

Ein Bewertungsschema für HyperText-Systeme und ein vergleichender Überblick über HyperTies, HyperPad, HyperCard und Guide

Ch. Meyer, M. Rauterberg, M. Strässler
 Institut für Arbeitspsychologie (IfAP)
 Eidgenössische Technische Hochschule Zürich
 Nelkenstrasse 11, CH-8092 Zürich

Das hier vorgestellte Bewertungsschema soll primär dazu dienen, dem nicht professionellen Hypertextanwender (-Autor und -Leser) einen systematischen Überblick über Eigenschaften von Hypertextsystemen zu geben. Das Bewertungsschema enthält Eigenschaften, welche bei nicht-trivialen Hypertextentwicklungen empfehlenswert, bzw. unabdingbar sind. Wie man an der Schwerpunktsetzung zwischen Benutzersicht und Autorensicht erkennen kann, werden die Probleme des Hypertextautors und eine adäquate Unterstützung seitens der Entwicklungsumgebung noch zu sehr unterschätzt. Erst wenn entsprechend komfortable Entwicklungsumgebungen zur Verfügung stehen, werden Hypertextsysteme die ihnen zustehende Rolle einnehmen können.

1 Einführung

Der Begriff 'Hypertext' hat viele Facetten, und es gibt verschiedene Ansätze, sich ihm zu nähern. Jedoch trifft man stets, wenngleich in verschiedenen Varianten, auf die beiden folgenden Grundaspekte von Hypertexten:

Die **Struktur**: Jeder Hypertext ist ein *beliebiges Netzwerk* von strukturierten, informationstragenden *Knoten*; Verbindungspunkte im Innern der Knoten setzen diese über *Querverweise* zueinander in Beziehung.

Der **Zugriff** auf die Struktur: Jedes Hypertextsystem ermöglicht neben dem gewöhnlichen Direktzugriff auf einzelne Knoten auch die *Navigation* im Hypertextnetz, wodurch man jede gewünschte Knoteninformation mittels einer Wanderung entlang einem geeigneten Pfad des Querverweis-Netzwerks erreicht.

Diesen beiden Aspekten stehen zwei Grundkonzepte zur technischen Umsetzung auf Computern gegenüber. **Datenorientierte Hypertextsysteme**: die Knoten sind *Informationsspeicher*, die nur gelesen, also nicht verändert werden können; Navigation im Hypertext entspricht einem Suchzugriff auf eine netzwerkartig organisierte Datenbank. **Objektorientierte Hypertextsysteme**: die Knoten sind *Zustandsautomaten*, d.h. nach aussen abgeschlossene Objekte, die aus einem Informationsspeicher (Datenstrukturen) sowie aus *Transformationsregeln* für diesen Speicher (Operationen auf den Datenstrukturen) bestehen. Die Transformationen werden ausgelöst, indem die Bildschirmdarstellungen der Objekte manipuliert werden und die betreffenden Knotenobjekte sich darauf gegenseitig *Nachrichten* mit entsprechenden Transformationsanforderungen zusenden. Dabei muss zwischen den Nachrichtenpfaden und den Netzwerkpfaden kein Zusammenhang bestehen. Navigation im Objektnetzwerk dient zur Inspektion des Objektzustandes, d.h. des Speicherinhaltes, und zur Auslösung von Nachrichtenflüssen.

2 Grundlegende Konzepte

2.1 Daten- versus Objekt-Orientierung

Das **datenorientierte Hypertextkonzept** ist ein neuer Lösungsansatz für ein altes Problem. In herkömmlichen Texten reiht sich Satz an Satz: die sichtbare Form des Textes, seine typographische Organisation ist der Inbegriff der *Sequenz*. Die Welt, die mit dem Text beschrieben werden soll, ist jedoch meist nicht von solch sequentieller Natur, sondern besitzt im günstigen Fall eine *hierarchische*, im allgemeinsten Fall jedoch eine beliebige *netzartige*, also *zirkuläre* Beziehungsstruktur zwischen den zu beschreibenden Objektbereichen.

Im herkömmlichen Text gibt einzig die in ihm enthaltene Information, der Textinhalt, Aufschluss über die strukturellen Beziehungen in der dargestellten Wirklichkeit — die *äussere Form* hingegen ist nicht sehr aussagekräftig. Zum Teil lässt sich dieser Mangel mit Methoden der *Informationstypographie* beheben

(Seitenraster, Schriftwahl und Schriftmischung, Titelgestaltung, Auszeichnung von Textpassagen, Farben, graphische Elemente, Illustrationen). Mit rein typographischen Hilfsmitteln lässt sich aber höchstens eine *sequentiell dargestellte hierarchische Organisation* erreichen, wie sie auch diesem Aufsatz zugrunde liegt: Kapitel gliedern sich in Abschnitte, diese in Absätze bzw. einzelne Punkte. Aber schon ein umfangreicher Fussnotenapparat oder gar eine echt nicht-hierarchische Kapitelstruktur lassen diese Bemühungen am sequentiellen Grundprinzip des gedruckten Textes scheitern.

Erheblich näher an die Lösung des Problems, die Form des Textes mit den abzubildenden Beziehungsstrukturen in Übereinstimmung zu bringen, kommt man mit der Methode der 'Strukturbilder' oder **Blockdiagramme**. Der sequentielle Text wird in Teilsequenzen, d.h. in Textblöcke zerlegt und diese mit einfachen graphischen Elementen wie Striche, Pfeile und Rahmen zweidimensional zueinander in Beziehung gesetzt. Bei grösseren Textmengen versagt aber auch dieser Ansatz, zwar keineswegs prinzipiell, jedoch aus rein praktischen Gründen.

Hingegen wird auf *Computersystemen* das Blockdiagramm im Rechnerspeicher repräsentiert und bei Bedarf auszugs- oder blockweise in gewohnter Darstellung auf einen Bildschirm gebracht. Das ist bereits im wesentlichen, was als 'Hypertext' bezeichnet wird: eine Menge von gespeicherten Textfragmenten (Kapitel, Absätze), die durch elektronische Verbindungen zu einer Beziehungsstruktur verbunden werden. Deshalb auch die Bezeichnung: ein 'Hypertext' ist ein '**Über-Text**', eine Textsammlung mit einer Überstruktur. Darüberhinaus umfasst das Hypertextkonzept nicht nur Information in Form von Schrift, sondern auch in Form von Bildern und bewegten Bildsequenzen, d.h. Animation. Kommt noch Ton und Film hinzu, so spricht man meist nicht mehr von 'Hypertext', sondern von '*Hypermedia*'.

Objektorientierte Hypertextsysteme sind dedizierte objektorientierte Programmiersysteme, die für die Programmierung von Interaktion im Knotennetzwerk ausgelegt sind. Ein objektorientierter Hypertext ist demnach ein Programm, bestehend aus untereinander vernetzten Knotenobjekten, welche die eigentliche Information in ihren Datenspeichern halten und bei Erhalt von entsprechenden Aufforderungen transformieren können. Die Knotenobjekte werden aus vorgegebenen Grundobjekten aufgebaut, z.B. Textfelder zur Informationsvermittlung oder Bedienungsfelder zur Auslösung von Transformationen, und auf dem Bildschirm graphisch dargestellt.

Das datenorientierte Konzept ist ein Spezialfall des objektorientierten, wo auf die Knoteninhalte ausser den Navigationsoperationen keine Transformationen wirken, die 'Datenobjekte' also unveränderlich sind. Trotz dieser scheinbaren Einschränkung verkörpert das datenorientierte Konzept das eigentliche Grundprinzip von Hypertexten; die Objektorientierung ist in diesem Sinne eine nicht unbedingt notwendige, aber ausserordentlich leistungsfähige und flexible Erweiterung. Natürlich sind in konkreten Anwendungsfällen beliebige Übergangsformen zwischen den beiden Positionen denkbar, d.h. datenorientierte Hypertexte können auch objektorientierte Komponenten aufweisen, indem sie auch Knoten oder Knotenbestandteile enthalten, die vom Benutzer transformiert werden können.

