

Hypertext-Systeme 1 (Lokale Systeme)

vorgelegt von:

Kevin Haack

Matrikelnummer: 7094226

Studiengang: Informatik (M.Sc.)

Thema betreut von:

Dr. Felix Winkelkemper

Paderborn, 18. Februar 2019

Hypertext-Systeme 1

(Lokale Systeme)

Abstract

In den 1980er Jahren entwickelten sich lokale (nicht zwangsweise netzbasierte) Hypertext-Systeme, wie Intermedia, HyperTIES und Hypercard.

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

Während der Begriff „Hypertext“ von Ted Nelson in den 1960ern geprägt wurde [Nelson, 1965], kann das Konzept von Hypertext auf die Beschreibung der „Memex“ von Vannevar Bush aus dem Jahr 1945 zurückgeführt werden [Bush, 1945].

„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.“ [Bush, 1945]

Fall Sie dieses Dokument in lesen, verwenden Sie auch Hypertext Hypertext kann man sich kaum vorstellen ohne Internet wie könnte das sonst aussehen? Diese Arbeit beschäftigt sich mit der Nutzungsgeschichte und ihre Entwicklung über die Zeit. Wie technische Entwicklungen Nutzungsformen hat entstehen lassen. Wie Probleme wie „dangling edges“ oder „lost in hyperspace“ entstehen und ob oder wie die Systeme mit diesen Problemen umgehen.

Inhaltsverzeichnis

| | | |
|----------|---------------------------------------|-----------|
| 1 | Was ist Hypertext? | 1 |
| 1.1 | Was ist Hypertext? | 1 |
| 2 | Geschichtliche Übersicht | 4 |
| 2.1 | NLS und AUGMENT | 5 |
| 2.2 | Problem: Dangling edges | 7 |
| 2.3 | HES | 7 |
| 2.4 | FRESS | 8 |
| 2.5 | Problem: Lost in Hyperspace | 9 |
| 2.6 | Document Examiner | 10 |
| 2.7 | HyperCard | 11 |
| 3 | Zusammenfassung | 15 |
| 4 | Literaturverzeichnis | 16 |

1 Was ist Hypertext?

1.1 Was ist Hypertext?

Was ist überhaupt Hypertext? Für eine Antwort auf diese Frage kann man sicherlich viele Antworten und Definitionen finden.

„[...] 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]

„A footnote is a classical form of a hyperlink. It's up to the reader to read – as you have just done – or to skip it. In this sense hypertext can be seen as the "generalized footnote", a metaphor taken from Jakob Nielsen's book Hypertext and Hypermedia.“ [Nielsen, 1990]

„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]

Gemeinsam haben viele, dass ein Hypertext aus diskreten Texten oder Textabschnitten besteht, zwischen denen eine Art Verlinkung existiere. Aber auch Verlinkungen zwischen Medien wie Audio-, Grafik- oder Videodateien können damit beschrieben werden. In diesem Zusammenhang verwendet Ted Nelson auch den Begriff „Hypermedia“ [Nelson, 1965]. Mit der Abbildung 1.1 zeigt Jakob Nielsen eine vereinfachte Darstellung eines Hypertextes im Ganzen. Anstatt eine feste Reihenfolge gibt es für den Leser mehrere Möglichkeiten die Texte aufzurufen. Nach Text *A* könne zum Beispiel sofort Text *D* folgen [Nielsen, 1995, S.1].

Eine Verlinkung könne, wie auch Jeff Conklin 1987 beschrieben hat, einen Token in einem Dokument als Start haben und als Ziel ein anderes Dokument [Conklin, 1987]. Wie auf Abbildung 1.2 zusehen, zeigt der Token *xxxx* im Dokument *A* auf das ganze Dokument *B*. Verlinkungen können aber auch von ganzen Dokumenten auf andere Dokumente zeigen, ohne dabei einzelne Token als Start zu haben.

