahiert und indiziert werden, zusammen mit den Wörtern des Titels (ebd., 17). Ähnlich wie in HTML gibt es in HTF ein Element, welches das Einfügen von Graphiken in verschiedenen Formaten erlaubt (ebd., 16).

Um HTF-Texte zu editieren, wird von dem jeweiligen HyperWave-Client (‚Amadeus‘ und ‚Harmony‘ genannt) ein externer Texteditor aufgerufen, in dem HTF im Quellformat geschrieben werden muß. Danach wird er auf den entsprechenden HyperWave-Server übertragen. Der Text kann vorher im Client betrachtet werden (HYPERWAVE 96b).

Bei Aufruf des UNIX-Browsers ‚Harmony‘ wird zunächst das ‚Session Manager‘-Fenster geöffnet (s. Abbildung 7, hinten links). Es stellt einen Ausschnitt der Hierarchie des HyperWave-Datenraumes graphisch als liegenden Baum dar, wie für Dateihierarchien üblich. Die Hierarchie ist in HyperWave zwar nicht der Ausgangspunkt für die Benennung von Daten, aber für ihre Bedienung (quasi umgekehrt zur Situation in Xanadu, wie wir noch sehen werden). Der Hierarchieausschnitt besteht aus Collections und der Angabe über in ihnen enthaltene weitere Collections und Dokumente. Wenn vorhanden, wird beim Öffnen gewisser Collections automatisch ein spezielles ‚Head-Document‘ geöffnet, sonst werden Dokumente nur nach Doppelklick angezeigt (MAYRHOFER U. ANDREWS 96, 13).

Neben der Liste der zuletzt besuchten Dokumente, die man sich auch in gängigen WWW-Browsern anzeigen lassen kann, bietet Harmony weitere Orientierungshilfen, die zusätzliche Darstellungen der gerade betrachteten Dokumente erlauben (ebd., 26). Die ‚Local Map‘ (s. Abbildung 7, vorne) zeigt alle Dokumente, die auf das gerade betrachtete Dokument mit einem Link verweisen oder auf die das aktuelle Dokumente verweist.<sup>16</sup> Diese Darstellung ist von der ‚Local Tracking Map‘ des Hypertextsystems Intermedia inspiriert (s. YANKELOVICH u.a. 88, 35). In der Local Map von HyperWave kann auch ein Dokument angefordert werden. Links können nach Typ (Referenzlinks, Anmerkungen, Bilder, über- oder untergeordnete Elemente) und Tiefe eingestellt werden (ebd., 27f). Die Übersichtsdarstellung ‚Information Landscape‘ erweitert die ‚Local Map‘ zu einer dreidimensionalen Sicht, über die mit Hilfe von Mausektionen ‚geflogen‘ werden kann. Angezeigte Objekte geben durch verschiedene Größen bzw. Farben über ihren Umfang bzw. ihre Art Auskunft (ebd., 29f).

<sup>16</sup> Schon beim Betrachten eines Literaturverzeichnisses eines fremden Dokumentes scheint es oft interessant, die Stelle zu finden, an denen ein bestimmtes Dokument zitiert wurde. Dies ist bei Büchern höchstens über einen Personenindex möglich.

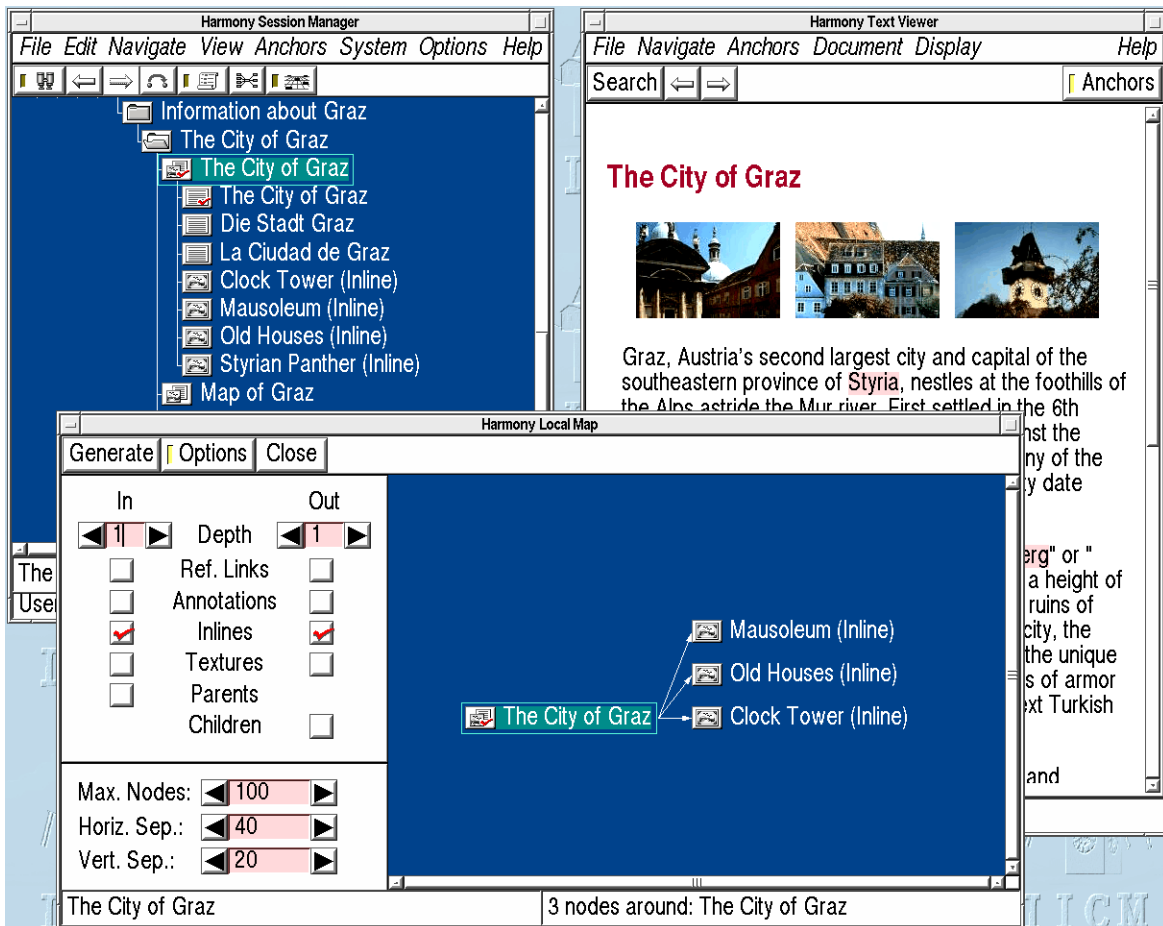


Abb. 7. HyperWave Browser Harmony unter UNIX mit (obligatorischem) Verzeichnisfenster (hinten links), Textbetrachter (hinten rechts) und Linkkontextfenster (vorne).

Dokumente verschiedenen Typs wie HTF-, Graphik- (GIF-, JPEG-, TIFF- und PNG-), Film- (MPEG-), VRML-, PostScript- und Tonsequenzdateien werden jeweils durch eigene Programme dargestellt bzw. abgespielt, die speziell für HyperWave entwickelt oder angepaßt wurden (ebd., 32ff). In allen optischen Formaten können interaktiv Anker angebracht werden, die, wie schon beschrieben, extern zu den eigentlichen Daten gehalten werden (ebd., 58ff).

In HyperWave können beliebige Objekte annotiert werden. Anmerkungen können aus Text, einem Bild, einem Film, einer Audiosequenz, einer dreidimensionalen Szene oder einem PostScript-Dokument bestehen. Sie sind immer einer Collection zugeordnet (ebd., 51ff). Die Anmerkung stellt eine der drei möglichen Typen eines Ankers dar. (Die anderen beiden sind ‚Include‘ für die Einbindung von Bildern und ‚Reference‘ für Verweise, s. KAPPE 94).

HyperWave wurde in C++ unter UNIX (und die Browser) mit dem Fenstersystem X implementiert (KAPPE 91, 19). Auf gängigen Workstations sind mindestens 64 MB RAM nötig, um den HyperWave-Server einzusetzen, auf Intel-PCs mit Linux oder FreeBSD nur 32 MB (HYPERWAVE 96).

### 3.4 Xanadu

Xanadu ist der Name der Vision TED NELSONS für ein weltweites Hypertextsystem, welches die heutige Art der Erzeugung, Verbreitung und Nutzung von Literatur ersetzen soll. Da Xanadu eine lange Geschichte aufweist und sich der mit dem Namen verbundene Inhalt ständig gewandelt hat,

sollen zunächst seine Vorgänger zusammenfassend vorgestellt werden. Danach wird die letzte dokumentierte Version Xanadus ausführlich dargestellt. Es wird außerdem kurz die Frage diskutiert, warum Xanadu bisher nicht umgesetzt worden ist.

### 3.4.1 Geschichte von Xanadu

Die schon zitierte erste Definition von ‚Hypertext‘ hat NELSON 1965 auf einer Konferenz der ACM vorgestellt. Er hatte 1960 in Harvard zum ersten Mal versucht, ein Textprogramm zu schreiben.<sup>17</sup> Fünf Jahre später war daraus ein Entwurf zu einem System geworden, welches dem Verwalten einer persönlichen Ablage (als „dream file“) und dem Erstellen von Texten dienen sollte, was sich nach NELSONS Philosophie nicht scharf trennen läßt (65, 85).

Obwohl an dem System nur jeweils eine Benutzerin gleichzeitig arbeiten konnte, sollte seine Benutzung in Bereichen wie „information retrieval and library science, motion pictures and the programming craft“ (ebd., 84) nützlich sein. NELSON hat also in seinem ersten System die Anfänge der Vision vom großen Hypertext gelegt, die er später ‚Xanadu‘ nennt. Das Xanadu-System wird weiter unten vorgestellt. Er nannte das erste System von 1965 „Evolutionary List File“, ELF (ebd., 89). Implementiert wurde es erst 1993 in Japan unter NELSONS Mithilfe auf Basis des WWW (s. S. 57). Zunächst wird hier das System von 1965 beschrieben, weil es den ersten veröffentlichten Entwurf NELSONS für ein Hypertextsystem darstellt.

Daten sollten in der Ablage in Elementen abgespeichert werden. Ein Element sollte aus einem Text beliebiger Länge, einem Bild oder einem Namen für ein physikalisches Objekt bestehen (ebd., 89f). Elemente sollten zu geordneten Listen zusammengefaßt werden können. Ein Element sollte sich dabei in einer beliebigen Anzahl von Listen befinden können, weil diese es nicht physikalisch enthalten, sondern nur auf es verweisen sollten. Die Reihenfolge einer Liste sollte geändert werden können. Listen sollten kopiert, auseinandergeschnitten und kombiniert werden können (ebd.).

Zwischen genau zwei Elementen aus zwei verschiedenen Listen sollte ein Link gesetzt werden können. Ein Link sollte also nur zwei Enden haben (ebd.). Die beiden betroffenen Listen sollten auch verschieden lang sein können. Die Anzahl der möglichen Links sollte sich dann nach der Länge der größeren Liste richten. Da sich die Links direkt auf die Elemente und nicht auf ihre Position in den Listen beziehen sollten, sollte die Reihenfolgen der Listen geändert werden können, ohne daß die Links davon betroffen wären (ebd.).

Listen, zwischen denen Links bestanden, wurden ‚Zippered Lists‘ genannt (später ‚Zipper Lists‘ oder ‚Zips‘, s. OOKUBO U. NELSON o.J., Abschn. „The HyperLab ZipEditor“).<sup>18</sup> Zwei Listen nennt NELSON ‚zippered‘, wenn sie aus Elementen bestehen, zwischen denen es paarweise Links gibt, wobei jedes Element nur in einem solchen Paar enthalten sein kann und die Links unabhängig von der Reihenfolge der Listen sind (ebd.). Listen sollten als Kategorien für Notizen, Textentwürfe, Gliederungen und Indizes genutzt werden (ebd., 89f).

In Abbildung 8 sind verbundene ‚Zippered Lists‘ dargestellt. Varianten und Teile eines Dokumentes sind über Links verbunden. Es gibt z.B. jeweils einen Link von einem Element der Outline zu einem anderen des Main Draft. Links sind hier über gestrichelte Linien dargestellt, durchgezogene Linien verbinden dieselben Elemente in verschiedenen Dokumenten. Nur Links werden als physische Objekte im System gespeichert. Dokumente sind indirekt über Elemente verbunden, die sie gemeinsam einbinden, es gibt jedoch keine Links von den Elementen zu den Dokumenten.

Links sollten von zwei Seiten her ausgewertet werden können, es sollte also in unserem Beispiel die jeweilige Gliederungsüberschrift zu einem Teil des Hauptdokuments oder der Abschnitt des

17 Dieser Versuch hatte im Rahmen einer Programmier Einführung statt gefunden und ihm gleich die erste Niederlage auf dem Feld der Informatik beschert: Weil er das System nicht fertigbekommen hatte, hatte er auch keinen Leistungsnachweis für diesen Kurs erhalten (NELSON 93, 1/26).

18 nach dem englischen Wort für Reißverschluß, ‚zip(per)‘.

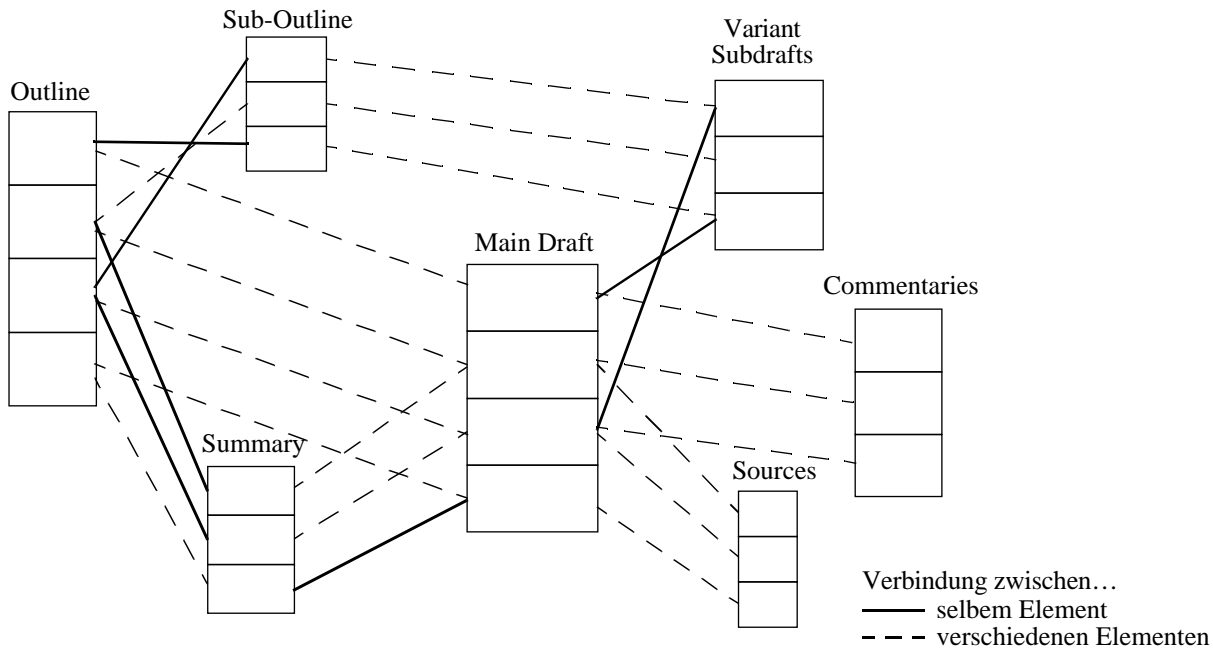


Abb. 8. Verbindung zwischen Dokumenten über ‚Zippered Lists‘ ( NELSON 65, 99)

Hauptdokumentes zu einer Überschrift gefunden werden können (ebd., 90). Wenn aus einem Eintrag oder einer Liste eine neue Version erzeugt wurde, sollte diese aus Verweisen auf das jeweilige Original und Änderungsdaten zu ihm bestehen, so daß sie relativ wenig Platz beansprucht (ebd., 92).

Das System sollte zwar die Bearbeitung von Gliederungen ermöglichen (Outline processing), aber nicht ausschließlich, was NELSONS Abneigung gegen Hierarchien entspricht. In seinem System sollte ein Element in einem Zusammenhang eine Trivialität, in einem anderen (einer anderen Folge) eine Überschrift bedeuten (93, 1/28). Später erweitert NELSON ‚Zippered lists‘ zu ‚Kollateralen Strukturen‘ und gibt an, daß Engelbart sie unabhängig von ihm entdeckt hat (87, *Dream Machines*, 50).

Das ELF sollte dem Verarbeiten ähnlicher Texte dienen, z.B. verschiedener Textversionen (NELSON 65, 92). Analog zu BUSHS ‚Trails‘ (s. NELSON 91, 103f) sollte das System die Erstellung von Pfaden in Briefen, Gerichtsurteilen und Verträgen ermöglichen (NELSON 65, ebd.). Generell sollten die Datenstrukturen der Ablage dem Dokumentieren dienen und selbst wieder dokumentiert werden können, z.B. als private und öffentliche Randbemerkungen (wobei Benutzerinnen nacheinander an dem Einzelplatzsystem arbeiten würden) und selbstdokumentierende Programme. Für letztere sollten Programmcode- mit Texteinträgen als Kommentare verbunden werden. Zusätzlich sollten mehrere Versionen eines Programmes verwaltet werden können, die über Listen beschrieben und miteinander verbunden sein sollten (ebd., 93f).

Auf das System sollte sowohl über ein Batchsystem, ein Drucker- und ein Bildschirmterminal zugegriffen werden können. Für letzteres stellte sich NELSON eine schnelle Suchfunktion und Bildschirmmenüs vor, hatte die genaue Bedienung jedoch nicht festgelegt (ebd., 95). Das System sollte außerdem programmierbar sein, um Kommandos gemeinsam auszuführen oder Daten wie Textstrings oder Zahlen zu verarbeiten (ebd.). Es sollte also eine Makrosprache geben, wie sie heute genannt werden würde.

Nach Meinung seines Entwicklers ist die Datenstruktur der über Links verbindbaren Listen wichtiger als die Festlegung einer Benutzungsoberfläche für sie. Er stellt sich vor, daß seine Daten-

strukturen in ein Betriebssystem eingebaut werden könnten (eigentlich „into the file capabilities of general utility software“) (ebd., 95f).

NELSON schreibt über die Benutzbarkeit nur allgemein, daß mächtige und vielseitige Programme, wie es das zur Bedienung der ELF sein sollte, oft schwer zu verstehen und zu bedienen sind. Sein System sollte jedoch v.a. von Computerlaien benutzbar sein. Die Listen sollten daher gleichzeitig einfach und mächtig sein (ebd., 89). Die folgende Aussage könnte er auch über Xanadu gemacht haben:

The original idea was to make a file for writers and scientists, much like the personal side of Bush's memex. ... These uses and considerations became so complex that the only answer is a simple and generalized building-block structure, user-oriented and wholly general-purpose. (ebd., 84)

Im Jahr 1969 hielt sich NELSON auf eigene Kosten an der Brown-Universität auf. Er entwickelte dort das ‚Hypertext Editing System‘ (HES) mit (unter der Leitung von ANDRIES VAN DAM), durch das einige seiner Hypertextideen ausprobiert werden sollten (NELSON 93, 1/31f, s. auch VAN DAM 88, 889). Daneben sollte es der einfachen und effizienten Erstellung von gedruckten Dokumenten dienen. Das System wurde später von der NASA für die Dokumentation der Apollo-Missionen benutzt (VAN DAM 88, ebd.).

Wie das System von 1965 war das HES für jeweils eine Benutzerin ausgelegt (CARMODY et al. 69, 291), die eine Schreiberin, Redakteurin oder Online-Leserin sein sollte (ebd., 294). Sie konnte ‚reisen‘ (travel), also zwischen Textstücken springen, wobei ein anderer Text den sichtbaren ersetzte. Außerdem konnte sie Text und Hypertextelemente erzeugen sowie ändern und Text für die Bildschirm- und Druckerausgabe formatieren (ebd., 305). Neben der Tastatur konnte das HES auch über einen Lichtgriffel bedient werden (ebd., 293).

Text wurde in Einheiten unterteilt, die durch Links verbunden waren (ebd., 292). Diese Einheiten wurden ‚Flächen‘ (Areas) genannt. Ein einfacher Text bestand nur aus einer Fläche (ebd., 302). Daneben gab es Links. Sie verwiesen von einer Stelle in einer Fläche zu einer anderen Stelle in derselben oder einer anderen Fläche. Links bildeten alternative Pfade (ebd., 302f) oder Sprungmöglichkeiten in einem sonst linearen Text (ebd., 309). Sie konnten wie Fußnoten und Querverweise eingesetzt werden, z.B. zu Definitionen und detailreicheren Darstellungen führen. In einem Hypertext erlaubten sie Sprünge zwischen verschiedenen Flächen (ebd., 309).

Neben Links gab es Verzweigungen (Branches) und ein Menü von Sprungzielen, welches sich am Ende einer Fläche befinden konnte (ebd., 302f). Verzweigungen erlaubten Multiple-Choice-Fragen, das Gliedern von Material und konnten ein Inhaltsverzeichnis bilden (ebd., 309f). Texte, Links und Verzweigungen konnten annotiert werden (ebd., 304). Es gab einen Befehl, der eine Zeichenkette von einem an einen anderen Ort kopierte, ähnlich ‚Copy and Paste‘, der aber zusätzlich am Ursprungs- und am Zielort einen Referenznamen einfügte (ebd., 319f), wodurch der Herkunftsort eines einkopierten Textes aufgerufen werden konnte (ebd., 303f).<sup>19</sup> Im HES konnten Überschriften zwar numeriert werden, es wurde aber keine hierarchische Struktur vorgegeben (ebd., 301).

Seit 1967 nennt NELSON die Idee seines Hypertextsystemes ‚Xanadu‘. Er hat den Namen wegen seiner literarischen Bedeutung gewählt: Anstatt des mystischen Palastes im Gedicht ‚Kubla Khan‘ von Coleridge sollte er den „magischen Platz des literarischen Gedächtnisses“ benennen (93, 1/30).<sup>20</sup> Der Name ist entstanden, während NELSON für den Verlag Harcourt Brace gearbeitet hat.

In den Jahren 1971 und 1972 erforschte NELSON Datenstrukturen für die schnelle Manipulation von Text. Zusammen mit anderen hat er 1972 die erste ‚Enfilade‘ entwickelt, eine Datenstruktur zur Verwaltung von großen Textmengen, die in späteren Versionen von Xanadu immer noch Verwendung findet. Sie ermöglicht die schnelle Änderung sequentieller Daten (ebd., 1/32f). Im September

<sup>19</sup> Dieses Konzept kann als Vorläufer der Transklusion (s. S. 71) gesehen werden.

1973 gab es eine lauffähige Version (ebd., 1/33). Ein Freund von NELSON hat die Enfilade 1974/75 überarbeitet und dahingehend verbessert, daß Links bei Änderungen des Textes, auf den sie sich beziehen, erhalten blieben (ebd., 1/34). Dieser neue Entwurf wurde im Sommer 1979 erneut überarbeitet und um eine Enfilade für die Verwaltung von Versionen und eine ‚allgemeine Enfiladentheorie‘ ergänzt, zu der NELSON keine weiteren Angaben macht. Durch die verschiedenen Enfiladen sollte das System ein Wachstum ohne Geschwindigkeitseinbußen erlauben (ebd., 1/36). NELSON hat von Ende 1981 bis 1984 für die Firma Datapoint<sup>21</sup> gearbeitet, zunächst als Softwareentwickler, später als technischer Schreiber (weil seine Ideen „zu futuristisch“ waren; NELSON 87, *Computer Lib*, 16). Er und andere aus dem Xanadu-Team haben bei Datapoint an einem Hypertextsystem gearbeitet, welches die Idee der ‚Zippered lists‘ (s. S. 30) umsetzen sollte, aber nie verwirklicht wurde.

Der Entwurf für Xanadu Version 1988, wie sie später nach dem Fertigstellungsjahr der Implementierung genannt wurde, war 1981 abgeschlossen. Die Implementierung stellte sich als schwieriger heraus als erwartet. Die Beschreibung des Systems in *Literary Machines 87.1* hat NELSON 1987 verfaßt (NELSON 93, Vorwort).<sup>22</sup> Auf der ‚Hacker’s Conference‘ im Oktober 1987 haben sich JOHN WALKER, der Direktor von Autodesk, und ROGER GREGORY, der damalige Direktor der Xanadu Operating Company (XOC), getroffen. Dieser Kontakt hat schließlich dazu geführt, daß Autodesk 1988 achtzig Prozent von Xanadu übernommen und die weitere Entwicklung des Systems finanziert hat. NELSON wurde ‚Distinguished Fellow‘ bei Autodesk (WALKER 88a, 416). Ein fertiges System sollte vor Ende 1989 auf den Markt kommen (WALKER 88b, S. 425f).<sup>23</sup>

Xanadu Version 1988 war, wie schon erwähnt, in dem entsprechenden Jahr fertiggestellt worden. Es stellte sich jedoch, so NELSON, als nicht mächtig genug heraus für ein weltweites Publikationssystem, außerdem als nicht robust genug. Die Xanadugruppe wollte jedoch nicht gleichzeitig diese Version warten und eine neue entwerfen und implementieren. Also wurde Version 1988 als Prototyp betrachtet und der Entwurf einer neuen Version begonnen (93, Vorwort).

Diese Entscheidung hat sich später als Fehler herausgestellt. Als Autodesk 1992 in finanzielle Schwierigkeiten geriet, waren nur der Entwurf und ein Prototyp des neuen Systems fertig. In *Literary Machines 93.1* nennt NELSON diese Version ‚Xanadu 1993‘, weil sie in diesem Jahr auf den Markt kommen sollte. Xanadu 1993 sollte ein „connection-and-history server“ sein, der in einem Netzwerk (WAN) arbeitet und Client-Programme auf anderen Computern mit Daten und Linkinformationen versorgt. Er sollte robuster als die Vorgängerversion sein, war intern anders strukturiert, hatte ein leicht geändertes Protokoll und neue Fähigkeiten. Die Internas sind nicht veröffentlicht. (Auch in *Literary Machines 93.1* wird Version 1988 beschrieben.) Der Server wurde auf einer Sun-

20 Für NELSONS Arbeit ist auch die folgende Bemerkung interessant: „... the name ‘Xanadu’ has both a magic and a curse; in the film ‘Citizen Kane’ it referred to the never-finished palace of the protagonist” (NELSON 93, Vorwort). BARRET schreibt:

The Coleridgian reference in the naming of his system is no mere grace note. Nelson’s conception of hypertext is highly Romantic: the computer tracks the mind as it interlinks seemingly disparate text objects and facilitates that motion of mind through highly developed annotation features; programming, in other words, permits the sculpting of imaginative insights. (91a, xii)

21 Diese war durchaus innovativ: “Datapoint’s ARCnet with personal computers with the Intel 8080 was clearly the first commercial workstation environment” (BELL 88, 17). Die Firma verschwand Mitte der achtziger Jahre.

22 SCHNUPP liegt also falsch, wenn er über Xanadu schreibt: „1991 war bereits die Versionsnummer 87.1 (!) erreicht“ (92, 121, Ausrufezeichen im Original).

23 Tatsächlich berichtet PAM von einem internen Autodesk-Papier vom 6.10.88 mit dem Titel „Sunrise on Xanadu“, in dem WALKER ein (funktionierendes, jedoch unvollständiges) graphisches Frontend für den Xanadu-Server von 1988 beschreibt. Es lief auf einer Sun-Workstation unter einem Fenstersystem und unterstützte Rasterbilder und AutoCAD-Datenbanken (PAM 96b).



Workstation entwickelt und sollte auch auf 386er-PCs und Apple Macintoshs laufen (ebd., Vorwort). Ob er es je getan hat, ist nicht veröffentlicht.

Das Warenzeichen ‚Xanadu‘ wurde 1992 an NELSON zurückgegeben. Die Weiterentwicklung des XOC-Servers wurde der Firma Memex überlassen (die sich 1994 in ‚Filoli‘ umbenannt hat). 1993 hat NELSON eine Reihe von Geschäftsvereinbarungen festgelegt, die unter dem Namen ‚Xanadu‘ firmieren (PAM 96c, Abschn. 4). Xanadu Version 1993 ist immer noch nicht auf dem Markt. NELSON schreibt jedoch in einer E-Mail vom 26.11.96 an die ‚Xanadu‘-Mailing-Liste, daß dieses System jetzt soweit implementiert ist, daß es öffentlich getestet werden kann. ROGER GREGORY, der immer noch dafür verantwortlich ist, sucht finanzielle Unterstützung, um es „ins Web zu bringen“.<sup>24</sup>

NELSON arbeitet inzwischen in Japan. Am HyperLab in Sapporo wurde seine Idee der ‚Zippered Lists‘ (s. S. 53) mehr als dreißig Jahre nach ihrer Formulierung in ein Programm umgesetzt, allerdings in eines, welches im Gegensatz zum alten Entwurf den Mehrbenutzerinnenbetrieb über mehrere Server erlaubt (OOKUBO U. NELSON o.J., Abschn. „The HyperLab ZipEditor“). Im ZipEditor ist es möglich, verbundene Elemente nebeneinander zu betrachten und zu bearbeiten (s. Abbildung 9). (Da es sich bei den Elementen um kleine Abschnitte handelt, können ihre Fenster so schmal eingestellt werden, daß zwei nebeneinanderpassen.) Wenn ein Element geändert wird, wird es durch ein neues ersetzt, wobei das alte erhalten bleibt und das neue über einen Link mit ihm verbunden wird. Gleiche (transkludierte) und voneinander abgeleitete Elemente werden durch verschieden dicke Linien verbunden (ebd.).

Alle Zips auf einem bestimmten Server werden in demselben Datenbereich verwaltet. Mit dem ZipEditor können Zips und Elemente von verschiedenen Servern gleichzeitig betrachtet und verbunden werden (ebd., Abschn. „User Interfaces“). Auf Listen und Elemente kann lesend auch über das WWW zugegriffen werden. Dabei kann nicht nur zwischen demselben Element in verschiedenen Listen oder entlang von Versionsketten gesprungen, sondern auch die jeweilige Autorinnen-schaft abgefragt werden. WWW-Browser greifen über einen WWW-Server per CGI auf den sogenannten ZipFileManager zu. Dieser erstellt HTML-Seiten aus seinen Daten, in die Links eingebettet sind (ebd.).

Daten werden im ZipFileServer verwaltet. ZipEditoren kommunizieren mit ihm durch ein eigenes Protokoll. Listen werden als Folge von Zeigern auf Elemente verwaltet. Eine Liste enthält neben dem Namen ihrer Autorin, dem Erstellungsdatum und einer Internetadresse die Folge ihrer Elemente und möglicherweise einen Verweis auf die Liste, von der sie abgeleitet wurde. Transklusion liegt vor, wenn zwei verschiedene Listen Verweise auf dasselbe Element enthalten (s. S. 71). Sie wird vom Server erkannt und vom Editor entsprechend dargestellt.

Wenn eine Autorin eine Liste kopiert, enthält diese dieselben Zeiger auf Elemente wie die ursprüngliche Liste, jedoch mit dem Namen der neuen Autorin und einem Verweis auf die ursprüngliche Version. Es werden jedoch keine Änderungsdaten gespeichert. Wenn also ein Element in einer Liste ersetzt wird, wird die Liste direkt geändert, ohne daß diese Änderung rückgängig gemacht werden kann. Jedes Element enthält ebenfalls Autorin, Erstellungsdatum, Internetadresse und den eigentlichen Text, außerdem Verweise auf die Liste, in der es erstellt wurde und auf das Element, von dem es abgeleitet ist (ebd., Abschn. „The Implementation“).

### 3.4.2 Xanadu Version 1988

Xanadu sollte nie ein abstraktes Konzept sein, wie z.B. das Dexter Hypertext Model (s. HALASZ U. SCHWARTZ 94), sondern immer praktisch umgesetzt werden. In diesem Abschnitt wird Xanadu in der Version von 1988 vorgestellt, weil dies die letzte dokumentierte ist. Die Vorstellung wird in

24 NELSON selbst nennt sich in der zitierten E-Mail übrigens „Professor for Environmental Information“.

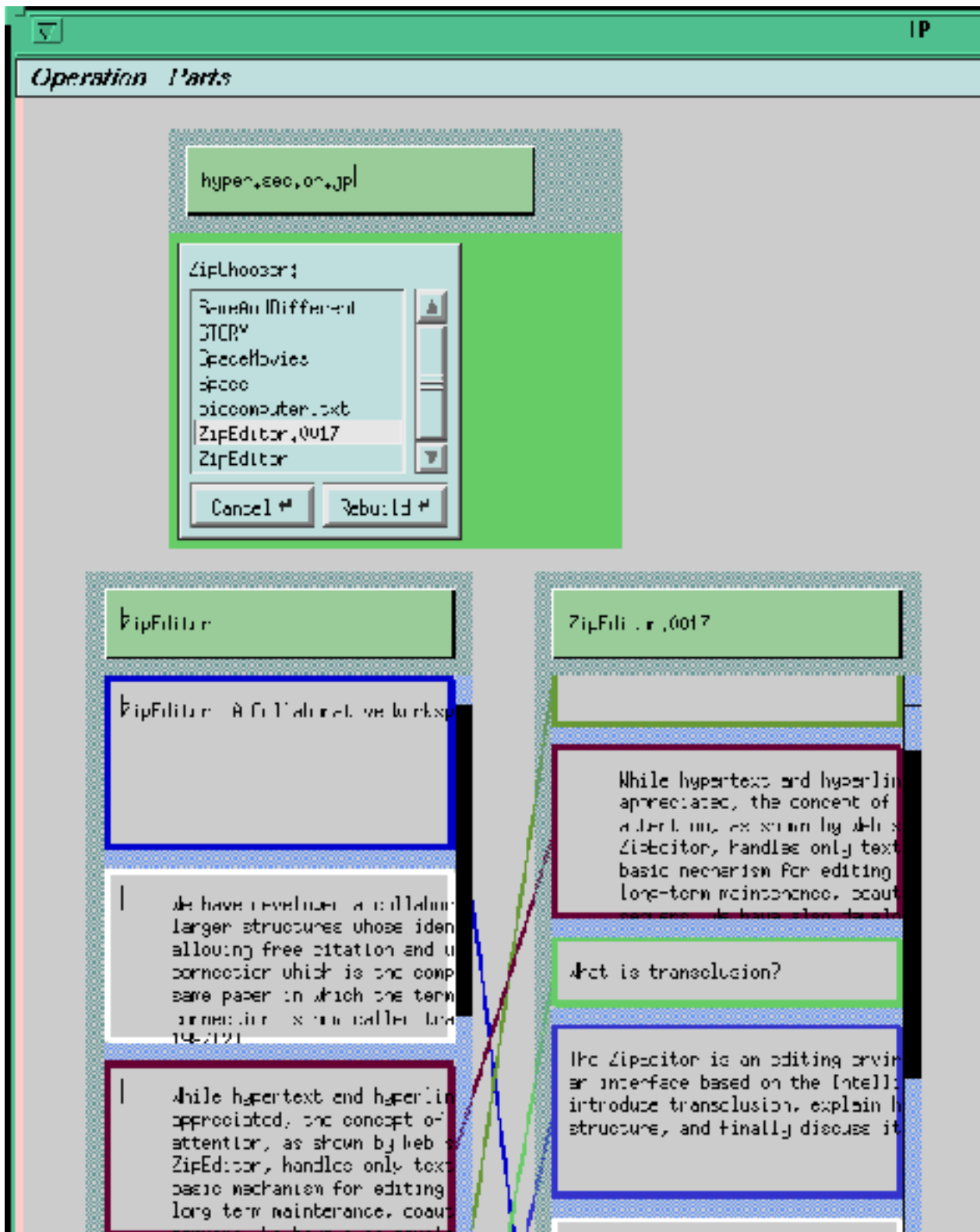


Abb. 9. Screenshot des ZipEditors (OOKUBO U. NELSON o.J., Qualität im Original)

Formulierungen erfolgen, die die Möglichkeit einer Realisierung von Xanadu zumindest nicht ausschließen.

Wie noch deutlich werden wird, ist es jedoch nicht bei allen von NELSON geschilderten Funktionen klar, wie sie umgesetzt werden sollen, weil die Beschreibung auf technische Einzelheiten nur selten eingeht. Bei manchen Funktionen ist es sogar unwahrscheinlich, daß sie überhaupt so implementiert werden können, daß sie in einem großen, weltweiten Hypertextsystem in benutzbarer Geschwindigkeit ablaufen. Dabei hatten die Xanadu-Entwicklerinnen die Version 1988 aufgegeben und den Entwurf einer neuen begonnen, gerade weil die alte Version nicht skalierbar war (s. S. 56). Die folgende Beschreibung von Xanadu ist immer wieder durch Kritik unterbrochen, wobei teilweise auf die beiden aktuellen Hypertextsysteme HyperWave und WWW Bezug genommen wird, die bereits beschrieben wurden.

### 3.4.2.1 Anforderungen

Die umfangreichste Dokumentation von Xanadu findet sich in dem Buch *Literary Machines* (NELSON 93). NELSON schwebt also vor, in seinem Hypertextsystem ‚Literatur‘ als Ganzes abzubilden und zu verbessern (93, 2/8). Er definiert „a literature“ sehr weit als „a system of interconnected writings“, welche damit z.B. auch Briefe, Bücher und Rechnungen umfaßt. Literatur enthält Verbindungen, die über Zitate, Literatur- und andere Verweise hergestellt werden (ebd., 2/9ff). Die heutige Literatur ist ausgereift, auch wenn wir ihre Struktur nicht vollständig verstehen können (ebd., 2/12), „LITERATURE IS DEBUGGED“ (NELSON 80, 1015). Sie ist außerdem in der Lage, sich Inhalten und Sichtweisen in verschiedenen Feldern anzupassen (ebd., 1015). Diese Eigenschaften klassischer Literatur soll auch die in Xanadu umgesetzte behalten.

Nach NELSON teilt die Informatik die Bearbeitung von Text künstlich auf in ‚Textverarbeitung‘, ‚E-Mail‘ und ‚Information Retrieval‘ (80, 1022). Xanadu soll die ersten beiden Anwendungen enthalten, zusätzlich Gliederungsverarbeitung, das Abhalten von Telekonferenzen, Elektronisches Publizieren und Archivieren (NELSON 93, 2/50f).<sup>25</sup> Das Bearbeiten von Text steht jedoch im Vordergrund.

Um Anforderungen an sein System formulieren zu können, spezifiziert NELSON zunächst den Prozeß des Schreibens als das Zusammenstellen und mehrmalige Überarbeiten von Textfragmenten. Diese Fragmente werden versuchsweise geändert, umgestellt, kopiert und angepaßt, wobei viele sich untereinander kaum unterscheiden, also Versionen voneinander darstellen (65, 87). An anderer Stelle schreibt er: „Writing is the process of making the tree of thought into a picket fence“ (NELSON 91, 254).

Klassische Textverarbeitungsprogramme sind dabei kaum eine Hilfe, weil sie nur das Arbeiten mit Textstücken erlauben, die schon organisiert sind. Gerade dieses Organisieren ist jedoch seiner Meinung nach der schwierigste Teil des Schreibens, wie NELSON schon 1965 darlegt. Ein gutes System sollte dabei helfen, in dem es das Auffinden, Aneinanderreihen, Nebeneinanderstellen und Vergleichen von Textstücken erlaubt (87f).

Von ‚What-You-See-Is-What-You-Get‘ (WYSIWYG) als Anforderung an Textverarbeitungssysteme hält NELSON nicht viel. Für ihn steht dahinter die Forderung, daß es auf jedes Objekt im Computer nur eine Sicht geben sollte. Er jedoch verlangt: „We need more ways of viewing *anything*: expanded, shrunken, sideways, showing only highlights, showing contents without highlights, showing links, animated“ (87, *Dream Machines*, 73).<sup>26</sup>

25 Xanadu ist nach NELSON „Literary System, Storage Engine, Hypertext and Hypermedia Server, Virtual Document Coordinator, Write-Once Network Storage Manager, Electronic Publishing Method, Open Hypermedium, Non-hierarchical Filing System, Linked All-Media Repository Archive, Paperless Publishing Medium, and Readdressing Software. The Magic Place of Literary Memory“ (87, *Dream Machines*, 141).

26 Wie schon erwähnt, gibt es in gängigen, papierzentrierten WYSIWYG-Textverarbeitungsprogrammen wie Word von Microsoft durchaus mehrere Ansichten auf ein Dokument, z. B. eine Gliederungssicht oder eine schnelle Arbeitsansicht ohne Papierrändersimulation.

Schon NELSONS erstes System von 1965 sollte als persönliche Ablage dienen. Für eine solche hat er gefordert, daß sie die beliebige Anordnung von abgelegtem Material erlauben soll, die vollständige Änderbarkeit, die Behandlung von Alternativen und die vollständige Dokumentation im System selbst (65, 84). Die Möglichkeit einer hierarchischen Ordnung gehört jedoch nicht dazu. Bereits BUSH hat sie in seinem Artikel, der die Idee des Hypertext zum ersten Mal ausgesprochen hat (91), abgelehnt:

Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome. Having found one item, moreover, one has to emerge from the system and re-enter on a new path. (101)

Nach CONKLIN kommen Hierarchien zwar der Fähigkeit der menschlichen Erkenntnis nach Abstraktion entgegen. Sie werden jedoch jeweils anhand der zu einer bestimmten Zeit bekannten Kriterien erstellt, die nach Dazukommen neuen Materials nicht ausreichen könnten (87, 35). CONKLIN vermutet Unterschiede in der Weltanschauung der Benutzerinnen als Gründe für oder gegen Hierarchien (87, 35).<sup>27</sup> Es muß also Benutzerinnen überlassen bleiben, ob sie Hierarchien anlegen möchten oder nicht. Das von HALASZ so genannte, bereits erwähnte „Problem der vorschnellen Organisation“ (s. S. 23) tritt bereits bei der Ablage von Material auf.

In Xanadu werden keine festen Kategorien oder Hierarchien für die Ordnung von Material festgelegt, verschiedene Kategorien müssen koexistieren und es müssen (für bereits eingeordnetes Material) neue hinzukommen können (NELSON 93, 1/24). NELSON spricht daher von „kumulativer Ordnung“, bei der nach dem Hinzukommen von neuem Material neue Kategorien erzeugt werden (ebd., 0/7f). Es gibt in Xanadu nur zwei Verzeichnisse, die vom System selbst verwaltet werden, eines von Autorinnennamen und eines von Dokumententiteln.<sup>28</sup> Da sie also formal/mechanisch ordnen (s. S. 15), eignen sie sich für die Ordnung aller Dokumente. Außer den Systemverzeichnissen können von Benutzerinnen über Links beliebige eigene Verzeichnisse erzeugt werden, für die, wie für andere Dokumente auch, Tantiemen kassiert werden (s. weiter unten) (ebd., 2/49f).

Xanadu ist keine Ablage im klassischen Sinn, sondern ein ‚Hyperfile‘. Als Hyperfile bezeichnet NELSON ein Computersystem, welches Daten in einzelnen Blöcken abspeichert, sie je nach verlangter Version auf Anfrage zusammensetzt und Vergleiche zwischen Versionen erlaubt, indem es gleiche Teile anzeigt. Ein solches System verwaltet also Links, historische und alternative Versionen (ebd., 2/21f). Es verwaltet Daten jedoch nicht in Dateien, weil diese in großer Zahl unübersichtlich sind und jeweils einen eigenen Namen benötigen (NELSON 87, *Computer Lib*, 102). Allerdings werden Daten in Xanadu in Dokumenten organisiert, die jeweils eine Nummer bekommen und daher als Dateien angesehen werden können. Dokumente und -nummern werden weiter unten beschrieben.

Beim Schreiben sind Versionen nützlich, weil sie zurückgeholt werden können, um Gedanken klarzustellen, neu anzufangen oder alte Ideen einzuarbeiten, wie wir im ersten Kapitel gesehen haben. Bestimmte Versionen können als Ausgangspunkt für weitere dienen (NELSON 93, 2/18). BRÜGGEMANN-KLEIN gibt Beispiele von sich schnell ändernden Dokumenten:

Studienordnungen und Systemdokumentationen befinden sich in beständigem Wandel, Aufgabenblätter werden, um Musterlösungen ergänzt, zu Aufgabensammlungen zusammenge-

27 Er schreibt: “As a generalization, it seems that engineering-oriented hypertext users prefer hierarchical organizations, whereas arts- or humanities-oriented users prefer cross-referencing organizations” (87, 35).

28 Diese beiden systemweiten Verzeichnisse sind u.a. nötig, damit ein Dokument gefunden werden kann, wenn noch kein Link auf es existiert.

stellt, Skripten werden zu Büchern ausgearbeitet, und aus internen Berichten werden Kurzfassungen für Tagungen extrahiert. (89, 151)

Das Aufbewahren von alten Versionen ist in bestehenden Computersystemen nur eine der beiden Funktionen vom Aufbewahren von Dateikopien. Die andere Funktion ist die, daß man in einem Schadensfall auf eine Kopie zurückgreifen kann (Backup). Diese Sicherheitskopien werden von Xanadu automatisch erstellt, ohne daß die Benutzerin davon belastet wird. Das gleiche gilt für das Verwalten von Versionen (NELSON 93, 2/13ff).

Wie schon erwähnt, ist Xanadu im Gegensatz zum ersten System NELSONS nicht nur eine Ablage, sondern auch ein Publikationssystem. Den Schritt zu einem solchen System sieht NELSON schon bei BUSH in seinem Memex angelegt, und er schreibt:

Having created for personal use a hyper-document on one's console, it will seem only natural to press a button passing this on to a colleague in its hyper-form, without chopping and aligning it into conventional writing. (91, 257)

An ein Publikationssystem sind größere Anforderungen zu stellen als an ein Ablagesystem, die außerdem schwerer zu erfüllen sind. NELSON nennt die Aspekte Privatheit, Urheberrechte, Tantiemen und Abrechnung, die ein Entwurf für ein Publikationssystem ansprechen muß (93, 2/41). Er ist gegen die Trennung in ein Autorinnen- und ein Leserinnensystem. Gerade in der Möglichkeit, beides zu sein, sieht NELSON das egalitäre Potential eines Hypertextsystemes (87, *Dream Machines*, 31). So können Benutzerinnen, die ursprünglich das System nur zum Lesen und als persönliche Ablage benutzt haben, in die Rolle von Autorinnen hineinwachsen.

Neben dem schon erwähnten Aspekt, daß gerade ein mächtiges System einfach zu bedienen sein muß (s. S. 30),<sup>29</sup> nennt NELSON als weiteren für die Entwicklung eines Hypertextnetzwerkes, daß es schnell genug auf der verfügbaren Hard- und Software sein muß (91, 252). Er bezieht sich dabei auf die Lieferung von Daten im Netzwerk (NELSON 93, 2/57). Unter ‚schnell‘ versteht er etwa die Wartezeit für eine Telefonverbindung, die er mit durchschnittlich 3-15, höchstens 30 Sekunden angibt. Diese Zeit wird zwar Benutzerinnen immer noch voll Ungeduld vor ihren Geräten sitzen lassen, ist aber in den meisten Fällen kleiner als die der Lieferung eines umfangreichen Dokumentes im WWW, wenn es zwischen zwei Kontinenten transportiert werden muß. Die genannte Geschwindigkeit kann nur durch Techniken erreicht werden, die den heute eingesetzten überlegen sind. NELSON gibt zwar an, daß für Xanadu solche entwickelt worden wären, hat sie aber nirgendwo dokumentiert, so daß Zweifel angebracht sind. Darauf wird weiter unten noch eingegangen.

NELSON hat seine Anforderungen an Informationssysteme in Regeln gefaßt, die er „Nelsons Kanon“ nennt. Er hat sie 1987 aufgestellt in der Erwartung des Aufkommens von Systemen, die „alle Informationen, die Sie brauchen, zu jeder Zeit, zu jedem Ort“ versprechen (*Dream Machines*, 149). Sie sollen hier als seine eigenen Anforderungen an Xanadu kurz aufgeführt werden (der leichten Lesbarkeit wegen als *Beschreibungen* eines idealen Systems formuliert):

1. *Browser*:<sup>30</sup> Sie sind leicht zu verstehen und zu bedienen („in 10 Minuten“). Text erscheint nicht plötzlich beim Scrollen, sondern bewegt sich in das Sichtfeld der Benutzerin. Es können beliebige Browser benutzt werden, die unabhängig von der Datenhaltung der Server funktionieren (ebd., 150).

