Hypertext-Systeme 1
(Lokale Systeme)

vorgelegt von:
Kevin Haack

Matrikelnummer: 7094226
Studiengang: Informatik (M.Sc.)

Thema betreut von:
Dr. Felix Winkelnkemper

Paderborn, 22. Februar 2019

Hypertext-Systeme 1 (Lokale Systeme)

Abstract

> „Consider a future device for individual use, which is a sort of mechanized private file and library. [...] A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.“ aus Vannevar Bushs Arbeit „As we may think“ , 1945[Bush, 1945]

Während der Begriff „Hypertext“ erst in 1960er Jahren von Theodor Holm Nelson [Nelson, 1965] geprägt wurde, er dachte Vannevar Bush in seiner Arbeit „As we may think“ die Memex. Das Konzept der Memex beschreibt eben genau diese Hypertext-Funktionalitäten, bevor die technischen Möglichkeiten überhaupt an so etwas denken ließen. Selbst jetzt in diesem Moment verwenden Sie als Leser ein Hypertext-Dokument. Je nachdem ob dieses Dokument ausgedruckt ist oder eine Anwendung Ihnen das Dokument anzeigt, haben Sie verschiedene Möglichkeiten den Text zu lesen. In ausgedruckter Form dieses Dokumentes sehen sie eine Referenz wie am Ende des Zitats und können ins Literaturverzeichnis in die entsprechende Zeile blättern. Von dort aus können Sie selbst entscheiden, ob Sie in der Quelle weiter lesen. In digitaler Form reicht meistens ein Klick und die Anwendung springt für Sie auf die entsprechende Seite. Doch diese Arbeit beschäftigt sich nicht mit den heutigen Möglichkeiten, sondern eher wie das Konzept Hypertext angefangen hat, was im Laufe der Zeit daraus geworden ist und welche Probleme dieses Konzept mit sich bringt.

Inhaltsverzeichnis

1 Was ist Hypertext? 1
1.1 Was ist überhaupt Hypertext? 1
1.2 Abgrenzung 4

2 Geschichtliches 5
2.1 NLS und AUGMENT 6
2.2 Problem: Dangling edges 8
2.3 HES 9
2.4 FRESS 10
2.5 Problem: Lost in Hyperspace 12
2.6 Document Examiner 12
2.7 HyperCard 14
2.8 Ist das noch Hypertext? 18

3 Literaturverzeichnis 20

1 Was ist Hypertext?

1.1 Was ist überhaupt Hypertext?

Was ist überhaupt Hypertext? Nach Jakob Nielsen in dem Buch „Multimedia and Hpertext: The Internet and Beyond“ aus dem Jahr 1995, ist:

> „The simplest way to define hypertext is to contrast it with traditional text like a book. All traditional text, whether in printed form or in computer files, is sequential, meaning that there is a single linear sequence defining the order in which the text is to be read. [...] Hypertext is nonsequential; there is no single order that determines the sequence in which the text is to be read.“ [Nielsen, 1995, S.1]

Jakob Nielsen beschreibt, genau wie Jeff Conklin in seinem Artikel „Hypertext: An Introduction and Survey“, Hypertext als Informationsquelle die dem Leser die Freiheit gibt die Informationen in beliebiger Reihenfolge abzurufen. Der Leser dürfe entscheiden, welche Links er nutze und in welcher Reihenfolge [Conklin, 1987, S.33] [Nielsen, 1995, S.1].

> „Hypertext allows and even encourages the writer to make such references, and allows the readers to make their own decisions about which links to follow and in what order. In this sense, hypertext eases the restrictions on the thinker and writer. It does not force a strict decision about whether any given idea is either within the flow of a paper's stream of thought or outside of it.“ [Conklin, 1987, S.33]

Also könnte ein Hypertext wie auf Abbildung 1.1 zu sehen, aus vielen einzelnen Informationen bestehen zwischen denen eine Verlinkung existiert. Durch diese Links kann der Leser entscheiden an welcher Stelle er weiterlesen möchte. Anstatt den Text in der festen Reihenfolge A, B, C, ... und dann G zu lesen, gibt es für den Leser mehrere Möglichkeiten die Texte aufzurufen. Nach Text A könne zum Beispiel sofort Text D folgen und so weiter [Nielsen, 1995, S.1].

