

NR. 4

**Heinz Lothar Grob**  
**Frank Bensberg**  
**Stefan Bieletzke**

**Hypertext**

INSTITUT FÜR WIRTSCHAFTSINFORMATIK DER WESTFÄLISCHEN WILHELMS-UNIVERSITÄT MÜNSTER  
GREVENER STR. 91, 48159 MÜNSTER, TEL. (0251) 83-9752, FAX. (0251) 83-9754  
EMAIL GROB@WI.UNI-MUENSTER.DE

Oktober 1995

# **INHALT**

<b>1 Der kognitive Lernprozeß</b>	1
<b>2 Das Hypertext-Paradigma</b>	3
<b>3 Realisierung des Hypertext-Paradigmas</b>	5
3.1 Entwicklungsgeschichte	5
3.2 Architektur von Hypertextsystemen	7
3.3 Der Entwurf von Hypertext-Dokumenten	10
<b>4 Realisation von Hypertext-Projekten</b>	13
<b>5 Modifikation von Hypertext-Dokumenten</b>	20
<b>Literatur</b>	22

# 1 Der kognitive Lernprozeß

Trotz neuerer Entwicklungen auf dem Gebiet der Multimediatechnologie, bei der Fotos, Videos, Sprache, Geräusche und Musik zur Verbesserung von Lernprozessen, auf der technologischen Plattform *Computer* präsentiert werden, kommt dem *Text* zur Dokumentation und Erarbeitung von Wissen nach wie vor eine herausragende Bedeutung zu. Im CAL-Konzept werden Textdokumente als Basis angesehen, in die spezielle *Objekte* (z.B. Modelle für Berechnungsexperimente) einzufügen sind. Allerdings wird der Text nicht in linearer Form abgebildet, sondern im Hinblick auf den kognitiven Lernprozeß des Benutzers als Hypertext aufbereitet. Unter Hypertext ist eine netzartige Verknüpfung von Informationselementen zu verstehen, so daß der kognitive Lernprozess eines Individuums wirkungsvoll unterstützt werden kann.

Der Vorgang des Lernens sei nun am Beispiel des Lesens wissenschaftlicher Abhandlungen näher beschrieben.

Das Ziel des Lernens besteht im Erwerb von Wissen und letztendlich in einer nutzbringenden Veränderung des Verhaltens. Die Psychologie beschreibt den Lernprozeß auf der Basis eines kognitivistischen Ansatzes, bei dem *Informationsverarbeitungsprozessen* des Individuums besondere Aufmerksamkeit gewidmet wird.

Das Erarbeiten von Wissen beim Lesen eines Buches setzt die Existenz von drei semiotischen Ebenen voraus. Auf der *syntaktischen* Ebene findet die Zusammensetzung einzelner Zeichen zu sinnvollen, definierten Aggregaten - also zu Wörtern und Sätzen - statt. Zentraler Gegenstand der Syntaktik ist dabei die Repräsentation der Zeichen auf dem Zeichenträger. Die inhaltliche Bedeutung der sprachlichen Aggregate wird erst auf der *semantischen* Ebene relevant. Die Semantik untersucht dabei die Beziehung der Zeichen zu den Objekten, für die sie stehen können.<sup>1</sup> Schließlich erfährt der semantische Ausdruck eine *pragmatische* Interpretation durch den situativen Kontext. Hierbei bringt der Leser die Information mit seinem vorhandenen Wissen in Verbindung. Dieser Vorgang wird als *kohärente mentale Repräsentation* des Textes bezeichnet.<sup>2</sup>

Die Art der Repräsentation eines Textes auf einem Medium liefert die für den Lernprozeß wichtigen Ankerpunkte zur *Navigation* im subjektiven Wissensraum. Das Erlernen neuer Ankerpunkte vollzieht sich dabei durch den Transfer der gelesenen Informationseinheiten („Chunks“) vom Kurzzeit- ins Langzeitgedächtnis. Dabei wird eine Verbindung zwischen der

---

<sup>1</sup> Handelt es sich bei den Zeichen um Wörter, so wird dabei die Beziehung des Stehens-für-etwas (*aliquid stat pro aliquo*) in besonderem Ausmaß erfüllt. Vgl. Rodi, F. (1992).

<sup>2</sup> Vgl. Haake, J. M., Hannemann, J., Thüning, M. (1991), S. 121.

neuen Informationseinheit, die im Kurzzeitgedächtnis gespeichert ist, und den bereits gespeicherten Informationen des Langzeitgedächtnisses aufgebaut. Als Ergebnis dieses Vorgangs wird entweder ein neues *semantisches Netz* gebildet oder ein existierendes semantisches Netz modifiziert.

In Abb. 1 wird zur Veranschaulichung ein einfaches semantisches Netz mit ausschließlich linearen Beziehungen abgebildet.

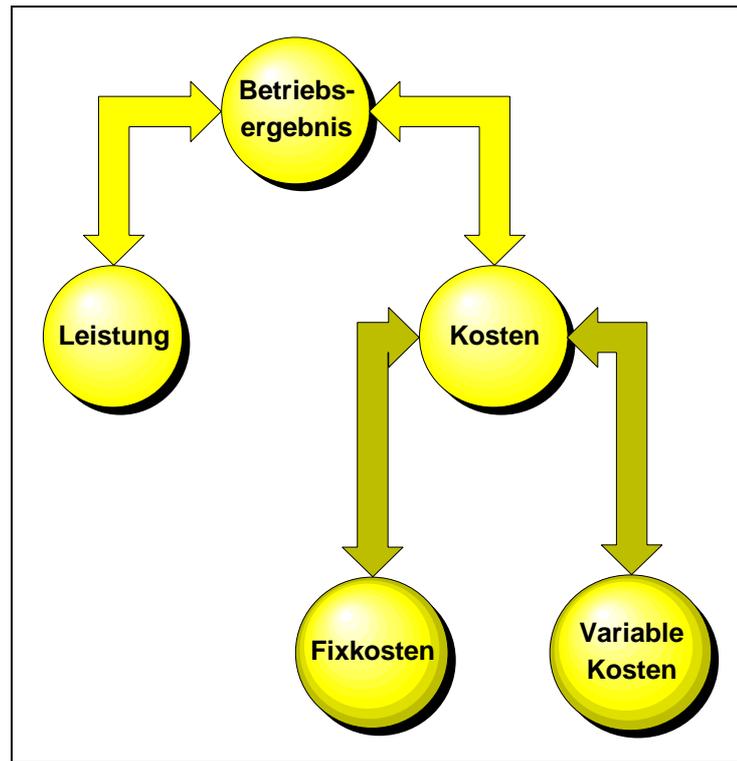


Abb. 1: Darstellung eines semantischen Netzes

Angenommen, die Termini „Fixkosten“ und „Variable Kosten“ seien zu erlernen. Erfolgt ein Transfer dieser beiden Informationseinheiten in das Langzeitgedächtnis, so wird der Lernende diese Informationen im Kontext mit dem bereits gespeicherten Begriff „Kosten“ verankern. Erfolgt anschließend durch die Kognition<sup>1</sup> des Lernenden eine Aktivierung der Informationseinheit „Betriebsergebnis“, können die Begriffe „Leistung“, „Kosten“ und anschließend auch die neu erlernten Termini „Fixkosten“ und „Variable Kosten“ aktiviert und vom Lernenden in anderen Kontext verwendet werden. Dieses einfache Beispiel soll im folgenden als Ausgangsbasis zum Verständnis des Hypertext-Paradigmas herangezogen werden.

