

Diplomarbeit

# Navigation in hypermedialen Systemen

Eine Betrachtung anhand des  
Industriemeisters 2000 und des  
Altenberger Doms

**Klaus Kneupner**

Diplomarbeit  
am Fachbereich Informatik  
der Universität Dortmund

28. September 1999

**Gutachter:**

Prof. Dr. E.-E. Doberkat  
Dipl.-Inform. K. Alfert

Universität Dortmund  
Lehrstuhl für Softwaretechnologie  
Fachbereich Informatik  
D-44221 Dortmund

---

**Zusammenfassung:** Als „Navigation in hypermedialen Systemen“ wird im Allgemeinen nur als das Folgen von Verknüpfungen verstanden. Dabei bieten moderne hypermediale Systeme zusätzlich die Möglichkeit, mit Hilfsmitteln wie „Lesezeichen“ oder „Historie“ zu navigieren. Ausgehend von der Analyse zweier hypermedialer Anwendungen werden allgemeine Eigenschaften von Hilfsmitteln der Navigation bestimmt. Eine Erweiterung des Dexter Modells für die Modellierung von Hilfsmitteln wird vorgestellt. Aufbauend auf dieser Erweiterung sowie auf den Hilfsmitteln wird abschließend eine Definition der Navigation erstellt.

**Schlüsselwörter:** Hypermedia, Navigation, Dexter Modell, Hilfsmittel der Navigation.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Kontext . . . . .	1
1.2	Aufgabenstellung . . . . .	1
1.3	Aufbau der Arbeit . . . . .	2
1.4	Danksagung . . . . .	3
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Hypermediale Systeme . . . . .	5
2.1.1	Hypertext . . . . .	5
2.1.2	Multimedia . . . . .	6
2.1.3	Hypermedia . . . . .	7
2.1.4	Adaptive hypermediale Systeme . . . . .	8
2.2	Modelle für hypermediale Systeme . . . . .	9
2.2.1	Dexter Hypertext Referenzmodell . . . . .	9
2.2.2	Amsterdamer Hypermediamodel (AHM) . . . . .	16
2.2.3	Modell adaptiver Hypermediaanwendungen (AHAM) . . . . .	18
2.2.4	Bewertung der Modelle . . . . .	19
2.3	Strukturen der Datenbasis . . . . .	20
2.4	Navigation . . . . .	22
2.4.1	Die Navigationsmetapher . . . . .	23
2.4.2	Desorientierung . . . . .	24
2.4.3	Hilfsmittel der Navigation . . . . .	24
2.4.4	Arbeitshypothese der Navigation . . . . .	25

<b>3 Die hypermediale Industriemeister 2000 Anwendung</b>	<b>27</b>
3.1 Projektbeschreibung . . . . .	27
3.1.1 Zielsetzung des rechnergestützten Lernsystems . . . . .	28
3.1.2 Die Realisierung . . . . .	29
3.2 Die Datenbasis . . . . .	32
3.3 Das hypermediale System . . . . .	34
3.3.1 Adaption . . . . .	34
3.3.2 Die Realisierung der Schichten des Dexter Modells . . . . .	35
3.3.3 Nicht unterstützte Merkmale des Dexter Modell . . . . .	36
3.3.4 Navigationsschritte . . . . .	36
3.4 Hilfsmittel der Navigation . . . . .	37
3.4.1 Die Beschreibungsform für Hilfsmittel . . . . .	38
3.4.2 Geführte Touren . . . . .	38
3.4.3 Glossar . . . . .	41
3.4.4 Historie . . . . .	43
3.4.5 Inhaltsverzeichnis . . . . .	44
3.4.6 Lesezeichen . . . . .	46
3.4.7 Reihenfolge . . . . .	48
3.4.8 Startauswahl . . . . .	50
3.4.9 Titelleiste . . . . .	51
<b>4 Die hypermediale Altenberger Dom Anwendung</b>	<b>53</b>
4.1 Projektbeschreibung . . . . .	53
4.1.1 Zielsetzung . . . . .	54
4.1.2 Derzeitiger Entwicklungsstand . . . . .	54
4.2 Die Datenbasis . . . . .	57
4.3 Das hypermediale System . . . . .	58
4.3.1 Die virtuelle ADML Maschine . . . . .	59
4.3.2 Die Schichten des Dexter Modells . . . . .	60
4.3.3 Nicht unterstützte Merkmale des Dexter Modell . . . . .	60
4.3.4 Navigationsschritte . . . . .	61

4.4	Hilfsmittel der Navigation . . . . .	61
4.4.1	Historie . . . . .	61
4.4.2	Lexikonzugriff . . . . .	63
4.4.3	Titelleisten . . . . .	64
4.4.4	Virtueller Dom . . . . .	65
4.4.5	Weitere geplante Hilfsmittel . . . . .	66
<b>5</b>	<b>Vergleich der beiden Anwendungen</b>	<b>69</b>
5.1	Datenbasis . . . . .	69
5.1.1	Knoten . . . . .	69
5.1.2	Verknüpfungen . . . . .	70
5.1.3	Weitere Strukturen . . . . .	70
5.1.4	Zusammenfassung . . . . .	70
5.2	Die hypermedialen Systeme . . . . .	71
5.2.1	Die genutzten Modelle . . . . .	72
5.2.2	Instanzierungsfunktion . . . . .	72
5.2.3	Navigationsschritte . . . . .	72
5.3	Klassifikationsschema für Hilfsmittel . . . . .	73
5.3.1	Darstellungsbeschreibungen . . . . .	73
5.3.2	Zustand . . . . .	73
5.3.3	Systeminteraktion . . . . .	74
5.3.4	Benutzerinteraktion . . . . .	75
5.3.5	Adaptivität . . . . .	75
5.4	Klassifikation der Hilfsmittel . . . . .	76
5.4.1	Die Hilfsmittel des Industriemeisters 2000 . . . . .	76
5.4.2	Die Hilfsmittel des Altenberger Doms . . . . .	78
5.4.3	Bewertung des Klassifikationsschemas . . . . .	79
5.5	Vergleich der Hilfsmittel . . . . .	80
5.5.1	Darstellungsbeschreibungen . . . . .	80
5.5.2	Zustände . . . . .	80
5.5.3	Systeminteraktion eines allgemeinen Hilfsmittels . . . . .	81
5.5.4	Benutzerinteraktion . . . . .	84
5.5.5	Adaptivität . . . . .	84
5.6	Bewertung des Vergleichs . . . . .	84

<b>6</b>	<b>Definition der Navigation</b>	<b>87</b>
6.1	Eine Erweiterung des Dexter Modells für Hilfsmittel . . . . .	87
6.1.1	Motivation . . . . .	87
6.1.2	Die überarbeitete Schichtenarchitektur . . . . .	88
6.1.3	Darstellungsbeschreibungen . . . . .	88
6.1.4	Laufzeitauflöserfunktion . . . . .	89
6.1.5	Die Navigationsschrittfunktion . . . . .	91
6.1.6	Zusammenfassung . . . . .	92
6.2	Definition . . . . .	93
6.3	Bewertung der Definition . . . . .	93
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>95</b>
7.1	Zusammenfassung . . . . .	95
7.2	Ausblick . . . . .	96
<b>A</b>	<b>Die Terminologie des Dexter Modells</b>	<b>99</b>
<b>B</b>	<b>Ein Vorgehensmodell zur Analyse der Navigation</b>	<b>105</b>
B.1	Dokumente der Analysephase . . . . .	105
B.2	Dokumente der Vergleichsphase . . . . .	106
B.3	Dokumente der Weiterentwicklungsphase . . . . .	107
B.4	Benutzung des Modells . . . . .	107
	<b>Literaturverzeichnis</b>	<b>109</b>

## Abkürzungsverzeichnis

2PC	2-Phasen-Commit
ADML	Altenberger Dom Markup Language (siehe Abschnitt 4.2)
AHAM	Adaptive Hypermedia Application Model (siehe Abschnitt 2.2.3)
AHM	Amsterdamer Hypermediamodell (siehe Abschnitt 2.2.2)
AHS	Adaptives hypermediales System (siehe Abschnitt 2.1.4)
CMIF	CWI (Centrum voor Wiskunde en Informatica) Multimedia Interchange Format
DIHT	Deutscher Industrie- und Handelstag
DTD	Document Type Definition (siehe Abschnitt 2.3)
HTML	Hypertext Markup Language
IHK	Industrie- und Handelskammer
VAM	Virtuelle ADML Maschine (siehe Abschnitt 4.3.1)
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language (siehe Abschnitt 2.3)

Weitere Abkürzungen richten sich nach [Duden96].

## Zeichenerklärung / Anmerkungen

- Namen von Funktionen oder Variablen, die aus den Abbildungen 2.4 und 6.2 in den Fließtext übernommen werden, werden in einer Schreibmaschinenschrift gesetzt (zum Beispiel: **FolgeVerknüpfung**).
- Neu eingeführte Begriffe werden in kursiver Schrift gesetzt (beispielsweise: *Komponente*).
- Englische Originalbegriffe werden in Klammern und Anführungszeichen hinter der deutschen Übersetzung angeführt.
- Definitionen und Arbeitshypothesen werden durchnummeriert und in kursiver Schrift gesetzt.
- Während der Klassifizierung der Hilfsmittel im Abschnitt 5.4 wird die jeweilige Einordnung in eine Klasse in Fettschrift gesetzt.
- Aufgrund der umfangreichen Terminologie werden die Begriffe des Dexter Modells im Anhang A zusammengefaßt.



---

# Kapitel 1

## Einleitung

In diesem einleitenden Kapitel wird die Aufgabenstellung und der Aufbau der Arbeit erläutert.

### 1.1 Motivation und Kontext

Informationen werden in Systemen, die auf einer Hypertextbasis beruhen, auf Knoten verteilt und mit Verknüpfungen zwischen Knoten verbunden. Im folgenden werden Systeme, die auf einer Hypertextbasis aufbauen, hypermediale Systeme genannt. Hypermediale Systeme erlauben die Betrachtung der Knoteninhalte eines (oder mehrerer) Knoten und das Folgen von Verknüpfungen zum Wechsel zu einem anderen Knoten, dessen Inhalte daraufhin präsentiert werden. Zur Beschreibung der aktiven Handlung durch den Benutzer des hypermedialen Systems wird meistens die „Navigationsmetapher“ herangezogen (siehe [Gay91]). Die Navigationsmetapher hebt die Rolle des Benutzers hervor, der aktiv durch ein hypermediales System „navigiert“. Nicht der Autor des Systems, sondern der Benutzer bestimmt den „Kurs“ durch das System, d. h. welche Knoten in welcher Reihenfolge „angesteuert“ werden. Der Autor kann aber, ähnlich einem Kartographen, Hilfsmittel bereitstellen, welche die Navigation unterstützen, um bestimmte Knoten und Informationen zu finden. Einige solcher Hilfsmittel werden ebenfalls in [Gay91] beschrieben. Dazu gehören „Index“ und „On-line Guidance“.

Die beiden hypermedialen Systeme, die jeweils als Teilziel der beiden Projekte Industriemeister 2000 (vgl. [IHK96]) und Altenberger Dom (vgl. [AD97]) realisiert werden, stellen dem Benutzer eine Reihe von Hilfsmitteln zur Vereinfachung der Navigation zur Verfügung. Dazu gehören Hilfsmittel mit den bildhaften Namen „Lesezeichen“ und „Inhaltsverzeichnis“.