29 NELSON nennt explizit NLS des sonst bewunderten DOUGLAS ENGELBART als Gegenbeispiel, welches viel Schulung erfordert (80, 1013).

30 NELSON nennt die Client-Programme von Xanadu ‚Front-Ends‘. Obwohl ‚Browser‘ dem nicht ganz entspricht, weil Front-Ends ausdrücklich auch Editierfunktionen bieten, soll dieser in der WWW-Welt gängige Begriff in der vorliegenden Arbeit verwendet werden.

2. *Datenzugriff*: Obwohl im Prinzip beliebig große Dateien im System zulässig sind, erfolgen alle Zugriffe schnell. Sie löst die Benutzerin durch das Anwählen von Links aus, welches ohne das Wissen um Dateinamen und -orte möglich ist (ebd.).
3. *Links*: Zwischen allen Teilen von Objekten im System können Links angelegt werden. Überall können Anmerkungen angebracht und Lesezeichen plazierte werden. Es gibt auch private Anmerkungen in öffentlichen Dokumenten (ebd.).
4. *Höhere Strukturen*: Text wird in mehreren Fenstern dargestellt. Es kann ein Link von einem Text in einem Fenster zu einem anderen Text in einem anderen Fenster gesetzt werden. Material kann in ‚Anthologien‘ zusammengestellt werden, die Benutzerinnen untereinander austauschen können. Material aus einer Anthologie kann in seinem ursprünglichen Kontext betrachtet werden. Alte Versionen werden aufgehoben, weil Links darauf existieren könnten. Diese Links überstehen auch Änderungen des Textes, auf den sie zeigen (ebd., 150f).
5. *Urheberschutz*: Er wird vom System gewährleistet (was weiter unten noch näher beschrieben werden wird). Material kann natürlich trotzdem verteilt werden, allerdings gegen eine bestimmte Gebühr (ebd., 151).
6. *„Freiheit von Spionage und Sabotage“*: Überwachung kann durch Pseudonyme, anonyme Zugangsberechtigungen (Accounts) und den Verzicht auf bestimmte Aufzeichnungen verhindert werden, was vom System unterstützt wird. Auch in anonymen Accounts werden Daten aufbewahrt. Gegen Sabotage gibt es Vorrichtungen zur Überprüfung der Authentizität von Daten (ebd.). Zu diesem Punkt werden im Xanadu-Entwurf keine weiteren Ausführungen gemacht, außer, daß aus anderen Dokumenten übernommene Daten in ihrem ursprünglichen Kontext betrachtet werden können.

#### 3.4.2.2 Spezifikation

Da Xanadu nie auf den Markt gekommen ist, ist von ihm nicht mehr als seine Spezifikation bekannt. NELSON hat Xanadu auf informelle Weise konzeptioniert. Er nennt eine konzeptionelle Spezifikation ‚Virtualität‘, die auch enthält wie sich ein System für die Benutzerin „anfühlt“ (ebd., 2/7f). Diese ‚obere Virtualität‘ erscheint als Text-, Film-, 3D- oder andere Dokumente, die Links besitzen. Sie hängt teilweise von der Darstellung der Daten durch einen Browser ab und nicht nur von der Systemstruktur selbst (ebd., 4/8). Auf der Ebene der ‚unteren Virtualität‘ besteht ein Dokument in Xanadu nur aus den zwei Elementarten Bytes und Links. Sie werden jeweils getrennt verwaltet (ebd.). Die untere Virtualität ist die Ebene der Datenverwaltung in Servern (‚Backends‘) (2/6f).

Die Einteilung in die drei Ebenen der oberen Virtualität, der unteren Virtualität und der physikalischen Datenspeicherung entspricht der Drei-Schema-Architektur für Datenbanken (auch ANSI/SPARC-Architektur genannt, s. Abbildung 10):

1. Die *interne Ebene* entspricht der physikalischen Speicherstruktur der Datenbank. Sie wird durch ein physikalisches Datenmodell beschrieben.<sup>31</sup> Nur auf dieser Ebene existieren Daten tatsächlich (ELMASRI U. NAVATHE 94, 26).
2. Die *konzeptionelle Ebene* entspricht der Sicht auf die Datenbank für alle Benutzerinnen. Statt physikalischen Datenstrukturen beschreibt sie Dateneinheiten (Entities), -typen, Relationen, Benutzerinnenoperationen und Constraints (ebd.).

<sup>31</sup> Diese und die anderen beiden Ebenen sind als sogenannte ‚Schemen‘ in der Datenbank selbst abgespeichert, w oher die Architektur ihren Namen bezieht.

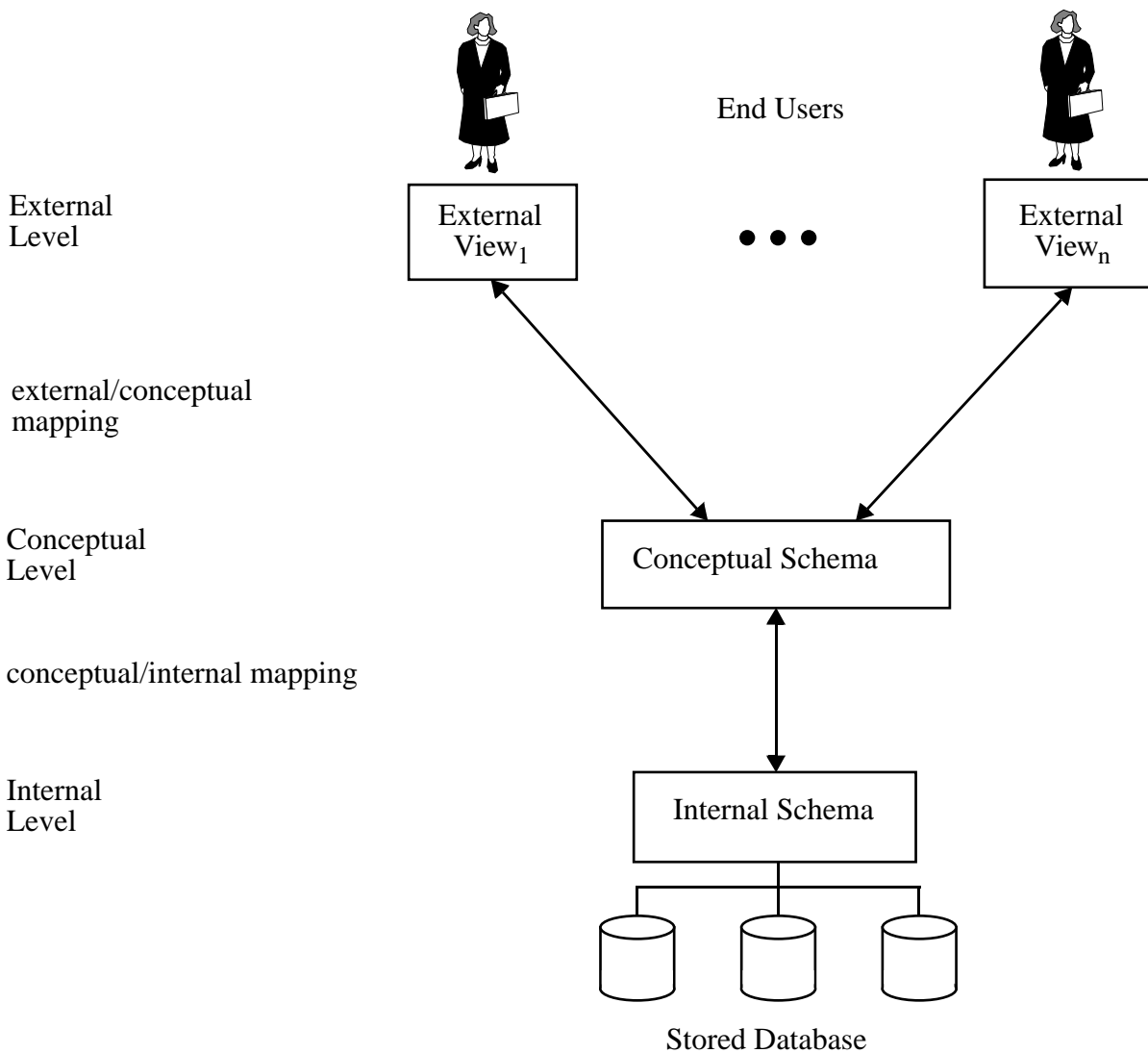


Abb. 10. Drei-Schema-Architektur (ELMASRI U. NAVATHE 94, 26)

3. Die *externe Ebene* besteht aus einer Reihe von Benutzerinnensichten. Diese Sichten beschreiben jeweils eine Ausschnitt der konzeptionellen Datenbank für eine bestimmte Benutzerinnengruppe und verstecken den Rest des konzeptionellen Modelles (ebd.).

Eine konzeptionelle Datenbankbeschreibung enthält z.B. die Typen der Daten einer Firma (z.B. einen Typ ‚Name‘, der 30 Textzeichen umfaßt) und die Verhältnisse zwischen ihnen. Die Daten werden durch Sachbearbeiterinnen der Firma erstellt und geändert, wobei die Bearbeiterinnen jeweils eine externe Sicht auf sie besitzen. Zwei externe Sichten können sich unterscheiden. In Datenbank-Management-Systemen (DBMS) werden zur Laufzeit Datenbankabfragen und die dazugehörigen Antworten zwischen den drei Ebenen übersetzt (ebd., 27).

Die obere Virtualität in Xanadu entspricht der externen Ebene der Drei-Schema-Architektur, die untere Virtualität der konzeptionellen Ebene und die Ebene der physikalischen Datenspeicherung der internen Ebene. Auf der konzeptionellen Ebene sind in Xanadu keine speziellen Dokumenttypen festgelegt. (Denkbar wären für seine Anwendungen ‚Schwarze Bretter‘, ‚Telekonferenzen‘ etc.). Sie werden auf der externen Ebene in den Browsern behandelt (NELSON 93, 2/50).

Xanadu soll einen weltweiten Zugriff erlauben und im Prinzip beliebig wachsen können. Daher muß es eine verteilte Datenbank enthalten. Konzeptionell soll diese ihren Benutzerinnen jedoch

einheitlich als ein ‚Docuverse‘ erscheinen (ebd., 2/53). Anwenderinnen sollen auf alle Dokumententeile auf gleiche Weise zugreifen können, egal wo sie gespeichert sind. Die Zugriffsgeschwindigkeit soll davon nicht wesentlich abhängen (NELSON 93, 2/53f), selbst dann nicht, wenn das System in Abhängigkeit von der Nachfrage wächst, wofür zunächst keine technischen Grenzen gesetzt sind (ebd., 2/56).

Auf das Xanadu-Netzwerk soll in den Xanadu-Stationen, über Telefonverbindungen, kommerzielle und professionelle Netzwerke<sup>32</sup> und über Gateways von anderen Netzen aus zugegriffen werden können. Benutzerinnen sollen dazu Bildschirm- und Druckerterminals und PCs verwenden können. Servern sollen über das Paketnetzwerk X.25 (in Deutschland Datex-P) und Satelliten verbunden sein (ebd., 2/55f). Aus Hardwaresicht hält NELSON das Xanadu-Netzwerk für einfach zu realisieren, wobei ihm auch die schnelle Datenlieferung über große Entfernungen als nicht allzu utopisch vorkommt (ebd., 2/54). Wie die Erfahrung mit dem WWW und der Übertragung anderer Multimediadaten im Internet lehrt, werden zwar teilweise hohe Übertragungsraten erreicht, aber v.a. bei Übertragungen über große Entfernungen, z.B. zwischen den USA und Deutschland, sind diese Raten teilweise so klein, daß gewisse Anwendungen unbenutzbar werden, z.B. die Abfrage von Hypertextdokumenten mit Bildern.

Aus Softwaresicht sind die Anforderungen an das Netzwerk nach NELSON höher als aus Hardwaresicht:

1. Dokumentkopien müssen aus Sicherheitsgründen im Netz verteilt werden, wobei Benutzerinnen auf alle Kopien eines Dokumentes wie auf ein einziges Dokument zugreifen können.
2. Alle Kopien müssen an ihren jeweiligen Orten geändert werden.
3. Änderungen müssen schnell im Netzwerk bekannt werden.
4. Für Dokumente, die nicht auf einem bestimmten Server liegen, muß es dort zumindest Informationen über eine Quelle geben (ebd., 2/54ff).

Die ersten beiden Punkte bedeuten, daß das System zwar Kopien von einem Dokument erstellt und auf verschiedene Server verteilt, die Autorin sie aber weiterhin nur an einer Stelle ändern muß. Leserinnen bekommen die Kopie eines Dokumentes zwar möglicherweise schneller geliefert als das Original, benutzen aber für die Anforderung dessen ursprüngliche Adresse. Kopien erhalten auf der externen Ebene (s. S. 62) keine eigenen Adressen. Der letzte Punkt der Anforderungen bezieht sich darauf, daß ein Browser immer über einen bestimmten Server auf alle Daten des Netzwerkes zugreift. Wenn von einer Benutzerin ein fremdes Dokument angefordert wird, daß ein Server nicht vorrätig hat, muß der Server wissen, woher er es bekommen kann.

### 3.4.2.3 Entwurf

Wie an anderer Stelle schon verlangt wurde, muß der Zugriff auf Daten in Xanadu schnell erfolgen. Die Zeitkomplexität der Algorithmen in Xanadu muß daher weniger als linear sein (in Abhängigkeit von der Anzahl der Server und der Menge der verwalteten Daten). NELSON gibt an, daß in Xanadu Algorithmen für das Speichern und Ändern von Daten und das Verbinden über Links mit einer dem Logarithmus bzw. der Quadratwurzel entsprechenden Zeitkomplexität verwandt werden. Wie er selbst sagt, ist die Komplexität in schnellen Baumstrukturen wie B-Bäumen für das Einfügen und Löschen von Daten ebenfalls logarithmisch, diese Strukturen waren daher Vorbild für die in Xanadu benutzten (ebd., 4/2f).

In einem Netzwerk kommt zu der Zeit für die Verwaltung von Daten noch die Zeit für die Übertragung von Kommandos und Daten (ebd.), die durch die Verwendung von Zwischenspeichern (Caches und Mirrors) gesenkt werden kann. Diese Speicher sind in Xanadu nicht dokumentiert,

32 Mit letzteren hat NELSON das ARPAnet, den Vorgänger des Internet gemeint.



auch wenn beschrieben wird, daß Daten zwischen Servern kopiert werden. Es findet sich bei NELSON auch kein Hinweis, daß die Leistungsangaben empirisch überprüft wurden. Es ist zweifelhaft, ob Xanadu sie wirklich erfüllen kann, wenn es jemals implementiert sein sollte.

Obwohl sich die spätere Arbeit an Xanadu auf das Netzwerk und seine Server beschränkt hat (worauf später eingegangen wird), stand zunächst die Arbeit an Benutzungsoberflächen der Client-Programme im Vordergrund. Eine Benutzungsoberfläche ist das ‚Parallel Textface‘.<sup>33</sup> Sie sollte der Darstellung von Texten dienen, zwischen denen Abhängigkeiten bestehen. Die Texte sollten nebeneinander in Fenstern dargestellt werden, mit Linien zwischen abhängigen Abschnitten. Abhängige Texte sollten gleichzeitig gescrollt werden (NELSON 87, *Dream Machines*, 42). Für Xanadu greift NELSON diese Idee wieder auf. In *Literary Machines* findet sich eine Abbildung, in der mehrere Texte, die sich aufeinander beziehen (z.B. eine Quelle und Anmerkungen dazu) nebeneinander dargestellt werden, indem sich die Rahmen ihrer Fenster berühren (93, 4/77).

Eine weitere fortgeschrittene Form von Hypertext wird von NELSON ‚Stretchtext‘ genannt. Bei Stretchtext handelt es sich um einen Textabschnitt, dessen Länge variabel ist. Eine Benutzerin kann sich aussuchen, in welcher Länge sie ihn lesen möchte. Stretchtext wird über ein Bildelement (etwa einen Schieber) bedient, welches den Umfang des dargestellten Textes verändert: Wenn das Element in die eine Richtung bewegt wird, wird der dargestellte Text länger (am selben Ort, wobei entsprechend Platz geschaffen werden muß), indem ein Wort nach dem anderen eingefügt wird. Wenn es in die andere Richtung bewegt wird, wird er entsprechend kürzer (ebd., 134). NELSON gibt nicht an, wie ein solcher Stretchtext erstellt werden kann. Die Vorstellung, daß eine Autorin viele verschiedene Versionen eines Textes produziert, die sich vom kürzesten bis zum längsten Text nur in jeweils einem Wort unterscheiden, scheint jedoch abwegig.

Im ersten populären Hypertextsystem für Personal Computer, ‚Guide‘ von Office Workstations (OWL), gibt es Ersetzungslinks, durch deren Aktivierung der Text, auf den sie sich beziehen, durch einen anderen ersetzt wird. Sie stellen nur eine einfache Art Stretchtext mit zwei Textlängen dar, also mit zwei Zuständen. Das generelle Konzept ist damit jedoch immerhin in einem kommerziellen Programm umgesetzt (NIELSEN 90, 91). Eine weitere Umsetzung findet sich in dem Entwicklungssystem Oberon/F von Oberon Microsystems. Dort werden Ersetzungslinks ‚Falten‘ genannt. Die beiden Zustände eines Faltenlinks wird durch die Anzahl der (französische) Anführungszeichen am Anfang und Ende einer Falte angezeigt (s. Abbildung 11 und Abbildung 12).

Um die obengenannten Anforderungen an das Xanadu-Netzwerk v.a. bezüglich der Zugriffsgeschwindigkeit auf Daten zu erfüllen, werden Übertragungen in Xanadu folgendermaßen realisiert:

1. Benutzerinnen haben über ihre Browser (Front-Ends) relativ langsame Verbindungen zum Netzwerk über Telefonleitungen, LANs etc. Die Browser kommunizieren mit den Servern (Back-Ends) über das Front-End-Back-End (FEBE)-Protokoll in ASCII.
2. Anfragen von Browsern werden über den Server, mit dem sie verbunden sind, auf weitere Server verteilt. Umgekehrt werden alle ankommenden Daten über diesen Server an den Browser zurückgeleitet.
3. Server sind über schnelle Verbindungen (s. S. 64) und das Back-End-Back-End (BEBE)-Protokoll (s. S. 83) miteinander verbunden.
4. Alle Server enthalten eine Abbildung der Organisation des gesamten Netzwerkes, welche jedoch nur an bestimmten Punkten mit Daten ‚ausgefüllt‘ ist.

<sup>33</sup> Das ‚Parallel textface‘ war der Aufhänger für die Entwicklung von Xanadu. Die beiden haben sich inzwischen auseinanderentwickelt, wobei Xanadu die Zugriffsfunktionen im Hintergrund bereitstellen soll. Das Design des ‚Parallel textface‘ stammt von 1972 (NELSON 87, *Dream Machines*, 43).

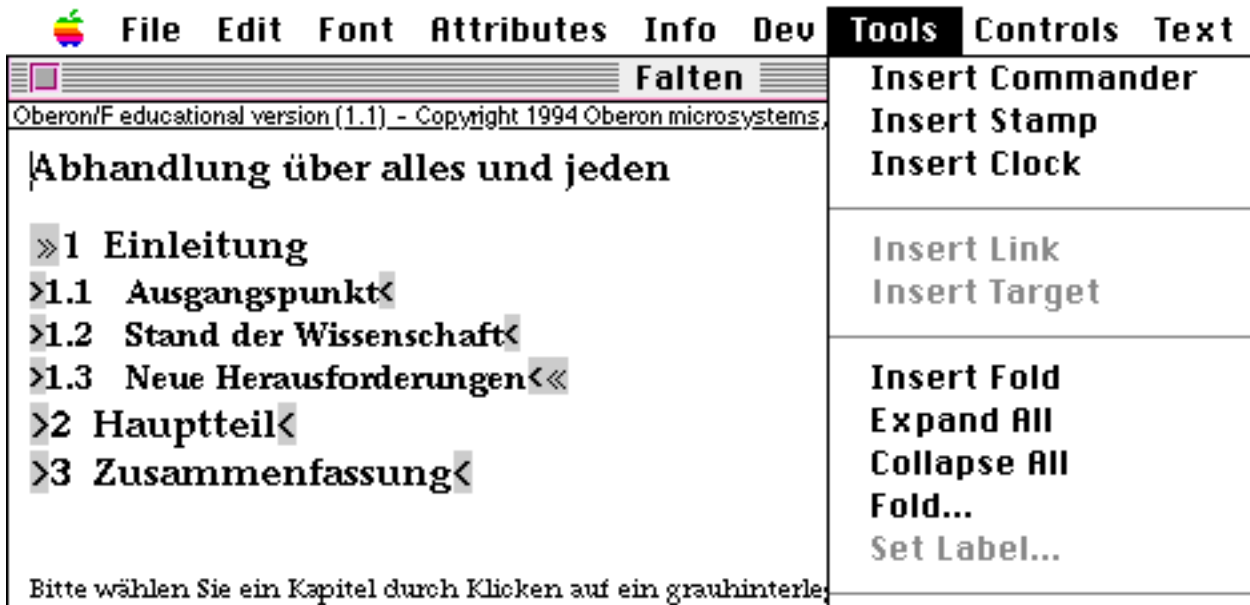


Abb. 11. Inhaltsverzeichnis mit ausgefalteter Ebene 1 (innerhalb der doppelten Anführungszeichen). Das Menü zeigt die Kommandos zur Verwaltung von Falten.

5. Bei Unerreichbarkeit eines Servers werden automatisch Kopien von Daten von anderen Servern angefordert (ebd., 4/74f).

Im Gegensatz zu HyperWave, bei dessen Entwicklung davon ausgegangen wurde, daß zwischen Clients und dem Server, mit dem sie verbunden sind, eine relativ schnelle Verbindung im Vergleich zu der zwischen zwei Servern existiert, gibt es in Xanadu also v.a. schnelle Verbindungen zwischen Servern. Der Xanadu-Ansatz erlaubt es daher, einen Server viele Einzeldaten von anderen anfordern zu lassen, die dann in einer „Schauer“ an den Client weitergereicht werden (auf ihn ‚einprasseln‘). Ein heutiges Xanadu würde jedoch auch auf dem Internet basieren und ähnliche Bedingungen wie HyperWave vorfinden.

NELSON gibt nicht an wie binäre Daten, z.B. Bilder, über ein ASCII-Protokoll (s. Punkt 1) übertragen werden sollen. Für die Datenübertragung in Mailboxen und dem Internet werden verschiedene Kodierungen benutzt, die Binärdaten auf ASCII-Zeichen abbilden. Die einfachste ist die, jedes Byte als Hexadezimalzeichen zu übertragen, wobei es dann zwei Zeichen im Datenstrom belegt. Es werden dadurch doppelt so viele Zeichen übertragen wie an Binärdaten vorliegen. Praktisch eingesetzte Kodierungen sind sparsamer, z.B. das Verfahren, um verschlüsselte Texte, die in Binärform vorliegen, per E-Mail verschicken zu können, in der nur druckbare ASCII-Zeichen erlaubt sind (s. LINN 93).<sup>34</sup>

ASCII-Datenverbindungen benötigen bei Einsatz eines solchen Verfahrens mehr Übertragungskapazität und Rechenzeit (für das Kodieren) als Binärverbindungen. Das haben auch die Xanadu-Entwicklerinnen erkannt. In der Dokumentation, die einer FEBE-Interface-Programmbibliothek für das Schreiben von Xanadu-Front-Ends beiliegt, wird betont, daß für den Einsatz von Xanadu eine Datenleitung bereitstehen muß, die beliebige Bytes unverändert übertragen kann. Eine reine ASCII-Übertragung ist dafür nicht geeignet (XANADU OPERATING COMPANY 87, Abschn. 8). Das BEBE (Back End-Back End)-Protokoll wird auf S. 83 näher beschrieben.

34 Dieses Verfahren wird ‚Base 64‘ genannt, nach den 64 ASCII-Zeichen (Groß- und Kleinbuchstaben, Ziffern, ‚+‘ und ‚/‘), auf die die 256 möglichen Bytes abgebildet werden. Dazu werden jeweils drei aufeinanderfolgende Bytes der Ursprungsdaten bei der Senderin auf vier aus den 64 genannten Zeichen abgebildet und bei der Empfängerin wieder zurückverwandelt (LINN 93, 13f).

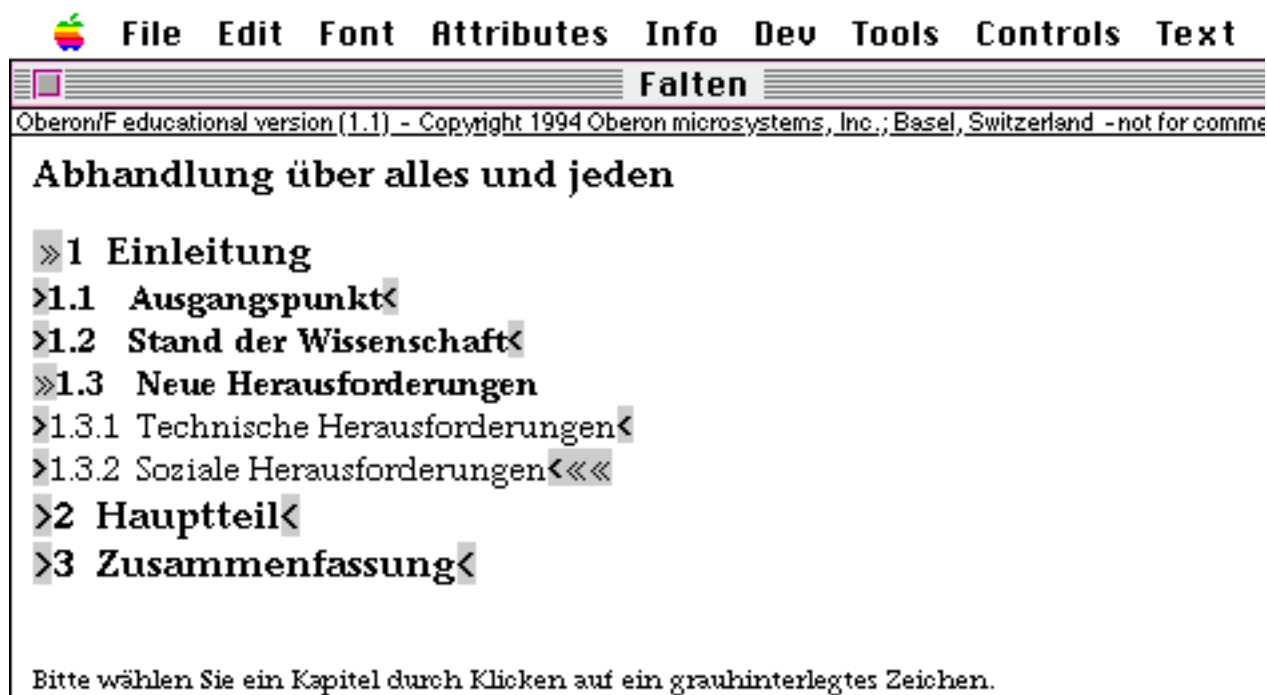


Abb. 12. Inhaltsverzeichnis mit zusätzlich ausgefalteter Ebene 1.3.

In Xanadu können verschiedene Daten gespeichert werden, z.B. Bilder, Filme, 3D-Objekte, Ton, DNA und Referenzen auf externe Objekte wie Menschen, Orte und Dinge (ebd.). Durch die redundante Speicherung<sup>35</sup> soll nach NELSON die Aufbewahrung kulturell wichtiger Dokumente in Xanadu sicherer sein als in heutigen Bibliotheken (93, 3/25). Alle Daten werden in separaten Dokumenten abgelegt. Jedes Dokument hat eine Eigentümerin und einen Namen (ebd., 2/29ff). Nur die Autorin kann Änderungen an einem Dokument vornehmen und erhält die anfallenden Tantiemen (ebd., 4/8f).

Es werden private und öffentliche Dokumente unterschieden. (Das Veröffentlichen und die Abrechnung von Dokumenten wird in Abschnitt 3.4.2.5 auf S. 84 beschrieben.) Alle Dokumente sind zunächst privat. Links, alternative und historische Versionen (s. weiter unten) können optional mitveröffentlicht werden (ebd., 2/42). Dokumente enthalten nicht nur ‚Inhalt‘, sondern auch Informationen über ihre eigene Manipulation und Darstellung (in Form von Daten oder Programmen), die über Links gespeichert sind und von Browsern ausgewertet werden (ebd., 4/55).

Xanadu unterstützt historische und alternative Versionen (‚Alts‘). Historische Versionen geben die Entstehungsgeschichte eines bestimmten Dokumentes an. Alternative Versionen sind aus demselben Dokument entstanden. In Xanadu kann durch die Entstehungsgeschichte eines Dokumentes ‚geblättert‘ und alternative Versionen können verglichen werden, indem gleiche Teile hervorgehoben werden (ebd., 3/12f). Es kann also nicht nur im Raum (klassisches Scrollen), sondern auch ‚in der Zeit geblättert‘ werden (ebd., 2/18).

Die Menge aller Teile eines Dokumentes inklusive ihrer historischen und alternativen Versionen heißt in Xanadu ‚Docuplex‘ (ebd., 2/21). Es gibt in Xanadu keine Dokumente an sich, sondern nur Docuplexe und Versionen:

Thus there is no ‚main‘ version of a thing, just the ongoing accumulation of pieces and changes, some of whose permutations have names and special linkages. ... You ask for a certain part of a certain version at a certain point in time. (ebd., 2/17)

35 inklusive der Speicherung in Servern, die sich unter der Erde oder im Weltraum befinden.

NELSON gibt nicht an, wie die Versionsverwaltung von Xanadu bedient werden könnte. Er beschreibt nur, wie verschiedene Versionen dem Namen nach unterschieden werden können. Dies geschieht durch Anhängen von Versionsnummern an den Grundnamen, wie wir noch sehen werden.

Die Benennung von Versionen ist eine, aber nicht die einzige Aufgabe des Configuration Management (CM). Unter diesem Namen wird die Versionsverwaltung als Teildisziplin des Software-Engineering erforscht, die sich mit dem „controlling of the evolution of complex systems“ befaßt (im Software-Engineering natürlich hauptsächlich als „Software Configuration Management“, „... its specialization for computer programs and associated documents“, s. TICHY 88, 1).<sup>36</sup> Weitere Aufgaben eines CM-Systems sind das Verfolgen von Änderungen (wer ein Objekt wann und warum geändert hat), die Auswahl einer Version aus einer Gruppe zusammenhängender Objekte, die Zusammenstellung für die Weiterverarbeitung und die Verwaltung ihrer Produkte (z.B. Kompilate bei der Softwareentwicklung) und schließlich die Verwaltung von gleichzeitigen Änderungen am selben Objekt, inklusive der Möglichkeit, die verschiedenen Versionen zu vergleichen und zu einer neuen Version zusammenzuführen (ebd., 2).

Versionen werden im CM in Revisionen und Varianten unterschieden. Ein Objekt wird eine Revision eines anderen genannt, wenn es durch Änderung einer Kopie des anderen entstanden ist. Die transitive, antisymmetrische und irreflexive Revisionsrelation zeichnet die Geschichte eines Objektes auf. Zwei Objekte werden als Varianten voneinander bezeichnet, wenn sie unter einer bestimmten Abstraktion ununterscheidbar sind. Die Variantenrelation ist also dreiwertig, weil neben den beiden Varianten auch die Abstraktion betrachtet wird. Für eine bestimmte Abstraktion, also zweiwertig betrachtet, handelt es sich um eine Äquivalenzrelation, da sie dann transitiv, symmetrisch und reflexiv ist (ebd., 5f).

Im Kontext des wissenschaftlichen Arbeitens kann es sich bei der Abstraktion z.B. um ein Thema handeln, welches die Varianten bearbeiten. Sie sind dann Varianten bezüglich dieses Themas (und nicht notwendigerweise bezüglich eines anderen, der Form der Arbeit etc.). Im Configuration Management wird mit Revisionen das bezeichnet, was NELSON ‚historische Versionen‘, mit ‚Varianten‘ das, was NELSON ‚alternative Versionen‘ nennt.<sup>37</sup> Ein Objekt kann neben einer Variante auch eine Revision eines anderen sein, in Abhängigkeit davon, ob die Variante durch Änderung des anderen Objektes entstanden ist (ebd., 6).

Als Versionsgruppe werden Objekte bezeichnet, die Revisionen oder Varianten voneinander sind. Durch die Revisionsrelation wird ein gerichteter, azyklischer Graph gebildet, der eine Entwicklungsgeschichte darstellt (s. Abbildung 13). Es handelt sich im einfachsten Fall um eine Liste, die mit dem ältesten Objekt beginnt und mit dem jüngsten abschließt (Revisionen 1.1 bis 3.1 in Abbildung 13). Die Variantenrelation hingegen verbindet parallele Arbeiten über ein gemeinsames Objekt (par.1 ist Variante von 1.3). Durch sie werden Seitenäste zu der Liste hinzugefügt, die teilweise wieder mit ihr verbunden werden (Objekt fix.1, das in Version 2.2 eingemischt wird, wodurch Version 3.1 entsteht).

Revisionen können in einem Konflikt zueinander stehen, wenn z.B. eine Revision aus einem Objekt weiterentwickelt wurde, ohne daß der Versionsprung in der Hauptversionsreihe abgewartet wurde (Revision con.1 steht im Konflikt zu 3.1). Solche Konflikte entstehen nur an Blättern eines Versionsbaumes. Sie können durch späteres Zusammenführen aufgelöst werden (ebd., 6f).

Die meisten Software-CM-Systeme benutzen das Dewey-Dezimalsystem<sup>38</sup> für die Benennung von Versionen, meist zweistellig in der Form *Hauptversion.Ebene* für Revisionen des Hauptzwei-

36 Als ‚Configuration‘ wird eine Ansammlung von Objekten bezeichnet, z. B. eine Menge von Dateien. Eine Konfiguration kann auch selbst Konfigurationen enthalten (ebd., 4f).

37 Bei ihm werden die Begriffe jedoch unschärfer verwendet und Varianten sind immer auch Revisionen. NELSON geht also generell davon aus, daß ein Dokument, das aus einem anderen entwickelt wurde, auch Bytes mit diesem teilt. Das kann man auch daran sehen, daß er *inhaltliche* Übernahmen von einem Dokument in ein anderes nie erwähnt.

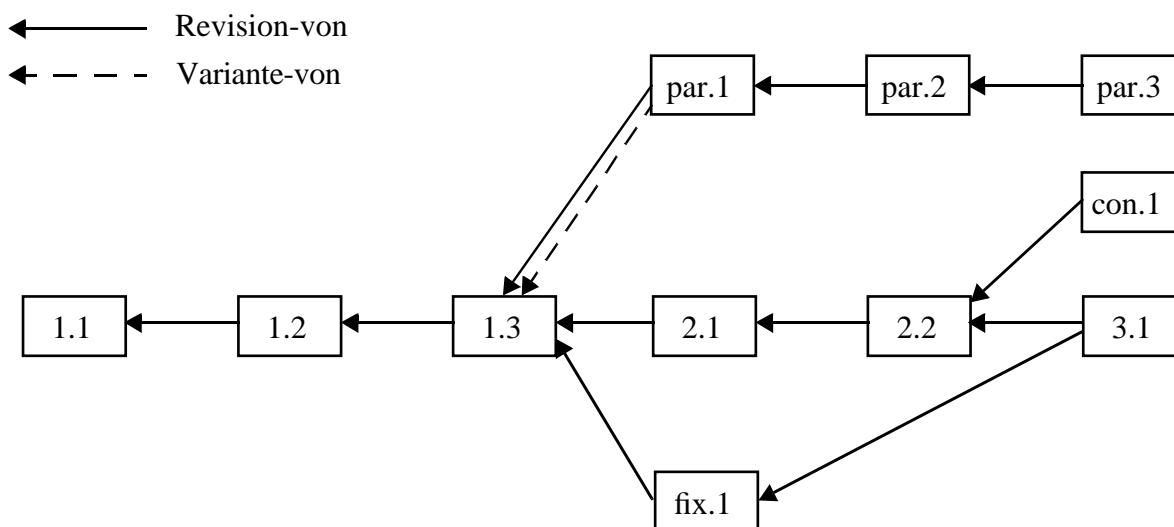


Abb. 13. Eine Versionsgruppe mit Revisionen und Varianten (nach TICHY 88, 7).

ges. Teilweise werden Versionen auf Nebenzweigen durch spezielle Numerierungen gekennzeichnet (z.B. durch 2, 4, ..., 2n durch Punkte getrennte Nummern, wenn Objekte des Hauptzweiges durch 1, 2, ..., n Nummern bezeichnet wurden. Objekte im Hauptzweig könnten also zweistellig, Objekte eines Nebenzweiges vierstellig nummeriert werden.) Diese Notationen hält TICHY jedoch für zu wenig übersichtlich (obwohl er sie in RCS, einem bekannten von ihm entwickelten CM-System, auch eingesetzt hat). Er schlägt vor, für jeden Unterzweig einen symbolischen Hauptbezeichner zu wählen und seine Objekte über eine Unternummer hochzuzählen (s. ‚par‘, ‚con‘ und ‚fix‘ in Abbildung 13). Die Lage eines Zweiges im Versionsbaum kann dann immer noch über die Revisionsrelation herausgefunden werden (ebd., 8), diese muß also übersichtlich darstellbar sein.

Revisionen werden durch Aus-Checken, Editieren und anschließendes Wiedereinchecken erzeugt. Beim Aus-Checken wird eine Kopie des betroffenen Objektes erstellt. Die Kopie kann dann bearbeitet werden. Solange sie ausgecheckt bleibt, kann sie (auch in einem Mehrbenutzersystem) nicht von anderen geändert werden. Wenn eine Benutzerin die ursprüngliche Version erneut aus-checkt, wird sie gewarnt, daß sie einen Nebenzweig im Versionsbaum erzeugt, der eventuell später wieder durch Mischen mit dem Hauptzweig verbunden werden muß (ebd., 7f).

Durch die Checkin-Operation wird eine Änderung in den Versionsbaum eingefügt, also für andere Benutzerinnen sichtbar. Zwischen Checkout und -in kann eine Version mehrmals überarbeitet werden. Wenn ein Editor eine Änderung sichert, wird eine neue Datei erzeugt. Alte Revisionen bis auf diejenige, die schließlich eingchecked wird, werden ‚Minor revisions‘ genannt und beim Einchecken gelöscht. Sie dienen der Datensicherheit (ebd.).

Beim Zusammenführen (‚Merging‘) werden eine Ausgangsversion und zwei ihrer Revisionen betrachtet. Es werden die Änderungen identifiziert, die die Revisionen von der Hauptversion unterscheiden und zusammengemischt. Konflikte können entstehen, wenn z.B. ein Absatz in einer Revision geändert und in einer anderen gelöscht wird. Konflikte können zwar automatisch identifiziert, jedoch nur manuell behoben werden. Um Änderungen besser verfolgen zu können, werden sie zusammen einem Kommentar der durchführenden Person aufgezeichnet (ebd., 8).

Um Platz zu sparen, wird nicht jedes einzelne Objekt einer Versionsgruppe vollständig gespeichert. Es wird vielmehr mit ‚Deltas‘ gearbeitet, die jeweils aus einem Satz von Änderungskommandos bestehen, um ein Objekt in ein anderes zu überführen. Objekte müssen also teilweise erzeugt

38 Dieses System wird auch in amerikanischen Bibliotheken verwendet. Bezeichner bestehen in ihm aus Zahlenfolgen, die durch Punkte getrennt sind.

werden, bevor auf sie zugegriffen werden kann. Durch spezielle Techniken wie das Komprimieren von Leerzeichen umfassen Deltas nur ein bis zwei Prozent der vollständigen Größe der betroffenen Dokumente (ebd.). (Wichtige Änderungskommandos sind solche zum Einfügen und Löschen von Teilen eines Objektes, also Textstücken in einem Textdokument.)

Deltas können vorwärts oder rückwärts gespeichert werden. Beim Vorwärtsspeichern wird die jeweils erste Version eines Zweiges vollständig gespeichert, während sich Deltas in späteren Versionen befinden. Beim Rückwärtsspeichern liegt entsprechend die letzte Version ohne Delta-Kommandos vor. Das Rückwärtsspeichern ist in der Praxis sinnvoller, weil die jeweils letzten Revisionen, die gewöhnlich öfter benötigt werden, schneller erstellt werden können (ebd., 9f).

NELSON gibt nicht an, in welcher Form Deltas in Xanadu in den Servern gespeichert werden. Da Versionen über Transklusion (s. weiter unten) erzeugt werden, also Textstücke, die sich in einem bestimmten Dokument gegenüber seiner Vorversion nicht geändert haben, in es eingebunden werden (s. NELSON 93, 2/38), ist es wahrscheinlich, daß Deltas in Xanadu vorwärts gespeichert werden, was vom Vertrauen NELSONS in die Effizienz seiner Algorithmen zeugen würde. Durch das Transkludieren kann in Xanadu auf eine Delta-Löschoperation verzichtet werden, weil gelöschte Dokumententeile einfach nicht eingebunden werden, in einem neuen Dokument also um sie ‚herumtranskludiert‘ wird. Damit ist aber nur die technische Umsetzung beschrieben. Wie die Versionsverwaltung in Xanadu bedient wird, ist immer noch unklar.

Deltas können im CM allgemein auf zwei verschiedene Arten erzeugt werden: direkt während der Erstellung von Objekten, also nach und nach in einem Editor, oder durch den Vergleich von Paaren fertiger Objekte. Für den ersten Ansatz sind spezielle (oder angepaßte) Editoren nötig, die jeden einzelnen Änderungsschritt aufzeichnen. (In Xanadu wird ein spezieller Editor eingesetzt, der die Schritte allerdings nicht selbst aufzeichnet, sondern jeweils direkt an einen Server weitergibt, der sie dann auf den eigentlichen Daten ausführt.)

Bei der Softwareentwicklung ist der erste Ansatz für praktische Anwendungen mit heterogenen Entwicklungsumgebungen ungeeignet. Es wird daher meist der zweite Ansatz verfolgt, bei dem Deltas durch Programme erzeugt werden, die (fertige) Dateien nach Abschluß eines Änderungsschrittes vergleichen, was natürlich eine kurze Wartezeit erfordert. Deltas werden gewöhnlich während des Checkin einer Datei erzeugt, wenn also eine neue Version in das CM-System zurückgespielt wird (TICHY 88, 9f).

Versionsgruppen und ihre Objekte können in Datenbanken oder Dateisystemen gespeichert werden. In den ersten Implementierungen von CM-Systemen (wie dem bereits erwähnten RCS) gibt es spezielle Kommandos, durch die eine Version einer Datei aus den vorhandenen Daten erzeugt und in eine temporäre Datei abgelegt wird, die dann mit klassischen Programmen (z.B. Editoren) bearbeitet werden kann.

Neuere Systeme integrieren die Verwaltung in Dateisysteme: Wenn eine Datei zum Lesen geöffnet wird, wird sie aus Deltas erzeugt. Wenn sie zum Schreiben geöffnet wird, wird sie zusätzlich ausgecheckt (und durch Schließen wieder eingecheckt). Dieser Ansatz bietet den Vorteil, daß Verwaltungsdaten des Systems für Benutzerinnen unsichtbar sind und daher nicht unabsichtlich zerstört werden können. Vor allem ist dieser Ansatz jedoch einfacher in bestehende Umgebungen einzufügen, weil er auf klassischen Dateioperationen basiert (ebd., 8f).

TICHY schildert leider nicht, wie ein ‚Three-way revision merging‘, also das Zusammenführen von zwei Revisionen einer vorliegenden Ursprungsdatei, bedient werden kann. Er nennt das UNIX-Programm ‚diff3‘, das ein solches Zusammenführen erlaubt. Dieses Programm stellt jedoch Unterschiede zwischen drei Dateien nur zeilenweise dar oder erstellt eine Datei für die Verarbeitung durch den Zeileneditor ‚ed‘, der das eigentliche Zusammenführen übernimmt.

Änderungen stellt ‚diff3‘ dar, indem es die Zeilen anzeigt, die sich in den drei Dateien unterscheiden. Die Zeilen werden in Blöcken angezeigt, in deren Kopf zunächst angegeben wird, ob sich

alle drei Dateien an der betrachteten Stelle unterscheiden oder nur eine Datei unterschiedlich ist. Danach folgt ein (ed-)Kommando als Delta-Befehl. Dabei handelt es sich entweder um einen Hinzufügungs- oder einen Änderungsbefehl, der sich auf eine Zeile bzw. einen Zeilenbereich bezieht und von neuen Zeilen gefolgt wird (SCHEURIG 94, 93f). Verschiebungen eines Textblockes werden also in zwei Kommandos umgesetzt, eines zum Löschen des Textes an seinem Ursprungsort und eines zum Einfügen desselben Textes an einer anderen Stelle.

In Xanadu kann ein Dokument Teile aus anderen Dokumenten enthalten, welche über Links verwaltet werden. Es heißt dann Verbunddokument (Compound document) (ebd., 2/32). Ein Verbunddokument hat wie jedes andere Dokument eine Besitzerin. Eingebundene Teile gehören jedoch den Besitzerinnen der Ursprungsdokumente. Aus veröffentlichten Materialien kann beliebig viel eingebunden werden (ebd., 2/33). Umgekehrt kann von jedem Zeichen in einem Dokument festgestellt werden, woher es ursprünglich stammt (ebd., 2/40). Ein besonderes Verbunddokument stellt das abgeleitete Dokument (Derivative document) dar. Es besteht aus Teilen eines ‚Originals‘ und Änderungen zu ihm, die eine Autorin geschrieben hat (ebd., 2/37). Ein anderes ist das Vergleichsdokument (Intercomparison document), welches das Verhältnis zwischen zwei Dokumenten zeigt. Es enthält dazu Auszüge aus diesen Dokumenten und Kommentare (ebd., 2/39).

Den Einschluß von Bytes aus einem Dokument in ein anderes nennt NELSON ‚Transklusion‘ (früher ‚Inklusion‘), wobei die Bytes in dem zweiten Dokument so behandelt werden, als ob sie in ihm physikalisch vorhanden wären (s. S. 73) (ebd., 4/10f). Eine Transklusion kann beliebig lang sein, z.B. auch das ganze Ursprungsdokument umfassen und verändert dieses Dokument nicht (ebd., 2/32). Transklusionen werden in Xanadu durch spezielle Links implementiert. Da sie zusätzlich das Springen zu den Ursprungsdokumenten ermöglichen (ebd., 2/34), werden sie im folgenden teilweise gemeinsam mit anderen Linkarten besprochen.

NELSON nennt folgende Anwendungsbeispiele für die Transklusion:

1. Der Jahresbericht eines Konzerns kann Statistiken aus Abteilungen enthalten, die zu den jeweiligen Teilberichten führen.
2. In einem Kinderbuch können Bilder eingebunden werden, die aus anderen Geschichten stammen und zu diesen führen.
3. Zitate aus einem wissenschaftlichen Text führen zu entsprechenden Originalen (ebd., 2/35).<sup>39</sup>

Diese Beispiele belegen, daß NELSON tatsächlich die ganze Literatur auf seinem System abbilden will. Für diese Arbeit ist v.a. das letzte Beispiel interessant.

Transklusionen sollten nicht mit Zitaten verwechselt werden. Transklusionen stellen die allgemeine technische Basis für alternative und historische Versionen und ‚Trails‘ im Sinne BUSHS dar, Zitate Vereinbarungen des Literaturbetriebes. Zitate sollen sich optisch vom sie umgebenden Text abgrenzen, durch Anführungsstriche oder Einrücken. Transklusionen können hingegen unsichtbar sein, z.B. wenn eine Autorin keine Vorstufen eines Textes offenlegen möchte, was sie bei der Veröffentlichung einstellen kann (s. weiter unten). Mittels einer Transklusion kann Text weiterhin nur wörtlich übernommen werden, ein Analogon zu inhaltlichen Übernahmen bietet das Konzept nicht. (Inhaltliche Übernahmen können jedoch durch Sprunglinks kenntlich gemacht werden.)