---

<sup>1</sup> Der Begriff der Kognition bezieht sich auf alle Prozesse, durch die Wahrnehmungen verarbeitet, gespeichert und verwendet werden. Vgl. Zimbardo, P.G. (1992), S. 235.

## 2 Das Hypertext-Paradigma

Die Erarbeitung sich permanent ändernder, komplexer und vernetzter Informationsbestände ist mit klassischen Medien, in denen Wissen in linearer Form repräsentiert wird, nur schwer möglich. Aus diesem Grunde liegt es nahe, *Konformität* zwischen der Wissenspräsentation *durch das Medium* und der *Wahrnehmung des Mediums* durch den Lernenden bzw. den Lehrenden zu schaffen. Auf diese Weise sollen umfangreiche vernetzte Wissenskomplexe effizienter zugänglich gemacht werden.

Um hinsichtlich der Wissenspräsentation eine möglichst hohe Annäherung an das kognitive Lernmodell zu erzielen, wurde ein Ansatz zur *nicht-linearen Textgestaltung* entwickelt, der unter dem Begriff „Hypertext“ diskutiert wird. Das Modell des Hypertext besteht dabei aus Knoten, die zueinander in einer semantischen Beziehung stehen. Ein Knoten steht für eine einzelne „Idee“, die Verweise zu anderen Knoten enthalten können. Derjenige Knoten, von dem ein solcher Verweis ausgeht, wird dabei als *Referenz*, der Empfängerknoten als *Referent* bezeichnet. Hinsichtlich des oben angeführten Beispiels läßt sich der Zusammenhang zwischen dem Referenz- und den Referentenknoten wie folgt verdeutlichen:

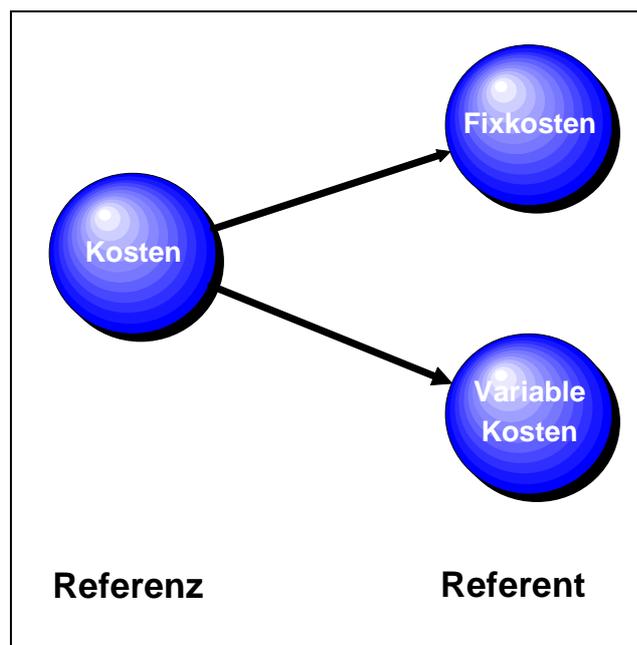


Abb. 2: Struktur von Hypertexten

Die dargestellte Konzeption erlaubt auch die Konstruktion *rekursiver Beziehungen*<sup>1</sup> zwischen mehreren Knoten. So ist eine komplexe Idee meist nicht atomarer Natur, sondern umfaßt weitere Ideen, die mit ihr vernetzt sind. Hypertexte müssen daher so strukturiert werden, daß die Zusammenfassung einzelner Ideen zu komplexen Knoten für den Lernenden nachvollziehbar ist. Wesentlich ist somit, daß zusammengesetzte Knoten („composite nodes“) generiert werden können. Die rekursive Strukturierung erlaubt die Einbettung des Beispiels in einen weiteren Knoten - den Meta-Knoten. In dem hier dargestellten Beispiel heißt der Meta-Knoten „Kostentheorie“. Dabei herrscht zwischen dem Meta-Knoten und den einzelnen Sub-Knoten eine implizite Teilmengenbeziehung (vgl. Abb. 3).

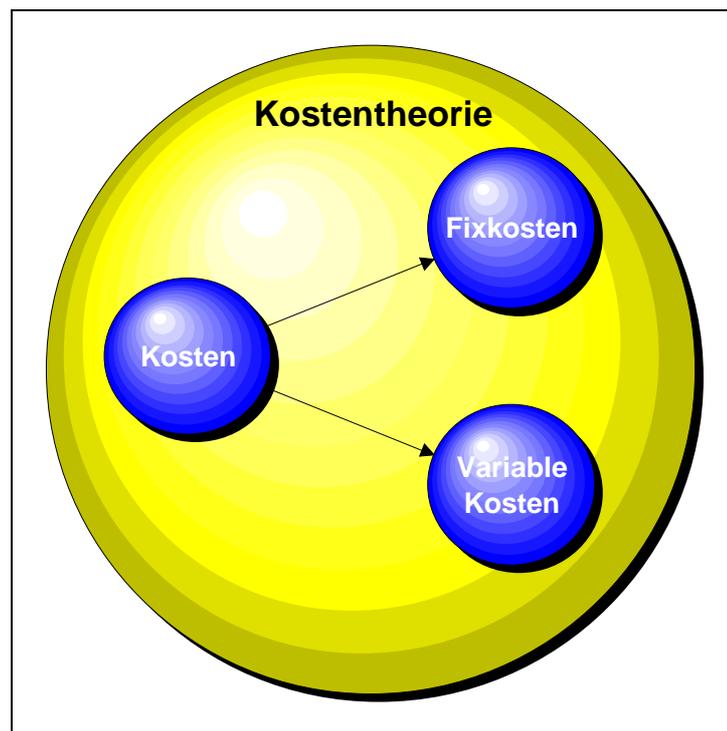


Abb. 3: Das Prinzip der rekursiven Strukturierung

Die *semantische Bedeutung* eines Verweises ist von dem abzubildenden Wissenskomplex abhängig. Aus dem Gebiet der Logik können allgemeine Relationen zwischen Objekten abgeleitet werden, die zur Formulierung eines konkreten Sachverhaltes dienen. Ein Katalog generischer Verweistypen wird in der folgenden Tabelle wiedergegeben:

---

<sup>1</sup> Rekursion bezeichnet dabei die Darstellung bzw. Definition eines Problems durch sich selbst.

<b>Verweistyp</b>	<b>repräsentierte Relation (A, B)</b>
Mengenzugehörigkeit	A ist in B enthalten
Demonstration	A ist ein Beispiel von B
Ursache	A verursacht B
Verwendung	A verwendet B
Besitz	A besitzt B
Enthaltensein	A ist Teil von B
Konsequenz	A folgt aus B
Ähnlichkeit	A ist ähnlich zu B

Abb. 4: Katalog generischer Verweistypen

### 3 Realisierung des Hypertext-Paradigmas

#### 3.1 Entwicklungsgeschichte

Die Idee einer nicht-linearen Verknüpfung von Texten, für die sich im Laufe der Zeit die Bezeichnung „Hypertext“ herauskristallisiert hat, wurde bereits in kulturgeschichtlichen Werken angewendet.<sup>1</sup> Die Konzeption eines Hypertext-Systems wurde erstmals im Jahre 1945 von Bush unter dem Namen „Memex“ vorgestellt. Dieses System sollte dem Zweck dienen, über assoziative Dokumentenverknüpfungen den Informationsbedarf, der auf einer rasch ansteigenden Anzahl wissenschaftlicher Veröffentlichungen basierte, zu bewältigen. Die auf Mikrofilmtechnologie basierende Memex-Konzeption wurde aufgrund technischer Probleme allerdings nie implementiert.<sup>2</sup>

Aufgrund der Verbreitung standardisierter Personal Computer (PC) steht für die Implementierung von Hypertextsystemen heute eine ausgereifte Technologie zur Verfügung. So besitzen

---

<sup>1</sup> So kann beispielsweise der Talmud mit seinen verschachtelten Anmerkungen und Kommentaren als altertümlicher Prototyp eines nicht-linearen Textsystems angesehen werden. Vgl. Balasubramanian, V. (1994).