### 1.2 Aufgabenstellung

Das Ziel der Diplomarbeit besteht in der Definition des Begriffes Navigation für die oben genannten hypermedialen Systeme Industriemeister 2000 und Altenberger Dom. Dazu sol-

len, ausgehend von einer Beschreibung und Analyse der beiden Systeme, die beiden Systeme und speziell die Hilfsmittel der Navigation miteinander verglichen werden. Aufbauend auf diesem Vergleich soll dann eine Definition erstellt werden, die auch Hilfsmittel mit in die Navigation einbezieht.

## 1.3 Aufbau der Arbeit

Dieser Abschnitt gibt einen Überblick über die Inhalte der einzelnen Kapitel der vorliegenden Diplomarbeit.

**Kapitel 1:** In dem vorliegenden ersten Kapitel werden Aufgabenstellung und Aufbau der Arbeit erläutert.

**Kapitel 2:** Im zweiten Kapitel werden wichtige Aspekte hypermedialer Systeme beschrieben. Die Begrifflichkeiten der Aufgabenstellung und wichtige Konzepte für die Arbeit werden erläutert. Im begriffsbildenden Abschnitt 2.1 werden die wesentlichen Charakteristika von Hypertext, Multimedia und Hypermedia, sowie adaptiver Hypermediasysteme beschrieben. Der Abschnitt 2.2 erläutert Referenzmodelle für Hypermediaanwendungen. Strukturierte Hypermedia-Datenmengen werden in Abschnitt 2.3 beschrieben. Das Kapitel schließt mit einem Abschnitt über Navigation. Hier werden Hilfsmittel erklärt und es wird eine Arbeitshypothese für Navigation vorgestellt, die bis zur Definition im Kapitel 6 den Begriff der Navigation mit einer Bedeutung versteht.

**Kapitel 3:** Das dritte Kapitel umfaßt eine Beschreibung der hypermedialen Anwendung (siehe Abschnitt 2.1.3) Industriemeister 2000. Die Darstellung des Projektes in Abschnitt 3.1 vermittelt einen Überblick über das Projekt und somit den Kontext für die weiteren Darstellungen. Die Analyse der Datenbasis im Abschnitt 3.2 umfaßt eine Beschreibung ihrer Struktur und einen Vergleich mit dem Dexter Referenzmodell. Das hypermediale System wird im Abschnitt 3.3 mit dem Dexter Modell und dem AHAM verglichen. Die im Abschnitt 3.4 beschriebenen Hilfsmittel bauen auf diesen Beschreibungen auf, da sie Strukturen der Datenbasis nutzen und mit dem hypermedialen System interagieren.

**Kapitel 4:** Das vierte Kapitel umfaßt eine Beschreibung der hypermedialen Anwendung des Altenberger Doms. Sie erfolgt analog zu der Beschreibung aus dem Kapitel 3.

**Kapitel 5:** Im fünften Kapitel werden die beiden Anwendungen miteinander verglichen. Zunächst erfolgt ein Vergleich der Datenbasis (siehe Abschnitt 5.1) und der hypermedialen Systeme (siehe Abschnitt 5.2). Für den Vergleich der Hilfsmittel wird ein Klassifikationsschema erläutert (siehe Abschnitt 5.3), welches charakteristische Besonderheiten der Hilfsmittel aus Sicht der Navigation beschreibt. Aufbauend darauf werden Hilfsmittel klassifiziert (siehe Abschnitt 5.4) und ihre allgemeinen Eigenschaften extrahiert (siehe Abschnitt 5.5).

**Kapitel 6:** Anhand der allgemeinen Eigenschaften von Hilfsmitteln wird im sechsten Kapitel im Abschnitt 6.1 eine Erweiterung des Dexter Modells für Hilfsmittel erstellt. Diese Erweiterung ist die Basis für die Definition der Navigation der beiden betrachteten hypermedialen Systeme (siehe Abschnitt 6.2).

**Kapitel 7:** Im letzten Kapitel wird eine Zusammenfassung und mögliche Fortsetzungen der Arbeit beschrieben.

## 1.4 Danksagung

Ich möchte mich bei allen bedanken, die mich bei dieser Arbeit durch Diskussionen und Anregungen unterstützt haben: Klaus Alfert, Lars Bunge, Michael Emmerich, Wolfgang Franke, Armin Freese, Matthias Heiduck, Georg Kneupner, Oliver Koster, Volker Mattick, Nicole Negwer, Daniela Püttmann, Frank Wächter, Jens Ziegler und Peter Ziesche.



---

# Kapitel 2

## Grundlagen

In diesem Kapitel werden wichtige Aspekte hypermedialer Systeme beschrieben. Die Begrifflichkeiten der Aufgabenstellung werden genauer erläutert, und wichtige Konzepte, die in weiteren Kapiteln genutzt werden, beschrieben. Im begriffsbildenden Abschnitt 2.1 werden die wesentlichen Charakteristika von Hypertext, Multimedia, Hypermedia und adaptiver Systeme erläutert. Der Abschnitt 2.2 beschreibt Referenzmodelle für Hypermediaanwendungen. Anhand der dort vorgestellten Modelle werden in Kapitel 3 und 4 die beiden hypermedialen Anwendungen Industriemeister 2000 und Altenberger Dom beschrieben. Der Abschnitt 2.3 erläutert strukturierte Hypermedia-Datenmengen. Abschließend wird Herkunft und Bedeutung des Begriffes Navigation im Kontext hypermedialer Systeme im Abschnitt 2.4 dargestellt. Es wird aufgezeigt, warum eine Definition der Navigation von Interesse ist, und es wird im Abschnitt 2.4.4 eine Arbeitshypothese für Navigation aufgestellt. Im Zusammenhang mit der Navigation werden in Abschnitt 2.4.3 die für alle weiteren Kapitel zentralen Hilfsmittel der Navigation beschrieben.

### 2.1 Hypermediale Systeme

Im folgenden werden Hypertext (siehe Abschnitt 2.1.1) und Hypermedia (siehe Abschnitt 2.1.3) beschrieben. Da Hypermedia aus der Kombination von Hypertext und Multimedia entsteht, wird letzteres ebenfalls erläutert (siehe Abschnitt 2.1.2).

#### 2.1.1 Hypertext

Der Begriff *Hypertext* geht zurück auf Theodor Nelson. Er verstand darunter eine Menge von Texten und Bildern, die auf eine so komplexe Art miteinander verknüpft sind, daß sie nicht einfach auf Papier dargestellt werden können (vgl. [Nelson65]). Die Idee von Hypertext wird häufig Vannevar Bush zugeschrieben (vgl. [Lennon97, S.17]), der im Jahre 1945 ein System mit dem Namen *Memex* (Memory Expander) postulierte, welches Datenmaterial, beispielsweise Bücher und Tonaufnahmen, miteinander verbunden speichern sollte [Bush45]. Seit diesen Anfängen hat sich Hypertext ständig weiterentwickelt (vgl.

[Lennon97]). „Das Ziel bestand darin, eine neue ‚Textualität‘ in Form eines Informationsmediums zu schaffen, das Lesern die Möglichkeit bieten sollte, einen Inhaltsbereich nicht in einer bereits vorab festgelegten traditionell linearen Form, sondern auf unterschiedlichen eigenen Pfaden zu erschließen“ [Tergan97]. Inzwischen besitzen die meisten Anwendungen für einen Rechner eine Online-Hilfe, die auf Hypertext basiert. Diese Dateien stellen einzelne Hyperdokumente dar. Fast jede größere Anwendung für das Betriebssystem Windows besitzt mindestens eine solche Hilfedatei.

**Definition 1 (Hypertext)**

*Ein Hypertext besteht aus miteinander verknüpften Knoten, die Objekte mit beliebigen Informationen umfassen können.*

Ein Hypertext wird auch *Hypertextdokument* genannt. Beziehungen jeder Art zwischen Knoten können explizit durch *Verweise*, auch *Verknüpfungen* genannt, hergestellt werden. Verknüpfungen können *gerichtet* oder *ungerichtet* sein. Die Knoten besitzen *Informationen* (oder auch *Inhalte*) in beliebigen Formen, beispielsweise als Text oder als Grafik (vgl. Abschnitt 2.1.3). Die Gesamtheit der Informationen eines Hypertextes, also Knoten, Inhalte und Verknüpfungen, wird *Datenbasis* genannt. Die Verknüpfungen sind grundlegende Informationen der Datenbasis, da sie beschreiben, welche Informationen zueinander in Bezug stehen. Ein Hypertext bedarf eines geeigneten *Hypertextsystems* zu seiner *Darstellung*. Oftmals stellt das System auch Funktionalität zur Erstellung und Veränderung eines Hypertextes zur Verfügung. Der *Betrachter* oder auch *Benutzer* liest den Hypertext, wie er von einem Hypertextsystem dargestellt wird und interagiert mit dem System. Ein Hypertextsystem stellt die Inhalte der Knoten dar, meistens mit Hilfe einer graphischen Benutzeroberfläche. Verknüpfungen werden in der Regel durch einfache Handlungen des Benutzers *aktiviert*, zum Beispiel durch einen Mausklick. Das Hypertextsystem stellt daraufhin den Zielknoten der aktivierten Verknüpfung dar. Will man Verknüpfungen leicht zugänglich machen, so brauchen sie eine geeignete Darstellung. Einzelne Wörter oder, im Falle graphischer Inhalte, bestimmte Bereiche einer Graphik können als (*Start-*) *Anker* genutzt werden. Eine Verknüpfung kann dann anhand des sichtbaren Startankers aktiviert werden. Mit Hilfe von (*Ziel-*) *Ankern* kann auf bestimmte Inhalte innerhalb des Zielknotens verwiesen werden.

Eine Verknüpfung verweist immer auf einen Zielknoten. Das Hypertextsystem muß den Verweis *auflösen*, um damit den Zielknoten zu ermitteln. Kann eine Verknüpfung nicht aufgelöst werden, zum Beispiel wenn der Zielknoten nicht mehr existiert, so nennt man dies einen *baumelnden* oder auch einen *toten Verweis* („dangling link“).

Ein Hypertext und ein dazu passendes Hypertextsystem bilden zusammen eine *Hypertextanwendung*.

**2.1.2 Multimedia**

Nach Steinmetz sind *Multimediasysteme* Systeme zur rechnergestützten, integrierten Erzeugung, Manipulation, Darstellung, Speicherung und Kommunikation von unabhängigen

Informationen, die in mindestens einem kontinuierlichen (zeitabhängigen) und einem diskreten (zeitunabhängigen) Medium kodiert sind [Steinmetz99, Seite 13]. Unter kontinuierlichen Medien versteht man vor allem digitale Audio- und Videoströme, während unter diskreten Medien vor allem Texte und Bilder verstanden werden.

Nach Schulmeister ist *Multimedia* eine „interaktive Form des Umgangs mit symbolischem Wissen in einer computergestützten Interaktion“ [Schulmeister96, Seite 18].

Beiden Definitionen ist gemeinsam, daß Multimedia nicht einfach durch die Kombination multipler Medien entsteht. Wichtig ist nach Steinmetz ein Zeitbegriff und nach Schulmeister die Interaktion des Benutzers mit der Software. Diese Arbeit schließt sich diesen beiden Kriterien an und definiert damit Multimedia wie folgt:

**Definition 2 (Multimedia)**

*Multimedia ist eine Kombination zeitabhängiger und zeitunabhängiger Medien, mit denen der Benutzer rechnergestützt interagieren kann.*

Multimedia verhält sich zu Multimediasystemen ähnlich wie Hypertext zu Hypertextsystemen. Erst mit einem rechnergestützten System kann Multimedia dargestellt und dem Benutzer präsentiert werden.

