

Heft 87

M. Nüttgens, G. Keller, A.-W. Scheer, S. Stehle

**Konzeption
hyperbasierter Informationssysteme**

Dezember 1991

Inhaltsverzeichnis

| | | |
|-----------|--|-----------|
| 1. | Problemstellung | 1 |
| 2. | Charakterisierung des Hyper-Konzeptes | 2 |
| 2.1 | Überblick | 2 |
| 2.2 | Hyper-Strukturen | 4 |
| 2.2.1 | Knoten (nodes) | 5 |
| 2.2.2 | Kanten (links) | 6 |
| 2.2.3 | Orientierungs- und Navigationshilfen | 8 |
| 3. | Gestaltungsaspekte hyperbasierter Informationssysteme | 10 |
| 3.1 | Nichtlinearität und informelle Informationsstrukturen | 10 |
| 3.2 | Multimedialität | 11 |
| 3.3 | Mensch-Computer Schnittstelle | 12 |
| 3.4 | Integrierende Aspekte | 14 |
| 4. | Umsetzungsaspekte hyperbasierter Informationssysteme | 15 |
| 4.1 | Aufgabenbezogene Navigationssysteme | 15 |
| 4.2 | Interaktive Tutorial- und Trainingssysteme | 19 |
| 4.3 | Integrierte Dokumentationssysteme | 24 |
| 5. | Modellierungsaspekte hyperbasierter Informationssysteme | 27 |
| 5.1 | Hyper-Engineering | 27 |
| 5.2 | Hyper-Informationsmodell | 29 |
| 6. | Zusammenfassung und Ausblick | 32 |
| 7. | Literaturverzeichnis | 33 |

1. Problemstellung

Mit der zunehmenden Verfügbarkeit leistungsfähiger Informations- und Kommunikationstechniken eröffnen sich für die Gestaltung der Mensch-Computer Schnittstelle neue Dimensionen. Unter den Schlagworten "Objektorientierung" und "Hypermedia" werden hierbei konzeptionelle Ansätze diskutiert, welche den Anwender integrierter Informationssysteme in die Lage versetzen sollen, die zunehmende Komplexität dieser Systeme zu beherrschen.

Mit dem Hyper-Konzept wird die Zielsetzung verfolgt, die Benutzerschnittstelle von der Systemfunktionalität zu entkoppeln und die Modellierung der Benutzerschnittstelle als eigenständige Entwurfsaufgabe abzugrenzen.

Das objektorientierte Paradigma kann mit seinen Schlüsselbegriffen Objekt, Nachricht, Klasse, Instanz und Vererbung charakterisiert werden.

Das Hyper-Konzept verkörpert die konsequente Übertragung des objektorientierten Paradigmas auf den Entwurf und die Implementierung integrierter Benutzerschnittstellen. Es kann unmittelbar auf die Problemstellung integrationsgerechter Benutzerschnittstellen angewandt werden und unterstützt den Entwurf realitätsnaher Modell-Handlungswelten.

Auch der Mensch geht bei der Erfassung komplexer Systeme intuitiv objektorientiert vor. Um Systeme verstehen zu können, strukturiert er sie nach Objekten mit gleichen Eigenschaften und Verhalten und deren Wirkungszusammenhänge. Somit besteht eine weitreichende Analogie zwischen der Denkweise des objektorientierten Ansatzes, den abzubildenden Strukturen der realen Welt und den mentalen Modellen eines Benutzers. Die Kommunikation zwischen Benutzer und DV-Anwendung rückt damit näher an die eigentliche Arbeitsaufgabe heran.

Nachfolgend werden die Charakteristika des Hyper-Konzeptes herausgearbeitet, die Gestaltungsaspekte hyperbasierter Informationssysteme aufgezeigt und Umsetzungsaspekte für betriebliche Anwendungen am Beispiel von Prototypen diskutiert. Die Hyper-Werkzeuge (Hyper-tools/-shells) zur Umsetzung des Hyper-Konzeptes werden nur insofern diskutiert, als sie das "Vehikel" zur Umsetzung des sachlogischen Konzeptes in eine DV-technische Realisierung darstellen. Im "Hyper-Informationsmodell" sind abschließend die zentralen Objekte eines hyperbasierten Informationssystems formal beschrieben.

2. Charakterisierung des Hyper-Konzeptes

2.1 Überblick

Eine erste Vision dessen, was heute als Hypertext bezeichnet wird, veröffentlichte Bush 1945 in seinem Artikel "As We May Think" [1]. Darin entwarf er, angesichts der Wissensexplosion, ein Gerät namens Memex (memory extender) auf der Basis von Mikrofilm- und Photozellentechnologie. Memex sollte sowohl zur Speicherung von vorgefertigten Informationen, wie mikroverfilmte Bücher, Zeitschriften und Abbildungen dienen als auch zur Speicherung von benutzerindividuellen Informationen wie handschriftliche Notizen und Korrespondenz. Vor allem sollte Memex "assoziatives Indizieren" unterstützen: " ... Grundgedanke ist eine Vorrichtung wodurch jeder Gegenstand auf Wunsch sofort und automatisch einen anderen Gegenstand auswählen kann" [2]. Memex wurde nie realisiert, der Grundgedanke des Verknüpfens von Informationseinheiten wurde jedoch in Hypertext übernommen.

Das Wort Hypertext selbst wurde von Ted Nelson geprägt. In einem 1967 erschienenen Beitrag definiert er Hypertext als "eine Kombination aus natürlichsprachlichem Text und der Fähigkeit des Computers zu interaktivem Verzweigen oder dynamischer Darstellung" [3]. Mit Xanadu entwarf er ein universales System in dem alle Texte der Welt gespeichert und miteinander verbunden werden sollten. Außerdem entwickelte Nelson für Xanadu bereits eine strikte Trennung von Datenbasis und Benutzerschnittstelle. Xanadu wurde später als Prototyp teilweise realisiert.

Vollständig realisiert wurden in den sechziger Jahren die Systeme NLS (oNLine System) und Hypertext Editing System. Das am Stanford Research Institute von Engelbart entwickelte NLS war eine Arbeitsumgebung für Wissenschaftler. Es gab diesen die Möglichkeit ihre Arbeitspapiere, Designentwürfe, Berichte und Notizen zu speichern und mit Querverweisen zu verbinden. Zudem unterstützte NLS alle Projektphasen der Softwareentwicklung, die Anwendung von Informationsfiltern, Mehrpersonenbetrieb und Kommunikation. NLS wurde 1968 auf der Fall Joint Computer Conference der Öffentlichkeit vorgestellt und später unter dem Namen Augment kommerziell weiterentwickelt [4].

Hypertext Editing System wurde an der Brown University von van Dam entwickelt. Es bot

[1]: Bush, *As We May Think*, 1945.

[2]: "... the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another." Bush, *As We May Think*, 1945; zitiert nach: Nielsen, *Hypertext*, 1990, S. 30.

[3]: "a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ..." Nelson, *Getting it Out*, 1967; zitiert nach: Conklin, *Introduction*, 1987, S. 17.

[4]: Conklin, *Introduction*, 1987, S. 22-23.

zusätzlich zu den üblichen Textverarbeitungsfunktionen die Möglichkeit Texte miteinander zu verknüpfen. Das System wurde zur Dokumentation der Apollo-Missionen eingesetzt [5].

In den siebziger und achtziger Jahren wurden erstmals Systeme mit Hypertext-Merkmalen für die Anwendung außerhalb universitärer Forschungslabore entwickelt. Interaktive Auskunftssysteme und Handbücher wie die Aspen Movie Map (1978) oder der Symbolics Document Examiner (1985), Systeme zur Unterstützung des Unterrichts (Intermedia [6], Hyperties) und Arbeitsumgebungen zum Sammeln und Auswerten von Text-Informationen (ZOG/KMS [7], Textnet, NoteCards [8]) sind nur einige Beispiele. Die meisten dieser Systeme wurden im Hinblick auf bestimmte Zielanwendungen konzipiert und bieten Möglichkeiten die über die allgemeine Hypertext-Funktionalität hinausreichen, beispielsweise zur Kommunikation, Software-Entwicklung oder Entscheidungsunterstützung. Seit 1986 Guide auf den Markt kam, sind allgemeine Hypertextsysteme kommerziell verfügbar. 1987 folgten HyperCard, 1989 ToolBook und die Anzahl der angebotenen Produkte wächst. 1987 und 1989 veranstaltete ACM (Association for Computing Machinery) jeweils eine Hypertext-Konferenz in den USA. 1990 fand die erste große europäische Konferenz in Paris statt. Somit ist Hypertext inzwischen ein etablierter Bereich sowohl in der EDV-Forschung als auch in der Anwendung [9].

Die Begriffe Hypertext und Hypermedia wurden in der frühen Geschichte des Feldes nicht eindeutig verwendet. Im Rahmen dieser Arbeit soll die von Conklin 1987 erarbeitete Definition übernommen werden, die auch den meisten späteren Arbeiten [10], zumindest implizit, zugrunde liegt.

Conklin definiert "...maschinen-unterstützte Kanten (sowohl innerhalb als auch zwischen Dokumenten) als die kennzeichnende Eigenschaft von Hypertext-Systemen..." [11]. **Nicht-linearität, Multimedialität und eine integrierende Benutzerschnittstelle sind Eigenschaften von Hypermedia, die sich aus dieser grundlegenden Idee der maschinengestützten Kanten ergeben.** Von Hypermedia spricht man, wenn außer Text-Dokumenten andere Medien wie Graphik, Animation, Video oder Ton verwendet werden. Da die Verwendung unterschiedlicher Medien keinen Einfluß auf die Grundidee der maschinengestützten Kanten hat, werden die Begriffe Hypertext und Hypermedia synonym gebraucht.

[5]: Dam, Address, 1988, S. 889.

[6]: Yankelovich u. a., Intermedia, 1988.

[7]: Akscyn/McCracken/Yoder, KMS, 1988.

[8]: Halasz, Notecards, 1988.

[9]: Die Konferenzen und Projektaktivitäten sind dokumentiert in: Smith u. a., Hypertext'87, 1987. ACM, Special, 1988. ACM, Hypertext'89, 1989. Gloor/Streitz, Hypertext, 1990. Rizk/Streitz/André, ECHT, 1990. Jonassen/Mandl, Designing Hypermedia, 1990. Maurer, Hypertext/Hypermedia'91, 1991.

[10]: z. Bsp.: Smith/Weiss, Hypertext, 1988, S. 816, Akscyn/McCracken/Yoder, KMS, 1988, S. 820.

[11]: "... machine-supported links (both within and between documents) as the essential feature of hypertext systems..." Conklin, Introduction, 1987, S. 18.

2.2 Hyper-Strukturen

Die Grundbausteine von hyperbasierten Anwendungen sind **Knoten**, welche in der Form eines semantischen Netzes miteinander verknüpft sind. Die **Kanten** (Hyperlinks) werden an der Benutzerschnittstelle durch **Anker** (direkt manipulierbare Objekte) repräsentiert. Der Anwender benötigt beim Navigieren durch das semantische Netz **keine Kenntnis über die physischen oder logischen Zugriffswege**.

Die nachfolgend beschriebenen, sehr unterschiedlichen Gestaltungen von Knoten, Kanten und Orientierungshilfen haben sich durch verschiedene Realisierungen hyperbasierter Anwendungen in der Praxis ergeben. Die vorgenommenen Einteilungen sind daher nicht immer überschneidungsfrei, die verwendeten Begriffe nicht immer exakt definiert [12]. Sie beschreiben den derzeitigen "state of the art".

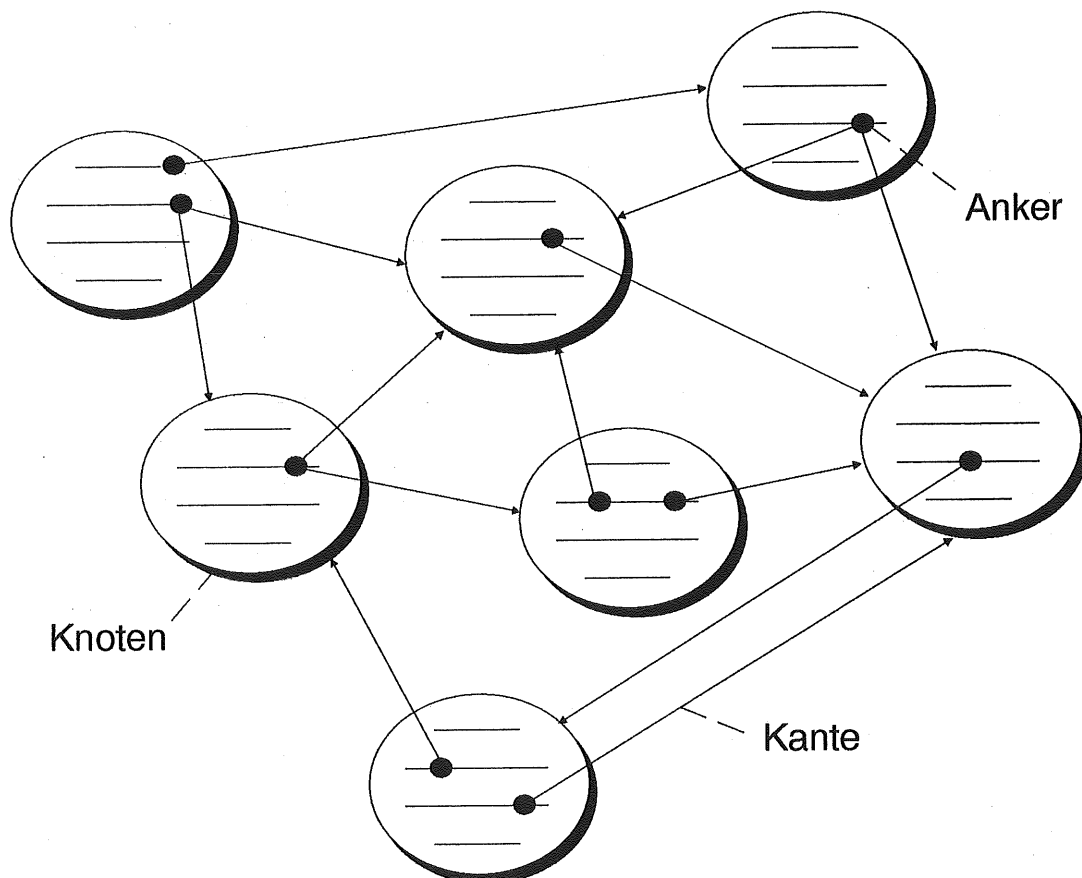


Abb. 1: Hyper-Struktur mit 7 Knoten, 12 Kanten und 10 Ankern

[12]: Zur Problemstellung vgl. Shneiderman/Kearsley, *Hypertext Hands-On*, 1989. Hoffmann, *Hypertextsystem*, 1991, S. 178 f. Mühlhäuser, *Hypermedia-Konzept*, 1991, S. 282 ff.

2.2.1 Knoten (nodes)

In hyperbasierten Anwendungen werden die Informationen durch Knoten (nodes) repräsentiert. Idealtypisch ist der Knoten die kleinste, für den Benutzer ansprechbare Informationseinheit. Entscheidend für die Bildung eines Knotens ist der **semantische Zusammenhang der Informationen**. Welche Informationen aufgrund ihrer inhaltlichen Bedeutung zusammen in einen Knoten gehören und welche Informationen auf verschiedene Knoten verteilt werden, hängt von der konkreten Anwendung ab.