Nodes und Edges erklären [Conklin, 1987]

1 Was ist Hypertext?

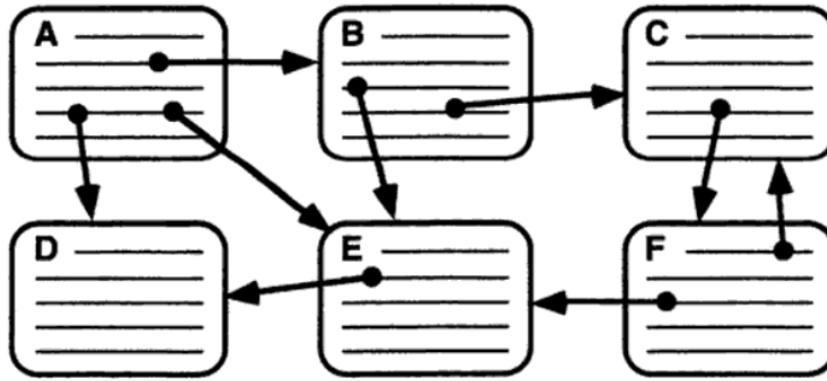


Abbildung 1.1: Vereinfachte Ansicht eines Hypertextes [Nielsen, 1995, S.1]

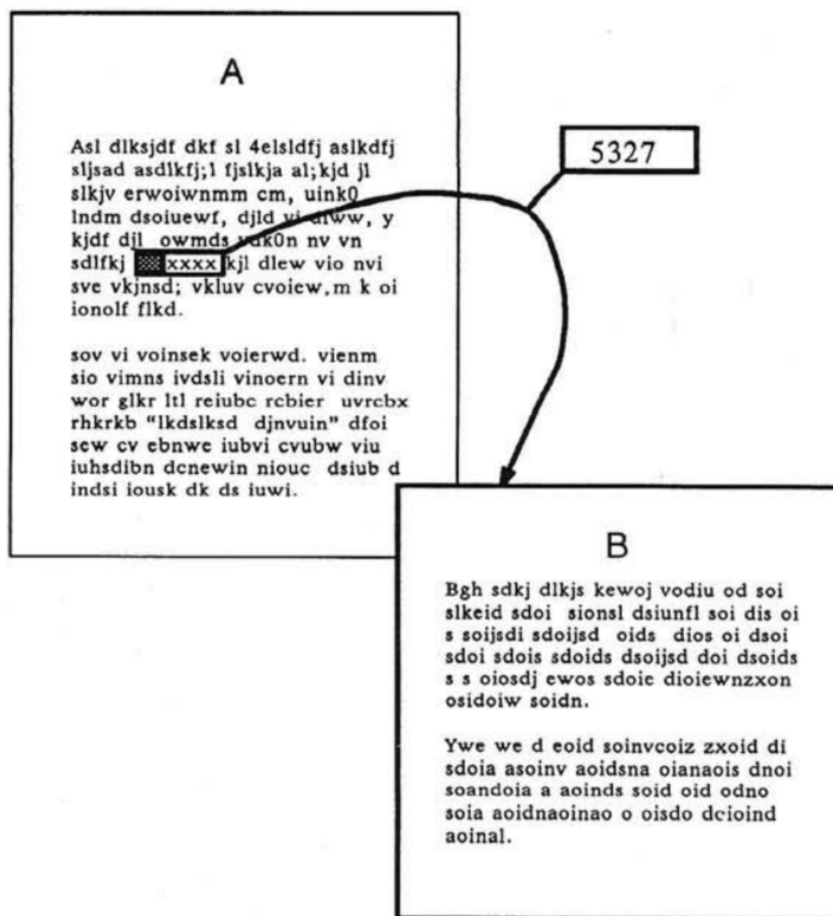


Abbildung 1.2: Ein Beispiel eines Links von in einem Text zu einem anderen Text [Conklin, 1987, S.34]

Auch viel später - 1995 - beschrieb Jakob Nielsen in dem Buch *Multimedia and Hptertext: The Internet and Beyond*, ein Hypertext-System ganz ähnlich: „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

1 Was ist Hypertext?

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]. Er beschrieb allerdings ein Hypertext-System als gänzlich ohne sequentielle Reihenfolge. Buchs „trails“ würden durch sogenannte „trail blazers“ erstellt [Nielsen, 1995, S.35].