Transklusionen stellen die technische Umsetzung des Prinzips der Übernahme von altem Text in neuen dar, wobei es sich bei dem alten Text sowohl um eigenen einer Autorin als auch fremden handeln kann. Die umgekehrte Einbindung von neuem in alten Text, als was z.B. das Annotieren verstanden werden kann, ist mittels der Transklusion dadurch umsetzbar, daß ein neues Dokument

<sup>39</sup> Weniger von der Anwendungs- als von der technischen Seite kann man die Transklusion mit der Funktion von Microsoft Word 6.0 vergleichen, eine Datei in eine andere per Referenz einzufügen, so daß sie wie ein Bestandteil der einbindenden Datei erscheint. Es können auch nur Teile einer Datei eingebunden werden, z. B. mit Namen versehene Textbereiche aus Word-Dateien oder über Koordinaten bestimmte Ausschnitte aus Microsoft Excel-Dateien.

erzeugt wird, welches den neuen Text enthält und in das der alte Text transkludiert wird. Ein Dokument kann nur solche Texte per Transklusion einbinden, die die gleiche oder eine geringere Sicherheitsstufe besitzen, z.B. können private Dokumente öffentliche einbinden, aber nicht umgekehrt (ebd., 2/46f).

Die Transklusions-Operation ist transitiv, eingebundenes Material kann also wieder eingebunden werden. Daraus ergeben sich verschiedene Herausforderungen an die Bedienbarkeit und Technik des Systems. Bezüglich der Bedienbarkeit stellt sich die Frage, wie Transklusionen übersichtlich dargestellt werden können. In Xanadu sollen Transklusionen in einem Browser-Fenster erst auf Anfrage hervorgehoben dargestellt werden (ebd., 4/53). Dies fällt für NELSON unter Browser-Funktionen, die er generell nicht näher beschreibt.

Transklusionsbeziehungen könnten in einem eigenen Browserfenster als Graph dargestellt werden, mit Dokumenten als Knoten und Transklusionsbeziehungen als Kanten. Eine weitere Möglichkeit bestünde darin, eine Transklusion direkt im Textfenster eines Browsers typographisch, farblich oder durch eine Einrahmung herauszustellen. Besonders eine typographische oder farbliche Hervorhebung würde jedoch bei langen und geschachtelten Transklusionen stören (wenn wir davon ausgehen, daß nicht nur die Transklusionen des aktuellen Dokumentes, sondern auch deren Transklusionen hervorgehoben werden).

Am übersichtlichsten ist also die Lösung mit Rahmen. Diese können aus speziellen Zeichen am Anfang und Ende des Textes (s. Abbildung 11 und Abbildung 12 für die Hervorhebung von Stretchtext) oder aus graphischen Kästen bestehen. Letztere benutzt Nelson übrigens, um in schematischen Zeichnungen das Transkludieren darzustellen, welches er auch ‚Windowing‘ nennt (s. ebd., S. 2/32ff). Da Kästen, die sich über mehrere Seiten erstrecken, unübersichtlich sind, könnten beide Rahmenformen eingesetzt werden, je nachdem, wie lange die Transklusion ist: Kästen für kurze und hervorgehobene eckige Klammern für lange Einbindungen.

Die Transitivitätseigenschaft der Transklusionsrelation bringt die Gefahr von Schleifen mit sich, also das Einbinden von Text (im folgenden ‚Quelle‘ genannt), der selbst den einbindenden Text (das ‚Ziel‘) enthält. Schleifen müssen verhindert werden, weil sie einen Browser dazu veranlassen würden, beliebig viel Text zu laden, bis er keinen Speicher mehr hat. Um Schleifen zu verhindern, muß entschieden werden, ob das Ziel in der Quelle (oder einer ihrer Transklusionen) bereits enthalten ist. Wenn dies der Fall ist, darf nicht eingebunden werden.

Um diese Frage entscheiden zu können, müssen alle Transklusionen der Quelle rekursiv durchsucht werden, wobei darauf geachtet werden muß, daß nur die Teile einer Transklusion betrachtet werden, die wiederum selbst in ein anderes Dokument transkludiert sind. Die Überprüfung auf Schleifen kann zu zwei Zeitpunkten erfolgen: Bei der Erstellung, wenn also eine Benutzerin die Einbindungsoperation im Browser festlegt, oder der Darstellung, wenn eingebundener Text im Browser dargestellt werden soll, nachdem er vom Server angefordert wurde. Obwohl der zweite Fall leichter implementiert werden kann, weil beim Laden alle Transklusionen eines Textes sowieso rekursiv abgearbeitet werden müssen, ist der erste vorzuziehen, weil dabei die Autorin (im Gegensatz zur Nutzerin) einer potentiellen Schleife gewarnt werden kann.

Transklusionen haben gegenüber Kopien (die z.B. durch ‚Copy-and-Paste‘ in ein Dokument eingefügt wurden) den Vorteil, daß Platz gespart wird und Updates des Ursprungsdokumentes durch das System verwaltet werden (ebd., 2/36f). Es gibt zeit- und ortsfeste Transklusionen. Fest in der Zeit ist eine Transklusion dann, wenn sich die Darstellung in dem einbindenden Dokument nicht ändert, wenn sich das eingebundene Dokument ändert. Ortsfest (und fließend in der Zeit) ist eine Transklusion, wenn sich Änderungen des eingebundenen Dokumentes direkt auf das einbindende Dokument auswirken (ebd., 2/37).

In einem wissenschaftlichen Kontext erscheint v.a. die ortsfeste Transklusion sinnvoll, weil als Zitat eingebundener, fremder Text sich nicht ändern darf, wenn die fremde Autorin ihren Text



ändert. NELSON gibt nicht an, wie diese Zeitfestigkeit der Transklusion erreicht werden soll. Wahrscheinlich jedoch mit Hilfe der Versionsverwaltung: Wenn eine Autorin ein Dokument ändern will, muß sie eine neue Version davon erstellen. Die Transklusion bezieht sich weiterhin auf die alte Version, zu der sie erzeugt wurde. Bei der Darstellung des transkludierten Textes erscheint jedoch eine Meldung, falls es eine neue Version des eingebundenen Dokumentes gibt, damit eine Leserin sich über Änderungen informieren kann. NELSON erklärt auch nicht, wie ortsfeste Transklusionen implementiert sind. Es ist vorstellbar, daß ihre Adressen in den einbindenden Dokumenten geändert und Spannen zu Spannenmengen aufgeteilt werden, wenn sich eine eingebundene Bytefolge ändert oder teilt. Allerdings schreibt NELSON explizit, daß sich Adressen nicht ändern, wenn mehr Material hinzukommt (s. die Besprechung von Adressen weiter unten).

Ein sofortiges und ständiges Anpassen von Transklusionen und anderen Links beim Ändern eines Dokumentes kann in Xanadu leicht umgesetzt werden, weil Dokumente direkt im Datenspeicher des Servers bearbeitet werden. Wenn also im Editorteil des Browsers Bytes gelöscht, verschoben oder hinzugefügt werden, geschieht dies wohl zunächst im Hauptspeicher des Computers, auf dem der Browser läuft (darüber läßt sich NELSON nicht weiter aus), Daten werden aber nicht auf einem lokalen Massenspeicher abgelegt, sondern auf dem des Servers. Ein Server übernimmt also tatsächlich die vollständige Verwaltung der Bytes, bis hinunter auf die Ebene, die der von Operationen gängiger Textverarbeitungssysteme im Hauptspeicher des entsprechenden Computers entspricht (ebd., 4/10).

Aus dem Gesagten ergibt sich, daß Texte in einem Xanadu-Browser nur bei Verbindung zu einem Datenserver erstellt werden können. Da diese Verbindung nicht immer gewährleistet werden kann (z.B. unterwegs oder bei technischen Störungen) und außerdem die Datenmanipulation im Server viele Übertragungen erfordert, scheint dieser Ansatz nicht sehr praktikabel. Auch wenn NELSON dies nicht ausdrücklich erwähnt, müßte lokaler Massenspeicher zumindest als Schreibcache benutzt werden, um größere Datenblöcke in bestimmten Zeitabständen auf einmal zum Server zu übertragen. Zum Speichern oft angeforderter Dokumentfragmente wäre ein solcher Puffer als Lesecache natürlich auch nützlich, gängige WWW-Browser benutzen ihn daher auch.<sup>40</sup>

Im Gegensatz zu Dateien in Dateisystemen existieren Xanadu-Dokumente nicht physisch, sondern nur virtuell. Ihr einziger physischer Aspekt ist der ‚Nucleus record‘ auf ihrem jeweiligen Ursprungsserver (Home server, s.u.). Dokumente werden erst auf Anforderung erstellt. Ein Xanadu-Dokument besteht logisch aus einer Folge von Bytes, von denen einige aus anderen Dokumenten stammen (mit möglicherweise anderen Ursprungsservern) und in ihm nur als Referenz auftauchen. Dokumente können daher über das ganze Xanadu-Netzwerk verteilt gespeichert sein (ebd., 4/6f).<sup>41</sup>

Wie schon beschrieben, bestehen Dokumente auf der konzeptionellen Ebene aus Bytes und Links. In den Bytes sind die ‚Inhalte‘ eines Dokumentes abgelegt, also Textzeichen, Pixel o.ä. Hingegen wird Mark-up in Text- oder Metadaten in Grafikdateien über Links gespeichert. Bytes und Links besitzen jeweils ein Ursprungsdokument, in dem sie erzeugt wurden und dem sie daher logisch gehören. Bytes und Links besitzen daneben möglicherweise mehrere Heimatdokumente

40 Interessanterweise wurde vor einiger Zeit das Konzept des ‚Net Computers‘ NC (als Gegenstück zum PC) vorgestellt, der selbst (zur Kostenminimierung) keine Festplatte enthält und Daten nur im Hauptspeicher oder auf einem Massenspeicher in einem Server bearbeiten kann. Dieser Ansatz wird dahingehend kritisiert, daß Festplattenspeicher inzwischen so preiswert ist, daß sich un vermeidbare Verzögerungen bei Verzicht auf ihren Einsatz nicht rechtfertigen lassen (s. den NC-Lobby-Server <http://www.nc.com>).

41 Dokumente werden daher erst bei der Abfrage zusammengesetzt. Eine solche Abfrage heißt in der Xanadu-Diktion ‚Pouncing‘, nach dem Sturz von (Trickfilm- ?)Katzen auf Gegenstände, welche erst während des Stürzens erscheinen (ebd., 2/16).

(Home documents), also solche, in die sie über Transklusion eingebunden sind, aber nur ein Ursprungsdokument (ebd., 4/9).

Bytes heißen bezüglich ihres Heimatdokumentes ‚native‘, eingebundene ‚nicht-native‘, beide zusammen ‚virtuelle‘ Bytes des Dokumentes. Bei der Anfrage eines Dokumentes werden alle seine virtuellen Bytes geliefert. Native und nicht-native unterscheiden sich nur in der Zugriffsgeschwindigkeit und möglicherweise der Eigentümerinnenschaft (ebd., 4/11). Nur die Eigentümerin darf ein Originaldokument ändern (ebd., 4/9). Wenn sie Bytes in einem solchen löscht, bleiben diese Bytes in allen Dokumenten erhalten, in welchen sie transkludiert sind (ebd., 4/11). NELSON bezieht sich dabei anscheinend auf zeitfeste Transklusionen (s. S. 72).

Positionen oder Verbindungen werden in Xanadu über Links gespeichert (ebd., 4/10). Sie sind so wichtig wie der Text, den sie verbinden (ebd., 2/62). Links können Kommentare, Lesezeichen, Platzhalter, Fußnoten, Randnotizen, Hypertextsprünge und Notizzettel darstellen (ebd., 2/24f). Sie können zwischen allen abgespeicherten Daten eingefügt werden (ebd., 2/28), bei öffentlichen Dokumenten oder privaten mit erlaubtem Zugriff nicht nur von den Besitzerinnen der entsprechenden Dokumente, sondern durch alle Benutzerinnen des Systems (ebd., 2/42).

Dies ist bei dem bereits beschriebenen System HyperWave, das weltweit von v.a. Universitäten eingesetzt wird, ebenfalls möglich. Damit das Setzen von Links bei einer Vielzahl von Servern und Benutzerinnen praktikabel bleibt und der Text von populären Dokumenten nicht in einer Vielzahl von Links untergeht, die darauf verweisen und zusammen mit dem Dokument angezeigt werden, ist in Xanadu vorgesehen, Links herauszufiltern, z.B. nach ihrem Ursprung (v.a. nach Autorinnen). Ein komplexer Filter, den sich eine Benutzerin konstruiert hat, führt dann dazu, daß sie nur auf eine ‚Subwelt‘ von Xanadu Zugriff hat. Bei der Bedienung eines Filter handelt es sich um eine Browserfunktion, die, wie alle anderen Eigenschaften von Browsern auch, nicht weiter beschrieben ist (ebd., 4/60).

Es sind Zweifel angebracht, wie sinnvoll der Aufbau einer ‚Subwelt‘ über komplizierte Filter wirklich ist. Zunächst stellt sich die Frage, wie die Filterfunktion bedient werden kann, so daß ihre Anwendung vorhersehbar bleibt. Vor allem scheinen aber weder das Herausfiltern noch das Zulassen von Texten bestimmter Autorinnen die Lösung des Problems der Linkflut darzustellen: Wenn nur bestimmte Autorinnen herausgefiltert werden, werden bei einem populären Dokument immer noch zu viele Dokumente angezeigt. Wenn andererseits nur bestimmte Autorinnen zugelassen werden, erhält eine Benutzerin keine neuen Hinweise von unbekanntem Autorinnen (die z.B. heute in den Internet-News sehr nützlich sein können). Das Anpassen des Filters an die Art der angefragten Dokumente schließlich erscheint zu aufwendig. Filtern wird also dazu führen, daß potentiell interessante Informationen nie bei einer Benutzerin ankommen.

Links verbinden nicht einzelne Bytes, sondern Folgen von solchen (ebd., 4/42). Diese Folgen werden ‚Spannen‘ (Spans) genannt (ebd., 4/24). Nach NELSON beziehen sich Links nicht auf Positionen von Daten, sondern auf die Daten selbst, wobei er wohl ortsfeste Links im Auge hat. Daher bleiben Links mit speziellen Bytes auch in geänderten Versionen verbunden, solange die Bytes dort noch vorhanden sind. Links können also auch ortsfest sein (s. S. 72). Wie bei ortsfesten Transklusionen ist jedoch nicht klar, wie sie implementiert sind. Wenn es verschiedene Versionen eines Dokumentes gibt, das Links enthält, kann von einer Textstelle, an der ein Link befestigt ist, zu allen entsprechenden Textstellen in den anderen Versionen gesprungen werden (ebd., 2/24ff).

Links, die von einem Dokument aus auf andere zeigen, werden ‚Out-Links‘ genannt, Links, die von anderen Dokumenten aus auf ein Dokument zeigen, entsprechend ‚In-Links‘ dieses Dokuments. Out-Links gehören der Eigentümerin des jeweiligen Dokuments, In-Links nicht (ebd., 2/31). Zu jedem Dokument können alle seine In-Links angezeigt werden (ebd., 2/47). Dies ist im schon beschriebenen HyperWave auch möglich. Dadurch kann zu Dokumenten gesprungen werden, die mit dem vorliegenden dadurch in Verbindung stehen, daß ein Link von ihnen auf es zeigt. Das Ver-

folgen von Links in beliebiger Richtung ist außerdem für die konsistente Datenhaltung nötig. Dabei handelt es sich keineswegs um eine theoretische Anforderung. Wenn ein Dokument nämlich stark überarbeitet wird, auf das Links gesetzt sind, müssen einige Links entfernt oder zumindest als unbenutzbar markiert werden.

NELSON nennt das sofortige Anzeigen von In-Links eines Dokumentes in einer Fußnote „the hardest feature“ bezüglich der Implementierung (ebd.). Alle Links im System zu untersuchen, ob sie auf das vorliegende Dokument zeigen, ist dafür kein brauchbarer Ansatz. Die In-Links eines Dokumentes müssen also gemeinsam dort gespeichert werden, wo das Dokument selbst abgelegt ist, genau wie Out-Links auch in der ‚Nähe‘ ihres Dokumentes gespeichert sein müssen, um effizient verwaltet werden zu können.

Bezüglich der Netzentfernung zwischen den betroffenen Computern entspricht das Setzen eines Out-Links der Änderung eines Dokumentes, z.B. in der Form von Hinzufügung von Text, das Setzen eines In-Links von einem anderen Dokument aus hingegen der *Abfrage* des ersten Dokumentes. (Die Datenübertragung, die ja nach einer Abfrage eines Dokumentes für mehr Netzbelastung sorgt als der Abfragebefehl selbst, soll für das Setzen von In-Links zu einem Dokument als vernachlässigbar angenommen werden, weil die betroffenen Datenpakete relativ klein sind.)

Dokumente werden in verteilten Hypertextsystemen in bandbreitenmetrischer Nähe ihres Servers geändert und dann auf ihn übertragen. Das gilt auch für das Setzen eines Out-Links, was daher effizient erfolgen kann. Das Abfragen von Dokumenten, und entsprechend das Setzen von Out-Links, ist deshalb so aufwendig bezüglich der entstehenden Netzbelastung, weil es von einem beliebigen Punkt des Netzes aus erfolgen kann, also teilweise große Strecken zwischen den beteiligten Netzstationen zu überbrücken sind. Außerdem können viele Netzteilnehmerinnen solche Abfragen gleichzeitig vornehmen.

Durch den Einsatz von Caches auf lokalen Computern und in speziellen Servern wird die Häufigkeit der Dokumentenübertragungen im WWW verringert. Dadurch wird auch die Schwierigkeit der Beantwortung vieler Abfragen aufgefangen, außerdem durch die Verwendung großer Server mit vielen Netzanschlüssen durch Anbieterinnen vielgefragter Daten (z.B. der Browseranbieterin Netscape oder der WWW-Verzeichnisanbieterin Yahoo).

Weil für das auf dem laufenden halten von Links Schreiboperationen nötig sind, sind diese beiden Ansätze dafür nicht zu gebrauchen. Bei den WWW-Caches handelt es sich nämlich um Lesecaches. In HyperWave werden lokale Server hingegen auch als Schreibcaches benutzt. Wie wir gesehen haben, erfolgt dort ein Linkupdate nicht sofort, sondern wird zu bestimmten Zeiten von Server zu Server weitergereicht, also ‚geflutet‘ (s. die Beschreibung des P-Flood-Algorithmus auf S. 48). Das Setzen eines In-Links in einem Browser wird also zunächst auf dem Server zwischengespeichert, mit dem der Browser verbunden ist. Der Link wird erst durch das Fluten zu dem Server des entsprechenden Dokumentes übertragen, wo der In-Link abgeschlossen wird.

Wenn die beschriebenen erfolgreichen Datenanbieterfirmen auch In-Links auf ihre Dokumente in ihren eigenen Servern speichern würden, müßten sie die Server mit zusätzlichem Massenspeicher ausrüsten. Vor allem aber müßte es eine Möglichkeit geben, sich In-Links gefragter Dokumente selektiv anzeigen zu lassen, weil sonst für erfolgreiche Dokumente mehr Link- als Dokumentendaten im Browser dargestellt würden. Das hat NELSON auch schon erkannt (93, 2/47). Sein schon erwähntes Konzept sieht daher vor, Links nach Herkunftsort, Autorin und Zeit zu sieben. Links müssen daher also zunächst die entsprechenden Attribute besitzen.

Ähnlich wie Bytes besitzen auch Links ein Urprungs- und ein Heimatdokument (ebd., 4/42).<sup>42</sup> Dort stellen sie neben den Bytes einen zweiten Strom von adressierbaren Daten dar, der (zumindest

42 Dies wird an anderer Stelle der genannten Quelle explizit anders dargestellt (S. 4/11). Möglicherweise stellt die Gleichbehandlung von Bytes und Links in dem genannten Sinn eine späte Entwicklung dar .

in der beschriebenen Version 1988 von Xanadu) nicht geändert werden kann (ebd., 4/11). Das Material, welches durch einen Link verbunden wird, muß sich nicht im Heimatdokument des Links befinden, dieses gibt nur an, wem der Link gehört. Ein Link kann also auch Teile von anderen Dokumenten verbinden (ebd., 4/12).

Der Vorteil dieses Konzeptes der Speicherung eines Links zwischen zwei Dokumenten in einem dritten liegt darin, daß damit jeder Link einen festen ‚Ort‘ und Adresse hat (was in Xanadu Voraussetzung für das Herausfiltern ist). Außerdem können Links so in Dokumenten zusammengefaßt, u.a. auch gemeinsam gelöscht werden (was in Xanadu jedoch nicht möglich ist, weil Dokumente ja nie gelöscht werden).

Das Problem des Konzeptes liegt jedoch darin, daß ein Link in seinem Heimatdokument zunächst nicht sichtbar ist. Generell sind ja nur die Anker von Links in den verbundenen Dokumenten sichtbar und nicht die Verbindung zwischen ihnen. Man kann sich jedoch eine Übersichtsdarstellung aller Links vorstellen, die in einem bestimmten Dokument gespeichert sind, ähnlich der Darstellung aller In- und Out-Links (aber damit nicht zu verwechseln!). Diese Übersichtsdarstellung würde alle Dokumente auflisten (jeweils als Icons oder Namen in einer Graphik oder Liste), die durch Links in einem bestimmten Dokument verbunden sind. Glücklicherweise dürfte diese Darstellung selten nötig sein, weil die meisten Links in den Dokumenten gespeichert werden dürften, von denen sie ausgehen, weil Autorinnen wohl am ehesten von eigenen auf andere Dokumente verweisen wollen.

Alle Standardlinks besitzen in Xanadu drei ‚Enden‘, die sogenannten Endmengen, die ‚Von-Menge‘ (From-set), die ‚Zu-Menge‘ (To-set) und die ‚Drei-Menge‘ (Three-set), wobei letztere den Typ des Links darstellt. Diese Mengen enthalten Adressen von Spannen. Linkenden können sich in Xanadu nicht nur, wie in HyperWave, auf Dokumententeile beziehen, sondern auch unzusammenhängend sein (ebd., 4/42f).

Die Drei-Menge kann mehrere Zahlen als Typen enthalten, die beliebig gewählt sind und wie Adressen behandelt werden, aber eben keine Bytes adressieren (s. die Beschreibung von Tumblern ab S. 77) (ebd., 4/43). NELSON gibt wieder keine Darstellung einer solchen Endmenge im Browser an. Eine einzelne Spanne könnte farblich oder typographisch hervorgehoben werden, was bei sich überlappenden Spannen jedoch unübersichtlich wäre.

Links sind gewöhnlich gerichtet, von der Von- zur Zur-Menge, es gibt jedoch auch ungerichtete (ebd., 4/42f).<sup>43</sup> Von fortgeschrittenen Nutzerinnen sollen auch Links mit mehr als drei Enden definiert werden können, die entweder gerichtet sind oder nicht (ebd., 2/27f). Eine dokumentierte Grundmenge von Linktypen wird den Benutzerinnen von Xanadu von Anfang an zur Verfügung gestellt. Es soll ein Prozedere eingerichtet werden, nach dem weitere Links standardisiert werden können (ebd., 4/52). Über dieses Prozedere werden aber keine weiteren Angaben gemacht.

Standardlinks lassen sich in Meta-, Text- und Literaturlinks einteilen. Metalinks beziehen sich jeweils auf ein ganzes Dokument. Vordefiniert ist ein Titel- und Autorinnenlink, der die entsprechenden Angaben eines Dokumentes darstellt und ein ‚Dokumentenersetzungslink‘, welcher, wenn vorhanden, auf eine neuere Version des vorliegenden Dokumentes verweist. Korrekturen, Kommentare, Gegenstücke, Übersetzungen, Überschriften, Absätze, Zitate, typographische Auszeichnungen und Fußnoten werden durch Textlinks gekennzeichnet (ebd., 4/53). Typographische Auszeichnungen und Absatzenteilungen können also in einem einbindenden Dokument dadurch geändert werden, daß die entsprechenden Links durch andere ersetzt werden, ohne das ursprüngliche Dokument zu verändern.

43 Einen ungerichteten Link kann man dafür benutzen, zwei sich entsprechende Passagen in zwei verschiedenen Dokumenten zu kennzeichnen, die jedoch nicht auseinander hervorgegangen sind, beispielsweise von zwei verschiedenen Autorinnen unabhängig voneinander gemachte Entdeckungen.

Hypertextlinks stellen Material zum Durchstöbern zusammen. In Xanadu vorhanden sind einfache Sprünge, getypte Sprünge, solche, die einen Lesepfad vorschlagen (einen ‚Trail‘ im Sinne BUSHS) und Links, die einen Text oder eine Grafik zu einer anderen Version oder Ansicht erweitern (ebd., 4/54). Literaturlinks schließlich ermöglichen den Sprung zu einer Quelle, verweisen auf eine alternative Version oder einen Kommentar zum aktuellen Dokument, zertifizieren und authentifizieren Material und ermöglichen das Verschicken von E-Mail<sup>44</sup> (ebd., 4/55). Über die letzten drei Funktionen werden sonst keine Angaben gemacht.

Wie wir gesehen haben, beziehen sich Links immer auf Punkte oder Abschnitte in Daten. Es muß in Xanadu also eine Möglichkeit geben, solche zu adressieren. NELSON gibt an, daß ein Adressierungssystem folgende Anforderungen erfüllen muß:

- Verwaltung vieler Server, Benutzerinnenaccounts, Dokumente und Versionen
- Verwaltung vieler Adressen
- Verzicht auf eine Obergrenze der Anzahl verwalteter Objekte
- Adressierbarkeit einzelner Bytes
- sparsamer Umgang mit Speicherplatz
- schnelle Adreßmanipulation
- Verwaltung von Versionen
- Verzicht auf eine zentrale Verwaltungsinstanz, die z.B. Adressen vergibt (ebd., 4/15ff)

Da in Xanadu Hauptadressen (Serveradressen) zentral vergeben werden, bezieht sich der letzte Punkt darauf, daß für die Ableitung von Server-, Account- und Dokumentadressen aus den Hauptadressen die Zentrale nicht benötigt wird.

Xanadu-Adressen werden ‚Tumbler‘ genannt<sup>45</sup> (ebd., 4/16). Das Tumblersystem ist vom Dewey-Decimal-System inspiriert (wie die Versionsnumerierung in Configuration Management Systemen, s. S. 68), „which, despite its faults, does not waste a lot of space on empty characters“ (ebd., 4/17). Tumbler bestehen aus positiven Ganzzahlen beliebiger Größe (ebd., 4/13). Jedes Objekt wird über seinen Tumbler adressiert, unabhängig von seinem physikalischen Ort, auch wenn es auf mehrere Server verteilt ist. Die jeweilige Adresse bleibt erhalten, auch wenn dem System mehr Material hinzugefügt wird (ebd., 4/24).

Tumbler lösen Probleme der Versionsverwaltung, der Verwaltung vieler Server und Benutzerinnen und des angenommenen dezentralen Wachstums des Xanadu-Netzwerkes (ebd., 1/38).<sup>46</sup> Sie haben die folgenden Eigenschaften:

- Dezentralisierung: Zu ihrer Anwendung ist keine zentrale Kontrollinstanz nötig.
- Eigentum: Nummern gehören natürlichen oder juristischen Personen, die Unternummern (s. weiter unten) definieren und weitergeben dürfen.
- Unabhängigkeit von der Inhaltsstruktur: Obwohl Tumbler baumartig strukturiert sind, zwingen sie diese Struktur nicht dem Inhalt auf.
- Unabhängigkeit von Kategorien: Tumbler indizieren nicht nach Kategorien oder Themen, sondern verwalten nur Speicherplatz.
- Unabhängigkeit von Systeminternas.

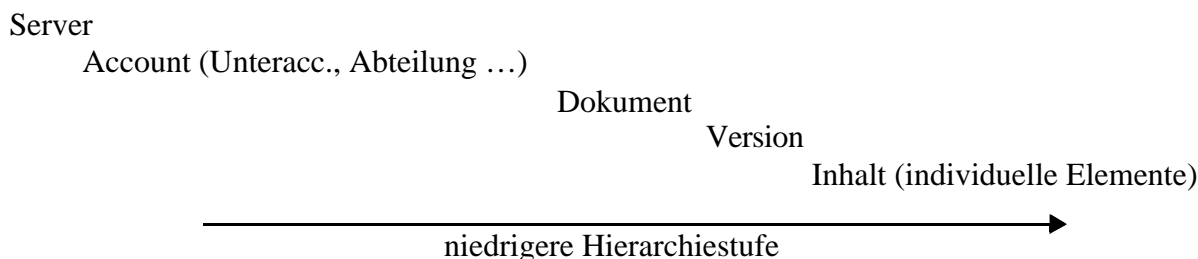
44 als Link auf eine Mail an ihrem Ursprungsort. Das Öffnen einer Mailbox bedeutet in Xanadu die Anzeige der seit dem letzten Mal hinzugekommenen Mails (ebd., 4/55).

45 weil dies nach ‚Number‘ klingt und der amerikanische Name für die sogenannte Zuhaltung in Schlössern ist, mit denen sie nach NELSON Ähnlichkeiten aufweisen (ebd., 4/16).

46 NELSON beschreibt sie in einer Überschrift blumig als „A Write-Once Address System of Forking Multipart Integers Specifying A Master Ever-Growing Tree-Address Space“ (ebd., 4/15).

- Beschränkung auf Orte: Tumbler verwalten keine Zeit. (ebd., 4/17f)

Der ‚Tumblerraum‘ (Tumblr space) ist die Menge aller möglichen Tumbleradressen. Das ‚Dokuversum‘ (Docuverse) ist die Menge aller *belegten* Tumbleradressen (ebd., 4/23). Durch die Tumblerverwaltung erscheint das ganze Dokuversum als ein sich gabelndes Dokument (ebd., 4/18). Eine Tumbleradresse stellt also eine Hierarchie dar, die die folgenden Ebenen enthält (ebd., 4/23):



Server, Accounts, Dokumente und Text können beliebig in den Tumblerraum eingefügt werden, ohne daß sich andere Adressen ändern. Daher sind alle Adressen permanent (ebd., 4/19). Server, Accounts und Dokumente haben jeweils genau ein Objekt, welches sich in der Hierarchie über ihnen befindet und möglicherweise mehrere unter sich. Unter einem Server können weitere Server eingerichtet werden, dasselbe gilt für Accounts. Server können Accounts enthalten (ebd., 4/19). Server, Accounts und Dokumente belegen zwar Adressen im Tumblerraum, im Gegensatz zu Bytes und Links aber sonst keinen Speicherplatz (ebd., 4/23).

Der hierarchische Aufbau des Adressierungssystems steht eigentlich im Widerspruch zur Ablehnung von Hierarchien durch NELSON. Er schreibt dazu:

... the tumblr system promotes in some ways a linking and searching hierarchy, and we endeavor to take best advantage of it, even though our materials are intrinsically non-hierarchical. (Whether to take advantage of any features not a part of our original virtuality design can be philosophically argued, but any method has certain directions of advantageous use, and we have endeavored to optimize the use of this one.) (ebd., 4/26)

Tumbler werden durch spezielle Zahlen mit einem Additions- und einem Subtraktionsoperator dargestellt (ebd., 4/18). Sie werden im ganzen System benutzt, auf der internen und der externen Ebene (ebd., 4/16) Es werden z.B. in Verträgen Benutzerinnen und Dokumente über Tumbler bezeichnet, jeweils neben den Klartextnamen (s. ebd., 5/19).

Ein Tumbler besteht aus einer Folge von positiven Ganzzahlen, ‚Humber‘ genannt. Seine Humber werden als ‚Ziffer‘ des Tumblers bezeichnet. Ziffern werden durch Punkte getrennt. Wenn Ziffern einzeln stehen, wird jeweils ein Punkt davor und einer dahinter geschrieben, z.B. „.373.“, „.675923.“ und „.40.“ (ebd., 4/19). Eine Ziffer kann durch das Anhängen weiterer Ziffern differenziert werden, z.B. „.2.4.6.312.“ für das 312. Objekt unter dem sechsten unter dem vierten unter dem zweiten Objekt (ebd., 4/20). Eine Folge von einer oder mehreren Ziffern ohne Nullziffer wird ‚Feld‘ genannt. Ein Feld oder Teilfeld kann ein bestimmtes Objekt adressieren, z.B. einen Server oder ein Account. Felder werden durch den Haupttrenner ‚.0.‘ verbunden, also die Ziffer Null (ebd., 4/20). Eine Tumbleradresse darf höchstens drei solche Ziffern Null enthalten, Differenzentumbler (s. weiter unten) beliebig viele (ebd., 4/28).

Ein Tumbler besteht aus vier Feldern. Das erste stellt einen Server dar. Dahinter verbirgt sich meistens ein Servercomputer, es kann sich aber auch um ein logisches Gerät handeln, z.B. um einen Bereich eines Servers oder eine Menge von Servern. Serveradressen beginnen immer mit einer Eins, z.B. „.1.2368.792.6“ (ebd., 4/28ff). Das zweite Feld bezeichnet die Eigentümerin von Dokumenten. Die Eigentümerin selbst kann es weiter unterteilen. Für kommerzielle Zwecke kann es sich bei dem zweiten Feld um ein Account handeln, mit Abteilungen, Projekten und Aufgaben

als Unterfeldern. Privat kann es sich um ein Individuum handeln, mit Interessensgebieten und Ablagen als Unterbereiche. Ein Beispiel für eine Nutzerinnenadresse ist

1.2368.792.6.0.6974.383.1988.352 (ebd.).

Dokumente werden durch das dritte Tumblerfeld dargestellt. Unterfelder dieses Feldes bedeuten Unterdokumente und Versionen. Adressen von Versionen eines Dokumentes können von der Benutzerin festgelegt werden, sind also Interpretationssache. Das Dokumentenfeld sollte eine Benutzerin für alle Dokumente mindestens zweistellig wählen, damit über die letzte Stelle Versionen hochgezählt und gleichzeitig alle Versionen über die vorletzte Stelle angesprochen werden können. Benutzerinnen steht es also zunächst frei, wieviele Ziffern sie für Dokumentenfelder festlegen. Das Xanadu-Server-Kommando für die Erzeugung einer neuen Version eines Dokumentes, ‚CREATE-NEWVERSION‘, liefert jedoch eine neue Tumbleradresse zurück, die vom Server aus der des Ursprungsdokumentes abgeleitet ist (ebd., 4/66). Die Verwaltung von frei interpretierbaren Versionsnummern ist insgesamt auch schwer vorstellbar, weil sich die Versionsverwaltung dann nur schwer automatisieren ließe. Beispiele für Tumbler mit drei Feldern:

1.2368.792.6.0.6974.383.1988.352.0.75 (Dokument) und  
1.2368.792.6.0.6974.383.1988.352.0.75.2 (Version) (ebd.).

Das vierte Feld eines Tumblers kann Bytes und Links (und weitere Inhalte) adressieren. Dokumententeile werden also ähnlich der Kurzbelege sakraler Texte (s. S. 19) bezeichnet. Die verschiedenen Inhaltsarten werden über die höchstwertige Stelle im vierten Feld unterschieden, ‚.1.‘ für Bytes und ‚.2.‘ für Links. Es werden logische, also auch transkludierte Bytes eines Dokumentes adressiert. Es wurde keine vernünftige Interpretation einer Unterteilung von Bytes gefunden, für Links gibt es zumindest keine Festlegung in dieser Hinsicht. Beispiele:

1.2368.792.6.0.6974.383.1988.352.0.75.2.0.1.9287 (Byte) und  
1.2368.792.6.0.6974.383.1988.352.0.75.2.0.2.352 (Link) (ebd.).

Unterbäume des Xanadu-Netzwerkes werden durch Spannen (Spans) adressiert. Dabei handelt es sich um Tumbler-Intervalle, die jeweils durch einen Anfangs- und Endtumbler angegeben werden. Eine einzelne Xanadu-Adresse bezeichnet einen Pfad durch einen Baum von Adressen, wie eine WWW-Adresse im Prinzip auch. Dieser Pfad muß nicht bis hinunter auf ein Blatt, also eine Bytefolge, reichen, sondern kann bei einem Zweig ‚stehenbleiben‘. Dies geschieht durch einen Tumbler, der aus dem Präfix von einer oder mehrerer Adressen besteht. Der Tumbler bezieht sich dann auf alle Daten, die mit Tumblern adressiert werden, die mit diesem Präfix beginnen. Auf diese Daten kann ein Link gesetzt, und es kann in ihnen gesucht werden.

Die Text- und Linkoperationen in Xanadu beziehen sich nicht auf einzelne Spannen, sondern auf Spannenmengen (Span sets). Diese Mengen dienen dazu, ganz bestimmte Elemente anzusprechen, im Gegensatz zu aufeinanderfolgenden Elementen, die durch eine Spanne allein adressiert werden können (ebd., 4/25). Anfang- und Endpunkt einer Spanne können aus Elementen verschiedener Typen (Server, Accounts etc.) bestehen. Aus diesen Typen ergeben sich dann die Typen der jeweiligen Spanne (ebd., 4/24f).

Eine Spanne kann entweder über zwei Tumbler oder einen Tumbler und einen Differenzentumbler angegeben werden. Bei einem Differenzentumbler handelt es sich um den ‚Abstand‘ zwischen zwei Tumblern. Die beiden Formen einer Spanne können durch Tumblerarithmetik ineinander überführt werden (ebd., 4/25). Tumblerarithmetik wird im Browser ausgeführt. Sie behandelt immer ganze Tumbler (deren Ziffern über Ganzzahlarithmetik verrechnet werden). Es kann entweder ein Differenzentumbler aus zwei Adreßtumblern oder ein zweiter Adreßtumbler aus einem ersten und

einem Differenzentumbler berechnet werden. Es gibt nur diese beiden Formen der Subtraktion, sie ist also nicht kommutativ (ebd., 4/34).

Die Verwendung von Nummern anstatt Namen als Adressen in Xanadu hat den offensichtlichen Vorteil, daß Benutzerinnen sich für kleine Dokumente jeweils keinen eigenen Namen ausdenken müssen. Technisch ermöglichen Nummern die Berechnung von Spannen aus den Angaben der Nummern von zwei Dokumente, die ‚überspannt‘ werden sollen. Durch Spannen können ad hoc mehrere Dokumente gleichzeitig adressiert werden, eine Spanne stellt also ein virtuelles Verzeichnis dar. (Sonst gibt es in Xanadu ja keine Verzeichnisse, die von Benutzerinnen erzeugt werden können.) Dieses Konzept funktioniert jedoch nur für Dokumente, deren Nummern im Tumblerraum aufeinanderfolgen oder voneinander abgeleitet sind (Versionen). Um andere Dokumente zusammenzufassen, muß eine Spannenmenge konstruiert werden, also eine Liste von Spannen, die diese Dokumente enthält. Dafür müssen jedoch keine Berechnungen angestellt werden. Wie Spannen in Browsern erstellt werden könnten, darüber schweigt sich NELSON aus.

Für Adressen höherer Ebene, also solche von Servern und Accounts, scheinen Spannen ebenfalls wenig sinnvoll. Server einer Institution (z.B. einer Universität oder Firma) werden nämlich alle aus einer einzigen Nummer als Unternummern abgeleitet, wodurch sie durch Angabe dieser Nummer gemeinsam adressiert werden können. Die gleichzeitige Adressierung der Server mehrerer Institutionen oder einer Reihe von Servern von einer Institution ist über Nummern unübersichtlich und in vielen Fällen sinnlos. Zwischen Servern, die mit aufeinanderfolgenden ersten Ziffern im Serverfeld beginnen, dürfte nämlich nur selten eine inhaltliche Verbindung bestehen. ‚Top-Level‘-Tumbler würden in Xanadu nämlich wahrscheinlich chronologisch vergeben werden.

Sinnvoll und auch von anderen Systemen bekannt ist die Verwaltung von Ausschnitten und Versionen von Dokumenten über Nummern. Adressen für Ausschnitte bestehen aus einer Reihe von Byte- oder anderen Adressen (wie z.B. beim ‚Byte Range Protocol‘ für das WWW, s. S. 128). Versionsadressen bestehen aus einer Folge von Nummern, die durch Punkte getrennt sind (s. S. 68f). Diese beiden Adreßformen beziehen sich jedoch immer auf bestimmte Grundbezeichner von Dokumenten. Die Grundbezeichner bestehen in den meisten Fällen aus Namen, und die Bereichs- oder Versionsnummern werden an sie angehängt.

Wo die Vorteile einer reinen Benennung von Objekten im Hypertextnetzwerk durch Nummern zweifelhaft sind, sollte sie zumindest keine Nachteile aufweisen. Solche gibt es jedoch. Nummern können zwar leicht maschinell verarbeitet werden, sind für Menschen jedoch unhandlich, weil sie eine geringe Redundanz aufweisen. Sie können v.a. schwer gemerkt und aufgeschrieben werden. Das gilt zwar teilweise auch für lange Adressen, die aus Namen bestehen. Zumindes der wichtigste Teil einer Adresse, im WWW der Name des Servers (s. S. 34), läßt sich meist leicht merken. Auf dem Server kann ein Dokument dann eher gefunden werden. Auch wenn Adressen in einem Hypertextsystem aus Namen bestehen, sollte die Benutzung von Nummern möglich sein. Dies gilt selbst dann, wenn Nummern bereits zur Bezeichnung von Versionen benutzt werden.

Transklusionen und Sprunglinks weisen einige Gemeinsamkeiten, aber auch viele Unterschiede auf, sowohl in ihrer jeweiligen Bedienung als auch ihrer technischen Umsetzung. Die Gemeinsamkeiten und Unterschiede sollen hier kurz diskutiert werden, um die Möglichkeiten der jeweiligen Bedienung und Implementierung, das Thema des nächsten Kapitels, vorzubereiten. In Xanadu ähneln sich Transklusionen und Sprunglinks darin, daß beide Dokumententeile verbinden, die über Tumbler adressiert werden, es sie beide in zeit- und ortsfesten Ausführungen gibt und daß die durch sie jeweils dargestellte Relation transitiv ist. Transklusionen beinhalten außerdem die Funktion von Sprunglinks, weil von einem eintranskludierten Text zu seinem Original gesprungen werden kann.

Wenn es in einem Hypertextsystem eine Versionsverwaltung gibt, können zeitfeste Transklusionen und Sprunglinks relativ einfach implementiert werden. Die Links beziehen sich dann immer auf eine bestimmte Version eines Dokumentes. Eine zeitfeste Transklusion eines Textabschnittes



kann jedoch für praktische Belange durch eine Kopie des Abschnittes ersetzt werden, wenn von dieser Kopie aus ein Link zum Original führt. Dies ist möglich, weil sich eine zeitfeste Transklusion nach Definition nie ändert. Ein solcher einkopierter Text hätte den Vorteil, daß er in den meisten Fällen schneller geladen wird als ein dynamisch eingebundener Text, weil er sich bereits in dem entsprechenden Dokument befindet. Allerdings würde dabei kein Server garantieren (soweit er es überhaupt könnte), daß der Text korrekt ‚zitiert‘ wurde, was NELSON sehr am Herzen liegt. Die Korrektheit könnte nur durch ein Verfolgen des Sprunglinks überprüft werden, was immerhin leichter als das Überprüfen eines Zitates in einem Papierdokument wäre.

Die Ortsfestigkeit sowohl von Transklusionen als auch Sprunglinks stellt eine besondere technische Herausforderung dar, weil die Spannenmengen, die eingebundenen bzw. Zieltext adressieren, bei der Änderung des Textes im Quelldokument mitgeändert werden müssen. Dies erscheint nur dann möglich, wenn direkt auf den Datenstrukturen für Text, Transklusionen und Sprunglinks in einem speziellen Editor oder Server (wie bei Xanadu) gearbeitet wird. Bezüglich des Erzeugens von Deltas treten im Configuration Management ähnliche Probleme auf (S. 70). Da dort jedoch nur die Unterschiede zwischen zwei Versionen interessant sind und es keine Verweise von außen auf einen Text gibt, also keine Änderungen von Textpositionen protokolliert werden müssen, reicht es aus, wenn die Deltas erst beim Einchecken erzeugt werden. Bei einem Hypertextsystem müßten Links jedoch laufend angepaßt werden.

Transklusionen und Sprunglinks unterscheiden sich auch in einigen Punkten. Links liegen zwischen (mindestens) zwei vorhandenen Textstücken, Transklusionen hingegen fügen ein Textstück an einer bestimmten Position ein und verschieben den dahinterliegenden Text, im eingefügten Zustand verbinden sie also zwei gleiche Texte. Daraus ergeben sich wesentliche Unterschiede: Da beim Einfügen einer Transklusion tatsächlich der Text verändert wird, in den sie eingefügt wird, kann diese Operation nur von der Besitzerin des jeweiligen Dokumentes ausgeführt werden. Links hingegen können von allen Benutzerinnen auf alle Dokumenten gelegt werden, weil sie sich nur auf Positionen in ihnen beziehen und Text höchstens typographisch verändern, damit sie leicht erkannt werden können.

Schleifen, die sich durch die bei sowohl der Link- als auch der Transklusionsrelation vorhandene Transitivitätseigenschaft bilden können, stellen bei Links kein Problem dar, weil die Links erst bei Aktivierung ausgewertet werden. Da Transklusionen jedoch schon bei ihrer Darstellung ausgewertet werden und Text einfügen, könnte eine Schleife dazu führen, daß ein Browser geschachtelten Text lädt, bis er keinen Speicher mehr hat (o.ä.). Mögliche Lösungen dieses Problems wurden oben bereits angedeutet. Weiterhin bedeutet die Tatsache, daß Transklusionen beim Laden des transkludierenden Dokumentes automatisch ausgewertet werden, daß für jede einzelne Transklusion ein Textstück von dem entsprechenden Server angefordert werden muß. Wenn es sich um entfernt voneinander liegende, verschiedene Server handeln, ist dieses Einbinden sehr aufwendig. NELSON ist zuversichtlich, durch spezielle Algorithmen und Datenstrukturen den Aufwand praktikabel zu halten, verrät aber nicht, wie.<sup>47</sup>

In der möglichen Bedienung unterscheiden sich (gerichtete) Sprunglinks und Transklusionen ebenfalls. Sprunglinks können in einem Browser unter einer graphischen Benutzungsoberfläche folgendermaßen erstellt werden: Zuerst wird das Textstück markiert, von dem der Link ausgehen soll.

47 In der Bibliographie im FAQ zu Xanadu (PAM 96c) wird kommentarlos ein Artikel aufgelistet, der ‚K-Bäume‘ beschreibt (BATES 94). Dabei handelt es sich um eine dynamische Datenstruktur, die der Verwaltung von linearen Datenfolgen dient. Die Adressierung eines Elementes, das Auseinanderschneiden eines Baumes und das Zusammenfügen von zwei Bäumen erfolgt in log arithmischer Zeit. Außerdem werden interne Baumteile durch Operationen nicht geändert, sondern solange aufgehoben, bis sie in k einem Baum mehr enthalten sind (ebd., 26). Zumindest ihren Eigenschaften nach ähneln K-Bäume also der v ermutlich in Xanadu verwendeten Datenstruktur für das Verwalten von Textblöcken und Transklusionen. Eine weitere Analyse würden jedoch den Rahmen dieser Arbeit sprengen.

Dann wird ein Kommando ausgelöst, das ihn als Ausgangsanker definiert. Dann wird entsprechend das Textstück markiert, zu dem der Sprung führen soll und ein Kommando ausgelöst, welches den Link setzt.

Da es sich bei Transklusionen jedoch um dynamische Einfügungen handelt, sollten sie eher wie die statischen Einfügungen in heutigen Textsystemen (,Copy and Paste‘) bedient werden. Bei ihnen wird zuerst das Textstück markiert, das eingefügt werden soll. Dann wird entweder eine Position gewählt oder ein weiteres Textstück markiert. Im ersten Fall wird der Text an der entsprechenden Stelle eingefügt, im zweiten anstelle des markierten Textes.

Stellen wir uns vor, eine Benutzerin würde entweder eine inhaltliche Übernahme aus einem anderen Text durch einen Sprunglink dokumentierten oder eine wörtliche Übernahme als Transklusion vornehmen. Im ersten Fall würde sie zuerst ihre Umformulierung und dann den fremden Text, im zweiten Fall jedoch zuerst den fremden Text und dann eine Position in ihrem Dokument auswählen. Die Reihenfolge der Aktivierung der beiden Dokumente wäre also jeweils unterschiedlich, was bei der Bedienung zu Verwechslungen führen könnte (z.B. das Verwechseln von Ausgangs- und Zielankern beim Setzen von Links).

Noch verwirrt wird die Bedienung von Transklusionen und Sprunglinks dadurch, daß transkludierter Text gleichzeitig dazu benutzt werden kann, an seinen Herkunftsort zu springen, also entgegen die Einbindungsrichtung. NELSON gibt auch für diesen Fall nicht an, wie die Bedienung erfolgen soll. Um die beschriebenen, fortgeschrittenen Linkarten einsetzen zu können, müssen also nicht nur Hypertextsysteme mit entsprechender Funktionalität erstellt, sondern auch praktisch erprobt und immer wieder weiterentwickelt werden (wie in HyperWave zunächst auch geschehen).