In Analogie zu Datenbanken entspräche der Knoten einem Datensatz. Wichtiger Unterschied ist jedoch, daß der Knoten weder in Form, Größe oder Inhalt den für Datensätze üblichen Beschränkungen unterliegt. Die zur Darstellung der Informationen verwendeten Medien können Texte, Graphiken, etc. und Kombinationen daraus sein. Auch Größe und Erscheinungsbild des Knotens sind, in den Grenzen der technischen Machbarkeit und dessen was für den späteren Anwender sinnvoll ist, frei bestimmbar.

Prinzipiell fehlen den Knoten jegliche Strukturen. Dies erschwert dem Benutzer jedoch die Orientierung in komplexen Systemen. Gleichzeitig erschwert es die Bildung eines mentalen Modells zum Verständnis eines bestimmten Hypertext-Systems [13]. Um dem entgegen zu wirken unterteilen konkrete Systeme die **Knoten in Klassen** (typed nodes), verwenden innerhalb von Knoten bestimmte wiederkehrende Strukturelemente (structured nodes) oder fassen einzelne Knoten zu neuen Knoten (composite nodes) zusammen [14]. Beispielsweise klassifiziert (typed nodes) NoteCards in Informations-Karten (mit bis zu 50 anwendungsspezifischen Unterklassen), Browser-Karten die eine graphische Übersicht der Informations-Karten und ihrer Verknüpfungen enthalten, und FileBox-Karten, die die Informations-Karten zusammenfassen und in eine hierarchische Struktur einbetten [15]. Orientierungshilfen wie die Browser-Karten sind eine besonders wichtige Klasse von Knoten und werden daher am Ende dieses Kapitels gesondert beschrieben.

Insbesondere in hyperbasierten Anwendungen zur Verwaltung größerer Datenmengen erleichtern structured nodes mit vorgegebenen Datenfeldern und wiederkehrenden Bildelementen dem Benutzer die Orientierung und ein schnelles Erfassen der dargestellten Informationen. Gleichzeitig ermöglichen sie dem System, standardisierte Prozeduren zur Verarbeitung der Informationen anzuwenden.

[13]: De Young, *Linking*, 1990, S. 239.

[14]: Die Einteilung wurde von Conklin, *Introduction*, 1987, S. 36-37 übernommen, da sowohl der Begriff "structured node", als auch "typed node" in der späteren Literatur meist ohne nähere Definition auftauchen. Die Begriffe sind jedoch nicht überschneidungsfrei. Eine bestimmte Klasse von Knoten (typed) unterscheidet sich von anderen Klassen anhand bestimmter Strukturmerkmale (structured).

[15]: Halasz, *Notecards*, 1988, S. 837-838.

Composite nodes sind in Systemen mit einer großen Anzahl von Knoten sinnvoll. Das Zusammenfassen einzelner Knoten auf einer höheren Organisationsebene verschafft dem Benutzer einen besseren Überblick. Die in Notecards verwendeten FileBox-Karten sind ein Beispiel für composite nodes.

Eine Ansammlung einzelner Knoten ergibt jedoch noch keine hyperbasierte Anwendung. Die spezielle Charakteristik einer hyperbasierten Anwendung entsteht erst durch die Verknüpfung der einzelnen Knoten zu einem semantischen Netz.

2.2.2 Kanten (links)

"Kennzeichnende Eigenschaft von Hypertext ist die maschinelle Verfolgung von Verweisen [16]". Der Autor einer hyperbasierten Anwendung hat die Möglichkeit jeden Punkt innerhalb eines Knotens mit einem anderen Knoten zu verbinden [17]. Aktiviert der Benutzer den Ausgangspunkt einer Kante [18], so springt er in den "gerufenen" Knoten. Auch für die Bildung von Kanten ist allein **der semantische Zusammenhang der Informationen entscheidend**. Welche Knoten oder Punkte innerhalb von Knoten aufgrund ihrer inhaltlichen Bedeutung verknüpft werden, hängt von der konkreten Anwendung ab.

Alle **Kanten** haben eines gemeinsam: Sie haben immer zwei Enden, einen **Ausgangs- oder Ankerpunkt** (link anchor) und mindestens ein **Ziel** [19]. Allerdings bestehen anwendungsspezifische Unterschiede zwischen den Kanten in Bezug auf Richtung, Ausgangspunkten, Zielen oder prozeduralen Eigenschaften.

Es gibt **eindirektionale und zweidirektionale Kanten**. Beim eindirektionalen link erreicht der Benutzer das Ziel, ohne die Möglichkeit diese Kante in die andere Richtung zurückzuverfolgen. Vom Zielknoten aus ist nicht ersichtlich, daß hierzu Kanten aus anderen Knoten existieren. Der zweidirektionale link dagegen unterscheidet nicht zwischen Ziel- oder Ausgangspunkt. Die Kante ist von beiden Knoten aus sichtbar, sie hat also zwei "Ausgangspunkte", und läßt sich auch in beide Richtungen verfolgen.

Die computergestützte Kante zweier Knoten an sich ist auf dem Bildschirm nicht sichtbar. Das Vorhandensein einer Kante muß dem Benutzer durch Ankerpunkte angezeigt werden.

[16]: "The most distinguishing characteristic of hypertext is its machine support for the tracing of references."
Conklin, *Introduction*, 1987, S. 33.

[17]: Einige Systeme erlauben auch die Verknüpfung von Punkten innerhalb eines Knotens.

[18]: Der englische Fachausdruck "link" wird oftmals auch mit "Verbindung" oder "Verknüpfung" übersetzt.

[19]: Nielsen, *Hypertext*, 1990, S. 107.

Prinzipiell kann jedes Objekt auf dem Bildschirm Ausgangspunkt einer Kante sein. Sinnvoll sind allerdings nur Ankerpunkte die als solche leicht identifizierbar sind. Meist werden hierzu besonders gekennzeichnete Worte, spezielle Symbole (icons), hervorgehobene Teile von Abbildungen oder kontextabhängige Menues verwendet [20].

Weiteres wichtiges Kriterium für die Wahl der Ankerpunkte ist ihre inhaltliche Bedeutung. Das heißt, der Benutzer sollte bereits durch die graphische Gestaltung oder die Beschriftung des Ankers Informationen zum Zielpunkt erhalten. Dies verleiht den Ankern eine Orientierungsfunktion [21].

Analog zu den typed nodes werden in vielen Anwendungen die Kanten klassifiziert (typed links) [22] und die einzelnen Klassen durch unterschiedlich gestaltete Ausgangspunkte gekennzeichnet. Ein interessantes Beispiel hierfür ist Guide, das die Kanten in Ersetzen, Pop-Ups und Sprünge [23] unterscheidet. Erkennbar sind diese unterschiedlichen Kanten-Klassen durch entsprechende Veränderungen des Cursors sobald dieser einen Ausgangspunkt berührt. Ziele können entweder ganze Knoten oder bei einigen Systemen auch Punkte innerhalb eines Knotens sein [24]. Letzteres ist bei Knoten sinnvoll die, aufgrund ihrer Größe, auf dem Bildschirm nicht vollständig darstellbar sind.

Im einfachsten Fall hat ein Ausgangspunkt einen Zielpunkt. Möglich sind aber auch mehrere Zielpunkte die zu einem Ausgangspunkt gehören. Wählt der Benutzer einen solchen Ausgangspunkt, so werden entweder mehrere Knoten gerufen oder aber das System wählt in Abhängigkeit von einem Benutzermodell den Zielknoten aus [25]. Im weitesten Sinne zählen auch Menues zu den Ausgangspunkten mit mehreren Zielen.

Fortgeschrittene Hyper-shells [26] verfügen über eigene Programmiersprachen mit deren Hilfe den Kanten Prozeduren zugewiesen werden können. Sobald der Benutzer eine derartige Kante auswählt, führt das System die Prozedur aus. Ein Beispiel hierfür ist die oben erwähnte automatische Auswahl aus mehreren Zielknoten. Prozedurale Kanten sind grundlegende Voraussetzung für fortgeschrittene interaktive Anwendungen wie Entscheidungsunterstützungssysteme [27].

[20]: Vgl. die Beschreibung von Hyperties in: Marchioni/Shneiderman, *Finding*, 1988, S. 74.

[21]: Vgl. weiter unten die Beschreibung der Orientierungs- und Navigationshilfen

[22]: De Young, *Linking*, 1990, S. 240.

[23]: " ... Replacements, pop-ups, and jumps." Nielsen, *Hypertext*, 1990, S. 91.

[24]: Yankelovich, *Intermedia*, 1988, S. 83. *Intermedia* erlaubt zweidirektionale Kanten zwischen zwei beliebig großen Punkten, sogenannten Blöcken.

[25]: Nielsen, *Hypertext*, 1990, S. 109.

[26]: Beispielsweise *Guide*, *Toolbook*, *Hypercard*, *KnowledgePro*; Vgl. Irler/Barbieri, *Anchors*, 1990, S. 265.

[27]: Vgl. beispielsweise die Beschreibung von *SPRINT* (Strategic Plan and Resource INTEgration) in: Carlson/Ram, *HyperIntelligence*, 1990.

2.2.3 Orientierungs- und Navigationshilfen

Orientierungs- und Navigationshilfen sind keine originären Elemente von hyperbasierten Anwendungen sondern besondere Formen von Informationseinheiten oder Kanten. Sie beinhalten **Informationen über Knoten und Kanten**. Da ihnen in der praktischen Anwendung eine herausragende Bedeutung zukommt, sollen sie hier gesondert beschrieben werden.

"Gleichzeitig mit der Möglichkeit die Informationen komplexer zu organisieren ergibt sich das Problem zu wissen (1) wo man sich im Netzwerk befindet und (2) wie man an eine andere Stelle gelangt, von der man weiß (oder vermutet), daß sie im Netzwerk existiert" [28]. Der Anwender benötigt also Hilfen (1) zur Orientierung und (2) zum Navigieren.

Einen ersten Einstieg in ein Netzwerk können geführte **Touren oder Pfade** geben. Durch entsprechend gekennzeichnete Anker schlägt der Autor der Anwendung dem Benutzer eine sinnvolle Reihenfolge wichtiger Knoten vor. Für verschiedene Benutzerinteressen können unterschiedliche Pfade vorgegeben werden. Eine weitere Navigationshilfe ist die Backtrack-Funktion [29]. Egal über welche Kanten der Benutzer sich zum aktuellen Knoten vorgearbeitet hat, die Backtrack-Funktion bringt ihn zu den vorher besuchten Knoten in umgekehrter Reihenfolge zurück. Da sie wichtig für den Eindruck der Beherrschbarkeit eines Systems ist, sollte sie jederzeit und immer auf die gleiche Weise aktivierbar sein [30].

Den gleichen Zweck erfüllen **History- (Geschichts-) Funktionen**. Bei History-Funktionen werden die vorhergehenden Knoten jedoch nicht Schritt für Schritt zurück verfolgt, sondern der Benutzer erhält eine Liste aus der er sich den interessierenden, bereits besuchten Knoten direkt aussuchen kann. Zusätzlich kann eine automatisch aufgezeichnete "Benutzer-Geschichte" zur Dokumentation und Auswertung verwendet werden.

Sowohl die oben beschriebenen, unterschiedlich gestalteten Knoten und Kanten, als auch geführte Touren und Backtrack-Funktionen geben dem Benutzer begrenzte, lokale Orientierungs- und Navigationsmöglichkeiten. Sie informieren ihn über den aktuellen und die nächstliegenden Knoten, geben jedoch keinen Überblick über den Gesamtzusammenhang des Netzes. Im komplexen "hyperspace" benötigt der Benutzer zusätzlich globale Orientierungshilfen. Eine solche Orientierungshilfe ist der **graphische Browser** [31]. Der

[28]: "Along with the power to organize information much more complexly comes the problem of having to know (1) where you are in the network and (2) how to get to some other place that you know (or think) exists in the network." Conklin, *Introduction*, 1987, S. 38.

[29]: Ein Fachausdruck der sich mit Rückverfolgung nur unzureichend übersetzen läßt.

[30]: Nielsen, *Hypertext*, 1990, S. 129.

[31]: wörtlich übersetzt: Blätterer.

Browser nutzt die beim Menschen hochentwickelte Fähigkeit zur Verarbeitung optisch/räumlicher Informationen [32]. Er bildet die Knoten und ihre Kanten im zwei- oder dreidimensionalen Raum symbolisch ab. Durch unterschiedliche Beschriftungen, Farben, Formen, Größen, oder Muster der einzelnen Objekte erhält der Benutzer visuelle Anhaltspunkte und kann sich wie mit einer Landkarte orientieren. Oftmals kann er auch durch Anklicken eines Symbols direkt in den entsprechenden Knoten springen.

Größere Netze lassen sich jedoch aufgrund der beschränkten Größe und Auflösungsfähigkeit von Bildschirmen nicht vollständig darstellen. Ein Lösungsansatz sind scrollende Fenster. Manche Browser bieten auch mehrere Diagramme mit unterschiedlichem Detaillierungsgrad, so daß der Benutzer zwischen Übersichtsdiagramm und einzelnen Regional-Karten wählen kann. Dadurch geht allerdings die geforderte globale Orientierung teilweise wieder verloren. Eine weitere Lösung ist die Fischaugen-Perspektive. Hierbei werden, ausgehend von der aktuellen Position des Anwenders, die naheliegenden Knoten detailliert, die entfernteren Netz-Regionen weniger detailliert dargestellt. Dadurch ergibt sich in besonderem Maße, wie insgesamt bei der Gestaltung eines Browsers, das Problem die semantische "Nähe" der einzelnen Knoten zueinander zu messen und optisch umzusetzen [33].

[32]: Conklin, *Introduction*, 1987, S. 38-39.

[33]: Ein Vorschlag "Affinität" zu messen findet sich in: Pintado/Tsichritzis, *SaTellite*, 1990.

3. Gestaltungsaspekte hyperbasierter Informationssysteme

Das Hyper-Konzept integriert nicht nur unterschiedliche Medien, sondern auch sehr verschiedene Anwendungsfelder unter einer Oberfläche. "... das Wesentliche an Hypertext ist genaugenommen die Kreuzung, die traditionelle Grenzen überwindet [34]."

Hypertext läßt sich als Datenbankmethode mit Zugriff über den semantischen Kontext beschreiben. Gleichzeitig bieten hyperbasierte Anwendungen eine Möglichkeit informelle Informationen mit formalisierten, computergestützten Prozeduren zu verbinden. Letztendlich drückt sich das Hyper-Konzept auch in einer Benutzerschnittstelle aus, die problemlos an unterschiedliche Benutzergruppen angepaßt werden kann.

3.1 Nichtlinearität und informelle Informationsstrukturen

Das Grundkonzept der semantischen Kanten zwischen Informationseinheiten führt zu nichtlinearen Informationsnetzen. Im Gegensatz dazu sind die meisten traditionellen Texte sowohl logisch also auch physisch linear strukturiert. Sie bauen inhaltlich von Anfang bis Ende aufeinander auf und werden auch in dieser einen Richtung, also linear gelesen. Nachschlagewerke, die einer gezielten Informationssuche dienen, bilden eine Ausnahme. Logisch bestehen sie aus einzelnen unabhängigen Einträgen (analog den Knoten) die durch Querverweise miteinander verbunden sind. In dieser vernetzten Weise werden sie auch verwendet, niemand liest ein Lexikon von vorn nach hinten. Physisch jedoch stehen die Einträge in alphabetischer Reihenfolge hintereinander. Das heißt sie sind anhand eines formalen, vom Inhalt unabhängigen, Kriteriums linear angeordnet.