### 2.1.3 Hypermedia

Hypermedia entsteht aus der Kombination von Multimedia und Hypertext, allerdings existieren unterschiedliche Ansichten über die Richtung der Integration. So sehen Newcomb et al. ein Hypermediadokument als ein Multimediadokument mit Verknüpfungen an [Newcomb91]. Nach Tergan [Tergan97] hingegen ist Hypermedia ein multimedial angereicherter Hypertext: „Die Unterscheidung zwischen Hypertext- und Hypermediasystemen bezieht sich – technologisch gesehen – primär auf Unterschiede in der Kodierungsform der in der Datenbasis repräsentierten Informationen.“

Andere Autoren nennen multimediale Merkmale als Unterscheidungsmerkmal von Hypertext und Hypermedia: Tochtermann baut auf der Multimediadefinition von Steinmetz [Steinmetz99] auf und versteht unter einem Hypermediasystem ein rechnergestütztes System, mit dessen Hilfe Hypermediadokumente erstellt, manipuliert, dargestellt und gespeichert werden können. Die Knoten dieser Hypermediadokumente können sowohl kontinuierliche als auch diskrete Medien besitzen [Tochtermann95]. Hardman et al. erachten die zeitliche Dimension als wichtigste Unterscheidung zwischen Hypertext und Hypermedia. Erst durch die zeitliche und räumliche Synchronisation von Medien mit einer hypertextuellen Grundstruktur erhält man ein Hypermediadokument [Hardman94]. Haack hingegen betrachtet, ähnlich Schulmeister, Interaktivität als zentrale Eigenschaft von Multimedia und Hypermedia [Haack97].

Auch im folgenden wird Hypermedia als eine Kombination von Multimedia und Hypertext angesehen. Dabei wird die Struktur von Hypertext mit multimedialen Inhalten ausgefüllt.

**Definition 3 (Hypermedia)**

*Hypermedia ist Hypertext mit multimedialen Erweiterungen, also Hypertext mit einer zeitlichen Dimension und der Interaktion mit den Inhalten.*

Ein *Hypermediadokument* ist somit ein Hypertextdokument, dessen Knoten multimediale Inhalte besitzen, oder dessen Anker auf multimediale Art dargestellt werden.

**Definition 4 (Hypermediales System)**

*Ein hypermediales System ist ein Werkzeug zur Darstellung eines Hypermediadokumentes.*

Ein hypermediales System (auch *Hypermediasystem*) umfaßt oftmals auch Funktionalität zur Erstellung und Veränderung eines Hypermediadokumentes. Zusammen bilden Hypermediadokument und Hypermediasystem eine *Hypermediaanwendung*.

Hypertext und Hypermedia basieren auf derselben Basis von Knoten und Verweisen. Die Inhalte der Knoten können sich unterscheiden, ebenso die Anker, weil Hypermediaanker sich zusätzlich auf zeitliche Informationen der Darstellung eines Knotens beziehen können. Verknüpfungen unterscheiden sich nicht.

Hypermedia wird im folgenden als Oberbegriff zu Hypertext verstanden. Hypertext steht damit für eine Sichtweise, die den Blickwinkel auf die Struktur, bestehend aus Knoten und Verknüpfungen, lenkt und die zeitliche Koordinierung von verschiedenen Medien als ein davon unabhängiges Problem versteht. Somit kann auch ein Hypermediadokument als ein Hypertextdokument bezeichnet werden, wenn die Inhalte zweitrangig sind.

Der Oberbegriff für Hypertextdokument und Hypermediadokument ist *Hyperdokument*.

### 2.1.4 Adaptive hypermediale Systeme

*Adaptive hypermediale Systeme (AHS)* sammeln Informationen über einen Benutzer des Systems, um ihn bei seiner Informationssuche zu unterstützen [DeBra99]. Brusilovsky [Brusilovsky96] beschreibt adaptive hypermediale Systeme folgendermaßen: „By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system, it should have a user model, and it should be able to adapt the hypermedia using this model.“

Adaptive hypermediale Systeme können Inhalte und Verweise *adaptieren*. Unter *Inhaltsadaptation* wird zum Beispiel das Einblenden zusätzlicher Inhalte (Texte, Bilder, usw.) und das Austauschen von Textstellen, Medien oder ganzer Knoten verstanden. Das Anzeigen zusätzlicher und das Austauschen oder Bewerten (über Farben) vorhandener Verknüpfungen wird *Verweisadaptation* genannt. Mit adaptivem Verhalten ist keine Anpaßbarkeit durch Systemeinstellungen, Optionsdialoge oder ähnlichen Benutzereinstellungen gemeint. Die Informationen werden implizit gesammelt, was bedeutet, daß der Benutzer nicht selbständig die Informationen ändert, die das adaptive System über ihn sammelt. Nach De Bra et al. [DeBra99] besitzt ein AHS folgende Bestandteile:

**Bereichsmodell:** Das *Bereichsmodell* („domain model“) beschreibt die Metadaten des Hyperdokumentes, also wie die Informationen im Hyperdokument strukturiert und miteinander verbunden werden. Das Bereichsmodell unterscheidet sich von der Datenbasis dadurch, daß die Inhalte der Knoten nicht zum Bereichsmodell gehören.

**Benutzermodell:** Das *Benutzermodell* („user model“) beschreibt welche Informationen über einen Benutzer dauerhaft gesammelt werden.

**Lehrmodell:** Das *Lehrmodell* („teaching model“) ist eine Sammlung von *pädagogischen Regeln* („pedagogical rules“). Diese wiederum bestimmen, wie Bereichsmodell und Benutzermodell genutzt werden, um das AHS anzupassen. Die Namensgebung impliziert nicht die ausschließliche Nutzung des Modells im Rahmen von Lehrsystemen.

**Adaptionsgestalter:** Der *Adaptionsgestalter* („adaptive engine“) führt die konkreten Adaptionen durch, indem Inhalte und Verknüpfungen dynamisch adaptiert oder erstellt werden.

Pädagogische Regeln aktualisieren das Benutzermodell und werden von dem Adaptionsgestalter zur Generierung von Darstellungsbeschreibungen (siehe Abschnitt 2.2.1) genutzt, welche die Darstellung von Knoten und Verknüpfungen beeinflussen. Pädagogische Regeln besitzen keine feste Form. De Bra et al. schlagen in [DeBra99] eine an Prädikatenlogik erster Stufe angelehnte Form vor.

## 2.2 Modelle für hypermediale Systeme

Referenzmodelle ermöglichen es, Hypermediaanwendungen zu vergleichen. Diese Arbeit vergleicht zwei hypermediale Anwendungen, die zunächst in den Kapiteln 3 und 4 mit Hilfe der nun vorgestellten Referenzmodellen beschrieben werden. Das bekannteste Referenzmodell ist das Dexter Modell, welches gleichzeitig die Basis für die weiteren hier vorgestellten Referenzmodelle ist, die ebenfalls vorgestellt werden. Das Amsterdamer Hypermediamodel (siehe Abschnitt 2.2.2) erweitert das Dexter Modell um Konzepte für die Modellierung von Hypermedia, und das AHAM (siehe Abschnitt 2.2.3) erweitert das Dexter Modell zur Modellierung adaptiver hypermedialer Systeme.

### 2.2.1 Dexter Hypertext Referenzmodell

Das Dexter Hypertext Referenzmodell [Halasz90, Halasz94] wurde mit dem Ziel entwickelt, Hypertextsysteme vergleichbar zu machen und Hypertexte austauschbar zu gestalten. Es umfaßt das Benutzen und Erstellen von Hypertexten, nicht aber die Art, auf der Knoten und Verknüpfungen dargestellt werden. Die möglichen Inhalte sowie ihre Darstellung erschienen zu vielfältig, als daß sie in einem allgemeinen Modell beschreibbar wären. Das Dexter Modell wurde formal mit der Spezifizierungssprache *Z* [Spivey98] erstellt. Die nun folgende Beschreibung des Modells basiert auf der natürlichsprachlichen Beschreibung aus [Halasz94].

Die zentrale Idee des Dexter Modells ist die separate Verwaltung der Verknüpfungen und der Knoten. Eine Verknüpfung ist im Dexter Modell eine eigenständige, den Knoten gleichgestellte Komponente. Hieraus ergeben sich eine Reihe von Vorteilen zu Systemen, welche die Verweise als Bestandteil der Knoten verwalten. Die Vorteile liegen darin begründet, daß ein System gezielt die vorhandenen Verweise abfragen kann. Somit können baumelnde

(tote) Verweise leicht identifiziert werden und das Hinzufügen und Entfernen von Verweisen vereinfacht sich (vgl. [Grønbaek97]). Anwendungen, die diese Trennung nicht besitzen, sind auch heute noch weit verbreitet. Das bekannteste Beispiel ist das World Wide Web, welches innerhalb einer textuellen Beschreibung der Knoten gleichzeitig auch die Verweise deklariert.

Dieser Abschnitt ist wie folgt gegliedert: Zuerst wird die Komponente beschrieben, die zentrale Datenstruktur des Dexter Modells für Knoten und Verweise. Daraufhin werden die Schichten erläutert, die jeweils eine bestimmte Funktionalität zur Verfügung stellen. Im Anschluß daran wird auf einige ausgewählte Themen im Zusammenhang mit dem Dexter Modell eingegangen, die für diese Arbeit von Interesse sind.

### 2.2.1.1 Komponenten

Die *Komponente* („component“) ist die zentrale Datenstruktur des Dexter Modells. Sie umfaßt sowohl die Knoten als auch die Verknüpfungen des Hyperdokumentes.

Eine Komponente besteht aus einer *Komponenten-ID* („unique identifier“), einer *Darstellungsbeschreibung* („presentation specification“), einer Menge von *Eigenschaften* („attribute/value pairs“), einer Menge von *Ankern* („anchors“) und einer *Basiskomponente* („base component“) wie in Abbildung 2.1 dargestellt. Diese Bestandteile sind im einzelnen:

**Komponenten-ID:** Die Komponenten-ID ist eine systemweit eindeutige Bezeichnung für die Komponente.

**Darstellungsbeschreibungen:** Darstellungsbeschreibungen werden nicht näher im Modell behandelt. Dieser Abschnitt der Komponenteninformationen ermöglicht es, In-

Komponenten-ID	eindeutiger Bezeichner				
Darstellungsbeschreibung	nicht genauer beschrieben				
Eigenschaften	<table border="1"> <tr><td>Name</td></tr> <tr><td>Wert</td></tr> </table>	Name	Wert		
Name					
Wert					
Anker	<table border="1"> <tr><td>Anker-ID</td></tr> <tr><td>Ankerwert</td></tr> </table>	Anker-ID	Ankerwert		
Anker-ID					
Ankerwert					
Basiskomponente	<table border="1"> <tr><td>Daten oder Zeiger auf Daten</td></tr> <tr><td>atomar</td></tr> <tr><td>Verknüpfung</td></tr> <tr><td>zusammengesetzt</td></tr> </table>	Daten oder Zeiger auf Daten	atomar	Verknüpfung	zusammengesetzt
Daten oder Zeiger auf Daten					
atomar					
Verknüpfung					
zusammengesetzt					

Abbildung 2.1: Komponenten (Darstellung in Anlehnung an [Hardman94])

formationen über die Darstellung der Komponente zur Verfügung zu stellen. Die Informationen werden nicht von dem Dexter Modell, sondern von der Instanzierungsfunktion (siehe unten) genutzt.