Abbildung 1.1: Vereinfachte Ansicht eines Hypertextes [Nielsen, 1995, S.1]

Einen Hypertext kann man oftmals als Graphen aus Knoten und Kanten dargestellen. Diese Darstellungsform kann vor dem Leser verborgen sein, muss es aber nicht [Conklin, 1987]. Eine Information, ein Absatz, ein Text oder ein Dokument ist ein Knoten, die Größe spielt hierbei keine Rolle. Ein Knoten kann nun Links auf andere Knoten haben und diese sind dann die Kanten in dem Graphen [Conklin, 1987, S.19] [Nielsen, 1995, S.2]. Auch John B. Smith und Stephen F. Weiss schrieben in dem Artikel „Hypertext“ vergleichbar über Hypertext als Netzwerk aus Informationen mit Links [Smith and Weiss, 1988]. Diese bezeichnen unter anderem auch Verlinkungen zwischen Medien wie Audio-, Grafik- oder Videodateien als Hypertext. In diesem Zusammenhang verwendet Theodor „Ted“ Holm Nelson auch den Begriff „Hypermedia“ [Nelson, 1965].

> „[...] Hypertext is an approach to information management in which data is stored in a network of nodes connected by links. Nodes can contain text, graphics, audio, video, as well as source code or other forms of data.“ [Smith and Weiss, 1988]

Wie auf Abbildung 1.2 zu sehen, müsse eine Link also nicht unbedingt von Wort zu Wort führen, sondern könne auch den Token $xxxx$ im Dokument A mit dem ganzen Dokument B verbinden [Conklin, 1987].

Abbildung 1.2: Ein Beispiel eines Links von in einem Text zu einem anderen Text [Conklin, 1987, S.34]

Das gleiche könne auch auf einen traditionellen Text auf Papier zutreffen, Fußnoten referenzieren auf andere Texte und geben dem Leser die Möglichkeit an anderer Stelle weiterzulesen [Nielsen, 1995]:

> „The same is true of footnotes in traditional printed texts, since readers have to determine upon reaching the footnote marker whether to continue reading the primary stream of text or to branch off to pursue the footnote. “ [Nielsen, 1995]

Auch Ted Nelson schrieb in „Literary Machines“ 1992 über die Vergleichbarkeit zu Texten auf Papier:

> „By hypertext I simply mean non-sequential writing. A magazine layout, with sequential text and insert illustrations and boxes, is thus hypertext.“ [Nelson, 1992]

1.2 Abgrenzung

Doch für eine Antwort auf die Frage was ein Hypertext ist, gibt es sicherlich noch viele andere Antworten und Definitionen. Doch im Jahr 1945 veröffentlichte Vannevar Bush den Artikel „As we may think“ und formulierte darin die Grundidee zu Hypertext. Ich beschränke mich in dieser Arbeit auf lokale Hypertext-Systeme insbesondere der achtziger Jahre. Ich nehme in dieser Arbeit Bezug auf ausgewählte Hypertext-Systeme, deren Benutzung, Funktionen und technische Umsetzung. Ein besonderes Augenmerk legt diese Arbeit hierbei auf die Umsetzung der Links. Vernetzte und verteilte Hypertext-Systeme klammere ich in dieser Arbeit explizit aus.

2 Geschichtliches

In der Geschichte von Hypertext bis Heute sind viele verschiedene Konzepte und Systeme entstanden. In diesem Kapitel werden für einen Überblick einige davon vorgestellt. Das Konzept hinter Hypertext kann auf einen Artikel aus dem Jahr 1945 zurückgeführt werden.

> „Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, memex will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility.“ [Bush, 1945, Section 6].

Vannevar Bush schrieb in dem Artikel „As we may think“ von dem Konzept „Memex“ , einer Maschine für eine individuelle, private und mechanisierte Bibliothek. Die Abbildung 2.1 zeigt eine Illustration aus dem Life Magazine 1945. Wenn der Nutzer ein bestimmtes Buch aufrufen möchte, könne er einen Code in das Keyboard eintippen und die Titelseite erscheint als Projektion [Life, 1945, S.121] [Bush, 1945, Section 6].

Abbildung 2.1: „Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. [...] At left is a mechanism which automatically photographs longhead notes, pictures and letters, then files them in the desk for future reference“ [Life, 1945, S.123]

In in diesem Konzept solle der Nutzer einen sogenannten „Main Trail“ aus Dokumenten erstellen. Dieser Trail sei eine Kombination aus Dokumenten, die aus der Perspektive des Nutzers von Interesse ist. Mit sogenannten „Side Tails“ sollte der Nutzer ein Main Trail mit anderen Dokumenten „assoziieren“ [Bush, 1945, Section 7]. Diese Side Trails könnten als erstes Konzept von Hyperlinks verstanden werden. Zwei Jahrzehnte nach der Veröffentlichung von „As we may think“ , in der 1960er Jahren, wurde erst der Begriff Hypertext von Theodor „Ted“ Holm Nelson geprägt.