Hyperbasierte Anwendungen bieten nun die Möglichkeit die Divergenz zwischen **logischer und physischer Struktur von Informationen zu überwinden**. Die Querverweise, der vielfältige semantische Zusammenhang einzelner Informationen, kann vom Benutzer maschinengestützt direkt nachvollzogen werden. Das bedeutet Informationen können entsprechend ihrer inhaltlichen Struktur, die nur selten linear ist, gespeichert und abgerufen werden. Dies bietet drei Vorteile:

- Erstens eignet sich das Hyper-Konzept in besonderem Maße zur Verarbeitung von Informationen, die nur **geringe oder keine formalen Strukturen** aufweisen anhand derer

[34]: "..., the essence of hypertext is precisely that it is a hybrid that cuts across traditional boundaries." Conklin, Introduction, 1987, S. 33.

sie geordnet und wiedergefunden werden könnten [35]. Dies trifft vor allem auf komplexe Sachverhalte zu wie sie beispielsweise im Bereich der Entscheidungsunterstützung anfallen.

- Zweitens hat die assoziative Informationsverarbeitung ihre Analogie im menschlichen Gedächtnis [36]. Die **informellen, kontextabhängigen Speicher- und Abrufmechanismen** sind daher intuitiv erfaßbar und machen hyperbasierte Anwendungen für Endanwender mit geringen Computer-Kenntnissen schnell nutzbar. Aber auch geübte Anwender profitieren von der Möglichkeit informelles Wissen über semantische Zusammenhänge schnell aufzufinden.
- Drittens läßt sich ein semantisches Netz jederzeit erweitern und anpassen ohne die Gefahr logische Strukturen zu verfälschen, wie dies zum Beispiel bei hierarchisch gegliederten Informationen der Fall sein kann. Vorhandene Wissensnetze können so **individuellen Benutzeranforderungen und unterschiedlichen mentalen Modellen** angepaßt oder um neue Erkenntnisse erweitert werden.

Die freie Verknüpfbarkeit birgt auch Gefahren. Der Autor einer hyperbasierten Anwendung ist nicht gezwungen systematisch und strukturiert vorzugehen. Analog der Gefahr bei den Sprungbefehlen (go to) in prozeduralen Programmiersprachen kann leicht ein nicht nachvollziehbarer "Spaghetti-Code" entstehen [37]. Dadurch ergibt sich das Risiko von Fehlinterpretationen durch den Benutzer, der aus den vom Autor vorgegebenen Kanten möglicherweise nicht beabsichtigte Schlußfolgerungen über deren inhaltliche Bedeutung zieht. Für den Endanwender besteht vor allem in großen Netzen die Gefahr des "Getting `lost in space" [38].

3.2 Multimedialität

Hyper-Anwendungen verbinden Informationseinheiten unabhängig vom verwendeten Medium. Fortgeschrittene Hyper-tools verarbeiten außer Texten auch Graphiken, Fotos, Animation, Video, Sprache und Ton [39]. Unterschiedliche Medien zur Informationsdarstellung sind vor allem in den Bereichen wichtig, in denen vielfältiges und komplexes Wissen verarbeitet wird. Dadurch können die Informationen so dargestellt werden, wie es ihrem jeweiligen Inhalt angemessen ist. Beispielsweise lassen sich das Verpackungsdesign

[35]: Marchioni/Shneiderman, *Finding*, 1988, S. 71.

[36]: Lucarella, *Information Retrieval*, 1990, S. 83.

[37]: De Young, *Linking*, 1990, S. 238.

[38]: annähernd übersetzbar mit: "Verlorengehen im Raum.", Conklin, *Introduction*, 1987, S. 38.

[39]: Ogawa/Harada/Kaneko, *Scenario-based*, 1990.

eines Produktes oder die Karosserieform eines Autos verbal nur unzureichend beschreiben. Hier sind ein Foto beziehungsweise eine Konstruktionszeichnung die angemessenen Medien. Gleichzeitig können inhaltlich zusammenhängende oder ergänzende Informationen die nur mit unterschiedlichen Medien darstellbar sind, in hyperbasierten Anwendungen unter einer einheitlichen Oberfläche miteinander verbunden werden.

Beispiel hierfür ist das von Ogawa [40] entwickelte, interaktive Englisch-Lernprogramm. Die einzelnen Lektionen werden auf Video in Form gespielter Szenen dargestellt. Parallel dazu kann sich der Benutzer den englischen Text der gesprochenen Sätze oder die japanische Übersetzung anzeigen lassen. Ogawa weist allerdings auch auf die noch nicht vollständig geklärten Probleme der Verknüpfung statischer (Texte, Abbildungen) und dynamischer Medien (Video, Ton) hin.

3.3 Mensch-Computer Schnittstelle

Bei der Gestaltung der Benutzerschnittstelle sind all jene Aspekte zu berücksichtigen, "... mit denen Benutzer konzeptuell (begrifflich, internes Wissen), perzeptuell (durch Sehen, Hören) und physikalisch in Berührung kommen" [41].

Die **konzeptuelle Ebene** beschreibt die Vorstellungen, die der Benutzer "im Kopf" (mentales Modell) vom System hat oder entwickeln soll. Mentale Modelle erlauben dem Benutzer, das Verhalten des Systems vorherzusagen, zu erklären und entsprechende Ziele und Aufgaben zu planen. Bei Hypermediasystemen besteht das konzeptuelle Modell lediglich aus den zwei Grundkomponenten Knoten und Verbindungen. Dieses revolutionär einfache, konzeptuelle Modell ist das hypermediaspezifische Kennzeichen und sein wesentlicher Vorteil. Es ermöglicht dem Benutzer die Bildung eines einfachen mentalen Modells und damit ein intuitives Verständnis des Systems.

Zur Gestaltung eines Informationssystems auf der **perzeptuellen Ebene** stehen verschiedene Interaktionstechniken wie Kommandosprachen, Menutechniken und die direkte Manipulation zur Auswahl [42]. Bei der direkten Manipulation wird eine Modell-Handlungswelt graphisch auf dem Bildschirm repräsentiert. Die Interaktion erfolgt über eine Simulation physischer Handlungen des Benutzers an den Objektrepräsentationen [43].

Die **physikalische Ebene** beschreibt die Ein- und Ausgabeaktionen, die zur Manipulation der Informationsobjekte und zum Informationsaustausch zwischen Benutzer und System

[40]: Ogawa/Harada/Kaneko, *Scenario-based*, 1990.

[41]: Tauber, *Mentale Modelle*, 1985, S. 49.

[42]: *Zum Befund empirischer Untersuchungen* vgl. Frese/Schulte-Göcking/Altman, *Lernprozesse*, 1987.

[43]: Vgl. Shneiderman, *User Interface*, 1987, S. 180.

notwendig sind [44].

Die meisten hyperbasierten Anwendungen verwenden **graphische Repräsentationen einer Modell-Handlungswelt** und damit die direkte Manipulation als wesentliche Interaktionsform. Dies ist aufgrund des Hyper-Konzeptes der Knoten und Verbindungen naheliegend, aber nicht zwingend. Bei einigen Aufgabenstellungen sind andere Interaktionsformen angemessener. Hyperbasierte Anwendungen und verschiedenartige, der jeweiligen Aufgabenstellung angemessene Interaktionstechniken sind kein Gegensatz. So können problemlos Menutechniken und Dateneingabemasken ergänzend zur direkten Manipulation implementiert werden, ohne das zugrundeliegende Konzept der Knoten und Verbindungen zu verändern. Damit lassen sich alle notwendigen Mensch-Computer Interaktionen effizient bewerkstelligen, also die Eingabe, Auswertung und Ausgabe von Informationen. Zusätzlich zu den in konventionellen Informationssystemen üblichen Techniken, bieten hyperbasierte Anwendungen die Vorteile des leicht verständlichen Konzeptes von Knoten und Verbindungen und einer graphischen Benutzeroberfläche. Dies hat vielfältige Auswirkungen:

- Der Informationszugriff erfolgt hauptsächlich **navigierend**. Der Benutzer kann ohne Kenntnis der zugrundeliegenden Datenstrukturen zu den gewünschten Informationen gelangen.
- Beim navigierenden Zugriff erhält der Benutzer durch die vielfältigen Gestaltungsmöglichkeiten einer graphischen bzw. multimedialen Oberfläche **optische Anhaltspunkte**.
- Die durch die begrenzte Bildschirmgröße bedingte Zerteilung der Informationen in mehrere Einheiten wirkt weniger störend als in konventionellen Systemen. Durch die direkten **semantischen Verbindungen** kann sie vom Benutzer jederzeit und ohne Kenntnisse des formalen Systemaufbaus (Problem des "cognitive overhead" [45]) überwunden werden.
- Funktionen können über **graphische Elemente der Oberfläche ausgelöst** werden. Dies entbindet den Benutzer von der Notwendigkeit, komplexe formale Sprachen zu erlernen oder sich in verschachtelten Menue-Strukturen zurechtzufinden.
- Informationen können graphisch aufbereitet werden. Vor allem bei **Auswertungsergebnissen**, also stark verdichteten Daten, verschafft dies dem Benutzer einen schnellen Überblick.

[44]: Zu den allgemeinen Forderungen der Soft- und Hardwareergonomie vgl. Fähnrich, *Software-Ergonomie*, 1987. DIN, DIN 66234 Teil 8, 1988. Balzert, *Software-Ergonomie*, 1988.

[45]: *Cognitive overhead* bezeichnet die zusätzliche kognitive Belastung des Benutzers, die durch die Notwendigkeit entsteht das Medium Computer allgemein bzw. eine konkrete Anwendung zu beherrschen.

3.4 Integrierende Aspekte

Das Hyper-Konzept integriert auf konzeptioneller Ebene die Informationen und Funktionen zu einzelnen, inhaltlich zusammengehörenden Einheiten, den Knoten. Jeder Knoten ist somit ein **eigenständiges Objekt**, das über die Verbindungen mit anderen Knoten kommuniziert. In dieses Netz lassen sich aber nicht nur die originären Hyper-Knoten einbinden, sondern genauso gut auch andere, externe Programm-Module. Das externe Programm-Modul besteht ebenfalls aus Daten und Funktionen und ist somit auch ein "Knoten" mit dem über Verbindungen kommuniziert werden kann.

Die Ausstattung der Knoten und Kanten mit **prozeduralen Eigenschaften** erweitert Hypertext um die Anwendungsmöglichkeiten objektorientierter Programmiersprachen. Dadurch können zusätzlich zur Datenspeicherung auch Funktionen zur Informationsanalyse und Entscheidungsunterstützung implementiert werden. Das Hyper-Konzept ermöglicht also die Integration der unterschiedlichen Module, Anwendungen oder technisch verschiedener Implementierungen unter einer einheitlichen Konzeption, der der Knoten und Verbindungen. Gleichzeitig bietet es dem Benutzer den Zugang zu diesen Modulen unter einer gemeinsamen Benutzerschnittstelle.

Für den Benutzer ergibt sich damit der Vorteil, daß sowohl die Datenspeicherung als auch die darauf anwendbaren unterschiedlichsten Funktionen und die Ablaufsteuerung unter einer **einheitlichen, konsistenten Oberfläche** integriert sind. Nachteilig ist, daß der Programmierer nicht vom System gezwungen wird Daten, Funktionen und Ablaufsteuerung systematisch zu trennen. Es besteht die Gefahr der unstrukturierten, redundanten Programmierung.

Zusammenfassend läßt sich sagen, daß das zugrundeliegende Prinzip der Verbindung einzelner, unabhängiger Informationsobjekte über prozedurale links Hyper-tools zu einem sehr flexiblen Werkzeug machen.

4. Umsetzungsaspekte hyperbasierter Informationssysteme

Die Einsatzschwerpunkte hyperbasierter betrieblicher Anwendungen sind:

- Aufgabenbezogene Navigationssysteme
- Interaktive Tutorial- und Trainingssysteme
- Integrierte Dokumentationssysteme

Mit dieser Einteilung wird jedoch der Eindruck erweckt, daß mit dem jeweiligen Einsatzgebiet auch inhaltlich differierende Anwendungsbereiche verbunden sind. Folgt man jedoch konsequent der Zielvorgabe einer durchgängigen **Entwicklungs- und Einführungsstrategie**, so sollte am Ende eines derartigen Weges ein Informationssystem stehen, welches über **alle drei Komponenten** verfügt. Nachfolgend werden zum jeweiligen Einsatzschwerpunkt die Umsetzungsmöglichkeiten anhand von am Institut für Wirtschaftsinformatik (IWi) entwickelten Prototypen diskutiert.

4.1 Aufgabenbezogene Navigationssysteme

Mit aufgabenbezogenen Navigationssystemen wird die Zielsetzung verfolgt, dem Anwender die Datenbasis und den Funktionsumfang eines Informationssystems zur Bewältigung einer **fest umrissenen Aufgabenstellung** zugänglich zu machen. Nachfolgend soll hierbei auf den Aspekt der Informationsanalyse und Entscheidungsunterstützung näher eingegangen werden.

Management ist das Treffen von Entscheidungen. "Die Qualität von Management-Entscheidungen ist von den, dem Entscheidungsträger zur Verfügung stehenden Informationen, direkt abhängig" [46]. Daher ist es wichtig, welche Daten und in welcher Form diese Daten von einem Informationssystem bereitgestellt werden.

Nachdem die Daten bereitgestellt sind, müssen sie im Rahmen eines Entscheidungsprozesses analysiert werden. Dazu werden die Daten mit Hilfe von Methoden und Modellen weiterverarbeitet. Oftmals geschieht dies unsystematisch, ohne daß sich der Entscheidungsträger darüber bewußt ist. Mockler berichtet, daß Manager meist nicht in der Lage sind, die einzelnen Schritte einer Entscheidungsfindung konkret zu verbalisieren [47].

In der Praxis zeigt sich auch, daß viele der theoretischen Möglichkeiten der

[46]: "The quality of managerial decisions is related directly to the information available to the decision maker." Burch/Strater/Grudnitski, *Information Systems*, 1979, S. 49.

[47]: Mockler, *Information Systems*, 1974, S. 4-6.

Entscheidungsunterstützung kaum realisiert werden. Dies ist zum einen auf fehlende Kenntnisse der Verfahren, zum anderen jedoch auch auf die für EDV-Laien umständliche Handhabung konventioneller Daten-, Methoden- und Modellbanken zurückzuführen. So berichtet Heinzelbecker "... daß sich Manager häufig bereits beim Abruf vorprogrammierter Datenoutputs überfordert fühlen" [48].

Erforderlich sind daher benutzerfreundliche Managementinformationssysteme die den Anwender bei der Informationsanalyse **aktiv** unterstützen. Hierfür bietet Hypermedia neue Gestaltungsansätze. Mit dem Prototypen "HyperMAIS" wurde ein aufgabenbezogenes Navigationssystem für den Anwendungsbereich Marketing/Vertrieb entworfen. Hierbei stand nicht ein großer Funktionsumfang, sondern die konsequente Umsetzung des Hyper-Konzeptes für einen kleinen Aufgabenausschnitt im Vordergrund. Abb. 2 gibt einen zusammenfassenden Überblick über die Knotenstruktur des Prototypen.