2 Geschichtliche Übersicht

In der Geschichte von Hypertext 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].

Im Jahr 1945 wurde „As we may think“ von Vannevar Bush veröffentlicht. Mit dem Konzept „Memex“ erdachte Vannevar Bush eine Maschine für die individuellen Verwendung - eine private, 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]. In in diesem Konzept könne der Nutzer einen sogenannten „main trail“ aus Dokumenten erstellen. Dieser Trail solle eine Kombination aus Dokumenten sein, die aus der Perspektive des Nutzers von Interesse ist. Mit sogenannten „side tails“ könne der Nutzer ein Main Trail mit anderen Dokumenten „verlinkt“ [Bush, 1945, Section 7]. Diese Trails könnten als erstes Konzept von Hyperlinks verstanden werden. Zwei Jahrzehnte nach der Veröffentlichung von „As we may think“ , in der 1960er Jahren, wurde der erst der Begriff Hypertext von Ted 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 würde 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].

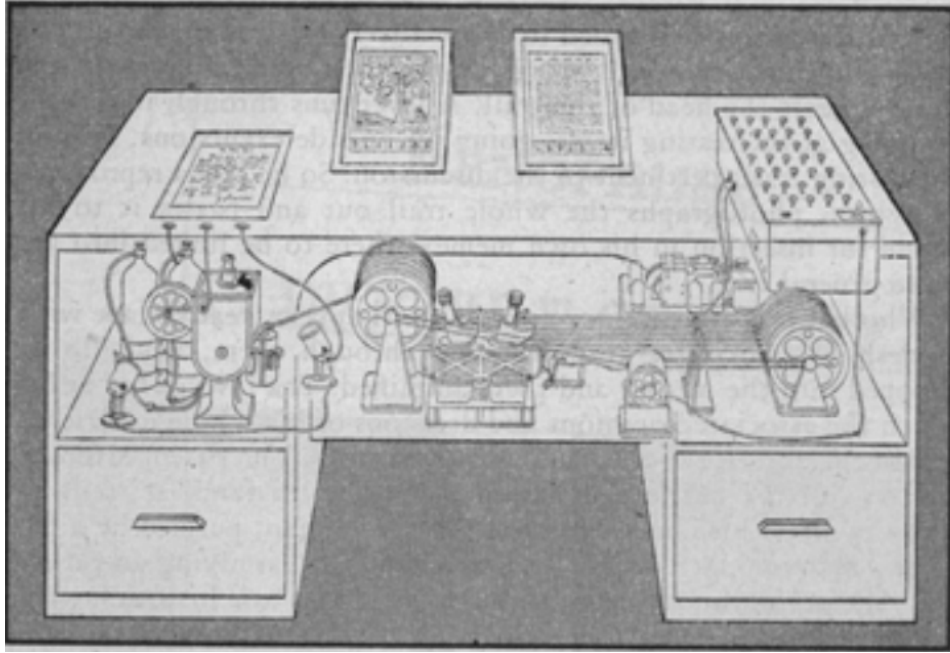


Abbildung 2.1: „Memex in the form of a desk would instantly bring files and material on any subject to the operator’s fingertips. [...] At left is a mechanism which automatically photographs longhead notes, pictures and letters, then files them in the desk for future reference“ [Life, 1945, S.123]

2.1 NLS und AUGMENT

Im gleichen Jahrzehnt erschien Doug Engelbarts „Augmenting Human Intellect: A Conceptual Framework“ .