2.2 Informationsstrukturierung

Allgemein betrachtet ist Hypertext eine *Methode zur Wissensrepräsentation*, d.h. ein Werkzeug zur gezielten Erschliessung von Wissen in Form strukturierter begrifflicher und anschaulicher Information über einen Ausschnitt aus unserer Innen- und Umwelt. Das Wissen soll in Form von Text und Bild in einer Maschine nachgebildet werden und gezielt abrufbar sein. Man könnte Hypertext zudem als einen graphentheoretischen Ansatz zur Informationsstrukturierung einordnen. Der allgemeinste Graph, das Netzwerk, ist gut geeignet, um das Beziehungsnetz zwischen in sich geschlossenen Informationseinheiten zu modellieren und damit Wissen auf einen Graphen abzubilden. Entsprechend der Terminologie der Graphentheorie werden die abgeschlossenen Informationseinheiten Knoten genannt, die Kanten des Graphen jedoch als Verweise ('Link') bezeichnet.

Ein Hypertext unterscheidet sich jedoch von einem gewöhnlichen Graphen darin, dass ein Hypertextknoten kein null-dimensionaler Punkt ist, sondern eine zwei-dimensionale (Bildschirm)- Fläche mit innerer Text-Bild-Struktur. Die Anfangs- und Endpunkte der Verweise, welche im folgenden als Verweispunkte bezeichnet werden, können im Knoten beliebig angeordnet sein. Die Verweise führen aus dem Inneren des Ausgangsknotens heraus in den Zielknoten. Dabei kann es sich auch um den gleichen Knoten handeln, denn in der Regel enthält ein Knoten eine Vielzahl von Verweispunkten, und diese können auch im Knoteninnern untereinander vernetzt sein. Im folgenden wird bei graphentheoretischen Interpretationen des Hypertextnetzes stets von der inneren Knotenstruktur abgesehen und innere Verweise als Schlingen im Graph betrachtet.

Die Informationsstruktur von Hypertexten beruht demnach auf drei Grundelementen:

- 2.2 a Der **Knoten** ist die abgeschlossene Informationseinheit des Hypertextes. Jeder Knoten ist durch einen Namen, einen Titel oder sonstige Klassifizierungsinformation eindeutig identifiziert und kann eine beliebige Sequenz von Text, Bild und Animation enthalten. Die Informationsmengen in den Knoten können sehr unterschiedlich sein, von einem einzigen Wort über einen Bildschirminhalt bis zu einem ganzen Buch; sinnvollerweise wird man jedoch den Knotenumfang auf ein Kapitel oder einen inhaltlich abgeschlossenen Abschnitt festlegen. Im Unterschied zu einem gedruckten Text sind die Knoteninhalte bei objektorientierten Hypertexten prinzipiell, bei datenorientierten in Sonderfällen (objektorientierte Bestandteile, z.B. Textfelder für Anmerkungen und Notizen) veränderbar.
- 2.2 b Jeder Knoten kann eine beliebige Anzahl **Verweispunkte**, d.h. Anfangs- und Endpunkte von Verweisen enthalten. Verweispunkte können natürlich auch geschachtelt werden, z.B. kann ein Satz, der als Anfangspunkt für einen Verweis dient, ein Stichwort enthalten, das ebenfalls als Anfangspunkt für einen weiteren Verweis verwendet wird. Als Verweispunkte können beliebige Knotenbestandteile dienen, im besonderen:
- 2.2 ba **STICHWÖRTER**, vergleichbar mit Querverweisen in Lexika, aber auch Wortgruppen und ganze Sätze.
- 2.2 bb in sich geschlossene **KNOTENTEILE**, wie Fussnoten, Absätze und Kapitel; auch der Knoten als Ganzes kann Verweispunkt sein (vgl. Punkt 2.2 d).
- 2.2 bc **BILDKOMPONENTEN**, die z.B. in der Art einer Explosionszeichnung dargestellt und ausgewählt werden können.
- 2.2 bd **KONTROLLFELDER** (z.B. Dialogboxen, etc.) zur Interaktion mit dem Knoten bzw. dem Knotennetzwerk; solche Simulationen von 'Bedienungskonsolen' mit Ein- und Ausgabefeldern sind für objektorientierte Hypertexte von grundlegender Bedeutung.
- 2.2 c **Verweise** verbinden genau je zwei Verweispunkte miteinander, innerhalb desselben Knotens oder ausserhalb. Navigation in einem Hypertext bedeutet, dass man im aktuellen Knoten einen Anfangspunkt auswählt und über den Verweis automatisch zum Endpunkt in einem neuen Knoten gelangt und hier erneut einen Anfangspunkt auswählt. Jeder Verweis ist wie ein Knoten eindeutig identifiziert und gehört einem bestimmten Typ an. Der **Verweistyp** ist bestimmt durch:
- 2.2 ca **RICHTUNG**. Analog zur gerichteten Kante in einem Graphen werden beim gerichteten Verweis Anfangs- und Endpunkt unterschieden, beim ungerichteten können die zugeordneten Verweispunkte sowohl Anfangspunkt als auch Endpunkt sein. Von jedem Anfangspunkt geht genau ein Verweis aus; in einen Endpunkt können beliebig viele gerichtete und, falls sonst kein Verweis wegführt, genau ein ungerichteter Verweis einlaufen.
- 2.2 cb **GEWICHTUNG**. Analog zu einem bewerteten Graphen kann einem Verweis ein Zahlengewicht zugeordnet werden, z.B. je nach Wichtigkeit oder Detailliertheitsgrad der zugehörigen Verweispunkte.
- 2.2 cc **ORDNUNG**. Je nachdem, ob der Verweis auf ein übergeordnetes Stichwort, einen untergeordneten Begriff, auf ein verwandtes Thema, ein Gegenbeispiel oder eine zugeordnete Fussnote verweist, hat er den Typ einer Überordnung, Unterordnung, Nebenordnung, Gegenordnung oder Zuordnung. Zusammen mit der Gewichtung ist die Ordnung von grosser Bedeutung für die Informationswiedergewinnung (siehe Kap. 2.3).
- 2.2 cd **DARSTELLUNGSFORM**. Wird ein Verweis aktiviert, d.h. ein Anfangspunkt auf dem Bildschirm ausgewählt, so gibt es verschiedene Möglichkeiten, den Endpunkt auf der zweidimensionalen Bildschirmfläche in sichtbaren Bezug zum Anfangspunkt zu setzen:
- cda **Einschub**. Ist der Endpunkt ein Textabschnitt oder ein Bild, das den Ausgangstext oder das Ausgangsbild inhaltlich und formal passend erweitert, so wird der Endpunkt in der Darstellung des aktuellen Knotens an der entsprechenden Stelle eingefügt. Der Anfangspunkt wird sozusagen 'aufgefaltet' und sein Inhalt, der Endpunkt, frei gelegt. Ist ein Hypertext rein hierarchisch organisiert, so kann er ausschliesslich mit (in sich geschachtelten) Einschüben dargestellt werden.
- cdb Liegt der Endpunkt in einem gänzlich anderen inhaltlichen Umfeld als der Ausgangspunkt oder hat eine unvereinbare äussere Form (z.B. bei Verweisen von Bildern heraus auf Text), so ist ein **Einschub nicht sinnvoll**:
- cdba **Ersetzung**. Der Knoten, der den Endpunkt enthält, ersetzt den aktuellen Ausgangsknoten. Diese Darstellung muss gewählt werden, wenn auf der Darstellungsfläche, d.h. auf dem Bildschirm nicht mehr genügend Platz verfügbar ist.
- cdbb **Paralldarstellung**. Der Knoten, der den Endpunkt enthält, wird parallel zum aktuellen Knoten dargestellt. Diese Darstellungsform ist wegen der Übersichtlichkeit gegenüber der Ersetzung vor

zuziehen. Ist ein Hypertext netzartig organisiert, so muss im allgemeinen eine dieser beiden Darstellungsformen verwendet werden.

cdc Überlagerung. Hat der Endpunkt den Charakter einer Fussnote oder Anmerkung, also einer erläuternden, nur kurzzeitig benötigten Hilfsinformation, so wird diese vorübergehend dem aktuellen Knoten in der Darstellung überlagert und nach Gebrauch wieder entfernt. Bei allen Darstellungsformen ist zu beachten, dass nur die Darstellung der aktuellen Knoten transformiert wird, nicht aber die Knoteninhalte an sich.