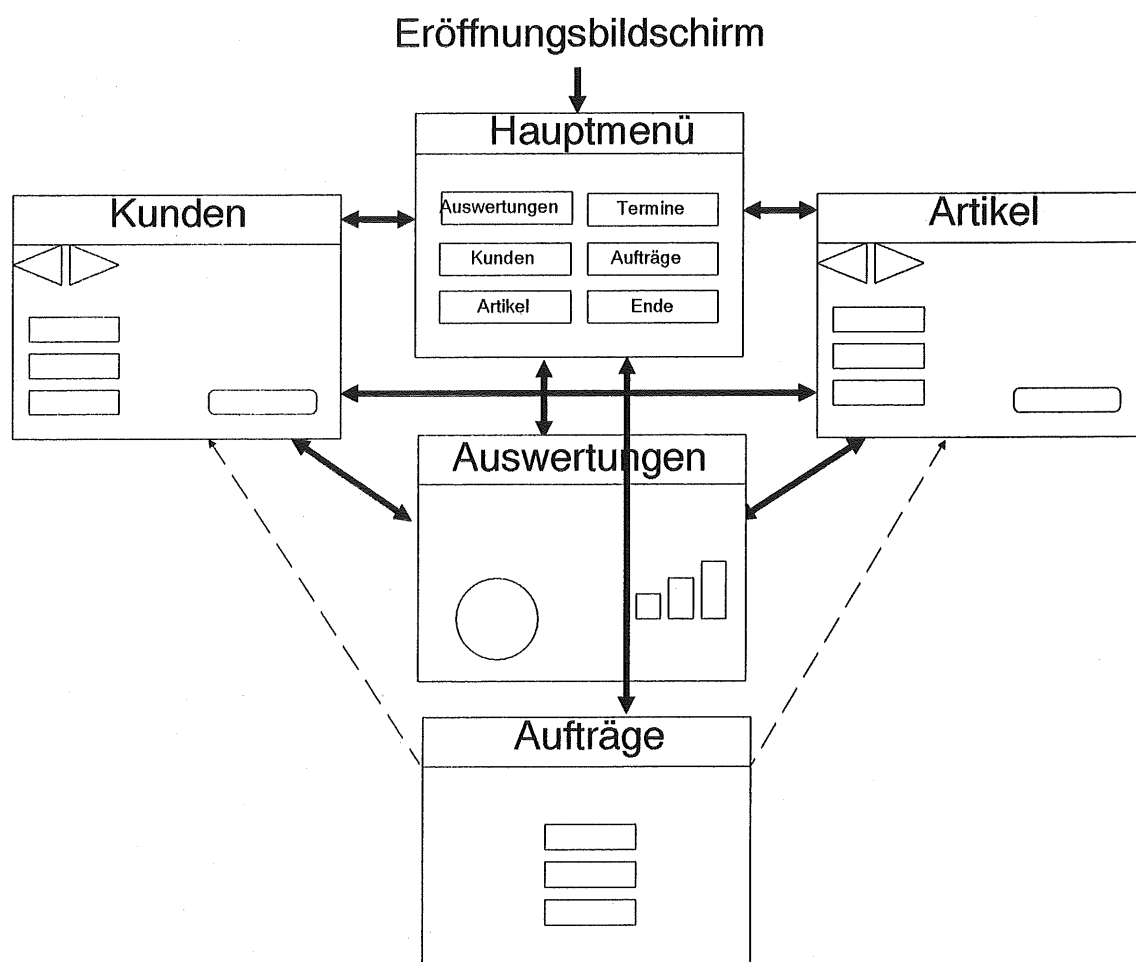


Abb. 2: Knotenstruktur des Prototypen "HyperMAIS"

Am Beispiel des Knotens "Auswertungen" können die hypermediaspezifischen Verknüpfungen besonders anschaulich dargestellt werden (vgl. Abb. 3). Innerhalb dieses Knotens stehen dem Benutzer zwei standardisierte Auswertungsfunktionen in der Form einer regionalen Umsatzauswertung nach Nielsen-Gebieten und eine Auswertung nach Artikelgruppen zur Verfügung.

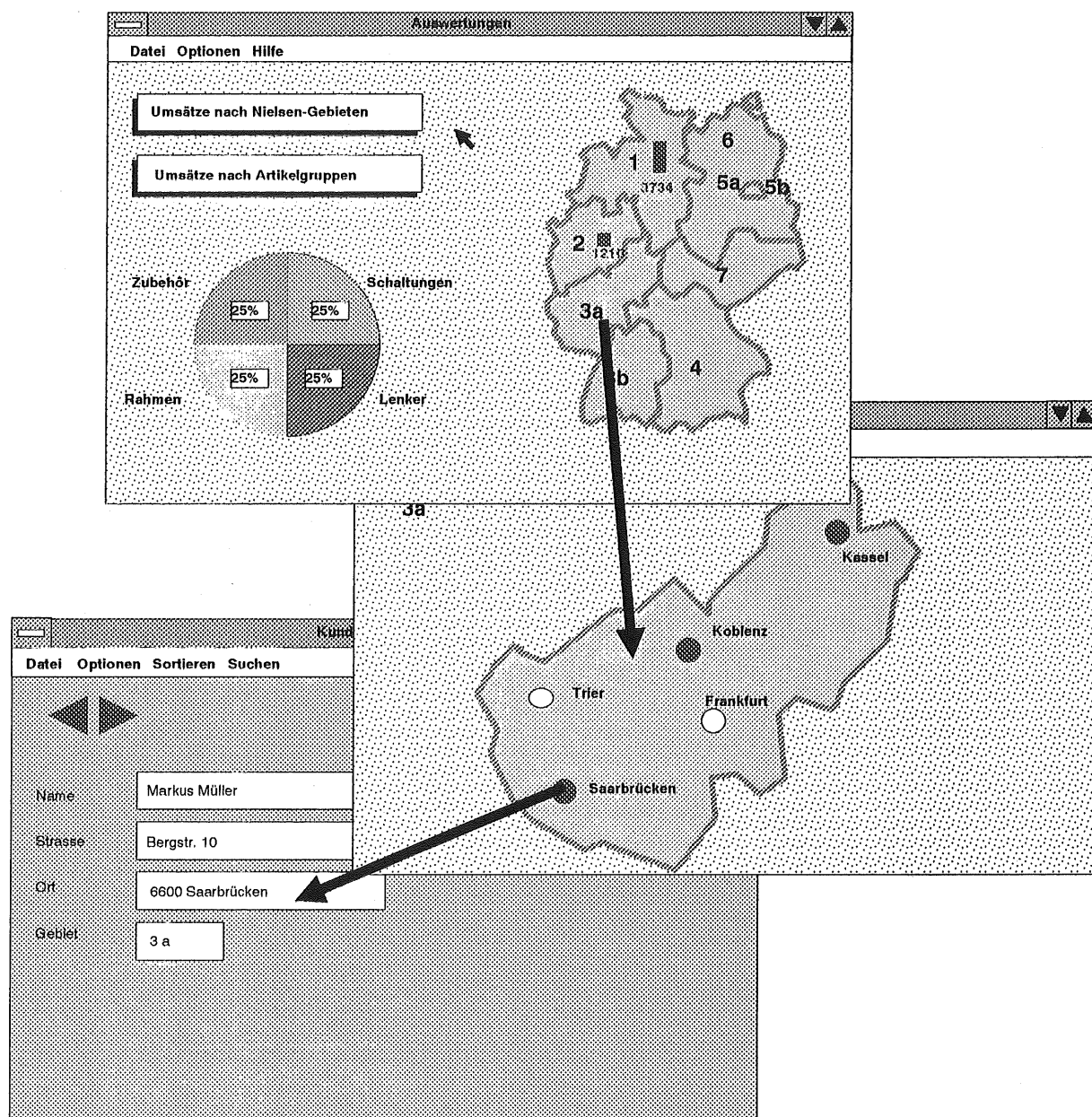


Abb. 3: Benutzerschnittstelle des Knotens "Auswertungen"

Sowohl die Tortenstücke als auch die einzelnen Nielsen-Gebiete sind Ankerpunkte von Verbindungen zu den entsprechenden Teilen der Artikel- bzw. Kundendatei. Möchte der Benutzer, z. B. auf Grund der Auswertungsergebnisse, die unverdichteten Daten von Teilbereichen sehen, so kann er er über einen Anker direkt in die entsprechende Datei springen. Das "Anklicken" des Tortenstücks "Zubehör" bewirkt einen Sprung in den Teil der Artikeldatei, der alle Artikel der Artikelgruppe Zubehör enthält. Analog bewirkt das Auswählen eines Nielsen-Gebietes den Sprung zu der dazugehörigen Gebietskarte der Kundendatei.

Hieraus kann der Benutzer dann eine der markierten Städte auswählen und gelangt zum eigentlichen Knoten "Kunden" mit den zugehörigen Datenbankmasken. Im Unterschied zu den klassischen Datenbankanfragesprachen erfolgt diese Zugriffsweise nach dem Prinzip der direkten Manipulation.

Wie auch in konventionellen Systemen, lassen sich also in einem datenbankgestützten hyperbasierten Managementinformationssystem strukturierbare Daten speichern, analysieren und zur Entscheidungsunterstützung verwenden. Zusätzlich können in einem hypermediabasierten Informationssystem jedoch auch nicht strukturier- und formalisierbare Daten gespeichert werden. Allerdings ergibt sich bei den nicht-strukturierten Informationen bezüglich der Informationsanalyse und Entscheidungsfindung ein prinzipielles Problem. Sie können kaum maschinengestützt erfolgen. Der Grund dafür ist die fehlende formale Struktur. So lassen sich zwar einzelne Informationseinheiten eines hypermedialen Netzes anhand ihrer Semantik von einem menschlichen Autor, sozusagen "von Hand", verknüpfen, das formale Programm jedoch kann, solange es sich nicht um ein Expertensystem handelt, keine semantischen Inhalte verarbeiten [49].

Die eigentliche Analyse der nicht-formalisierten Informationen muß vom Entscheidungsträger vorgenommen werden. Das hypermediabasierte Managementinformationssystem unterstützt ihn dabei als gemeinsames Speichermedium der formalisierten und nicht-formalisierten Informationen und durch die Möglichkeit diese semantisch (wenn auch von Hand) in Beziehung zu setzen.

Hyperbasierte Ansätze bieten "durch ihre Integrationsfähigkeit damit den Ausgangspunkt, für den Manager weit intelligentere, individuelle [!] und flexible "front ends" zur bisher existierenden Welt der Datenbanken und netzverfügbaren Großapplikationen in betrieblichen Informations- und Kommunikationssystemen zu gestalten [50]".

[49]: Daraus ergibt sich auch die Unmöglichkeit Hypertext-Verbindungen automatisch zu erzeugen die z. B. in Bernstein, *Apprentice*, 1990 beschrieben wird. Sein Lösungsvorschlag beruht auf einem automatischen String-Vergleich unter Aufsicht eines menschlichen Experten.

[50]: Nastansky/Seidensticker, *Hypermedia-basiertes Informationsmanagement*, 1990, S. 520 u. 521.

4.2 Interaktive Tutorial- und Trainingssysteme

Bereits in den 60er Jahren gab es erste Ansätze zur Entwicklung computergestützter Lernsysteme. Diese Ansätze scheiterten jedoch seinerzeit, da die notwendigen Hard- und Softwarevoraussetzungen wie dezentrale Arbeitsplatzrechner und graphische Benutzerschnittstellen fehlten. Seit den 70er Jahren bemühte man sich um Ansätze, welche unter dem Schlagwort "Computer Based Training (CBT)" diskutiert werden. Hierbei steht ein sequentieller Ablauf der Wissensvermittlung im Vordergrund.

Zu Beginn der 90er Jahre bietet nun das Hyper-Konzept die Möglichkeit, wesentlich freiere Konzepte für das computergestützte Lehren und Lernen zu entwickeln, die zu einer aktiven Einbindung des Nutzers und zu einem individuellen Lehr- bzw. Lernstil führen können.

Nachfolgend werden die Umsetzungsmöglichkeiten eines hyperbasierten Tutorials am Beispiel des Prototypen "WITEC-Tutorial" beschrieben. Dieser Prototyp wurde im Rahmen eines vom BMBW geförderten Forschungsvorhabens gemeinsam mit dem Institut der Gesellschaft zur Förderung der Angewandten Informationsforschung e. V. (IAI) an der Universität des Saarlandes und der Fakultät für Wirtschaftswissenschaften der Universität Leipzig, Lehrstuhl für Wirtschaftsinformatik, entwickelt. Zielsetzung dieses Projektes war die Überführung von in Buchform vorliegenden Lehrinhalten in ein hyperbasiertes Tutorial.

Das WITEC-Tutorial setzt sich aus den Komponenten Wissensbasis und Lernsystem zusammen. Die Knotenstruktur des WITEC-Tutorials ist in Abb. 4 widergegeben. Nachfolgend werden relevante Aspekte beim Entwurf des WITEC-Tutorials erläutert.

Der Aufbau der Wissensbasis eines hyperbasierten Tutorials erfordert die Umsetzung der Lehrinhalte in Hyper-Strukturen [51]. Diese Umsetzung kann in folgenden Schritten erfolgen:

1. Zerlegung des Wissensbereiches in Informationseinheiten
2. Verknüpfung der Informationseinheiten
3. Schaffung von externen Zugriffspfaden

Bei der Aufbereitung der Lehrinhalte hat sich eine Zerlegung in die Knotentypen "Basis-Informationen" und "Erläuterungen" bewährt. Hierzu werden die Erläuterungen aus den Lehrinhalten extrahiert und die verbleibenden Basis-Informationen linear in Knoten aufgespalten.

[51]: zu dieser Problematik vgl. Jonassen, *Designing structured Hypertext*, 1988. Jonassen/Mandl, *Designing Hypermedia*, 1990.