> „Let me introduce the word hypertext to mean a body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented on paper. It may contain summaries, or maps of its contents and theier interrelations; it may contain annotations, additions and footnotes from scholars who have examined it. [...]“ [Nelson, 1965]

Nach Ted Nelson „habe ein System auf Papier gravierende Einschränkungen beim Organisieren oder Präsentieren von Ideen. Ein Buch könne nie perfekt zu einem Leser passen. Der eine Leser sei gelangweilt, während ein anderer von den gleichen Seiten verwirrt werde. [...] Ein solches System könne das Potenzial haben, das Gefühl der Freiheit, die Motivation und das intellektuelle Verständnis des Lesenden zu vergrößern“ [Nelson, 1965].

2.1 NLS und AUGMENT

Im gleichen Jahrzehnt erschien Doug Engelbarts „Augmenting Human Intellect: A Conceptual Framework“ . Er schrieb unter anderen von dem „digitalen Computer als Werkzeug für die persönliche Benutzung. Hier verspricht man nicht nur große Flexibilität beim Erstellen und Neuanordnen von Texten und Diagrammen direkt vor den Auge des Benutzers, sondern auch viele andere Verarbeitungsmöglichkeiten [...]“ [Engelbart, 1962, S. 17].

> „By augmenting human intellect we mean increasing the capability of a man to approach a complex problem situation [...].“ [Engelbart, 1962, S. 1]

Während seiner Arbeit am Augmentation Research Center am Stanford Research Institute (SRI-ARC) wurde unter anderem das „NLS“ (the oN-Line System) entwickelt. Wobei „on-line“ in den 60er Jahren nicht die gleiche Bedeutung gehabt haben dürfte als Heute. Vorgestellt wurde das NLS auf der Fall Joint Computer Conference in San Francisco 1968. Diese Präsentation wird oft „mother of all demos“ genannt. Neben dem NLS wurden auch die interaktive Textverarbeitung, die Computermaus und die Organisation von Windows auf dem Bildschirm vorgestellt (Siehe Abbildung 2.2). Das NLS wurde präsentiert als „ein mächtiges Tool für die Arbeit eines Individuums, wenn er studiert, plant, designt, debuggt oder dokumentiert.“ [Engelbart, 1968]. Es

Abbildung 2.2: Doug Engelbart präsentiert auf der Fall Joint Computer Conference in San Francisco die Forschung vom Stanford Research Institute [Engelbart, 1968].

gab Möglichkeiten für kollaboratives Arbeiten, gemeinsame Dokumente und man konnte Nachrichten an Dokumente heften [Engelbart, 1968].
Gezeigt wurden aber auch die Hypertext Funktionalitäten, Dokumente seien verknüpft mit Links und der Nutzer könne entscheiden welchen „Branch“ er betrete [Engelbart, 1968]. Verlinkungen konnten durch verschiedene Adressiermöglichkeiten realisiert werden. Jedes Dokument in NLS ist in „Statements“ unterteilt und diese Statements konnten auf verschiedene Weisen verlinkt werden. Um ein Statement eindeutig zu identifizieren, wurden dem Nutzer vier Möglichkeiten zur Verfügung gestellt [Engelbart, 1984]:

- Structural Statement Number
  Die „Structural Statement Number“ bezeichnet die strukturelle Position des Statements. Das System vergibt diese Nummber anhand der Position innerhalb eines Dokumentes. Diese Nummern konnten zum Beispiel dargestellt werden als 1, 1a, 1a1 und 1b. Durch das Verändern der Dokumentenstruktur wurden vom System allerdings neue Nummern vergeben, so konnten Verweise auf falsche oder nicht mehr existierende Statements zeigen.

- Statement Identifier (oder SID)
  Der „Statement Identifiers“, als globale ID war unabhängig von der Position des Statements, diese Nummer wurde sequenziell vom System vergeben und wurde nie verändert. Diese Nummern wurden immer mit der „0“ als Präfix angezeigt.

- Worker-Assigned Statement Name (oder label)
  Die „Labels“ konnten vom Nutzer definiert werden. Der Nutzer konnte Trennzeichen angeben, mit denen das System dem Statement einen

Namen gibt. Wenn zum Beispiel die Trennzeichen „(“ und „)“ gewählt wurden, erkannte das System das erste eingeklammerte Wort als Label.

- Direct Screen Selection
  Die letzte Möglichkeit war die „Direct Screen Selection“, der Nutzer konnte mit der Maus den Cursor auf ein Statement bewegen und mit dem Drücken von „Select“ auf der Maus das entsprechende Statement auswählen. Diese letzte Möglichkeit diente zur Textmanipulation.