Bisher war nur von einem einzigen, gesamthaften Netzwerk die Rede. Analog zu gedruckten Publikationen lässt sich die Vernetzung jedoch in weitere **Strukturebenen** aufspalten:

2.2 d Das **Knotennetz** besteht aus denjenigen Verweisen, die ausschliesslich ganze Knoten sowohl zum Anfangs- als auch zum Endpunkt haben. Das Knotennetz repräsentiert die thematische Gliederung des im Hypertext gespeicherten Wissens, es ist sozusagen das Inhaltsverzeichnis des Hypertextes. Analog zur Schachtelung von Kapiteln in konventionellen, hierarchischen Inhaltsverzeichnissen können Teilnetze des Knotennetzes zu Überknoten zusammengefasst und als ein einzelner Knoten behandelt werden.

2.2 e Das **Verweisnetz** besteht aus allen Verweisen, die keinen ganzen Knoten zum Anfangs- und Endpunkt haben. Das Verweisnetz bestimmt die detaillierte Beziehungsstruktur des Hypertextes und entspricht z.B. dem System von Querverweisen in Lexika oder den kapitelübergreifenden Hinweisen in Lehrbüchern. Es wäre denkbar, das Verweisnetz nochmals aufzuspalten, und zwar in abgeschlossene, untereinander unabhängige Verweisnetzebenen, die sich alle auf dasselbe Knotennetz beziehen, aber z.B. für verschiedene Benutzerkreise bestimmt sind.

2.2 f Der **Thesaurus** erlaubt die direkte Lokalisierung aller Verweispunkte, die den Charakter eines Stichwortes haben, also nur aus einzelnen Zeichen, Wörtern oder kurzen Wortverbindungen bestehen. Ausserdem hat er die Funktion eines Glossars, enthält also Kurzerklärungen zu den verzeichneten Stichwörtern. Der Thesaurus besteht aus zwei Komponenten:

2.2 fa ALPHABETISCHES STICHWORTVERZEICHNIS (Index). Bei der automatischen Ableitung dieses Verzeichnisses müssen orthographische Abweichungen, unterschiedliche Flexionen und Numeri der Stichwörter auf die entsprechenden Grundformen zurückgeführt werden.

2.2 fb SYSTEMATISCHES STICHWORTVERZEICHNIS. Dieses Verzeichnis lässt sich zum Teil aus dem Ordnungstyp der Verweise herleiten. Die Stichwortbegriffe lassen sich nach folgenden Beziehungsgruppen klassifizieren:

fba sinngleiche (synonyme) und sinnverwandte Begriffe, gegensinnige (antonyme) Begriffe;

fbb übergeordnete (hyperonyme) und untergeordnete (hyponyme) Begriffe;

fbc sachverwandte Begriffe.

2.3 Informationswiedergewinnung

'*Information Retrieval*', d.h. Wiedergewinnung von Information, ist in allen Informationssystemen eine zentrale Tätigkeit. Es gibt im wesentlichen zwei Verfahren zur Informationswiedergewinnung. Das **deskriptive Verfahren**: Die gewünschte Information wird in einem iterativen Ablauf immer genauer umrissen, bis eine ausreichende Genauigkeit der Beschreibung erreicht ist und die gesuchte Information abgerufen werden kann. Zur Beschreibung werden mengentheoretische, boolesche und prädikatenlogische Operatoren eingesetzt, die auf die sogenannten *Deskriptoren* angewendet werden. Die Menge der Deskriptoren stellt eine kompakte Beschreibung der zu durchsuchenden Textmenge dar; vor allem handelt es sich dabei um Stichwörter und Begriffe, die mit Zeichenkettenoperationen zugegriffen werden können. Das deskriptive Verfahren ermöglicht somit einen Direktzugriff, der von der konkreten Netztopologie unabhängig ist, jedoch entscheidend von der zur Verfügung stehenden Deskriptorenmenge bestimmt wird. Dabei muss die inhaltliche Struktur sowohl der gesuchten Information als auch ihres Umfeldes ungefähr bekannt sein, da sich die Beschreibung darauf bezieht und nur so eine Lokalisierung möglich ist. Der deskriptive Ansatz eignet sich für die systematische, flächendeckende Suche in unbeschränkten Themenbereichen bei fortlaufender Einengung des Themenkreises. Für die oberflächliche Durchsicht und die Übersichtsgewinnung in unbekanntem Bereichen ist diese Methode ungeeignet. Das deskriptive Verfahren ist die klassische Methode zur Informationswiedergewinnung in herkömmlichen Datenbanken. In Hypertextsystemen wird es bei Zugriffen auf Thesauri benötigt (vgl. Kap. 2.2 f).

Das **navigierende Verfahren**: Obwohl diese Methode auch bei herkömmlichen Datenbanken, z.B. relationalen, eingesetzt werden kann, erreicht sie ihre volle Leistungsfähigkeit und den grössten Kontrast zur

deskriptiven Methode erst mit dem Hypertext-Konzept, da die Netzwerkstruktur optimal mit der anschaulichen Vorstellung von 'Navigation' übereinstimmt. Deshalb wird im folgenden die ganz spezielle, hypertextspezifische Variante des navigierenden Verfahrens beschrieben.

Es handelt sich um eine *exploratorische* Methode. Ausgehend von einem Einstiegspunkt in das Netzwerk bewegt sich die Suche entlang eines frei wählbaren Verweispfades durch die Knoten des Netzwerkes. In jedem Knoten muss entschieden werden, welcher Verweispunkt als nächster ausgewählt wird; bei mehreren Möglichkeiten müssen die Endpunkte mittels geeigneter Deskriptoren zur Auswahl gestellt werden. Im Gegensatz zum deskriptiven Verfahren, wo ein bestimmtes Vorwissen über die abzurufende Information notwendig ist, kann man sich beim selektiven Vorgang der Navigation auf den inhaltlichen Kontext des umgebenden Knotens und des bisher beschrittenen Pfades abstützen und so problemlos auch in unbekanntes Gebiet vorstossen. Navigation ermöglicht, ganz im Unterschied zur Deskription, dass Umfang und Beziehungsstruktur der Information, die irgendwo unsichtbar in den Tiefen des Systems lagert, dem Benutzer offenbar und fassbar werden.

Das navigierende Verfahren ermöglicht im wesentlichen zwei verschiedene **Zugriffsarten** auf das Netzwerk:

Suchen nach einem unklaren Begriff oder einem nicht näher bestimmten Themenbereich: Befindet man sich in einem unbekanntem Wissensgebiet ohne begriffliche Anhaltspunkte, so kommt man nur mit einer navigierenden Methode zum Ziel. Bei bekanntem Begriff oder klar umrissenem Thema ist der deskriptive Zugriff über Thesauri dagegen einfacher und effizienter. Man kann deshalb zuerst über den Thesaurus eine Groblokalisierung vornehmen, anschliessend mit navigierendem Suchen das Ziel genau ansteuern.