#### 3.4.2.4 Technische Realisierung des Xanadu-Netzwerkes

Wie schon beschrieben, soll Xanadu durch ein Client-Server-Netzwerk realisiert werden, in dem ein Client immer nur mit einem Server verbunden ist. Das FEBE (Front End-Back End Language)-Protokoll enthält die Kommandos, die Browser an Server schicken können. Diese Kommandos beschäftigen sich hauptsächlich mit den *Adressen* von Text und Links, von den siebzehn Kommandos von Xanadu Version 1988 dient nur eines der Anforderung von Inhalt. Die anderen dienen dem Einfügen und dem Löschen von Text aus einem Dokument und dem Verschieben von Text in einem Dokument. (Die Kommandos übernehmen also Funktionen, die in andere Hypertextsystemen von den Editierfunktionen von Browsern übernommen werden.) Wieder andere Kommandos befassen sich mit Links. FEBE-Kommandos werden nicht von einer Benutzerin direkt, sondern von ihrem Browser aufgerufen (ebd., 4/61), ähnlich wie im WWW und in HyperWave. Browser können weiterhin nicht auf die physikalische Speicheradresse von Daten zugreifen, sondern nur auf die davon verschiedene Tumbleradresse (ebd., 4/12).

Browser dienen der Darstellung und Veränderung von Daten. Sie müssen Links herausfiltern und selektiv anzeigen können. Sie müssen keine Datenstrukturen für die Textbehandlung verwalten, weil dies vom Server geleistet wird, wie schon beschrieben wurde. Browser können Abhängigkeiten zwischen Text anzeigen (ebd., 4/76f). Die Darstellung von Links, Linkzielen und Rändern von Transklusionen und das Merken von Leseorten (Bookmarks) werden ebenfalls von Browsern durchgeführt (ebd., 2/24ff). Die Unterscheidung von Transklusionslinks ist ebenfalls eine Browserfunktion. Clientprogramme können von Transklusionen, die zeitfest sind, anzeigen, ob sich das ursprüngliche Dokument geändert hat (ebd., 2/37).

Die Grundabfrage von Clients an Server bezieht sich auf Links mit bestimmten Eigenschaften. Die Spannenmenge für eine Anfrage heißt Anfragemenge (Request set). Die Anfragemenge enthält auch die Heimatmenge (Home set), in dem sich die Links befinden sollen. Ein Link erfüllt eine Anfrage, wenn Elemente aus jeder Endmenge den entsprechenden Teil der Anfrage erfüllen, also in einer der entsprechenden Spannen liegen. Es handelt sich also um eine Konjunktion von Disjunk-

tionen. Es können darüberhinaus z.B. alle Links von oder zu einem Dokument oder alle Links eines bestimmten Typs gefunden werden (ebd., 4/58). Diese Operationen werden ‚Suche‘ genannt. Eine Suche nach Text ist in Xanadu zunächst nicht vorgesehen (!) und soll durch ‚Middle-Ends‘ genannte Server erfolgen, die noch erläutert werden.

Weil bei einer Suche nur die Links Suchzeit kosten, die eine Anfrage erfüllen, erfolgt sie in Xanadu effizient, auch wenn das System voll ‚Müll‘ ist. Benutzerinnen können sich aus Effizienzgründen in ‚Teilwelten‘ aufhalten, indem sie ihre Anfragen entsprechend stellen oder von ihrem Browser stellen lassen. Die Heimatmenge ist besonders wichtig für das Herausfiltern, weil dadurch Links nach Herkunft ausgeschlossen werden können (ebd., 4/60). Wie dieses Herausfiltern jedoch auf Serverebene erfolgen soll, damit es die beschriebene Effizienz ermöglichen kann, wird nicht angegeben. Eine Abfrage nach Dokumenten, auf die *zwei* bestimmte Links verweisen, kann nicht Server erfolgen, sondern nur über Clientprogramme, weil sie als kombinatorisch zu aufwendig angesehen wird (ebd., 4/58f).

Browser können den Sprung mittels eines Links zu einem Dokument, von dem eine neuere Version existiert, zu der neuen Version leiten. Außerdem werden die ‚Darstellungsanweisungen‘ (Presentation instructions) für ein Dokument von Browsern interpretiert und können z.B. beeinflussen, ob bei Abfrage eines Dokumentes gleich die neueste Version erscheint (ebd., 2/43). Bitmap-Graphiken sollen von Browser inkrementell dargestellt, z.B. gezoomt werden (ebd., 3/4).

Server kommunizieren über das schon erwähnte BEBE (Back End-Back End)-Protokoll miteinander. Sie leiten darüber Daten untereinander weiter und gleichen Informationen über Daten von anderen Servers ab (‚Subrepresentation‘ und ‚Co-Modelling‘) (ebd., 4/70f). Weiterleiten bezieht sich auf Anfragen von Browsern an andere Server, die sie erfüllen können und die Rückgabe des gesammelten Materials an den Browser (ebd.).

Aus Geschwindigkeits- und Sicherheitsgründen enthalten alle Server eine Unterrepräsentation des ganzen Netzwerkes und modellieren sich gegenseitig mit. Das bedeutet, daß jeder Server eine Abbildung des Netzwerkes mit einigen Daten enthält, er spiegelt also Daten, wie die Bezeichnung im Internet lautet. Über Daten, die ein Xanadu-Server nicht gespeichert hat, enthält er Informationen, von welchem Server er sie bekommen kann (ebd., 4/71f). Das Netzwerkmodell ändert sich ständig, bleibt aber immer benutzbar (ebd.).

Unter Berücksichtigung der Dokumentenzahl des WWW als größtes im Einsatz befindliches Hypertextnetzwerk scheint es kaum möglich, daß Server Informationen über jedes Dokument des Netzes enthalten. Die Xanadu-Gruppe hat keine weiteren Details über das BEBE-Protokoll veröffentlicht. Nach ihrem Konzept soll Server-Software nur von ihr erstellt werden (siehe die Ausführungen im nächsten Abschnitt).

Zusätzlich zu Front-Ends und Back-Ends soll es in Xanadu Middle-Ends geben. Sie stellen im Netzwerk jeweils eigene Server dar und bieten Suchfunktionen (die in den Back-Ends zuviel Leistung und in Browsern zuviel Bandbreite kosten würden). Zusätzlich können sie Material in Xanadu einchecken, welches nicht in Xanadu selbst erstellt wurde, damit es komprimiert abgelegt wird und später die Abhängigkeiten in ihm dargestellt werden können (ebd., 4/72f). Die Middle-Ends werden nicht weiter beschrieben und sollen auch erst nachträglich dem Xanadu-Netzwerk hinzugefügt werden.

Kompatibilitätserwägungen haben bei der Entwicklung von Xanadu kaum eine Rolle gespielt. Das steht im Gegensatz zu NELSON Kritik an „geschlossenen Hypermediasystemen“, die nicht kombiniert und deren Inhalte nicht zusammen betrachtet werden können. Als Gegenbeispiel nennt er Bücher, die sich, wie er sagt, zumindest gegenseitig zitieren und zusammen auf einem Schreibtisch liegen können (87, *Dream Machines*, 67). Nach NELSON sind Dateien dafür verantwortlich, daß Daten nicht zwischen verschiedenen Programmen austauschbar sind. In Xanadu hingegen kann Material zwischen Dateien (sic) geteilt werden (87, *Computer Lib*, 103).

Xanadu soll einen offenen Hypermediaserver bieten, auf dem alle Hypermedien gespeichert werden können (87, *Dream Machines*, ebd.). Tatsächlich ist Xanadu jedoch wegen völlig neuer Datenstrukturen inkompatibel zu herkömmlicher Software. Es muß also entweder ein Dateisystem für Xanadu geschrieben werden, so daß vorhandene Programme darauf zugreifen können, oder neue Programme entwickelt werden (NELSON 87, *Computer Lib*, ebd.).

Kompatibilität soll also nicht durch die Abbildung oder Konversion von Daten erreicht werden, sondern durch das zur Verfügung stellen einer umfassenden Hypermedia-Infrastruktur, auf die verschiedene Programme zurückgreifen können. Das Xanadu-System kann immerhin dazu benutzt werden, Daten aus anderen Systemen oder privates Material, welches z.B. nicht den eigenen Rechner verlassen soll, zu indizieren, zu annotieren, über Links zu verbinden, historische Versionen von ihm zu verwalten etc. Zusammengebracht werden müßten die Daten aus unterschiedlichen Quellen allerdings in speziell angepaßten Front-Ends (ebd., 2/50f). Die Verwaltung von Nur-Lese-Speichern (z.B. CD-ROMs) ist explizit zugelassen (ebd., 4/12).

#### 3.4.2.5 Anwendung von Xanadu

Xanadu soll „all the world's published work“ aufnehmen: Bücher, Zeitschriften, Fotografien, Tonaufnahmen, Filme und Veröffentlichungen neuer Art (NELSON 93, Vorwort). Explizit ist auch die Erstellung von Sekundär- zu wissenschaftlicher Primärliteratur eingeplant (ebd., 2/61).

In Xanadu werden für jedes übertragene Byte Tantiemen berechnet. Diese werden von der Anfragerin der Übertragung an die Eigentümerin des entsprechenden Dokumentes bezahlt. Im System gibt es eine vertraglich festgelegte Standardtantieme, die pro Byte berechnet wird, z.B. mit einem tausendstel Cent. Diese Einheit heißt ‚Nib‘. Die Tantiemen werden der Nutzerin mit den Kosten an die sogenannte Service-Providerin ihres jeweiligen Servers bezahlt und betragen 10-20% der Byte delivery charges, die außerdem Übertragungskosten von den Back-Ends enthalten, jedoch keine Leitungskosten (ebd., 2/43f). In Version 1988 wurde keine Abrechnung von Links festgelegt. Sie könnte einige Nibs pro Link betragen, wobei Links innerhalb von Dokumenten (z.B. für Paragraphen oder typographische Auszeichnungen) frei sein könnten (ebd., 5/13).

Jedes Dokument besitzt in Xanadu einen Tantiemenzähler, die sogenannte Kasse (Cash register). Er wird hochgezählt, wenn Links oder Bytes des Dokumentes über das Netz geschickt werden. Der Zähler ist unbegrenzt und liegt im geschützten Bereich der Benutzerin, wo auch ihr Paßwort und andere Abrechnungsdaten liegen (was wiederum nicht weiter ausgeführt ist). Die Kasse eines Dokumentes kann von der Benutzerin jederzeit abgefragt werden. Von der Kasse werden die durch das Dokument entstandenen Kosten abgezogen, um den Autorinnengewinn übrigzulassen (ebd., 5/12f). Tantiemen für öffentliche Dokumente gehen in den Autorinnenfonds. Aus diesem wird die Eingabe und das Korrekturlesen von öffentlichen Dokumenten und die Unterstützung für Künstlerinnen bezahlt oder andere Zwecke, die die „Gelehrsamkeit und Kreativität fördern“ (ebd.).

Von allen Daten werden Zugriffsrechte verwaltet. Wenn die entsprechende Eigentümerin dies erlaubt, können andere Nutzerinnen auf ihre Daten zugreifen, sowohl lesend als auch durch das Setzen von Links. (Das gemeinsame Schreiben von Dokumenten durch mehrere Benutzerinnen ist also nicht eingeplant.) Dokumente können jedoch auch durch einen formalen Akt veröffentlicht werden. Da mit einer Veröffentlichung die Aufgabe von gewissen Rechten verbunden ist, geschieht sie nicht auf Knopfdruck, sondern erst nach Unterschreiben eines (papiernen!) Vertrages (NELSON 93, 2/42f).

Nach dem Veröffentlichlichen wird ein Dokument in die schon erwähnte (nach Titeln bzw. Autorinnen sortierte) Systemliste aller Dokumente aufgenommen. Die Autorin akzeptiert durch das Veröffentlichlichen weiterhin das Setzen von In-Links auf ihr Dokument und daß sie das Dokument höchstens nach einem langwierigen Prozedere zurücknehmen kann.<sup>48</sup> Sie kann ein veröffentlichtes Dokument jedoch auch durch eine neue Version ersetzen (ebd.). Anonyme Veröffentlichungen sind

erlaubt (ebd., 2/60). Sie werden nur nach hinreichender Begründung entfernt, normale Veröffentlichungen sogar nur nach Übereinkunft oder einem Gerichtsurteil (ebd.).

Benutzerinnen können nicht nur eigene, sondern durch Transklusion auch geänderte Versionen von Werken anderer Autorinnen veröffentlichen. Diese geänderten Versionen können sich auch nur in ein paar Wörtern vom Original unterscheiden. Wenn auf das neue Dokument zugegriffen wird, werden die Tantiemen je nach Anteil auf die jeweiligen Autorinnen verteilt (ebd., 2/45f).

Ein Dokument kann teilveröffentlicht werden, indem der öffentliche Teil als eigenes Dokument veröffentlicht und dieses Dokument in ein vollständig privates weiteres Dokument per Transklusion eingebunden wird. Für historische und alternative Versionen eines Dokumentes kann ausgewählt werden, ob sie mitveröffentlicht werden. Es steht einer Autorin also frei, die ‚Entwicklungsgeschichte‘ eines Dokumentes zu veröffentlichen oder nicht.

Um mehr als die Standardtantiemen zu erhalten, können Eigentümerinnen von Dokumenten diese privat speichern oder verschlüsseln und nur über zusätzliche Zahlungen den Zugriffscode weitergeben (ebd., 2/44). ‚Privazieren‘ (privashing) bedeutet das Setzen von Zugriffsrechten eines Dokumenten, so daß beliebige Xanadu-Benutzerinnen es lesen können. Ein ‚privaziertes‘ Dokument kann jederzeit zurückgezogen und beliebig verändert werden. Dafür können jedoch keine Tantiemen kassiert werden (ebd., 2/44).

Auch in Xanadu soll es neben Autorinnen Verlegerinnen geben. Letztere tragen finanzielle und rechtliche Risiken. Sie bezahlen die Herausgabe und Speicherung von Dokumenten. Urheberrechte und Tantiemen werden zwischen Autorin und Verlegerin außerhalb von Xanadu geregelt (ebd., 2/48). Auch Bücher („a comparatively large unit of writing or anthology, written and publ[i]shed by specific people on specific dates“) und Zeitschriften („a collection of shorter pieces which is regularly edited and published by the same person, and regularly published at a specific time“) werden durch ein System wie Xanadu nicht verschwinden. Sie sind wichtig, weil sie einen stabilen Blick auf die Welt bieten, vorhersehbares Material enthalten und Ideen sammeln und bewerten (ebd., 2/49).

In Xanadu gibt es ein internes ‚Urheberrecht‘, das alle Nutzerinnen des Systems per Unterschrift anerkennen (ebd., 2/42). Es gibt keine Einschränkungen, was Xanadu-Benutzerinnen mit durch das System geliefertem Material machen dürfen (ebd., 2/47f). Wenn Material jedoch auf Papier oder Magnetspeicher übertragen wird, gilt das normale Urheberrecht. Wenn fremdes Material über einen externen Speicher wieder in das System eingebracht wird, werden die Erstellungsdaten der beiden Dokumente verglichen und die beiden Herausgeberinnen aufgefordert, ihre Urheberinnenschaft zu beweisen (ebd., 2/60f).

Wenn ein Xanadu-Dokument auf der lokalen Festplatte einer Nutzerin gespeichert wird, gehen alle Links verloren und Transklusionen werden durch den entsprechenden Text ersetzt. Die Kopie ist „eingefroren“, im Gegensatz zu dem „freifließenden“ Original, welches sich z.B. ändert, wenn eines seiner eingebundenen Teile geändert wird (ebd., 2/48). Hypertext nach NELSON kann also nicht nur schlecht gedruckt werden, wie seine Definition verlangt (s. S. 30), sondern verliert auch beim Speichern auf einem Rechner außerhalb des Netzes.

Auf das Xanadu-System sollen Benutzerinnen zwar hauptsächlich von zu Hause oder dem Büro aus zugreifen, es soll aber auch bestimmte Gebäude mit Xanadu-Terminals geben, ‚Silverstands‘ genannt, eine Mischung aus „Star Trek and McDonald’s hamburger restaurants“. In den Stationen werden neue Benutzerinnen eingewiesen (auch Heimbenutzerinnen), neue Accounts eröffnet, Paßwörter gewechselt (!), Veröffentlichungen vorgenommen und Schulungen abgehalten. Man kann dort drucken, auf Videoband kopieren und scannen.

48 Solch eine Rücknahme muß ein Jahr vorher angekündigt werden und kostet eine Gebühr (ebd., 5/21).

Neben Räumen mit Xanadu-Terminals gibt es in den Stationen Konferenzräume und ein Café (ebd., 5/6). Abgerechnet wird über Debitkarten. In den Stationen arbeiten junge Menschen in Uniformen (wie in McDonald's). In den Gebäuden sind auch die Server für die Einwahl von außen untergebracht. Silverstands sollen in Wohngegenden, Freizeitparks, in der Nähe von Universitäten und entlang von Autobahnen stehen (ebd., 5/6f).

Obwohl auch andere Lizenzen vergeben werden können, soll das Xanadu-Netzwerk hauptsächlich über Franchising wachsen, über ein Modell wieder ähnlich dem von McDonald's: geführt von den Eigentümerinnen selbst, mit Kosten einer ausgebauten Station vergleichbar mit denen für ein Hamburgerrestaurant (1987 \$1 Mio. exklusive technischer Ausstattung). Die Geschäftsführerinnen werden von der Xanadu-Zentrale ausgebildet, die auch das komplette Xanadu-System liefert. Stationen sind für tausend Telefonverbindungen und zweihundert Leute im Gebäude ausgelegt (ebd., 5/8f). Die Lizenznehmerinnen bezahlen für jedes gespeicherte und für jedes an die Kundinnen ausgelieferte Byte eine bestimmte Gebühr an die Xanadu-Zentrale (ebd., 5/15). Die Xanadu-Providerinnen erstellen selbst keine Inhalte im Xanadu-System, nicht einmal Indizes (ebd., 3/6).

Die ersten, prototypischen Xanadu-Stationen sollen nur \$50.000 für Computer und Peripherie kosten (wieder nach Preisen von 1987) und einen Server mit acht Modems bieten. Sie würden sehr technisch aussehen und noch nicht vernetzt sein (!), später möglicherweise über Standleitungen (ebd., 5/9). Die ersten Nutzerinnen von Xanadu werden Wissenschaftlerinnen und Hacker sein „who recognize their own information control needs and will understand at once the potential of xanalogical storage and delivery for their programs, research notes, electronic mail, etc.“ (ebd., 5/3).

Der Aufbau einer Xanadu-Station würde heute keine gute Investition darstellen, falls sie es je gewesen wäre. Der Zugriff auf das WWW funktioniert ohne, wenn man von den Internet-Cafés absieht, die den Wunsch nach Browsen in Gemeinschaft auch erfüllen können. Dem Anspruch nach entsprechen die Xanadu-Stationen Bibliotheken, wenn auch privat finanziert und weniger verstaubt.

Xanadu-Benutzerinnen müssen Gebühren an verschiedene Dienstleisterinnen entrichten. Extern zu dem System sind die für Zugriffsterminals und die Übertragungskosten an ein Telekomunternehmen. An die Eigentümerinnen von Xanadu-Servern gehen die Kosten für die Bytelieferung (inklusive der Tantiemen), Kosten für das Suchen, für die private Speicherung, für eine Veröffentlichung und für den Speicherplatz einer Veröffentlichung (ebd., 2/44).

Die Preise sind Ausdruck der wirtschaftlichen Grundannahme hinter Xanadu, daß alle Beteiligten an dem System verdienen können (Benutzerinnen durch Veröffentlichungen) und daß sich alle Dienste selbst finanzieren (ebd., 4/5). Da das System nicht zwischen Arten von gespeichertem Material unterscheidet, ist Werbung erlaubt. Da diese aber technisch ausgeblendet werden könnte, so NELSON, wird für die Verbreitung des Systems nicht auf Werbung gesetzt (ebd., 4/5).

Die beschriebene Form der Finanzierung erscheint heute nicht mehr realistisch. Das WWW wird z.B. hauptsächlich durch Werbung finanziert. Es gibt allerdings Bestrebungen, die Benutzung von WWW kostenpflichtig zu machen, wie es der Zugriff auf es über einen ‚Service Provider‘ auch schon ist. Es gibt bereits WWW-Inhalte, zu deren Benutzung man ein Paßwort erworben haben muß.

#### 3.4.2.6 Warum wurde Xanadu nicht umgesetzt?

Der Erfolg von Programmen kann daran gemessen werden, daß sie zunächst selbst im technischen Sinn funktionieren und dann v.a. von Nutzerinnen angenommen werden. Diese Kriterien sind auch auf Hypertextsysteme anwendbar. Nach ihnen ist das WWW ein sehr erfolgreiches System, Xanadu ist es hingegen nicht. Dem Entwurf von Xanadu kann höchstens einiger Einfluß zugeschrieben werden, der jedoch in den meisten Fällen nicht belegbar ist.<sup>49</sup>

Die Durchsetzung des Hypertextgedankens erfolgte insgesamt relativ spät. Die Gründe sah CONKLIN 1987 nicht nur darin, daß die nötige Hardware lange nicht verbreitet war, sondern auch in der vorherrschenden Sicht auf Computersysteme als zahlen- und datenverarbeitende Systeme. In dem genannten Jahr seien die Konzepte von Hypertext besser bekannt und Computer als Werkzeuge für die Verarbeitung von Ideen, Wörtern und Symbolen akzeptiert gewesen (32).

Nach NELSON hat die Entwicklung von Xanadu Version 1988 so lange gedauert, weil alle wichtigen Funktionen von Anfang an vorhanden sein und nicht nach und nach eingefügt werden sollten (93, 0/5). Es könne auf keine Funktion des Xanadu-Systems verzichtet werden, um es etwa leichter implementierbar oder auf kleinerer Hardware lauffähig zu machen:

... a hypermedia server without generalized links, transclusion or versioning can perform only small and shallow functions. And electronic publishing *without transclusion* is retrograde. (ebd., Vorwort, Hervorhebung im Original)

Es ist NELSONS Meinung nach nicht möglich, mit einem einfacheren Entwurf eines weltweiten Hypertextsystemes anzufangen und später Funktionen hinzuzufügen (ebd.), wofür er den amerikanischen Fernsehstandard NTSC als schlechtes Beispiel nennt (87, *Dream Machines*, 150). Die Existenz des WWW widerlegt ihn teilweise, weil es Erfolg hat und nach und nach erweitert wird, z.B. um Browserfunktionen (allerdings nicht um alternative Möglichkeiten für die Verwaltung von Links und Hypertexten).

Im Jahr 1988, als Xanadu fertiggestellt werden sollte, hat der damalige Autodesk-Direktor und berühmte Programmierer JOHN WALKER es als ein System ohne Grenzen bezüglich Allgemeingültigkeit, Aufnahmefähigkeit und Erweiterbarkeit beschrieben, was es schwer zu implementieren mache. Es sollte daher zunächst von einem kleineren Xanadu-System ausgegangen werden, das in Abhängigkeit von der Nachfrage weiterentwickelt werden sollte, mit der globalen Bibliothek als Ziel (88a, 418).

Nach dem Ende der Finanzierung von Xanadu durch Autodesk schreibt WALKER, daß es bei den Xanadu-Entwicklerinnen (und bei fast allen Softwaredesignerinnen) einen ‚Designkult‘ gebe, nämlich die Idee, daß mittels Forschung, geistiger Anstrengung, innovativem Management und Tools vollständige Produkte auf den Markt gebracht werden könnten, deren Qualität, Preis, Kundinnen und Ertrag von Anfang an feststünden (93). MEYROWITZ schreibt, daß es auf dem Gebiet der Hypertextforschung insgesamt zu viele Visionen gäbe (91, 310).

RASKIN, der für die Entwicklung des Apple Macintosh und der Canon Cat berühmt wurde, nennt die Benutzungsoberfläche eines Hypertextsystems wichtiger als die dahinterstehenden Datenstrukturen, weil sie zunächst darüber entscheide, ob Benutzerinnen das System annehmen. Das Xanadukonzept spricht hingegen hauptsächlich Datenspeicherung und -verwaltung an (88, 328).

Obwohl diese Kritik grundsätzlich zutrifft, ist NELSON zumindest keine generelle Abneigung gegenüber der Entwicklung von Oberflächen vorzuwerfen. Im Gegenteil, er hat durchaus einen Namen als ‚Human Computer Interface Designer‘ (s. z.B. LAUREL 91). Außerdem hat sich Xanadu aus der Oberfläche des ‚Parallel Textface‘ (s. Fußnote Nr. 33 auf S. 65) entwickelt.

MEYROWITZ unterstützt Nelson in seiner Konzentration auf die Serverentwicklung, indem er als Begründung dafür, daß sich das Wählscheibentelefon so lange gehalten hat, angibt, daß in den achtziger Jahren (in den USA) hauptsächlich in die Telefoninfrastruktur investiert wurde (91, 298).

49 ANDRIES VAN DAM nennt NELSON, der mit ihm in Swarthmore auf dem College war, neben ENGELBART als den Pionier des Hypertexts (88, 887ff). RANDOLL TRIGG, der durch ‚Guided Tours‘ berühmt wurde, gibt an, daß sein Interesse an Hypertext durch die Lektüre von *Computer Lib/Dream Machines* (NELSON 87) geweckt wurde (TRIGG 91). Dieses und *Literary Machines* (NELSON 93) waren laut MEYROWITZ bis 1987 die einzigen beiden Bücher über Hypertext (91). (Er meint damit anscheinend Hypertext im engeren Sinn, ältere Vorläufer wurden ja bereits genannt.)

Er schreibt jedoch auch, daß Hypertextsysteme nicht nur erdacht, sondern auch gebaut und getestet werden müssen (ebd., 310). Es bleibt auf jeden Fall festzuhalten, daß das Projekt Xanadu weder einen Server noch einen Browser hervorgebracht hat. NELSON schreibt daher auch: "In my life I have as yet no real achievements, but I do have a peculiar distinction: a history of thirty-five years in pursuit of a transclusive hypermedia ideal" (95a).

### 3.5 Vergleich von Xanadu, WWW und HyperWave

Xanadu wurde in dieser Arbeit als utopisch-durchdachtes, das WWW als pragmatisch-entwicklungsfähiges Hypertextsystem beschrieben. HyperWave schließlich stellt den empirischen Nachweis dafür dar, daß es möglich ist, ein effizientes, vernetztes Hypertextsystem zu implementieren, welches Links von Dokumenten trennt und dadurch automatisch ihre Konsistenz sicherstellen kann.<sup>50</sup> Diese drei Systeme sollen nun zusammenfassend verglichen werden (s. Tabelle 2), wobei auf die Beschreibungen aus den vorherigen Abschnitten zurückgegriffen wird.

Das WWW und HyperWave basieren auf dem Internet, bei Xanadu stellt es (eigentlich das ARPAnet) nur eines der verschiedenen Zugriffsmöglichkeiten dar. Alle drei Systeme setzen das Client-Server-Konzept um. Zu diesem gibt es z.Z. keine Alternative, auch wenn Net Computer als ‚Thin Clients‘ in einigen Bereichen, z.B. Unternehmen mit einer großen Anzahl von Computern, die ‚Fat Clients‘ in Form heutiger PCs verdrängen und damit das Gewicht auf Server verschieben könnten (s. Fußnote Nr. 40 auf S. 73).

Bezüglich der Hierarchieeigenschaft eines Hypertextsystems muß zwischen der Verwaltung und Benennung von Daten unterschieden werden. Davon unabhängig ist die Frage nach der zentralen Verwaltung des Netzes. Ein Netz wird zentral verwaltet, wenn ein neuer Server bei einer Zentrale angemeldet werden muß. Bei Xanadu sollten sogar die Server komplett von ‚Xanadu Central‘ geliefert werden, womit eine Registrierung verbunden sein sollte. Bei HyperWave werden immerhin noch Servernummern verwandt, die nicht mit den IP-Adressen der entsprechenden Computer übereinstimmen und daher bei einer Zentrale angemeldet und auf Eindeutigkeit überprüft werden müssen.

Ein WWW-Server kann in das Internet eingefügt werden, ohne daß (außer auf der Ebene der lokalen Rechnerverwaltung) eine zentrale Instanz oder Verwaltungen anderer Rechner benachrichtigt werden müssen.<sup>51</sup> Damit ein Server über einen Namen und nicht nur eine IP (Internet Protocol)-Nummer angesprochen werden kann, muß er in das DNS (Domain Name Service)-System des Internet eingebunden werden. Das DNS basiert auf einer gestaffelt hierarchischen Verwaltung. Da ein WWW-Server gewöhnlich in das bestehende Netz einer Institution eingebunden wird, werden die nötigen Anmeldungen seines Namens auch in dieser vorgenommen. Für einen Server mit einem neuen Domainnamen muß dieser jedoch bei einer Zentrale beantragt werden.

Daten werden hierarchisch verwaltet, wenn sie in Verzeichnissen gespeichert sind, die wiederum weitere Verzeichnisse enthalten können. Daten werden hierarchisch benannt, wenn sich ihre Namen aus der Folge der Verzeichnisse ergeben, die ‚durchwandert‘ werden müssen, um zu einer Datei zu gelangen. Im WWW werden Daten hierarchisch verwaltet, weil Dokumente als Dateien in einem hierarchischen Dateisystem gespeichert sind und sich daher immer in einem Verzeichnis befinden. Dokumente besitzen im WWW auch einen hierarchischen Pfadnamen. Dieser besteht aus den Namen der Verzeichnisse, die durchlaufen werden müssen, um zu der Datei zu gelangen und aus einem Dateinamen. Die hierarchische Dateiverwaltung im WWW ist nach dem UNIX-Dateisystem modelliert,

<sup>50</sup> wenn auch vielleicht zu kleinen Umfängen.

<sup>51</sup> Wie Benutzerinnen von seiner Existenz in Kenntnis gesetzt werden, ist wieder eine andere Frage. Wenn in der betreffenden Institution, z.B. einer Firma oder Universität bereits Server existieren, können von dort aus Links auf den neuen gesetzt werden.



Aspekt	WWW	HyperWave	Xanadu
hierarchische Datenverwaltung	ja	ja	nein
Struktursicht	nein	ja	?
hierarchische Benennung	ja	nein	ja
Servertail in Adresse	ja	ja	ja
Benennung über Namen	ja	nein	nein
eindeutige, ewige IDs	nein	ja	ja
‚ewiges‘ Aufheben von Dokumenten	nein	nein	ja
verschiedene Sprachen	ja	ja	ja
Versionsverwaltung	nein	nein	ja
Eigentumsverwaltung	tlw.	ja	ja
Abrechnung	nein	ja	ja
Texterstellung im System	nein	nein	ja
Offline-Benutzung	tlw.	tlw.	nein
Transklusion	(nur Bilder)	(nur Bilder)	ja
Anmerkungen	nein	ja	ja
Stretchtext	nein	nein	ja
getypte Links	nein	nein	ja
Links von/auf Dokumententeile	tlw.	ja	ja
unzusammenhängende Linkenden	nein	nein	ja
dokumentenübergreifende Suche	Zusatz	ja	nein
Sitzungsorientierung	nein	ja	ja
Metainformationen zu Dokumenten	tlw.	ja	ja
Kompatibilität zu anderen Internetdiensten	ja	tlw.	nein
Zugriff auf vorhandene Datenbanken	ja	ja	nein
zentrale Netzverwaltung	nein	ja	ja
Caching	ja	ja	nein
Mirroring	nein	nein	ja
Linkdatenbank	nein	ja	ja
kleiner Server	ja	nein	nein
Server-Server-Protokoll	nein	ja	ja

Tabelle 2. Vergleich von Transactions mit dem WWW, HyperWave und Xanadu

das in vielen Servern auch tatsächlich die technische Basis bildet. Wie schon erwähnt, stellen Verzeichnisse im WWW bedingt auch eigenständige Namen von Objekten dar.

Daß die Eigenschaften der Verwaltung und Benennung tatsächlich orthogonal sind, zeigt sich im Vergleich von HyperWave und Xanadu. In HyperWave muß sich jedes Objekt in einer Collection, also in einem Verzeichnis befinden. Daten werden in HyperWave also hierarchisch verwaltet. Der Name eines Objektes besteht jedoch aus einer für seinen Server eindeutigen Nummer. Eine globale Adresse ergibt sich zusammen mit einer Servernummer. Namen sind also nicht hierarchisch (wenn sie auch aus zwei Teilen bestehen).

In Xanadu gibt es keine Verzeichnisse, in denen sich ein Objekt befinden könnte (außer den beiden systemweiten Listen nach Autorinnen und Titeln), Objekte werden also nichthierarchisch verwal-

tet. Daten werden wie in HyperWave über Nummern adressiert, allerdings nicht wie dort über Nummernpaare, sondern hierarchische Nummernketten. Objekte, deren vorderer Adreßteil übereinstimmt, können über dieses Präfix gemeinsam adressiert werden. Dadurch stellen solche Teiladressen Pseudoverzeichnisse dar, die ein Objekt nicht wechseln kann, weil Objekte und Adressen in Xanadu nicht gelöscht werden können. Daten werden in Xanadu also hierarchisch benannt.

Benutzerinnen können in Xanadu Verzeichnisse durch das Sammeln von Links in Dokumenten zusammenstellen. In solchen Dokumenten können im Gegensatz zu Verzeichnissen in HyperWave weiterer Text, z.B. Anmerkungen, eingefügt werden. Im WWW ist es auch möglich, Verzeichnisse auf der Basis von Dokumenten selbst zu erstellen, was für die Organisation von WWW-Verzeichnissen (wie z.B. Yahoo) auch ausgenutzt wird. Solche selbsterstellten, möglicherweise hierarchischen Verzeichnisse konkurrieren im WWW jedoch mit den Dateiverzeichnissen, in denen sich die verzeichneten Hypertextdokumente befinden. Über Verzeichnisse oder Links erstellte Strukturen können völlig unabhängig voneinander sein. Xanadu verzichtet daher auch auf die explizite Organisation von Dokumenten in hierarchischen Verzeichnissen, damit nur eine Art von Ordnungsstruktur existiert.

Die Adressen im WWW, in HyperWave und Xanadu bestehen jeweils aus einem Server- und einem Objektteil. Diese Aufteilung verleiht auch HyperWave-Adressen einen gewissen hierarchischen Aspekt, beim WWW und Xanadu ist jedoch der Name eines Objektes auch ohne den Servernamen hierarchisch. Die besondere Behandlung eines Serverabschnitts in Xanadu-Adressen ist zwar teilweise nützlich für die Organisation von Daten, erklärt jedoch nicht die Aufteilung von Adressen in zwei Hälften beim WWW und HyperWave. Sie ist auf die technische Organisation der beiden Hypertextnetzwerke zurückzuführen, bei denen scharf zwischen dem Heimatserver eines Objektes und seiner Lage auf ihm unterschieden wird.

In Xanadu ist der Serverabschnitt einer Adresse nur ein Feld von mehreren, auch wenn er am Anfang steht. Er bezeichnet nicht den Speicherort eines Objektes, sondern seinen Ursprungsort. Xanadu-Objekte werden nämlich zwischen Servern kopiert und jeweils vom nächstliegenden Server angefordert, wenn ein Server sie nicht vorrätig hat. Daten besitzen, so deutet NELSON an, eine interne Adresse, die sich von ihrem jeweiligen Tumbler unterscheidet und wahrscheinlich den Speicherort eines Objektes angibt. Zu einem Tumbler gibt es also mehrere interne Adressen.

Unabhängig vom Ort gibt es in Xanadu (auf externer Ebene) und in HyperWave eine eindeutige, ewige Adresse für ein Objekt. Da die Adresse eines Objektes im WWW davon abhängig ist, in welchem Verzeichnis es sich befindet, kann sie sich leicht ändern. In Xanadu kann ein Dokument nicht gelöscht werden, in HyperWave und im WWW ist dies jedoch möglich. Der Verzicht auf Löschung hat den Vorteil, daß ein Dokument immer adressiert werden kann. Bei der externen Linkverwaltung in Xanadu und HyperWave kann allerdings technisch festgestellt werden, ob ein Link auf ein Dokument existiert, und eine Löschoperation nur dann ausgeführt werden, wenn dies nicht mehr der Fall ist. Das Verschwinden von Dokumenten, auf denen sich Links befinden, kann also einfach verhindert werden, ohne auf das Freimachen von Speicher zu verzichten.

Im WWW werden Objekte über Namen, in HyperWave und in Xanadu über Nummern adressiert. Es wurde bereits erwähnt, daß Nummern zwar für Computer leichter zu handhaben sind, nicht jedoch für Menschen. Nummern haben den Vorteil, daß sie maschinell eindeutig erzeugt werden können. Ihre Zulassung als Namen oder das Ableiten weiterer Namen aus einem vorhandenen durch Anhängen einer Nummer scheint jedoch sinnvoll, ebenfalls die Verwendung von Nummern für die Adressierung von Teildokumenten und Versionen. In allen drei Systemen gibt es neben den eigentlichen Adressen Titel für Dokumente. Darüber können Dokumente meist erkannt und aus Listen ausgewählt werden, wenn es zusätzlich Hinweise auf einen Kontext gibt, z.B. die Autorin.

In allen drei Systemen werden Sprachen von Dokumenten unterschieden. In Xanadu existiert ein spezieller Link, der auf Übersetzungen eines Dokumentes verweisen kann. In HyperWave- und

WWW-Browser kann eine Sprachprioritätenliste für Dokumente eingestellt werden. Diese Liste wird Servern bei der Abfrage von Dokumenten mitgeteilt, die entsprechend reagieren, wenn sie verschiedene Sprachvarianten eines Dokumentes zur Verfügung haben. Die Sprache an oberster Stelle wird in HyperWave-Browsern auch für die Benutzungsoberfläche übernommen.

Nur in Xanadu ist eine Versionsverwaltung eingebaut. Versionsnummern werden an den entsprechenden Adreßtumbler eines Dokumentes angehängt, so daß alle Versionen des Dokumentes gemeinsam durch Weglassen dieser Nummern adressiert werden können. HyperWave und Xanadu besitzen eine Eigentumsverwaltung. Im WWW können einzelnen Dokumenten keine Eigentümerinnen zugewiesen werden, es existiert nicht einmal ein Attribut, welches über die Autorinnenschaft Auskunft gibt. Über Servereinstellungen ist es allerdings möglich, Zugriffsrechte zu vergeben. In Xanadu wird die Übertragung von Daten nach Umfang und unabhängig von der Datenart abgerechnet. In HyperWave war eine Abrechnung je Dokument von Anfang an vorgesehen und ist anscheinend inzwischen implementiert. Im WWW gibt es gar keine Abrechnung.

In Xanadu und HyperWave gibt es Links von und auf Dokumententeile, weiterhin Anmerkungen und getypte Links. In Xanadu können Linktypen jedoch von fortgeschrittenen Benutzerinnen selbst definiert werden, außerdem können Links auf unzusammenhängende Dokumententeile zeigen (die sich nicht einmal im selben Dokument befinden müssen).

Es ist im WWW möglich, Linkanker als Sprungziele zu einem Punkt innerhalb eines Dokumentes zu definieren, deren Aufruf dazu führt, daß das entsprechende Dokument so gescrollt wird, daß der Punkt am Fensteranfang erscheint. Ein Anker kann jedoch nur von der Autorin eines Dokumentes eingefügt werden, weil er wie ein Link im Dokumententext steht. Neben dem Sprunglink gibt es im WWW keine weiteren Linkarten.

Im WWW werden keine Anmerkungen unterstützt, und Links befinden sich immer in Dokumenten. In HyperWave werden Links (auf der technischen Ebene) hingegen getrennt von den entsprechenden Dokumenten verwaltet. In den Beschreibungen zu Xanadu gibt es dazu keine Aussage, die Möglichkeit der Abfrage aller ‚In-Links‘ eines Dokumentes beruht aber wahrscheinlich ebenfalls auf einer derartigen Trennung. Der Verzicht auf einen Linkserver im WWW ist Hauptgrund dafür, daß seine Server prinzipiell relativ hardwaregenügsam sind, im Gegensatz zu den Servern der anderen beiden Systeme. In Xanadu und HyperWave werden Daten auch zwischen Servern ausgetauscht, im WWW nicht.

Das Einbinden von Grafiken kann im WWW und in HyperWave als Transklusion betrachtet werden, weil in dem entsprechenden WWW-Dokument nur die Adresse (URL) der Grafik angegeben wird, die dann der Browser bei Darstellung des Gesamtdokumentes anfordert. Andererseits sollen ihrem Konzept nach Transklusionen hauptsächlich Ausschnitte von Texten zusammenstellen. Dies erlaubt nur Xanadu. In HyperWave können immerhin Links auf beliebige Objekte und ihre Teile gelegt werden, auch von anderen Nutzerinnen als der Autorin. Das WWW und HyperWave gestatten das Zusammenstellen von Dokumenten nur über klassisches ‚Copy and Paste‘ und einen Editor.

In Xanadu können Dokumente in Browsern editiert werden. Es sind Textverarbeitungsfunktionen in die Server eingebaut, die damit selbst einzelne Zeichenketten manipulieren, was mit der heutigen Technik zu wenig effizient für die praktische Arbeit sein dürfte. Die Texterstellung geschieht im WWW und in HyperWave in Programmen, die extern zu den eigentlichen Browsern sind. Für das WWW gibt es allerdings immer mehr kombinierte Browser/Editoren.

In Editoren oder Serververwaltungsprogrammen können HTML-Dateien zusammen mit möglichen Links zwischen ihnen betrachtet und verschoben werden. Zum Ordnen von Material im wissenschaftlichen Schreibprozeß eignen sie sich aber nicht. In HyperWave ist in Browsern das Verschieben innerhalb von Kollektionen möglich. In dem System können Browser standardmäßig Daten an Server übertragen, die mit anderen Programmen erstellt wurden. Dem ‚Harmony‘-

HyperWave-Browser für Microsoft-Windows liegt eine Datenbank bei, in der Daten offline organisiert und online zusammen an einen Server übertragen werden können.

In Xanadu können Texte und Textausschnitte veröffentlicht werden, inklusive der Aufnahme in einem Verzeichnis. Dazu gibt es ein offizielles Ritual. Im WWW und HyperWave sind das Erstellen und Publizieren von Dokumenten technisch getrennt, weil das Erstellen extern zu dem jeweiligen System geschieht und das Einbringen in einen Server als Publizieren verstanden werden kann. Das Veröffentlichen in Auszügen wird im WWW und HyperWave jedoch generell nicht unterstützt. Ob das Veröffentlichen fremden Textes in eigenem Text mittels Transklusion und die dazugehörige Abrechnung in Xanadu den Anforderungen entspricht, die Autorinnen und Leserinnen an ein digitales Veröffentlichungssystem stellen, wird von manchen Autorinnen bezweifelt (s. z.B. SAMUELSON U. GLUSHKO 91).<sup>52</sup>

Für öffentliche Verzeichnisse gilt das bereits über private Zusammenstellungen gesagte: In Xanadu ist es uneingeschränkt, in den anderen beiden Systemen nur bedingt möglich. Das Veröffentlichen wird in Xanadu nicht nur technisch, sondern auch organisatorisch (und wenn man will: sozial) unterstützt. Dazu gibt es im Umfeld von WWW und HyperWave Überlegungen, aber auch erste praktische Projekte (s. z.B. DENNING U. ROUS 95 und ENDRES 95).

In Xanadu und HyperWave werden Anmerkungen durch einen speziellen Link unterstützt. Dieser Link kann sowohl in eigenen als auch fremden Dokumenten angebracht werden. Obwohl mit dem ersten erfolgreichen graphischen Browser für das WWW, Mosaic vom NCSA, sowohl private als auch öffentliche Anmerkungen an beliebigen Dokumenten angebracht werden können (LALIBERTE U. BRAVERMAN 95), wird dies von den meisten heute gebrauchten WWW-Browsern und -Servern nicht unterstützt.

In Xanadu und HyperWave beinhaltet jeder Server einen Cache für Daten aus anderen Servern. Beim WWW ist das nicht so, es werden jedoch immer mehr sogenannte Proxy-Caches eingesetzt, die gegenüber anderen Servern wie ein Browser auftreten und Daten für anfragende Browser speichern und weiterleiten. Browser müssen für die Benutzung von Cache-Servern vorbereitet sein, was aber bei den neueren Browsern der Fall ist. Proxy-Caches werden häufig in eigenen Netzen zusammengeschlossen, so daß die Beantwortung einer Anfrage dem Cache überlassen wird, der gerade am wenigsten beansprucht ist.

In Xanadu können zunächst nur Dokumente nach verschiedenen Links gefiltert werden, z.B. nach Autorinnen. Volltextsuche soll über spezielle Server, die ‚Middle-Ends‘, umgesetzt werden. Im WWW kann über mehrere Server ebenfalls nur per speziellen ‚Suchmaschinen‘ gesucht werden. Diese bestehen gewöhnlich aus einem Indizierungs- (einem ‚Roboter‘) und einem Datenbankteil. HyperWave erlaubt standardmäßig die Suche in Attributen von Objekten und dem Text von Dokumenten, die in HyperWave selbst gespeichert sind. Es ist jedoch wahrscheinlich, daß in einem HyperWave-Netz von der Größe des aktuellen WWWs ebenfalls nur noch mittels dedizierter Server gesucht werden kann, mit viel Speicher und schnellen Verbindungen.

In Xanadu und HyperWave ist ein Client für eine Sitzung direkt mit einem Server in bandbreitenmetrischer Nähe verbunden. Auf andere Server greift ein Browser nur mittelbar über den Heimatserver zu. Über die Server in Xanadu und HyperWave ist es Benutzerinnen außerdem möglich, Dokumente zu veröffentlichen, auf die andere Benutzerinnen ständig zugreifen können, weil die entsprechenden Server immer mit dem Netz verbunden sind.

Im WWW gibt es keine Sitzungen, Clients verbinden sich mit verschiedenen Servern für jeweils einen Zugriff. WWW-Browser sind also autonomer als die in den anderen Systemen. Andererseits befindet sich in der Nähe jedes WWW-Browsers gewöhnlich auch ein Server: Universitäten und

<sup>52</sup> Die „ACM Interim Copyright Policies“ legen für Veröffentlichungen der ACM eindeutig fest, daß Einbindungen von (im Gegensatz zu Sprunglinks auf) Material, das dem Urheberrecht unterliegt, nur mit Zustimmung der entsprechenden Autorin und einem Urheberrechtsvermerk gestattet sind (DENNING U. ROUS 94, 104).

Firmen haben meist eigene, oft auch Proxy- und Cache-Server. Private Nutzerinnen wählen sich über kommerzielle Service-Provider sowieso in einen Server in ein Netzwerk ein, meist erfolgt dann auch der Zugriff auf WWW-Server nicht direkt, sondern wiederum über Proxy-Server.

In Xanadu wird der Zugriff auf bestehende Daten nur durch spezielle Browser unterstützt, die sowohl auf einen Xanadu- als auch auf einen Server mit den externen Daten Zugriff haben müssen. Im WWW können beliebige Daten gespeichert und übertragen werden. Browser erlauben die Einbindung von Viewern für andere Formate als das Standardtextformat HTML, z.B. PostScript, ohne jedoch in den fremden Dokumenten das Einfügen von Links zu erlauben. In HyperWave hingegen werden Formate wie PostScript durch spezielle Programme nicht nur zum Betrachten unterstützt, sondern es können in den entsprechenden Dateien auch Links eingefügt werden.

Im WWW war es von Anfang an vorgesehen, daß Server als Gateways für bestehende Datenbankserver dienen und deren Daten für die Darstellung in WWW-Browsern konvertieren können. In HyperWave scheint eine ähnliche Funktion inzwischen auch eingebaut zu sein. In Xanadu hingegen soll das Zusammenführen von bestehenden Daten und Verwaltungsinformationen aus dem Hypertextsystem in speziell angepaßten Browsern erfolgen. Eine solche Anpassung ist jedoch aufwendig und führt zu inkompatiblen Browsern, die für die Erledigung jeweils einer Spezialfunktion erweitert wurden, wie bei experimentellen Erweiterungen von NCSA Mosaic zu beobachten.

Xanadu, das WWW und HyperWave sollen nun nach diesen technischen Aspekten bezüglich ihrer Eignung für den wissenschaftlichen Publikationsprozeß anhand der in Abschnitt 2.5.2 auf S. 27 formulierten Anforderungen verglichen werden. Xanadu hat es als über viele Jahre konzipiertes, jedoch nicht implementiertes System selbstredend leichter, umfangreiche Anforderungen zumindest auf dem Papier zu erfüllen.