Das Userinterface bot Kommandos zum Manipulieren der Oberfläche. Die interaktive Textbearbeitung unterstützte Funktionen wie Insert, delete, Move and Copy. Es konnten Outlines und eine Vorschau der ersten Zeilen von jedem Statement angezeigt werden. Die Verlinkungen und Bedienmöglichkeiten ermöglichten dem Nutzer ein „herein und heraus zoomen “ [Engelbart, 1968]. Die kommerziellen Rechte an dem System wurden 1978 an Tymshare abgegeben und NLS wurde zu AUGMENT umbenannt [Engelbart, 1984]. Im Vergleich zum Konzept Memex hatte das Team um das NLS ähnliche Ziele, denn in beiden Projekten sollte die die Arbeit des Individuums optimiert werden [Life, 1945] [Engelbart, 1968]. Die Verknüpfungen der Dokumente in der Memex sind vergleichbar mit den verlinkten Statements im NLS, beide Verlinkungen wurden über IDs realisiert [Engelbart, 1984] [Bush, 1945]. Nur die Navigation durch die Links unterschied sich bei beiden, statt berührungssensitive Anzeige und Spracheingabe [Bush, 1945], nutzte das Team rund um Doug Engelbart Maus und Tastatur [Engelbart, 1968].

2.2 Problem: Dangling edges

Allerdings wäre Vannevar Bush bei einer Realisierung der Memex wahrscheinlich auf das gleiche Problem gestoßen wie Doug Engelbart. Ein Hypertext ist vergleichbar mit einem Graphen aus Knoten und Kanten. Ein Text ist ein Knoten und die Verlinkungen zwischen den Texten sind die Kanten zwischen den Knoten. Durch entfernen von Knoten können sogenannte „dangling edges“ entstehen, eine Kante, die ein Start hat aber kein Ziel mehr. Die Abbildung 2.3 zeigt zum Beispiel durch das Entfernen des Textes D zwei entstandenen Dangling Edges von A und E. Bei NLS konnte durch das Bearbeiten der Dokumente oder Statements Dangling Edges entstehen. Dies könnte durch die verschiedenen Adressiermöglichkeiten zumindest eingedämmt werden [Engelbart, 1984].

Abbildung 2.3: Modifizierte Darstellung von Nielsen, zur Darstellung von zweier Dangling Edges [Nielsen, 1995, S.1].

2.3 HES

Ebenfalls in dem 1960er Jahren wurde an der Brown Universität, im US-amerikanischen Bundesstaat Rhode Island, ein Hypertext-System Namens „HES“ (Hypertext Editing System) entwickelt. 1967 begannen Andries van Dam und Ted Nelson die Entwicklung an HES. Eigentlich sei der Zweck von HES gewesen, das Hypertext Konzept zu erforschen, später sei es von IBM aber auch an die NASA verkauft worden und zur Dokumentation der Apollo Missionen verwendet worden [van Dam, 1988].

> „One of the most important things (Nelson) taught me was that this is a new medium and you really can't be constrained to thinking about it in the old ways. Don‘t copy old, bad habits; think about new organizations, new ways of doing things, and take advantage of this new medium.“ [van Dam, 1988]

Texte konnten in HES in separate Sektionen unterteilt werden. Grundlegend gab es zwei verschiedene Verlinkungen in HES. Es wurde unterschieden zwischen „Links“ und „Brachnes“ . Branches wurden organisiert im Branch Menü oder waren im späteren Ausdruck als Fußnoten zu sehen. Links konnten optional vom Nutzer besucht werden. Beide waren im Text markiert mit einem Sternchen (*) und zeigten einen sogenannten Explainer als Vorschau an. Der Explainer eines Links tauchte in der Annotaion Area auf, der Branch Explainer wurde in-line beim Branch Sternchen angezeigt. Der Nutzer konnte Sektionen mit dem Lightpen auswählen und einem einzigartigem Label versehen. Dieses Label wurde dann in eine Label-Tabelle eingetragen und konnte mit einem Befehl direkt abgerufen werden. Außerdem konnte das Label als Link in einem Text verwendet werden [van Dam, 1969]. Die Texte selbst waren sogenannte Instanzen und wurden unidirektional verlinkt. „Die Bearbeitung selbst sei mehr eine Pointer- als Textmanipulation gewesen“ [van Dam, 1988, S. 890].

Abbildung 2.4: Foto einer von HES in der Brown Universität [Lloyd, 1969].