Sichten eines ganzen Bereichs: Soll ein unbekannter Teil des Verweisnetzes gezielt oder spielerisch erforscht werden, so ist das Sichten oder 'Browsen', wie es auch genannt wird, eine sehr intuitive und effiziente Methode. Navigierendes Sichten ist die eigentliche Stärke des Hypertext-Konzepts, denn es entspricht genau der gewöhnlichen Methode, ohne bestimmte Anhaltspunkte auf gedruckte Informationssammlungen zuzugreifen: man durchstöbert die Seiten, liest hier etwas und dort, folgt einem Querverweis, merkt sich interessante Stichwörter. Bei komplizierten, hochgradig vernetzten, d.h. zyklischen Strukturen ist es jedoch sehr schwierig, einen grösseren Bereich systematisch und flächendeckend zu sichten.

Bei der Navigation, sowohl beim Suchen als auch beim Sichten, ist es sehr wichtig, während der Wanderung durch das Verweisnetz die Übersicht zu behalten und den eigenen Standort bestimmen zu können (deshalb auch die Bezeichnung 'Navigation'). Wie die Erfahrung zeigt, können schon einfache zyklische Strukturen ungemein verwirren. Deshalb ist mit der Informationsgewinnung untrennbar stets auch die Übersichtsgewinnung verbunden. Grundprobleme bei der Gewinnung von Übersicht sind: Reduktion der Informationsmenge, Verringerung der Strukturkomplexität, Visualisierung komplizierter Beziehungsgeflechte.

3 Die beiden Sichten auf HyperTexte

3.1 Die Benutzersicht: Interaktion mit Hypertexten

Der Benutzer eines Hypertextes entspricht dem herkömmlichen Leser von gedruckten Publikationen. Ein Buch zu benutzen bereitet dem geübten Leser keine Probleme. Allein dadurch, dass er das Buch in der Hand hält, seine Seitenzahl ermittelt und die Schriftgrösse in Betracht zieht, kann er den Informationsgehalt des Werkes abschätzen. Auch über die Struktur ist er sich völlig im klaren: das Buch ist physisch eine Folge von nummerierten Seiten, logisch gemäss dem hierarchischen Inhaltsverzeichnis organisiert und über ein Stichwortverzeichnis direkt zugreifbar. Die Orientierung zu verlieren ist nicht möglich, da die Stelle, an der das Buch aufgeschlagen ist, bereits rein optischen Aufschluss über die ungefähre Lese-Position gibt. Hingegen sieht sich der Leser eines Hypertextes einer ganzen Reihe von Problemen gegenüber. Der Bildschirm bietet dem Benutzer stets nur einen kleinen Ausschnitt dar; die gesamte zur Verfügung stehende Information ist in ihrem Umfang und in ihrer physischen Struktur anschaulich nicht fassbar. Die neuen Möglichkeiten zur Informationswiedergewinnung führen zu einem Orientierungsverlust, der durch eine künstliche Visualisierung der logischen Organisation und der Position im Netzwerk ausgeglichen werden muss. Ein Hypertextsystem muss daher gewissen Anforderungen genügen. An erster Stelle kommen sicher Anforderungen bezüglich **Leistungsfähigkeit und Ergonomie der Hardware**. Wichtig sind vor allem: eine möglichst intuitive, direkte Schnittstelle zum Rechner, da die Navigation ein höchst interaktiver Vorgang ist; eine möglichst angenehme Informationsaufnahme ab Bildschirm, da man in Hypertexten sehr oft auch längere Texte lesen muss; möglichst kurze Antwort- und Reaktionszeiten, besonders bei Massenspei-

cherzugriffen, da Hypertext sehr grosse Datenmengen mit sich bringen kann. Ebenso wichtig sind Forderungen bezüglich **Leistungsfähigkeit und Ergonomie der Software**. Neben allgemeinen Anforderungen wie Betriebseffizienz und Zuverlässigkeit sind speziell folgende Punkte zu beachten:

- 3.1.a **Interaktion**. Das Dialogsystem deckt die interaktiven Aspekte des Hypertextsystemes ab und bestimmt damit unmittelbar die ergonomischen Qualitäten von Hypertext. Eine Kommandooberfläche ist für die Zwecke von Hypertext ungeeignet und für Gelegenheitsbenutzer unzumutbar; besonders bei einer ausgesprochen interaktiven Tätigkeit wie Navigation (siehe Punkt b) ist die direkte Manipulation mittels Zeigens auf die Verweispunkte unbedingte Voraussetzung.
- 3.1.b **Navigation**. Das Hypertextbetriebssystem ist für die Navigation zuständig. Im einzelnen sind folgende Punkte zu betrachten:
- 3.1.ba **MODELLVORSTELLUNG DES BENUTZERS**. Der Mensch hat allgemein die hochentwickelte Fähigkeit, Vorgänge und Sachverhalte, die ihm nicht oder nur teilweise einsichtig sind, zu abstrahieren und sich eine Modellvorstellung zu bilden. Von der Qualität dieses gedanklichen Modells hängt direkt die Qualität des sachbezogenen Verhaltens und Handelns ab. Das gilt im besonderen für die Benutzer von Software-Systemen, da die Programme unsichtbar in der Maschine ablaufen, die Datenmengen meist sehr kompliziert strukturiert sind und die Maschinenlogik der Alltagserfahrung widersprechen kann. Auch bei Hypertext muss der Benutzer dazu angeleitet werden, sich eine klare, adäquate Modellvorstellung von der vorliegenden Netzstruktur und den darauf wirkenden Operationen zu bilden. Vor allem müssen gedankliche Strukturkonflikte verhindert werden, da sie die Modellbildung und somit die rationelle Bedienung erschweren. So muss z.B. der Unterschied zwischen Knotennetz, Verweisnetz und Thesaurus dem Benutzer jederzeit präsent sein, denn jede der drei Strukturebenen ermöglicht einen eigenen gedanklichen Zugang zum Hypertext, was die Navigationsstrategie des Benutzers erheblich beeinflusst.
- 3.1.bb **FORTBEWEGUNG IM NETZWERK**. Informationswiedergewinnung in einem Hypertext erfordert Fortbewegung in dessen Informationsnetzwerk, d.h. Suchen entlang geeigneter Verweispfade. Bei Verzweigungsalternativen müssen leicht handzuhabende Entscheidungshilfen angeboten werden. Falls der Pfad in einer Sackgasse endet, sollte dem Benutzer ein brauchbarer Mechanismus zur Wegrückverfolgung, zum 'Backtracking', zur Verfügung stehen. Um rationell arbeiten zu können, müssen häufig benutzte Zielpunkte und Pfade markiert werden können.
- 3.1.bc **ORIENTIERUNG IM NETZWERK**. Da der Bildschirm nur einen eng beschränkten Ausschnitt der Netzwerklandschaft darstellen kann, ist der Benutzer stets gefährdet, die Orientierung zu verlieren. Wichtigster Grundsatz ist: der Benutzer muss jederzeit seinen Standort im Netzwerk bestimmen und notfalls zu einem ihm wohlbekanntem Ausgangspunkt zurückkehren können. Damit die Position bestimmt werden kann, muss in der Bildschirmdarstellung eine Informationsreduktion erfolgen. Dazu wird mit Hilfe der Verweisgewichtungen und -ordnungen (Kap. 2.2 c) die strukturelle Information des betreffenden Netzwerkausschnittes in Übersichtsdarstellungen soweit reduziert, bis charakteristische Strukturmerkmale zutage treten, die der Benutzer mit seiner Modellvorstellung identifizieren und lokalisieren kann.
- 3.1.bd **SICHERN UND WEITERVERWENDEN VON ARBEITSRESULTATEN**. Will man nicht nur im Datenbestand herumstöbern, so müssen komfortable Möglichkeiten vorhanden sein, Informationsfragmente aus dem Netzwerk dauerhaft zu extrahieren, zu ordnen und weiterzuverwenden. Dabei ist zu beachten, dass weder das bestehende Netzwerk verändert wird noch ein neues Netz vom Benutzer konstruiert werden soll. Die interessanten Knotenteile werden lediglich aus dem Netzwerk herauskopiert und in konventioneller Weise sequentiell, allenfalls hierarchisch zusammengestellt, formatiert und nach Bedarf ausgedruckt.
- 3.1.c **Anpassungsfähigkeit**. Ein weiteres Mass für die ergonomische Qualität bilden die Möglichkeiten, die dem Benutzer offenstehen, um einen gegebenen Hypertext seinen Bedürfnissen individuell anzupassen.
- 3.1.ca **ANPASSUNG DES DIALOGSYSTEMS**. Soll Hypertext einer breiten Benutzerschicht zugänglich werden, muss das Dialogsystem sehr anpassungsfähig sein. Das heisst: eine ausführliche, schrittweise Dialogführung für den Anfänger, eine knappe, abgekürzte für den Fortgeschrittenen. Im Idealfall ist das Dialogsystem komplett nach individuellen Ansprüchen reorganisierbar.
- 3.1.cb **ANPASSUNG DER NETZSTRUKTUR**. Das Hypertextkonzept lässt veränderliche Knoteninhalte zu (Objektorientierung), aber es verbietet, dass der Benutzer die Vernetzungsstruktur modifiziert. Solchen Anpassungswünschen sind deshalb enge Grenzen gesetzt. Denkbar wäre, selten verwendete Netz