**Eigenschaft:** Eine Eigenschaft ist ein Paar, bestehend aus einem Namen und einem Wert. Sie wird genutzt werden, um eine Komponente mit beliebigen Informationen auszustatten. Unter anderen können Schlüsselwörter und Typbezeichnungen als Eigenschaften zusammen mit den Komponenten verwaltet werden.

**Anker:** Ein Anker besteht aus einem komponentenweit eindeutigen Bezeichner, der *Anker-ID* („anchor id“), und einem *Ankerwert* („anchor value“). Der Ankerwert ist eine Information zur Beschreibung der Darstellung und wird deswegen nicht näher vom Modell behandelt. Er beschreibt eine Stelle, eine Region, ein Objekt oder eine Substruktur innerhalb einer Basiskomponente. Ein Anker kann systemweit eindeutig durch ein (Komponenten-ID, Anker-ID)-Tupel identifiziert werden. Innerhalb einer Komponente können beliebig viele Anker definiert werden. Diese Anker dienen als Endpunkte für Verknüpfungen.

**Basiskomponenten:** Die Basiskomponenten enthalten die Daten des Hypertextes: *atomare Komponenten* („atomic components“) besitzen Knoten, *Verknüpfungskomponenten* („link components“) Verknüpfungen als Basiskomponenten und die Basiskomponente der *Kompositum-Komponenten* („composite components“) ist aus Komponenten zusammengesetzt. Die Darstellung von Basiskomponenten gehört nicht zum Dexter Modell, sie werden im Modell unabhängig von Ihrer Darstellung beschrieben.

Die Darstellungsbeschreibung, die Liste der Anker und die Eigenschaften bilden zusammen die *Komponenteninformationen* („component information“). Die einzelnen Komponententypen sind im einzelnen:

**Atomare Komponente:** Die atomaren Komponenten entsprechen den Knoten eines Hyperdokumentes. Sie besitzen zu der Komponenten-ID und den Komponenteninformationen konkrete Inhalte als Basiskomponente. Inhalte und ihre Strukturen werden nicht näher beschrieben. Über die Liste der Anker kann auf Start- und Zielanker zugegriffen werden.

**Verknüpfungskomponente:** Verknüpfungen werden im Dexter Modell als Basiskomponenten angesehen. Somit besitzt eine Verknüpfungskomponente eine eigene Komponenten-ID, Komponenteninformationen und eine Verknüpfung als Basiskomponente. Eine Verknüpfung kann im Dexter Modell nicht nur einen Start- und einen Zielpunkt besitzen, sondern kann mehrere Startpunkte mit beliebig vielen Zielpunkten verknüpfen. Für jeden Start- und Zielpunkt wird eine *Endpunktbeschreibung* („specifier“) verwaltet. Eine Endpunktbeschreibung besteht aus einem *Komponentenverweis* („component specification“), einer Anker-ID, einer *Richtung* („direction“) und einer Darstellungsbeschreibung. Ein Komponentenverweis kann eine Komponenten-ID sein oder eine beliebige Beschreibung, solange sie durch die weiter unten beschriebene

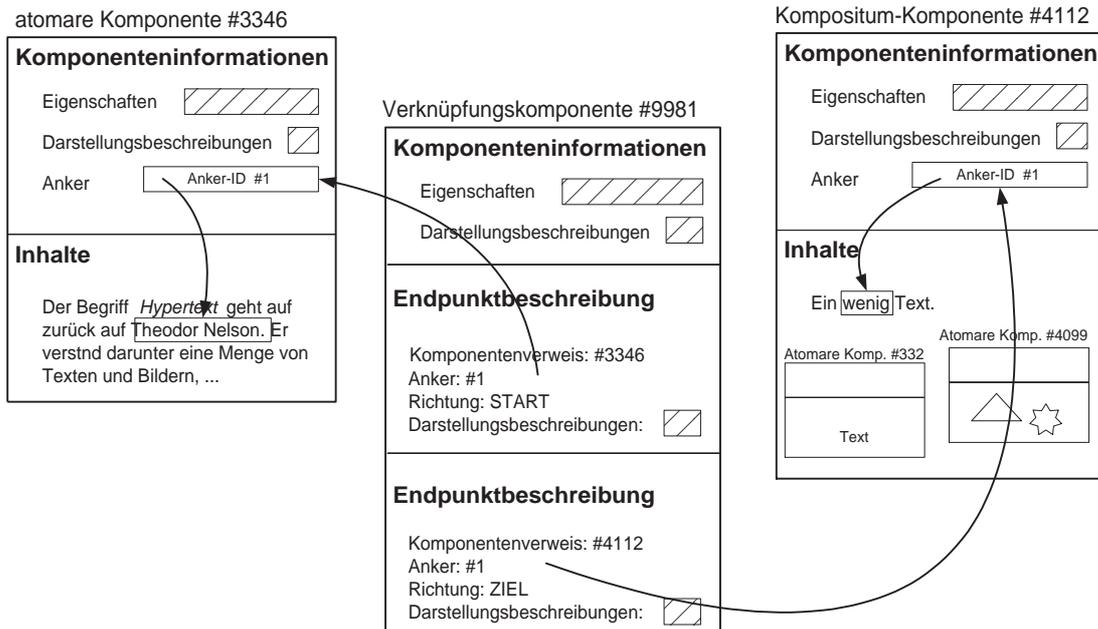


Abbildung 2.2: Beispiel für eine Verknüpfung mit zwei Endpunktbeschreibungen (Darstellung in Anlehnung an [Halasz94])

Auflöserfunktion zu einer Komponenten-ID aufgelöst werden kann. Die Richtung der Endpunktbeschreibung besagt, ob es sich um einen Zielpunkt *ZIEL* („TO“), einen Startpunkt *START* („FROM“), *BEIDES* („BIDIRECT“) oder *KEINES* („NONE“) von beidem handelt. Die Darstellungsbeschreibung besteht wiederum aus Informationen, die nicht von dem Modell verwendet werden.

Eine *Verknüpfung* („link“) ist eine Liste von zwei oder mehr Endpunktbeschreibungen. Verknüpfungen können nicht nur Beziehungen zwischen atomaren Komponenten beschreiben, sondern zwischen beliebigen Komponenten. Es sind somit auch Verknüpfungen, die auf Verknüpfungen verweisen, möglich. Besteht die Verknüpfung aus mehr als zwei Endpunktbeschreibungen, so wird sie als *mehrwertig* bezeichnet.

Eine beispielhafte Verknüpfung ist in Abbildung 2.2 dargestellt.

**Kompositum-Komponente:** Mit den Kompositum-Komponenten wurde ein hierarchischer Strukturierungsmechanismus (siehe Abschnitt 2.3) in das Dexter Modell aufgenommen. Eine Kompositum-Komponente besitzt wie jede Komponente eine Komponenten-ID und Komponenteninformationen. Als Basiskomponente besitzt sie weitere Komponenten, weshalb die Komponente ein rekursiver Typ ist. Damit sind auch Verknüpfungen als Teil einer Kompositum-Komponente erlaubt. Wie hierarchische Strukturen interpretiert werden bleibt dem hypermedialem System überlassen. Die Darstellungsbeschreibungen könnten derartige Informationen beschreiben. Eine Kompositum-Komponente darf als einzige Einschränkung sich selbst auch transitiv nicht enthalten.

### 2.2.1.2 Die Schichten

Das Modell unterteilt ein Hypertextsystem in drei Schichten (siehe Abbildung 2.3): eine *Laufzeitschicht* („run-time layer“), eine *Speicherschicht* („storage layer“) und eine *Komponenteninhaltsschicht* („within-component layer“). Die Schnittstelle zwischen der Komponenteninhaltsschicht und der Speicherschicht wird *Ankerschnittstelle* („anchoring“), die Schnittstelle zwischen der Speicherschicht und der Laufzeitschicht *Darstellungsbeschreibung* („presentation specifier“) genannt.

Die Schichtenarchitektur des Dexter Modell gleicht einer Instanz des Layers-Musters von Buschmann et al. [Buschmann98]. Jede Schicht baut auf die Funktionalität der darunterliegenden Schicht auf und stellt eine Funktionalität für die darüberliegende Schicht zur Verfügung. Insofern gleicht dieser Aufbau dem des OSI-7-Schichtenmodelles (siehe [Mühlhäuser97b]). Die Aufgaben der Schichten sind im einzelnen:

**Komponenteninhaltsschicht:** Die Komponenteninhaltsschicht verwaltet die Inhalte sowie die Struktur der Inhalte innerhalb der Komponente. Hier werden die verschiedenen Inhalte wie Texte, Bilder und Animationen gespeichert. Da die möglichen Datentypen sehr unterschiedlich sind, liegt die Komponenteninhaltsschicht außerhalb des Modells und es wird nicht versucht, eine Struktur innerhalb der Komponenten auf dieser Ebene zu modellieren. Diese Schicht stellt der Speicherschicht die Ankerschnittstelle zur Verfügung, über die auf Ankerinformationen innerhalb von Komponenten zugegriffen werden kann. Über diese Schnittstelle kann die Speicherschicht Informationen über die Anker erhalten, ohne Struktur und Inhalte der Komponenten zu kennen.

**Speicherschicht:** Der Schwerpunkt des Modells liegt auf der Speicherschicht. In dieser



Abbildung 2.3: Die drei Schichten des Dexter Modells (Darstellung in Anlehnung an [Halasz94])

Schicht werden die Komponenten zu einem Hypertext miteinander verbunden. Die Speicherschicht verwaltet eine Datenbank der Komponenten. Sie sieht Funktionen zum Erstellen und Löschen von Komponenten vor, sowie für den Zugriff auf Komponenten. Die zentralen Funktionen für den Zugriff auf Komponenten sind die *Zugriffsfunktion* („accessor function“) und die (*Verweis-*) *Auflöserfunktion* („resolver function“). Die Zugriffsfunktion liefert die zu einer Komponenten-ID zugehörige Komponente. Die Auflöserfunktion liefert eine (oder mehrere) Komponenten-ID(s) zu einer Verknüpfung, indem sie die Komponentenverweise auflöst. Dies ist von Bedeutung, wenn Verknüpfungen nicht die Komponenten-ID als Komponentenverweis benutzen, sondern beispielsweise einen Verweis auf eine Seite mit einem bestimmten Eigenschaftswert. Die Speicherschicht stellt die Darstellungsbeschreibung als Schnittstelle zur Verfügung. Diese Schnittstelle wird von dem Modell nicht näher beschrieben.