<sup>2</sup> Vgl. Bush, V. (1945).

PCs die Möglichkeit, Daten kostengünstig zu speichern und durch entsprechende Algorithmen in aufbereiteter Form dem Lernenden zur Verfügung zu stellen. So kann ein Hypertext unter einer fensterorientierten Benutzeroberfläche, wie Microsoft Windows, folgendermaßen abgebildet werden:

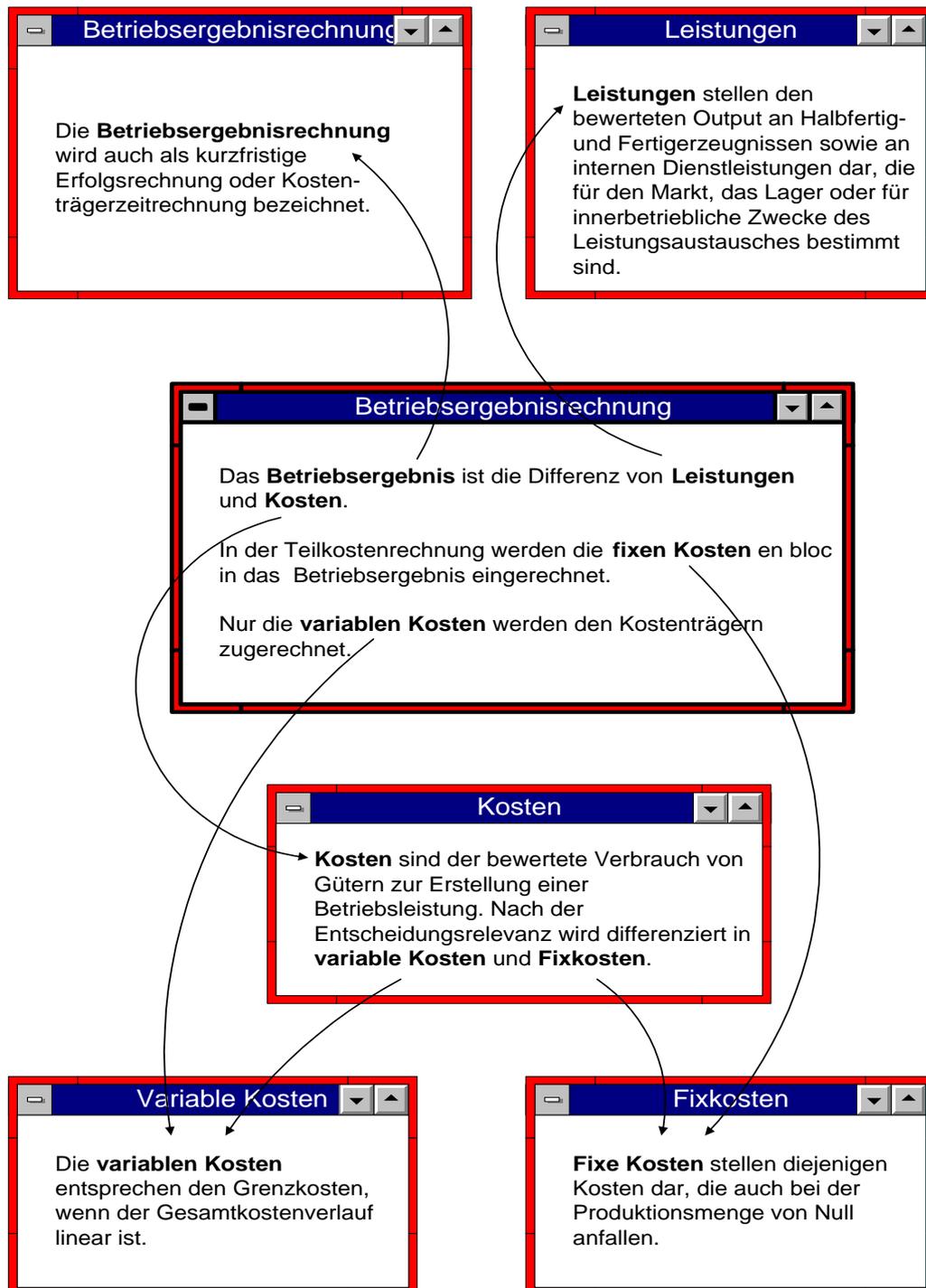


Abb. 5: Beispiel zur Gestaltung von Hypertexten

Die Markierung der existenten Verweise erfolgt durch eine textliche Gestaltung. Eine besondere Explikation des Typs der semantischen Beziehungen, wie in Abb. 4 systematisiert, ist jedoch hierbei nicht möglich.

Hypertext-Systeme weisen im Vergleich zu herkömmlichen Informationssystemen bestimmte Merkmale auf. Aus der oben angeführten Struktur wird deutlich, daß Informationen netzartig verknüpft werden können und nicht nur linear oder hierarchisch organisiert sein müssen. Neben dieser strukturellen Dimension müssen Hypertext-Systeme dem Anwender die Navigation durch den Hypertext erlauben. Für den Benutzer sind deshalb bestimmte - möglichst einfach zu erlernende - Operationen zur Navigation bereitzustellen. Zur Visualisierung bietet sich dabei insbesondere die grafische Präsentationsform an.

Bei der Realisierung von Hypertexten findet zunehmend eine Abkehr von rein textuellen Ansätzen statt. Konsequenterweise wird eine Integration heterogener Datentypen verfolgt. So findet bei multimedialen Anwendungen die Vernetzung von Text, Tabelle, Grafik, Fotografie, Video und Ton statt. Die Synthese von Hypertext und multimedialen Objekten wird als HyperMedia-System bezeichnet. Derartige Systeme bilden den Schwerpunkt der CAL+CAT-Projekte.

### **3.2 Architektur von Hypertextsystemen**

Die Architektur von Hypertextsystemen kann in einem Top-Down-Ansatz wie folgt dargestellt werden. Anzumerken ist, daß beim CAL+CAT-Konzept mit Anwendern Dozenten und Studenten gemeint sind.