> „Instances are references, so that if you changed, for example, a piece of legal boilerplate that was referenced in multiple places, the change would show up in all the places that referenced it. “ [van Dam, 1988]

Dem Nutzer war die Möglichkeit gegeben, nach dem Drücken eines „Function Keys“ mit dem Lightpen einen Link zu berühren und so dem Link zu folgen [van Dam, 1969, S.23]. Wie auch das NLS unterstützte die Textbearbeitung Funktionen wie Insert, Delete, Move and Copy [van Dam, 1969, S.10-14] [van Dam, 1988, S. 889]. Beim Entfernen von Texten entstanden keine Dangling Edges, wie in Abschnitt 2.2 beschrieben, da jeder Link und Branch (der auf die gelöschte Sektion zeigte) ebenfalls gelöscht wurde [van Dam, 1969, S.12].

2.4 FRESS

> „The other important thing he really led me to focus on is that there is a great application for hypertext and hypermedia, not just for research, not just for maintaining your own personal database, but in teaching. And that :is one of my major themes here: how we have used hypertext and hypermedia in teaching at Brown University.“ [van Dam, 1988, S.888]

1968 startete van Dam zusammen mit seinen Studenten an der Brown Universität die Entwicklung von „a File Retrieval and Editing System“ (FRESS). Inspiriert von Doug Engelbart sollte FRESS die besten Ideen von NLS und HES

vereinen [van Dam, 1988, S.887 und 890]. Im Gegensatz zu vorherigen Systemen solle FRESS auch auf verbreiteten kommerziellen Systemen laufen. Van Dam zufolge sei FRESS das erste System gewesen, dass eine UNDO Funktion unterstützte. Jede Bearbeitung wurde in einener Schattenversion gespeichert und erlaubte so ein Autosave und ein UNDO [van Dam, 1988, S.891].

Abbildung 2.5: Engelbart an einer NLS/Augment Workstation. Das Key Set unter seiner linken Hand und die Maus unter seiner rechten Hand. [Conklin, 1987].

Im Gegensatz zum Vorgänger unterstützte FRESS bidirektionale Links, so genannte „Jumps“ . Aber weiterhin gab es auch on-way Links, die sogenannten „Tags“ . Paul Delany und George P. Landow schrieben hierzu:

> „FRESS had two types of links - indicated a connection to a single element such as an annotation, definition, or footnote. When the reader pointed to a tag with a light pen, the associated text appeared in another window on the screen for reference while the reader remained in the main document. Unlinke a tag, a jump - a bidirectional link - indicated a path to another document. By following a jump, the reader was transferred from one document to another, and up to seven windows could be used to display documents simultaneously. “ [Paul Delany, 1995, S.68]

FRESS unterstütze so den Nutzer beim Navigieren durch die Links, außerdem konnte so der Autor, wie schon bei der Memex angedacht, eine Art Main Track

oder Side Tracks erstellen. Im Dokument wurden die „cross-reference markers“ angezeigt, diese zeigten dem Nutzer alle Verlinkungen an und gaben dem Leser einen „Backtrack“ durch seine besuchten Links. Diese waren in FRESS eigene Objekte und nicht mehr einfach Teile eines Textes. Es sei möglich gewesen Links zu typisieren oder zu labeln, indem man sie mit „key words“ versehe [van Dam, 1988, S.891]. Das System wurde auf Grund des hohen Funktionsumfangs nicht fertig gestellt, van Dam schrieb:

> „It was overkill, and we haven't done it since, but it certainly was interesting.“ [van Dam, 1988, S.891]

2.5 Problem: Lost in Hyperspace

Je komplexer und größer ein Hypertext Graph wird, desto einfacher kann man sich auf der Vielzahl von Branches verlieren. So auch van Dam: „Wir hatten bereits Bemerkt, dass je größer der Hypertext ist, desto größer das Navigationsproblem ist“ [van Dam, 1988]. Nach van Dam sei das „come from“ genauso wichtig wie das „go to“ [van Dam, 1988]. Er schrieb über HES:

> „[...] The real diagram was about the size of a large blueprint [...]. We had already come to the point where Ted, who designed it, was able to go through the hypertext pretty well, but some of the rest of us had difficulty following it - it was not exactly obvious where you were. This, of course, is the classical lost-in-hyperspace problem [...]“ [van Dam, 1988]

FRESS nutzte hingegen beim Navigieren Windows und bidirektionale Links damit der Nutzer jederzeit einen Weg zurück fand. Bei einer Umsetzung der Memex ist zu vermuten, dass dieses Problem durch die zwei Anzeigen zumindest entschärft worden wäre, aber auch hier wäre man beim Navigieren durch die Dokumente irgendwann „lost in hyperspace“gewesen. Erst die bidirektionalen Links und Windows lösen dieses Problem und geben dem Nutzer die Möglichkeit seinen Weg zurückzugehen.