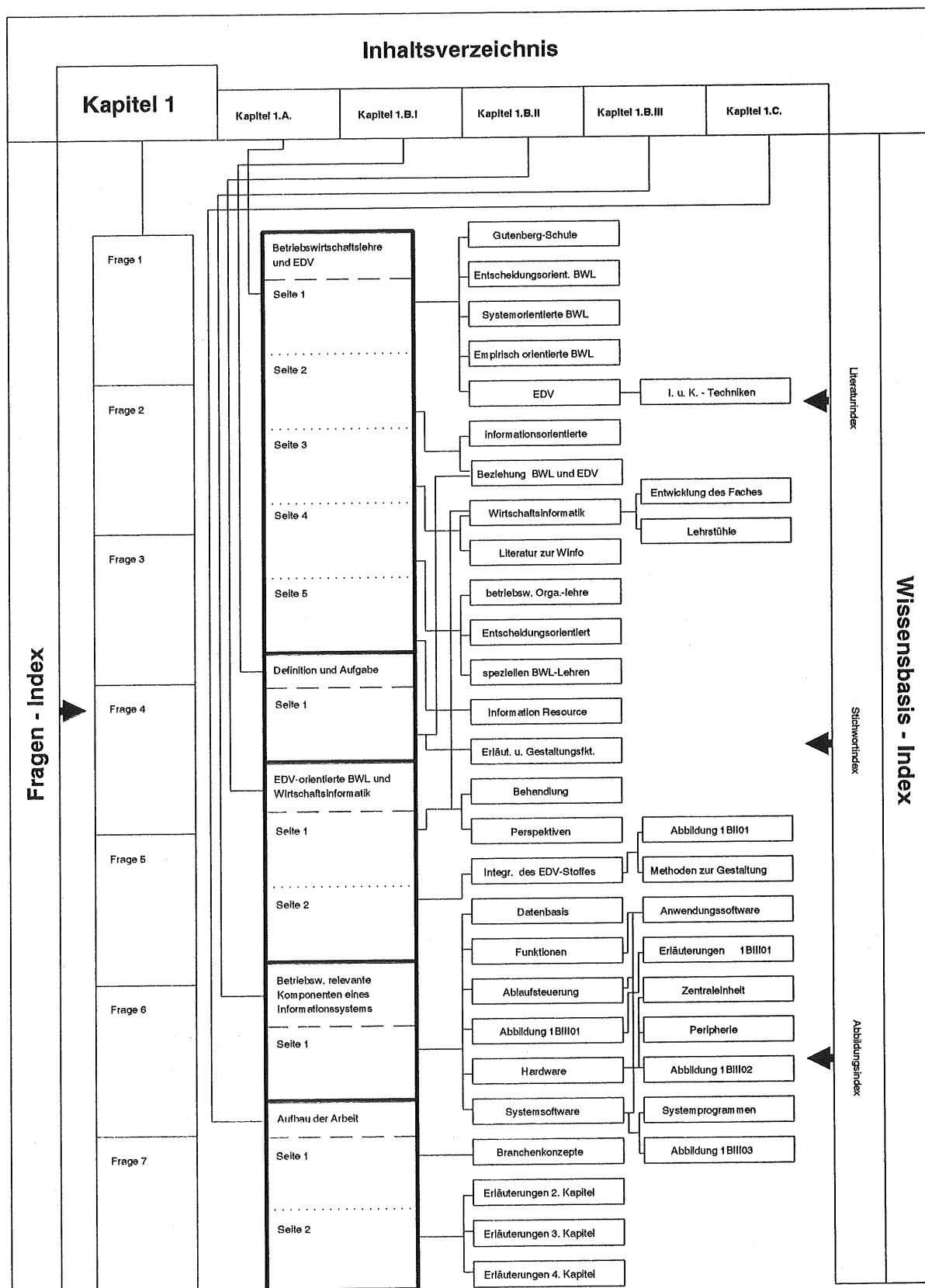


Abb. 4: Knotenstruktur des Prototypen "WITEC-Tutorial"

Diese Vorgehensweise hat folgende Vorteile:

- Die semantische Konsistenz zwischen Druckwerk und computergestütztem Tutorial bleibt gewährleistet, da der Zerlegungsprozeß jederzeit rückverfolgt werden kann.
- Der Zerlegungsprozeß kann in mehreren Schritten erfolgen und ist besser beherrschbar.
- Aufgrund der hierarchischen Struktur sind bereits Verknüpfungen zwischen den Informationseinheiten explizit vorgegeben.

Die Basis-Informationen repräsentieren das gedankliche Gerüst der Lehrinhalte. Sie sind für das Verständnis unabdingbar und stellen quasi ein "Konzentrat" des Wissensbereiches dar. Aufgrund der beschriebenen Vorgehensweise sind sie bereits linear miteinander verknüpft. Diese Linearität stellt eine wichtige Orientierungs- und Navigationshilfe für den Benutzer dar und wird im System durch die Blätterfunktion realisiert. Über das Inhaltsverzeichnis kann der Einstieg in den jeweiligen Bereich der Basis-Informationen erfolgen.

Auf Erläuterungen greift ein Benutzer erst dann zurück, wenn er zu einem bestimmten Sachverhalt eine Vertiefung wünscht. In der realisierten Benutzeroberfläche ist sowohl den Basis-Informationen als auch den Erläuterungen jeweils ein Fenster zugeordnet. Dies gestattet dem Benutzer die gleichzeitige Ansicht von Basis-Informationen und gerufenen Erläuterungen. Diese Konzeption gewährleistet ein hohes Maß an Flexibilität und wird den Anforderungen verschiedener Benutzergruppen gerecht.

Auf der Grundlage der bereits im Hierarchiebaum festgelegten Verknüpfungen erfolgt nun die Ergänzung um Verknüpfungen außerhalb der Baumstruktur. Das Ergebnis ist eine Netzstruktur, in der der Benutzer frei zwischen den Informationseinheiten navigieren kann. Diese Netzstruktur charakterisiert letztlich auch die Hyper-Struktur und ermöglicht den wahlfreien Zugriff innerhalb des Informationsraums.

Im letzten Arbeitsschritt erfolgt die Aufbereitung der direkten Zugriffspfade auf Informationseinheiten innerhalb der Hyper-Struktur. Neben dem Zugriff über das Inhaltsverzeichnis werden Direkt-Zugriffe wie Schlagwort-, Abbildungs- und Autoren-Index unterstützt.

In Abb. 5 ist die Objekt-Funktions-Matrix der Wissensbasis des WITEC-Tutorials dargestellt. Diese kann im Rahmen einer weiteren Detaillierung problemlos um Informationen zur DV-technischen Umsetzung erweitert werden und als Ausgangsbasis für eine Implementierung dienen. Abb. 6 zeigt die DV-technische Umsetzung der Benutzerschnittstelle für die Komponente "Wissensbasis".

| Komponente | | Wissensbasis | | | | | | |
|---------------------------|--------|---|--|--|---|--|---|-----------------|
| Funktionen | Objekt | Informationsleiste | Basistext | Basiserläuterungen | Abbildungsicon | Abbildungen | Abbildungserläuterungen | Literaturstelle |
| Blättern in Basistext | | durch Sprung zu einem neuen Basistext ändert sich die Infoleiste | einzelne Seiten innerhalb eines Themengebietes sind durch Blättern miteinander verbunden | mit Hilfe von Howwords werden Erläuterungen aufgerufen | mit Hilfe von Howwords werden neue Erläuterungen aufgerufen | | mit Hilfe von Howwords Erläuterungen aufrufen | |
| Text/Erläuterungsaufzuruf | | | mit Hilfe von Howwords werden Erläuterungen aufgerufen | mit Hilfe von Howwords werden Erläuterungen aufgerufen | | | | |
| Text/Abbildungsicon | | | zum Basistext werden direkt jeweilige Abbildungsicons aufgerufen | | | | | |
| Icon/Abbildungsaufzuruf | | durch Sprung zu einer Abbildung ändert sich die Infoleiste | | | nur jeweilige Abbildung auf | | | |
| Text/Literaturaufzuruf | | | mit Hilfe von Howwords werden jeweilige Literaturstellen angezeigt | mit Hilfe von Howwords werden jeweilige Literaturstellen angezeigt | | | | |
| Abbildungserläut.aufzuruf | | | | | | | | |
| Notizblock | | | eine Notiz kann einer bestimmten Basistextseite zugeordnet sein | eine Notiz kann einer bestimmten Erläuterungskarte zugeordnet sein | | in t. Hilfe von Howwords werden jeweilige Literaturstellen angezeigt | | |
| Mark | | | markiert eine Stelle im Basistext als Lesezeichen | markiert eine Basiserläuterung als Lesezeichen | | durch sensitive Felder wird eine Abbildung bzw. Teile einer Abb. erläutert | | |
| Weitere Basistexte | | durch Sprung zu einem neuen Basistext ändert sich die Infoleiste | zu einer Basistextseite wird angezeigt, welche weiteren Basistexte es gibt | | | eine Notiz kann einer bestimmten Abbildung zugeordnet sein | | |
| Schlagwortverzeichnis | | ermöglicht den direkten Zugriff auf bestimmte Sachverhalte, Schlagwörter können einzelne Begriffe innerhalb eines Basistextes und Howwords innerhalb eines Basistextes sein | | | | in t. Hilfe von Howwords werden jeweilige Literaturstellen angezeigt | | |
| Abbildungsverzeichnis | | ermöglicht den direkten Zugriff auf eine Abbildung über die Abbildungsnummer | | | | | | |
| Literaturverzeichnis | | alle Literaturquellen können direkt angesprochen werden | | | | | | |
| Inhalt | | ermöglicht dem Benutzer über das Inhaltsverzeichnis in einzelne Basistexte einzusteigen, mit Überblick über die gesamte Wissensbasis: Browser, bzw. direkt über einzelne Kapitelbezeichnungen | | | | | | |
| Back | | führt die zuletzt ausgeübte Funktion rückgängig | | | | | | |
| History | | zeigt an, welche Informationseinheiten vom Benutzer während einer Sitzung bereits bearbeitet wurden | | | | | | |
| Drucken | | einzelne Informationseinheiten können ausgegeben werden | | | | | | |
| Wechsel in Lernkontrolle | | ermöglicht den direkten Wechsel von der Wissensbasis in die Lernkontrolle nach einem zusammenhängenden Themengebiet | | | | | | |
| Standardwerte | | hier kann z.B. festgehalten werden, ob der Benutzer beim nächsten Systemaufruf wieder an der Stelle einsteigen möchte, an der | | | | | | |
| Ende | | Verlassen der Wissensbasis | | | | | | |
| Uhr, Datum | | zeigt aktuelle Uhrzeit und Datum an | | | | | | |
| Hilfe | | kann Hilfe in Bezug auf einzelne Funktionen geben oder gesamtes Tutorial aufrufen | | | | | | |
| WITEC-Info | | gibt eine Kurzinformation über das Projekt WITEC | | | | | | |
| Terminkalender | | gibt dem Benutzer die Möglichkeit, sich einen Terminplan, z.B. zum Erlernen der Wissensgebiete, anzustellen | | | | | | |
| Tour | | führt den Benutzer durch ein bestimmtes Themengebiet | | | | | | |

Abb. 5: Objekt-Funktions-Matrix der Komponente "Wissensbasis"

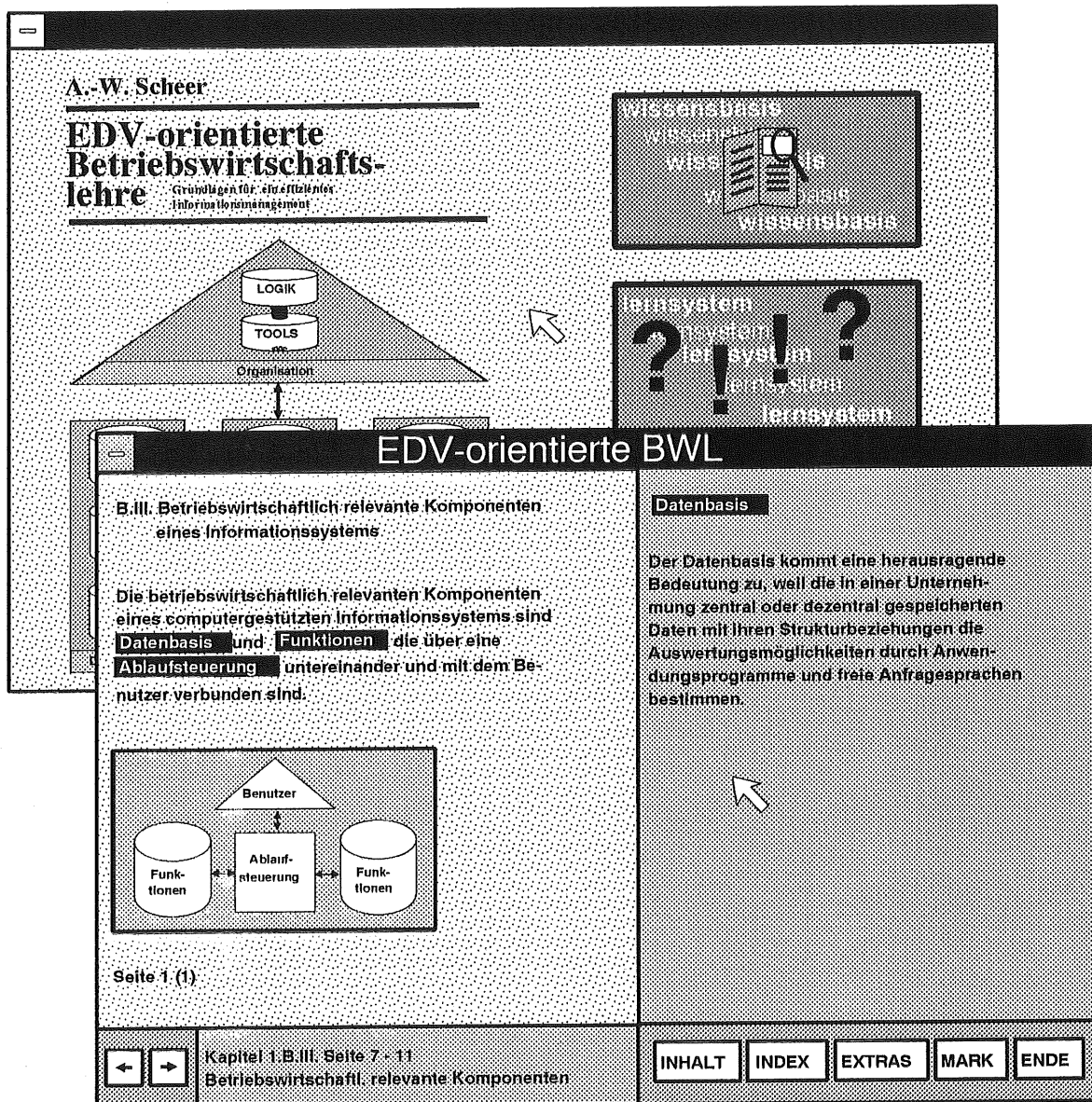


Abb. 6: Benutzerschnittstelle der Komponente "Wissensbasis"

Hyperbasierte Lernsysteme unterstützen zwar den Benutzer auch bei einer explorativen Suche, führen aber nicht zwangsläufig zum systematischen Aufspüren von Wissenslücken. Interaktive Lernkontrollen in der Form von Prüfungsfragen und Beurteilungen fördern daher die Akzeptanz computergestützter Lernsysteme in besonderem Maße und heben sie von konventionelle Lehr- und Lerntechniken ab. Die inhaltliche Ausgestaltung der einzelnen Fragen erfolgt im WITEC-Tutorial in enger Anlehnung an die Informationseinheiten der Wissensbasis.

Die Konzeption der Lernkontrolle ermöglicht den Aufruf der einzelnen Fragen auf folgenden Zugriffspfaden:

- über ein Inhaltsverzeichnis (analog dem Inhaltsverzeichnis der Wissensbasis)
- über einen Fragenindex
- über Verknüpfungen aus der Wissensbasis

Der wahlfreie Zugriff nach Abarbeitung bestimmter Lehrinhalte in der Wissensbasis auf die zugehörigen Teile der Lernkontrolle stellt hierbei die wohl didaktisch interessanteste Alternative dar. Die Integration einer Lernkontrolle wirft neben Fragen der inhaltlichen Ausgestaltung in besonderem Maße Fragen der Gestaltung der Interaktionstechnik auf. Um dem hohen Maß an Interaktion und Bediensicherheit Rechnung zu tragen, können in der realisierten Lernkontrolle alle Fragen direkt-manipulativ mittels Maus-Zeiger beantwortet werden.

Ein Instruct-Programm gewährleistet, daß auch im Umgang mit diesem Tutorial ungeübte Benutzer sich innerhalb weniger Minuten zurechtfinden.