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

Engelbart schrieb in „Augmenting Human Intellect: A Conceptual Framework“ von dem „digital computer as a tool for the personal use of an individual. Here there is not only promise of great flexibility in the composing and rearranging of text [...]“ [Engelbart, 1962, S. 17]. 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. Vorge stellt 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 Computermouse und die Organisation von Windows auf dem Bildschirm (Siehe Abbildung 2.2). Das NLS wurde präsentiert als „ein mächtiges Tool für die Arbeit eines Individuums, wenn er studiert, plant, designt, debuggt oder dokumentieren.“

2 Geschichtliche Übersicht

[Engelbart, 1968]. Es gäbe Möglichkeiten für das kollaborative Arbeiten, es gäbe gemeinsame Dokumente und man könne Nachrichten an Dokumente heften [Engelbart, 1968].

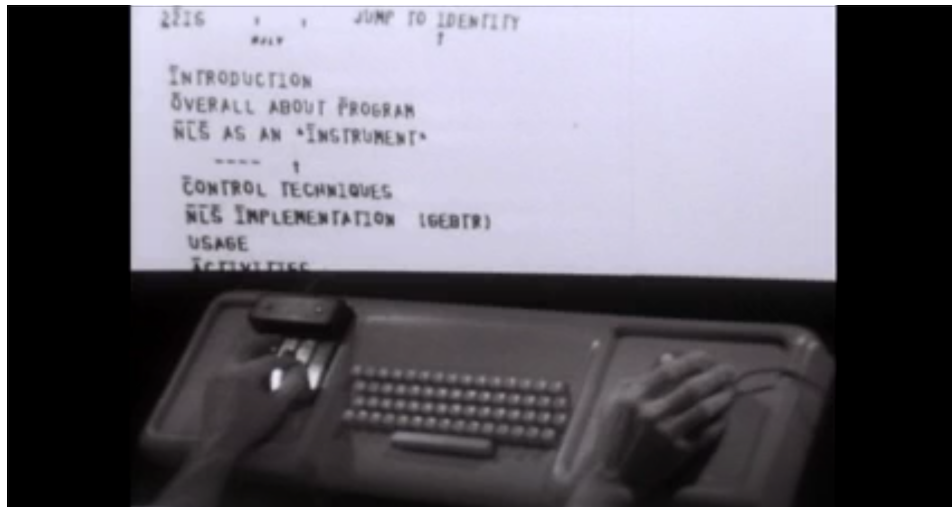


Abbildung 2.2: Doug Engelbart präsentiert auf der Fall Joint Computer Conference in San Francisco die Forschung vom Stanford Research Institute [Engelbart, 1968].

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 Adressierungsmöglichkeiten realisiert werden. Jedes Dokument in NLS ist in „Statements“ unterteilt und diese Statements konnten auf verschiedene Weisen verlinkt werden [Engelbart, 1984]. Jedes Statement hat eine „structural statement number“, eine eindeutige ID, die sich nach der Position in einem Dokument richtet. Nach dem Verändern des Textes konnten so Links zu falschen Statements führen. Der „statement identifiers“, als globale ID war unabhängig von der Position des Statements, diese ID blieb unverändert und „Labels“ die vom Nutzer definiert werden. Das Userinterface bot Kommandos zum Manipulieren der Oberfläche. Die interaktive Textbearbeitung unterstützte Funktionen wie Insert, delete, Move and Copy. Es konnten Outlines und eine Art 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. Die Arbeit des Individuums sollte optimiert werden. Die Memex als analoges Konzept und das NLS als digitale Lösung für ein Hypertext-System. Die Verknüpfungen der Dokumente in der Memex Maschine sind vergleichbar mit den verlinkten Statements im NLS, beide Verlinkungen wurden auch über IDs realisiert [Engelbart, 1984], [Bush, 1945]. Nur die Navigation durch die Links unterschied sich bei beiden, statt Touch 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 Nodes und Edges. Ein Text könnte ein Node sein und eine Verlinkungen zwischen zwei Texten ist eine Edge zwischen zwei Nodes. Durch entfernen von Nodes können „dangling edges“ entstehen. 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].