2.6 Document Examiner

Die ersten Hypertext-Systeme wie NLS und FRESS haben gezeigt, dass sie nicht nur ein Konzept sind, sondern auch als echte Systeme implementiert werden können. In den achtziger Jahren kamen dann einige mehr auf. Eines davon war im Jahr 1985 „Document Examiner“ von Symbolics. Document Examiner diente zur Anzeige der Handbücher und der Dokumentation der Symbolics Systeme, war also nicht für akademische oder Forschungszwecke, sondern kommerzielle Zwecke gedacht [Walker, 1987] [Nielsen, 1995, S. 50]. Anders als bei vorherigen System sei es Ziel bei der Entwicklung von Document Examiner gewesen, die Oberfläche handhabbar wie ein Buch zu gestalten und dabei Vorteile einer nicht-Papier-Lösung hinzuzufügen [Walker, 1987]. Durch die

Buchmetapher solle die Anwendung benutzerfreundlicher als andere Systeme werden [Walker, 1987].

> „The most fundamental decision in the interface was to make the material that a person was reading look essentially as it would in a paper book. The reason for doing this was ease of use. We saw no reason to have the underlying information structure be reflected in the user interface model unless that structure was a good model for interacting with information. My experience in trying to help users with a tree-structured information interface [...] led me to believe that a book-like interface would be more palatable for many people. “ [Walker, 1987]

Abbildung 2.6: Userinterface von Document Examiner. Zeigt die vier Abschnitte: Content Area, Bookmark Pane, Candidates Pane und Command Region [Walker, 1987].

Document Examiner wurde als „window-based“ bezeichnet, wie auf Abbildung 2.6 zu sehen, unterteilen die Windows die Anwendung in vier Abschnitte. Die „Conent Area“, zum Anzeigen des Inhalts. Die „Bookmark Pane“, enthält eine chronologische Aufzählung der Records, die der Benutzer gelesen hat. Die „Command Pane“, die dem Nutzer es ermöglicht Befehle abzusetzen [Walker, 1987]. Die „Candidates Pane“, enthält die vom Nutzer zusammengestellten Records, die noch zu lesen sind. Die Liste der Candidates war wie ein Suchergebnis, der Nutzer konnte einen Befehl absetzen, mit dem er nach bestimmten Begriffen im Titel oder in den Keywords suchen konnte. Abbildung 2.7 zeigt eine Suchanfrage und das Ergebnis eines solchen Befehls.

Command line:

Show Candidates finding help

Candidates offered:

Finding Out About Zmacs Commands with HELP
Finding Out What a Prefix Command Does
Method for Searching for Appropriate Zmacs Commands
More HELP Commands for Finding Out About Zmacs Commands

Keywords for "Method for Searching for Appropriate Zmacs Commands":

m-X Apropos
Searching for appropriate commands
Finding the right command
Help A Zmacs command

Abbildung 2.7: Zeigt die vorgeschlagenen Records, die mit dem Befehl *showcandidates* gesucht wurden [Walker, 1987].

Die Inhalte in Document Examiner sind in sogenannte „Records“ unterteilt, jeder Record hat eine eindeutige ID, einen Namen und ein Typ. Die Records sind unidirektional mit anderen Records verlinkt. Diese Verlinkung kann entweder ganze Records verlinken oder verlinkt eine Stelle im Text mit einem anderen Record. Durch die verwendete Buchmetapher ist die ganze Dokumentation in Kapitel, Abschnitte und Unterabschnitte unterteilt [Walker, 1987]. Hintereinander gelesen ergeben die Records, analog zur Memex, einen Main Trail und weitere Verlinkungen sind die Side Trails. Das Problem der Dangling Edges (aus Abschnitt 2.2) bestand auch sicherlich in Document Examiner, aber da die Erstellung und Wartung des Inhalts nur durch Symbolics geschah [Walker, 1987] konnte sichergestellt werden, dass jeder Link stets verbunden war. Auch das Problem Lost in Hyperspace (aus Abschnitt 2.5) konnte in dem System verringert werden. Die Candidates und Bookmark Pane gab dem Nutzer einen Überblick, was bereits gelesen war und was noch folgen kann. Auch die Verwendung der Buch Metapher solle nach Walker die Übersicht und Benutzerfreundlichkeit im Vergleich zu anderen Anwendung verbessert haben [Walker, 1987].