Die vor den tatsächlichen Umsetzungen des WWW und HyperWave formulierten Visionen bezüglich ihrer Leistungsfähigkeit kommen einem idealen wissenschaftlichen Publikationssystem ebenfalls wesentlich näher als die tatsächlichen Systeme. Allerdings bestand für die beiden letztgenannten Systeme auch die Möglichkeit, sie nach ihrer Einführung weiterzuentwickeln. Bei HyperWave war dies allein deshalb nötig, weil die Trennung von Links und Dokumenten nur durchgehalten werden konnte, indem die Server und die Verbindungen zwischen ihnen immer ausgefeilter und komplexer wurden. Das hat u.a. den Nachteil, daß die Hardware-Anforderungen für den Einsatz eines HyperWave-Servers relativ hoch sind.

Über Bedienungsfreundlichkeit bzw. Einarbeitungszeit und Antwortverhalten von Xanadu kann keine Aussage gemacht werden. WWW- und HyperWave-Browsern kann eine hohe Benutzungsfreundlichkeit attestiert werden, wenn sie auch beim WWW zumindest teilweise auf die einfache Struktur des Systems zurückzuführen ist.<sup>53</sup> Für die Erstellung von Texten ist bei WWW und HyperWave im Vergleich zu Textverarbeitungsprogrammen eine relativ hohe Einarbeitungszeit nötig, weil der jeweilige SGML-Dialekt gelernt werden muß.

Die Übertragung von Daten dauert im WWW und in HyperWave teilweise bis zur Unbenutzbarkeit lange. Inwieweit dies auf die Systemkonzepte oder das Internet als benutzte Infrastruktur zurückzuführen ist und z.B. durch größere Cache-Systeme aufgefangen werden kann, kann in dieser Arbeit nicht geklärt werden. Ebenso wenig, ob die umfangreiche Datenmanipulation in den Xanadu-Servern überhaupt schnell genug für produktives Arbeiten erfolgen kann.

Im nächsten Kapitel werden die Anforderungen an ein Publikationssystem für das wissenschaftliche Schreiben auf Basis des Hypertextkonzeptes weiterentwickelt, das in diesem Kapitel an Beispielen beschrieben wurde. Danach wird ein solches Publikationssystem spezifiziert und auf Basis des WWW entworfen.

53 und bei neueren Funktionen wie Frames nur noch bedingt gegeben ist.

## 4 Ein wissenschaftliches Hypertext-Publikationssystem

Wie wir im letzten Kapitel gesehen haben, können mit dem Computer Texte sowohl flexibler geschrieben als auch gelesen werden, als mit heutigen Textverarbeitungsprogrammen möglich. Der Computer erlaubt Veröffentlichungen, in denen auf verschiedene Arten gesprungen werden kann, also Hypertexte. In diesem Kapitel wird ein verteiltes Hypertext-Publikationssystem auf Basis des Internets spezifiziert. Zuerst wird seine Benutzung skizziert. Nach der Systemspezifikation folgt am Ende des Kapitels ein Entwurfsansatz auf Basis des WWW.

### 4.1 Ein Anwendungsszenario

Obwohl NELSON immer wieder betont, wie nützlich ein Hypertextsystem für das Erstellen und Publizieren von Texten ist, wird aus seinen Beschreibungen nicht deutlich, wie es konkret dazu eingesetzt bzw. generell bedient werden kann. In diesem Abschnitt soll daher ein Szenario beschrieben werden, in dem eine Informatikstudentin mit dem Namen Christine mit einem Hypertext-Publikationssystem, das weiter unten spezifiziert wird, einen Aufsatz schreiben und publizieren kann. Dafür werden zwar in einigen Fällen spezielle Annahmen gemacht, generell sollen die in dem Szenario ausgedrückten Vorgänge jedoch exemplarischen Charakter haben, also auf andere Benutzungsvorgänge übertragbar sein.

Das System besteht aus Clients (Browsern) und Servern. Daten können nicht nur in Servern, sondern auch in einem privaten Bereich einer Benutzerin auf ihrem Arbeitsplatzrechner gespeichert werden, ‚Ablage‘ genannt. Es wird davon ausgegangen, daß Dokumente in den meisten Fällen in einer Ablage erstellt werden. Es gibt Abteilungsserver und solche, die von Bibliotheken und Verlagen verwaltet werden und Veröffentlichungen enthalten. Dokumente können auf Abteilungsservern erstellt werden, damit mehrere Personen gemeinsam an ihnen arbeiten können. Die Eigentümerin eines Dokuments kann es auch so sperren, daß es von anderen Benutzerinnen nur gelesen werden kann. Wenn ein Dokument auf einem Veröffentlichungsserver liegt, kann selbst seine Autorin es nicht mehr ändern.

Die Beispielbenutzerin Christine hat ein Thema für ihren Aufsatz vorgegeben. Sie kann das Thema jedoch noch nicht vollständig einordnen und muß es außerdem konkretisieren. Da Christine an ihrem Studiengang als wissenschaftliche Hilfskraft beschäftigt ist, hat sie einen eigenen Schreibtisch mit einem Personal Computer darauf zur Verfügung. Sie meldet sich über den Computer auf dem Hypertextserver des Studienganges an, indem sie ihren Benutzerinnennamen und das dazugehörige Paßwort eingibt.

Sie bekommt als erstes das Hauptdokument ihrer privaten Ablage präsentiert. Die Ablage enthält Verweise auf Dokumente in dem Netzwerk, Ausschnitte von Dokumenten und Kommentare dazu und fertige und in Arbeit befindliche Texte von ihr selbst, die sie für das Studium oder ihren Job schreibt. Das Hauptdokument enthält Notizen über Aufgaben, die sie noch zu erledigen hat. Als Christine ihr Account eingerichtet bekommen hatte, enthielt die Hauptseite eine Einführung in das System und Informationen zur Systemhilfe. Das Hauptdokument enthält außerdem einen Link auf eine Systembeschreibung, auf die Universitätsbibliothek und auf einige andere elektronische Bibliotheken, die sich schwerpunktmäßig mit Informatik befassen.

Christine hat sich ihre Ablage selbst organisiert. Zunächst hat sie sich drei Dokumente mit den Namen ‚Studium‘, ‚Job‘ und ‚privat‘ angelegt, in denen nur sie lesen und schreiben kann. Dazu hat sie ein Dokument mit dem Namen ‚öffentlich‘ erzeugt, in dem alle Benutzerinnen im Studiengang lesen können, und eines mit dem Namen ‚Zusammenarbeit‘, in dem andere Studentinnen auch schreiben können. Diese Dokumente dienen als Verzeichnisse von anderen, sie bestehen also haupt-

sächlich aus Namen von Dokumenten, wobei von jedem Namen aus ein Link auf das entsprechende Dokument gelegt ist.

Einige Verzeichnisse in der Ablage besitzen solche Benutzungsrechte, daß sie von Mitgliedern von Gruppen, die sich Studentinnen selbst anlegen können und die gewöhnlich mit Arbeitsgruppen zusammenfallen, gelesen und beschrieben werden können. Christine ist selbst Mitglied in verschiedenen Arbeitsgruppen, in denen sie im Laufe ihres Studiums mehrere Referate verfaßt hat.

Zur Erstellung des Referats sucht Christine zuerst in ihrer eigenen Ablage nach Informationen zum Thema. Sie hat jedoch keine vorliegen. Die Professorin, zu deren Vorlesung Christine den Aufsatz schreibt, hat veranlaßt, daß ein kommentiertes Literaturverzeichnis zu dem Thema der Vorlesung zusammengestellt wurde. Das Verzeichnis enthält Links auf weitere Verzeichnisse (Dokumente, die wieder hauptsächlich aus Links bestehen) und auf Dokumente, die sich entweder im System oder der lokalen Bibliothek befinden. Es wird u.a. auf ein Verzeichnis verwiesen, welches Übersichtsartikel zur Informatik enthält. Es wurde aus verschiedenen Verzeichnissen und Dokumenten zusammengestellt. In ihm liest sich Christine in das Thema ein, bis sie ihre Aufgabe versteht.

Die Übersichtsartikel zeigen nach dem Laden (neben Titel und Autorin) nur eine Zusammenfassung und eine Inhaltsübersicht. Diese beiden Elemente können zu dem vollständigen Text bzw. Inhaltsverzeichnis erweitert werden. Artikel enthalten teilweise Anmerkungen von anderen Benutzerinnen des Hypertextsystems. Diese Anmerkungen kann Christine abschalten, wenn sie sie zu sehr von den eigentlichen Texten ablenken.

In Artikeln wird auf andere verwiesen, von denen Christine einige ebenfalls mit ihrem Browser anfordert. Bei einem besonders interessanten Artikel läßt sich Christine außerdem das Linkkontextfenster anzeigen. Es stellt nicht nur die Links dar, die von dem Artikel ausgehen, sondern auch die, die auf es zeigen. Christine wählt als Umgebung alle Dokumente mit Links auf das Dokument und auf Dokumente, die auf es zeigen, also zwei Ebenen von In-Links (s. S. 74). Sie kann Dokumente anfordern, indem sie in der Übersicht auf sie klickt. Die Dokumente werden dann in das aktuelle Fenster geladen und der ursprüngliche Inhalt ersetzt, falls er nicht geändert wurde. Das Übersichtsfenster bleibt, es wird jedoch so erneuert, so daß nun das neue Dokument in der Mitte steht.

Christine speichert Links auf Dokumente, Zitate und andere Dokumentenausschnitte in ein eigenes Dokument in ihrer Ablage. Sie überträgt die Ausschnitte mit ‚Copy and Paste‘, ihr Browser fügt dabei automatisch Links zu den Ausgangsdokumenten ein, damit sie sich keine Adressen von Quellen notieren muß. Nach dem Stöbern schreibt sie jeweils eine Überschrift und einen kleinen Kommentar zu den einzelnen Abschnitten. Wenn sie Text aus anderen Dokumenten nur inhaltlich statt wörtlich übernimmt, erzeugt sie einen Link auf den Ursprungstext.

In Abhängigkeit vom Thema, das sie inzwischen weit genug eingegrenzt hat, löscht Christine einige Notizen, die ihr nun nicht mehr relevant erscheinen. Die anderen liest sie sich genauer durch. Sie achtet darauf, ob Artikel in ein Verzeichnis lesenswerter Literatur eines wissenschaftlichen Instituts oder einer entsprechenden Organisation aufgenommen sind oder als ‚richtige‘ Veröffentlichungen auf einem Server eines Verlags liegen, was Christine wieder über das Linkkontextfenster feststellen kann. Sie fügt an einigen Stellen in Dokumenten private Anmerkungen ein, die also nur für sie sichtbar sind und in ihrer Ablage gespeichert werden. Sie sichert weitere Textpositionen durch Links in ihrem Ablagedokument, zu denen sie Kommentare schreibt. An das Ende ihres Dokuments schreibt sie eigene Ideen, die ihr beim Lesen der Texte kommen.

Christine ordnet die Textausschnitte in der Gliederungssicht ihres Browsers, immer noch mit dem Thema im Hinterkopf. Sie experimentiert mit verschiedenen Gliederungen, indem sie Varianten der ersten Gliederung erzeugt. Die Varianten enthalten zunächst dieselben Daten wie die Ausgangsversion. Sie können dann in der Gliederungsansicht überarbeitet werden, indem Gliederungspunkte als Kategorien erzeugt und verschoben werden. Einige Überschriften auf der untersten

Ebene und die dazugehörigen Absätze (die zusammen jeweils einer Karteikarte beim Papiermodell entsprechen) ordnet Christine in mehreren Abschnitten ein, indem sie sie im selben Dokument kopiert.

Nachdem sie ihr Material gegliedert hat, besitzt Christine eine konkrete Vorstellung über die Argumentationslinie ihrer Arbeit. Sie verfaßt daher durch Weiterentwicklung des Themas eine Einleitung, die sie im weiteren Verlauf der Arbeit immer wieder überarbeitet, indem sie jeweils eine neue Version erstellt, wenn die alte nicht mehr dem aktuellen Stand ihrer Arbeit entspricht. Um ein Zettelmanuskript zu erstellen, erzeugt sie aus der Gliederung, die ihr am besten erscheint, eine neue Version. Sie fügt dazu zwischen die schon vorhandenen Absätze neue ein.

Ein Rohmanuskript entsteht auch als neue Version, indem der Text des Zettelmanuskriptes überarbeitet, u.a. geglättet wird. Quellenverweise müssen nicht gesondert eingefügt werden, weil sie noch als Links von der Übernahme des fremden Textes vorhanden sind. Um sie von Übernahmen aus Texten von derselben Autorin wie der jeweils vorliegende Text zu unterscheiden, deren Hervorhebung im Browser ja abgeschaltet werden kann, setzt Christine Zitate in Anführungszeichen und kennzeichnet Auslassungen und Einfügungen (was für die Erstellung einer Anthologie aus einer Reihe von fremden Texten nicht nötig ist, wenn die Übernahmen anders gekennzeichnet sind).

Für die Endredaktion muß Christine keine Verzeichnisse erstellen: Eine Leserin kann sich ein Inhaltsverzeichnis ansehen, indem sie einen Text in Gliederungssicht betrachtet, was sie selbst dann machen kann, wenn sie keine Schreibrechte auf das Dokument hat. Anstatt eines Literaturverzeichnisses kann sie sich im Linkkontextfenster des Dokuments alle Quellenlinks anzeigen lassen, die von dem Dokument ausgehen. Bei Quellen außerhalb des Hypertextsystems wurde ein Link auf ein eigenes oder fremdes Dokument mit den bibliographischen Angaben gelegt. Für ein annotiertes Literaturverzeichnis muß Christine die Quellen natürlich immer noch einzeln auflisten (oder mit Anmerkungen versehen), ebenso muß sie eigene Links für Personen- und Sachregister erstellen.<sup>1</sup>

Während des Schreibens hat Christine ihren Aufsatz bereits mit typographischen Auszeichnungen versehen. Es hängt vom Aufsatz ab, wie er veröffentlicht wird. Wenn es sich um ein einfaches Referat für die Erlangung eines Leistungsnachweises handelt, wird sie ihn nach Fertigstellung auf den Server des Studienganges übertragen (falls sie ihn nicht bereits dort erstellt hat) und ihrer Betreuerin den Zugriff erlauben.

Eine Diplomarbeit oder Dissertation wird entsprechend veröffentlicht, indem sie auf einen Server der Universitätsbibliothek abgelegt wird. Artikel werden auf einem Server einer Fachgesellschaft oder eines Verlages publiziert. Um eine Veröffentlichung auf einem Server zu veranlassen, auf den eine Autorin keinen direkten Zugriff hat (im Beispiel alle Server außer dem des Studienganges), stellt sie den entsprechenden Hypertext zunächst auf ihren Server und teilt seine Adresse der veröffentlichenden Stelle mit. Diese überträgt ihn auf ihren Server, nachdem sie ihn vorher hat begutachten lassen.

## 4.2 Anforderungen

Nachdem in Abschnitt 2.5.2 auf S. 27ff erste Anforderungen an ein technisches System zum Schreiben formuliert wurden, werden diese nun zu einer Spezifikation der Benutzungsmöglichkeiten und der -oberfläche eines solchen Systems weiterentwickelt.

### 4.2.1 Weiterentwicklung der Anforderungen

Für die Spezifikation des Publikationssystems werden die genannten Anforderungen unter Berücksichtigung der technischen Möglichkeiten von Hypertextsystemen, wie wir sie im letzten Kapitel

<sup>1</sup> wobei ein Personenregister wiederum durch Personenlinks auf Hypertexte über sie (z. B. ‚Homepages‘ im heutigen WWW) ersetzt werden könnte.



kennengelernt haben, erweitert. Das System muß für den gesamten Ablauf des wissenschaftlichen Schreibens benutzt werden können, von der Literaturbeschaffung und -übersicht bis zur Veröffentlichung (s. Abbildung 5 auf S. 27). Autorinnen müssen mit dem System also bestehende digitale Texte lesen, auf sie verweisen und neue Texte schreiben können.

Um die Vielfalt der möglichen Dokumente und der Verbindungen zwischen ihnen abzubilden, soll das Publikationssystem als Hypertextnetzwerk implementiert werden. Dokumente stellen Hypertextknoten, Verweise zwischen ihnen Kanten dar. Manuskripte und ihre Spezialisierungen Reinschriften, gesetzte Texte und Bücher (s. Abbildung 4) sollen also im System nicht explizit unterschieden werden. Die Funktion der Unterscheidungen wird von der Versionsverwaltung und dem Veröffentlichungskonzept übernommen, die weiter unten erläutert werden. Im System selbst müssen Dokumente geschrieben und Verweise zwischen ihnen erstellt werden können, das System muß Benutzerinnen also Editierfunktionen bieten. Abbildung 14 stellt einen ersten Entwurf der Oberfläche eines Zugriffsprogrammes für das System dar.

The screenshot shows a browser window with a menu bar (Datei, Ablage, Format, Ansicht, Navigation, Extras, Fenster) and a toolbar. The main content area displays a document titled "Frege und Turing" by Christine Pauli. The document text includes sections for "1 Einleitung", "2 Logisches", and "3 Maschinelles". A "Linkkontext" window is open on the right, showing a "Verzeichnis der Berichte" with a central "Bericht" node and arrows pointing to "Quelle 1", "Bibliographie 1", "Quelle 2", and "Quelle 3". The interface includes a toolbar with "Schließen" and "Gehe zu" buttons.

Abb. 14. Browser für ein Hypertext-Publikationssystem unter einer graphischen Benutzeroberfläche

Bei einem Dokument soll es sich um einen durchgängig zu lesenden Text von einer bestimmten Autorin mit einem Titel und einem Erstellungsdatum handeln. Ein Dokument soll so abgeschlossen sein, daß es eindeutig identifiziert und benannt werden kann. Ein Dokument soll neben dem Autorinnennamen, dem Titel und dem Erstellungsdatum weiterhin als Metainformation beinhalten, in welcher Sprache es geschrieben ist, damit aus mehreren Ausgaben eines Dokuments in verschiedenen Sprachen ausgewählt werden kann. Es muß außerhalb des Systems möglich sein, einen Bezeichner eines Dokuments zu notieren. Bezeichner müssen daher eindeutig und feststehend sein und außerdem eine gewisse Redundanz aufweisen können, indem sie aus anderen Zeichen als Ziffern bestehen dürfen, damit sie für Benutzerinnen leichter merk- und notierbar sind.

Auf Dokumente im System muß von anderen Systemen aus zumindest lesend zugegriffen werden können. Dokumente müssen gedruckt werden können, weil sie auf Papier generell einfacher

und auch von solchen Leuten lesbar sind, die keinen Zugriff auf das System haben. Auf dem Bildschirm sollen Dokumente jedoch nicht an die Vorgaben des möglichen Ausgabemediums Papier gebunden sein. Texte müssen als Hypertextstrukturen mit mehreren Dokumentenknoten angelegt werden können. Eine solche Struktur könnte z.B. aus einem Dokument mit dem eigentlichen Text und einem mit Endnoten bestehen.

Um die verschiedenen möglichen Arten von Verweisen zwischen Dokumenten auszudrücken, muß einem Link ein Typ zugewiesen werden können, an dem eine Leserin erkennen kann, in welchem Verhältnis das Ziel des Links zu seinem Ausgangspunkt steht. Ein Linkziel kann z.B. eine Erklärung, eine Quelle oder einen Gegenstandspunkt zum Linkursprung darstellen. Um in Hypertextnetzen navigieren und sie verwalten zu können, muß eine graphische Darstellung eines Unter-Netzwerkes von Dokumenten und Links angezeigt werden können (s. das kleine Fenster in Abbildung 14), in der Linktypen durch Farben unterschieden werden (die in der Abbildung nicht sichtbar sind). Dadurch kann beispielsweise Sekundärliteratur zu einem vorliegenden Text gefunden werden.

Aus allen Texten müssen Autorinnen Auszüge erstellen und in eine Ablage einsortieren, ebenso muß die Ablage eigene Texte und -stücke aufnehmen können. Eigene Texte werden als Hypertextdokumente und Teile davon gespeichert. Wenn sich eine Quelle im System befindet, kann ein Quellenverweis direkt auf sie zeigen, ein Verweis auf eine Quelle außerhalb des System hingegen auf einen klassischen Eintrag in einem Literaturverzeichnis zu einem Hypertext-Dokument.

Da sich Links auf und Auszüge aus einer Quelle in einem fertigen Dokument nicht ändern sollen, wenn eine neue Ausgabe der Quelle erstellt wird, müssen diese Verweise zeitfest sein (s. S. 80), damit sie eine Verbindung zu einer bestimmten Version des jeweiligen Originals behalten. Dokumententeile müssen auch außerhalb des Systems bezeichnet werden können, als Fundstellenangabe. Sie werden als Name des Dokuments, auf welches sie zeigen (inklusive Versionsnummer) und Adresse in ihm notiert.

Die Ablage muß frei gestaltet werden können. Dokumente müssen daher nicht in ein hierarchisches Verzeichnis eingeordnet werden. Verzeichnisse beliebiger Gestalt können jedoch als Hypertexte mit Links auf Dokumente und weitere Verzeichnisse erzeugt werden. Das System muß das Erstellen von Gliederungen unterstützen. Dabei soll es sich um Hypertextdokumente handeln, die Gliederungspunkte als speziell dargestellten (u.a. numerierten) Text enthalten. Gliederungspunkte müssen geschrieben und verschoben werden können.

Autorinnen müssen Textteile zu Texten montieren und immer wieder überarbeiten können. Die Editierfunktionen müssen dabei so vielseitig und bequem zu bedienen sein wie in gängigen Textverarbeitungsprogrammen, allerdings auf Hypertexte zugeschnitten. Die Verwaltung von Textstrukturen ist wichtig, die Orientierung an Papier hingegen nicht. In Gliederungen müssen Teile von eigenen und fremden Texten eingeordnet werden können, um Zettelmanuskripte zu erstellen. Dieses Einordnen erfolgt durch Setzen von Links auf die Textteile. Die Links müssen fließend in der Zeit bzw. ortsfest sein dürfen, damit die Umformulierung eines Abschnittes einen Verweis auf ihn nicht zerstört.

Damit Autorinnen Übergänge zwischen einzelnen Arbeitsschritten nachverfolgen und Links auf sich verändernde Dokumente setzen können, müssen automatisch Versionen von Dokumenten verwaltet werden. Texte müssen für die Endredaktion optisch attraktiv gestaltet, d.h. typographisch ausgezeichnet werden können. Arbeitsmaterialien und andere Dokumente müssen schon vor der endgültigen Veröffentlichung anderen zugänglich gemacht werden können, damit die anderen Benutzerinnen sie bewerten oder mit ihnen arbeiten können. Umgekehrt muß es Autorinnen auch möglich sein, anderen den Zugriff auf private Daten zu verwehren. Ein Dokument muß eine Kurz- und eine Langfassung eines Textes enthalten können, zwischen denen gewechselt werden kann, also Stretchtext.

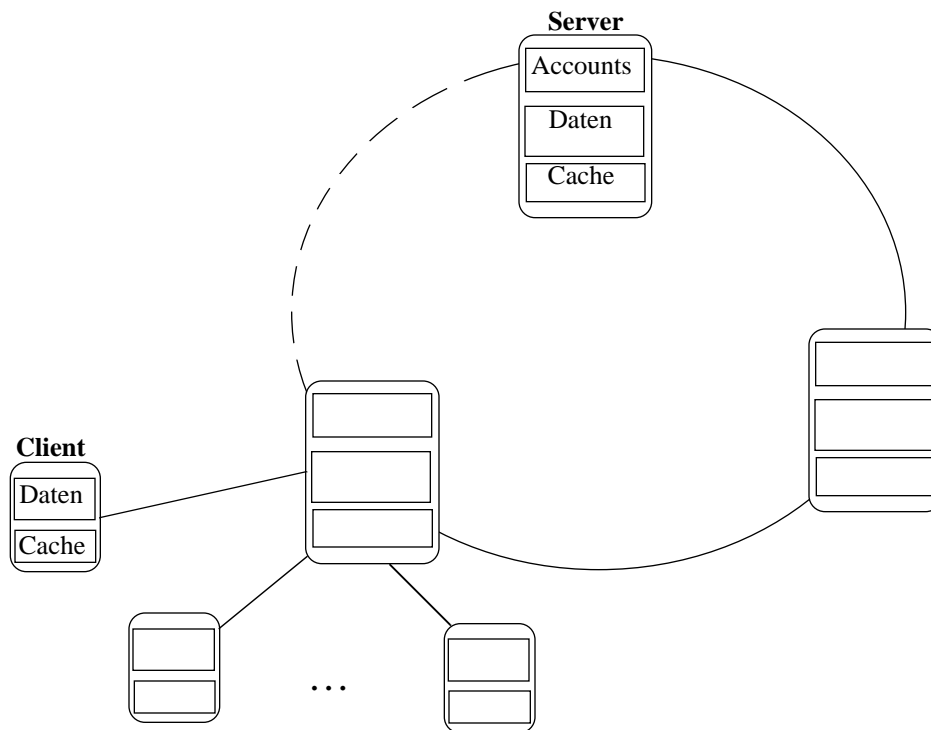


Abb. 15. Systemmodell des Publikationssystems.

Für Pflicht- (z.B. Dissertationen) oder nicht bibliographisch erfaßte Veröffentlichungen (z.B. technische Berichte) muß ihre jeweilige Autorin den letzten Schritt der Publikation als Bereitstellen für die Öffentlichkeit und Aufnahme in Verzeichnisse selbst vornehmen können. Sie muß Dokumente dazu auf offizielle Publikationsserver legen können. Diese Server müssen selbst formal/mechanisch geordnete Verzeichnisse anlegen (s. S. 15), damit im Zweifelsfall entschieden werden kann, ob ein Dokument überhaupt vorhanden ist. Weitere Verzeichnisse können als Hypertexte mit Links auf andere erstellt werden.

Die inhaltliche Qualität von Dokumenten muß im System gesichert werden können, z.B. durch eine Begutachtungshierarchie. Um Begutachtungen zu ermöglichen und das Verstehen von fremden Texten zu erleichtern, müssen beliebige Benutzerinnen an eigenen und fremden Texten Anmerkungen zu bestimmten Passagen anbringen können, die von ihnen und anderen Benutzerinnen gelesen werden können.

Eine Anmerkung muß jeweils neben dem Abschnitt, auf den sie sich bezieht, angezeigt werden können, in einer eigenen Spalte bzw. einem eigenen Fenster. Eine Anmerkung zu einem ganzen Dokument soll in einem eigenen Dokument stehen, auf das ein Link von dem Quelldokument aus verweist. (Wenn eine Anmerkung am Anfang des Quelldokuments dargestellt würde, würde sie den Ursprungstext verdrängen. Wenn sie hingegen am Ende dargestellt würde, müßte sowieso ein Link auf sie hinweisen.) Damit Anmerkungen nicht vom Lesen des jeweiligen Textes ablenken, muß es private geben, die nur die jeweilige Autorin wahrnimmt, lesen und schreiben kann. Außerdem müssen öffentliche Anmerkungen eine Angabe über ihre jeweilige Autorin beinhalten und muß die Anzeige von Anmerkungen abgeschaltet werden können, auch in Abhängigkeit von ihren jeweiligen Autorinnen.

Durch einen Begutachtungsprozeß selektierte Texte sollen in öffentlichen Verzeichnissen aufgeführt werden, die von Bibliotheken und Verlagen verwaltet sind. Die Archivierung von Dokumenten ist sowohl ein technisches als auch ein soziales Problem, v.a. wegen der Flüchtigkeit digitaler Daten, weil nicht nur Archivierungsserver eingerichtet, sondern auch festgelegt werden muß, wann

ein Dokument auf ihnen durch ein anderes ersetzt oder gar gelöscht werden darf. Wenn beispielsweise eine Autorin ein Dokument veröffentlicht hat, soll sie unter demselben Namen kein anderes veröffentlichen dürfen, allerdings eine neue Auflage des Dokuments, die eine höhere Auflagennummer erhält.

Anforderung	WWW	HyperWave	Xanadu	<i>Transactions</i>
Benutzerinnen- und Zugriffsverwaltung	tlw.	ja	ja	<i>ja</i>
Abrechnung	nein	ja	ja	<i>nein</i>
Nutzung bestehender digitaler Texte	ja	ja	nein	<i>ja</i>
Editierfunktion im System	nein	nein	ja	<i>ja</i>
typographische Auszeichnungen	ja	ja	ja	<i>ja</i>
Versionsverwaltung	nein	nein	ja	<i>ja</i>
Gliederungsfunktion	nein	nein	ja	<i>ja</i>
Hypertextknoten als Dokumente	ja	ja	ja	<i>ja</i>
Metainformationen zu Dokumenten	tlw.	ja	ja	<i>ja</i>
verschiedene Sprachen	ja	ja	ja	<i>ja</i>
Benennung über Namen	ja	nein	nein	<i>ja</i>
eindeutige, ewige IDs	nein	ja	ja	<i>ja</i>
hierarchische Datenverwaltung	ja	ja	nein	<i>nein</i>
Sprunglinks	ja	ja	ja	<i>ja</i>
Transklusionen	nein	nein	ja	<i>nein</i>
Anmerkungen	nein	ja	ja	<i>ja</i>
Stretchtext	nein	nein	ja	<i>ja</i>
getypte Links	nein	nein	ja	<i>ja</i>
Links von/auf Dokumententeile	nein	ja	ja	<i>ja</i>
zeitfeste Links auf fremde Dokumente	nein	ja	ja	<i>ja</i>
ortsfeste Links auf eigene Dokumente	ja	tlw.	ja	<i>ja</i>
graphische Darstellung eines Unternetzwerkes	nein	ja	?	<i>ja</i>
Drucken eines Unternetzwerkes	nein	nein	nein	<i>ja</i>
Zugriff von anderen Netzwerken aus	ja	ja	nein	<i>ja</i>
private Ablage	nein	tlw.	nein	<i>ja</i>
Bereitstellen von Dokumenten für andere	ja	ja	ja	<i>ja</i>
Veröffentlichung durch spezielle Dokumente und Server	nein	nein	ja	<i>ja</i>

Tabelle 3. Vergleich von Transactions mit dem WWW, HyperWave und Xanadu

Veröffentlichungsserver müssen ständig am Netz sein und Veröffentlichungen lange aufheben. Falls sich das System oder die dazu nötige Technik so weiterentwickeln, daß auf alte Dokumente nicht mehr zugegriffen werden kann, müssen die alten Dokumente auf ein neues System konvertiert werden und der Zugriff über die alten Namen möglich bleiben. Die Abrechnung von Dokumenten

stellt keine Anforderung dar, weil sie für die Verbreitung wissenschaftlicher Informationen hinderlich ist, solange diese Verbreitung auch ohne funktioniert (s. S. 9). Der Zugriff auf bestimmte Dokumente soll jedoch durch Paßwörter gesperrt werden können, die auch verkauft werden können.

Als Benutzerinnen des Systems werden Wissenschaftlerinnen in Universitäten und ähnlichen Institutionen angenommen. Eine gewisse Einarbeitungszeit wird als zumutbar angesehen. Server und Daten auf ihnen sollen von Bibliotheken, Verlagen, Fachbereichen und anderen wissenschaftlichen Instituten verwaltet werden. In Tabelle 3 werden das WWW, HyperWave und Xanadu bezüglich der eben geschilderten Anforderungen verglichen. Die Tabelle entspricht in etwa Tabelle 2 auf S. 89, außer daß die Einträge den obigen Ausführungen entsprechend formuliert sind und eine fünfte Spalte hinzugekommen ist. Wie man sieht, können weder das WWW noch HyperWave alle Anforderungen erfüllen. Daher soll im nächsten Abschnitt ein neues Hypertextsystem spezifiziert werden, das dazu in der Lage ist (und bereits die fünfte Spalte der Tabelle belegt).

#### 4.2.2 Spezifikation von Transactions

In diesem Abschnitt wird ein Hypertextsystem für das wissenschaftliche Arbeiten spezifiziert. Um leichter über es reden zu können, soll es einen Namen bekommen, ‚Transactions‘, nach dem englischen Ausdruck für Sitzungsprotokolle von wissenschaftlichen Gesellschaften, der im Titel vieler wissenschaftlicher Zeitschriften zu finden ist.<sup>2</sup> Nachdem in Abschnitt 2.5.1 auf S. 24 der Ist-Zustand beschrieben und im letzten Abschnitt die Anforderungen unter Berücksichtigung des Hypertextkonzeptes weiterentwickelt wurden, sollen nun die Benutzungsschnittstelle und die Funktionalität des Systems festgelegt werden. Ein Browser für das System soll eine graphische Benutzungsoberfläche aufweisen, mit Pulldown-Menüs und Dialogboxen. Die Oberfläche wird in Anlehnung an die von bekannten Betriebssystemen, aber bewußt schematisch skizziert, um ihren vorläufigen Charakter zu betonen (s. Abbildung 16). Auf das Verhalten des Systems bei Fehlern wird ebenfalls nur am Rande eingegangen.

Noch ein paar Worte zur Systemumgebung: Die Anforderungen an Transactions als weltweites Publikationssystem werden nach dem Client-Server-Konzept mit verteilten Autorinnen- und Publikationscomputern umgesetzt (s. Abbildung 15). Benutzerinnen greifen auf es über ihre grafikfähigen Arbeitsplatzrechner und Server von Studiengängen, Bibliotheken, Verlagen und Rechenzentren zu. Vorhandene Postscript- und HTML-Dokumente auf FTP- und WWW-Servern sind im System lesbar. Es können Verweise auf diese externen Dokumente erstellt werden. Um möglichst vielen Netzbenutzerinnen den Zugriff auf Texte im System zu ermöglichen, müssen die Texte von einem anderen Netz mit verbreiteten Programmen zumindest gelesen werden können, v.a. mit WWW-Browsern.

##### 4.2.2.1 Betrachtung und Erstellung von Dokumenten

Wie in Abbildung 14 zu sehen war, stellt der Transactions-Browser eine Mischung aus einem WWW-Browser und einem Textverarbeitungsprogramm dar. Er bietet weiterhin Funktionen eines Gliederungsprogrammes, Überschriften können also als Gliederungspunkte verschoben und Text unter einer bestimmten Gliederungsebene ausgeblendet werden. Schließlich erlaubt der Browser die Verwaltung einer persönlichen Hypertextablage auf dem lokalen Computer einer Benutzerin. In ihrer Ablage kann die Benutzerin Daten speichern, auf die sie andere Benutzerinnen nicht zugreifen lassen möchte (bzw. noch nicht, wenn es sich um Dokumente in Entstehung handelt). Damit das System leicht bedient werden kann, benutzt der Browser das Fenstersystem des jeweiligen Rechners für die Ein- und Ausgabe.

<sup>2</sup> Dies galt schon für die Vorläuferin der heutigen wissenschaftlichen Zeitschriften, *Philosophical Transactions*, die am 6. März 1665 zum ersten Mal erschienen ist ( WALKER U. HURT 90, 9ff).

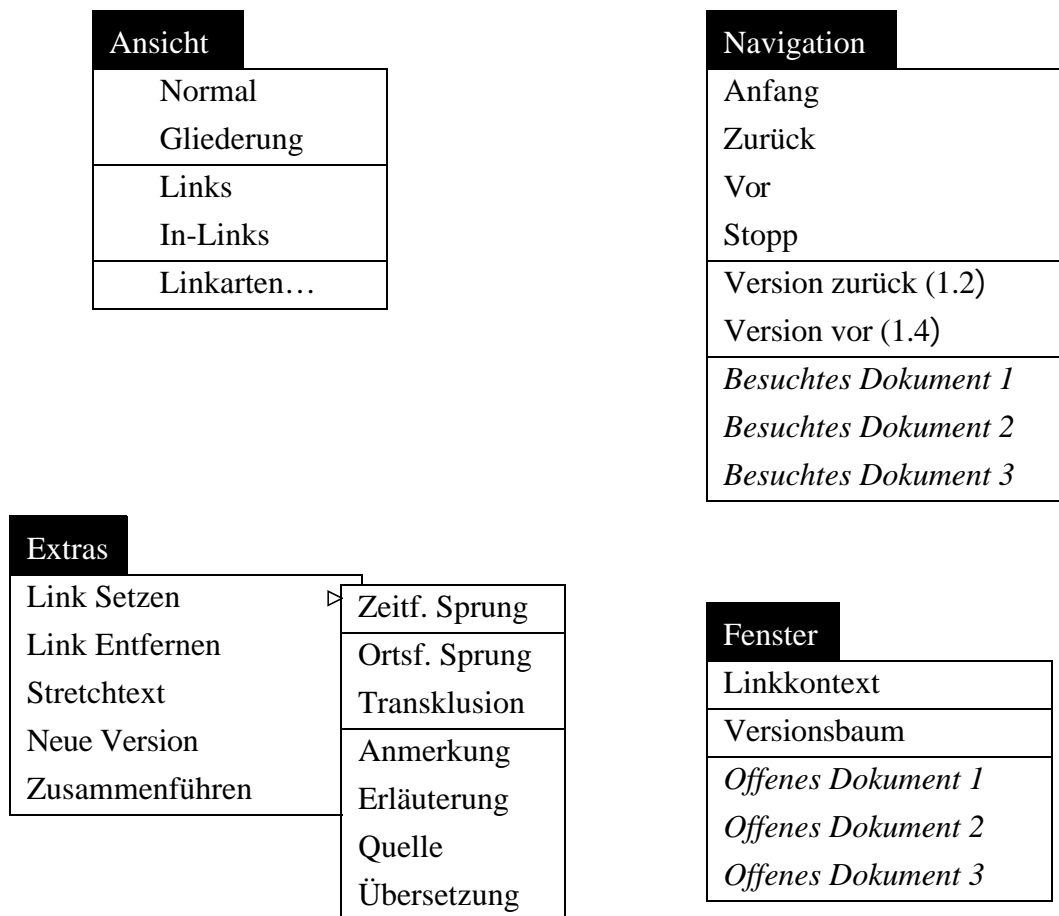
Datei	Ablage	Format
Neu	Rückgängig	Zeichensatz ▷
Öffnen ...	Ausschneiden	Größe ▷
Schließen	Kopieren	Stil ▷
Sichern	Einfügen	Zeichen ▷
Sichern unter ...	Löschen	Absatz ▷
Attribute	Alles Markieren	Link ▷
Papierformat	Suchen/Ersetzen ...	Sprunglink
Drucken	Weitersuchen	Transklusion
Beenden	Eigenschaften ...	Stretchlink
		Anmerkung
		Erläuterung
		Quelle
		Übersetzung
		Veröffentlichung
		Neu ...

Abb. 16. Erster Entwurf von Pulldown-Menüs für Transactions ...

Als Browser können mit dem Transactions-Client Dokumente im System betrachtet und annotiert werden. Wenn er gestartet wird, lädt er zunächst ein voreingestelltes Dokument (s. Abbildung 17), damit die Benutzerin über es zu gerade in Arbeit befindlichen Dokumenten gelangen kann. Ein Dokument kann auf zwei Arten geöffnet werden, über die Eingabe seines Namens oder das Verfolgen eines Links auf es in einem anderen Dokument (s. Abbildung 16). Außerdem kann, wie in WWW-Browsern auch, in einer Liste der zuletzt besuchten Dokumente vor- und zurückgesprungen werden, damit die Benutzerin sich leichter in einer Hypertextstruktur orientieren kann (s. Abbildung 16; es werden auch die Namen von besuchten Dokumenten zum möglichen Wiederaufruf angeboten). In ihrer Ablage und dem Server, mit dem eine Benutzerin gerade verbunden ist, kann sie nach Wörtern und Metainformationen (wie der Autorin, dem Titel, den Erstellungs- und Änderungsdaten) von Dokumenten suchen.

Als Textverarbeitungsprogramm ermöglicht der Client das Erstellen von neuen Dokumenten auf gängige Weise. Textstellen können verschiedene typographische Attribute zugewiesen werden, um sie z.B. hervorzuheben. Der Name und Speicherort eines Dokuments kann, wie bei anderen Editoren, beim Speichern des Dokuments festgelegt werden. Bei dem Speicherort kann es sich um die lokale Ablage oder einen Server handeln. Wenn beim Speichern eines neuen Dokuments ein Dokument gleichen Namens auf dem Server oder in der Ablage vorhanden ist, muß ein anderer Name gewählt werden, damit Dokumente eindeutig identifiziert werden können.

Diese Spezifikation geht davon aus, daß Dokumente in den meisten Fällen in der privaten Ablage erstellt werden, wo sie vom Zugriff durch andere geschützt sind. Wenn sie fertiggestellt sind, können sie auf einen Server übertragen werden. Dokumente können aber auch direkt auf einem Server erstellt werden, z.B., um anderen Benutzerinnen Entwürfe verfügbar zu machen. Dokumente auf einem Server und der privaten Ablage werden nach ihrem Namen unterschieden: Namen von ersteren enthalten auch einen Servernamen, Namen von letzteren nur den eigentlichen Dateinamen (siehe die Ausführungen zur Benennung weiter unten).



Fortsetzung von Abbildung 16

Dokumente können in der Ablage auch dann editiert werden, wenn der Rechner, auf dem der Browser läuft, vorübergehend nicht an das Internet angeschlossen ist. Dies empfiehlt sich jedoch nicht, weil dann nicht auf andere Dokumente im Netz zugegriffen werden kann. Wenn ein Dokument geändert wird, das In-Links von anderen Dokumenten enthält, werden die Server dieser Dokumente benachrichtigt, sobald der Client-Rechner wieder mit dem Netzwerk verbunden ist (s. die Beschreibung der Links weiter unten). Die Besitzerinnen der Dokumente können diese dann entsprechend anpassen.

In dem System kann eine Benutzerin beliebige Texte in ihrer lokalen Ablage sichern, um z.B. schneller auf fremde Dokumente zugreifen zu können. Ein gesichertes Dokument erhält einen ortsfesten Link auf das jeweilige Ursprungsdokument, das dadurch angefordert und darauf überprüft werden kann, ob es sich geändert hat.

Jedes Dokument enthält als Metadaten einen Namen, die Autorin, Zugriffsrechte, das Datum der Erstellung und das der letzten Änderung. Dadurch kann es, wie eine Datei in einem Dateisystem auch, leicht identifiziert werden. Die Zugriffsrechte werden jeweils bei der Erzeugung eines Dokuments festgelegt und können auch später noch gesetzt werden. Benutzerinnen können sich in Gruppen zusammenschließen, deren Mitglieder bezüglich Zugriffsrechten gleichbehandelt werden. Dadurch können sie u.a. gemeinsam ein bestimmtes Dokument bearbeiten, wenn es sich auf einem Server befindet (s. die Beschreibung der Versionsverwaltung weiter unten).

Es werden verschiedene Sprachen für Dokumente unterstützt, damit eine Benutzerin ein Dokument automatisch in der Sprache geliefert bekommt, die sie bevorzugt, falls es in dieser Sprache

Eigenschaften	
Server:	<input type="text" value="berichte.informatik.uni-bremen.de"/>
Anfangsdokument:	<input type="text" value="Anfang"/>
Sprache:	<input type="text" value="Deutsch"/>
Anzahl gesicherter Revisionen:	<input type="text" value="2"/>
Anzahl angezeigter In-Links:	<input type="text" value="100"/>
<input type="button" value="Abbrechen"/> <input type="button" value="OK"/>	

Abb. 17. Dialogbox, um Eigenschaften des Transactions-Browsers zu ändern.

vorhanden ist. Im Client kann eine Benutzerin eine Sprachpräferenz angeben (s. Abbildung 17). Ein Dokument wird vom Server ausgewählt, falls es in der Sprache verfaßt ist, die in den Eigenschaften des anfordernden Browsers gewählt ist und es selbst oder ein anderes Dokument mit anderer Sprache angefordert wurde, das mit einem Übersetzungslink auf es zeigt. (Es kann also vorkommen, daß ein Dokument mit einer anderen Adresse geliefert wird als der, welche angefordert wurde. Dies kann verhindert werden, indem die Präferenzsprache des Browsers auf ‚keine‘ gestellt wird.) Auf eine Übersetzung eines Dokuments kann über einen Übersetzungslink verwiesen werden. Die Übersetzungslinks zu einem Dokument werden im Linkkontextfenster dargestellt (s. weiter unten).

Zu jedem Dokument können seine Metadaten angezeigt und geändert werden (s. Abbildung 19), seine Adresse nur durch Speichern des Dokuments unter einem anderen Namen oder an einem anderen Ort. Der Titel eines Dokuments, der im Kopf eines Dokumentenfensters und im Linkkontext-Fenster angezeigt wird, kann unabhängig von der Adresse gewählt werden. Das Erstellungsdatum kann nicht, das Änderungsdatum nur indirekt durch Speichern des Dokuments geändert werden. Die Eigentümerin bzw. -gruppe und die Zugriffsrechte eines Dokuments für die Eigentümerin selbst, die Gruppe und alle anderen Benutzerinnen des Systems können hingegen geändert werden, um anderen den Zugriff auf das Dokument zu ermöglichen bzw. zu verwehren.

Transactions-Dokumente besitzen keine feste Breite. Diese wird in Abhängigkeit von der Fenstergröße gewählt, wie in WWW-Browsern auch. Dadurch können selbst auf 15-Zoll-Bildschirmen in entsprechend schmalen Fenstern zwei Dokumente nebeneinander betrachtet oder es kann zu einem Dokumentenfenster die Übersichtskarte mit Links (s. weiter unten) dargestellt werden. Wörter am Zeilenende werden automatisch getrennt. Für schwierig automatisch zu trennende Wörter können die Trennstellen beim Schreiben des entsprechenden Dokuments angegeben werden.

In Transactions kann eigener und fremder Text gedruckt werden, um ihn leichter lesen zu können. Wenn ein vernetzter Hypertext gedruckt werden soll, werden zunächst die einzelnen Dokumente ausgewählt (in einem Linkkontext-Fenster, s. Abbildung 18). Das System macht dann einen Vorschlag über die Reihenfolge der Ausgabe der Dokumente, indem es sie zu einem neuen Dokument zusammenstellt, in welchem die Reihenfolge in der Gliederungssicht (s. weiter unten) geändert werden kann.

Das zusammengestellte Dokument wird zunächst in einem Fenster angezeigt, dessen Breite der des Ausdrucks entspricht, so daß Umbrüche überprüft werden können. Sprünge zu Dokumenten,



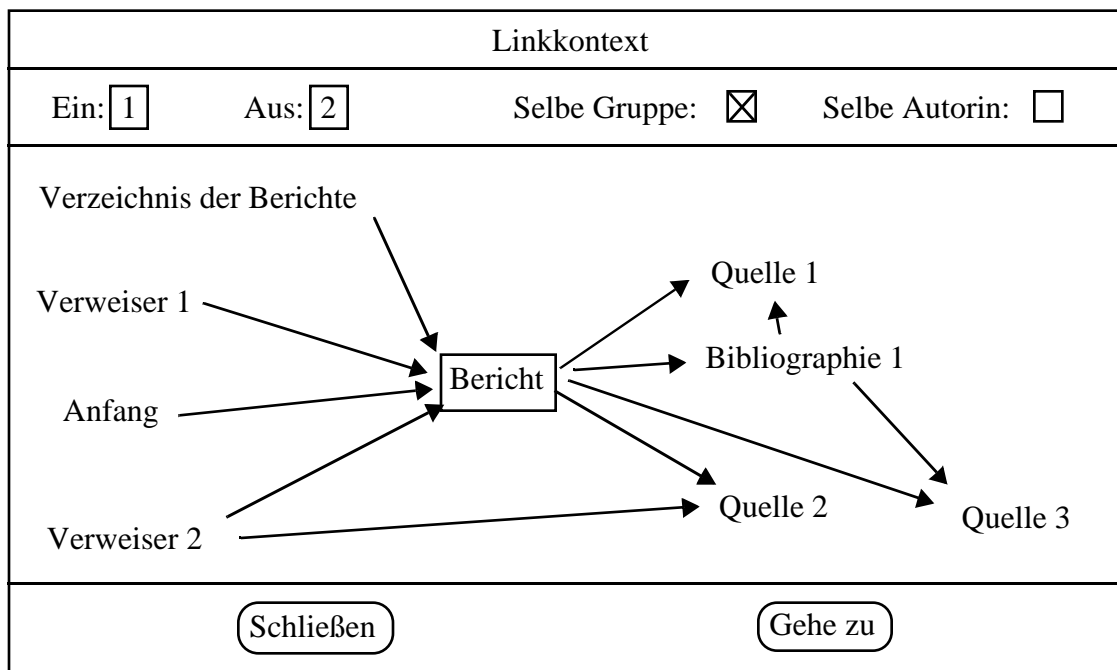


Abb. 18. Dialogbox, um In- und Out-Links eines Dokuments darzustellen.

die mitgedruckt werden, werden in einem Ausdruck durch einen Querverweis der Form ‚(s. Seite  $n$ )‘ ersetzt.