Je nachdem wer die Sachen bearbeitet, wird diese Problem erst relevant In most cases so far, the people building the delivery interface are also the people creating the underlying information structure that it is delivering.

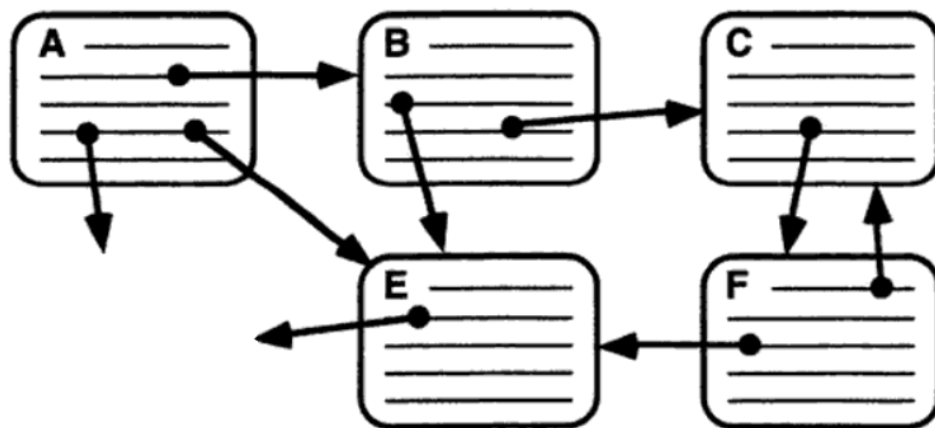


Abbildung 2.3: Modifizierte Darstellung von Nielsen, zur Darstellung von zweier Dangling Edges [Nielsen, 1995, S.1].

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, „Links“ und „Branches“. 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 Annotation Area auf, der Branch Explainer wurde in-line beim Branch Sternchen angezeigt. Der Nutzer konnte Sektionen mit dem Lightpen auswählen und einem einzigartigen Label versehen. Dieses Label wurde dann in eine Label Tabelle eingetragen werden und konnten mit dem Befehl besucht werden und als Pointer 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 Manipulation als Text Manipulation gewesen [van Dam, 1988, S. 890].

„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 vom Text entstanden keine Dangling Edges, wie in Abschnitt 2.2 beschrieben. Jeder Link oder Branch der auf die gelöschte Sektion zeigt wird ebenfalls gelöscht [van Dam, 1969, S.12].

2.4 FRESS

1968 startete van Dam zusammen mit seinen Studenten 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]. FRESS sollte auch auf verbreiteten kommerziellen Systemen laufen. Van Dam zufolge sei FRESS das erste System gewesen, dass eine UNDO Funktion unterstützte. Jede Bearbeitung wurde in einer Schattenversion gespeichert und erlaubte so ein Autosave und ein UNDO [van Dam, 1988, S.891]. 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



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

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. “ [und George P. Landow, 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 Text wurden die „cross-reference markers“ angezeigt, diese zeigten dem Nutzer alle Verlinkungen an und gaben dem Leser einen „Backtrack“ durch seine besuchten Links. Links 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 Funktionsumfang 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 viel zahl von Branches verlieren. So auch van Dam: „We already

started getting the notion that the richer the hypertext, the greater the navigational problem. “ . Nach van Dam sei das „come from“ genauso wichtig wie das „go to“ [van Dam, 1988].

„[...] 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-inhyperspace problem [...]“ [van Dam, 1988]

FRESS nutzte beim Aufruf von Links beispielsweise Windows und bidirektionale Links damit der Nutzer sich nicht im Hyperspace verloren geht. In der Memex wäre dieses Problem eventuell durch die zwei Anzeigen ein wenig entschärft worden, aber auch hier wäre man beim Navigieren durch die Dokumente irgendwann „lost in hyperspace“. Erst die bidirektionalen Links und Windows lösen dieses Problem und geben dem Nutzer die Möglichkeit seinen Weg zurück zugehen.