**Laufzeitschicht:** Der Benutzer interagiert mit der Laufzeitschicht. Während die Speicherschicht und die Komponenteninhaltsschicht den Hypertext als eine statische Datenstruktur beschreiben, dient die Laufzeitschicht der Erstellung, Betrachtung und Veränderung des Hypertextes durch den Benutzer. In dieser Schicht wird die Darstellung des Hypertextes verwirklicht, die Benutzerinteraktion behandelt und die dabei anfallenden dynamischen Daten verwaltet. Die Laufzeitschicht verwaltet für jeden Benutzer des Hypertextes eine *Sitzung* („session“). Für die Darstellung einer Komponente wird eine *Instanz* („instantiation“) von ihr erstellt, mit *Ankerinstanzen* („link marker“) für die Startanker, anhand derer Verknüpfungen aktiviert werden können. Instanzen können verändert werden. Änderungen an einer Instanz können gespeichert werden. Hierbei müssen einige *Hypertext-Invarianten* („invariants of hypertext“) (siehe Abschnitt 2.2.1.4) bewahrt bleiben. Die wichtigsten Funktionen dieser Schicht sind die *Instanziierungsfunktion* („instantiator function“) und die *Laufzeitauflöserfunktion* („run-time resolver function“). Die Instanziierungsfunktion liefert anhand einer Komponenten-ID und einer Darstellungsbeschreibung eine neue Instanz der Komponente zurück. Die Laufzeitauflöserfunktion besitzt dieselbe Aufgabe wie die Auflöserfunktion der Speicherschicht, kann aber zusätzlich noch Sitzungsdaten zum Auflösen des Verweises verwenden.

### 2.2.1.3 Folgen einer Verknüpfung im Dexter Modell

Die Darstellung eines neuen Knotens erfolgt im Dexter Modell als Reaktion auf das Aktivieren einer Verknüpfung. Das Modell beschreibt die Reihenfolge der genutzten Funktionen, die in der Darstellung neuer Knoten münden. Für die weiteren Betrachtungen wurde die textuelle Beschreibung aus [Halasz94] in eine Pseudoprogrammiersprache übertragen (siehe Abbildung 2.4). Nach der Aktivierung wird die *FolgeVerknüpfung*-Funktion („followLink“) aufgerufen, welche die Verknüpfung anhand der übergebenen Instanz-ID und der aktivierten Ankerinstanz auf nicht näher spezifizierte Weise ermittelt (hier als *getVerknüpfung* beschrieben). Daraufhin werden die Zielverweise (dazu gehören auch die bidirektionalen Verweise) der Verknüpfung ermittelt und zusammen mit nicht näher spezifizierten Darstellungsbeschreibungen der *Komponentendarstellung*-Funktion („present component“) übergeben. Diese wiederum nutzt die Laufzeitauflöserfunktion zum Bestimmen der Komponenten-ID und übergibt sie zusammen mit den Darstellungsbeschreibungen

```

procedure FolgeVerknüpfung (IID : Instanz-ID, aktivierterAnker : Ankerinstanz);
var Verweis : Endpunktbeschreibung;
    DB : Darstellungsbeschreibung;
    aktivierteVerknüpfung : Verknüpfung;
begin
    {die Funktion getVerknüpfung wird nicht spezifiziert!}
    aktivierteVerknüpfung = getVerknüpfung(IID, aktivierterAnker);
    for all Verweis in aktivierteVerknüpfung do
        if Verweis.Richtung = "ZIEL" or Verweis.Richtung = "BEIDES" then
            Komponentendarstellung (Verweis, DB) {DB wird nicht spezifiziert!}
        end if;
    end for;
end;

procedure Komponentendarstellung (KV : Komponentenverweis,
                                   DB : Darstellungsbeschreibung);
var KID : Komponenten-ID;
begin
    KID = Laufzeitauflöserfunktion(KV);
    Instanzierungsfunktion(KID, DB);
end;

```

Abbildung 2.4: Die FolgeVerknüpfung-Funktion und die Komponentendarstellung-Funktion des Dexter Modells.

gen der Instanzierungsfunktion. Das Modell sieht die Verwaltung mehrerer Instanzen durch das System vor, so daß der Aufruf einer Verknüpfung in der gleichzeitigen Präsentation mehrerer Zielkomponenten münden kann.

Diese FolgeVerknüpfung-Funktion beschränkt sich auf die Darstellung eines Knotens und wird durch die Aktivierung einer Ankerinstanz durch den Benutzer aufgerufen. Nicht betrachtet wird dabei eine Aktualisierung des Systems. Wenn, wie bei WWW-Browsern üblich, der Titel einer WWW-Seite in der Titelzeile des Browser angezeigt werden soll, so kann dies nicht mit dem Dexter Modell beschrieben werden, da eine solche Funktion nicht vorgesehen ist. Zwar kann eine Titeleigenschaft einer Komponente spezifiziert werden, aber es ist keine Funktionalität vorgesehen, die das System selbst beeinflußt. Die Instanzierungsfunktion erzeugt eine Instanz einer Komponente, die Titelzeile des Browsers ist aber Teil des Systems und wird deshalb auch von der Instanzierungsfunktion nicht beeinflußt.

#### 2.2.1.4 Hypertext-Invarianten

Ein Hypertext muß nach dem Dexter Modell fünf Invarianten erfüllen. Die Invarianten werden für diese Arbeit durchnummeriert. Eine solche Numerierung und die dadurch ent-

stehende Bezeichnung wurde von den Autoren des Dexter Modells nicht vorgenommen.

Die Invarianten lauten im einzelnen:

- I1:** Die Zugriffsfunktion ist eine umkehrbare Funktion, die Komponenten-IDs auf Komponenten abbildet. Das impliziert, daß jede Komponente eine Komponenten-ID besitzt.
- I2:** Der Bildbereich der Auflöserfunktion umfaßt alle Komponenten-IDs.
- I3:** Keine Komponente ist (direkt oder transitiv) Teil von sich selbst.
- I4:** Die Anker-IDs der Endpunktbeschreibungen stimmen mit den Anker-IDs der Komponenten überein, zu denen die Komponentenverweise der Endpunktbeschreibungen aufgelöst werden können.
- I5:** Alle Komponentenverweise können zu Komponenten-IDs aufgelöst werden.

#### 2.2.1.5 Mächtigkeit des Modells

Das Modell beschreibt Hypertext auf sehr abstrakter Ebene. Einige Merkmale des Modells, wie Verknüpfungen, die auf Verknüpfungen verweisen, mehrwertige Verknüpfungen oder zusammengesetzte Komponenten werden nur in wenigen hypermedialen Systemen verwirklicht. Die Autoren des Modells verweisen darauf, daß ihr Modell mächtiger sei als notwendig („our model is much more powerful than necessary“ [Halasz94]). Weitere Ausführungen besagen, daß zum Zeitpunkt der letzten Veröffentlichung (1994) kein ihnen bekanntes System alle Merkmale des Modells erfüllt hätte. Die Autoren des Modells folgern daraus, daß nicht alle Merkmale zwingend notwendig sind für ein hypermediales System, und deswegen obengenannte Merkmale des Modells als optional anzusehen seien. Die Autoren führen weiter aus, daß hypermediale Systeme anhand der von ihnen unterstützten Merkmale verglichen werden können.

#### 2.2.2 Amsterdamer Hypermediamodel (AHM)

Das *Amsterdamer Hypermediamodel* (AHM, siehe [Hardman94]) ist eine Weiterentwicklung des Dexter Referenzmodells. Die grundlegende Erweiterung ist die ausdrückliche Betrachtung und Modellierung zeitlicher Informationen. Damit ist das AHM ein Modell speziell für Hypermediaanwendungen. Es entstand aus einer Kombination des Dexter Hypertextmodells [Halasz90, Halasz94] mit dem CMIF-Multimediamodell [Bulterman91, Bulterman93]. Während beim Dexter Modell (vgl. Abschnitt 2.2.4) zeitliche Informationen als Teil der Komponenten angesehen und damit nicht betrachtet werden, kann in dem AHM die Synchronisierung von Medien innerhalb einer Gruppe ebenfalls modelliert werden. Das AHM sieht eine Aufteilung der Medien in Gruppen vor. Die zeitliche Synchronisation aller Medien einer Komponente wird durch die Synchronisation innerhalb der Gruppen und zwischen den Gruppen beschrieben. Zur Beschreibung von komplexerer Synchronisation innerhalb oder zwischen Gruppen werden Synchronisationsbögen eingesetzt. Synchronisationsbögen ähneln Verknüpfungen, weil sie die zeitliche Relation zwischen Komponenten beschreiben.

Die Gruppen werden durch die Kompositum-Komponenten des Amsterdamer Hypermediamodells beschrieben. Abbildung 2.5 stellt die abgewandelten Komponenten des AHMs dar. Die wesentliche Änderung der atomaren Komponenten sind die umfangreichen Darstellungsbeschreibungen. Die Kompositum-Komponenten wurden um Synchronisationsbögen erweitert, die Komponenten in zeitlichen Zusammenhang setzen. Eine Kompositum-Komponente kann nun keine eigenen Inhalte mehr besitzen, sie dient ausschließlich der Strukturierung und Synchronisation.

Das CMIF-Multimediamodell basiert auf einer hierarchischen Struktur von multimedialen Komponenten und ihrer Präsentation in *Kanälen* („channels“). Ein Kanal ist dabei ein abstraktes Ausgabegerät, zum Beispiel ein Bereich des Bildschirms oder eine Audiospur. Die Komponenten wurden diesem Modell angepaßt. Es wird nun der Zeitpunkt, die Dauer und der Kanal einer Komponentendarstellung spezifiziert.

Diese Änderungen wurden in der Speicherschicht des Modells vorgenommen, die Laufzeitschicht muß nun natürlich die zusätzlichen Informationen verarbeiten. Die im Abschnitt 2.2.1.3 beschriebene *FolgeVerknüpfung*-Funktion des Dexter Modells wird dadurch aber nicht verändert, da die Instanzierungsfunktion diese Funktionalität kapselt.

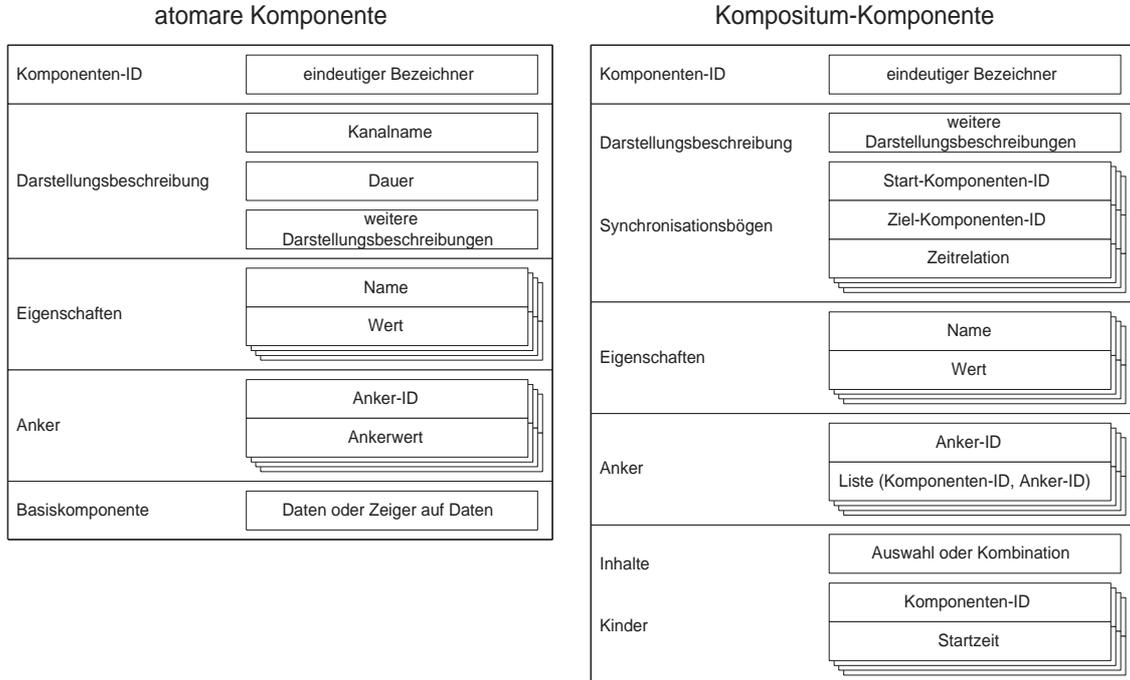


Abbildung 2.5: Die Komponenten des Amsterdamer Hypermediamodells (Darstellung in Anlehnung an [Hardman94])

### 2.2.3 Modell adaptiver Hypermediaanwendungen (AHAM)

Das *AHAM* („adaptive hypermedia application model“, siehe [DeBra99]) ist eine Erweiterung des Dexter Modells (siehe Abschnitt 2.2.1) für adaptive hypermediale Systeme (siehe Abschnitt 2.1.4).