#### 4.2.2.2 Übersichtsdarstellungen für Dokumente

In Transactions gibt es Verzeichnisse nicht als eigene Dokumentenart, sie können jedoch als Hypertextdokumente erstellt werden, die Links auf Dokumente und andere Verzeichnisse enthalten. Server halten jeweils zwei Dokumente vor, in denen alle auf ihnen gespeicherten Dokumente verzeichnet sind, einmal nach Autorinnennamen und einmal nach Titeln sortiert. Von diesen beiden Verzeichnissen werden auch Versionen verwaltet (s. weiter unten), so daß das Hinzukommen von neuen Dokumenten auf einem Server nachvollziehbar ist.

Als Ersatz für Verzeichnisfenster kann zu jedem Dokument das ‚Linkkontextfenster‘ angezeigt werden. In diesem Fenster werden alle Links, die auf ein Dokument zeigen und von ihm ausgehen, dargestellt, inklusive der darüber verbundenen Dokumente (s. Abbildung 18). Links können im Linkkontextfenster vom Ziel zum Ausgang verfolgt werden. Dokumente werden in der Übersicht durch ihre Namen und Links als Linien zwischen Dokumenten dargestellt.

Das Dokument, auf das sich eine Linkkontextdarstellung bezieht, ist durch einen Kasten markiert. Es kann gewählt werden, wieviele In- und Out-Links im Linkkontextfenster jeweils angezeigt werden, außerdem, ob nur Dokumente angezeigt werden, die derselben Benutzerin oder derselben Benutzerinnengruppe wie das gerade aktive Dokument gehören. Dadurch kann eine Überfrachtung der Anzeige verhindert werden. Es wird in dieser Spezifikation allerdings davon ausgegangen, daß auf die meisten wissenschaftlichen Dokumente nicht allzuvielen Links zeigen. Im Linkkontextfenster können Dokumente zum Drucken, Übertragen auf einen Server oder Speichern in der eigenen Ablage ausgewählt werden.

Die Gliederungssicht eines Dokuments entspricht der gängiger Textverarbeitungssysteme und läßt sich auch so bedienen. Überschriften werden als Gliederungspunkte verschiedener Ebenen interpretiert, die verschoben werden können. Diese Überschriften können auch Links auf andere Dokumente enthalten. Außerdem können Überschriften und anderer Text unterhalb einer bestimm-

Attribute des aktuellen Dokuments	
Adresse:	berichte.informatik.uni-bremen.de:Christine.Pauli/Diplomarbeit/3.10
Eigentümerin:	<input type="text" value="Christine"/>
Titel:	<input type="text" value="Hypertext und Theoretische Informatik"/>
Datum: Erstellung:	23.11.96
Letzte Änderung:	03.01.97
Größe:	537 Kilobytes
Gruppe:	<input type="text" value="Studentinnen"/>
Rechte: Autorin:	<input checked="" type="checkbox"/> Lesen <input checked="" type="checkbox"/> Schreiben
Gruppe:	<input checked="" type="checkbox"/> Lesen <input checked="" type="checkbox"/> Schreiben
Welt:	<input checked="" type="checkbox"/> Lesen <input type="checkbox"/> Schreiben
Sprache:	<input type="text" value="Deutsch"/>
<input type="button" value="Abbrechen"/> <input type="button" value="OK"/>	

Abb. 19. Dialogbox, um Attribute eines Transactions-Dokuments im Browser zu ändern.

ten Ebene ausgeblendet werden. Ein Dokument kann so als kleiner Karteikasten benutzt werden, indem ein Abschnitt mit einer Überschrift wie eine Karteikarte benutzt wird. (Eine hierarchische Gliederungsfunktion wird unterstützt, obwohl Dateien nicht in Verzeichnissen verwaltet werden, weil Dokumente aufgrund ihrer Größe relativ übersichtlich sind und als Verzeichniseratz benutzt werden sollen.) Eine Datei kann in der Gliederungssicht auch geändert werden.

#### 4.2.2.3 Dokumentennamen

Dokumentennamen werden nicht (wie in HyperWave) als Nummern vom System vorgegeben, sondern können aus Buchstaben und anderen Zeichen (außer dem Doppelpunkt) von der Benutzerin gewählt werden. Dadurch können Namen festgelegt werden, die Benutzerinnen den Austausch von Adressen von Dokumenten auch außerhalb von Rechnernetzen erlauben, z.B. auf Papier. Intern besitzen Dokumente eine Nummer, die auf ihrem Server eindeutig ist. Diese kann nicht geändert werden und ist für Benutzerinnen unsichtbar.

Der Name eines Dokuments wird von seiner Besitzerin vergeben. Er besteht aus zwei Feldern, einem Server- und einem Dokumentenfeld. Die Felder sind durch Doppelpunkte voneinander getrennt. Das Serverfeld enthält den Namen des Servers, auf dem sich das Dokument befindet, seinen Heimatserver. Dies ist der Technik des Internets geschuldet, auf dem das Transactions-Netz basiert und in dem Daten relativ zu einem Server adressiert werden.

Nach dem Serverfeld folgt der eigentliche Dokumentenname. Dieser kann wieder durch beliebige Zeichen (außer dem Doppelpunkt) unterteilt werden. Vorgeschlagen wird die Unterteilung des Namens in Unterfelder durch Schrägstriche und die Unterteilung der Unterfelder durch Punkte. Ein Dokumentenname sollte den Autorinnennamen enthalten, muß es aber nicht. Eine bestimmte Version eines Dokuments hat z.B. die Adresse

berichte.informatik.uni-bremen.de:Christine.Pauli/Diplomarbeit/3.10

Linkarten Bearbeiten					
Linkart:	Quelle		basiert auf:	Zeitf. Sprung	
Erste Form (In-Link)					
Stil:	Unterst.	Farbe:	Grau	Davor:	
		Dahinter:		<input type="checkbox"/>	Kasten
Zweite Form (Out-Link)					
Stil:	Normal	Farbe:	Schwarz	Davor:	»
		Dahinter:	«	<input checked="" type="checkbox"/>	Kasten
<input type="button" value="Abbrechen"/>		<input type="button" value="Neu"/>		<input type="button" value="OK"/>	

Abb. 20. Dialogbox, um die Darstellung von Links (In- und Out-Links) einzustellen

(siehe die folgenden Erläuterungen zur Versionsverwaltung).

#### 4.2.2.4 Versionsverwaltung

In Transactions werden Versionen von Dokumenten verwaltet. Versionen dienen der Unterscheidung verschiedener Stufen eines Dokuments (wie Entwurf, Endfassung etc.) und dem Aufbewahren von Dokumenten, auf die Links zeigen und die geändert werden. Wenn ein Dokument ohne Versionsangabe angefordert wird, erscheint seine letzte Version. Diese enthält eine Versionsnummer, die nach einem Schrägstrich an den Namen angefügt wird.

Wenn eine Versionsnummer eines Dokuments größer als ‚1.1.‘ ist, es also eine oder mehrere Vorgängerversionen gibt, kann die Benutzerin über einen Menüpunkt ihres Editors versionsweise zurückgehen, wenn es Nachfolgeversionen gibt, entsprechend vorgehen. Es kann außerdem ein Fenster angefordert werden, in dem die Versionsgruppe eines Dokuments, sein Versionsbaum, dargestellt wird, ähnlich zur Darstellung eines Versionsgraphen in Abbildung 13 auf S. 69.

Wenn ein neues Dokument erzeugt wird, erhält es die Versionsnummer ‚1.1‘ und es kann sofort in ihm geschrieben werden, es muß also nicht entsperrt werden, wie in manchen Versionsverwaltungssystemen nötig (s. S. 68). Wenn ein Dokument gesichert wird, wird die Versionsunternummer mit dem niedrigsten Rang (die am weitesten rechts steht) erhöht. Höherrangige Versionsnummern können durch einen eigenen Befehl erhöht werden, über den auch Varianten eines Dokuments erzeugt werden können (die, wie in Xanadu, immer auch Revisionen sind, s. S. 68ff). Wenn eine Benutzerin eine Variante erzeugt, werden die Revisionen eines Dokuments (bis auf eine einstellbare Anzahl) gelöscht.

Eine Variante ist eine Version auf einem anderen Versionszweig, parallel zum Versionshauptstrang. Die Variante erhält einen beliebigen symbolischen Bezeichner als Hauptversionsbezeichner, der auch eine Zahl sein kann, solange diese Zahl nicht auf der Hauptversionslinie liegt. Die Nummer muß daher eine andere Stelligkeit als die der Hauptversionslinie erhalten. Wenn die Hauptversionen z.B. ‚1.1.‘, ‚1.2‘ ... hochgezählt werden, kann eine Unterversion beispielsweise ‚Versuch.1‘, ‚Versuch.2‘ ... heißen oder ‚1.2.1‘, ‚1.2.2‘ ... Um die Eindeutigkeit der Versionsbeziehung zu erhalten, muß außerdem eine Variante erzeugt werden, wenn ein Dokument geändert werden soll, das schon einen Nachfolger hat.

Linkarten anzeigen		
Linkarten	1. Form	2. Form
	(In-Link)	(Out-Link)
Zeitf. Sprung Ortsf. Sprung Transklusion Stretchtext		
Anmerkung Erläuterung Quelle Übersetzung		
<div style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block; margin: 10px 0;">Linkarten Bearbeiten...</div>		
<div style="display: flex; justify-content: space-around; width: 100%;"> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block;">Abbrechen</div> <div style="border: 1px solid black; border-radius: 15px; padding: 5px; display: inline-block;">OK</div> </div>		

Abb. 21. Dialogbox, um zu wählen, welche Linkarten angezeigt werden.

Da in Transactions Dokumente direkt auf Servern bearbeitet werden können, auf die vom ganzen Netz aus zugegriffen werden kann (vorausgesetzt, die Rechte sind entsprechend gesetzt), kann es vorkommen, daß mehrere Autorinnen gleichzeitig an einem Dokument arbeiten. Diese Spezifikation geht davon aus, daß sich die Autorinnen in einem solchen Fall absprechen. Sie können größere Überarbeitungen eines Dokuments in ihrer Ablage vornehmen und von Zeit zu Zeit auf den Server übertragen. Eine Autorin wird sofort vom Server gewarnt, wenn sie versucht, ein Dokument zu überschreiben, dessen Versionsnummer gleich oder höher der ihres eigenen Dokuments ist. Sie kann dann das Dokument auf dem Server in ihren Browser laden und ihre Version in es einmischen, um die entstehende Version unter einer neuen Versionsnummer auf den Server zurückzuschreiben.

Zwei Varianten, also Versionen, zwischen denen keine (unmittelbare oder mittelbare) Vorgänger/Nachfolger-Beziehung besteht, können im Editor manuell zusammengeführt werden. Zusammenführungen beziehen sich immer auf ein einziges Dokument, das Grunddokument, in das Änderungen eines anderen (Löschungen und Einfügungen) teilweise übernommen werden. Obwohl das Grunddokument und seine Variante nicht unmittelbar auseinander abgeleitet sind, sind sie doch (nach Definition) aus einer gemeinsamen Abstraktion entstanden (s. S. 68). Bei dieser handelt es sich in Transactions, wie in Xanadu auch, immer um ein Dokument. (Sie ist nicht nur immateriell, beispielsweise als gemeinsames Thema zweier Dokumente.)

Es kann in Transactions daher eine neue Version eines gemeinsamen Ausgangsdokuments von zwei oder mehr Varianten erzeugt werden. Die Unterschiede der anderen Dokumente zum Grunddokument werden dazu typographisch hervorgehoben im Grunddokument selbst dargestellt. Sinnvoll erscheinen Unterstreichungen für Einfügungen und Durchstreichungen für Löschungen.

Eine Verschiebung eines Textes aus dem Grunddokument wird im Prinzip als Löschung eines Blockes an einer und Einfügung desselben Blockes an anderer Stelle angezeigt, allerdings mit einer zusätzlichen Unterstreichung. Ein verschobener Textblock erscheint an seinem Ursprungsort also gleichzeitig unter- und durchgestrichen, am Zielort doppelt unterstrichen. Es besteht so kaum die

Gefahr, daß neue Benutzerinnen des Systems durch den zusätzlichen Unterstrich verwirrt werden, weil die mit ihm verwendeten einfachen Unter- bzw. Durchstreichungen ihre alte Bedeutung behalten.

Änderungen aus verschiedenen Dokumenten werden zum Zusammenmischen farblich unterschieden, mit einer Farbe je Variantendokument. Teile des Grunddokuments, die in mehreren Varianten gelöscht wurden, werden auch mehrmals dargestellt, jeweils in einer anderen Farbe. Änderungen können im Grunddokument bleiben oder entfernt werden. Die Änderungsanzeige kann auch abgeschaltet werden, um die Darstellung des Dokuments im zusammengeführten Zustand zu erhalten. Schließlich können die nichtentfernten Änderungen in eine neue Version übernommen werden.

#### 4.2.2.5 Links

Verweise zwischen Dokumenten werden in Transactions durch Sprunglinks dargestellt. Links sind immer gerichtet, von einem Ausgangs- zu einem Zieldokument. Sie werden in beiden Dokumenten gespeichert, wobei sie im Zieldokument als In-Links bezeichnet werden, im Ausgangsdokument entsprechend als Out-Links. In Transactions werden neben Sprunglinks auch Stretchtext unterstützt.

Einen Out-Link, den eine Autorin von einem ihrer Dokumente auf ein anders zeigen läßt, wird bezüglich der Benutzung so behandelt, als ob er in ihrem Dokument gespeichert wäre. Er wird z.B. gelöscht, wenn das Ausgangsdokument gelöscht wird. Analog werden In-Links im Zieldokument gespeichert. Eine Autorin kann auch einen Sprunglink zwischen zwei Dokumenten legen, auf welche sie keinen Schreibzugriff hat. Dabei kann sie wählen, ob der Link privat oder öffentlich sein soll. Private Links zwischen fremden Dokumenten werden in der Ablage im Client der Benutzerin, öffentliche Links auf dem Zugriffsserver der Benutzerin abgelegt.

Alle Links werden durch typographische Hervorhebungen oder bestimmte, hervorgehobene Zeichen am Anfang und Ende vom übrigen Text abgesetzt. Dadurch können auch verschachtelte Links dargestellt werden (s. Abbildung 20). Der Text, auf den ein Link verweist, wird ebenfalls kenntlich gemacht. Wenn mehr als eine festgelegte Anzahl von In-Links auf einen bestimmten Text zeigen, wird dies gemeldet, wenn der Text angefordert wird. Die Anzahl der ohne Nachfrage darzustellenden Links kann im Browser eingestellt (s. Abbildung 17), die Heraushebung von Links in einer eigenen Dialogbox nach Linktypen abgeschaltet werden (s. Abbildung 21). In einem Übersichtsfenster werden Links in Abhängigkeit von ihrem Typ ebenfalls farblich hervorgehoben (s. Abbildung 18). Diese Hervorhebung kann auch unterdrückt werden.

Wenn eine Benutzerin ein Dokument, auf das Links zeigen, löschen möchte, wird sie vom Transactions-Browser gewarnt. Wenn sie die Löschung dennoch ausführt, werden die Besitzerinnen der Dokumente, von denen die Links ausgehen, benachrichtigt. Diese Besitzerinnen der anderen Dokumente erhalten dann eine Meldung, wenn sie gerade mit dem System arbeiten oder sich das nächste Mal einloggen. Dadurch können sie ihre Dokumente anpassen, die einen Link auf das gelöschte Dokument enthalten.

Links können in Transactions auf ein beliebiges Dokument gelegt werden, entweder auf das ganze Dokument oder nur einen Teil von ihm. Sowohl Linkausgangspunkte also auch Linkziele werden wie typographische Auszeichnungen in Textverarbeitungsprogrammen beim Editieren mitgeführt. Für das System werden zeit- und ortsfeste Links unterschieden. Zeitfeste Links beziehen sich auf eine bestimmte Version eines Dokuments, ortsfeste Links auf die aktuelle Version.

Für das wissenschaftliche Schreiben sind zeitfeste Links besonders nützlich, weil eine Autorin gewöhnlich nur auf Dokumente verweist, die ihr schon bekannt sind (und nicht auf die unbekannteste neueste Version eines Dokuments). Ortsfeste Links sind hingegen wichtig, um überhaupt einen Link auf ein Dokument legen zu können, das noch in Arbeit ist. Eine Autorin kann mit ortsfesten Links z.B. ein Verzeichnis ihrer aktuellen Projekte erstellen.

Zeitfeste Links werden in den meisten Fällen auf eine Veröffentlichung gelegt, bei der die veröffentlichende Stelle (z.B. ein Verlag) sicherstellt, daß sie nicht mehr geändert wird oder auf ein Dokument, das schon eine Nachfolgedition hat, das also eigentlich auch nicht mehr geändert werden kann. Wenn eine neue Version eines Dokuments erzeugt wird, auf dem sich ein zeitfester Link befindet, bleibt der Link auf dem alten Dokument erhalten und wird nicht auf das neue kopiert.

Wenn sich das Zieldokument eines zeitfesten Links ändert, also eine Version mit neuer Nummer aus dem Dokument erzeugt wird, werden die Autorinnen aller Dokumente benachrichtigt, die einen Link auf das Zieldokument enthalten. Das System bietet diesen Autorinnen dann die Möglichkeit an, die Texte an beiden Enden eines Links zu vergleichen. Dafür werden zunächst die beiden Texte in zwei Fenstern nebeneinander dargestellt. Wenn eine Autorin Text aus einem anderen zitiert hat, kann sie so feststellen, wo sich der Ursprungstext geändert hat. Das System bietet ihr auch die Möglichkeit, die neue Version des Textes in ihr Dokument zu kopieren. Wenn eine Autorin Text inhaltlich übernommen hat, kann ihr das System beim Vergleichen allerdings nicht helfen.

Ein zeitfester Sprunglink kann analog zu ‚Copy and Paste‘ gelegt werden. Dazu wird zunächst die Stelle in einem Dokument (mit Tastatur oder Maus) markiert, von welcher der Link ausgehen soll. Danach wird ein zeitfester Link gewählt. Schließlich wird der Zieltext (im selben oder einem anderen Dokument) markiert und der Link durch ‚Einfügen‘ fertiggestellt.

Wenn eine Autorin einen ortsfesten Link auf ein nichtveröffentlichtes Dokument oder einen Teil von einem solchen legt, muß sie damit rechnen, daß das Linkziel sich ändert. Sie wird daher nur informiert, wenn das Linkziel gelöscht wird, also entweder der Dokumententeil oder das ganze Dokument, damit sie ihre Links auch löschen oder auf ein anderes Dokument legen kann. Ortsfeste Sprunglinks werden gesetzt, indem die Stelle markiert wird, von der der Link ausgehen soll und eine (ortsfeste) Linkart gewählt wird. Schließlich wird eine Stelle im Zieldokument gewählt und der Link durch ‚Einfügen‘ fertiggestellt.

Anmerkungen werden als spezielle Sprunglinks unterstützt, die von der Stelle in einem Text, auf die sich eine Anmerkung bezieht, auf einen Anmerkungstext in einem anderen Dokument verweisen. Dieses andere Dokument kann eine einzige oder mehrere Anmerkungen enthalten. Es gibt private und öffentliche Anmerkungen, wie es auch private und öffentliche Links gibt. Wenn ein Anmerkungslink in einem Dokument aktiviert wird, werden der ursprüngliche Text und der mit den Anmerkungen nebeneinander dargestellt. Die gewählte Anmerkung wird in einem eigenen Fenster auf dieselbe Höhe wie der Text gescrollt, auf den sie sich bezieht.

In Transactions gibt es keine Transklusion. Wie wir gesehen haben (s. S. 80), können zeitfeste Transklusionen durch das Kopieren des einzubindenden Textes und einen Sprunglink ersetzt werden. Die Benutzung von Links für die wörtliche Textübernahme wird in Transactions dadurch unterstützt, daß beim Kopieren von Text in ein Dokument aus einem anderen ein Link vom einkopierten zum Ursprungstext gelegt wird. Dieser Link kann entfernt werden.

Ortsfeste Transklusionen, die sich ändern, wenn das Ursprungsdokument geändert wird, erscheinen im Zusammenhang des wissenschaftlichen Publizierens nicht sinnvoll. Eine ortsfeste Transklusion würde Text in ein Dokument einbinden, den die Autorin des Dokuments noch nicht kennen kann. Sie erscheint nur für den Fall nützlich, daß ein eigener Text eingebunden wurde, der einen Fehler enthält, der dann nur einmal im Ursprungstext korrigiert werden muß. In Transactions muß dieser fehlerhafte Text überall dort geändert werden, wo der Text einkopiert wurde. Die entsprechenden Stellen können aber leicht gefunden werden, da In-Links im Ursprungstext auf sie verweisen.

In Dokumenten kann der Umfang von Textteilen als Stretchtext geändert werden (s. S. 65). Über Stretchtext ist es z.B. möglich, eine kurze und eine lange Beschreibung eines bestimmten Sachverhaltes in ein Dokument einzubinden, oder ein Abstract und eine Langfassung eines Textes. Stretchtext wird erzeugt, indem jeweils ein bestimmtes graphisches Symbol als Anfangs- bzw. Endbegren-

zer in das Dokument eingefügt wird. Zwischen diese beiden Symbole kann Text geschrieben werden, der ein- und ausgeblendet werden soll.

Ein Stretchtext hat zwei Zustände, von denen der aktuelle durch die Form der Begrenzungselemente oder durch eine farbliche Hervorhebung angezeigt wird. Zwischen den Zuständen kann durch Klicken auf eines der beiden Elemente gewechselt werden. Die Hervorhebung von Stretchtextelementen in einem Dokument kann abgeschaltet werden. Die beiden Zustände eines Stretchtextes können jeweils beliebig viel Text enthalten, ein Stretchtext kann geschachtelt werden und beliebige andere Links enthalten. Umgekehrt kann Stretchtext in beliebigen Linkenden enthalten sein.

Links können neben den Standardlinkarten Sprunglink und Stretchtext weitere Typen zugewiesen werden. Diese werden ähnlich benannten Zeichenvorlagen in Textverarbeitungsprogrammen bedient und jeweils aus einer der Standardlinkarten abgeleitet. Anmerkungs-, Erläuterungs-, Quell- und Veröffentlichungslinks sind bereits vordefinierte Ableitungen des Sprunglinks, die auf jeweils eigene Weise dargestellt werden.

#### 4.2.2.6 Publikation und Finanzierung

Bibliotheken und Verlage veröffentlichen Dokumente, indem sie sie auf ihren eigenen Servern speichern. Die Dokumente können auf diesen Servern nicht mehr geändert, es können jedoch Links auf sie gelegt werden. Veröffentlichungen werden in Verzeichnissen geführt. Diese bestehen aus Links auf Dokumente, zusammen mit Beschreibungen zu ihnen.

Dokumente auf beliebigen Servern können außerdem besonders gewürdigt werden, indem ein vordefinierter Sprunglink mit dem Namen ‚Veröffentlichung‘ von einem Dokument einer veröffentlichenden Stelle aus auf sie gelegt wird. Bei der veröffentlichenden Stelle kann es sich um einen Verein, eine Privatperson etc. handeln. Obwohl Autorinnen ein veröffentlichtes Dokument immer noch ändern können, solange es nicht auf einem speziellen Veröffentlichungsserver gesichert wurde, wird erwartet, daß sie eine neue Version erzeugen, wenn sie das Dokument weiterbearbeiten möchten.

Die Übertragung und Nutzung von Daten wird in Transactions nicht finanziell abgerechnet. Verlage können jedoch Veröffentlichungen für den allgemeinen Zugriff sperren und Zugriffspañwörter verkaufen. Die Programme des Hypertextsystems sollen auf vorhandenen Computern in Studiengängen und Rechenzentren eingesetzt werden können. Institutsserver sollen Nutzerinnen Speicherplatz zur Verfügung stellen, auf denen sie ihre Arbeiten anderen zugänglich machen können, wie es über ‚Seiten‘ auf Instituts-WWW-Servern heute auch schon möglich ist. Wissenschaftlerinnen erstellen ihre Texte auf Computern, die ihnen die Universität (bzw. das wissenschaftliche Institut, an dem sie arbeiten) zur Verfügung gestellt hat und legen sie auf Servern ihres Studienganges oder einer wissenschaftlichen Bibliothek ab.

Bibliotheksserver müssen neu angeschafft werden. Sie können Kosten zur Verwaltung und Lagerung von Diplomarbeiten u.ä. und des Fernleihverkehrs einsparen, jedoch zumindest in einer Übergangszeit nicht soviel, daß sich ihre Anschaffung finanziell rechnet. Auf lange Sicht wird die Abbestellung wissenschaftlicher Zeitschriften Kosten senken, wenn einige dieser Zeitschriften über das Hypertextsystem erscheinen, auch wenn diese Abbestellungen sowieso erfolgen und nicht notwendigerweise zu Investitionen in ein Hypertextsystem führen.

#### 4.2.2.7 Kompatibilität und technische Aspekte

Transactions basiert auf dem Internet als dem auf absehbare Zeit erfolgreichsten Computernetzwerk. Server und Clients sind über es verbunden, außerdem Server untereinander. Neue Server können einfach in das Netz eingefügt werden, wenn ein entsprechender Domainname (wie ‚uni-bre-

men.de') existiert. Server müssen also nicht bei einer zentralen Stelle registriert werden, wie es bei Xanadu und HyperWave der Fall ist.

Server kommunizieren untereinander mit einem eigenen Protokoll, über welches Linkinformationen abgeglichen werden. Aus technischer Sicht werden Links in Transactions-Servern extern zu Dokumenten verwaltet, in einer Datenbank, die in jeden Server eingebaut ist. Dadurch kann u.a. die referentielle Integrität der Hypertextdaten sichergestellt werden. Browser erhalten Linkdaten gewöhnlich in Dokumente eingemischt, sie können aber auch nur die zu einem Dokument gehörenden Links anfordern, um sie in einem Linkkontextfenster darzustellen.

Der Transactions-Browser kann direkt auf beliebige Server zugreifen. Benutzerinnen loggen sich jedoch gewöhnlich bei einem einzigen Server ein, der sich in der Nähe ihres Arbeitsplatzrechners befindet. (Beispielsweise im Rechenzentrum, wenn sie an einer Universität arbeiten.) Nur auf einem Server, auf dem sie bekannt sind und sich identifizieren ('einloggen'), können sie öffentliche Links und eigene Dokumente ablegen. Ohne Anmeldung auf einem Server können Benutzerinnen nur private Links und Dokumente in ihrer Ablage speichern. Um ein zügiges Antwortverhalten in dem verteilten System zu gewährleisten, enthalten Server und Clients jeweils einen Cache für die zuletzt benutzten Dokumente.

In Transactions kann auf HTML- und PostScript-Dokumente zugegriffen werden, die bereits auf FTP- und WWW-Servern veröffentlicht sind. Bestehende WWW-Server mit z.B. bibliographischen Daten können also auch im Transactions-Netz genutzt werden. Auf externe Dokumente können Links gesetzt werden, jedoch nur auf ganze Dokumente. Die Gültigkeit von Links auf externe Dokumente kann in Transactions technisch nicht sichergestellt werden.

Zur Zusammenfassung der obigen Spezifikation wird in der bereits vorgestellten Tabelle 3 Transactions (in der letzten Spalte) mit dem WWW, HyperWave und Xanadu verglichen (s. S. 100).

## 4.3 Entwurf von Transactions

### 4.3.1 Objektentwurf

Einige der obigen Überlegungen, v.a. bezüglich der verschiedenen Linkarten, sollen nun in einen Objektentwurf von Transactions überführt werden, der wieder als Objektdiagramm nach OMT dargestellt wird (s. Abbildung 22). Das Objektdiagramm dieses Entwurfes besitzt mit dem Diagramm aus der Analyse (Abbildung 4 auf S. 25) nur wenig Ähnlichkeit, weil die verschiedenen Material- und Dokumentenarten hier durch Hypertextelemente ersetzt worden sind (s. Abschnitt 4.2.1 für die Ersetzung im einzelnen).

Das Publikationssystem verwaltet als Hypertextsystem Zeichenblöcke<sup>3</sup> und Links, inklusive deren Speicherung in Clients und Servern. Die folgende Beschreibung der einzelnen Objekte des Diagramms in Form eines Datenverzeichnisses läßt sich aus der obigen Spezifikation ableiten:

*Account*: Eintrag einer Benutzerin auf einem Server. Erlaubt den Zugriff auf ihren privaten Datenbereich. Enthält ein Paßwort, daß für diesen Zugriff eingegeben werden muß.

*Benutzerin*: Person, welche in dem System liest oder schreibt. Besitzt einen Namen. Hat ein oder mehrere Accounts.

*Cache*: Unterklasse von 'Datenbank'. Speichert Daten, auf die zuletzt zugegriffen wurde, im Client oder Server für schnellen erneuten Zugriff.

*Client*: Unterklasse von 'Station'. Netzwerkeinheit, von der aus auf das Netz zugegriffen wird.

*Datenbank*: Speichert 'private Datenbereiche' von Benutzerinnen.

3 Der Einfachheit halber wird von der Unterstützung von graphischen und anderen nichttextlichen Daten abgesehen, obwohl sie im Prinzip möglich sein sollte.



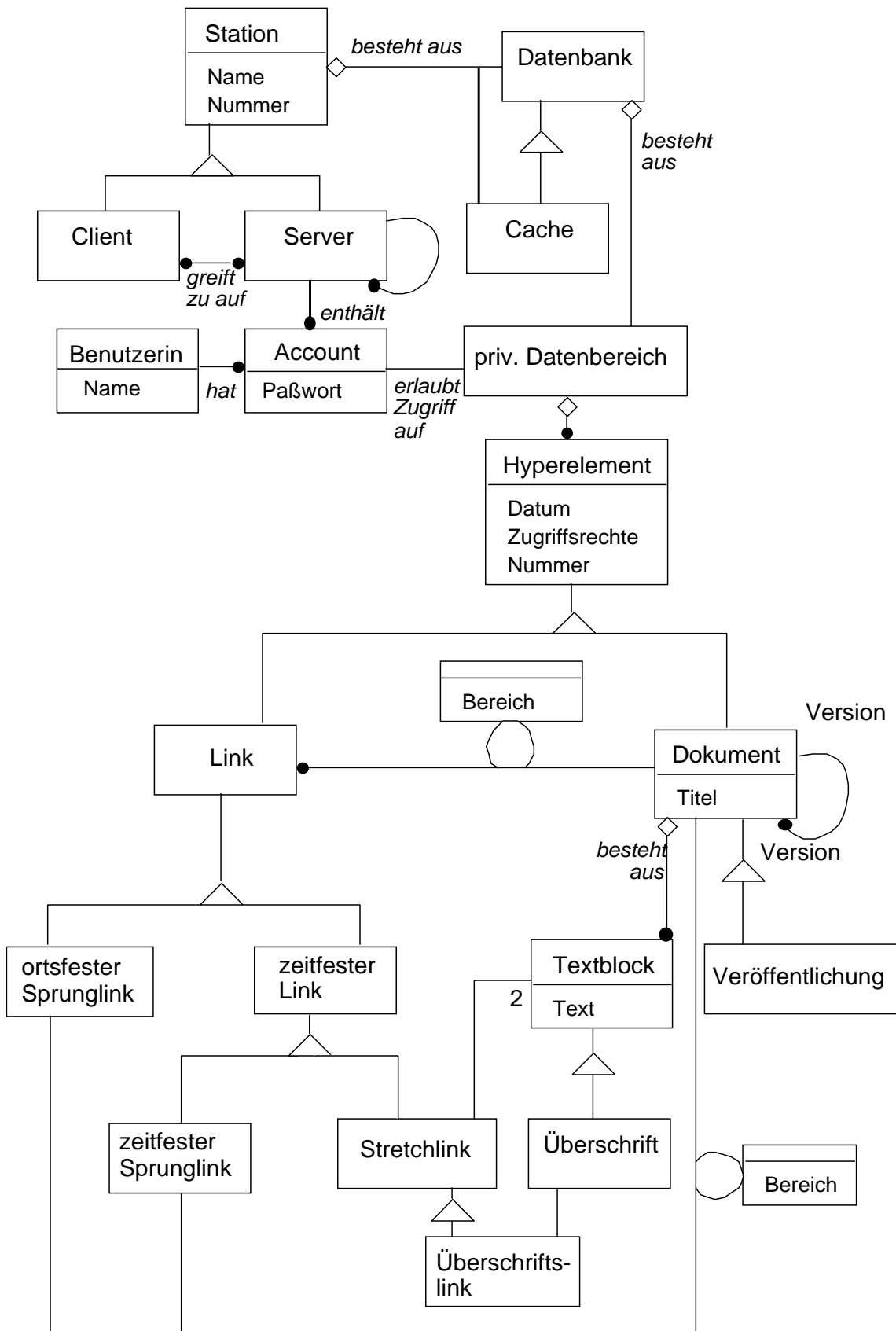


Abb. 22. Objektmodell für das Publikationssystem Transactions

*Dokument*: Unterklasse von ‚Hyperelement‘. Enthält einen Titel, besteht aus Textblöcken und ist mit alten Versionen von sich verbunden.

*Hyperelement*: Abstrakte Klasse. Gemeinsame Basis für ‚Link‘ und ‚Dokument‘. Enthält das Datum der Erstellung des Elements, Zugriffsrechte und eine eindeutige Nummer.

*Link*: Unterklasse von ‚Hyperelement‘. Bezieht sich auf einen Bereich in einem Dokument. Dieser Bereich wird als Position und Länge angegeben.

*ortsfester Sprunglink*: Unterklasse von ‚Link‘. Erlaubt den Sprung von einem Bereich in einem Dokument zu einem anderen Bereich in demselben oder einem anderen Dokument.

*Priv. Datenbereich*: Private Daten einer Benutzerin (oder des Systems). Zugriff nur über das entsprechende Account.

*Server*: Unterklasse von ‚Station‘. Netzwerkeinheit, auf der Daten für den Zugriff durch Clients gespeichert sind.

*Station*: Hardwareeinheit mit eindeutigem Namen und eindeutiger Nummer, die eine Datenbank und einen Cache enthält.

*Stretchlink*: Unterklasse von ‚zeitfester Link‘. Umfaßt zwei Textblöcke in demselben Dokument, aus denen die Benutzerin einen zur Darstellung auswählen kann.

*Textblock*: Verwaltungseinheit für Text. Dokumente bestehen aus Textblöcken.

*Transklusion*: Unterklasse von ‚ortsfester Link‘. Bindet Text aus einem Dokument in dasselbe oder ein anderes Dokument ein.

*Überschrift*: Unterklasse von ‚Textblock‘. Text, der hervorgehoben dargestellt wird und zu dem ein Überschriftslink gehört.

*Überschriftslink*: Unterklasse von ‚Stretchlink‘. Hat einen Überschriftstext. Enthält im zusammengefalteten Zustand keinen zusätzlichen Text und im ausgefalteten Zustand die in der Gliederungshierarchie unter der Überschrift liegenden Textteile (also Überschriften und Textblöcken) auf verschiedenen Gliederungsebenen. Die Gliederungsfunktion wird intern über Stretchtexte implementiert, auch wenn die Gliederungsteile anders bedient werden.

*Veröffentlichung*: Unterklasse von ‚Dokument‘. Ist im Gegensatz zu einem Dokument nicht änderbar.

*zeitfester Link*: Unterklasse von ‚Link‘. Ein Link, der weiterhin auf eine bestimmte Version eines Dokuments zeigt, auch wenn eine neue Version eines Dokuments erstellt wird.

*zeitfester Sprunglink*: Unterklasse von ‚zeitfester Link‘. Erlaubt den Sprung von einem Bereich in einem Dokument zu einem anderen Bereich in demselben oder einem anderen Dokument.

#### 4.3.2 Funktionaler Entwurf

Um die technischen Aspekte von Transactions zu entwerfen, muß über die Funktion von Clients und Servern entschieden werden, über Algorithmen für die Datenverwaltung, über das Protokoll zwischen Clients und Servern und zwischen Servern und über die Datenformate von Texten und Links. Ein vollständiger Entwurf würde jedoch den Rahmen dieser Arbeit sprengen.

Vor allem ist fraglich, ob die Implementation eines völlig neuen Hypertextsystems angebracht ist, selbst wenn es zum WWW abwärtskompatibel wäre, das ja heute schon für eine einfache Art des wissenschaftlichen Publizierens eingesetzt wird. Es müßte sich nämlich nicht nur gegenüber dem WWW selbst behaupten, sondern auch gegenüber HyperWave, welches wichtige Funktionalitäten bereits beinhaltet und immerhin eine gewisse Verbreitung an Universitäten gefunden hat, die auch die potentiellen Hauptanwenderinnen von Transactions wären.

HyperWave kann technisch in den meisten Punkten als Vorbild für ein Hypertext-Publikationssystem dienen. Seine Browser warten mit einer einheitlichen Oberfläche für den Zugriff auf verschie-

dene Datentypen auf, die Kompatibilität zum WWW wird in beiden Richtungen unterstützt, Links können (vorgegebene) Typen haben, auf Dokumententeile zeigen und in einem Linkkontextfenster dargestellt werden. Weiterhin gibt es einen Cache-Mechanismus und es wird ein spezielles Protokoll eingesetzt, über welches Verweise zwischen Servern abgeglichen werden.

Allerdings enthält der HyperWave-Browser keinen Editor, es gibt keine Versionsverwaltung und keine Transklusionen oder andere erweiterte Links. Weiterhin mußten die Navigationsübersichten ihre Benutzbarkeit in einem Netz von der Größe des WWW noch nicht unter Beweis stellen, was auch für andere Aspekte des Systems gilt (andererseits auch für die Spezifikation von Transactions). Schließlich beharren die Entwicklerinnen von HyperWave immer noch auf Eigenheiten wie dem Textformat HTF, die sich nicht durchsetzen werden.

#### 4.4 Weiterentwicklung des WWW

Vor dem Hintergrund der Spezifikation von Transactions soll nun beschrieben werden, wie das WWW weiterentwickelt werden kann, um wissenschaftliches Publizieren im Sinne dieser Arbeit zu erlauben. Dazu wird untersucht, ob Forderungen aus Abschnitt 4.2.1, die das WWW nicht erfüllt, dem Geiste nach doch mit ihm umgesetzt werden können, also unter Berücksichtigung der ursprünglichen Anforderungen aus Abschnitt 2.5.2. Die jeweilige Umsetzung kann technisch oder organisatorisch erfolgen.

Um möglichst vielen Benutzerinnen den Zugriff auf technische Erweiterungen zu erlauben, setzen die meisten der hier beschriebenen an den WWW-Servern an, auf denen die wissenschaftlichen Dokumente abgelegt werden. Dadurch kann auf die Dokumente über unveränderte WWW-Browser zugegriffen werden. Teilweise sind jedoch auch Erweiterungen auf Browserseite nötig, v.a. für die Erstellung von Dokumenten. Durch die Möglichkeit, in einem Browser einen Proxy-Server einzutragen (s. S. 41), können bei Zugriff über ein Proxy-Programm, das auf demselben Computer wie der Browser läuft, nicht nur die Adressen von WWW-Dokumenten gespeichert werden, die der Browser anfordert, sondern auch die Dokumente selbst vor Lieferung an den Browser geändert werden, z.B. durch das Einfügen von Links.

Manche Erweiterungen können schließlich nur durch ein eigenständiges, größeres Programm umgesetzt werden. Dabei handelt es sich um ein Ablage-Verwaltungsprogramm als Browser und Editor, wie es für Transactions gefordert wurde. Ein solches Programm hätte viele Vorteile, selbst wenn Server kaum oder gar nicht geändert würden. Die Ablage dient nämlich einer Benutzerin zur Verwertung fremden und zur Erstellung eigenen Textes auf ihrem Arbeitsplatzrechner.

Einerseits sollen die im folgenden beschriebenen Erweiterungen als Editoren und kleine Werkzeuge das Schreiben von Dokumenten erlauben, wobei es sich auch um Makros für erweiterbare Texteditoren handeln kann. Andererseits sollen bestehende, erweiterbare Browser/Editoren durch entsprechende Plug-ins das komfortable Lesen und Schreiben von Dokumenten gestatten.

##### 4.4.1 Umsetzungen der Anforderungen durch ein erweitertes WWW

In diesem Unterabschnitt wird untersucht, wie die Anforderungen an ein Hypertext-Publikationssystem aus Abschnitt 4.2.1 unter Berücksichtigung der einzelnen Phasen der Erstellung eines Dokuments (s. Abschnitt 2.5) durch Erweiterung des WWW umgesetzt werden können. Technische Einzelheiten folgen in Abschnitt 4.4.2.

###### 4.4.1.1 Texte und Verbindungen zwischen ihnen

Im WWW sind Dokumente gleichzeitig Hypertextknoten, wie für ein wissenschaftliches Publikationssystem gefordert. Zur Identifikation besitzen die Dokumente einen Titel und können bei den meisten Servern verständlich benannt werden.<sup>4</sup> Neben ihrem Titel und Zugriffsnamen (z.B. einer

Signatur in einer Bibliothek) besitzen wissenschaftliche Dokumente jedoch noch andere wichtige Merkmale: den Namen ihrer Autorin, ihr Publikationsdatum und ihre Sprache. Diese Merkmale werden im WWW nur teilweise unterstützt.

Über das ‚Link‘- oder das ‚Meta‘-Element kann im Kopf eines HTML-Dokuments seine Autorin angegeben werden (s. S. 36). Diese Angabe wird von den meisten Browsern ignoriert. Ein spezielles Ablageprogramm für das WWW kann jedoch so programmiert werden, daß es alle Metadaten eines Dokuments auf Anfrage zusammen darstellt.

Der Name der Autorin kann in HTML außer im ‚Meta‘- über das ‚Address‘-Element im Körper (dem ‚Body‘-Abschnitt) eines Dokuments angegeben werden, also im eigentlichen Dokumententext. Das ‚Address‘-Element bewirkt eine einfache typographische Hervorhebung des Textes, den es umschließt, z.B. durch Kursivdruck. Für wissenschaftliche Dokumente im WWW ist eine Übereinkunft sinnvoll, im ‚Address‘-Element am Anfang des Hauptteils des Dokuments ihre Autorinnen anzugeben. Die Namen wären dann an ähnlich prominenter Stelle wie in gedruckten Artikeln.

Das letzte Änderungsdatum einer HTML-Datei wird einem Browser zusammen mit der Datei vom Server übermittelt. Die Benutzerin kann es sich in einem Informationsfenster ansehen (NETSCAPE COMMUNICATIONS o.J.c, Abschn. „Document Information“). Dabei handelt es sich jedoch nicht notwendigerweise um das Datum, an dem die entsprechende Autorin das Dokument inhaltlich geändert hat, sondern um das Datum des letzten Schreibzugriffs. Als Publikationsdatum kann es daher nur bedingt angesehen werden.

In einem Publikationssystem auf Basis des WWW sollen Dokumente das letzte Änderungsdatum daher im Textkörper enthalten. (Diese Forderung ist zwar offensichtlich, wird aber von den wenigsten Autorinnen von WWW-Dokumenten beachtet.) Außerdem soll aus dem Bezeichner eines Dokuments seine ‚Auflage‘ hervorgehen, weil die meisten HTML-Dokumente immer wieder überarbeitet werden. Als Auflagenkennung könnte das Veröffentlichungsdatum an den Grundnamen des Dokuments angehängt werden, was im WWW teilweise praktiziert wird. Eleganter, und daher für das Publikationssystem vorgesehen, ist es jedoch, die Auflage- bzw. Versionsnummer an den Dokumentennamen anzuhängen. (Auf die Versionsverwaltung wird später noch eingegangen.) Das Datum muß dann aus dem Dokumententext hervorgehen, wie bereits gefordert.

Viele WWW-Serverprogramme können Dokumente nach in Browsern eingestellten Sprachen auswählen und liefern. Leider gibt es keinen Mechanismus, um von einem Dokument zu einer seiner Übersetzungen zu gelangen, was die Sprachverwaltung für das wissenschaftliche Arbeiten weitgehend unbrauchbar macht. In WWW-Dokumenten, von denen Übersetzungen vorliegen, befinden sich jedoch oft Hinweise mit Sprunglinks auf die Übersetzungen. Für wissenschaftliche Texte soll festgelegt werden, wo sich diese Links auf Übersetzungen eines Dokuments befinden sollen. Wenn sie sich am Ende eines Dokuments befinden, ist nicht sofort klar, ob es zu dem Dokument Übersetzungen gibt. Es wird daher vorgeschlagen, Hinweise auf Übersetzungen an den Anfang des Dokuments zu setzen, wie es viele Autorinnen von WWW-Dokumenten bereits tun.

Verweise zwischen Dokumenten werden im WWW als Links dargestellt. Es gibt davon jedoch nur eine Art, Sprunglinks von jeweils einem Bereich in einem Dokument zu einem ganzen anderen Dokument oder einem benannten Anker in ihm (s. RAGGETT 96b). Ein Zielanker wird von WWW-Browsern nicht (z.B. farblich) hervorgehoben und muß von der Autorin des Zieldokuments in es eingefügt worden sein. Es ist also nur möglich, auf eine Stelle in einem Dokument zu verweisen, die die Autorin des Dokuments dafür vorbereitet hat.

Wenn ein Dokument allerdings durch den Server, auf dem es gespeichert ist, oder einen Proxy zwischen Browser und Server geändert werden kann, bevor es bei einer Benutzerin ankommt, kann

4 Eine Ausnahme stellen Dokumente auf MS-DOS-basierten Servern dar, weil in diesem Betriebssystem Dateinamen nur acht Zeichen lang sein dürfen, gefolgt von einem Dateityp, der drei Zeichen lang ist.

ein solcher Anker auch erst beim Laden eingefügt werden. Dann kann auch ein gewöhnlicher WWW-Browser zu einer bestimmten Stelle in einem Dokument springen. Dafür muß jedoch die Position in dem Dokument, an die gesprungen werden soll, dem einfügenden Server oder Proxy bekanntgegeben werden. Eine Position wird dazu an die URL des Dokuments angehängt, auf das sie sich bezieht.

Schwieriger als die Adressierung eines Bereiches in einem fremden Dokument ist es, seine Adresse herauszusuchen. Dies erfolgt in Abhängigkeit der Art einer Adresse. Als Arten sind Zahlen und Suchstrings möglich. Zahlenadressen beziehen sich in einem bestimmten Dokument auf einen Bereich von Bytes (z.B. ‚Zeichen 5456 bis 6522‘), Zeilen (z.B. ‚Zeile 17 bis 28‘) oder von speziellen HTML-Elementen (PAM 97). Ein Beispiel für die Adressierung von Paragraphenelementen ‚<P>‘ wäre ‚den fünften bis siebten Paragraphen‘.

Wenn sich ein Dokument ändert, sind in den meisten Fällen alle Zahlenadressen im Textfluß hinter der Änderungsstelle falsch. Eine Suchstringadresse bezieht sich hingegen auf das erste Vorkommen eines bestimmten Textes, unabhängig von seiner Position. Dadurch ist bei dieser Adressierungsart die Chance größer, daß die Adresse bei einer Änderung des Dokuments korrekt bleibt. Auch eine Suchstringadresse kann jedoch falsch werden, wenn der adressierte Text gelöscht, geändert oder so verschoben wird, daß es nach der Änderung vor ihm im Dokument einen Text gibt, auf den der Suchstring auch paßt.

Suchstrings sind nicht notwendigerweise eindeutig, bei ihrer Erstellung muß immer überprüft werden, ob sie wirklich das Textstück adressieren, auf das sich eine Autorin beziehen möchte. Je länger der Suchstring ist, desto größer ist die Chance, daß er eindeutig ist und es nach einer Änderung des Dokuments bleibt. Zur Platzersparnis kann als Suchstring ein Anfangs- und Endstück des Textes genommen werden, der adressiert werden soll. Wenn dieser Text kurz ist, muß er in den meisten Fällen ganz als Adresse angegeben werden. In diesem Fall erscheint der Text eines kurzen Zitates zwar zweimal in einem HTML-Dokument, einmal im Dokumententext und einmal als Teil der Adresse der Quelle, generell reicht jedoch die Angabe eines Anfangs- und Endstückes zur Adressierung aus.