2.6 Document Examiner

Die ersten Hypertext-Systeme wie NLS und FRESS haben gezeigt, dass Hypertext nicht nur ein Konzept sind, sondern auch als echte Systeme implementiert werden können. In den achtziger Jahren kamen dann einiger 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 forschungs Zwecke, sondern kommerzielle Zwecke gedacht [Walker, 1987], [Nielsen, 1995, S. 50]. Anders als bei den 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 Buch Metapher 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]

Document Examiner wurde als „window-based“ bezeichnet, wie auf Abbildung 2.6 zusehen, unterteilen die Windows die Anwendung in vier Abschnitte.

Die Content 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 2.5 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 2.5: 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 Buch Metapher 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 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 sollte 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

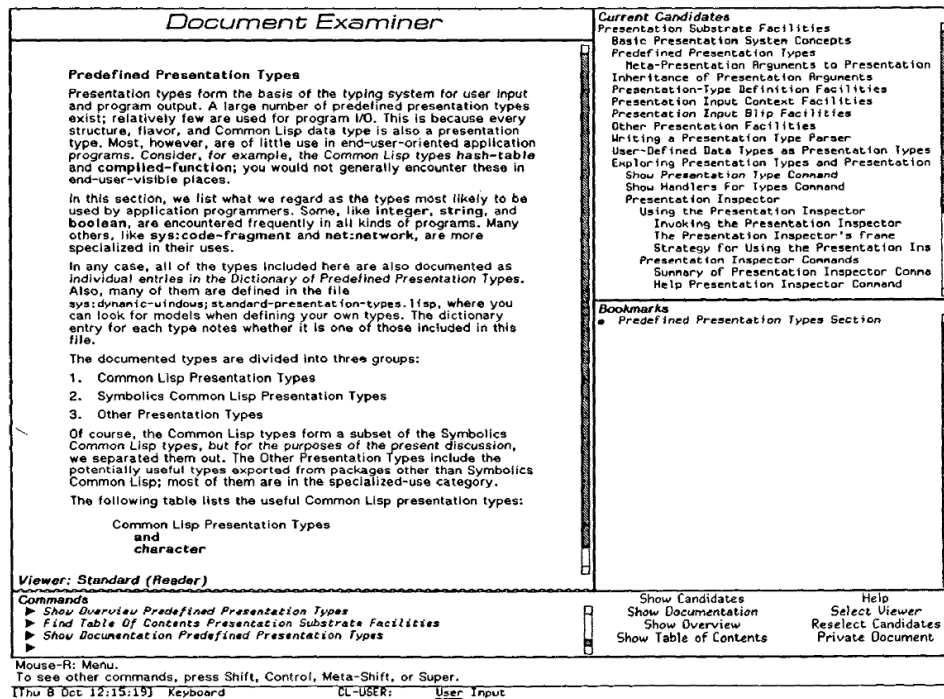


Abbildung 2.6: Userinterface von Document Examiner. Zeigt die vier Abschnitte: Content Area, Bookmark Pane, Candidates Pane und Command Region [Walker, 1987].

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]. Grundlegend besteht HyperCard aus zwei Sub-Systeme, zum Einen die Grafische Oberfläche und zum Anderen die HyperTalk Programmiersprache. Es gibt unter anderem fünf basis Element: Stacks, Backgrounds, Cards, Fields, Buttons [Goodman, 1988]. HyperCard macht sich in vielen Punkten die Karteikarten Metapher zunutze. Alle Informationen sind gleich Strukturiert und passen genau auf eine „Card“. Diese Card passt genau auf einen 9" Macintosh Bildschirm und in ein einen „Stack“ einsortiert. 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 2.7). 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].