4.3 Integrierte Dokumentationssysteme

Die Beurteilung eines integrierten Informationssystems wird sich in Zukunft weniger an DV-technischen Kriterien als an der durchgängigen Unterstützung der betriebswirtschaftlichen Anforderungen ausrichten. Diese Anforderungen spiegeln sich in semantischen Informationsmodellen wieder, welche beim Entwurf und der Implementierung von Informationssystemen zunehmend eine Schlüsselrolle einnehmen.

Informationsmodelle vermitteln zwischen dem betriebswirtschaftlichen Konzept und der DV-technischen Umsetzung. Semantische Informationsmodelle eignen sich aufgrund ihrer graphischen Beschreibungssprache in besonderem Maße zur transparenten Dokumentation betriebswirtschaftlich relevanter Tatbestände. Sie sollen bei wachsender Komplexität integrierter Informationssysteme die Beherrschbarkeit des Entwicklungs- und Anwendungsprozesses sicherstellen und die semantischen Strukturen eines Informationssystems offenlegen.

Daten-, Funktions-, Organisations- und Steuerungsmodelle bilden jedoch immer nur eine Teilsicht auf den gesamten Anwendungsbereich ab. Um die Komplexität des abzubildenden Sachverhaltes zu beherrschen und die Konsistenz innerhalb und zwischen den Teilmodellen

sicherzustellen, können computergestützte Modellierungswerkzeuge eingesetzt werden. Zielsetzung solcher Meta-Informationssysteme ist, die gewonnenen Informationen über primär statische Strukturen (Funktions-, Daten- und Organisationsmodelle) und ihr dynamisches Verhalten (Steuerungsmodelle) abzubilden.

Im Rahmen von CASE (Computer Aided Software Engineering) sind hierbei eine Vielzahl von Modellierungswerkzeugen entwickelt worden, welche alle jedoch nur bedingt den Anforderungen einer integrierten Informationsmodellierung genügen. Insbesondere der Gestaltung benutzerfreundlicher "front ends" zur Dokumentation der Fachinhalte wurde bisher kaum Beachtung geschenkt. Mit der Verfügbarkeit von Werkzeugen zur Gestaltung hyperbasierter Anwendungen bieten sich Möglichkeiten, die bisher unterrepräsentierten Funktionen der integrierten Dokumentation und Darstellung von Informationsmodellen umzusetzen. Ein solches integriertes Dokumentationssystem kann auch als "hyperbasiertes Meta-Informationssystem" oder "hyperbasiertes Repository" bezeichnet werden.

Mit dem Prototyp "ARIS-Repository" wird die Zielsetzung verfolgt, die einem anwendungsbezogenen Informationssystem zugrundeliegende "Meta-Struktur" in einem solchen übergreifenden hyperbasierten Repository abzulegen. Um eine anwendungsübergreifende Dokumentation zu unterstützen, folgt dieses Dokumentationssystem der ARIS-Architektur [52].

Während konventionelle Dictionary-Systeme nur eine maskenorientierte alphanumerische Darstellung der Meta-Struktur unterstützen, verfügt das "ARIS-Repository" über eine objektorientierte Benutzerschnittstelle. Es unterstützt den Anwender beim zielgerichteten Navigieren durch die jeweiligen Beschreibungssichten und -ebenen unter einer durchgängigen Zugriffsstruktur, wobei die zugrundeliegenden methodischen Grundlagen und Konstrukte in einem elektronischen Methodenhandbuch niedergelegt sind.

Durch die lockere Kopplung von datenbankgestütztem Repository und hyperbasierter Benutzerschnittstelle ist das System sowohl zur Übernahme von Daten aus aktiven CASE-tools als auch zur Online-Dokumentation geeignet. Hierbei werden in einem ersten Schritt die Daten mittels entsprechenden Dateneingabemasken bzw. Schnittstellen im Repository erfaßt und dann um die entsprechenden Navigationsstrukturen ergänzt.

In Abb. 7 ist die Benutzeroberfläche des Prototypen wiedergegeben

[52]: Zur ausführlichen Darstellung vgl. Scheer, *Architektur*, 1991.

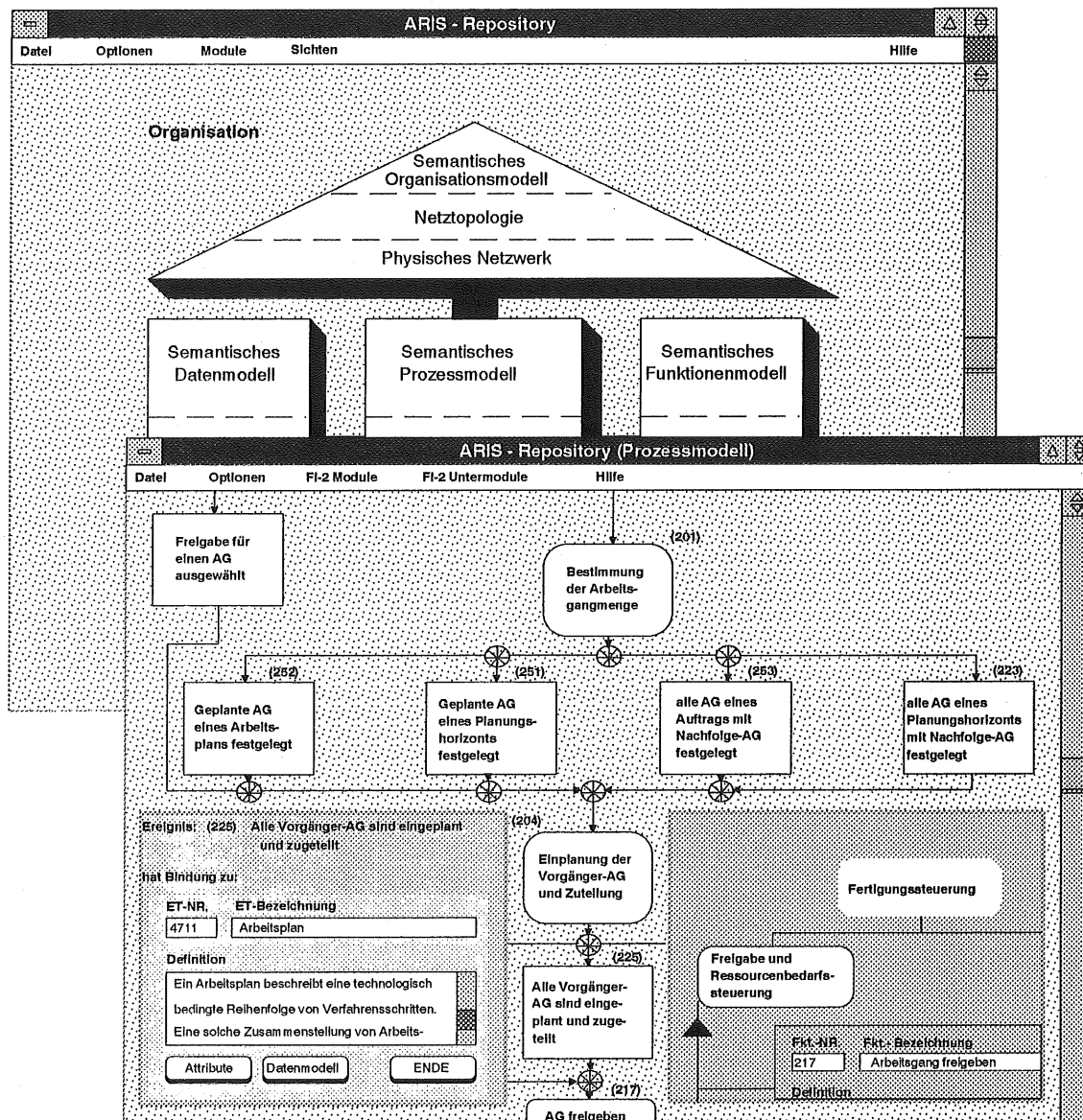


Abb. 7: Benutzeroberfläche des Prototypen "ARIS-Repository"

Neben einer Dokumentation der semantischen Struktur eines Informationssystems (build-time control) kann das ARIS-Repository auch zur Steuerung eines konkreten Informationssystems eingesetzt werden (run-time control). Dies betrifft auch die Umsetzung einer semantischen Steuerung zur zielgerichteten Navigation durch den Dialogteil eines Informationssystems unter den am IWi entwickelten Ereignisgesteuerten Prozeßketten (EPK's) [53] oder aber den koordinierten Zugriff auf Datenbankinhalte über semantische Datenmodelle. Der Aufruf einer Bearbeitungsmaske erfolgt dann ereignisorientiert und nicht funktionsorientiert, wodurch eine zielbezogene Reduktion der angebotenen Bearbeitungsfunktionen ermöglicht wird.

[53]: Vgl. Keller/Kirsch/Nüttgens/Scheer, *Informationsmodellierung*, 1991.

5. Modellierungsaspekte hyperbasierter Informationssysteme

Die Verfügbarkeit geeigneter Verfahren im Sinne des Software-Engineering wird **entscheidend die Entwicklung größerer hyperbasierter Informationssysteme** beeinflussen; das Defizit wird heute insofern verdeckt, als der Entwurf und die Implementierung von Prototypen noch selten Größenordnungen erreicht, welche eine durchgängige Unterstützung des Softwareentwurfes notwendig erscheinen läßt.

5.1 Hyper-Engineering

Der Aufbau eines hyperbasierten Informationssystems setzt eine adäquate Dokumentation der relevanten Objekttypen und ihrer Beziehungen voraus [54]. Der Umfang eines solchen Systems übersteigt schnell die Dimension von tausend Objekten, welche wiederum über Beziehungen miteinander verknüpft sind. Da die Komplexität eines solchen Systems mit der Anzahl der Objekte überproportional wächst, gewährleistet nur eine übergreifende Dokumentationssystematik die Konsistenz des Gesamtsystems. Dies ist auch Voraussetzung für den Entwurf und die Implementierung durch ein Team oder an räumlich verteilten Standorten.

Beim Entwurf und der Implementierung eines hyperbasierten Informationssystems ist es zweckmäßig, schrittweise vorzugehen. Hierbei hat sich eine Vorgehensweise nach den Beschreibungsphasen der ARIS-Architektur bewährt (vgl. Abb. 8) [55]:

1. Entwurf der semantischen Strukturen, welche im System abgebildet werden sollen (Knotenstruktur, Ankerpunkte, Funktionstypen, Informationsobjekte, Navigationspfade)
2. Spezifizierung der DV-technischen Umsetzung (Benutzeroberfläche, Interaktionstechnik, Programmodule, Datenzugriff)
3. Implementierung mittels geeigneter EDV-Werkzeuge wie Hyper-shells, objektorientierten Programmiersprachen, wissensbasierten Technologien etc.

Obwohl das Hyper-Konzept primär dem objektorientierten Paradigma folgt, erscheint es sinnvoll, den anwendungsbezogenen Systementwurf sichtenorientiert durchzuführen. Mit der ARIS-Architektur wird die Zielsetzung verfolgt, sichtenbezogene Komponenten eines Informationssystems zu integrieren und der hohen Gestaltungswirkung von Datenbanken,

[54]: Vgl. Dürr/Neske, *Datenbanken*, 1990, S. 154ff.

[55]: Vgl. Scheer, *Architektur*, 1991.

Programmbibliotheken und verteilten Anwendungen Rechnung zu tragen. Im Rahmen einer sichtenorientierten Beschreibung nach ARIS kann das Hyper-Konzept der Steuerungssicht zugeordnet werden (vgl. Abb. 8).

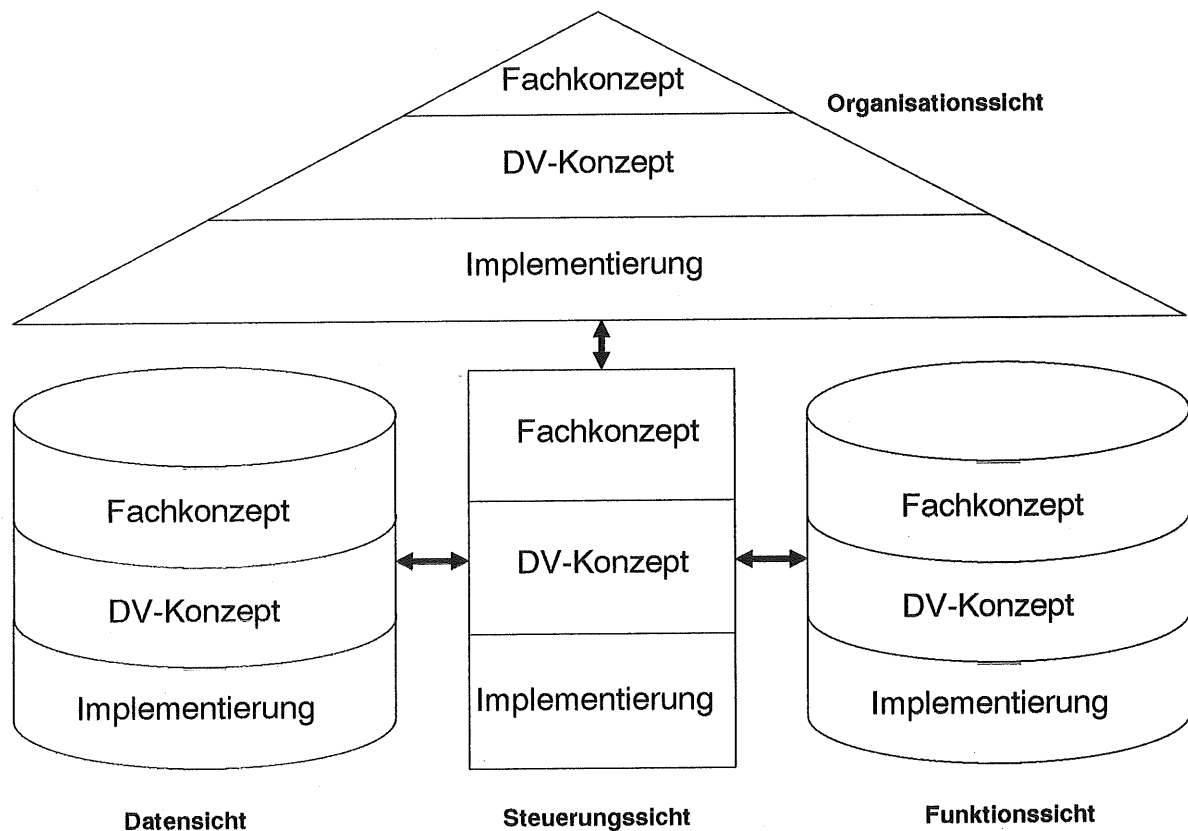


Abb. 8: ARIS-Architektur [56]

Weitere in der Literatur diskutierte Architekturvorschläge sind das Dexter-Reference Model [57] und das Dreistufenmodell von Campbell und Goodman [58]. Im Gegensatz zur ARIS-Architektur vertreten diese Ansätze aber kein Sichtenkonzept und haben keine eigene semantische Beschreibungsebene, welche von den DV-technischen Umsetzungsaspekten abstrahiert.

[56]: Vgl. Scheer, *Architektur*, 1991.

[57]: Vgl. Halasz/Schwartz, *Dexter*, 1990.

[58]: Vgl. Campbell/Goodman, *HAM*, 1988.