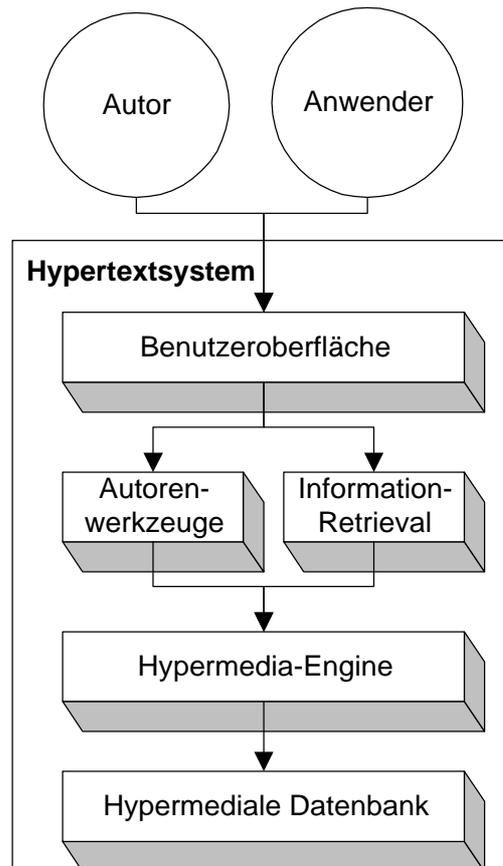


Abb. 6: Architektur von Hypertextsystemen

Die einzelnen Elemente des Hypertextsystems führen dabei die folgenden Funktionen aus:

### *(1) Benutzeroberfläche*

Die Benutzeroberfläche stellt das Erscheinungsbild eines Anwendungsprogrammes dar und muß die Ein- und Ausgabe des Systems in geeigneter Form präsentieren. Dies gilt sowohl für die Entwicklung als auch für die Anwendung des Hypertext-Systems.

Bei Hypertextsystemen werden besonders hohe Qualitätsanforderungen an die Benutzeroberfläche gestellt, da die Gefahr besteht, daß sich der Anwender im Informationsraum verliert<sup>1</sup> oder daß die sog. kognitive Last bei der Navigation zu groß wird. Die Gefahr der Desorientierung ist um so größer, je komplexer die zu recherchierenden Informationen sind. Einige Instrumente zur Behebung dieser Orientierungsproblematik finden dabei in neueren Programmsystemen Anwendung:

---

<sup>1</sup> Dieser Sachverhalt wird in der Literatur häufig als „Getting lost in hyperspace“ bezeichnet.

- Grafische Bäume und Netze zur Darstellung des Hypertext-Netzwerkes,
- Vorgabe von Informationspfaden und
- Rückwärtstraversierung des Navigationspfades („Backtracking“).

Grafische Bäume und Netze werden schwerpunktmäßig in modernen Entwicklungsumgebungen eingesetzt, um Beziehungen zwischen einzelnen Systemkomponenten darzustellen.<sup>1</sup> Die Vorgabe von Informationspfaden und die Rückwärtstraversierung werden häufig bei Lern- und Hilfesystemen neuerer Anwendungsprogramme genutzt. Die Rückwärtstraversierung erlaubt dabei die Wiederholung der aktuell bearbeiteten Informationen.

## **(2) Autorenwerkzeuge**

Autorenwerkzeuge dienen als Produktionsumgebung der Entwicklung und Pflege der Wissensdatenbank. Dabei ist das Problem zu lösen, daß heterogene Datentypen (wie Text, Grafik, Ton und Video) vernetzt werden müssen. Das Autorenwerkzeug muß daher hinsichtlich des Imports dieser Datentypen in das Hypertext-System eine ausreichende Offenheit besitzen.

## **(3) Information Retrieval (IR)**

Zur komfortablen Unterstützung des Benutzers bei der Anwendung von Hypertextsystemen gehört eine ausgereifte Abfrage- bzw. Suchtechnik. Einfache Navigationstechniken über grafische Browser<sup>2</sup> können in komplexen Informationsräumen zu einer Desorientierung des Nutzers führen. Bei der Implementierung von Abfrage- und Suchtechniken besteht die Möglichkeit einer Kombination von inhalts- und strukturorientierten Suchstrategien. Die inhaltsorientierte Suche erlaubt die Lokalisierung zusammenhängender Informationsknoten, wogegen die strukturorientierte Suche nach formalen Gliederungsaspekten der Wissensdatenbank durchgeführt wird. Voraussetzung für diese Art des Information Retrieval ist das Vorhandensein eines permanent zu administrierenden Index.

---

<sup>1</sup> Diese Technik wird schwerpunktmäßig im Bereich der objektorientierten Programmiersprachen (z. B. Smalltalk, C++) verwendet, um Klassenabhängigkeiten darzustellen.

<sup>2</sup> Als Browser wird dabei ein Instrument verstanden, das dem Anwender komplexe Wissensdatenbanken schematisiert und zusammenhängend darstellt und über Eingaben die Navigation erlaubt.

#### ***(4) HyperMedia-Engine***

Die Aufgabe der HyperMedia-Engine besteht darin, die Verweise zwischen den einzelnen Knoten zu verwalten. Dazu gehört die Definition der Verweise durch den Autor als auch die Darstellung der Verweise bei einer Präsentation der einzelnen Knoten. Ein besonderes Problem besteht in der Synchronisation von Verweisen. So sind beispielsweise Videosequenzen mit gleichzeitiger akustischen Untermalung in Einklang zu bringen.

#### ***(5) Hypermediale Datenbanken***

Die Aufgabe hypermedialer Datenbanken ist die Verwaltung und Bereitstellung aller in einem Hypertext-System repräsentierten Medien. Dabei sind folgende Anforderungskriterien an das Datenbanksystem zu formulieren:

- Offenheit in Bezug auf den Datenaustausch („Datenimport- und -export“),
- geografische Verteilung der Datenbestände („Fragmentierung“),
- Bedienung der Abfragetechniken des Information Retrieval („Query“) und das
- Management multimedialer Datentypen („Integration“).

### **3.3 Der Entwurf von Hypertext-Dokumenten**

Bei dem Entwurf von Hypertext-Dokumenten steht der Autor in der Regel vor dem Problem, eine Vielzahl von Quellen mit einer einheitlichen Systematik zu erfassen und entsprechend aufzubereiten. Dabei werden verschiedene Phasen durchlaufen, die hier in Form eines Wasserfallmodells abgebildet werden:<sup>1</sup>

---

<sup>1</sup> Die Phaseneinteilung erfolgt in Anlehnung an die drei Felder der klassischen Rhetorik: *Inventio*, *Dispositio* und *Elocutio*. Dabei wird die Phase *Dispositio* durch zwei Teilphasen zur Auswahl und Ordnung repräsentiert. Vgl. dazu Haake, J. M., Hannemann, J., Thüring, M., (1991), S. 119 ff.

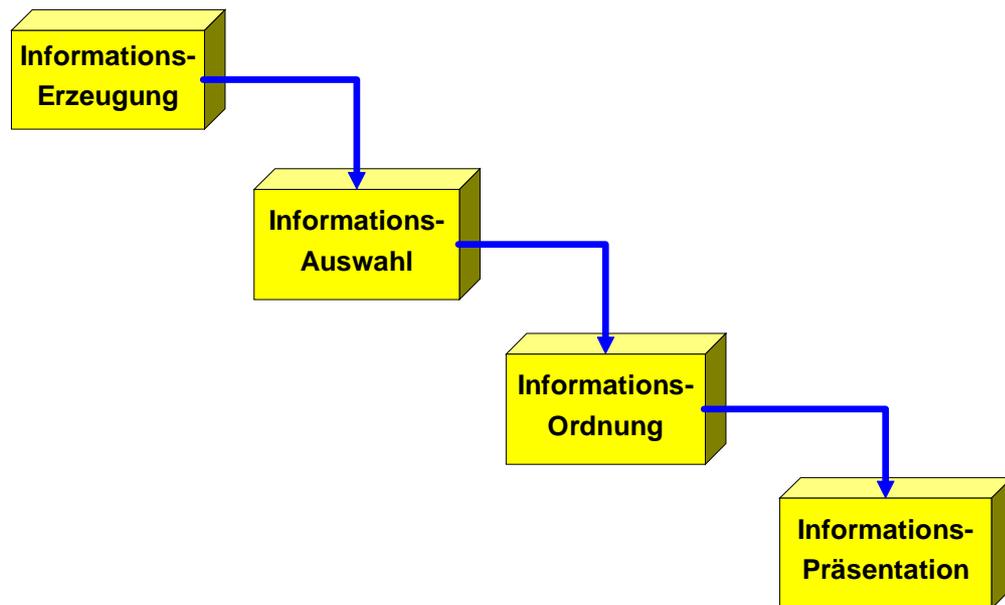


Abb. 7: Phasenkonzept beim Entwurf von Hypertext-Dokumenten

In den beiden ersten Phasen der *Informationserzeugung* und *Informationsauswahl* muß sich die Entwicklung an der Inhaltskomponente des Hypertext-Systems orientieren. Diese inhaltliche Zielorientierung umfaßt die folgenden Dimensionen:

- Definition und Segmentierung der Zielgruppe(n),
- Definition der zu vermittelnden Wissenskomplexe nach Inhalt und Umfang und
- Bestimmung des zeitlichen Horizontes für die Wissensvermittlung.