Verlinkungen wurden HyperCard mit HyperTalk implementiert. Ein Beispiel hierfür ist auf der Abbildung 2.8 zusehen. In diesem Beispiel wurde der Hintergrund „Six Monthly“ gewählt. Beim Klicken auf eine Woche bestimmt das HyperTalk Skript auf welche Woche geklickt wurde und leitet den Nutzer auf die

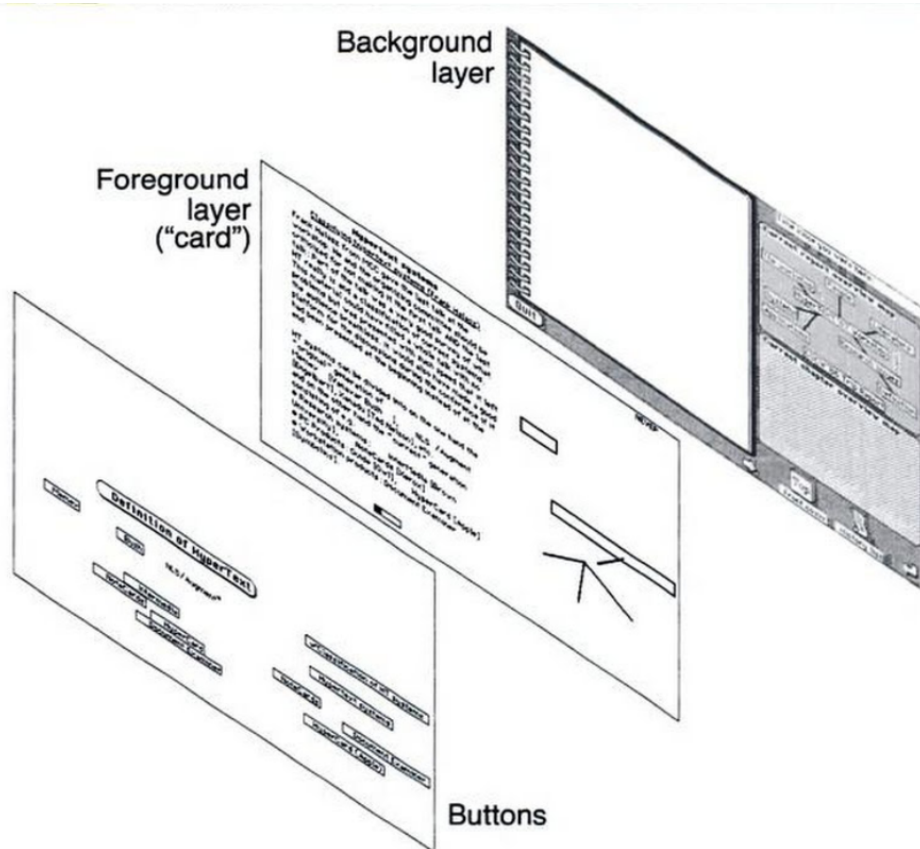


Abbildung 2.7: Vereinfachte Ansicht der HyperCard Layer [Nielsen, 1995].

entsprechende Wochenkarte weiter. Es könnten aber auch weitere Berechnungen und Bedingungen die Verlinkung beeinflussen [Goodman, 1988], so konnten beliebig komplexe Hypertext entstehen. Ein solcher Link war allerdings immer nur unidirektional, ein Skript auf der Zielkarte konnte aber eine Link zurück anbieten. HyperCard identifiziert seine Cards mit der Position innerhalb eines Stacks. Das Problem ist hierbei (wie in Abschnitt 2.2 beschrieben), dass wenn man zum Beispiel Card zwei in einem Stack löscht, rücken alle späteren Cards 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].