5.2 Hyper-Informationsmodell

Das nachfolgend aufgezeigte objektorientierte Hyper-Informationsmodell stellt einen **Vorschlag für eine semantische Beschreibung** hyperbasierter Informationssysteme dar. Es abstrahiert sowohl von einer konkreten Anwendung als auch von Aspekten der DV-technischen Umsetzung. Als Beschreibungssprache wurde das Entity-Relationship-Modell (ERM) von Chen gewählt [59]. Mit Hilfe des ERM können die Objekte eines Informationsmodells in der Form von Entitytypen und Beziehungstypen einheitlich dargestellt werden [60]. Nachfolgend wird das in Abb. 9 wiedergegebene Hyper-Informationsmodell erläutert.

Die zentralen Objekte des Hyper-Informationsmodells sind der **KNOTEN** und der **ANKER**. Die Definition eines Knotens orientiert sich an der Präsentation von Informationseinheiten gegenüber einem Benutzer. Die Anker verkörpern den Steuerungsaspekt, welcher sich in der Ausführung von Funktionen oder dem Aufruf von Knoten äußert. Der Zusammenhang zwischen Knoten und Anker wird durch die Beziehungstypen **ANKERZUORDNUNG** und **AUFRUFZUORDNUNG** ausgedrückt. Die Ankerzuordnung beschreibt die Zuordnung der Anker zu den Knoten, wobei ein Anker mindestens einem Knoten zugewiesen sein muß. Die Aufrufzuordnung beinhaltet diejenigen Anker, welche Knoten rufen können. Diese Differenzierung ist notwendig, da Anker nicht zwangsläufig den Aufruf eines Knotens repräsentieren. Durch die Ankerzuordnung und die Aufrufzuordnung zu den Knoten kann ein semantisches Netz erzeugt werden, welches die Gesamtheit aller Kanten in der Form eines gerichteten Referenzgraphen verkörpert. Es ist auch möglich, über diese Datenstruktur die semantischen Informationen eines Browsers zu generieren.

Zwischen Knoten können durch die Bildung zusammengesetzter Knoten überlappende Beziehungen (Cluster) bestehen. Diese werden durch eine m:n-Beziehung innerhalb des Objektes **KNOTEN** ausgedrückt. Der Beziehungstyp **KNOTENSTRUKTUR** erlaubt allgemeine Objekteigenschaften auf hierarchisch höheren Ebenen festzulegen, während dies für spezielle Objekteigenschaften auf tieferliegenden Ebenen geschieht.

Knoten können aufgrund bestimmter Strukturmerkmale klassifiziert werden. Diese Klassifizierung wird durch das Objekt **KNOTENTYP** zum Ausdruck gebracht. Eine derartige Zuordnung ermöglicht auch eine konsistente Umsetzung des sachlogischen Konzeptes in eine DV-technische Beschreibungsebene in der Form gleicher Fenstertypen (Größe, Form, Farbe) und wiederkehrender Strukturmerkmale (Masken, Symbole, Anordnung).

[59]: Vgl. Chen: *Entity-Relationship Model*, 1976

[60]: *Zum Entwurf und zur Darstellung von Informationsmodellen vgl.: Scheer, Architektur, 1991.*

Anker können ebenfalls wie Knoten klassifiziert werden. Der semantische Gehalt des Objektes **ANKERTYP** kann zur Vereinheitlichung der DV-technischen Umsetzung der Interaktionstechnik und der Manipulationsobjekte (Hotwords, Hyperregions, Buttons etc.) genutzt werden.

Das Objekt **INFORMATIONSOBJEKT** repräsentiert die Semantik der Daten, welche in strukturierter und unstrukturierter Form vorliegen können. Die Strukturbeziehungen zwischen Daten werden in dem rekursiven Beziehungstyp **DATENSTRUKTUR** zum Ausdruck gebracht. Durch die Klassifizierung von Medientypen in arithmetische Daten, Texte, Graphiken, Bildsequenzen und Tonsequenzen werden medienspezifische Eigenschaften den Informationsobjekten zugeordnet. Sie stellen die Schnittstelle zu DV-technischen Aspekten wie Standardformaten und Speicherformen dar (verteilte multimediale Datenbanken).

Das Objekt **FUNKTION** repräsentiert im Sinne der Objektorientierung die Methoden, welche über den Beziehungstyp **FUNKTIONZUORDNUNG** den Anker zugeordnet sind. Hierdurch ist es auch möglich, komplette externe Standardfunktionen miteinzubeziehen. Funktionen können untereinander verflochten sein, was durch den rekursiven Beziehungstyp **FUNKTIONSTRUKTUR** zum Ausdruck gebracht wird. Funktionen können nach Funktionstypen klassifiziert werden. Das Objekt **FUNKTIONSTYP** repräsentiert hierbei typische Hyperfunktionalitäten oder aber auch externe Funktionstypen wie komplexe Datenbankabfragen, statistische Berechnungen etc.

Die Organisationssicht wird durch den **BENUTZER** und das **BENUTZERMODELL** charakterisiert. Einem **BENUTZERMODELL** wird über den Beziehungstyp **SICHT** ein Ausschnitt des Referenzgraphen zugeordnet. Dies ist beispielsweise der Fall, wenn eine Benutzermodellierung nach Kriterien wie Entwickler, Experte oder Gelegenheitsnutzer erfolgt. Der Benutzer kann im Rahmen des ihm zugewiesenen Ausschnittes eines Referenzgraphen interaktiv die Funktionsausführungen und Knotenzugriffe steuern oder durch vordefinierte Sequenzen automatisch geführt werden (guided tour).

Die Zuordnung eines Benutzers zu einem Benutzermodell drückt sich durch den Beziehungstyp **KLASSIFIZIERUNG** aus. Der Weg eines Benutzers durch das semantische Netz entspricht dem Beziehungstyp **PFAD**. Die Dokumentation eines Pfades wird z. B. zur Realisierung einer Backtrack- oder History-Funktion benötigt. Das Objekt **BENUTZERGRUPPIERUNG** beschreibt Organisationseinheiten im Zusammenhang kooperativer Gruppenarbeit. In diesem Sinne werden auch Aspekte der rechnergestützten Kooperation mit dem Schlagwort "computer-supported cooperative work" (CSCW) diskutiert.

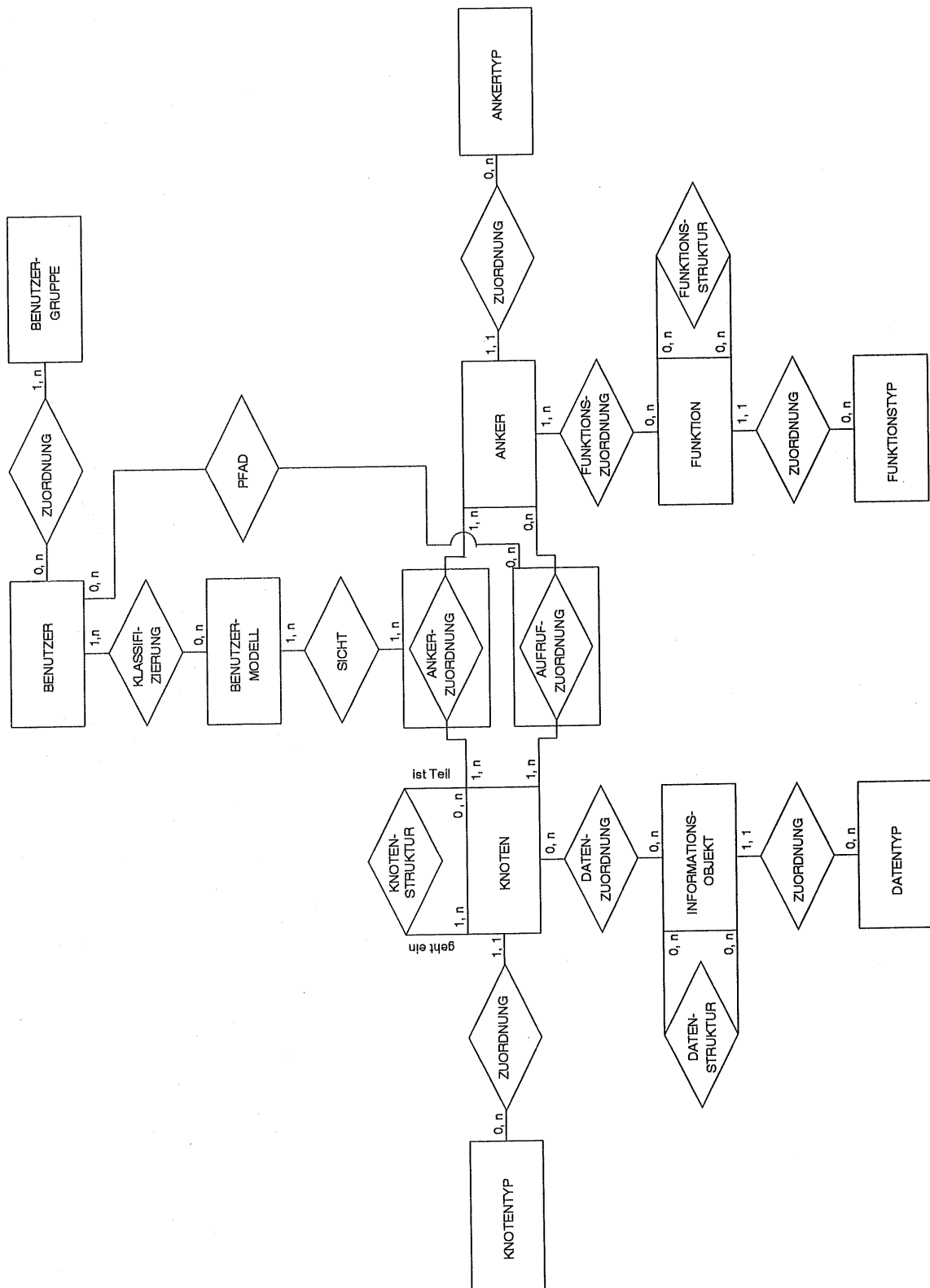


Abb. 9: Hyper-Informationsmodell

6. Zusammenfassung und Ausblick

Der größte Vorteil eines hyperbasierten Informationssystems ist der fehlende Zwang zur formalen Strukturierung der Informationen. Informationen werden statt dessen anhand semantischer Kriterien verknüpft. Dies ermöglicht nichtlineare semantische Informationsnetze, den benutzerfreundlichen navigierenden Informationszugriff und die Integration verschiedenartiger Anwendungen. Weiterer Vorteil ist die Multimedialität. Sie eröffnet neue Gestaltungsmöglichkeiten für eine aufgabenbezogene, objektorientierte Benutzeroberfläche.

Die konzeptionell bedingten Vorteile des Hyper-Konzeptes führen bei der praktischen Umsetzung gleichzeitig zu konzeptionell bedingten Problemen. Zur Lösung dieser Probleme sind zwei Ansätze möglich:

- Die Entwicklung einer eigenständigen Disziplin "Hyper-Engineering", welche sich mit Prinzipien, Methoden und Werkzeugen zur Entwicklung hyperbasierter Informationssysteme auseinandersetzt.
- Die Integration bereits bestehender Disziplinen, sei es der Einbezug der KI, insbesondere zum intelligenten Navigieren oder der Einbezug konventioneller datenbankgestützter Ansätze.

Der unmittelbare Bezug zwischen Benutzerschnittstelle und Anwendung hat auch zur Folge, daß neben informationstechnischen Gesichtspunkten zunehmend verhaltenswissenschaftliche Aspekte beim Design hyperbasierter Systeme berücksichtigt werden müssen. Dies führt auch zu der Forderung, daß Entwickler von hyperbasierten Informationssystemen über eine **interdisziplinäre Kompetenz** verfügen müssen. In diesem Forschungsbereich sind noch weitreichende Arbeiten zu leisten und werden das Design von hyperbasierten Informationssystemen stark beeinflussen.

Es darf mit Spannung erwartet werden, wie das Hyper-Konzept Impulse für die Entwicklung einer neuen Softwaregeneration geben und das Gesicht integrierter Informationssysteme verändern wird.

7. Literaturverzeichnis

ACM (Hrsg.): *Special 1988*

Communications of the ACM, Special Issue on Hypertext. Vol. 31 (1988) No. 7.

ACM (Hrsg.): *Hypertext'89 1989*

Proceedings ACM Conference Hypertext'89. Pittsburgh (VA) November 1989.

Akscyn, R. M.; McCracken, D. L.; Yoder E. A.: *KMS 1988*

KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. In: Communications of the ACM, Vol. 31 (1988) No. 7, S. 820-835.

Bernstein, M.: *Apprentice 1990*

An Apprentice That Discovers Hypertext Links. In: Rizk, A.; Streitz, N. A.; André, J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First European Conference on Hypertext, INRIA, France, November 1990. Cambridge, 1990, S. 212-223.

Balzert, H. u. a. (Hrsg.): *Software-Ergonomie 1988*

Einführung in die Software-Ergonomie. Berlin u. a. 1988.

Burch, J. G. Jr.; Strater, F. R.; Grudnitzki, G.: *Information Systems 1979*

Information Systems: Theorie and Practice. New York u. a. 1979.

Bush, V.: *As We May Think 1945*

As We May Think. In: Atlantic Monthly 176, July 1945, S. 101-108. Neu veröffentlicht in: CD-ROM: The New Papyrus, Microsoft Press 1986.

Campbell, B.; Goodman, J. M.: *HAM 1988*

HAM: A General Purpose Hypertext Abstract Machine. In: Communications of the ACM, Vol. 31 (1988) No. 7, S. 856-861.

Carlson, D. A.; Ram, S.: *HyperIntelligence 1990*

HyperIntelligence: The Next Frontier. In: Communications of the ACM, Vol. 33 (1990) No. 3, S. 311-321.

Chen, P. P.: *Entity-Relationship Model 1976*

The Entity-Relationship Model: Towards a Unified View of Data. In: ACM Transactions on Database-Systems, Vol. 1 (1976) No. 1, S. 9-36.

Conklin, J.: *Introduction 1987*

Hypertext: An Introduction and Survey. In: IEEE Computer, Vol. 20 (1987) No. 9, S. 17-41.

Dam, A. v.: *Address 1988*

Hypertext '87 Keynote Address. In: Communications of the ACM, Vol. 31 (1988) No. 7, S. 887-895.

De Young, L.: *Linking 1990*

Linking Considered Harmful. In: Rizk, A.; Streitz, N. A.; André, J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First European Conference on Hypertext, INRIA, France, November 1990. Cambridge 1990, S. 238-249.

DIN (Hrsg.): *DIN 66234 Teil 8 1988*

DIN 66234 Teil 8 - Bildschirmarbeitsplätze - Grundsätze ergonomischer Dialoggestaltung. Berlin 1988.

Dürr, M.; Neske, R.: *Datenbanken 1990*

Hypertext und Datenbanken: Gegensatz oder Symbiose? In: Gloor, P. A.; Streitz, N. A. (Hrsg.): Hypertext und Hypermedia - Von theoretischen Konzepten zur praktischen Anwendung. Informatik Fachberichte Nr. 249, Berlin u. a. 1990, S. 149-161.

Fährnich, K. P. (Hrsg.): *Software-Ergonomie 1987*

Software-Ergonomie. München-Wien 1987.

Frese, M.; Schulte-Göcking, H.; Altman, A.: *Lernprozesse 1987*

Lernprozesse in Abhängigkeit von der Trainingsmethode, von Personenmerkmalen und von der Benutzeroberfläche (Direkte Manipulation vs. konventionelle Interaktion). In: Schönplüg, W.; Wittstock (Hrsg.): Software-Ergonomie '87, Berichte des German Chapter of the ACM, Band 29, Stuttgart 1987, S. 55-73.