Einen Suchstring als Linkziel kann eine Benutzerin auch ohne die Hilfe von Programmen erstellen, am einfachsten durch Kopieren des Zieltextes und sein Einfügen in das Kommando für den Linkaufruf (das HREF-Attribut eines HTML-Ankerelementes), also durch ‚Copy and Paste‘. Sie muß sich allerdings vergewissern, daß der Suchstring eindeutig bezüglich des Dokuments ist. Wenn ein spezieller Browser für den Zugriff auf das System benutzt wird, können Adressen durch Markieren des entsprechenden Textes herausgesucht werden. Dann sind Positionsadressen, die aus Zahlen bestehen, besser, weil sie eindeutig sind.

Die Angabe einer Zeilennummer in dem Zieldokument als Sprungziel hat den Vorteil, daß diese Nummer mit den meisten Texteditoren leicht herausgefunden werden kann, dafür also kein spezieller oder erweiterter Browser nötig ist. Allerdings müssen zur Adressensuche die Zeilen im HTML-Sourcecode gezählt werden, weil WWW-Browser unabhängig von ihnen umbrechen. Vor allem sind die Zeilenendzeichen für HTML-Dateien nicht standardisiert, sie können sogar ganz weggelassen werden, wodurch ein HTML-Dokument aus nur einer ‚Zeile‘ bestehen kann. Bereiche von Dokumenten sollen daher für den hier besprochenen Ansatz durch Suchstrings oder nach Bytepositionen in den Dateien adressiert werden. Wie Adressen für diese Bereiche erstellt werden können, wird in Abschnitt 4.4.1.3 besprochen.

Im WWW besitzen Links keine Typen. Die Art eines Verweises wird in HTML-Dokumenten jedoch teilweise durch typographische Konventionen angedeutet, wie es bei Druckwerken auch getan wird. Ein Literaturverweis wird z.B. in runden (wie in diesem Text) oder eckigen Klammern eingeschlossen. Für Anmerkungen ist eine Hervorhebung ebenfalls sinnvoll, als Ersatz für ein Fußnotenzeichen bzw. das Schreiben auf dem Seitenrand. Für den hier besprochenen Ansatz wird vor-

geschlagen, die Art eines Verweises, den die Autorin eines Textes eingefügt hat, durch verschiedene Klammern anzudeuten, Quellverweise in eckigen und Anmerkungen (s. weiter unten) in spitzen Klammern unter Nennung der jeweiligen Autorin.

#### 4.4.1.2 Lesen

Als Erweiterung des WWW ist es mit dem vorgeschlagenen System leicht möglich, auf bestehende Dokumente zuzugreifen. Im WWW können nämlich existierende Dokumente auf FTP-Servern angefordert werden, z.B. im Postscript-Format. Dokumente eines anderen Formats als HTML werden von einem Browser nach dem Laden einem entsprechenden Programm zur Darstellung übergeben. Weil diese Dokumente über eine URL adressiert werden können, können auf sie (jeweils als Ganzes) auch Links gelegt werden.

Viele wissenschaftliche Texte liegen inzwischen selbst in HTML vor. Dadurch ist es durch die oben beschriebene Erweiterung möglich, auch Teile von Texten zu adressieren, wobei jedoch die Gefahr besteht, daß sich diese Dokumente ändern und Verweise dadurch ungültig werden. Für normale HTML-Dokumente, die sich nicht auf einem nach den in Abschnitt 4.4.2 beschriebenen Vorschlägen erweiterten WWW-Server befinden, muß also im Einzelfall abgeschätzt werden, ob es sinnvoll ist, einen Link auf einen Dokumententeil zu legen.

Um eine Reihe von über Links verbundenen Dokumenten leichter auswerten zu können, muß es möglich sein, sich die Verbindungen zwischen ihnen graphisch darstellen zu lassen. In einer solchen Darstellung können Dokumente auch ausgewählt werden, um sie zu drucken oder auf dem lokalen Rechner zu speichern. Es gibt für das WWW bereits Erweiterungen, die genau das ermöglichen: Mit dem schon beschriebenen HTML-Editor AOLPress (s. S. 43) können Dokumente gemeinsam gedruckt werden, die in einem Übersichtsfenster angezeigt werden. Allerdings werden Links zwischen den Dokumenten nicht in eine adäquate Form für Papier umgesetzt (z.B. zu Querweisen mit Seitenzahlen).

Das Browser-Hilfsprogramm ‚WebWhacker‘ für das WWW erlaubt es, ein Netz von ausgewählten Dokumenten auf der lokalen Festplatte zu speichern, also auch Dokumente, auf die die ausgewählten mit einem Link verweisen. Zu den gespeicherten Dokumenten und den Links zwischen ihnen wird von WebWhacker ein Baumdiagramm erstellt. Der WebWhacker ist mit einem WWW-Browser über die Proxy-Schnittstelle des Browsers verbunden (MENDELSON 96), ähnlich dem ‚OreO‘ (s. S. 41). Weil zu vermuten ist, daß die Internet-Hilfsprogramme immer besser werden, muß für die genannten Funktionen des gemeinsamen Ladens und Druckens von WWW-Seiten keine eigene Softwarekomponente geschrieben werden.

Aufgrund der Speicherung von Links in den Dokumenten selbst ist es allerdings im WWW nicht möglich, sich in einer Übersichtsdarstellung Dokumente anzeigen zu lassen, die *auf* ein vorliegendes Dokument verweisen. Im Kontext des wissenschaftlichen Arbeitens ist eine solche Funktion v.a. für das Finden von Sekundärliteratur zu einem Dokument nützlich.

In-Links können in das WWW eingeführt werden, indem auf Servern ein Mechanismus implementiert wird, durch den Benutzerinnen In-Links in Dokumenten von anderen Benutzerinnen setzen können. In einem speziellen Editor kann das Setzen eines In-Links gleichzeitig mit dem Setzen eines Out-Links auf eines dieser anderen Dokumente erfolgen. Normale WWW-Browser können In-Links darstellen, wenn der jeweilige Server des Dokuments sie ihnen in den Text des Dokuments als HTML-Links einmischt. Bei den entstehenden Dokumenten handelt es sich also um dynamische Dokumente, die zum Grundkonzept des WWW gehören (s. S. 40).

Damit In-Links in einem Dokument auf einem Server nicht ungültig werden, wenn sich das Dokument verändert, sollte eine solche Änderung gar nicht erst erfolgen. Stattdessen sollte eine geänderte Version eines Dokuments als neue ‚Auflage‘ mit einem entsprechenden Namenszusatz gespeichert werden, was später noch weiter ausgeführt wird.

Für eine Leserin muß es offensichtlich sein, ob sich eine Autorin an die Konvention des Änderns von Dokumenten nur über Auflagen hält. Die Leserin kann das an dem Namen eines Dokuments feststellen, indem sie prüft, ob er einen Auflagenzusatz enthält oder nicht. Wenn sich die Autorin nicht daran hält, aber die Datei trotzdem für das Setzen von Umkehrlinks freigegeben hat, kann ein Link immer noch auf die ganze Datei gesetzt und am Ende von ihr dargestellt werden.

Links auf ein veröffentlichtes Dokument können nur auf Anker im Dokument gelegt werden, die die Autorin des Dokuments dort eingefügt hat. Für die Einfügung eines Ankers muß ein Dokument geändert werden, was bei einer Veröffentlichung nicht erlaubt sein soll. Positionen in einem veröffentlichten Dokument müssen also ‚von außen‘ angegeben werden können, durch einen Suchstring oder eine Adresse.

#### 4.4.1.3 Auswerten

Zum Auswerten von Dokumenten gehört das Erstellen von Auszügen, Anmerkungen und eigenen kleinen Texten. Wie schon beschrieben, können Autorinnen WWW-Dokumente entweder in Text- oder speziellen HTML-Editoren erstellen. Grundsätzlich muß es auch im neuen System jeder Autorin freigestellt bleiben, wie sie Texte schreibt.

Es ist zu aufwendig, einen neuen Browser/Editor mit erweiterten Funktionen für alle gängigen Betriebssysteme zu entwickeln, wie er für unseren Zweck nötig wäre. Ein solcher Browser/Editor muß nämlich nicht nur die Leistungsmerkmale heutiger WWW-Browser und Textverarbeitungsprogramme bieten. Er muß weiterhin, wenn er von Benutzerinnen angenommen werden soll, mit neuen Erweiterungen von HTML und anderen Teilen des WWW schritthalten können. Wie schwierig das sein würde, läßt sich daran ersehen, wie kurz die Produktzyklen der Führer auf dem Browsermarkt, Netscape Navigator und Microsoft Explorer, sind. Es muß also in dem neuen System möglich sein, Dokumente und Links mit einem vorhandenen WWW- oder Texteditor zu erzeugen.

Auszüge aus einem veröffentlichten WWW-Dokument können durch Kopieren des entsprechenden Textbereiches aus der Quelle und Einfügen in ein neues Dokument in einem Editor erstellt werden. In Standard-WWW-Editoren würden dadurch die Adressen der entsprechenden Quellen nicht notiert werden. In einem Hypertextsystem kann ein Quellenverweis jedoch als Link von einem Auszug als Sprungausgang zum Originalabschnitt als Sprungziel gespeichert werden.

Solche Links können durch ein Programm oder eine Editorerweiterung erstellt werden, denen die Adresse eines zu exzerpierenden Dokuments mitgeteilt wird. Wenn eine Benutzerin immer nur aus dem letzten mit dem Browser angeforderten Dokument Textstücke kopiert, kann ein Programm dessen Adresse leicht ermitteln, falls der Browser über das Erweiterungsprogramm (als Proxy) Daten anfordert oder das Programm im lokalen Cache des Browsers nachsieht, in dem Dokumente zusammen mit dem Datum ihrer letzten Anforderung gespeichert sind.

In dem hier beschriebenen System können Auszüge mit Links zu ihren Quellen folgendermaßen zusammengestellt werden:

1. Es wird das Dokument in einen Browser geladen, aus dem Auszüge erstellt werden sollen.
2. In dem Dokument wird ein Textbereich mit der Maus markiert und in die Zwischenablage kopiert, durch dem Befehl ‚Copy‘ aus dem Menü ‚Edit‘ o.ä.
3. Es wird ein für die hier beschriebene Aufgabe erweiterter Texteditor oder ein eigenes kleines Programm aufgerufen. Im aufgerufenen Programm wird ein neues Dokumentenfenster geöffnet.
4. In dieses Fenster wird der kopierte Textbereich aus der Zwischenablage eingefügt, durch dem Befehl ‚Paste‘ aus dem Menü ‚Edit‘ o.ä.

5. Das zum Fenster gehörige Programm sieht in dem zuletzt mit dem Browser angeforderten Dokument nach, von welcher Stelle in ihm der eingefügte Textbereich stammt. Wenn die Suche der Position eindeutig gelingt, wird der Ausschnitt als Suchstringadresse für den Link übernommen.
6. Wenn der Textbereich mehrmals im Ursprungsdokument vorkommt, muß entweder der Suchstring geändert werden oder die Benutzerin den richtigen Textausschnitt aus einer Reihe von möglichen auswählen. Im letzten Fall wird ein Ausschnitt nicht über einen Suchstring, sondern über eine Byteposition adressiert.
7. Um den eingefügten Ausschnitt legt das benutzte Programm einen Link auf denselben Text im Ursprungsdokument, mit dem ermittelten Suchstring oder der Byteposition als Adresse. Der Link wird in eckigen Klammern eingeschlossen, um ihn als Quellverweis zu kennzeichnen, wie in Abschnitt 4.4.1.1 besprochen.
8. Wenn das Dokument mit den Auszügen auf einem Server gesichert wird, werden In-Links im Ursprungsdokument gesetzt bzw. erneuert. Das ist natürlich nur dann möglich, wenn der Server, auf dem sich das Ursprungsdokument befindet, das Setzen von In-Links unterstützt.
9. Es können mehrere Auszüge erstellt werden, indem der beschriebene Vorgang ab Punkt 2 wiederholt wird. Dadurch entsteht nach und nach ein neues HTML-Dokument mit Auszügen aus einem bestehenden Dokument, wobei von den Auszügen jeweils Links auf die Quellabschnitte zeigen.

Das Erstellen von Auszügen wird nach dem ‚Copy and Paste‘-Prinzip durchgeführt. Es kann natürlich auch durch das ähnliche ‚Drag and Drop‘ erfolgen. Außerdem können mehrere Links von Ausschnitten auf die jeweiligen Ursprungsabschnitte gemeinsam erstellt werden. Dazu werden Ausschnitte aus einem Dokument untereinander in eine einzige Datei kopiert und zwischen den Ausschnitten Trennzeichen eingefügt. Das Einfügen von Links kann dann von einem eigenen kleinen Programm vorgenommen werden, das nicht auf Benutzerinneneingaben angewiesen ist, wenn die Ausschnitte, die wieder selbst als Suchstringadressen genommen werden sollen, jeweils nur einmal im Dokument vorkommen.

Anmerkungen werden in Transactions durch Links dargestellt. Im Gegensatz zu einem Link von einem Auszug zu einer Quelle zeigt ein Anmerkungslink von einem Abschnitt in einem bestehenden Dokument zu einer Anmerkung. Ein Anmerkungslink muß also in das Dokument eingefügt werden, auf das er sich bezieht. Wenn das Dokument geändert werden darf, kann ein Anmerkungslink direkt in es eingefügt werden.

Wenn eine Benutzerin ein Dokument nicht auf seinem Server ändern darf, kann sie private Anmerkungen zu ihm erstellen, indem sie es in ihrer Ablage speichert und die Anmerkungen einfügt. Wenn sich das Ursprungsdokument ändert, sind ihre Anmerkungen ungefährdet, in ihrer Ablage befindet sich dann jedoch nicht mehr die neueste Version. Auf öffentliche Anmerkungen wird in Abschnitt 4.4.1.6 eingegangen.

Im WWW können Links nicht nur auf fremde Dokumente auf Servern, sondern auch auf eigene oder gesicherte fremde Dokumente im lokalen Dateisystem des Zugriffscomputers gelegt werden. Diese Links sind mit beliebigen WWW-Browsern benutzbar, weil es für lokale Dateien ein eigenes URL-Präfix gibt (s. Fußnote Nr. 5 auf S. 34).

Auch wenn lokale Verzeichnisse eigene und fremde Texte enthalten können, ergeben lokale Dateien in hierarchischen Verzeichnissen eines Betriebssystems noch keine Ablage, wie sie für diese Arbeit gefordert wurde (s. Abschnitt 4.2.1). Dafür sind hierarchische Verzeichnisse zu starr. Die gleichzeitige Verwendung von Links und Dateisystemverzeichnissen als Ordnungselemente scheint wenig hilfreich. Die beiden Arten von Ordnungselementen würden dann konkurrieren und Verzeichnisse würden einen zu starken Einfluß auf die Ordnung ausüben. Für ein Publikationssy-



stem müssen Dokumente in weiteren Hypertextdokumenten verzeichnet und Links zwischen Dokumenten in einem Übersichtsfenster dargestellt werden können.

Ein Ablageprogramm muß zumindest mit einem WWW-Editor zusammenarbeiten können. Wenn das Programm selbst einen Editor enthält, können Dateiverzeichnisse als Ordnungselemente ignoriert bzw. nur dazu genutzt werden, strukturierte Namen zu erzeugen. Dokumentennamen können dann in der Ablage als Separatoren die Verzeichnistrenner des jeweiligen Betriebssystems enthalten. Unter Unix könnten Namen also Schrägstriche enthalten. Durch einen solchen getrennt könnten z.B. ein Autorinnen- und der eigentliche Dokumentenname zu einem Gesamtnamen aneinandergesetzt werden. Das der Autorinnenname dann eigentlich ein Verzeichnisname des Dateisystems darstellt, ist für eine Benutzerin unwichtig, wenn sie nur über das Ablageprogramm auf die Dateien zugreift.

Es muß in dem Ablageprogramm möglich sein, lokal gespeicherte Dokumente nach Titel und Autorin zu finden. Formale Verzeichnisse nach diesen beiden Attributen können in einem Ablageprogramm jeweils bei der Speicherung eines Dokuments aufgefrischt werden. Dokumente müssen dann allerdings auch in diesem Programm gelöscht werden, damit sie wieder aus den Verzeichnissen entfernt werden. Alternativ können die beiden Verzeichnisse auch in regelmäßigen Abständen durch Untersuchung der ganzen Ablage erstellt werden.

Die Benutzung von Links als Quellenverweise wird im WWW dadurch erschwert, daß die Links dort nicht zeitfest sind. Wenn jedoch Dokumente so veröffentlicht werden, daß ihre Namen die jeweilige ‚Auflage‘ enthalten, können dadurch auch Links zeitfest gemacht werden, weil sie sich ja immer über seinen Namen auf ein Dokument beziehen. Bei der Besprechung der Veröffentlichung wird darauf noch näher eingegangen.

Obwohl es der Idee des Hypertextes (nach NELSON) entspräche, zeitfeste Einbindungen von Textteilen (Transklusionen) aus beliebigen Dokumenten für Zitate einzusetzen, erscheinen sie im WWW nicht sinnvoll. Sie können nämlich leicht durch das Einkopieren des entsprechenden Textes und einen Link auf das Original ersetzt werden. Nach Definition ändert sich zeitfest eingebundener Text nie, er muß daher auch nicht von einem anderen Server als dem des einbindenden Dokuments angefordert werden.

Gegen Transklusionen im WWW spricht, daß sie im WWW nur aufwendig umgesetzt werden können. Dies gilt einmal für die Erstellung und Verbreitung von erweiterten Browsern, aber auch für die Netzbelastung von speziellen Transklusionsservern (s. PAM 96a).

#### 4.4.1.4 Texterstellung

Der Editor des vorgeschlagenen Ablageprogrammes soll das Erstellen von Text unter einer graphischen Oberfläche erlauben. In ihm sollen Links auf Textabschnitte und andere neue Strukturen direkt erzeugt werden können. Wenn Text mit anderen WWW-Editoren geschrieben wurde, muß er entsprechend nachbearbeitet werden, indem z.B. zu Links Positionsadressen hinzugefügt werden. Im WWW können Texte typographisch ausgezeichnet werden, was von WWW-Editoren besonders komfortabel unterstützt wird.

Um Texte montieren und überarbeiten zu können, müssen ortsfeste Links und Versionen von der Ablage verwaltet werden. Zielanker von ortsfesten Links können in Standard-HTML als zweiteiliges Element angegeben und daher (wie typographische Auszeichnungen) beim Editieren mit dem entsprechenden Text mitbewegt werden. Sie eignen sich also für Links, die auf ein Dokument gelegt werden, solange es nicht veröffentlicht ist und noch geändert werden kann, weil sie in das Dokument selbst eingefügt werden müssen.

Im WWW gibt es keine Versionsverwaltung. Für die Erstellung von WWW-Dokumenten in HTML- oder Texteditoren können jedoch existierende Werkzeuge zur Versionsverwaltung von Programmsourcecode, z.B. RCS (s. S. 68ff), eingesetzt werden, weil HTML wie Programmsource aus

Zeilen in ASCII besteht.<sup>5</sup> Das Ablageprogramm soll die Verwaltung von Versionen lokaler Dokumente erlauben, die wie die Versionsverwaltungsfunktion in Transactions bedient wird (s. Abschnitt 4.2.2.4 auf S. 107).

Es ist kein WWW-Browser oder -Editor mit Gliederungsfunktion bekannt. Sie ist jedoch für das Schreiben von Texten nützlich und soll daher in einem Editor für die Benutzung des hier beschriebenen Publikationssystems vorhanden sein. Die Gliederungsfunktion wirkt auf die Überschriftselemente in HTML und erlaubt ihr Verschieben, sowohl vertikal (mit dem Textfluß) als auch horizontal (als Wechsel der Ebene einer Überschrift). Jeweils unter einer Überschrift liegende Überschriften und Absätze sollen in der Gliederungsansicht eines Dokuments ausgeblendet werden können.

Eine Gliederungsfunktion ist bereits für das Bearbeiten von HTML-Sourcecode sinnvoll, es kann also auch ein Texteditor mit HTML-Unterstützung in dieser Hinsicht erweitert werden. Die Funktion könnte auch in ein eigenes Programm eingebaut werden, weil hauptsächlich Überschriften dargestellt werden müßten. Allerdings entfaltet eine Gliederungsfunktion ihre echte Nützlichkeit erst in einem WYSIWYG-Editor, in dem zwischen ihr und normalem Schreiben übergangslos gewechselt werden kann, wie es in einigen Textverarbeitungsprogrammen auch möglich ist.

In Dokumenten für das neue System könnte ein Stretchlink über ein neues HTML-Element eingebunden werden, das sowohl die Kurz- als auch die Langfassung des jeweiligen Textes enthält. Dokumente mit einem solchen Stretchtext-Element könnten jedoch nur schwer so umgeformt werden, daß mit vorhandenen WWW-Browsern auf sie zugegriffen werden kann.<sup>6</sup> Stretchtexte wurden auch für Transactions nur in der eingeschränkten Form spezifiziert, daß sie die Auswahl zwischen zwei Formen eines bestimmten Textabschnittes erlauben, v.a., damit zwischen einer Kurz- und Volldarstellung eines Dokuments gewechselt werden kann.

Insgesamt scheint es zu aufwendig, ein eigenes Stretchtextelement für das Publikationssystem auf Basis des WWW einzuführen, das zu wenig Nutzen verspricht. Als Ersatz kann für die geplante Hauptanwendung ein Textabschnitt in HTML als ‚Abstract‘ gekennzeichnet werden, durch das gleichnamige Element. Dadurch wird der Abschnitt typographisch hervorgehoben. Es kann außerdem ein Link von einem Kurzttext aus auf das vollständige Dokument gelegt werden. Schließlich kann in einem erweiterten Browser in der Gliederungsfunktion Text unter einer Überschrift versteckt und wieder angezeigt werden.

#### 4.4.1.5 Veröffentlichung

Um Materialien und Dokumente der Öffentlichkeit zur Verfügung zu stellen, kann eine Autorin sie auf einen Server in ihrem Institut etc. legen. In neueren WWW-Editoren kann die Übertragung zu einem Server durch ein einziges Kommando ausgelöst werden, ohne daß ein externes Programm aufgerufen werden muß.

Autorinnen können für ihre Dokumente auf WWW-Servern nur bedingt Zugriffsrechte einstellen, weil diese Rechte (bei UNIX) in zentralen Konfigurationsdateien des Servers stehen. WWW-Verzeichnisse einer Benutzerin können jedoch von der Verwalterin des Servers so eingestellt werden, daß Dokumente in ihnen von allen gelesen, aber nur von der Eigentümerin selbst geschrieben und gelöscht werden können, nachdem sie ein Paßwort eingegeben hat.

Veröffentlichungsserver des hier beschriebenen Systems sollen formal/mechanische Verzeichnisse nach Autorinnennamen und Titeln von Dokumenten unter Standardnamen (z.B. ‚Autorinnenverzeichnis‘ und ‚Titelverzeichnis‘) anbieten. Diese Verzeichnisse können automatisch

<sup>5</sup> Mit RCS verwaltete WWW-Seiten kann man oft daran erkennen, daß sie zwischen den Begrenzern ‚\$Id:‘ und ‚\$‘ Versionsinformationen enthalten.

<sup>6</sup> Dafür müßte zu einem Dokument eine dynamische Ausführung in Abhängigkeit vom Zustand aller Stretchtexte im Dokument generiert werden. Bei Aktivierung eines Stretchtextelementes müßte das ganze dynamische Dokument neu erzeugt werden.

beim Übertragen eines Dokuments und von Zeit zu Zeit durch Überprüfen des ganzen Servers auf dem laufenden gehalten werden. Die Erstellung von anderen Verzeichnissen auf großen Veröffentlichungsservern stellt eine bibliothekarische Aufgabe dar.

Selbst wenn ein Dokument nicht mit Hilfe eines Versionsverwaltungssystems erstellt wurde, kann immer noch auf einem Server seine Version (Auflage) im Dateinamen angegeben werden. Wenn die Autorin ein Dokument auf einem Server nie ändert, sondern stattdessen eine neue Auflage mit einer anderen Versionsangabe erzeugt, können zeitfeste Links auf Dokumente gelegt werden. Die Version eines Dokuments kann als Versionsnummer an seinen Namen angehängt werden. Zu Versionen desselben Grunddokuments muß ein Dokument ohne Versionszusatz aufrufbar sein, das einen Link auf die neueste Version des Dokuments enthält, damit diese leicht gefunden werden kann.

Ähnlich wie in Transactions kann es in einem erweiterten WWW vorkommen, daß zwei Autorinnen dasselbe Dokument bearbeiten. Ein Konflikt tritt erst dann auf, wenn eine Autorin eine neue Version (Variante) des Dokumentes einer anderen Autorin überschreiben möchten. Der Konfliktfall ist leicht feststellbar, weil dann ein Dokument mit einem anderen mit gleicher oder niedrigerer Versionsnummer überschrieben werden soll. Ein erweiterter Server kann eine Benutzerin also warnen, wenn sie auf ihm ein Dokument mit einer Variante überschreiben möchte. Es soll ihr dann möglich sein, die beiden Versionen, die miteinander in Konflikt stehen, zu mischen, wie für Transactions bereits beschrieben wurde (s. Abschnitt 4.2.2.4).

Veröffentlichungsserver können durch Bibliotheken, Fachgesellschaften, Verlage etc. unterhalten werden. Für Bibliotheken macht sich der Aufwand für ihre Unterhaltung teilweise dadurch bezahlt, daß Server Pflichtveröffentlichungen aufnehmen können.<sup>7</sup> Fachgesellschaften und Verlage können durch die bereits erwähnte Zugriffsrechtsverwaltung von WWW-Servern den Zugriff auf bestimmte Dokumente nur nach Eingabe eines Paßwortes erlauben, wobei sie dieses Paßwort verkaufen und in gewissen Abständen wechseln können.

#### 4.4.1.6 Begutachtung

Private Bewertungen von veröffentlichten Dokumenten können in dem vorgestellten Publikationssystem als Anmerkungen erstellt werden, analog zu privaten Anmerkungen (s. Abschnitt 4.4.1.3). Begutachtungen im heutigen Peer Reviewing-Prozeß sind nichtöffentlich und werden von einzelnen Begutachterinnen unabhängig voneinander geschrieben. Sie können also in einem Hypertextsystem als persönliche Anmerkungen abgebildet werden, die an die entsprechende Stelle, z.B. eine Redaktion einer Zeitschrift, weitergereicht werden.

In einem Hypertextpublikationssystem ist es möglich, neben dem anonymen Peer-Reviewing Dokumente öffentlich zu bewerten. Eine Bewertung zu einem Dokument besteht dann aus einer öffentlichen Anmerkung. Es würde für eine Bewertung genügen, wenn sich eine Anmerkung nur auf ein ganzes Dokument beziehen kann und nicht auf Teile von ihm. Da es aber weitere Anwendungsmöglichkeiten der Anmerkung gibt (z.B. ein Hinweis auf eine neue Quelle) und sie ähnlich wie ein In-Link behandelt und verwaltet wird, der sich auf Bereiche in einem Dokument beziehen kann, soll sich eine Anmerkung in dem hier beschriebenen System ebenfalls auf einen Bereich beziehen können.

Die Schwierigkeit bei der Umsetzung von öffentlichen Anmerkungen liegt darin, daß eine Leserin eines Dokuments Hinweise auf vorhandene Anmerkungen erhalten muß, andererseits die Anmerkungen jedoch nicht zum Dokument selbst gehören und in den meisten Fällen von einer anderen Autorin erstellt worden sind. Hinweise auf Anmerkungen zu einem Dokument können als

<sup>7</sup> auch in Postscript, wenn die Dokumente mit normalen Textverarbeitungsprogrammen erstellt wurden.

Links ähnlich wie In-Links in es eingefügt werden (s. Abschnitt 4.4.1.2). Die Anmerkungen selbst können dann auf dem WWW-Server ihrer jeweiligen Autorin liegen.

Die obigen Ausführungen bezogen sich auf Anmerkungen zu veröffentlichten Dokumenten. Anmerkungen und In-Links zu nichtveröffentlichten Dokumenten auf einem Server, auf die trotzdem andere Benutzerinnen als die Autorin zugreifen können, können entweder gar nicht erzeugt oder in den Texten der Dokumente selbst gesichert werden, je nachdem, ob das Dokument für beliebige Benutzerinnen schreibgeschützt ist oder nicht. Wenn Anmerkungen oder In-Links in einem Dokumententext selbst gespeichert werden, sind sie so öffentlich wie das sie enthaltende Dokument.

#### 4.4.2 Technische Umsetzung

Erweiterungen des WWW können bei Servern oder Browsern ansetzen. Wenn nur Server erweitert werden, hat das den Vorteil, daß dann immer noch mit beliebigen Browsern auf sie zugegriffen werden kann, also auch von weniger gängigen Rechnerplattformen aus. Servererweiterungen bünden jedoch möglicherweise aufwendige Datenmanipulationen einer zentralen Instanz auf, wobei während einer Verarbeitung ein anfragender Browser unbeschäftigt ist. Außerdem führen Servererweiterungen zu einem insgesamt höheren Datenaufkommen, weil sie Daten vollständig zur Darstellung aufbereitet übermitteln.

WWW-Erweiterungen, bei denen Daten anders gehalten werden als im Standard-WWW, erfordern ein spezielles Programm auf Serverseite. Wenn die Daten auch noch editiert werden sollen, müssen zusätzlich Browser erweitert werden, weil sich das Editieren nur bedingt auf Standard-Browser-Funktionen (z.B. Formulare) abbilden läßt. Browsererweiterungen sind bereits dann nötig, wenn neue Strukturen nicht zufriedenstellend auf HTML abgebildet werden können. Anmerkungen können z.B. durch jeden WWW-Browser gelesen werden, wenn entweder ihr Text direkt oder ein Link auf ihn in das Dokument eingebettet wird, auf welches sie sich beziehen. So erscheinen Anmerkungen u.a. beim Ansatz des NCSA (LALIBERTE U. BRAVERMAN 95). Wenn sie jedoch in einem separaten Fenster dargestellt werden sollen, das synchron zum Hauptfenster gescrollt wird, oder jeweils in einem kleinen Popup-Fenster, dann ist eine Browsererweiterung unumgänglich.

Gewisse erweiterte Funktionen, die sich nur auf eine Benutzerin beziehen, z.B. private Anmerkungen, die Darstellung von Links zu Dokumenten und teilweise die Versionsverwaltung, können in einem entsprechend vielseitigen Clientprogramm erstellt und betrachtet werden. Andere Funktionen können erweiterte Server zwar beliebigen WWW-Browsern zur Verfügung stellen, benötigen dafür aber spezielle Daten, z.B. Adressen von Textabschnitten für Links. Solche Daten könnten in einem speziellen Clientprogramm für das Publikationssystem erstellt werden. Manche Funktionen können schließlich durch Konventionen umgesetzt werden, z.B. die Anbringung bibliographischer Informationen in einem Dokument und die Unterscheidung von Links nach ihren Typen. Autorinnen können auf die Einhaltung dieser Konventionen selbst achten und durch einen erweiterten Editor darin unterstützt werden. Verwalterinnen von Veröffentlichungsservern können ebenfalls auf die Befolgung der hier beschriebenen Regeln achten.

Die genannten Vorteile können das Hauptargument gegen die Eigenentwicklung eines Editors und Browsers, daß sie nämlich mit der schnellen Entwicklung des WWW und seiner Standards nicht mithalten kann, kaum aufwiegen. Selbst dann nicht, wenn man bedenkt, daß viele Erweiterungen von HTML dem wissenschaftlichen Arbeiten kaum dienen. Sinnvoller erscheint es daher, einen bestehenden Browser/Editor zu erweitern. Vielversprechend in dieser Hinsicht ist Amaya (s. Abschnitt 3.2.4 auf S. 42), der vom W3C selbst hervorgebracht wurde.<sup>8</sup>

#### 4.4.2.1 Texterstellung

Für ein Publikationssystem ist ein kombinierter Browser/Editor unbedingt notwendig. Außerdem müssen Texte auf dem lokalen Rechner einer Benutzerin in einer Ablage gespeichert und verwaltet werden können. Der WWW-Editor Amaya vom W3C kann in dieser Richtung erweitert werden. Er enthält nämlich nicht nur eine Programmierschnittstelle (API), sondern wird auch im Source-Code veröffentlicht, kann also notfalls direkt geändert werden.<sup>9</sup> Die wichtigste Eigenschaft von Amaya für unsere Zwecke ist jedoch, daß er bereits Editierfunktionen eingebaut hat, für die das W3C angibt, sie auf dem neuesten Stand zu halten, also mit HTML weiterzuentwickeln.

Da es sich bei Amaya um einen WWW-Browser handelt, kann er sofort auf WWW-Server zugreifen. Dokumente können in ihm lokal erstellt werden. Sie werden dazu im Dateisystem eines Client-Rechners gespeichert und können später an einen Server übertragen werden, der den ‚PUT‘-Befehl kennt (s. S. 34). Wenn sie ihre Dokumente mit einem anderen Editor erstellt, kann eine Autorin sie entweder über das UNIX-Dateiübertragungsprotokoll FTP, über Form-based file upload (s. S. 37) oder die HTTP-Methode ‚PUT‘ auf den Server übertragen. Viele WWW-Editoren und -Server beherrschen inzwischen die PUT-Methode, so daß HTTP als Standardübertragungsprotokoll für die hier besprochene Erweiterung des WWW gewählt wird.

Für das hier beschriebene System muß Amaya um Möglichkeiten der Erzeugung und Darstellung von Links, der Eigentums- und Versionsverwaltung und der lokalen Ablage von Daten erweitert werden. Für einige dieser Funktionen gibt es Vorbilder. So unterstützt z.B. der UNIX-Texteditor Emacs bereits die Versionsverwaltung, indem er eine integrierte Benutzungsoberfläche für die Versionsverwaltungssysteme SCCS, RCS und CVS darstellt. Um die Gliederungsfunktion und andere Erweiterungen zu programmieren, muß über das API von Amaya auf die internen Datenstrukturen von Text (s. QUINT 97) zugegriffen werden, damit z.B. Überschriften verschoben und Absätze ausgeblendet werden können.

#### 4.4.2.2 Metadaten

Informationen über die Autorin und das Veröffentlichungsdatum eines Dokuments werden im hier vorgeschlagenen Publikationssystem in seinem HTML-Kopf zusammen mit seinem Titel gespeichert. Der Autorinnennamen und das Datum werden dabei als ‚Meta‘-Element angegeben, wodurch sie einerseits von Browsern ignoriert werden, die das Element nicht verstehen, andererseits in solchen Browsern immerhin im HTML-Quellcode des Dokuments gelesen werden können. Wenn ein Dokument für das hier vorgestellte System mit einem gewöhnlichen HTML-Editor erstellt wird, müssen die Metadaten ebenfalls in der Quellcode-Ansicht des Dokuments geschrieben werden.

Da die beschriebenen Daten im Dokumentenkopf stehen, können sie angefordert werden, ohne das ganze Dokument zu laden. Fast alle WWW-Server verstehen nämlich den HTTP-Befehl ‚HEAD‘ (s. S. 34), der diesen Kopf auch ohne den Dokumentenkörper liefert. Die Sprache eines Dokuments wird im WWW im Dokumentennamen angegeben (s. S. 40), muß also nicht im Kopf gespeichert werden.

#### 4.4.2.3 Zugriffsrechte auf Servern

Zugriffsrechte für Dokumente auf WWW-Server können (unter UNIX) nur in speziellen Konfigurationsdateien eingestellt werden, die jeweils beim Starten des Serverprogrammes gelesen werden. Auf diese Dateien haben gewöhnlich nur Systemverwalterinnen Zugriff. Die Zugriffsrechte können daher nicht kurzfristig für ein einzelnes Dokument gesetzt werden. Wenn eine Benutzerin allerdings

8 Wobei das W3C allerdings nur bedingt die tatsächlichen Standards für das WWW setzt, was eher durch Firmen wie Netscape und Microsoft geschieht.

9 Dies empfiehlt sich jedoch nicht, weil Amaya vom W3C ständig weiterentwickelt werden soll und daher ständig neu für das hier beschriebene System angepaßt werden müßte.

Zugriff auf das Dateisystem des Servers hat (weil sich ihr Rechner im selben LAN befindet), kann sie einzelne Dateien über die Zugriffsrechtsverwaltung des Dateisystems sperren. Sie kann dann jedoch selbst auch nur noch über das LAN auf die Dateien zugreifen, nicht über den Server.

Für ein Publikationssystem mit erweiterten WWW-Servern ist es sinnvoll, daß jede Benutzerin auf der obersten Ebene ihrer WWW-Verzeichnisse jeweils ein Dateiverzeichnis für die Zugriffsrechte ‚Schreiben für die Eigentümerin und Lesen für alle‘, ‚Schreiben und Lesen für alle‘ und ‚Schreiben für alle und Lesen nur für die Eigentümerin‘ anlegt, die durch entsprechende Servereinstellungen geschützt werden. (Diese Einstellungen kann nur die Systemverwalterin des Servers setzen.)

In dem Verzeichnis, das fremde Benutzerinnen nur lesen dürfen, kann eine Autorin Veröffentlichungen ablegen. Das Verzeichnis, in dem alle Benutzerinnen lesen und schreiben können, muß durch ein Paßwort geschützt werden (das wieder nur die Systemverwalterin setzen kann). In diesem Verzeichnis kann seine Eigentümerin mit anderen Autorinnen zusammenarbeiten, wenn diese das Paßwort kennen, selbst wenn sie nicht auf ihr LAN zugreifen dürfen.

Das Verzeichnis schließlich, in das andere nur schreiben dürfen, kann als ‚Briefkasten‘ benutzt werden: Beliebige Benutzerinnen können Dokumente für die Eigentümerin dort ablegen, aber nur die Eigentümerin selbst kann sie lesen. Wenn die Verwalterin eines WWW-Servers dessen Einstellungen auch kurzfristig ändert, können Zugriffsrechte für spezielle Anwendungen eingestellt werden. Beispielsweise kann ein Verzeichnis auf einem WWW-Server technisch so eingerichtet werden, daß eine Arbeitsgruppe in ihm gemeinsam lesen und schreiben kann.

#### 4.4.2.4 Versionsverwaltung

Die Versionsverwaltung auf ihrem Clientcomputer obliegt der Verantwortung einer Autorin. Sie kann entweder in einem erweiterten Browser/Editor wie bei Transactions erfolgen (s. Abschnitt 4.2.2.4) oder durch einen beliebigen Editor und ein Standard-Versionsverwaltungssystem wie RCS. Wie ein Versionsverwaltungssystem implementiert werden kann, ist bei der Beschreibung des Configuration Management kurz angeklungen (s. S. 68). Für Transactions wurde in Abschnitt 4.2.2.4 beschrieben, wie Dokumente zusammengeführt werden können.

Auf Servern sollen ebenfalls Versionen von Dokumenten verwaltet werden. Wenn über einen erweiterten Browser/Editor auf Server zugegriffen wird, kann es sich dabei um denselben Mechanismus wie für eine Ablage handeln, einschließlich des Zusammenführens von Dokumenten. Wenn eine Autorin keinen solchen Editor hat, muß sie selbst darauf achten, daß sie Dokumente unter passenden Versionsnummern auf Servern abspeichert. Sie muß also zumindest Unterversionsnummern erhöhen, wenn sie eine neue Version erstellt. Wenn sie dies nicht tut, wird sie vom Server gewarnt.

Die Erweiterung des WWW für die Versionsverwaltung ist die Aufgabe einer Arbeitsgruppe von Spezialistinnen für Versionsverwaltung in Hypertexten unter dem Namen ‚WEBDAV‘ (Working Group on Distributed Authoring and Versioning on the World Wide Web). Deren Mitglieder DURAND U. VITALI beschreiben für die HTTP Working Group die Notwendigkeit der Erweiterung des WWW um Versionsverwaltung und die funktionalen Anforderungen für deren Umsetzung (96, Abschn. 1). Die Versionsverwaltung wird u.a. deshalb als wichtige Erweiterung des WWW dargestellt, weil damit zwischen historischen und alternativen Versionen von Dokumenten geblättert werden kann, sie stabile Namen für sich ändernde Dokumente bereitstellt und generell verschiedene Versionen eines Dokuments unter einem Namen (dem Grundnamen ohne Versionsnummer) zusammenfaßt (ebd., Abschn. 2). Diese Anwendungen der Versionsverwaltung entsprechen also grundsätzlich den in dieser Arbeit beschriebenen.

Versionen eines WWW-Dokuments sollen in einem Versionsverwaltungssystem ‚eingefroren‘ werden, sobald die Arbeit an ihnen abgeschlossen ist. Ein eingefrorenes Dokument kann nicht mehr geändert werden, ohne daß eine neue Version erzeugt wird. Anfragen eines eingefrorenen Doku-

ments sollen immer dieselben Daten liefern. Alle versionsfähigen Clients sollen mit allen versionsfähigen Servern zusammenarbeiten können. Die Anforderung von Daten (z.B. über HTTP und FTP) und die Reservierung und Freigabe von Änderungsrechten sollen getrennt voneinander erfolgen. Versionsserver sollen mit bisherigen WWW-Daten und -Adressen zusammenarbeiten können, ebenso mit Browsern ohne Versionsfunktionalität (ebd., Abschn. 2).

Der Zugriff auf eine spezielle Version eines Dokuments soll über einen URL erfolgen, ebenso der auf die Gesamtheit aller Versionen eines Dokuments. Zu jedem Dokument mit verschiedenen Versionen soll es eine Standardversion geben. Diese, die erste, nächste und vorherige Version eines Dokuments sollen ebenfalls über URLs abgefragt werden können. Einem Client soll es möglich sein, den Versionsbaum eines Dokuments anfordern zu können. Zu einem vorliegenden Dokument soll neben dem Grund-URL auch seine Versionsnummer aufrufbar sein können. Dokumente sollen gesperrt und reserviert werden können. Schließlich gibt es in der vorgeschlagenen WWW-Erweiterung einen Befehl, um eine neue Version an der entsprechenden Stelle des Versionsgraphen einzufügen. Zu Versionen gehören Zusatzdaten, z.B. Kommentare über den Zweck der Erstellung dieser Version (ebd., Abschn. 4).

Die WEBDAV (Working Group on Distributed Authoring and Versioning on the World Wide Web) hat ausgehend von den Vorschlägen von DURAND U. VITALI (96) ein Protokoll entworfen, welches ebenfalls WEBDAV heißt und HTTP um Fähigkeiten für die Versionsverwaltung erweitern soll. Dieser Entwurf befindet sich jedoch noch im Entwicklungsstadium (GOLAND et al. 96).

VITALI U. DURAND stellen mit VTML (Versioned Text Markup Language) eine Erweiterung zu HTML vor, durch die eine Versionsverwaltung in das WWW eingebracht werden kann (95, 37). Ihr Ansatz sieht die parallele, asynchrone Bearbeitung verschiedener Versionen eines Dokuments durch mehrere Autorinnen vor, wie sie für Transactions bereits gefordert wurde. Dadurch können unabhängige Versionen entstehen, die durch einen nachgeschalteten Vorgang wieder zu einer neuen Version vereint werden können. Da von einer asynchronen Bearbeitung ausgegangen wird, können neue Versionen eines Dokuments offline auf den jeweiligen Rechnern der Autorinnen erstellt werden (ebd., S. 39f). Änderungen werden in VTML als Deltaoperationen gespeichert (s. S. 70). Wenn Deltas nicht in einem Editor erzeugt wurden, werden sie durch den Vergleich zweier Versionen ermittelt (ebd., 41).

VTML-Befehle werden als SGML-Kommentare in HTML abgelegt, so daß sie von Browsern ignoriert werden, die VTML nicht verarbeiten können. Von VTML-fähigen Browsern werden die Befehle von einem Parser zusammen mit dem entsprechenden Dokument analysiert. Versionsdaten werden entweder in dem HTML-Dokument gespeichert, auf welches sie sich beziehen, oder in einem eigenen Dokument. Alle VTML-Dokumente enthalten einen Namen und die aktuelle Versionsnummer, optional eine Angabe über die beteiligten Autorinnen.

Es können sich mehrere VTML-Dokumente in einer Datei befinden. Als Änderungsoperationen (Deltaoperationen) sind bisher Einfügen, Löschen und Zusammenführen vorgesehen. Diese Operationen enthalten jeweils Angaben über die Autorin der Änderung, die Version, die sie erzeugen, das Änderungsdatum, einen Status, welcher die Wichtigkeit der Änderung angibt und einen Kommentar. Deltas schließen den Text ein, den sie einfügen bzw. löschen (ebd., 42f).

VTML soll nach der Vorstellung seiner Entwickler nur mit Hilfe von Programmen erstellt und ausgewertet werden, nicht direkt von Endbenutzerinnen. Unterstützung dafür soll in einen Editor eingebaut werden. Für die Darstellung soll VTML als MIME-Typ registriert werden, so daß erweiterte WWW-Browser darauf reagieren können. Für herkömmliche Browser wandelt ein CGI-Skript eine Dokumentenversion in VTML nach HTML, als Voreinstellung die letzte Version. Eine bestimmte Version wird durch Anhängen eines Schrägstrichs und der Nummer der Version an den URL des Grunddokuments adressiert, z.B. „http://my.host/dir/file.vtml/3.1.1“. Zunächst gibt es nur einen Parser für VTML, ein Browser ist jedoch geplant (ebd., 47ff).

Die Existenz einer ‚Working Group‘, die eine Versionsverwaltungserweiterung für das WWW erstellen soll, läßt vermuten, daß es in absehbarer Zeit in dieser Hinsicht erweiterte Browser und Server geben wird. Zunächst muß jedoch ein entsprechendes Protokoll definiert werden. Dafür existieren bisher nur Ansätze. Für das hier beschriebene Publikationssystem wird die Adressierung von Versionen von VTML übernommen. Versionsnummern werden also nach einem Schrägstrich am Ende eines Dokumentennamens angegeben.

Technisch handelt es sich bei dem Dokumentennamen dann um ein Verzeichnis auf einem WWW-Server und bei einer Version um eine Datei in diesem Verzeichnis, deren Dateiname nur aus der Versionsnummer besteht. Wenn keine Versionsnummer angegeben wird, wird vom Server automatisch das Standard-Dokument des Verzeichnisses geliefert, das hier einen Verweis auf die neueste Version des durch das Verzeichnis dargestellten Dokuments enthält. Diese neueste Version liegt vollständig auf dem Server vor. Ältere Versionen werden durch Deltas gespeichert und auf Anfrage vom Server generiert.

#### 4.4.2.5 Links

WWW-Browser sind in der Lage, bei Aufruf eines Dokuments mit Angabe eines Zielankers im Dokument den betroffenen Text durch Scrollen am Anfang des Fensters darzustellen (s. S. 37). Das Springen zu einem Anker kann für eine Erweiterung von WWW-Servern ausgenutzt werden. Nachdem man sich nämlich eine Adressierung von Teilen eines WWW-Dokuments überlegt hat, kann ein Server während der Lieferung bei Angabe einer Adresse an der entsprechenden Stelle eines Dokuments einen Zielanker anbringen, mit der Adresse als Namen. Dadurch springt der Browser nach der Darstellung des Dokuments an die adressierte Stelle. Wenn ein Adreßbereich des Dokuments angegeben wird, kann der entsprechende Bereich zusätzlich farblich hervorgehoben werden, durch das Einfügen von HTML-Anweisungen durch den Server.

Es existiert im WWW bereits ein Ansatz für die Adressierung von Bereichen und Bereichsfolgen (Spannen und -mengen im Sinne NELSONS, s. S. 74) in beliebigen Daten, nicht nur in HTML-Dokumenten. LUOTONEN U. FRANKS stellen mit dem ‚Byte Range Protocol‘ eine Erweiterung der URL-Syntax vor (95). Dazu wird am Ende eines URLs ein Strichpunkt und das Schlüsselwort ‚bytes‘ angefügt, gefolgt von durch Kommata getrennte Bytebereiche, z.B. „http://host/dir/foo;bytes=0-99,500-1499,-200“ für die ersten 100 Bytes, gefolgt von 1000 Bytes ab dem Byte Nummer 500 und den letzten 200 Bytes des Dokuments (ebd.).

Die Syntax des Byte Range Protocols soll für den hier entwickelten Ansatz übernommen werden. Da ein Link jedoch nur auf eine Spanne und nicht auf eine Spannenmenge zeigen soll, werden alle Spannen nach der ersten ignoriert, falls mehrere angegeben wurden. Der Adreßbereich am Ende einer URL soll dabei vom Server ausgewertet werden, auf dem sich das adressierte Dokument befindet. Es sind spezielle Hilfsmittel nötig, um einer Autorin die Möglichkeit zu geben, einen Bytebereich in einem Dokument herauszufinden, weil z.B. Texteditoren gewöhnlich nur Zeilen- und Spaltenadressen in einem Dokument anzeigen können.