In der nachgelagerten Phase der *Informationsordnung* erfolgt die Hierarchisierung der selektierten Informationen. Das Produkt dieser Phase ist ein ungeordnetes Netz oder ein hierarchischer Baum mit einer impliziten Sequenzierung. Die Entscheidung über die Organisation des Wissens ist dabei abhängig von der vorgelagerten Zielorientierung. So ist bei einer gering motivierten Zielgruppe die Sequenzierung der Lerneinheiten vorzugeben, wogegen bei höher motivierten Zielgruppen auch die freie Exploration über ungeordnete Netze sinnvoll sein kann. Beim CAL+CAT-Projekt wird grundsätzlich von der zweiten Variante ausgegangen und der Eigenschaft der Optionalität bei der Entwicklung von Software eine hohe Priorität eingeräumt.<sup>1</sup>

---

<sup>1</sup> Grob, H. L. (1995).

In der anschließenden Phase der *Informationspräsentation* wird die Aufgabe wahrgenommen, das Hypertext-Netz darzustellen. Dabei sind verschiedene Modelle zu unterscheiden. So kann zum einen die rein *textuelle* bzw. *knotenzentrierte Darstellung* gewählt werden, bei der für den aktuellen Knoten der Inhalt und alle ausgehenden und endenden Verweise präsentiert werden. In diesem Modus erfährt der Leser nur einen begrenzten Ausschnitt aus dem Hypertext-Dokument, da kein Überblick auf der Netzebene verfügbar ist. Im folgenden Beispiel wird dieser Darstellungsmodus wiedergegeben:

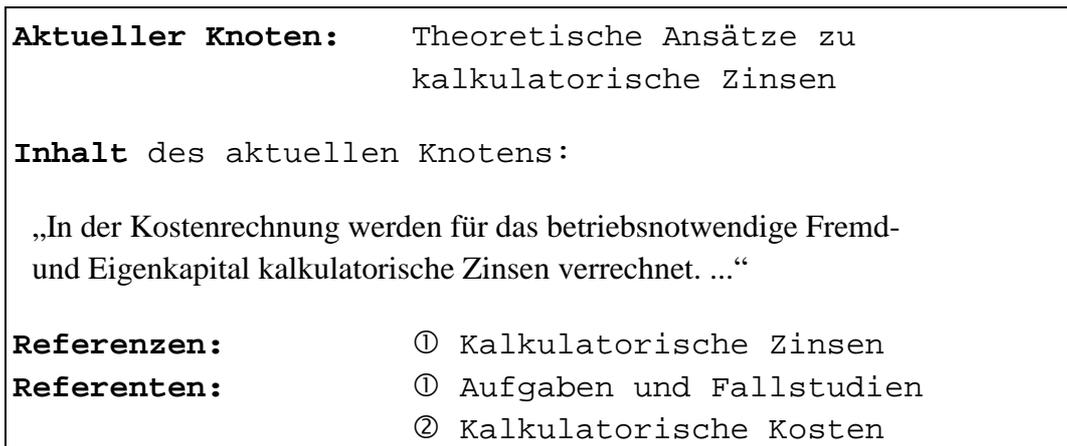


Abb. 8: Textuelle Darstellung von Hypertexten

Wird dagegen die *grafische Darstellung* gewählt, so erfolgt die Visualisierung des Teilnetzes durch Visualisierung der Umgebung des aktuellen Knotens. Diese Variante wird in Abb. 9 dargestellt:

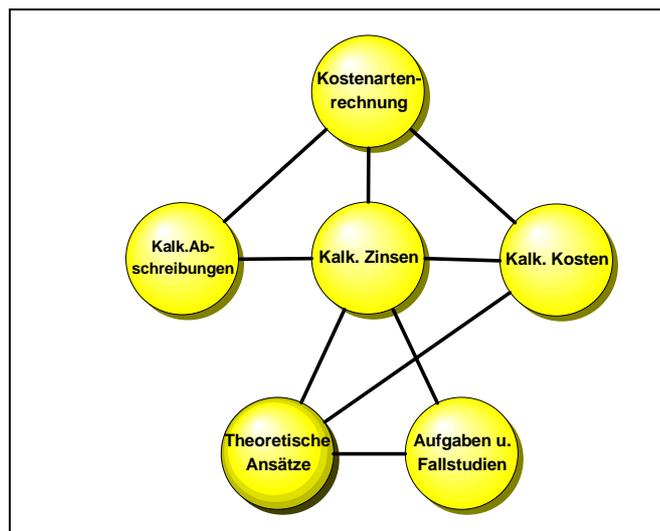


Abb. 9: Grafische Darstellung von Hypertexten

Schließlich ist auch eine Kombination textueller und grafischer Darstellungsarten möglich, wobei dem Nutzer ein einfacher Wechsel zwischen den beiden Varianten erlaubt sein sollte.

## 4 Realisation von Hypertext-Projekten

Die Leistungsfähigkeit bei der Realisation des Hypertext-Paradigmas ist von der Anwendungs- und Entwicklungsumgebung abhängig. Die Anwendungsumgebung definiert mit den technischen Voraussetzungen die Mindestausstattung eines Nutzers bezüglich Hardware und Software, um die HyperMedia-Software nutzen zu können. Neben diesem technischen Kompatibilitätsaspekt ist der persönliche Kompatibilitätsaspekt zu berücksichtigen, d.h. insbesondere die Oberfläche der Anwendung darf keinen hohen Einarbeitungsaufwand vom Benutzer verlangen. Je höher die Kompatibilitäten, desto höher wird die Akzeptanz und letztlich die Diffusion der Software prognostiziert.<sup>1</sup>

Die Entwicklungsumgebung für HyperMedia Software wird weitgehend durch die Anwendungsumgebung der Zielgruppe determiniert. Die Architektur der Entwicklerrechner sollte technisch abwärtskompatibel zu den Anwendungsrechnern sein, um aufwendige Portierungen zu vermeiden.<sup>2</sup> Die weiteren Ausführungen beziehen sich ausschließlich auf die Entwicklung von Hypertexten als Ausgangsobjekt für HyperMedia-Software. Dabei wird im folgenden das Arbeiten mit dem Windows-Help-Compiler HCP gezeigt.<sup>3</sup> Eine Übertragung der Bearbeitungsschritte auf andere Entwicklungsumgebungen ist möglich.