werkteile zu entfernen bzw. das wirklich benötigte Teilnetzwerk aus dem Gesamtnetz herauszulösen, was einen entsprechend modularen Netzentwurf voraussetzt (siehe Kapitel 3.2).

Nachstehend werden aus den obigen Betrachtungen die Schlüsse gezogen und **konkrete Realisierungsvorschläge** bezüglich der entsprechenden Systemkomponenten aufgelistet:

3.1.d **Bedienungsschnittstelle**

3.1.da Der BILDSCHIRM muss unbedingt Grossformat besitzen, mindestens A4-hoch, aber noch besser A3-quer. Eine grosse Darstellungsfläche ist eine einfache und sehr effiziente Massnahme, einen Teil der Desorientierungsprobleme vollständig zu lösen. Ausserdem muss der Bildschirm eine exzellente Darstellungsqualität besitzen, d.h. er muss den Text in hoher Auflösung, randscharf, flimmer- und reflexfrei in schwarzer Schrift auf papierweissem Hintergrund abbilden. Einen Hypertext zu benutzen heisst auch vor allem zu lesen, und deshalb dürfen bezüglich Bildschirmqualität keine Kompromisse eingegangen werden, da sonst die Gebrauchstauglichkeit von Hypertexten ernsthaft gefährdet ist. Farbbildschirme können im Bereich der Orientierung sehr gute Dienste leisten, allerdings wirkt sich ein übermässiger oder unzweckmässiger Einsatz von Farbe eher kontraproduktiv aus. Als Pendant zum Bildschirm muss der DRUCKER gleich hohe Qualitätsansprüche erfüllen, denn er verhilft der gewonnenen Information erst zu einer dauerhaften, computerunabhängigen Repräsentation.

3.1.db Als EINGABEGERÄTE eignen sich die herkömmliche Tastatur und Zeigegeräte, wie Maus und Steuermakel. Die Tastatur wird hauptsächlich zur Dateneingabe verwendet und dient nur optional zur Verkürzung von direktmanipulativen Sequenzen. Für öffentlich zugängliche Hypertextanwendungen sind auch berührungsempfindliche Bildschirme geeignet, da sie den intuitivsten, direktesten Zugriff auf die dargestellten Objekte ermöglichen.

3.1.dc Die BEDIENUNGSOBERFLÄCHE muss den Punkten 3.1.da und 3.1.db Rechnung tragen und deshalb für direkte Manipulation und Fenstertechnik ausgelegt sein. Die direkte Manipulierbarkeit ergibt sich aus der Verwendung von Zeigegeräten, die Fenstertechnik aus den grossen Bildschirmformaten und den hypertextspezifischen Erfordernissen: jeder Knoten lässt sich als Fenster darstellen und zu anderen Fenstern des Bildschirms in Beziehung setzen (2.2 cdb). Ausserdem sind Kontrollfelder (2.2 bd) und überlagernde Meldungen (2.2 cdc) integrale Bestandteile von direkt manipulierbaren Fensteroberflächen, welche damit als einziger Typus von Bedienungsoberflächen alle darstellungsmässigen Anforderungen erfüllen.

3.1.e Das **Dialogsystem** muss von allen Möglichkeiten der direkt manipulierbaren, fensterorientierten Bedienungsoberfläche Gebrauch machen: durchgehende Menüsteuerung, konsequente Fensterdarstellung, graphische Repräsentation der zu manipulierenden Objekte. Das Dialogsystem muss ausserdem möglichst einfach und möglichst umfassend an die sehr unterschiedlichen Bedürfnisse der einzelnen Benutzer anpassbar sein. Dazu bieten sich an: Parametermenüs zur Feinabstimmung des Systemverhaltens, Tastaturmakros zur Verkürzung von langen Aktionsfolgen, Ausblenden von nicht benötigten Menüpunkten bzw. Direktzugriff auf häufig benötigte, selbstdefinierbare Menüs zur Zusammenstellung von wichtigen Funktion und Tastaturmakros. Im Endeffekt ist das Dialogsystem vollständig reorganisierbar; um dies zu ermöglichen, ist es naheliegend, das Dialogsystem in der flexiblen Form eines objektorientierten Hypertextes zu implementieren.

3.1.f **Hypertextbetriebssystem**

3.1.fa ZUGRIFFSFUNKTIONEN. Dazu gehören alle Funktionen, die nötig sind, um in suchender oder sichtender Weise auf einen Hypertext zuzugreifen.

faa Direktzugriff. Ist man mit dem vorliegenden Hypertext bereits vertraut, hat sich also bereits eine Modellvorstellung des Netzwerks und des darin enthaltenen Wissens gebildet, so wird man eine bestimmte Position direkt ansteuern oder zumindest in unmittelbarer Umgebung einsteigen:

faaa *Zugriff über Einstiegspunkte*. Das 'elektronische Buchzeichen', also eine speziell definierte Position in einem bestimmten Knoten, ist der schnellste Weg, eine gewünschte Information wieder aufzufinden. Allerdings sollte die Zahl der Einstiegspunkte beschränkt sein, analog den richtigen Buchzeichen.

faab *Zugriff über den Thesaurus*. Im einfachsten Fall entspricht dies dem gewohnten Nachschlagen in einem alphabetischen oder thematischen Stichwortverzeichnis. Im allgemeinen Fall wird auf den Thesaurus mit dem in Kapitel 2.3 beschriebenen deskriptiven Verfahren zugegriffen.

faac *Zugriff über ein benutzerdefiniertes Stichwortverzeichnis*. Dies entspricht dem Thesauruszugriff, mit dem Unterschied, dass das dabei verwendete Verzeichnis neu erzeugt wurde, sei es automatisch mittels statistischer Auswertungen durch das System oder durch explizite Anweisung des Benutzers.