2.7 HyperCard

Während der achtziger Jahre arbeitete Bill Atkinson für Apple Computers und entwickelte mit an dem 1987 veröffentlichtem „HyperCard“. HyperCard ist ein System zum Entwickeln von Hypertextdokumenten und zugleich zur Entwicklung von grafischen Oberfläche. Das Hauptziel des System war es „Lernen einfach zu gestalten“ [Jakob Nielsen, 1991]:

> „One of the main design goals of HyperCard was to make it simple and easy to learn.“[Jakob Nielsen, 1991]

Grundlegend besteht HyperCard aus zwei Sub-Systeme, zum Einen die Grafische Oberfläche und zum Anderen die HyperTalk Programmiersprache. In

dem System gab es neun „Hauptelemente“, unter anderem fünf davon waren [Goodman, 1988]:

- Stacks
- Backgrounds
- Cards
- Fields
- Buttons

HyperCard macht sich in vielen Punkten die Karteikarten-Metapher zunutze. Alle Informationen sind gleich strukturiert und passen genau auf eine „Card“. Diese Karte sollte genau auf einen 9”Macintosh Bildschirm und passend zur Metapher in einen „Stack“ einsortiert sein. Der Nutzer kann in dieser Standardanordnung von Cards blättern auch ohne Hyperlinks zu verwenden. Die eigentliche Hypertextfunktionalität kommt erst in Kombination mit der Programmiersprache „HyperTalk“ [Goodman, 1988] [Jakob Nielsen, 1991]. Eine Card war - vereinfacht gesagt - aus drei Schichten aufgebaut (Siehe Abbildung 2.8). Der „Background Layer“ der den gemeinsamen Hintergrund für alle Cards auf dem Stack zeigt. Der „Foreground Layer“ der den Text, sowie Illustration für diese Card beinhaltet. Der „Button Layer“ der aktive Regionen enthält, die HyperTalk Skripte triggern können [Nielsen, 1995].

Abbildung 2.8: Vereinfachte Ansicht der HyperCard Layer [Nielsen, 1995].

Um einen Link zu erstellen verwendete der Nutzer die Programmiersprache HyperTalk. Ein Beispiel hierfür ist auf der Abbildung 2.9 zu sehen. In diesem Beispiel wurde der Hintergrund „Six Monthly“ gewählt, beim Klicken auf eine Woche bestimmt das HyperTalk Skript die gewählte Woche und leitet den Nutzer auf die entsprechende Wochenkarte weiter. Es könnten aber auch weitere Berechnungen und Bedingungen die Verlinkung beeinflussen [Goodman, 1988], so konnten beliebig komplexe Hypertexte entstehen. Ein solcher Link war allerdings immer nur unidirektional, ein Skript auf der Zielkarte konnte aber einen Link zurück anbieten.

Abbildung 2.9: Ein Beispiel für einen Monat, dargestellt als Card in HyperCard. Beim Klick auf eine Woche wird die entsprechende Card im Termin Stack geöffnet [Goodman, 1988].

HyperCard identifiziert seine Cards mit der Position innerhalb eines Stacks. Das Problem ist hierbei ist ähnlich wie in Abschnit 2.2 beschrieben. Wenn der Nutzer zum Beispiel die Karte auf Position Zwei in einem Stack löscht, rücken alle späteren Karten nach [Goodman, 1988]. Allerdings bot HyperCard noch eine andere Möglichkeit auf Cards zu referenzieren. Jede Card erhielt bei der Erstellung eine eindeutige Card ID, keine andere Card in dem Stack konnte diese ID haben. Selbst wenn eine Card gelöscht wurde, wurde die ID nicht erneut vergeben. So konnte zumindest nicht auf eine falsche Card verlinkt werden [Goodman, 1988]. Durch das Löschen einer Karte konnte das Problem einer Dangling Edge allerdings trotzdem auftreten (siehe Abschnitt 2.2).

2.8 Ist das noch Hypertext?

Die Literatur erwähnt HyperCard oftmals als Hypertext-System [Nielsen, 1995] [van Dam, 1988], [Smith and Weiss, 1988], an dieser Stelle könnte man sich aber folgende Frage stellen: Ist ein System zum Entwickeln grafischer Oberflächen überhaupt noch ein Hypertext-System oder eher schon Programmiersprache? Zunächst einmal, was unterscheidet HyperCard von anderen Systemen und Konzepten? Jakob Nielsen schrieb in „The learnability of HyperCard as an object-oriented programming system“ über das System als Programmiersprache, Nutzer mit Programmiererfahrung hätten es leicht die Benutzung von HyperTalk zu verstehen [Jakob Nielsen, 1991]. Aber auch in anderen Systemen, wie das HES, müsse der Nutzer zunächst eine Syntax beherrschen um Text zu verfassen und Links zu erstellen [van Dam, 1969].