Im folgenden sollen der Arbeitsablauf bei der Realisierung eines Hypertext-Projektes kurz dargestellt werden. Vorbereitend ist eine Grobkonzeption zu erstellen, die Struktur und Inhalt der Informationen umreißt. Die Einrichtung der Programme muß zu einer funktionsfähigen Entwicklungsumgebung führen. Der Beginn der Hypertext-Produktion ist durch die Erstellung der Themendateien (topic files) gekennzeichnet. Die einzelnen Themen (topic) der Themendatei werden durch Sprünge (Jumps), Querverweise (cross-references) und temporäre Zusatzfenster (popups) vernetzt. Makros werden in die Themendatei und in die Projektdatei (project file) eingefügt, um spezielle Effekte, Hilfen oder Verknüpfungen zu realisieren. Parallel hierzu werden ergänzende Programme (application objects) und Grafiken (graphics) erstellt, die -

---

<sup>1</sup> Innerhalb des CAL-LKR-Projektes wird die Kompatibilität zur Zielgruppe mit der Wahl von 80x86 PC-DOS/Windows Rechnern mit einer typischen Windows Oberfläche angestrebt.

<sup>2</sup> Im CAL-LKR-Projekt werden Pentium DOS/Windows Rechner mit englischen Winword und Winword Druckformatvorlagen und DOS-Compilern verwendet.

<sup>3</sup> Im CAL-LKR-Projekt wird Winword als Editor verwendet, und zusätzlich als Entwicklungshilfe die Druckformatvorlage hypertrk.dot. Als Compiler steht HC505 zur Verfügung. Compiler und Druckformatvorlagen sind jeweils aktuell im Internet per FTP von ftp.microsoft.com zu beziehen.

ggf. mit HotSpots versehen - zur Verknüpfung bereitgestellt werden. Die Kompilation führt nach Maßgabe der Steuerbefehle der Projektdatei zu einer Verknüpfung der Themendatei mit den Grafiken und Objekten. Ergebnis der Kompilation ist ein Hypertext, der unabhängig von der Entwicklungsumgebung lauffähig ist.

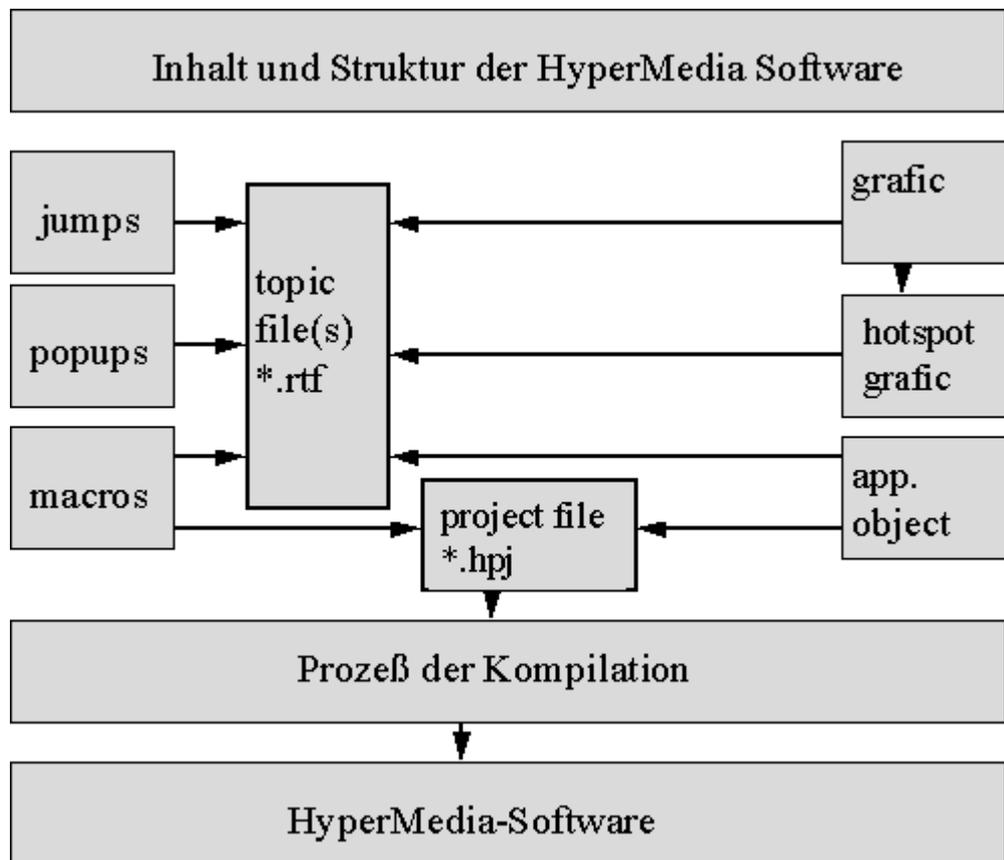


Abb. 10: Komponenten bei der Erstellung eines Hypertextes

### ***(1) Programme einrichten***

Die Entwicklungsumgebung für HyperMediaSkripte erfordert

- einen Texteditor für Rich Text Format RTF.
- einen Hilfecompiler

und optional

- eine Entwicklungshilfe

Ohne die Nutzung einer Entwicklungshilfe müßten die Verknüpfungen im RTF-Format programmiert werden. Beispielsweise würde der RTF-Syntax für einen Jump vom Text „Testtext“ zum topic „Fortsetzung“ in der Themendatei „thema.rtf“ wie folgt lauten:

```
{\uldb testtext}{\v {%|*}Fortsetzung{@thema.rtf}{>mainwindow}
```

Das Beispiel macht deutlich, daß die Verwendung einer Entwicklungshilfe die Erstellung eines Hypertextes stark beschleunigen kann, da diese im Gegensatz zu dem RTF-Format eine effiziente, gekoppelt textuell-grafische Entwicklung ermöglicht.<sup>1</sup>

## ***(2) Projekte definieren und bearbeiten***

Ein Projekt umfaßt eine thematisch zusammengehörende Einheit von Hilfedateien. Die Projektdatei ist eine ASCII-Steuerdatei für den Hilfecompiler. Sie faßt alle Informationen zusammen, die der Hilfecompiler benötigt, um die Hilfequelldateien in eine Hilfedatei umzuwandeln. Jedes Hilfeprojekt muß über genau eine Projektdatei verfügen. Der wichtigste Abschnitt der Projektdatei ist derjenige, der die Informationen mit den Namen der Themendateien beinhaltet, aus denen die Hilfedatei erstellt wird. Hinzu kommen Informationen, um die Konstruktion der Hilfedatei zu steuern sowie die Erscheinungsform und das Verhalten der Hilfeanwendung anzupassen und zu erweitern. Diese werden zu verschiedenen Abschnitten zusammengefaßt.