fab Navigierender Zugriff. Der navigierende Zugriff entspricht nichts anderem als einer Folge von Sprüngen von Verweispunkt zu Verweispunkt. Die Funktionalität des Sprunges wird bestimmt durch:

faba *Operationelle Funktionen*. Dazu gehören Grundfunktionen wie Zielauswahl bei mehreren Möglichkeiten und Pfadkorrektur durch Rücksprung um mehrere Schritte oder an einen Ausgangspunkt. Soll der beschrittene Weg später nochmals wiederholt werden, z.B. als 'Einführungstour' für einen ungeübten Benutzer, so muss der Pfad fortlaufend protokolliert und nach Wunsch dauerhaft gespeichert werden können.

fabb *Informatorische Funktionen*. Darunter fallen: Anzeige von Art und thematischer Zuordnung des angewählten Verweispunktes, Anzeige des Verweistyps, Hinweis auf allfällig vorhandene Synonyme, Hyponyme etc. Damit ausserdem der Benutzer sofort bemerkt, wenn er seinen Weg kreuzt oder gar in einen Zyklus geraten ist, müssen die bereits besuchten Knoten mit einer temporären Markierung versehen werden.

3.1.fb ORIENTIERUNGSFUNKTIONEN. Sich Orientieren heisst, sich in einer Darstellung zu lokalisieren. Deshalb geht es in den folgenden Punkten um die Visualisierung der verschiedenen Hypertextstrukturen:

fba Knotendarstellung. Der Inhalt des Knotens ist durch die zweidimensionale Sequenz des Textes bestimmt. Deshalb ist der Knoten ausschliesslich mit den Mitteln der Typographie übersichtlich zu gestalten. Besonders bei hochauflösenden Grossformatbildschirmen ist den typographischen Regeln und Gesetzmässigkeiten besondere Beachtung zu schenken. Leider werden oft die typographischen und graphischen Möglichkeiten entweder gar nicht wahrgenommen oder aber bis zum Exzess ausgereizt.

fbf Thesaurusdarstellung. Ist man im Verlauf der Navigation bei einem Stichwort angelangt, dessen Bedeutung oder thematische Zuordnung unklar ist, so müssen die entsprechenden Thesauruseinträge angezeigt werden. Dies geschieht am besten mittels überlagernder Hinweise, die anschliessend wieder entfernt werden.

fbc Netzdarstellung. Die Visualisierung des Knoten- und Verweisnetzes ist ein Kernproblem von Hypertextsystemen. Gewiss lässt sich mit rein numerischen, statistischen Beschreibungen bereits eine gewisse Übersicht verschaffen: Anzahl Knoten, Anzahl Verweise, Durchschnitt der Verweise pro Knoten, durchschnittliche Anzahl einlaufender Verweise pro Verweispunkt usw. Dazu gehören auch kompliziertere graphentheoretische Aussagen wie Anzahl und Grösse der Zyklen im Netzgraphen, durchschnittliche und maximale Pfadlängen. Für die Orientierung im Detail wird man jedoch das Netz in expliziter Form, textmässig oder graphisch, darstellen müssen.

fbca *Knotenetzdarstellung*. Ist das Knotennetz im wesentlichen hierarchisch, so lässt es sich rein textmässig darstellen, mit Einrückungen usw. Im allgemeinen Fall jedoch ist das Knotennetz nur graphisch in der Art eines Blockdiagramms darstellbar; die einzelnen Blöcke enthalten dabei den Knotentitel und allenfalls weitere Information. Um auch bei grossen Knotennetzen die Übersicht zu behalten, muss der aktuelle Netzausschnitt in verkleinerter Form in Zusammenhang mit seiner Umgebung oder im Gesamtzusammenhang betrachtbar sein; zudem müssen Netzteile temporär in Überknoten zusammengefasst werden können. Das System sollte automatisch diejenige Darstellung des Netzwerkgraphen wählen, die möglichst planar, d.h. überkreuzungsfrei ist.

fbcb *Verweisnetzdarstellung*. Während für das Knotennetz eine Blockdiagrammdarstellung noch einigermaßen realistisch ist, kann das Verweisnetz nur in einfachen Fällen mit Pfeilen und Kästchen visualisiert werden. Im allgemeinen scheidet dieses Vorhaben jedoch in einem undurchdringlichen Netzgewirr. Ausserdem zeigt sich ein Effekt, der bereits bei der Darstellung des Knotennetzes zu Schwierigkeiten führen kann: die optische Wirkung eines grossen Blockdiagramms, das im wesentlichen nur aus linearen Elementen, nämlich Linien und Text besteht, ist zu wenig differenziert, als dass es als charakteristische Form wahrgenommen werden könnte. Deshalb muss zumindest für grosse Verweisnetze eine alternative Form der Visualisierung gefunden werden. Es ist denkbar, dass mit der Metapher der 'Landkarte' und den Methoden der Topographie und Kartographie ein neuer Zugang zur Visualisierung von Verweisnetzen gefunden werden könnte. Der Benutzer könnte durch die verschiedenen Gegenden des Hypertextlandes schweifen, Planquadrat um Planquadrat erforschen und bekannte Punkte auch direkt auf dem Luftweg erreichen.

3.1.fc AUSGABEFUNKTIONEN

fca Extraktion nach einem Suchvorgang. Recherchen in Hypertexten sollen nicht nur zur 'Instant-Information' dienen, sondern auch dauerhafte Resultate hervorbringen. Zu diesem Zweck kann man die interessierenden Knoten oder Knotenteile aus dem Netzwerk extrahieren, d.h. kopieren, in eine beliebige lineare Folge einreihen, formatieren und abspeichern bzw. ausdrucken.

fcf **Zusammenfassung** während des Sichtens. Dieser Vorgang ist ähnlich zur Extraktion, aber mit folgenden Unterschieden: es wird fortlaufend extrahiert, also während der Wanderung durch das Netzwerk, und es werden nur kleine Knotenteile, meist nur Stichworte kopiert. Am Schluss wird die automatisch erstellte Liste der extrahierten Knotenabschnitte nach Wunsch umgestellt und weiterverarbeitet.

fcc **Linearisierung** eines Hypertexts. Soll ein Hypertext in Buchform umgesetzt werden, so muss das Netzwerk automatisch oder manuell in eine Sequenz aufgelöst werden. Die Linearisierung kann durch die Extraktion von Knoten des Netzwerkes erfolgen. Die Formatierung ist von grosser Wichtigkeit; sie gibt dem gedruckten Hypertext die erforderliche typographische Konsequenz, fügt die Fussnotenverweispunkte am richtigen Ort ein und ergänzt automatisch alle anderen Verweispunkte mit einem konventionellen Querverweis zur entsprechenden Seite. Zum Schluss werden ebenfalls automatisch das Inhaltsverzeichnis und der Thesaurus erzeugt.

3.2 Die Autorensicht: Konstruktion von Hypertexten

Der in Analogie zum Buchautor benannte Hypertextautor ist der Konstrukteur und Produzent des Hypertextes. Die gesamten Ausführungen dieses Kapitels gelten zwar ganz allgemein für Hypertexte, jedoch richtig bedeutsam werden sie erst für kompliziert strukturierte Hypertexte mit grossem Informationsgehalt. Die Konstruktion von Hypertexten ist ein technischer Vorgang. Die dabei auftretenden Probleme und die entsprechenden Methoden zur Lösung dieser Probleme haben sehr viele Gemeinsamkeiten mit der Softwaretechnik, bzw. dem Software-Engineering.