Die drei zentralen Modelle eines adaptiven hypermedialen Systems (Benutzer-, Lehr- und Bereichsmodell) werden in der Speicherschicht des Dexter Modells modelliert (siehe Abbildung 2.6). Dabei entspricht das Bereichsmodell im wesentlichen der Datenbank des Dexter Modells. Im Rahmen des AHAM werden Komponenten *Konzepte* („concepts“) genannt. Es werden nicht nur Verknüpfungen zwischen Konzepten modelliert, sondern beliebige Relationen, welche durch eine speziell eingeführte Typ-Eigenschaft der Relation unterschieden werden.

Das Benutzermodell besteht aus Eigenschaften, die zu den Konzepten gehören. Im Unterschied zu den Eigenschaften des Dexter Modells werden diese Eigenschaften von dem System verändert und für jeden Benutzer getrennt gespeichert. Das Benutzermodell besteht aus einem Schema, nach dem für jeden Benutzer eine Instanz angelegt wird.

Das Lehrmodell besteht aus einer Sammlung von pädagogischen Regeln. Die Ausführung pädagogischer Regeln wird durch die Zugriffsfunktion ausgelöst. Bei jedem Zugriff auf ein Konzept (Komponente) wird das Benutzermodell aktualisiert. Dazu werden die pädagogischen Regeln ausgewertet, welche aus einem Trippel  $(R, PH, PR)$  für allgemeine Fälle, oder aus einem Quadrupel  $(R, SC, PH, PR)$  für spezielle Fällen bestehen. Dabei steht  $R$  für eine Regel in einer der logischen Programmierung ähnlichen Notation,  $SC$  für eine Menge von Konzepten (Komponenten),  $PH$  für die Phase ihrer Ausführung („Vor“ oder „Nach“ dem Zugriff auf eine Komponente) und  $PR$  besagt, ob die Regel zur Ausführung weiterer Regeln führt („Ja“ oder „Nein“). Die Notation besteht aus einer Vorbedingung

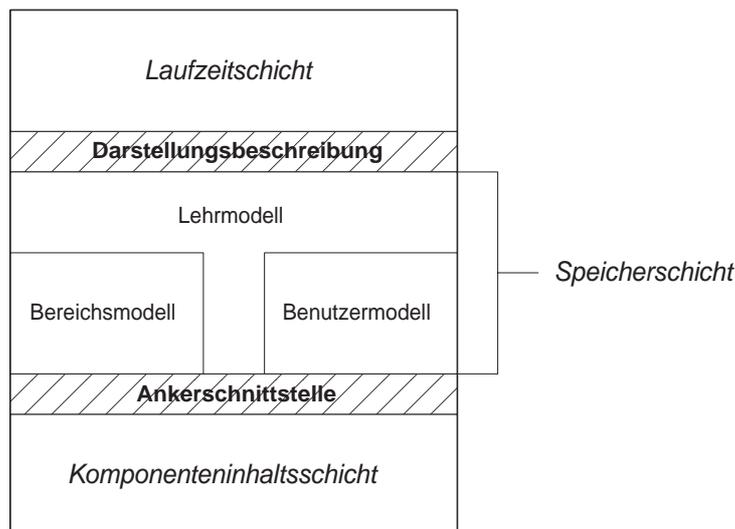


Abbildung 2.6: Das AHAM (Darstellung in Anlehnung an [DeBra99])

(linke Seite) und einer Aktion (rechte Seite). Wird eine Regel R1 aufgerufen und die Vorbedingung erfüllt, so wird die Aktion ausgeführt. Werden dabei Werte geändert, durch die die Vorbedingung einer Regel R2 erfüllt würde, so wird auch R2 aufgerufen, wenn R1 die Ausführung weiterer Regeln erlaubt (PR gleich „Ja“). Existieren mehrere ähnliche Regeln, so werden die speziellen Regeln, also Regeln für spezielle Konzepte, den allgemeinen Regeln vorgezogen.

Der Adaptionsgestalter ist eine Funktion, welche dynamisch Darstellungsbeschreibungen für die Instanzierungsfunktion des Dexter Modells erzeugt.

Das AHAM erweitert die im Abschnitt 2.2.1.3 beschriebene **FolgeVerknüpfung**-Funktion um die Ausführung pädagogischer Regeln. Die Ausführung pädagogischer Regeln erfolgt vor und nach dem Zugriff auf eine Komponente. Der Adaptionsgestalter generiert vor dem Zugriff Darstellungsbeschreibungen, die von der Instanzierungsfunktion zur Adaption genutzt werden.

## 2.2.4 Bewertung der Modelle

Das Dexter Modell dient dem Vergleich von Hypertextsystemen und dem Austausch von Hypertexten zwischen ihnen. Für diese Arbeit ist vor allem der Vergleich von Bedeutung. Da bereits die Autoren des Modells bemerkten, daß das Modell zu mächtig ist, ermutigen sie dazu, das Modell auch für nicht modellkonforme Systeme zu nutzen. Bereiche, in denen das System nicht mit dem Modell übereinstimmt, werden ausdrücklich aufgeführt, und der Rest wird mit dem Modell beschrieben. Der Austausch von Hypertexten auf Basis des Dexter Modells wurde von Leggett und Schnase kritisch bewertet. Sie stellen heraus, daß das Dexter Modell sich als Referenz für den Austausch der Datenbasis eignet, aber nicht als Basis für das Zusammenspiel mehrerer hypermedialer Systeme [Leggett94].

Ein Schwachpunkt des Modells ist die Beschreibung der Laufzeitauflöserfunktion. Das Dexter Modell erlaubt im Zusammenhang mit dieser Funktion die Verwendung dynamischer Verweise. Ein Beispiel für einen dynamischen Verweis ist „die zuletzt dargestellte Komponente mit der Eigenschaft: Typ = Thema“. Gleichzeitig verlangen die Invarianten des Dexter Modells, daß keine baumelnden Verweise innerhalb des Hypertextes existieren. Wie kann die Auflösbarkeit dynamischer Verweise bereits während der Erstellung des Hypertextes gesichert werden? Oder, wenn die dynamischen Verweise zur Laufzeit in den Hypertext unter Wahrung der Invarianten eingefügt werden, wie erfolgt dies? Es wird keine Möglichkeit zur Adaption des Hypertextes im Dexter Modell beschrieben. Auch die **FolgeVerknüpfung**-Funktion wurde nicht vollständig spezifiziert. Allgemein werden die dynamischen Aspekte des Dexter Modell nur unvollständig behandelt, was mit dem Schwerpunkt des Modells auf der Speicherschicht zu erklären ist.

Das Dexter Modell ist ein reines Hypertextmodell, da es keine zeitlichen Informationen modelliert. Allerdings schränkt das Modell auch nicht die Menge der möglichen Medientypen in den Komponenten ein. Da die Medien innerhalb der nicht näher modellierten Komponenteninhaltsschicht behandelt werden, können hier sowohl zeitabhängige als auch zeitunabhängige Medientypen gleichzeitig genutzt werden. Das Modell kann also hypermediale Knoten umfassen, kann aber keine hypermedialen Besonderheiten beachten. Nach

Hardman et al. [Hardman94] wird dies als Hypermediamodellierung über den „Ansatz der versteckten Struktur“ angesehen. Multimediainformationen sind in diesem Fall Teil einer Komponente. Da die beiden betrachteten Anwendungen dieser Arbeit aber hypermediale Datenbasen im Sinne der Definition 3 auf Seite 7 besitzen, könnte dieser Ansatz nicht ausreichen. Deshalb wurde ein anderer Ansatz vorgestellt, das Amsterdamer Hypermediamodell. Dieses betrachtet Zeitinformationen innerhalb des Modells, indem zeitliche Informationen innerhalb von Kompositum-Komponenten modelliert werden. Nach Hardman et al. wird dies als „Ansatz der zusammengesetzten Struktur“ für die Hypermediamodellierung genannt. Durch die Modellierung der zeitlichen Beziehungen ist das AHM komplexer und somit nur vorzuziehen, wenn die Modellierung der zeitlichen Dimension notwendig ist.

Das AHAM ist eine weitere Ergänzung des Dexter Modell. Das Dexter Modell erlaubt die Veränderung der Inhalte und Verweise ausschließlich während der Erstellung eines Hypertextes mit einem Hypertextsystem und modelliert keine Speicherung von Daten über den Benutzer der Anwendung. Adaptive Systeme ändern aber dynamisch während einer Sitzung Inhalte und Verweise und speichern sitzungsübergreifend Informationen über den Benutzer. Das AHAM beschreibt nun, wie das Dexter Modell dennoch für eine solche Modellierung genutzt werden kann, in dem zusätzliche Modellierungsmöglichkeiten eingeführt werden. Das AHAM ist somit spezieller und komplexer als das Dexter Modell, aber für adaptive Systeme vorzuziehen, da ein AHS nicht konsistent mit dem Dexter Modell zu beschreiben ist. Das AHAM könnte, statt auf dem Dexter Modell, auch auf dem Amsterdamer Modell aufsetzen, wenn dies als notwendig erachtet wird.

Mit den drei Modellen dieses Abschnittes besitzt man eine Sammlung von Modellen mit denen hypermediale Systeme beschrieben werden können. Sollte das Dexter Modell nicht flexibel genug sein, so können die anderen Modelle als optionale Erweiterungen herangezogen werden. Da die beiden Ergänzungen unabhängig voneinander sind, spricht nichts gegen ihre Kombination. Aufgrund dieser Kombinierbarkeit und der weiten Verbreitung des Dexter Modells wurde auf das ebenfalls weit bekannte „Tower-Modell“ von De Bra et al., beschrieben in [DeBra92], bewußt verzichtet.

Alle drei Modelle konzentrieren sich auf eine Betrachtung der Speicherschicht. Die dynamischen Aspekte werden, wie zum Beispiel bei der Spezifikation der Laufzeitauflöserfunktion, weitgehend offen gelassen.

## 2.3 Strukturen der Datenbasis

Für die weitere Arbeit sind Strukturen der Datenbasis von Interesse, da sie Hilfsmittel und Navigation beeinflussen können.

Besitzt die Datenbasis weitere Strukturen als Knoten und Verknüpfungen, so wird im folgenden von einer *erweiterten Grundstruktur* gesprochen. Diese Grundstrukturen können beispielsweise hierarchischer Natur sein oder verschiedene Verknüpfungstypen mit unterschiedlicher Semantik beschreiben. Aus der Interpretation der Strukturen durch ein hypermediales System können weitere Informationen über den Inhalt oder implizite Verknüpfungen gewonnen werden. Das Dexter Modell besitzt mit den Kompositum-Komponenten eine hierarchische Strukturierungsmöglichkeit.