Wie beschrieben, können Auszüge und Anmerkungen erstellt werden, indem der Text, auf den sie sich jeweils beziehen (das Ziel für Links) angewählt und in einem speziellen Programm (oder einem erweiterten Editor) eingefügt wird. Wenn dieses Programm zu einem Programmpaket für die Benutzung des Publikationssystems gehört, hat es Zugriff auf die Daten eines Proxies im System (oder auf den lokalen Cache eines Browsers), um das aktuelle Dokument festzustellen und eingefügte Textstücke mit ihm in Verbindung zu bringen. Wenn kein solches Programm vorhanden ist, kann eine Autorin immer noch einen Suchstring als Linkziel angeben.



Dies sieht auch WALKER<sup>10</sup> für seine WWW-Erweiterung ‚Hack links‘ vor. Unter diesem bezeichnenden Titel beschreibt er „a crude mechanism“, über den eine Benutzerin in beliebigen HTML-Dokumenten Umkehrlinks (In-Links) zu eigenen setzen kann (95). WALKER hat die benötigten Programme offensichtlich nicht implementiert (ebd., Abschn. ‚Conclusion‘).

Umkehrlinks zu einem Dokument werden bei ‚Hack links‘ nicht in dem Dokument selbst, sondern in einer externen Datei zu ihm gespeichert. Um ein Dokument für das Setzen von Umkehrlinks freizugeben, legt seine Besitzerin eine Datei an, in die der Server die Links abspeichert (ebd., Abschn. ‚Design Constraints‘). Wenn von einem Textstück im Zieldokument mehrere Sprung- und Umkehrlinks ausgehen, sieht der Ansatz vor, daß nur ein Link dargestellt wird, dessen Aktivierung ein neues HTML-Dokument erzeugt, in dem alle Links in dem Textstück angezeigt werden. Aus diesen Links kann die Benutzerin dann einen auswählen (ebd., Abschn. ‚How it Works‘).

Umkehrlinks können beim Ansatz von WALKER entweder mit einem speziellen Editor oder einem WWW-Browser in einem HTML-Formular-Dokument (s. S. 37) erstellt werden, das der betroffene Server anbietet. Es ist vorgesehen, die Position eines Umkehrlinks über Suchstrings oder Wortnummern angeben zu lassen (ebd., Abschn. ‚How it Works‘). WALKER ist sich des Problems bewußt, daß sich ein Dokument ändern kann, zu dem Umkehrlinks existieren, wodurch seine In-Links ungültig werden können. Neue Linkpositionen können dann nur über Heuristiken gefunden werden, indem das Ursprungsdokument und seine neue Version maschinell verglichen werden (ebd., Abschn. ‚What if the Document Changes?‘).

Eine Besitzerin eines Dokuments kann allerdings durch einen Blick in die Linkdatei des Dokuments feststellen, ob überhaupt Links in es eingefügt wurden. Sie kann weiterhin überprüfen lassen, ob die entsprechenden Umkehrlinks auf noch existierende Dokumente verweisen. Wenn dies beides nicht der Fall ist, kann sie bedenkenlos das Dokument verändern.

WALKER ändert die beteiligten Elemente des WWW so wenig wie möglich, das Datenformat HTML und Browser überhaupt nicht. Die Erweiterung soll nur auf den Servern in Form von CGI-Programmen installiert werden müssen, in deren Dokumenten Links angebracht werden sollen. Dokumente selbst müssen nicht verändert werden (ebd., Abschn. ‚Design Constraints‘).

‚Hack Links‘ besteht aus einer Reihe von kleinen Programmen, die auf den entsprechenden Servern installiert und über CGI-Aufrufe aktiviert werden (ebd., Abschn. ‚How it Works‘). Ein Programm dient dazu, HTML-Dokumente zusammen mit den in sie eingefügten Links an einen Browser zu liefern. Dazu erstellt es aus einem Dokument eine dynamische (virtuelle) Variante, die aus den Daten des Ursprungsdokuments und In-Links an den entsprechenden Positionen besteht.

Ein weiteres Programm auf dem Server dient dem Erstellen von Back links an einer bestimmten Position eines Dokuments. Falls die Eigentümerin eines Dokuments eine Datei für die Aufnahme von Links zu dem Dokument angelegt hat, wird der Link dort eingefügt, wenn zusätzlich das Sprungziel des In-Links vorhanden ist. Wenn weder Linkdatei noch Linkziel gefunden werden, wird auf das Setzen eines Umkehrlinks verzichtet (ebd.).

Die Speicherung von Umkehrlinks außerhalb von Dokumenten hat den Nachteil, daß die Links bei jedem Liefern eines Dokuments vom Server in die Dokumentendaten eingemischt werden müssen. Die Abspeicherung von Links in einer eigenen Datei zu jedem Dokument hat jedoch auch zwei Vorteile: Einmal gibt die Existenz der Datei an, daß eine Autorin das Einfügen von Umkehrlinks für ein Dokument akzeptiert. Weiterhin wird das Dokument selbst nicht von außen verändert, sondern nur die Linkdatei. Umkehrlinks können dadurch gemeinsam behandelt, z.B. gelöscht werden.

Gewöhnliche WWW-Browser können Umkehrlinks (und die weiter unten besprochenen Anmerkungslinks) anzeigen, wenn die Links in einen Dokumententext eingemischt werden. Dokumente mit In-Links sollen im hier vorgestellten System außerdem auch ohne diese Links gelesen werden

10 der für die Übernahme von Xanadu durch Autodesk verantwortlich war (s. S. 56).

können. Daher wird die externe Speicherung von Links für ein Publikationssystem auf Basis des WWW übernommen. Die Speicherung erfolgt in einer Datei, die denselben Namen wie die entsprechende HTML-Datei trägt. Der Name der Linkdatei endet aber nicht auf ‚.html‘, sondern auf ‚.link‘. Diese Datei soll nur vom Server geändert werden.

Anmerkungslinks werden in ein Dokument eingemischt, wenn an sein URL ein spezieller Parameter, z.B. ‚?Annotations‘ angehängt wird. Analog können In-Links durch das Anhängen von, z.B., ‚?Inlinks‘ und beide durch das Anhängen beider Parameter an den Dokumenten-URL angefordert werden. Da der Server das Einmischen vornimmt, muß dafür die Position eines Links als Zahl oder Suchstring angegeben werden.

Wenn ein Dokument geändert wird, in dem sich Umkehrlinks befinden, sollen die Benutzerinnen, die diese Links gesetzt haben, über E-Mail benachrichtigt werden. Sie müssen dazu eine Mailadresse besitzen und sie beim Setzen eines Links angegeben haben. In-Links in einem Dokument können von der Institution, die das Dokument veröffentlicht hat, unter bestimmten Umständen auf Anfrage gelöscht werden. Dafür müssen jedoch spezielle Regeln vereinbart werden.

#### 4.4.2.6 Anmerkungslinks

Der bekannteste Ansatz für die Umsetzung von Anmerkungen im WWW ist der des NCSA, das in seinen weit verbreiteten WWW-Browser Mosaic die entsprechenden Funktionen eingebaut hat (LALIBERTE U. BRAVERMAN 95). Sowohl die zu annotierenden Dokumente als auch die Annotationen selbst können beim Anmerkungskonzept des NCSA aus allen Objekten des WWW bestehen, die durch einen URL adressierbar sind (ebd., Abschn. ‚Annotations‘).

Es gibt für Mosaic persönliche, gruppenweite und öffentliche Anmerkungen. Persönliche Anmerkungen werden vom Browser der entsprechenden Autorin verwaltet, gruppenweite in Anmerkungsservern im LAN der Gruppe und öffentliche in öffentlichen Servern (ebd.). Da es derzeit keinen öffentlichen Server gibt, dürften die Anmerkungsfunktionen von Mosaic selten benutzt werden.

Bei der ‚Platform for Internet Content Selection‘ (PICS) handelt es sich um einen neuen Standard für die Bewertung von Internet-Dokumenten, der von führenden Software- und Inhaltsanbietern wie Netscape, Microsoft und AOL (America Online) unterstützt wird. PICS besteht aus einer Reihe von Daten- und Übertragungsformaten, also weder aus Bewertungen selbst noch einem Vokabular dafür (RESNICK U. MILLER 96). Funktionen nach diesem Standard sollen nicht nur für die Bewertung von WWW-Dokumenten benutzt werden können, sondern auch, um Anmerkungen in Dokumente einzufügen. Die entsprechende Technik wird wahrscheinlich bald weit verbreitet sein, weil sie Eltern ermöglichen soll, ihre Kinder von ‚schädlichen‘ WWW-Dokumenten fernzuhalten.

PICS besteht aus einer Beschreibungssprache für Bewertungsdienste, einer Sprache für maschinell lesbare Bewertungen, Festlegungen, wie Bewertungen in Mails und HTML-Dokumente eingefügt werden können, Erweiterungen von HTTP, so daß Bewertungen zu einem Dokument geladen werden können und einem Protokoll, mit dem Bewertungen aus einer Datenbank abgefragt werden können (ebd.).

Alle Daten, die durch einen URL adressiert werden können, können durch PICS bewertet werden. Bewertungsdaten können durch eine digitale Unterschrift gesichert sein. Bewertungen werden entweder direkt in Dokumente eingebettet, vom Server übertragen, der auch das Dokument bereithält oder von einem anderen Server (z.B. auch eines unabhängigen Bewertungsdienstes) (ebd.). Benutzerinnen stellen in PICS-fähigen Browsern ein, welchen Bewertungsdienst sie benutzen möchten. Jede Bewertung kann gleichzeitig mehrere Aspekte eines Objektes bewerten (ebd., Abschn. ‚Flexible Blocking‘).

PICS-Unterstützung kann auf verschiedenen Ebenen des WWW-Zugriffs implementiert werden, z.B. auf der von Servern (Proxies), der Netzwerksoftware eines Computers oder in Browsern. Bewertungen müssen im Prinzip für jedes Dokument einzeln angefordert werden, falls sie jeweils getrennt von Dokumenten gehalten werden, können aber auch nach einmaliger Anforderung zwischengespeichert oder sogar in einem Browser vorinstalliert sein (ebd., Abschn. „What PICS Doesn't Specify“).

Wie sinnvoll die Bewertung von Internetinhalten und eine Zugriffssperre auf ihrer Basis ist, ist nicht Thema dieser Arbeit. PICS ist jedoch so spezifiziert, daß es auch für andere Dienste benutzt werden kann:

PICS is a new resource available to anyone who wishes to associate data with documents on the Internet, even documents that others control. (ebd., Abschn. „Other Uses for Labels“)

Die Technik kann also auch dazu dienen, elektronisch publizierte Dokumente zu bewerten. Diese Bewertungen sind zunächst auf ein bestimmtes Vokabular festgelegt (wie „bahnbrechend“ etc.), sie können jedoch auch einen URL als Link auf eine Begründung beinhalten. Bewertungen dienen also nicht nur der Erlaubnis oder dem Verbot eines Zugriffes, sondern auch der Klassifikation von betroffenen Dokumenten. Bewertungen sollen in PICS ausdrücklich nicht nur durch eine zentrale Instanz, sondern auch durch eine Gruppe von Benutzerinnen vergeben werden können (ebd.).

Der PICS-Standard legt keinen Mechanismus für das Erstellen von Bewertungen durch Einzelpersonen fest. Er kann also insgesamt nur als Grundlage betrachtet werden, auf dem ein Anmerkungsdienst aufbauen oder über den ein PICS-kompatibler Browser auf einen solchen Dienst zugreifen kann. Ähnlich wie für Umkehrlinks wird für das hier entworfene Publikationssystem festgelegt, daß Anmerkungen in zu Dokumenten externen Dateien und durch Server im Körper von veröffentlichten Dokumenten eingefügt werden.

#### 4.4.2.7 Gliederungsfunktion und Verzeichnisse

Eine Gliederungsfunktion kann leicht in einen erweiterbaren Editor eingebaut werden. Sie wirkt nämlich nur im aktuellen Dokument, in dem sie sich auf vorhandene HTML-Elemente, nämlich auf die für die Kennzeichnung von Überschriften bezieht. Die Gliederungsfunktion kann als spezielle Darstellung eines HTML-Dokuments implementiert werden, in der Überschriften numeriert und besonders hervorgehoben sind. In dieser Ansicht müssen Überschriften und die darunterliegenden Ebenen durch ‚Drag and Drop‘ ohne vorheriges Markieren mit der Maus verschoben werden können. Überschriften müssen in der Gliederungssicht also nach ihrer jeweiligen Ebenennummer interpretiert werden. Wenn auf eine Überschrift weitere mit niedrigerrangigen Nummern folgen, müssen diese als ersterer untergeordnet angesehen werden.

Sowohl in Ablagen als auch Servern soll es formale Verzeichnisse nach Autorinnennamen und Titeln geben, um ein Dokument finden zu können, wenn gerade kein Link auf es vorliegt. Diese Verzeichnisse stellen Hypertextdokumente dar, in denen sich Links befinden, die von den Namen bzw. Titeln auf die jeweiligen Dokumente verweisen. Da ein erweiterter Server jeweils beim Sichern und Löschen eines Dokuments die Einhaltung von korrekten Versionsbeziehungen überwacht, können die Verzeichnisse beim Sichern auf dem laufenden gehalten werden. In einer einfachen Ablage, die auf dem Dateisystem des jeweiligen Betriebssystems basiert, müssen die Verzeichnisse von Zeit zu Zeit durch Durchsuchen der ganzen Ablage erstellt werden.

## 4.5 Zusammenfassung und Schlußbemerkung

In dieser Arbeit wurden die praktischen Aspekte des wissenschaftlichen Schreibens dargelegt. Es wurden verteilte Hypertextsysteme beschrieben und die Nützlichkeit von Versionen und erweiterten

Linkarten für das wissenschaftliche Schreiben gezeigt. Weiterhin wurde eine pragmatische Spezifikation eines Hypertextsystems mit diesen Funktionen vorgestellt und die Bedienbarkeit des Systems erläutert. Schließlich wurde ein Implementationsansatz des Hypertextsystems auf Basis des WWW beschrieben, unter Berücksichtigung aktueller Erweiterungsvorhaben für das WWW.

Wie schon einmal betont wurde, müssen Hypertextsysteme sowohl technisch umsetzbar sein als auch von Benutzerinnen und Verantwortlichen angenommen werden. Das im letzten Abschnitt beschriebene Publikationssystem kann technisch relativ leicht umgesetzt werden, weil es auf dem WWW als praktisch erprobter Technik basiert. Die Implementation des Systems erfordert nur kleine Erweiterungen von WWW-Servern, für die Erstellung und die Abfrage von Links in Dokumenten und für die Versionsverwaltung. Dokumente im System sollen mit gewöhnlichen WWW-Browsern gelesen werden können. Das Schreiben von Dokumenten soll mit kleinen Editorerweiterungen oder speziellen Programmen ermöglicht werden, notfalls mit gewöhnlichen Texteditoren. Mit für es erweiterten WWW-Editoren soll besonders komfortabel auf das System zugegriffen werden können.

Ein verteiltes Hypertextsystem muß skalierbar sein. Dabei geht es um die Frage, ob das System nicht nur Daten kleinen Umfangs und eine kleine Anzahl von beteiligten Computern, sondern (praktisch) beliebig viele Daten und Netzstationen verwalten kann. Weil WWW-Server auf einer einfachen Technik aufbauen und unabhängig voneinander arbeiten, tritt für sie das Problem der Skalierbarkeit nicht bereits bei einer bestimmten Zahl von Servern im Netz auf, sondern erst bei einer hohen Rate von Browserzugriffen auf dieselben Server. Bestehende WWW-Angebote, die häufig nachgefragt werden, werden daher auf mehreren Servern unter einer Adresse angeboten, zwischen denen die Last dynamisch verteilt wird. Wissenschaftliche Publikationsserver müßten voraussichtlich nur wenige Anforderungen von Dokumenten pro Zeiteinheit beantworten, es wären dafür also keine speziellen Vorkehrungen nötig.

Wie das zugrundeliegende WWW ist das neue System skalierbar, weil Server ebenso wie im WWW autonom arbeiten. Es ist daher kein Austausch von Verwaltungsinformationen zwischen ihnen nötig, der generell nur schlecht skalierbar ist. Benutzerinnen des Systems sollen per E-Mail benachrichtigt werden, wenn sich ein Dokument geändert hat, auf das sie In-Links gelegt haben. Diese Mitteilungen müßten jedoch nicht in Echtzeit erfolgen, sondern könnten per E-Mail dann abgesetzt werden, wenn ein Server wenige Anfragen zu bearbeiten hat.

Die im letzten Abschnitt beschriebenen Erweiterungsansätze für das WWW wie das der WEB-DAV und PICS stellen nur Ausgangspunkte für die Entwicklung des WWW zu einem echten Publikationssystem in dem in dieser Arbeit formulierten Sinn dar. Sie sind in den wenigsten Fällen ausgereift und schon gar nicht in größerem Maßstab erprobt. Andererseits besteht die Hoffnung, daß durch die verschiedenen Forschungen zu diesen für das wissenschaftliche Schreiben nützlichen Funktionen die kritische Masse erreicht wird, so daß die Erweiterungsvorschläge z.B. vom W3C aufgegriffen werden. Wenn z.B. Funktionen der Versionsverwaltung in kommerzielle WWW-Editoren und -Server eingebaut würden, könnte ein wissenschaftliches Publikationssystem auf ihrer Basis leichter implementiert werden. Außerdem könnte auf es mit gängigen WWW-Browsern zugegriffen werden.

Benutzerinnen des Hypertext-Publikationssystems müßten jedoch generell keine völlig neuen Programme benutzen, weil es auf der Erweiterung von bestehenden Programmen basiert. Das System würde dadurch von Benutzerinnen leichter angenommen werden. Wissenschaftlerinnen als potentielle Hauptbenutzerinnen des Systems könnten mit ihm übergangslos Aufsätze aus Notizen in ihrer Ablage erstellen. Sie könnten Texte, die nicht vollständig linearisiert sind, deren ursprüngliche Bestandteile also noch teilweise sichtbar sind, als Hypertexte veröffentlichen.

Wissenschaftlerinnen müßten Endredaktion und Satz ihrer Dokumente im System selbst vornehmen, das müssen sie für den Abdruck eines Artikels in einer Zeitschrift jedoch auch. Autorinnen

könnten Materialien und Entwürfe anderen Benutzerinnen zur Verfügung stellen. Leserinnen könnten über veröffentlichte Artikel im System selbst diskutieren, mit direktem Verweis auf Stellen in den Artikeln. Hypertextverweise in einem Artikel zu anderen Dokumenten könnten für die Anforderung dieser Dokumente benutzt werden (wenn die Dokumente auch im System vorliegen).

Wissenschaftliche Institutionen, Fachgesellschaften und Verlage müßten für das System Veröffentlichungsserver bereitstellen. Wissenschaftliche Institutionen und Fachgesellschaften würden mit dem System die Möglichkeit erhalten, Veröffentlichungen ihrer Mitglieder selbst zu betreuen. Dadurch würden Verlagen Geschäfte entgehen. Allerdings könnten auch sie den Veröffentlichungsprozeß inklusive der Kommunikation für Begutachtungen im System durchführen, sowohl für Veröffentlichungen auf Papier als auch in digitaler Form. Gängige Monographien als Einnahmequellen für Verlage werden außerdem noch einige Zeit in Buchform verkauft werden.

Verlage könnten in dem Hypertextsystem neue Autorinnen anwerben. Verlage könnten außerdem Informationen anbieten, auf die sie nur zahlende Nutzerinnen zugreifen lassen. Dabei könnte es sich um bestimmte Fachinformationen handeln. Schließlich könnten Verlage für wissenschaftliche Institutionen und Fachgesellschaften die Betreuung von Veröffentlichungsservern übernehmen. Für eine solche Betreuung wären Systemverwalterinnen, Bibliothekarinnen, Lektorinnen etc. nötig. Für das System würden also immer noch Fachkräfte aus dem Verlags- und Bibliothekswesen benötigt, die sich mit der Produktion und Verwaltung von Büchern auskennen, sich allerdings auch auf die neue Technik einlassen müßten.

Ein weltweites wissenschaftliches Hypertext-Publikationssystem würde einen weiteren Teil der Visionen BUSHS und NELSONS verwirklichen, nachdem das WWW als einfaches verteiltes Hypertextsystem etabliert ist. Das wissenschaftliche Arbeiten würde sich durch ein Hypertext-Publikationssystem nachhaltig verändern, wenn auch noch nicht absehbar ist, wie genau das geschehen würde. Es ist jedoch wahrscheinlich, daß im Laufe der Zeit monolithische Veröffentlichungen durch echte Hypertexte abgelöst würden. Solche Hypertexte würden jedoch völlig neue Herausforderungen an den Wissenschaftsbetrieb darstellen, weil bei ihnen die Zuordnung eines Forschungsergebnisses zu einer einzelnen Wissenschaftlerin in den Hintergrund treten würde. Diese Möglichkeit der Zuordnung ist aber gerade bei wissenschaftlichen Veröffentlichungen nötig. Sie muß in einem wissenschaftlichen Hypertext-Publikationssystem technisch und organisatorisch sichergestellt werden.

## 5 Literatur

- ACHTERT, W. S., J. GIBALDI. 1985. *The MLA Style Manual*. New York: Modern Language Association.
- APACHE. O. J.a. „Apache Core features“. HTML-Dokument.  
URL: <http://apache.zit.th-darmstadt.de/docs/mod/core.html>.
- . O. J.b. „Apache module mod\_include“. HTML-Dokument.  
URL: [http://apache.zit.th-darmstadt.de/docs/mod/mod\\_include.html](http://apache.zit.th-darmstadt.de/docs/mod/mod_include.html).
- . O. J.c. „Apache module mod\_log\_referer“. HTML-Dokument.  
URL: [http://apache.zit.th-darmstadt.de/docs/mod/mod\\_log\\_referer.html](http://apache.zit.th-darmstadt.de/docs/mod/mod_log_referer.html).
- . O. J.d. „Module mod\_actions“. HTML-Dokument.  
URL: [http://apache.zit.th-darmstadt.de/docs/mod/mod\\_actions.html](http://apache.zit.th-darmstadt.de/docs/mod/mod_actions.html).
- AMERICAN INSTITUTE FOR PHYSICS. 1970. *A program for a National Information System for Physics and Astronomy 1971-75*. Information Division, American Institute for Physics. Zitiert in WALKER U. HURT 90.
- AMERICAN PSYCHOLOGICAL ASSOCIATION. 1963, 1965, 1969. Reports of the Project on Scientific Information Exchange in Psychology, Vol. I (1963), Vol. II (1965), Vol. III (1969). Washington, D.C.: American Psych. Assoc. Zitiert in WALKER U. HURT 90.
- BAGER, J., S. EHRMANN, K. OBERMAYR. 1996. „Seitenspinner: HTML-Werkzeuge für Anfänger und Profis“. *c't Magazin für Computertechnik* Nr. 7/96 (Juli 1996), S. 190-196.
- BATES, R. 1994. „K-Tree Container Data Structures“. *Dr. Dobb's Journal* 19, Nr. 9 (September 1994), S. 26-34.
- BELL, G. 1988. „Towards a history of (personal) workstations“, in A. Goldberg, Hrsg. *A History of Personal Workstations*. New York, NY: ACM Press, S. 4-47.
- BERNERS-LEE, T. 1996. „Information-Management: a Proposal“. Projektantrag. O. O. HTML-Dokument. URL: <http://www.w3.org/pub/WWW/History/1989/proposal.html>.
- BERNERS-LEE, T., R. CAILLIAU. 1990. „WorldWideWeb: Proposal for a Hypertext Project“. 12.11.90. HTML-Dokument. URL: <http://www.w3.org/pub/WWW/Proposal.html>.
- BERNERS-LEE, T., R. CAILLIAU, J.-F. GROFF, B. POLLERMANN. 1992. „World-Wide Web: the Information Universe“. *Electronic Networking: Research, Applications and Policy* 2, Nr. 1 (Spring 1992), S. 52-58.
- BERNERS-LEE, T., R. CAILLIAU, A. LUOTONEN, H. NIELSEN, A. SECRET. 1994. „The World-Wide Web“. *Comm. of the ACM* 37, Nr. 8 (August 1994), S. 76-82.
- BERNERS-LEE, T., R. FIELDING, H. FRYSTYK. 1996. „Hypertext Transfer Protocol — HTTP/1.0“. Request for Comment (RFC) 1945. Mai 1996. HTML-Dokument. URL: <http://www.ics.uci.edu/pub/ietf/http/rfc1945.html>.
- BROOKS, C., M. S. MAZER, S. MEEKS, J. MILLER. 1995. „Application-Specific Proxy Servers as HTTP Stream Transducers“. *Fourth International World Wide Web Conference Proceedings, Boston, MA, December 11-14, 1995*. World Wide Web Journal 1. Hg. The World Wide Web Consortium. Sebastopol, CA: O'Reilly & Associates, S. 539-548.
- BROOKS, T. A. 1985. „Private acts and public objects: an investigation of citer motivation“. *Journal of the American Society for Information Science* 36, S. 223-229. Erwähnt in MOON 88.
- BRORS, D. 1996. „Feinschliff: Erste Beta-Versionen von MS Office 97 und Lotus SmartSuite 97“. *c't Magazin für Computer-Technik* Nr. 11/96 (November 1996).
- BRÜGGEMANN-KLEIN, A. 1989. *Einführung in die Dokumentenverarbeitung*. Stuttgart: Teubner.
- BRÜGGEMANN-KLEIN, A., A. ANDRES, H. SCHWEPPE. 1995. „Informatik und die Informationsgesellschaft der Zukunft“. *Informatik-Spektrum* 18, Nr. 1, S. 25-30.

- BUSH, V. „As we may think“. 1991. *Atlantic Monthly* 176, Nr. 1 (1945), S. 641-649. Nachdruck in NYCE u. KAHN 91, S. 85-110.
- CARMODY, S., W. GROSS, T. H. NELSON, D. RICE, A. VAN DAM. 1969. „A hypertext editing system for the /360“. In: M. F. Aiman, J. Nievergelt: *Pertinent Concepts in Computer Graphics*. Urbana u. a.: Univ. of Illinois Press, S. 291-330.
- CONKLIN, J. 1987. „Hypertext: An Introduction and Survey“. *Computer* 20, Nr. 9 (September 1987), S. 17-41.
- VAN DAM, A. 1988. „Hypertext '87 Keynote Address“. In: *Comm. of the ACM* 31, Nr. 7 (Juli 88), S. 887-895.
- DENNING, P., B. ROUS. 1995. „The ACM Electronic Publishing Plan“. *Comm. of the ACM* 38, Nr. 4 (April 1995), S. 97-109.
- DREXLER, K. E. 1986. *Engines of Creation: the Coming Era of Nanotechnology*. New York: Anchor.
- DURAND, D. G., F. VITALI. 1996. „Functional Requirements and Frame work for Versioning on the WWW“. HTTP Working Group Internet-Draft. 6.11.96 (gültig bis April 97).  
URL: <http://www.cs.bu.edu/students/grads/dgd/draft-durand-www-versreq-00.html>.
- ECO, U. 1992. *Wie man eine wissenschaftliche Abschlußarbeit schreibt: Doktor-, Diplom- und Magisterarbeit in den Geistes- und Sozialwissenschaften*. 5., durchges. Aufl. d. dt. Ausgabe. Heidelberg: C. F. Müller.
- EHRMANN, S. 1996. „Web-Fabrik: PageMill und SiteMill für den Macintosh“. *c't Magazin für Computer-Technik* Nr. 1/96 (Januar 1996), S. 80-81.
- ELMASRI, R., S. B. NAVATHE. 1994. *Fundamentals of Database Systems*. 2. Aufl. Redwood City, CA u. a.: Benjamin/Cummings.
- ENDRES, A., Verantw. 1995. „Entwicklung und Erprobung offener volltext-basierter Informationsdienste für die Informatik“. Techn. Bericht. Gesellschaft für Informatik, Fachinformationszentrum Karlsruhe, Springer-Verlag. 23.6.95. PostScript-Dokument. URL: <http://medoc.informatik.tu-muenchen.de/publications/ovid-1.ps>.
- GETTYS, J. O. J. „Hypertext Transport [sic] Protocol HTTP/1.1“. HTML-Dokument. URL: <http://www.w3.org/pub/WWW/Talks/9608HTTP/>.
- GINSPARG, P. 1996. „Winners and Losers in the Global Research Village“. Inv. contrib. for Conf. held at UNESCO HQ, Paris, 19-23 Feb. 96. 21.2.96. HTML-Dokument. URL: <http://xxx.lanl.gov/blurb/pg96unesco.html>.
- GOLAND, Y., E. J. WHITEHEAD, U. C. IRVINE, A. FAIZI, D. JENSEN. 1996. „Extensions for Distributed Authoring and Versioning on the World Wide Web“. 26.11.96. ASCII-Dokument. URL: <http://www.ics.uci.edu/~ejw/authoring/draft-goland-http-webdav-00.txt>.
- GOLDFARB, C. F. 1990. *The SGML Handbook*. Oxford: Clarendon.
- GOSLING, J., H. MCGILTON. 1995. „The Java Language Environment: a White Paper“. Oktober 1995. PostScript-Dokument. URL: <ftp://ftp.javasoft.com/docs/whitepaper.A4.ps.tar.Z>.
- HALASZ, F. 1988. „Reflection on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems“. *Comm. of the ACM* 31, Nr. 7. (Juli 1988), S. 836-852.
- HALASZ, F., M. SCHWARTZ. 1994. „The Dexter Hypertext Reference Model“. *Comm. of the ACM* 37, Nr. 2 (February 1994), S. 30-39.
- HARNAD, S. 1995. „Implementing Peer Reviewing on the Net: Scientific Quality Control in Scholarly Electronic Journals.“ HTML-Dokument.  
URL: <http://www.princeton.edu/~harnad/Papers/harnad95.peer.review.html>.  
Abgedr. in: R. PEEK, G. NEWBY, Hg. *Electronic Publishing Confronts Academia: The Agenda for the Year 2000*. Cambridge, MA: MIT Press.

- HAYES, J. R., L. S. FLOWER. 1980. „Identifying the organization of writing processes“. In: L. W. Gregg, R. Steinberg, Hg. *Cognitive Processes of Writing*. Hillsdale, NJ: Lawrence Erlbaum, S. 2-30. Zitiert in RIEHM u. a. 92.
- HEEKS, R. 1986. *Personal Bibliographic Indexes and their Computerisation*. London: Taylor Graham. Erwähnt in MOON 88.
- HESLOP-HARRISON, J. 1978. „Learned Society Publishing in the United Kingdom“. In: *The Future of Publishing by Scientific and Technical Societies: Proc. of the Seminar Held in Luxembourg April 3-4, 1978*. Luxemburg: Kommission d. Europ. Gemeinschaften, S. 5-17. Zitiert in WALKER u. HURT 90.
- HYPERWAVE. 1996a. „Attributes“. 05.09.96. HTML-Dokument. HyperWave-Id 0x82496a1f0x00136ed2. (URL: [http://elib.zib.de/0x82496a1f\\_0x00136ed2](http://elib.zib.de/0x82496a1f_0x00136ed2)).
- . 1996b. „Editing HTF Texts“. 05.09.96. HTML-Dokument. HyperWave-Id 0x82496a1f0x00136f56. (URL: [http://elib.zib.de/0x82496a1f\\_0x00136f56](http://elib.zib.de/0x82496a1f_0x00136f56)).
- . 1996c. „The Go to Command“. 05.09.96. HTML-Dokument. HyperWave-Id 0x82496a1f0x00136ece. (URL: [http://elib.zib.de/0x82496a1f\\_0x00136ece](http://elib.zib.de/0x82496a1f_0x00136ece)).
- . 1996d. „HyperWave Server – Technical Specifications“. 25.4.96. HTML-Dokument. URL: <http://www.hyperwave.com/hyperwave/server/specifications.html>.
- JECHLE, T. 1992. *Kommunikatives Schreiben: Prozeß und Entwicklung aus der Sicht kognitiver Schreibforschung*. Tübingen: Narr.
- KAPPE, F. 1994. „Anchor“. 08.12.94. HTML-Dokument. HyperWave-Id 0x82496a1f0x00003e60 (URL: [http://elib.zib.de/0x82496a1f\\_0x00003e60](http://elib.zib.de/0x82496a1f_0x00003e60)).
- . 1991. „Aspects of a Modern Multi-Media Information System“. Dissertation. Inst. f. Information Processing and Computer Supported Media (ICCM). Graz: Technische Universität. PostScript-Dokument.  
URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/report308.ps.gz>.
- . 1993. „Hyper-G: A Distributed Hypermedia System“. Proc. INET '93, San Franc., CA, S. DCC-1–DCC-9 (Aug. 93). PostScript-Dokument.  
URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/inet93.ps>.
- . O. J. „A Scalable Architecture for Maintaining Referential Integrity in Distributed Information Systems“. Techn. Bericht. Inst. f. Information Processing and Computer Supported Media (ICCM). Graz: Technische Universität. PostScript-Dokument. URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/pflood.ps.gz>.
- KAPPE, F., M. GAISBAUER. 1994. „Hyper-G Text Format (HTF) Version 2.10“. Techn. Bericht. Inst. f. Information Processing and Computer Supported Media (ICCM). Graz: Technische Universität. 21. Mai 1994. PostScript-Datei.  
URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/HTF.ps>.
- KAPPE, F., G. PANI. 1994. „Hyper-G Client/Server Protocol (HG-CSP) Version 7.05“. Techn. Bericht. Inst. f. Information Processing and Computer Supported Media (ICCM). Graz: Technische Universität. 28. Dez. 1994. PostScript-Datei.  
URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/Protocol.ps.gz>.
- . O. J. „The Architecture of a Massively Distributed Hypermedia System“. Techn. Bericht. Inst. f. Information Processing and Computer Supported Media (ICCM). Graz: Technische Universität. PostScript-Datei. URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/report341.ps>.
- LALIBERTE, D., A. BRAVERMAN. 1995. „A Protocol for Scalable Group and Public Annotations“. The Third International World-Wide Web Conference: Technology, Tools and Applications. April 10-14 1995, Darmstadt, Germany. HTML-Dokument.  
URL: <http://www.igd.fhg.de/www/www95/papers/100/scalable-annotations.html>.
- LAUREL, B. 1991. *Computers as Theatre*. Reading, MA u. a.: Addison-Wesley.



- LEVY, D. M. 1994. „Fixed or Fluid? Document Stability and New Media“. *Europ. Conf. on Hypermedia Technology 1994 Proceedings*. New York, NY: ACM Press.
- LINN, J. 1993. „Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures“. Request for Comment (RFC) 1421. February 1993. ASCII-Dokument. URL: <ftp://ds.internic.net/rfc/rfc1421.txt>.
- LUOTONEN, A., J. FRANKS. 1995. „Byte Ranges With HTTP URLs“. Internet-Draft. 20.6.95. URL: [http://home.netscape.com/comprod/development\\_partners/plugin\\_api/byterang.html](http://home.netscape.com/comprod/development_partners/plugin_api/byterang.html).
- MCLUHAN, M. 1994. *Die magischen Kanäle: Understanding Media*. Dresden, Basel: Verlag d. Kunst.
- MAYRHOFER, V., K. ANDREWS. 1996. „Harmony User Guide“. Institute for Information Processing and Computer Supported New Media (IICM), Graz: Technische Universität. 27.6.1996. PostScript-Dokument. URL: <ftp://ftp.iicm.tu-graz.ac.at/pub/Hyper-G/papers/hug.ps>.
- MAURER, H., W. SCHINNERL, I. TOMEK. 1990. „Kommunikation in einem Hypermedia-System“. In: P. A. GLOOR, N. A. STREITZ, Hg. *Hypertext und Hypermedia: von theoretischen Konzepten zur praktischen Anwendung*. Heidelberg: Springer, S. 124-133.
- MEADOWS, A. J. 1974. *Communication in Science*. London: Buttersworth.
- MENDELSON, E. 1996. „WebWhacker 2.0: a Complete Off-line Environment“. 1.08.96. HTML-Dokument. URL: <http://www.pcmag.com/iu/browse/offline/fgwwha20.htm>.
- MEYROWITZ, N. 1991. „Hypertext – Does It Reduce Cholesterol, Too?“. Keynote address Hypertext '89. In: NYCE U. KAHN 91, S. 287-818.
- MOLITOR, S. 1985. „Personen- und aufgabenspezifische Schreibstrategien: fünf Fallstudien“. *Unterrichtswissenschaft* 4, S. 334-345.
- MOON, C. 1988. „Computerized Personal Information Systems for Research Scientists“. *International Journal of Information Management* 8, S. 265-273.
- NATIONAL CENTER FOR SUPERCOMPUTING APPLICATIONS (NCSA). O. J.a. „CGI: Common Gateway Interface“. HTML-Dokument. URL: <http://hoohoo.ncsa.uiuc.edu/cgi/intro.html>.
- . O. J.b. „CGI Environment Variables“. HTML-Dokument. URL: <http://hoohoo.ncsa.uiuc.edu/cgi/env.html>.
- NEBEL, E., L. MASINTER. 1995. „Form-based File Upload in HTML“. Request for Comment (RFC) 1867. November 1995. ASCII-Dokument. URL: <ftp://ds.internic.net/rfc/rfc1867.txt>.
- NEGROPONTE, N. 1970. *The Architecture Machine*. Cambridge, MA: MIT Press.
- NELSON, T. 1965. „A File Structure for the Complex, the Changing and the Indeterminate“. In: *Proc. of the ACM 20th National Conference*, S. 84-100.
- . 1967. „Getting It Out of Our System“. In: Hg. G. SCHECTER. *Information Retrieval, a Critical View*. Washington, DC: Thompson, S. 191-210.
- . 1980. „Replacing the printed word: a complete literary system“. In: Hg. S. H. LAVINGTON, *Information Processing 80*. Amsterdam u. a.: North-Holland Publishing Company. S. 1013-1023.
- . 1987. *Computer Lib/Dream Machines*. Überarb. Ausgabe. Redmond, WA: Microsoft Press.
- . 1991. „As we will think“. *Online 72 Conference Proceedings, Vol. 1*. S. 439-454. Abgedruckt in NYCE U. KAHN 91, S. 245-260.
- . 1993. *Literary Machines 93.1*. Sausalito, CA: Mindful Press (Eigenverlag des Autors).
- . 1995a. „Lemonade“. 9.6.95. HTML-Dokument. URL: <http://xanadu.net/wolfsbane>.
- . 1995b. „Summary 1: What Transcopyright Is“. Email in der „Xanadu“-Mailingliste. 26.11.95. Archiviert als HTML-Dokument. URL: <http://www.xanadu.com.au/hypermail/xanadu/0058.html>.

- . 1995c. „Summary 2: Protocol Discussion“. Email in der „Xanadu“-Mailingliste. 26.11.95  
Archiviert als HTML-Dokument.  
URL: <http://www.xanadu.com.au/hypermail/xanadu/0059.html>.
- . 1996. „Re: Hello everyone, tell me state of the project please.“ 26.11.96. Email in der  
„Xanadu“-Mailingliste. Message-Id: <v01540b1baebfbec4640@[133.27.108.156]>.
- NETCRAFT. 1996. „Netcraft Web Server Survey“. Dezember 96. HTML-Dokument. URL:  
<http://www.netcraft.co.uk/Survey/>.
- NETSCAPE COMMUNICATIONS CORP. 1996. „Netscape Handbook: Mail, News, and Book-  
marks“. HTML-Dokument.  
URL: <http://home.netscape.com/eng/mozilla/2.0/handbook/docs/mnb.html>.
- . O. J.a. „Introduction“. HTML-Dokument.  
URL: <http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/introd.html>.
- . O. J.b. „Navigator Objects“. HTML-Dokument.  
URL: <http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/navobj.html>.
- . O. J.c. „Netscape Handbook: Onscreen Fundamentals“. HTML-Dokument.  
<http://home.netscape.com/eng/mozilla/3.0/handbook/docs/graphics.html>.
- . O. J.d. „Overview“. HTML-Dokument.  
URL: <http://home.netscape.com/eng/mozilla/3.0/handbook/plugins/pr1.htm>.
- NEUWIRTH, C., D. KAUFER, R. CHIMERA, T. GILLESPIE. 1988. „The Notes Program: a Hyper-  
text Application for Writing from Source Texts“. *Proc. Hypertext '87*, TR 88-013. Chapel Hill:  
Univ. of North Carolina. S. 121-135.
- NIELSEN, J. 1990. *Hypertext and Hypermedia*. Boston u. a.: Academic Press.
- NYCE, J. M., P. KAHN, Hg. 1991. *From Memex to Hypertext: Vannevar Bush and the Mind's  
Machine*. Boston u. a.: Academic Press.
- OOKUBO, K., T. NELSON. O. J. „ZipEditor: A Collaborative Workspace for Sharable Material“.  
HTML-Dokument. URL: <http://hyper.sec.or.jp:9090/documents/coll-zip/Overview.html>.
- PAM, A. 1996a. „Methods for implementing transclusion of text into HTML pages“. Xanadu  
Australia. 4.08.96. HTML-Dokument.  
URL: <http://www.xanadu.com.au/xanadu/transclude.html>.
- . 1996b. „Sunrise on Xanadu“. Email in der „Xanadu“-Mailingliste. 6.10.96.  
Message-Id: <199610060113.LAA18820@fizzgig.glasswings.com.au>.
- . 1996c. „Xanadu FAQ“. Version 1.45. HTML-Dokument.  
URL: <http://www.xanadu.com.au/xanadu/faq.html>.
- . 1997. „Fine-Grained Transclusion in the Hypertext Markup Language“. Internet-Draft. 2  
March 1997. ASCII-Dokument.  
URL: <http://www.internic.net/internet-drafts/draft-pam-html-fine-trans-00.txt>.
- PITKOW, J. E., R. K. JONES. 1996. „Supporting the Web: A Distributed Hyperlink Database  
System“. Fifth International World Wide Web Conference May 6-10, 1996, Paris, France.  
HTML-Dokument. URL: [http://www5conf.inria.fr/fich\\_html/papers/P10/Overview.html](http://www5conf.inria.fr/fich_html/papers/P10/Overview.html).
- POENICKE, K. 1988. *Wie verfaßt man wissenschaftliche Arbeiten? Ein Leitfaden vom ersten Studi-  
ensemester bis zur Promotion*. 2., neu bearb. Aufl. Mannheim u. a.: Dudenverlag.
- QUINT, V. 1996. „Activity on Amaya“. 16.07.96. HTML-Dokument.  
URL: <http://www.w3.org/pub/WWW/Amaya/Activity.html>.
- QUINT, V. 1997. „Amaya architecture“. 10.02.1997. HTML-Dokument.  
URL: <http://www.w3.org/pub/WWW/Amaya/User/AmayaArchi.html>.
- RADA, R. 1991. *HYPertext: from Text to Expertext*. London u. a.: McGraw-Hill.
- RAGGETT, D. 1996a. „Client-Side Scripting and HTML: W3C Working Draft“. 09.06.96. HTML-  
Dokument. URL: <http://www.w3.org/pub/WWW/TR/WD-script.html>.

- . 1996b. „HTML 3.2 Reference Specification: W3C Working Draft“. 09.09.96. HTML-Dokument. URL: <http://www.w3.org/pub/WWW/TR/WD-html32.html>.
- RASKIN, J. 1988. „The Hype in Hypertext: a Critique“. *Proc. Hypertext '87*, TR 88-013. Chapel Hill: Univ. of North Carolina. S. 325-330.
- RAU, C. 1994. *Revisionen beim Schreiben: zur Bedeutung von Veränderungen in Textproduktionsprozessen*. Tübingen: Niemeyer.
- RESNICK, P., J. MILLER. 1996. „PICS: Internet Access Controls Without Censorship“. HTML-Dokument. URL: <http://www.w3.org/pub/WWW/PICS/iacwcv2.htm>.
- RIEHM, U., K. BÖHLE, I. GABEL-BECKER, B. WINGERT. 1992. *Elektronisches Publizieren: eine kritische Bestandsaufnahme*. Berlin u. a.: Springer.
- SAMUELSON, P., R. J. GLUSCHKO. 1991. „Intellectual Property Rights for Digital Library and Hypertext Publishing Systems: an Analysis of Xanadu“. *Proc. Hypertext '91*, New York: ACM, S. 39-50.
- SCHEURIG, ROLF. 1994. *UNIX System V. Die Befehle von A-Z*. Hg. CDI Computer Data. München, Wien: Oldenbourg.
- SCHNUPP, P. 1992. *Hypertext*. Handbuch der Informatik Bd. 10.1. Hg. A. Endres. München, Wien: Oldenbourg.
- TANNENBAUM, A. S. 1992. *Computer-Netzwerke*. 2. Aufl. Attenkirchen: Wolfram's Fachverlag.
- THEISEN, M. R. 1993. *Wissenschaftliches Arbeiten: Technik – Methodik – Form*. 7., überarb. u. aktualis. Aufl. München: Vahlen.
- Tichy, W.F. 1988. „Tools for Software Configuration Management“. In: Winkler, J. F.H., Hg. *Proc. of the International Workshop on Software Version and Configuration Control January 27-29, 1988 Grassau*. Berichte des German Chapter of the ACM 30. Stuttgart: Teubner, S. 1-20.
- TRIGG, R. H. 1991. „From Trailblazing to Guided Tours: The Legacy of Vannevar Bush's Vision of Hypertext Use.“ In: NYCE U. KAHN 91, S. 353-367.
- TRIGG, R. H., P. M. IRISH. 1987. „Hypertext Habitats: Experiences of Writers in NoteCards.“ *Proc. Hypertext '87*, TR 88-013. Chapel Hill: Uni v. of North Carolina. S. 89-108.
- VITALI, F., D. G. DURAND. 1995. „Using Versioning to Provide Collaboration on the WWW“. *Fourth International World Wide Web Conference Proceedings, Boston, MA, December 11-14, 1995*. World Wide Web Journal 1. Hg. The World Wide Web Consortium. Sebastopol, CA: O'Reilly & Associates, S. 37-50.
- WALKER, J. 1988a. „Xanadu“. In: WALKER 89, S. 416-419. Als HTML-Dokument: URL: [http://www.fourmilab.ch/autofile/www/chapter2\\_64.html](http://www.fourmilab.ch/autofile/www/chapter2_64.html)
- WALKER, J. 1988b. „Valedictory“. Rede auf der Aktionärsversammlung von Autodesk. 10.6.88. In: WALKER 89, S. 423-429. Als HTML-Dokument: URL: [http://www.fourmilab.ch/autofile/www/chapter2\\_66.html](http://www.fourmilab.ch/autofile/www/chapter2_66.html).
- WALKER, J., Hg. 1989. *The Autodesk File: Bits of History, Words of Experience*. 3. Auflage. Thousand Oaks, CA: New Riders Publishing.
- WALKER, J. 1993. „Creation/Evolution“. 5.11.93. HTML-Dokument. URL: [http://www.fourmilab.ch/autofile/www/chapter2\\_108.html](http://www.fourmilab.ch/autofile/www/chapter2_108.html).
- WALKER, J. 1995. „Hack Links“. July 12th, 1995. HTML-Dokument. URL: <http://www.fourmilab.ch/documents/hacklinks.html>.
- WALKER, R., C. D. HURT. 1990. *Scientific and Technical Literature: an Introduction to Forms of Communication*. Chicago u. London: American Library Association.
- V. WERDER, L. 1993. *Lehrbuch des wissenschaftlichen Schreibens. Ein Übungsbuch für die Praxis*. Berlin, Milow: Schibri.
- WORLD WIDE WEB CONSORTIUM (W3C). O. J. „A Little History of the World Wide Web“. HTML-Dokument. URL: <http://www.w3.org/pub/WWW/History.html>.

- . 1996. „UR\* and The Names and Addresses of WWW Objects“. 24.5.96. HTML-Dokument.  
URL: <http://www.w3.org/pub/WWW/Addressing/Addressing.html>.
- XANADU OPERATING COMPANY, INC. „FEBE 87.1 README File“. 29.12.87. ASCII-Dokument in Dateiarchiv. URL: <ftp://ftp.byte.com/1988/JAN88.ARC>.
- YANKELOVICH, N., B. J. HAAN, N. K. MEYROWITZ, S. M. DRUCKER. 1988. „Intermedia: the Concept and the Construction of a Seamless Information Environment“. *Computer* 21, Nr. 1 (Januar 88), S. 81-96.
- ZIVADINOVIC, D. „Katzengold: Netscape Navigator Gold Beta 2“. 1996. *c't Magazin für Computertechnik* Nr. 5/96 (Mai 1996), S. 112-113.