Problemanalyse. Wie auch bei einem Softwareprodukt muss bei einem Hypertextprojekt zuerst das gestellte Problem analysiert, d.h. Art, Umfang und Interaktivität des Wissens bestimmt werden, das durch den Hypertext vermittelt werden soll. Das Pflichtenheft konkretisiert dann die Informationsbedürfnisse, die der Hypertext gegenüber dem Benutzer befriedigen soll. Wie bei herkömmlicher Software sind in dieser Phase nicht so sehr hypertexttechnische Kenntnisse vonnöten, sondern vor allem fachtechnische und produktbezogene. Hingegen müssen nun alle folgenden Konstruktionsschritte in eine integrierte Produktionsumgebung eingebettet werden, wie dies für den Fall von herkömmlicher Software gilt. Dies ist eine unbedingte Voraussetzung, denn die meisten potentiellen Hypertextautoren werden kaum mit den elementaren und unabdingbaren Prinzipien der Softwaretechnik vertraut sein. Auf den Hypertextautor bezogen heisst das: Er muss dazu angehalten werden, sich bereits in einem frühen Entwurfstadium die Strukturierung seines Hypertexts zu überlegen und ihn nicht ziellos zu einem Netzgewirr wuchern zu lassen. In dieser Phase wird die Gebrauchstauglichkeit des Hypertextes entscheidend vorbestimmt.

Systemstrukturierung bedeutet in Hypertext, die Struktur des Knotennetzes zu entwerfen. Allgemein gelten bei der Netzkonstruktion folgende zwei Grundsätze: möglichst einfache Knotennetzstruktur und möglichst lose gekoppelte Netzknöten. Der erste Grundsatz verlangt, dass das darzustellende Gesamtthema mit einem minimalen Vernetzungsaufwand in die Einzelthemen, in die Kapitel aufgelöst wird, was sich direkt und sehr förderlich auf die Übersichtlichkeit auswirkt. Der zweite Grundsatz entspricht eigentlich dem ersten, nur dass nicht das Knotennetz, sondern das Verweisnetz betrachtet wird: Die Netzknöten enthalten in sich geschlossene Informationseinheiten, sozusagen 'Informationsmodule'. Vermeidet man unnötige Abhängigkeiten, d.h. Verweise zwischen den Knöten, so vereinfacht sich das Verweisnetz und die Orientierung wird unmittelbar erleichtert. Daraus folgt, dass man der Desorientierung nicht nur mit ausgefeilten Navigationsoperationen begegnen muss, sondern auch mit einer handlichen, problemangemessenen Netzstruktur. Diese an sich selbstverständliche Erkenntnis muss nun die Produktionsumgebung in den Entwurfsprozess einbringen, indem der Hypertextautor in seinen Strukturentscheidungen unterstützt und geführt wird, das heisst, indem der Autor stets die Übersicht über den Entwurfsprozess behält und angehalten wird, nur im tatsächlichen Bedarfsfall kompliziertere Netzstrukturen zu benutzen.

Im folgenden werden einige der wesentlichen Eigenschaften für den Autor von Hypertexten aufgeführt.

3.2.a **eingebettete Programmiersprache:** da der Hypertextautor sich während der Erstellungsphase ständig in der zu schaffenden Hypertextstruktur aufhält, ist es wichtig, dass der Wechsel zwischen dem Ausführungsmodus und dem Entwicklungsmodus rasch von statten gehen kann. Dieser Umstand hat auch zur Folge, dass der Hypertextautor auf dieselben Unterstützungs- und Hilfsmittel angewiesen ist, wie der spätere Hypertextleser.

3.2.aa Die **EINBINDUNG EXTERNER PROZEDUREN** ist deshalb nützlich, weil es oftmals für spezielle Aspekte nicht die entsprechenden Ausdrücke in der jeweiligen Hypertextprogrammiersprache gibt.

- 3.2.ab Ein Hypertextsystem, welches den IMPORT VON TEXTEN nicht zumindest teilweise automatisch unterstützt, ist für echte Hypertextanwendungen ungeeignet. Hierzu zählt generell der Import von vorhandenem Datenmaterial.
- 3.2.b Da heutzutage mit die wichtigsten Güter 'Informationen' sind, sollte ein Hypertextsystem als Informationssystem gegen unbefugten Gebrauch mittels eines **Passwortsystems** schützbar sein.
- 3.2.c Bei grossen Hypertexten wird man um ein kooperatives Arbeiten und Entwickeln nicht umhin kommen. Eine wesentliche Voraussetzung dafür ist die **Netzwerkfähigkeit**.
- 3.2.d Je nach konkretem Verwendungszusammenhang des Hypertextes ist es nötig, eine **Runtime-Version** erstellen zu können. Dies ist z.B. bei reinen Auskunftssystemen gegeben. Wenn jedoch auf dem Hypertext selbst fortlaufend Veränderungen vorzunehmen sind, so muß die Entwicklungsumgebung zur Verfügung stehen.

4 Fazit

Je nach Verwendungszweck des zu erstellenden Hypertextes wird die Bewertung der zur Auswahl stehenden Hypertextsysteme unterschiedlich ausfallen. Das allen Ansprüchen gerechtwerdende und preiswerte Hypertextsysteme gibt es bisher (noch) nicht. Die wohl wichtigste Unterscheidung ist die Wahl zwischen einem *daten-* und einem *objekt-*orientierten Hypertextsystem. Während das daten-orientierte Hypertextsystem die Erstellung von Hypertexten mit vorwiegendem Textcharakter dadurch optimal unterstützt, dass es die Verweispunkte unmittelbar an den semantischen Kontext koppelt, wird demgegenüber bei den hier untersuchten objekt-orientierten Hypertextsystemen primär nur eine rein syntaktische Lösung für die Koppung zwischen Verweispunkten und semantischem Kontext angeboten. Dies wirkt sich unmittelbar auf den Entwicklungsaufwand aus. So können zwar bei HyperCard auch Texte grundsätzlich mit Verweispunkten ausgestattet werden, dennoch müssen bei einer Änderung der Textstruktur alle Verweispunkte 'von Hand' nachjustiert werden; dies gilt z.B. bei HyperCard auch für die Verweispunkte mit visuellem semantischen Kontext. Bei grossen Hypertexten z.B. mit HyperCard sind Änderungen ohne entsprechende automatische Unterstützung seitens der Entwicklungsumgebung praktisch undurchführbar; deshalb erscheint es notwendig, die Konzepte von Guide in HyperCard einzubringen. Die Stärke von objekt-orientierten Hypertextsystemen liegt sicherlich darin, dass die von den Objekten ausgesandten Nachrichten den Hypertext zum 'Leben erwecken'. Dies führt dann unmittelbar zu verstärktem Einsatz von Bildern, bis hin zu Animationseffekten (siehe 'Hypermedia'-Systeme).

Hypertextsysteme nehmen eine Schlüsselstellung zwischen traditionellen Datenbanksystemen, modernen Textverarbeitungssystemen und Systemen zur Verwaltung semantischer Netzwerke ein. Erst wenn eine entsprechende automatische Unterstützung des Hypertext-Lesers und -Autors zur Gewährleistung ihrer Orientierung in den verschiedenen Netzstrukturen eines Hypertextes gegeben ist, wird die Entwicklung und Nutzung von Hypertexten einen breiten Einsatz erfahren können. Diese automatische Unterstützung setzt voraus, dass das Wissen über die hypertext-spezifischen Strukturen in das Hypertextsystem implementiert ist.

5 Ein vergleichender Überblick

Im folgenden werden die wesentlichsten Beschreibungskategorien aus dem zuvor dargestellten Bewertungsschema hergenommen, um vier am Markt erhältliche Hypertextsysteme mit einander zu vergleichen. Das Produkt 'Guide' lag sowohl in der Version für den Macintosh, als auch in der Windows-Version für MS-DOS Rechner vor. HyperPad ist das entsprechende Gegenstück auf MS-DOS Maschinen zu dem bekannten HyperCard auf Macintosh-Rechnern.