Die *Extensible Markup Language* (XML) dient der strukturierten Beschreibung textueller Daten. Sie wurde im Februar 1998 von dem World Wide Web Consortium (W3C) in der ersten Version veröffentlicht [W3C98]. Die Strukturierung erfolgt mit Hilfe von *Markup-Tags*. Markups sind zusammenhängende Paare, die Inhalte umschließen und ineinander verschachtelt werden. Die Bedeutung eines Markups wird nicht festgelegt, sie obliegt dem benutzenden System, genauso wie eine eventuelle Präsentation von Tag-Inhalten.

Abbildung 2.7 zeigt einen Ausschnitt aus einer XML-Datei. Ein Markup steht in spitzen Klammern (< >) und besitzt seine Gültigkeit bis zu seiner Auflösung (</ >). Markups können ineinander verschachtelt werden, wie hier die einzelnen Daten zu einem Buch.

XML stellt mit den *Dokumenttypbeschreibungen* („Document Type Definition“, DTD) einen Mechanismus zur Verfügung, um Beschränkungen der Struktur zu definieren und die Verwendung von vordefinierten Markups zu unterstützen. Die Dokumenttypbeschreibung enthält oder verweist auf Markup-Deklarationen, die eine Grammatik für eine Klasse von Dokumenten bilden (vgl. [W3C98]). DTDs werden in der Regel für bestimmte Aufgabenbereiche eingesetzt, wie zum Beispiel die Mathematical Markup Language zum Setzen von mathematischen Formeln genutzt wird (vgl. [Bosak99]). Die Interpretation der Markups eines DTDs wird in einem solchen Fall festgelegt und beispielsweise als natürlichsprachlicher Text festgehalten.

Die beiden betrachteten Hypermediaanwendungen der nächsten Kapitel basieren auf einer erweiterten Grundstruktur (vgl. Abschnitte 3.2 und 4.2). Die Altenberger Dom Anwendung nutzt zur Beschreibung der Grundstruktur XML, für die Industriemeister 2000 Anwendung wurde eine eigenständige Beschreibungssprache entwickelt, auf die nicht näher eingegangen wird, die aber einfach in ein XML-Format konvertiert werden könnte.

Aus der Struktur der Datenbasis können Informationen über den Inhalt entnommen werden. So schreibt zum Beispiel Tergan [Tergan97]: „Eine hierarchische Struktur eignet sich beispielsweise zur Repräsentation unterschiedlicher Ebenen der Abstraktheit, Feinkörnigkeit und Bedeutsamkeit der Inhalte eines Gegenstandsbereichs.“ Nürnberg et al. gehen noch einen Schritt weiter, in dem sie behaupten, daß die Aufteilung in Knoten und

```
<book>
  <title>Discover Java</title>
  <author>Ed Tittel</author>
  <author>Bill Brogden</author>
  <publisher>IDG Books Worldwide</publisher>
  <ISBN>0-7645-8024-8</ISBN>
  <pagecount>316</pagecount>
  <price>$24.99</price>
  <pubdate>1997</pubdate>
  <recommendation>Buy It????</recommendation>
</book>
```

Abbildung 2.7: Eine Buchbeschreibung in XML

Verknüpfungen sogar hinderlich sei und das Daten stärker strukturiert werden sollten [Nürnberg 97]. Die designierte Websprache XML beschreitet genau diesen Weg, indem Daten nicht mehr, wie mit HTML üblich, graphisch aufbereitet werden, sondern inhaltlich strukturiert werden.

## 2.4 Navigation

Unter Navigation wird im allgemeinen die Bewegung des Benutzers innerhalb eines Hyperdokumentes verstanden. In vielen Arbeiten wird der Begriff der Navigation als allgemein bekannt vorausgesetzt (beispielsweise Lowe und Hall [Lowe99] und Faber [Faber93]). Dabei ist der Begriff der Navigation nicht eindeutig belegt. Für das unterschiedliche Verständnis von Navigation seien beispielhaft die Definitionen und begrifflichen Abgrenzungen von Mühlhäuser und Tergan vorgestellt.

Nach Mühlhäuser ist Navigation eine Weise des Lesens eines Hyperdokumentes. Navigation wird als ein planmäßiges Durchlaufen und Lesen von Knoten verstanden, das mit dem Lesen in Papierdokumenten vergleichbar ist. Hingegen entspricht Stöbern („browsing“) dem Blättern innerhalb eines Buches mit dem Ziel, einen Überblick über Inhalt und interessante Knoten zu gewinnen. Als dritte Möglichkeit des Lesens eines Hyperdokumentes nennt Mühlhäuser noch das Abfragen eines Hyperdokumentes [Mühlhäuser97a].

Tergan definiert Navigation als Mittel zum Informationszugriff, der wiederum in Browsing (auch Stöbern genannt), Gezielte Suche und Folgen von Pfaden unterteilt wird. Weiterhin wird von Tergan zwischen *interner* und *externer Navigation* unterschieden, wobei interne Navigation auf Grundlage von Verknüpfungen innerhalb des Hyperdokumentes und externe Navigation unter Nutzung von externen Orientierungshilfen wie Stichwort- und Inhaltsverzeichnissen oder graphischen Übersichten definiert wird [Tergan97].

Zusammenfassend ist nach Mühlhäuser Navigation eine Art des Lesens, nach Tergan ist Navigation eine Grundlage für den Informationszugriff. Dabei wird Navigation auch als Grundlage für Stöbern verstanden, während Mühlhäuser Stöbern wiederum als eine der Navigation gleichgestellte Art des Lesens bezeichnet.

Im folgenden wird sich der Meinung von Tergan angeschlossen, Navigation als eine Grundlage für den Informationszugriff durch den Benutzer zu verstehen. Dies wird aus dem Blickwinkel der technischen Realisierung geschehen, d.h., daß die Grundlage für den Informationszugriff als ein implementierbarer Mechanismus verstanden wird. Dies ist für die Arbeit aus nachfolgenden Gründen sinnvoll:

- Eine Definition der Navigation über das Leseverhalten von Benutzern beschreibt nicht die Navigation speziell für die beiden betrachteten Systeme. Aus welcher Motivation heraus ein Benutzer die Knoten betrachtet, ist abhängig von den Inhalten, die dargestellt werden, nicht aber von dem darstellendem System. Das System kann zwar spezielle Möglichkeiten zur Verfügung stellen, die es dem Benutzer erlauben, sich einen Überblick über die Informationen zu verschaffen, es kann aber das Verhalten des Benutzer kaum beeinflussen.

- Eine Betrachtung des Benutzerverhaltens könnte zu keinem Ergebnis führen, weil keine Informationen über das Benutzerverhalten mit den beiden Anwendungen existieren. Da sich beide Anwendungen zur Zeit noch in der Realisierungsphase befinden, können auch keine Informationen rechtzeitig bereitgestellt werden.
- Zur Beschreibung der beiden Anwendungen wird das Dexter Modell herangezogen. Dieses betrachtet die technische Realisierung auf einem abstraktem Niveau. Das Dexter Modell eignet sich somit als Grundlage, um einen Mechanismus zu beschreiben, der ein Teil eines hypermedialen Systems ist.

Somit ist für die weiteren Betrachtungen die Motivation des Benutzers zur Navigation nicht wichtig.

Der Begriff „Navigation“ wurde der *Navigationsmetapher* entnommen, die im folgenden eingehender besprochen wird.

### 2.4.1 Die Navigationsmetapher

Die Navigationsmetapher vergleicht die Bewegung innerhalb eines Hyperdokumentes mit der Navigation eines Schiffes. Nielsen [Nielsen90] schreibt zum Beispiel: „to help users navigate the hyperspace we must help them understand and recognize their present location“.

Diese Analogie wird von Geri Gay und Joan Mazur fortgesetzt. Sie vergleichen den Benutzer eines hypermedialen Systems mit dem Steuermann eines Schiffes. Der Benutzer des hypermedialen Systems, und nicht der Autor der Datenbasis, bestimmt den Kurs durch das Meer der Informationen. Der Benutzer steht am Steuerruder und entscheidet, welche Inhaltsinseln angesteuert oder umschifft werden. So wie der Kartograph sich der Größe der Meere bewußt sein sollte und unabhängig von den verschiedenen Routen seine Karten erstellen sollte, so sollte auch der Entwickler eines hypermedialen Systems sich des Umfanges seines Systems bewußt werden und dem Benutzer Hilfsmittel zur einfachen Verwendung zur Verfügung stellen. Ebenso wie ein Kartograph bestimmt auch der Entwickler nicht den Kurs, sondern die Möglichkeiten. Der Benutzer wird letztendlich auf sich selber gestellt sein [Gay91].

Ein modernes Beispiel für die Verwendung der Navigationsmetapher ist das hypermediale System Netscape Navigator [Pott97] für das World Wide Web. Das als Browser bezeichnete System enthält bereits im Namen das Wort „Navigator“ und benutzt als Programmsymbol das Steuerrad eines Schiffes. Das Programmpaket für das Internet, in das der Navigator eingebunden ist, zeigt einen Leuchtturm als Programmsymbol.

Dieser bildliche Vergleich zwischen dem Benutzer eines hypermedialen Systems und dem Steuermann eines Schiffes kann intuitiv eingängig sein, doch sind die Möglichkeiten von Hypertext und Hypermedia, wie zum Beispiel eine beliebige Strukturierung der Datenbasis, nicht vollständig in die Begriffswelt von Meeren und Schiffen übertragbar.

### 2.4.2 Desorientierung

*Desorientierung* („lost-in-hyperspace“) ist ein Problem das häufig im Zusammenhang mit Navigation in Hyperdokumenten genannt wird. Desorientierung bedeutet, daß der Benutzer den Überblick über seinen Standort innerhalb des Hyperdokumentes verloren hat. Desorientierung äußert sich darin, daß der Benutzer keine Möglichkeit sieht, die Informationen eines Hyperdokumentes zu finden, die er sucht und von denen er weiß, daß diese im Dokument enthalten sind. Dies kann seinen Ursprung an einer Unkenntnis der Navigationsmöglichkeiten oder einer mangelnden Übersichtlichkeit der Knoten des hypermedialen Systems haben (vgl. [Tergan97]).

### 2.4.3 Hilfsmittel der Navigation

Um diesem Problem Abhilfe zu schaffen, wird Navigation durch Hilfsmittel unterstützt. Die Betrachtung dieser Hilfsmittel ist ein Schwerpunkt dieser Arbeit. Sie werden deshalb nun vorgestellt und definiert.

Die Bezeichnung „Hilfsmittel“ wurde gewählt, weil sie dem Benutzer einer hypermedialen Anwendung Hilfestellung geben, die nicht im Dexter Modell beschrieben wird. Sie können dem Benutzer Informationen über seinen Standort innerhalb des Hyperdokumentes präsentieren und ihn gezielt zu Knoten leiten. Sie dienen damit der Navigation. Da sie nicht von dem Dexter Modell modelliert werden, sind sie für die Basisfunktionalität einer hypermedialen Anwendung, im Sinne des Dexter Modells, nicht notwendig.