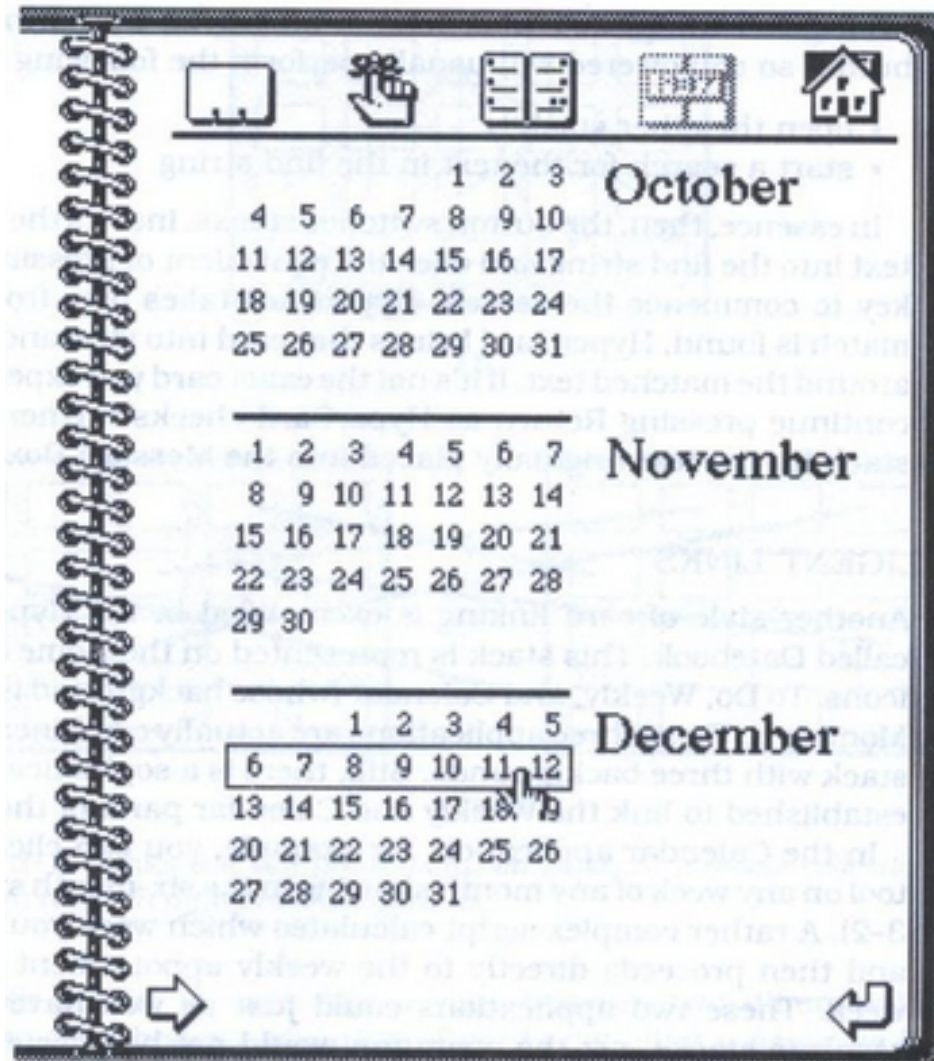


Abbildung 2.8: Ein Beispiel für einen Monat, dargestellt als Card in Hyper-Card. Beim Klick auf eine Woche wird die entsprechende Card im Termin Stack geöffnet [Goodman, 1988].

3 Zusammenfassung

Das Lost in Hyperspace Problem aus 2.5 zeigt, dass ein Hypertext-System eventuell nicht eine Erweiterung des Menschlichen Intellekts ist.

„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]

4 Literaturverzeichnis

- [Bush, 1945] Bush, V. (1945). As we may think. *The Atlantic Monthly*.
- [Conklin, 1987] Conklin, J. (1987). Hypertext: An introduction 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,] Lloyd, G. 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.*
- [Nielsen, 1990] Nielsen, J. (1990). *Hypertext and Hypermedia*. Academic Press, Boston, MA.
- [Nielsen, 1995] Nielsen, J. (1995). *Multimedia and Hypertext: The Internet and Beyond*. Academic Press, Boston, MA.
- [Smith and Weiss, 1988] Smith, J. B. and Weiss, S. F. (1988). Hypertext.
- [und George P. Landow, 1995] und George P. Landow, P. D. (1995). *Hypermedia and Literary Studies*. MIT Press.
- [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*.

4 *Literaturverzeichnis*

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