		Programm-Name	Hyper- ties Ms-DOS	Hyper- Pad Ms-DOS	Guide Windows	Guide Macintosh	Hyper- Card Macintosh
Symbol	Bedeutung	Versions-Nr.	*	1.0	2.00	2.0	2.0
++	gut unterstützt		10	12	24	24	19
+	teilweise unterstützt		3	3	1	1	9
(+)	"von Hand" behelfsmäßig möglich		3	5	4	4	8
leer	nicht vorhanden		30	31	22	22	15

		Betriebssystem des Rechners	MS-DOS	MS-DOS	MS-DOS Windows	Macintosh	Macintosh
		Programm-Name	Hyper-ties	Hyper-Pad	Guide	Guide	Hyper-Card
Kapitel	Bewertungsbegriffe	Versions-Nr.	*	1.0	2.00	2.0	2.0
2.1	DATENORIENTIERT		++		++	++	
2.1	OBJEKTORIENTIERT			++			++
2.2a	Knoten		++	++	++	++	++
2.2b	Verweispunkte						
2.2ba	Stichwörter, Wortgruppen		++		++	++	(+)
2.2bb	beliebige Textbereiche				++	++	(+)
2.2bc	Bilder (als Ganzes)			++	++	++	++
2.2bc	Bildteile						+
2.2bd	Kontrollfelder, Dialogboxen			++			++
2.2c	Verweisattribute						
2.2ca	gerichtet		++	++	++	++	++
2.2ca	ungerichtet			(+)			(+)
2.2cb	Gewichtung						
2.2cc	Ordnung				++	++	(+)
2.2cd	Darstellung eines Verweises						
2.2cda	Einschub				++	++	
2.2cdba	Ersetzung		++	++	++	++	++
2.2cdbb	Paralleldarstellung				++	++	+
2.2cde	Überlagerung				++	++	+
	Strukturebenen						
2.2d	Knotennetz		++	++	++	++	++
2.2e	Verweisnetz						
2.2f	Thesaurus						
2.2fa	alphabetisches Stichwortverzeichnis		++	(+)	(+)	(+)	(+)
2.2fb	systematisches Stichwortverzeichnis						
2.2fba	nach Synonymen / Antonymen		(+)	(+)	(+)	(+)	(+)
2.2fbb	nach Über-/ Unterordnungen		(+)	(+)	(+)	(+)	(+)
2.2fbc	nach sachverwandten Begriffen		(+)	(+)	(+)	(+)	(+)
2.3	deskriptiver Zugriff		++	+	+	+	+
2.3	navigierender Zugriff		++	++	++	++	++
3.1d	Bedienungsschnittstelle						
3.1da	Grossformatdarstellung (A3)				++	++	++
3.1da	Positivdarstellung				++	++	++
3.1da	Farbdarstellung			++			
3.1da	hochauflösende Grafikdarstellung				++	++	++
3.1da	verschiedene Schriftarten und -grössen				++	++	++
3.1db	Mauseingabe			++	++	++	++
3.1dc	Fenstertechnik				++	++	++
3.1e	Menüs		+	++	++	++	++
3.1e	selbstdefinierbare Menüs						++
3.1fa	Zugriffsfunktionen						
3.1faaa	über Einstiegspunkte						
3.1faab	über Thesaurus						
3.1faac	über benutzerdef. Stichwortverzeichnis						
3.1faba	Backtracking			+	++	++	+
3.1faba	Pfadprotokollierung		++				+

		Betriebssystem des Rechners	MS-DOS	MS-DOS	MS-DOS Windows	Macintosh	Macintosh
		Programm-Name	Hyper- ties 1)	Hyper- Pad	Guide	Guide	Hyper- Card
Kapitel	Bewertungsbegriffe	Versions-Nr.	*	1.0	2.00	2.0	2.0
	Informationsfunktionen						
3.1fabb	Anzeige Verweistyp		+		++	++	+
3.1fabb	Markierung besuchter Punkte						
3.1fbc	Netzdarstellung						
3.1fbca	Knotennetzdarstellung						
3.1fcb	Verweisnetzdarstellung						
3.1fc	Ausgabefunktionen						
3.1fca	Extraktion			+			+
3.1fcb	Zusammenfassung						
3.1fcc	Linearisierung				++	++	
3.2	Funktionen für Autoren						
3.2a	eingebettete Programmiersprache		*	++			++
3.2aa	Einbindung externer Prozeduren		*				++
3.2ab	automatischer Import von Texten		*				
3.2b	Passwortschutz		*				++
3.2c	netzwerkfähig		*				
3.2d	Runtime-Version möglich		+		++	++	+

++ gut unterstützt:

+ teilweise unterstützt:

(+ 'von Hand' behelfsmässig möglich, jedoch unpraktikabel;

leer nicht vorhanden.

* nur Runtime-Version "Hands-On!" untersucht;

6 Weiterführende Literatur

- Barrett E. (1988): Text, ConText, and HyperText - writing with and for the computer. Cambridge: MIT Press.
- Conklin J. (1987): Hypertext: An Introduction and Survey. IEEE Computer. Sept. 87, pp.17-40.
- Fiderio J. (1988): A Grand Vision. Byte. Oct. 88, pp. 237-244.
- Frisse M. (1988): From Text to Hypertext. Byte. Oct. 88, pp. 247-253.
- Goodman D. (1988): The complete HyperCard Handbook. (2nd edition), New York: Bantam Books.
- Gloor P. & Streitz N. (1990; Hrsg.): Hypertext und Hypermedia. Heidelberg New York: Springer Verlag.
- Hofmann M., Cordes R. & Langendörfer H. (1989): Hypertext/Hypermedia. Informatik-Spektrum. Band 12 Heft 4, Aug. 89, S. 218-219.
- Hypertext'87 Papers. Technical Report no 88-013. Department of Computer Science, The University of North Carolina at Chapel Hill, USA, 1988.
- Marchionini G. & Shneiderman B. (1988): Finding Facts vs. Browsing Knowledge in Hypertext Systems. IEEE Computer. Jan. 88, pp.70-80.
- McAleese R. (1989; ed.): Hypertext: theory into practice. Osney Oxford: Blackwell Scientific Publ.
- Monk A. (1989): The Personal Browser: a tool for directed navigation in hypertext systems. Interacting with Computers. vol. 1 no. 2, pp.190-196.
- Nelson T. (1967): Getting It Out of Our System. in: Information Retrieval: A Critical Review. (Schechter G., ed.), Washington: Thompson Books.
- Nielson J. (1990): Trip Report: Hypertext'89. SIGCHI Bulletin. vol. 21 no. 4, pp. 52-61.
- Pollitt A.S. (1989): Information Storage and Retrieval Systems - origin, development and applications. Chichester: Ellis Horwood.
- Ranade J. (1989): On-Line Text Management - Hypertext and other Techniques. New York: Intertext Publ.
- Shneiderman B. (1987): Designing the User Interface. Reading, Mass.: Addison-Wesley Publ.
- Shneiderman B. & Kearsley G. (1989): Hypertext Hands-On! Reading, Mass.: Addison-Wesley Publ.
- Stepno B. (1989): A HyperCard for the PC. Byte. Sept. 89, pp.189-192.

J. Encarnação (Hrsg.)

Telekommunikation und multimediale Anwendungen der Informatik

GI - 21. Jahrestagung
Darmstadt, 14. – 18. Oktober 1991

Proceedings



Springer

1991

Berlin Heidelberg New York London Paris
Tokyo Hong Kong Barcelona Budapest