Das folgende Beispiel zeigt eine CAL-LKR-Projektdatei mit Kommentaren, die die einzelnen Abschnitte und Optionen näher erläutern:

```
[OPTIONS]
```

```
;REMARK: Projektverzeichnis ist C:\CAL\TEST
```

```
ROOT=C:\CAL\TEST
```

```
;REMARK: Bitmaps sind in den Projekt-Unterverzeichnissen \BMP1, \BMP2
```

```
BMROOT=C:\CAL\TEST\BMP1;C:\CAL\TEST\BMP2
```

```
;REMARK: Die Titelleiste lautet „Hilfebuch“
```

```
TITLE=Hilfebuch
```

```
;REMARK: Der Kontextstring des Einstiegsthemas lautet EINSTIEG
```

```
CONTENTS=EINSTIEG
```

---

<sup>1</sup> Im CAL-LKR-Projekt wurde durch die Verwendung von Druckformatvorlagen eine Beschleunigung der Hypertext Erstellung um Faktor 12 erreicht.

```
; REMARK:Als Symbol der Hilfedatei soll DEMO.ICO verwendet werden.  
ICON=C:\CAL\TEST\BMP1\DEMO.ICO  
; REMARK:Abschnitt mit der Lokalisation der Themendateien text.rtf und text2.rtf  
[FILES]  
C:\CAL\TEST\text.RTF;c:\cal\test\text2.rtf  
; REMARK: Abschnitt mit Anpassung der Benutzeroberfläche  
[CONFIG]  
;REMARK: Makro zur Definition einer zusätzlichen Schaltfläche  
CreateButton(„BTN_G“, „&Glossar“, „JumpContents(`GLOSSAR.HLP`)“)  
; REMARK:Blätter-Schaltflächen aktivieren  
BrowseButtons()  
; REMARK: Eigene Hilfe-über-Hilfe-Datei definieren  
SetHelpOnFile(„DEMO.HLP“)  
; REMARK: Sekundäres Hilfefenster definieren  
[WINDOWS]  
wingloss=„Glossar“, (123,123,256,256), 0, (0,255,255), (255,0,0)
```

### ***(3) Themen definieren und bearbeiten***

Ein Projekt kann eine oder mehrere Themendateien beinhalten. Jede Themendatei enthält eine oder mehrere Einzelthemen. Ein Einzelthema (topic) ist dabei die Informationseinheit, die dem Benutzer letztlich in einem Bildschirmfenster zu einem beliebigen Zeitpunkt präsentiert wird. Die Topics sind somit die kleinste Einheit der Präsentation.

Jedes Thema bekommt zwingend eine Definition über

- eine eindeutige Überschrift

und optional Zusatzinformationen über

- eine Seitenzahl für traditionelles lineares Blättern
- Schlagworte
- Makros
- Identifikationshilfen

Diese Definitionen werden in den Fußnoten eines jeden Themas verwaltet.

Durch die Angabe einer Seitenzahl könnte die Themendatei nun kompiliert werden, um ein lineares Äquivalent zum Medium „Buch“ zu erhalten. Die Vernetzung -also die Erzeugung eines Hypertextes - geschieht in einem nächsten Schritt durch die Definition von HotSpots.

#### ***(4) HotSpots definieren und bearbeiten***

Ein HotSpot ist ein Bereich in einem Thema, der dem Benutzer die Möglichkeit gibt, durch Anklicken in ein anderes Thema zu verzweigen. HotSpots können auf Texten oder Grafiken liegen und alternativ drei Aktionsarten ausführen:

- die Sprungverknüpfung, um ein altes topic durch ein neues topic zu ersetzen (jump)
- die temporäre Ansicht, um eine anderes topic zusätzlich anzuzeigen (popup)
- die Makroverknüpfung, um einen Makro auszuführen.

Neben der Intradateiverknüpfung sind auch Interdateiverknüpfungen zu anderen Hilfedateien oder ausführbaren Programmen möglich.

Die Erstellung eines Text-HotSpot unter Zuhilfenahme einer Entwicklungshilfe soll kurz skizziert werden: Nach der Markierung des Textes, der als „Absprung“-adresse dienen soll, wird die Sprungart per Mausclick gewählt, um anschließend eine Liste dargeboten zu bekommen, aus der man die „Einsprung“-adresse wählt. Wird ein Thema als „Einsprung“-adresse gewählt, so kann zwischen Jump und popup gewählt werden, wird hingegen eine Makroverknüpfung gewählt, so muß aus einer Makroliste gewählt werden und die Parameter sind syntaxgerecht zu ergänzen.

#### ***(5) Hilfemakros einfügen***

Hilfemakros oder kurz Makros sind Befehle, mit denen man das Standardverhalten des Hilfeprogramms ändern und/oder erweitern können. Mit Makros kann man u.a.

- die Funktion von Schaltflächen verändern
- neue Schaltflächen definieren
- Aussehen, Position und Verhalten der Hilfefenster kontrollieren
- zwischen verschiedenen Themen und Hilfedateien navigieren
- andere Anwendungen aufrufen
- den Funktionsumfang des Hilfeprogramms mit DLLs erweitern

Ein Hilfemakro besteht dabei aus einem Makronamen und einer Liste von Parametern, die in runde Klammern eingeschlossen sind. Die allgemeine Form eines Hilfemakros lautet:

Makroname(„parameter1“, „parameter2“, ...)

Der Makroname bezeichnet die Aktion, die ausgeführt werden soll, wenn der Makro aktiviert wird. Die Parameter bezeichnen die Objekte, die von der Aktion des Makros betroffen sind. Beispiele für solche Objekte sind Dateien, Themen, Menüs, Schaltflächen oder andere Makros. Zum Beispiel erstellt der folgende Makro eine neue Schaltfläche mit der Aufschrift „Glossar“:

```
CreateButton(„btn_glossar“, „&Glossar“)
```

Werden Makros zur Erweiterung des Hilfeprogramms eingesetzt, kann gewählt werden, wann und wo die Makros ausgeführt werden sollen. Im folgenden werden einige Beispiele aufgeführt.

- Makros beim Öffnen einer Hilfedatei ausführen
- Makros vor Anzeige eines Themas ausführen
- Makros von einem Menübefehl aus ausführen
- Makros von einer Schaltfläche aus ausführen
- Makros von einem Hotspot aus ausführen
- Makros von einem Grafik-Hotspot aus ausführen

In Abhängigkeit von der Art der Ausführung muß das Makro entweder in der Projektdatei, in den Fußnoten oder in einem Hotspot abgelegt sein. Eine Dokumentation der wichtigsten Makros kann die Funktionen innerhalb einer Hilfedatei gut verdeutlichen:

Makro	Funktion
Annotate	Dialogfeld „Anmerken“ anzeigen
Back	Vorhergehendes Thema in der Historie anzeigen
BrowseButtons	Blätter-Schaltflächen in Funktionsleiste einfügen
CloseWindow	Fenster schließen
Contents	Einstiegsthema anzeigen
CreateButton	Neue Schaltfläche in Funktionsleiste einfügen
DeleteItem	Menübefehl löschen
DestroyButton	Schaltfläche löschen
DisableButton	Schaltfläche deaktivieren
DisableItem	Menübefehl deaktivieren
EnableButton	Schaltfläche aktivieren
EnableItem	Menübefehl aktivieren
ExecProgram	Programm ausführen
Exit	Hilfeanwendung beenden
FocusWindow	Fokus auf ein Fenster richten
HelpOn	Hilfe für das Hilfeprogramm anzeigen
History	Liste „Hilfe - Bisherige Themen“ anzeigen
IfThen	Makro bedingt ausführen
JumpContents	Zum Einstiegsthema einer Hilfedatei springen
JumpContext	Zu einem bestimmten Kontext springen
JumpId	Zu einem bestimmten Thema springen
JumpKeyword	Zu einem bestimmten Stichwort springen
Next	Vorblättern
PopupContext	Thema über Kontextnummer in einem Popup-Fenster anzeigen
PopupId	Thema über Kontextstring in einem Popup-Fenster anzeigen
PositionWindow	Position und Größe eines Fensters festlegen
Prev	Zurückblättern
Print	Aktives Thema drucken
RegisterRoutine	DLL-Funktion registrieren
Search	Dialogfeld „Suchen“ anzeigen
SetContents	Thema einer Hilfedatei als Einstiegsthema definieren
SetHelpOnFile	Andere Hilfe-über-Hilfe-Datei definieren

Abb. 11: Wichtige Makrofunktionen

**(6) Kompilation**

Nach der Erstellung der Projektdatei, der Themendateien und dem Einfügen der HotSpots und Makros ist das Projekt zu kompilieren. Hierzu sind folgende Schritte notwendig:

- Starten einer DOS-Sitzung unter Windows
- Starten des Kompilers mit dem Pfad und Parameter der Projektdatei
- Anzeigen von Fehlern und Dokumentation in einer Datei (z. B. in der Datei test.err)
- Darstellung des Ergebnisses der Kompilation in einer Hilfedatei (z. B. test.hlp)

Der Hypertext kann nun unabhängig von der Entwicklungsumgebung verbreitet werden. Der Aufruf geschieht mit dem Windows-Programm Winhelp.