> „I remember this particular demo we did at Time/Life when our audience said, “That's great, but it will take us at least 10 years before people will be willing to sit down behind tubes and do anything on-line.”“ [van Dam, 1988]

Auch das Konzept des Main Trails aus der Memex, kann man im wesentlichen mit den Stacks von HyperCard vergleichen. Auch unsere Definitionen aus Kapitel 1 treffen auf HyperCard zu, zum Beispiel schrieben John Smith und Stephen Weiss in dem Artikel „Hypertext“:

> „[...] Hypertext is an approach to information management in which data is stored in a network of nodes connected by links. Nodes can contain text, graphics, audio, video, as well as source code or other forms of data.“ [Smith and Weiss, 1988]

Genau diese Eigenschaften treffen auf HyperCard zu. Informationen sind auf den Cards, die durch Links untereinander verbunden sind. Ein Hypertext-System hat neben der nötigen Benutzerfreundlichkeit für eine hohe Akzeptanz auch mit andern Problemen zu kämpfen. Da ein Hypertext wie ein Graph zu betrachten ist und dieser sich bei einigen Anwendungen während der Lebenszeit verändert, können Links ihr Ziel verlieren oder ein Link auf ein falsches Ziel zeigen (Siehe Abschnitt 2.2). Janet Walker schrieb 1987 in „Document Examiner: Delivery Interface for Hypertext Documents“:

> „In most cases so far, the people building the delivery interface are also the people creating the underlying information structure that it is delivering.“Walker1987

Ob diese wertende Aussage nun wirklich zutrifft oder nicht soll an dieser Stelle nicht beurteilt werden. Allerdings würde bei einer Vermeidung der Dangling Edges schon bei der Erstellung und vor der Auslieferung, der letztendliche Nutzer dieses Problem niemals zu spüren bekommen.

Vannevar Bush schrieb vom menschlichen Verstand, „er funktioniert nicht auf diese Weise. Er arbeitet assoziativ.“ [Bush, 1945] [Life, 1945]. Doch das Lost-in-Hyperspace Problem aus Abschnitt 2.5 zeigt, dass Hypertext-Systeme rein nach seinem Konzept nicht unbedingt die optimale Lösung für den menschlichen Verstand sind. Zum Beispiel Hypertext-System wie Document Examiner und HyperCard helfen dem Nutzer mit Metaphern wie einem Buch oder einem Karteikartenstapel, sowie mit Funktionen wie dem Candidates Pane und einer Textsuche.

3 Literaturverzeichnis

[Bush, 1945] Bush, V. (1945). As we may think. *The Atlantic Monthly.*

[Conklin, 1987] Conklin, J. (1987). Hypertext: An intriduction and survey.

[Engelbart, 1968] Engelbart, D. (1968). Fall Joint Computer Conference, San Francisco.

[Engelbart, 1962] Engelbart, D. C. (1962). Augmenting human intellect a conceptual framework. *SRI Summary Report AFOSR-3223.*

[Engelbart, 1984] Engelbart, D. C. (1984). Authorship provisions in augment. *Proceedings of the COMPCON Conference, San Francisco.*

[Goodman, 1988] Goodman, D. (1988). *The complete HyperCard Handbook.* Bantam Books, second edition.

[Jakob Nielsen, 1991] Jakob Nielsen, Ida Frehr, H. O. N. (1991). The learnability of haypercard as an object-oriented programming system. *Technical University of Denmark, Department of Computer Science.*

[Life, 1945] Life (1945). As we may think. *Life Magazine.*

[Lloyd, 1969] Lloyd, G. (1969). Photo of the hypertext editing system (hes) console in use at brown university.

[Nelson, 1965] Nelson, T. H. (1965). A file structure for the complex, the changing, and the indeterminate. *Vassar College Poughkeepsie, N.Y.*

[Nelson, 1992] Nelson, T. H. (1992). *Literary Machines.* University of Michigan.

[Nielsen, 1995] Nielsen, J. (1995). *Multimedia and Hypertext: The Internet and Beyond.* Academic Press, Boston, MA.

[Paul Delany, 1995] Paul Delany, G. L. (1995). *Hypermedia and Literary Studies.* MIT Press.

[Smith and Weiss, 1988] Smith, J. B. and Weiss, S. F. (1988). Hypertext.

[van Dam, 1969] van Dam, A. (1969). *Manual for the Edit Phase of the Hypertext Editing System, for the IBM System/360 Model 50.*

[van Dam, 1988] van Dam, A. (1988). Hypertext '87 keybite address. *Communications of the ACM.*

[Walker, 1987] Walker, J. H. (1987). Document examiner: Delivery interface for hypertext documents.