Gloor, P.; Streitz, N. (Hrsg.): *Hypertext 1990*

Hypertext und Hypermedia - Von theoretischen Konzepten zur praktischen Anwendung; Proceedings der Hypertext/Hypermedia-Tagungen in Basel und Darmstadt 1990. Informatik Fachberichte, Bd. 249, Berlin u. a. 1990.

Halasz, F. G.: *Notecards 1988*

Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems. In: Communications of the ACM, Vol. 31 (1988) No. 7, S. 836-852.

Halasz, F. G.; Schwartz, M.: *Dexter, 1990*

The Dexter Hypertext Reference Model. In: Moline, J. u. a.; Proceedings of the NIST Hypertext Standardization Workshop, Gaithersburgh 1990, S. 95-133.

Heinzelbecker, K.: *Marketing-Informationssysteme 1985*

Marketing-Informationssysteme. Stuttgart u. a. 1985.

Hoffmann, M.: *Hypertextsystem 1991*

Hypertextsysteme - Begrifflichkeit, Modelle, Problemstellungen. In: Wirtschaftsinformatik, 33 (1991) 3, S. 177-185.

Irler, W. J.; Barbieri, G.: *Anchors 1990*

Non-Intrusive Hypertext Anchors and Individual Colour Markings. In: Rizk, A.; Streitz, N. A.; André, J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First European Conference on Hypertext, INRIA, France, November 1990. Cambridge 1990, S. 261-273.

Jonassen, D. H.: *Designing structured Hypertext 1988*

Designing structured Hypertext and structuring access to Hypertext. In: Educational Technology Vol. 28 (1988) No. 11, S. 13f.

Jonassen, D. H.; Mandl, H. (Hrsg.): *Designing Hypermedia 1990*

Designing Hypermedia for Learning, Proceedings of the NATO Advanced Research Workshop (ARW) in Rottenburg am Neckar, Juli 1989, NATO ASI Series F, Vol. 67. Berlin u. a. 1990.

- Keller, G.; Kirsch, J.; Nüttgens, M.; Scheer, A.-W.: *Informationsmodellierung 1991*
Informationsmodellierung in der Fertigungssteuerung. In: Scheer, A.-W. (Hrsg.):
Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 80, Saarbrücken 1991.
- Lucarella, D.: *Information Retrieval 1990*
A Model for Hypertext-Based Information Retrieval. In: Rizk, A.; Streitz, N. A.; André,
J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First
European Conference on Hypertext, INRIA, France, November 1990. Cambridge 1990,
S. 81-94.
- Marchioni, G.; Shneiderman, B.: *Finding 1988*
Finding Facts vs. Browsing Knowledge in Hypertext Systems. In: IEEE Computer, Vol.
21 (1988) No. 1, S. 70-80.
- Maurer, H. (Hrsg.): *Hypertext/Hypermedia'91 1991*
Hypertext/Hypermedia'91, Proceedings des gleichnamigen Symposiums der GI, SI und
OCG in Graz, Mai 1991, Informatik-Fachberichte, Bd. 276. Berlin u. a. 1991.
- Mockler, R. J.: *Information Systems 1974*
Information Systems for Management, Columbus (Ohio) 1974.
- Mühlhäuser, M.: *Hypermedia-Konzept 1991*
Hypermedia-Konzepte zur Verarbeitung multimedialer Information. In Informatik
Spektrum 14 (1991) 5, S. 281-290.
- Nastansky, L.; Seidensticker, F.-J.: *Hypermedia-basiertes Informationsmanagement 1990*
Anwendungen und Konzepte für Hypermedia-basiertes Informationsmanagement am
netzintegrierten Managerarbeitsplatz. In: Wirtschaftsinformatik 32 (1990) 6, S. 519-
537.
- Nelson, T. H.: *Getting it Out 1967*
Getting it Out of Our System. In: Schechter, G. (Hrsg.): Information Retrieval: A
Critical Review. Washington D.C. 1967.
- Nielsen, J.: *Hypertext 1990*
Hypertext and Hypermedia. Boston u.a. 1990.

Ogawa, R.; Harada, H.; Kaneko, A.: *Scenario-based 1990*

Scenario-based Hypermedia: A Model and a System. In: Rizk, A.; Streitz, N. A.; André, J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First European Conference on Hypertext, INRIA, France, November 1990. Cambridge 1990, S. 38-51.

Pintado, X.; Tschritzis, D. *SaTellite 1990*

SaTellite: Hypermedia Navigation by Affinity. In: Rizk, A.; Streitz, N. A.; André, J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First European Conference on Hypertext, INRIA, France, November 1990, Cambridge 1990, S. 274-287.

Rizk, A.; Streitz, N. A.; André, J.: *ECHT 1990*

Rizk, A.; Streitz, N. A.; André, J. (Hrsg.): Hypertext: Concepts, Systems and Applications - Proceedings of the First European Conference on Hypertext, INRIA, France, November 1990. Cambridge 1990.

Schcer, A.-W.: *Architektur 1991*

Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung. Berlin u.a. 1991.

Shneiderman, B.: *User Interface 1987*

Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading (Massachusetts) u. a. 1987.

Shneiderman, B.; Kearsley, G.: *Hypertext Hands-On 1989*

Hypertext Hands-On!: An Introduction to a New Way of Organizing an Accessing Information. Reading (Massachusetts) u. a. 1989.

Smith, J. B. u. a.: *Hypertext'87 1987*

Smith, J. B.; Halasz, F.; Yankelovich, N.; Schwartz, M.; Weiss, S. F. (Hrsg.): Proceedings ACM Workshop Hypertext'87. Chapter Hill (NC), November 1987.

Smith, J. B.; Weiss, St. F.: *Hypertext 1988*

Hypertext. In: Communications of the ACM, Vol. 31 (1988) No. 7, S. 816-819.

Tauber, M.: *Mentale Modelle 1985*

Mentale Modelle: Benutzen und Lernen. In: Handbuch der modernen Datenverarbeitung, 22 (1985) 126, S. 45-53.

Yankelovich, N. u. a.: *Intermedia 1988*

Intermedia: The Concept and the Construction of a Seamless Information Environment. In: IEEE Computer, Vol. 21 (1988) No. 1, S. 81-96.

Die Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Institut für empirische Wirtschaftsforschung an der Universität des Saarlandes erscheinen in unregelmäßiger Folge.

* Die Hefte 1 - 31 werden nicht mehr verlegt.

- Heft 32: A.-W. Scheer: Einfluß neuer Informationstechnologien auf Methoden und Konzepte der Unternehmensplanung, März 1982, Vortrag anläßlich des Anwendergespräches "Unternehmensplanung und Steuerung in den 80er Jahren in Hamburg vom 24. - 25.11.1981
- Heft 33: A.-W. Scheer: Dispositon- und Bestellwesen als Baustein zu integrierten Warenwirtschaftssystemen, März 1982, Vortrag anläßlich des gdi-Seminars "Integrierte Warenwirtschafts-Systeme" in Zürich vom 10. - 12. Dezember 1981
- Heft 34: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS - Ein Ansatz zur Entwicklung prüfungsgerechter Software-Systeme, Mai 1982
- Heft 35: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Konzept einer computergestützten Prüfungsumgebung, Juli 1982
- Heft 36: A.-W. Scheer: Rationalisierungserfolge durch Einsatz der EDV - Ziel und Wirklichkeit, August 1982, Vortrag anläßlich der 3. Saarbrücker Arbeitstagung "Rationalisierung" in Saarbrücken vom 04. - 06. 10.1982
- Heft 37: A.-W. Scheer: DV-gestützte Planungs- und Informationssysteme im Produktionsbereich, September 1982
- Heft 38: A.-W. Scheer: Interaktive Methodenbanken: Benutzerfreundliche Datenanalyse in der Marktforschung, Mai 1983
- Heft 39: A.-W. Scheer: Personal Computing - EDV-Einsatz in Fachabteilungen, Juni 1983
- Heft 40: A.-W. Scheer: Strategische Entscheidungen bei der Gestaltung EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 41: H. Krcmar: Schnittstellenprobleme EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 42: A.-W. Scheer: Factory of the Future, Vorträge im Fachausschuß "Informatik in Produktion und Materialwirtschaft" der Gesellschaft für Informatik e. V., Dezember 1983
- Heft 43: A.-W. Scheer: Einführungsstrategie für ein betriebliches Personal-Computer-Konzept, März 1984
- Heft 44: A.-W. Scheer: Schnittstellen zwischen betriebswirtschaftlicher und technische Datenverarbeitung in der Fabrik der Zukunft, Juli 1984
- Heft 45: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Ein Werkzeug zur Messung der Qualität von Software-Systemen, August 1984
- Heft 46: H. Krcmar: Die Gestaltung von Computer am-Arbeitsplatz-Systemen - ablauforientierte Planung durch Simulation, August 1984
- Heft 47: A.-W. Scheer: Integration des Personal Computers in EDV-Systeme zur Kostenrechnung, August 1984

- Heft 48: A.-W. Scheer: Kriterien für die Aufgabenverteilung in Mikro-Mainframe Anwendungssystemen, April 1985
- Heft 49: A.-W. Scheer: Wirtschaftlichkeitsfaktoren EDV-orientierter betriebswirtschaftlicher Problemlösungen, Juni 1985
- Heft 50: A.-W. Scheer: Konstruktionsbegleitende Kalkulation in CIM-Systemen, August 1985
- Heft 51: A.-W. Scheer: Strategie zur Entwicklung eines CIM-Konzeptes - Organisatorische Entscheidungen bei der CIM-Implementierung, Mai 1986
- Heft 52: P. Loos, T. Ruffing: Verteilte Produktionsplanung und -steuerung unter Einsatz von Mikrocomputern, Juni 1986
- Heft 53: A.-W. Scheer: Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung, Juli 1986
- Heft 54: U. Leismann, E. Sick: Konzeption eines Bildschirmtext-gestützten Warenwirtschaftssystems zur Kommunikation in verzweigten Handelsunternehmen, August 1986
- Heft 55: D. Steinmann: Expertensysteme (ES) in der Produktionsplanung und -steuerung (PPS) unter CIM-Aspekten, November 1987, Vortrag anlässlich der Fachtagung "Expertensysteme in der Produktion" am 16. und 17.11.1987 in München
- Heft 56: A.-W. Scheer: Enterprise wide Data Model (EDM) as a Basis for Integrated Information Systems, Juli 1988
- Heft 57: A.-W. Scheer: Present Trends of the CIM Implementation (A qualitative Survey) Juli 1988
- Heft 58: A.-W. Scheer: CIM in den USA - Stand der Forschung, Entwicklung und Anwendung, November 1988
- Heft 59: R. Herterich, M. Zell: Interaktive Fertigungssteuerung teilautonomer Bereiche, November 1988
- Heft 60: A.-W. Scheer, W. Kraemer: Konzeption und Realisierung eines Expertenunterstützungssystems im Controlling, Januar 1989
- Heft 61: A.-W. Scheer, G. Keller, R. Bartels: Organisatorische Konsequenzen des Einsatzes von Computer Aided Design (CAD) im Rahmen von CIM, Januar 1989
- Heft 62: M. Zell, A.-W. Scheer: Simulation als Entscheidungsunterstützungsinstrument in CIM, September 1989
- Heft 63: A.-W. Scheer: Unternehmens-Datenbanken - Der Weg zu bereichsübergreifenden Datenstrukturen, September 1989
- Heft 64: C. Berkau, W. Kraemer, A.-W. Scheer: Strategische CIM-Konzeption durch Eigenentwicklung von CIM-Modulen und Einsatz von Standardsoftware, Dezember 1989
- Heft 65: A. Hars, A.-W. Scheer: Entwicklungsstand von Leitständen^[1], Dezember 1989
- Heft 66: W. Jost, G. Keller, A.-W. Scheer: CIMAN - Konzeption eines DV-Tools zur Gestaltung einer CIM-orientierten Unternehmensarchitektur, März 1990
- Heft 67: A.-W. Scheer: Modellierung betriebswirtschaftlicher Informationssysteme (Teil 1:

Logisches Informationsmodell), März 1990

- Heft 68: W. Kraemer: Einsatzmöglichkeiten von Expertensystemen in betriebswirtschaftlichen Anwendungsgebieten, März 1990
- Heft 69: A.-W. Scheer, R. Bartels, G. Keller: Konzeption zur personalorientierten CIM-Einführung, April 1990
- Heft 70: St. Spang, K. Ibach: Zum Entwicklungsstand von Marketing-Informationssystemen in der Bundesrepublik Deutschland, September 1990
- Heft 71: D. Aue, M. Baresch, G. Keller: URMELE, Ein Unternehmensmodellierungsansatz, Oktober 1990
- Heft 72: M. Zell: Datenmanagement simulationsgestützter Entscheidungsprozesse am Beispiel der Fertigungssteuerung, November 1990
- Heft 73: A.-W. Scheer, M. Bock, R. Bock: Expertensystem zur konstruktionsbegleitenden Kalkulation, November 1990
- Heft 74: R. Bartels, A.-W. Scheer: Ein Gruppenkonzept zur CIM-Einführung, Januar 1991
- Heft 75: M. Nüttgens, St. Eichacker, A.-W. Scheer: CIM-Qualifizierungskonzept für Klein- und Mittelunternehmen (KMU), Januar 1991
- Heft 76: Ch. Houy, J. Klein: Die Vernetzungsstrategie des Instituts für Wirtschaftsinformatik - Migration vom PC-Netzwerk zum Wide Area Network (noch nicht veröffentlicht)
- Heft 77: W. Kraemer: Ausgewählte Aspekte zum Stand der EDV-Unterstützung für das Kostenmanagement: Modellierung benutzerindividueller Auswertungssichten in einem wissensbasierten Controlling-Leitstand, Mai 1991
- Heft 78: H. Heß: Vergleich von Methoden zum objektorientierten Design von Softwaresystemen, August 1991
- Heft 79: A.-W. Scheer: Konsequenzen für die Betriebswirtschaftslehre aus der Entwicklung der Informations- und Kommunikationstechnologien, Mai 1991
- Heft 80: G. Keller, J. Kirsch, M. Nüttgens, A.-W. Scheer: Informationsmodellierung in der Fertigungssteuerung, August 1991
- Heft 81: A.-W. Scheer: Papierlose Beratung - Werkzeugunterstützung bei der DV-Beratung, August 1991
- Heft 82: C. Berkau: VOKAL (System zur Vorgangskettendarstellung und -analyse) - Struktur der Modellierungsmethode - Juni 1991 (wird nicht verlegt)
- Heft 83: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - Theoretical Foundations - 1991
- Heft 84: A. Hars, R. Heib, Ch. Kruse, J. Michely, A.-W. Scheer: Concepts of Current Data Modelling Methodologies - A Survey - 1991
- Heft 85: W. Hoffmann, M. Nüttgens, A.-W. Scheer, St. Scholz: Das Integrationskonzept am CIM-TTZ Saarbrücken (Teil 1: Produktionsplanung), Oktober 1991
- Heft 86: A.-W. Scheer: Koordinierte Planungsinself: Ein neuer Lösungsansatz für die Produktionsplanung, November 1991

Heft 87: M. Nüttgens, G. Keller, A.-W. Scheer, S. Stehle: Konzeption hyperbasierter Informationssysteme, Dezember 1991