## 5 Modifikation von Hypertext-Dokumenten

Während bei Büchern und CD ROMs inhaltliche Änderungen dem Endbenutzer erst bei einer neuen Auflage sichtbar gemacht werden, bieten elektronisch gespeicherte Dokumente die Möglichkeit einer permanenten Modifikation. Dabei sind für Hypertext-Dokumente einige Besonderheiten zu berücksichtigen, die im folgenden als inhaltliche, funktionelle und strukturelle Modifikationen erörtert werden sollen.

Eine *funktionelle Modifikation* liegt vor, wenn in einem vorhandenen Hypertext die Steuerbefehle der Projektdatei oder der Themendatei geändert werden sollen, z. B. wenn ein Makro zum Programmaufruf eingefügt, geändert oder gelöscht werden soll. Die Struktur des Hypertext-Dokumentes bleibt durch eine funktionelle Modifikation unberührt. Der Modifikationsaufwand ist gering, da die Änderungen unter Einsatz des Texteditors und der optionalen Entwicklungshilfe durchgeführt werden können. Funktionelle Modifikationen in der Steuerdatei wirken sich auf das gesamte Hypertext-Dokument aus, Modifikationen in der Themendatei wirken sich nur auf das jeweilige Thema der Themendatei aus.

Eine *strukturelle Modifikation* liegt vor, wenn Themen der Themendatei gelöscht oder neu erstellt werden. Die *Löschung* eines kompletten Themas kann im Texteditor durchgeführt werden. Alle Verweise von anderen Themen auf das gelöschte Thema müssen ebenfalls gelöscht oder geändert werden. Dies erfordert den Aufruf der Suche mit dem Parameter „Name des gelöschten Themas“ im Texteditor und die anschließende Veränderung der gefundenen Verweise. Wenn das Hypertext-Dokument über die Map-Funktion mit einem ausführbaren Programm zur Navigation verbunden war, so muß im Map-Abschnitt der Steuerdatei das entsprechende Thema gelöscht werden. Die *Neuerstellung* eines Themas kann ebenfalls im Texteditor durchgeführt werden. Unter Zuhilfenahme der Entwicklungshilfe muß eine eindeutige Überschrift vergeben und in der Fußnote des Themas verankert werden. Bei Bedarf muß eine Blättersequenz hinzugefügt werden. Ein vorhandener Map-Abschnitt der Steuerdatei muß um das neue Thema ergänzt werden.

Eine *inhaltliche Modifikation* von Hypertext-Dokumenten liegt vor, wenn Textteile oder Grafiken in vorhandenen Themen der Themendatei zu verändern sind. Die inhaltliche Modifikation kann unter Nutzung des Texteditors durchgeführt werden. Sprünge werden unter Einsatz der Entwicklungshilfe neu erstellt.

Ein Sonderfall der inhaltlichen Modifikation liegt vor, wenn Änderungen gleichzeitig im Hypertext *und* einer traditionellen Textdatei durchgeführt werden sollen. Der Änderungsprozeß kann dabei entweder in der Textdatei oder im Hypertext ansetzen. Wenn keine dynamische Verknüpfung zwischen den Dokumenten besteht, so liegt ein doppelter Änderungsprozeß in zwei ähnlichen Dokumenten vor. Eine Änderung der Textdatei kann dann beispielsweise mit dem Texteditor Winword unter Nutzung der Überarbeitungsfunktion durchgeführt werden. Die Überarbeitungsfunktion ermöglicht es, Änderungen automatisch optisch zu markieren. Geänderte Textpassagen werden als „durchgestrichen“ markiert, neue Textpassagen werden als „unterstrichen“ markiert. Eine Änderung des Hypertextes kann dann erfolgen, indem die als „durchgestrichen“ markierten Passagen des Textdokumentes im Hypertext gesucht und manuell gelöscht werden bzw. die als „unterstrichen“ markierten Passagen im Hypertext ergänzt werden. Der Arbeitsaufwand für die doppelte Textänderung ist jedoch als relativ hoch einzustufen. Eine Möglichkeit zur Verringerung des Aufwandes besteht darin, die originären Änderungen im Hypertext zu erstellen und direkt aus dem Hypertext direkt die Textdatei zu generieren. Der Hypertext muß hierzu bei der Erstellung linear aufgebaut worden sein, d.h. über die linearen Blättersequenzen muß sich ein inhaltlich-logischer Textaufbau ergeben. Für CAL+CAT-Projekte ist die Organisation zur Modifikation von Hypertext-Dokumenten effizient zu gestalten, da gerade im Lehr-/Lernbereich häufig Verbesserungen funktioneller, struktureller und inhaltlicher Art umzusetzen sind. Selbst wenn die Themenbereiche der „herrschenden Lehre“ zuzuordnen sind, fallen doch unter didaktischen Aspekten aufgrund von Erfahrungen und Rückkopplungen zu den Studenten permanent Modifikationen an.

## Literatur

- Balasubramanian, V., State of the Art Review on Hypermedia Issues And Applications, 1994, <ftp.isg.sfu.ca/pub/duchier/HT.tar.gz>.
- Bush, V. (1945), As we may think, in: The Atlantic Monthly, Juli 1945.
- Engel, E (1994), Elektronisch publizieren mit dem Microsoft Help Compiler, Microsoft Press, Unterschleißheim 1994.
- Grob, H. L. (1995), CAL+CAT, Arbeitsbericht Nr. 2 der Reihe CAL+CAT, Münster 1995.
- Haake, J. M., Hannemann, J., Thüring, M. (1991), Ein Ansatz zur Organisation von Hyperdokumenten, in: Hypertext/Hypermedia 1991 Proceedings, S. 119-134.
- Rodi, F., Stichwort Semiotik, in: Handlexikon der Wissenschaftstheorie, Hrsg. : H. Seiffert, G. Radnitzky, München 1992, S. 296-301.
- Zimbardo, P. G. (1992), Psychologie, 5. Auflage, Berlin u. a. 1992.

## **Arbeitsberichte der Reihe CAL+CAT**

- Nr. 1 Grob, H. L., Griebhaber, W., Computergestützte Lehre an der Universität, Arbeitsbericht Nr. 1, Münster 1995.
- Nr. 2 Grob, H. L., CAL+CAT, Arbeitsbericht Nr. 2, Münster 1995.
- Nr. 3 Grob, H. L., Bensberg, F., Multimedia, Arbeitsbericht Nr. 3, Münster 1995.
- Nr. 4 Grob, H. L., Bensberg, F., Bielezke, S., Hypertext, Arbeitsbericht Nr. 4, Münster 1995.
- Nr. 5 Grob, H. L., Seufert, S., Vorgehensmodelle bei der Entwicklung von CAL-Software, Arbeitsbericht Nr. 5, Münster 1995 (in Vorbereitung).