Hilfsmittel werden von Gay und Mazur ebenfalls behandelt. Sie definieren Hilfsmittel („tools“) aber nicht, sondern nutzen die Navigationsmetapher um einen Eindruck von Hilfsmitteln zu erhalten. Sie vergleichen Hilfsmittel mit Meereskarten [Gay91]. Dieser Vergleich ist aber nicht sehr zweckdienlich für die weitere Arbeit, da Meereskarten passive Hilfsmittel sind, während die im folgenden besprochenen Hilfsmittel dem Benutzer auch Navigationsmöglichkeiten (externe Navigation nach Tergan) zur Verfügung stellen. Deswegen wird nun eine eigene Definition vorgestellt:

#### **Definition 5 (Hilfsmittel der Navigation)**

*Ein Hilfsmittel der Navigation ist ein Bestandteil einer hypermedialen Anwendung, welches einen Bezug zum Hyperdokument besitzt und dem Benutzer Funktionalität zur Verfügung stellt.*

Diese Definition umfaßt folgende wesentliche Punkte:

**Bestandteil einer hypermedialen Anwendung:** Ein Hilfsmittel ist integriert in einer hypermedialen Anwendung. Es ist keine externe Anwendung, welche mit derselben Datenbasis arbeitet.

**Bezug zum Hyperdokument:** Ein Hilfsmittel verarbeitet statische oder dynamische Informationen aus Datenbasis oder Sitzung. Es kann statische Informationen aus

der Datenbasis verarbeiten, wie beispielsweise Komponenteneigenschaften, Komponentenverweise und Strukturinformationen (vgl. Abschnitt 2.3) oder dynamische Informationen, die Teil einer Sitzung sind, wie aktuell dargestellte Instanzen. Es ist also ein Bezug zum Hyperdokument oder seiner Darstellung gegeben. Somit sind beispielsweise Teile einer Anwendung, die Kommunikation mit anderen Benutzern des Systems erlauben, keine Hilfsmittel der Navigation.

**Benutzerfunktionalität:** Der Benutzer kann die Funktionalität des Hilfsmittels nutzen. Ein Bestandteil einer Anwendung, welches Benutzertätigkeit protokolliert und die so gewonnenen Informationen einer dritten Person zur Verfügung stellen würde, wäre kein Hilfsmittel.

#### 2.4.4 Arbeitshypothese der Navigation

Obschon das Ziel der Arbeit die Definition von Navigation für die beiden betrachteten Systeme ist, wurde der Begriff „Navigation“ bereits zuvor und wird auch im folgenden genutzt. Um diesem Umstand Rechnung zu tragen, wird an dieser Stelle eine Arbeitshypothese vorgestellt, welche für die folgenden Betrachtungen hinreichend ist und durch eine abschließende Definition gegen Ende der Arbeit ersetzt wird. Zur Beschreibung der beiden Anwendungen wird das Dexter Modell herangezogen, deshalb basiert auch die Arbeitshypothese auf dem Dexter Modell.

Navigation wird zusammengesetzt aus Navigationsschritten. Bevor nun die Arbeitshypothese der Navigation aufgestellt wird, wird eine solche für Navigationsschritte formuliert:

##### **Arbeitshypothese 1 (Navigationsschritt)**

*Ein Navigationsschritt ist ein Wechsel des aktuell dargestellten Knotens mit der Folge-Verknüpfung-Funktion des Dexter Modells.*

Damit wird unter einem Navigationsschritt eine Operation des hypermedialen Systems verstanden, die dem Benutzer einen Knoten präsentiert. Der Sonderfall, in dem mehrere Knoten gleichzeitig präsentiert werden (zum Beispiel im Falle einer mehrwertigen Verknüpfung), wird nicht näher betrachtet, da die Instanzierungsfunktion diesen Fall behandelt. Das Dexter Modell sieht in einem solchen Fall mehrere gleichzeitig präsentierte Instanzen vor.

Die Navigationsschritte während einer Sitzung (das Dexter Modell speichert keine Daten zwischen Sitzungen) bilden zusammen die Navigation:

##### **Arbeitshypothese 2 (Navigation)**

*Navigation ist die sequentielle Abfolge der Navigationsschritte während einer Sitzung.*

Diese Arbeitshypothese beschreibt Navigation ausschließlich als Folge einer aktivierten Verknüpfung. Es wird nicht auf Navigation mit Hilfsmitteln eingegangen. An dieser Stelle setzt diese Arbeit an: Wie können Hilfsmittel mit in die Navigation einbezogen werden? Die Folge-Verknüpfung-Funktion wird weder durch Hilfsmittel angesteuert, noch werden Hilfsmittel über einen Navigationsschritt informiert. Die Integration der Hilfsmittel in die

Navigation und eine Erweiterung des Dexter Modells werden im folgenden Teilziele der Arbeit darstellen, um darauf aufbauend Navigation definieren zu können.

---

## Kapitel 3

# Die hypermediale Industriemeister 2000 Anwendung

Dieses Kapitel umfaßt eine Beschreibung des Industriemeisters 2000, einer hypermedia-  
len Anwendung. Die Darstellung des Projektes in Abschnitt 3.1 dient als Überblick und  
Kontext für die weitere Darstellung. Die Analyse der Datenbasis im Abschnitt 3.2 umfaßt  
eine Beschreibung ihrer Struktur und einen Vergleich mit dem Dexter Referenzmodell.  
Das hypermediale System wird im Abschnitt 3.3 mit dem Dexter Modell und dem AHAM  
verglichen. Der Schwerpunkt dieses Kapitels liegt auf der Dokumentation der Hilfsmittel  
(siehe Abschnitt 3.4). Die beschriebenen Hilfsmittel nutzen Strukturen der Datenbasis und  
arbeiten mit dem hypermedialen System zusammen.

Die Ergebnisse dieses Kapitels werden in Kapitel 5 mit den Ergebnissen des Kapitels 4  
verglichen. Damit ein solcher Vergleich möglich ist, werden Referenzmodelle genutzt.

### 3.1 Projektbeschreibung

Im folgenden wird das Industriemeister 2000 Projekt beschrieben, dessen Teilziel die Er-  
stellung einer hypermedialen Anwendung ist, auf die in diesem Kapitel näher eingegangen  
wird. Weitere Angaben zu dem Projekt dienen der besseren Einordnung der Anwendung  
und der Begriffsbildung.

Das Projekt „Aufbau multimedialer Netze zur Entwicklung und Evaluation von Modulen  
im naturwissenschaftlichen Bereich für das DIHT/IHK Konzept Industriemeister 2000“,  
kurz Industriemeister 2000, begann am 1. Oktober 1996 und wurde am 31. August 1999  
abgeschlossen. Das Projektziel bestand darin, „multimediale Bestandteile in die Weiterbil-  
dung zum Industriemeister zu integrieren und sie über schnelle Datennetze verfügbar zu  
machen“ [IHK96]. Die Produkte des Projektes sind ein rechnergestütztes Lernsystem und  
die während der Entwicklung entstandenen Konzepte, Methoden, Entwürfe und sonstige  
Ergebnisse [ICD98, Seite 9].

Für die weitere Betrachtung ist im besonderen Maße das rechnergestützte Lernsystem von Interesse. Die anderen Produkte sind für die weitere Arbeit nicht wichtig. Sie enthalten keine für diese Arbeit interessanten Ergebnisse oder Entwurfsentscheidungen.

### 3.1.1 Zielsetzung des rechnergestützten Lernsystems

Die Zielsetzung des Lernsystems ist in der Gesamtkonzeption/Anforderungsdefinition für das Projekt Industriemeister 2000 beschrieben (siehe [ICD98, Kapitel 2.1 - 2.3]). Die für das rechnergestützte Lernsystem wichtigen Kernaussagen aus diesen Abschnitten lauten:

- „Das rechnergestützte Lernsystem wird in der Aus- und Weiterbildung eingesetzt.“
- „Das Lernsystem besteht aus mehreren Grundsystemen, aus multimedialen Lerneinheiten zur Unterstützung konkreter Industriemeister 2000-Lehrgänge sowie aus Komponenten zur Erstellung und Pflege von Lehrgangseinheiten.“
- „Die Grundsysteme für Lernende, Lehrende und sonstige Anwendergruppen dienen dazu, multimediale Bestandteile von Kursen oder Lehrgängen mit Unterstützung von Diensten auf der Basis von Datennetzen verfügbar zu machen.“
- „Die multimedialen Lerneinheiten unterstützen die Vermittlung von naturwissenschaftlichen Grundlagen – Physik und Chemie – im Rahmen von Lehrgängen der Weiterbildung zum Industriemeister Metall und Elektrotechnik.“
- „Das Lernsystem wird als Lern-Informationssystem konzipiert, das eine ausgewogene Mischung an Informationsseiten, Lernseiten und Simulationen umfaßt.“
- „Das Lernsystem bietet dem Lernenden grundsätzlich die Möglichkeit zur freien Navigation. Die Navigation sollte durch zusätzliche Möglichkeiten wie geführte Touren, Inhaltsverzeichnisse, Übersichtsdiagramme, Lesezeichen, etc. unterstützt werden.“
- „Durch das Lernsystem werden die derzeit gängigen Kommunikations- und Datenübertragungsmöglichkeiten unterstützt. Hierbei ist primär die Verfügbarkeit von schnellen Datennetzen vorauszusetzen. Es sind jedoch zusätzlich auch Möglichkeiten für die Nutzung des Lernsystems in Netzen mit geringen Bandbreiten oder im Offline-Betrieb vorzusehen.“

Die verschiedenen Grundsysteme sollen unterschiedliche Funktionalitäten für die Anwendergruppen zur Verfügung stellen. Als Anwendergruppen oder Benutzergruppen werden Teilnehmer (auch Lernende genannt), Dozenten (Lehrer), Veranstalter, Autoren, Administratoren und Wartungspersonal genannt. Verschiedene Benutzergruppen sollen verschiedene Grundsysteme nutzen, um unterschiedliche Aufgaben mit dem System zu bewältigen. So erstellen zum Beispiel Autoren neue Inhalte, welche die Administratoren und das Wartungspersonal den Teilnehmern zur Verfügung stellen.

Die zentrale Zielgruppe des Lernsystems ist die der Lernende. Der Lernende kann mit der Hypermediaanwendung des Industriemeisters 2000, also einem Teil des Lernende-Grundsystems, begleitend zum Unterricht Lehrstoff aufarbeiten und sich auf Prüfungen vorbereiten, wozu er eine möglichst lückenlose Aufbearbeitung des Lehrstoffes benötigt. Weitere



Abbildung 3.1: Die Oberfläche des Lernsystems

Funktionalität des Lernende-Grundsystems dient der Kommunikation und Kooperation mit anderen Benutzer.

Anstatt der ursprünglich vorgesehenen zwei Lehrgänge, Physik und Chemie, wurde nur der Lehrgang Physik erstellt.

### 3.1.2 Die Realisierung

Die weiter unten vorgenommene Beschreibung des hypermedialen Systems mit Hilfe des Dexter Modells setzt Erläuterungen voraus, weshalb auch nur die für das Dexter Modell wichtigen Bereiche hier abgedeckt werden.

Das hypermediale System des Industriemeisters 2000 ist Teil des Lernende- und des Lehrer-Grundsystems, die sich nicht in der Darstellung der Lehreinheiten, sondern im Zugriff auf Teilnehmerdaten unterscheiden. Das Erstellen von neuen Lerneinheiten benötigt ein spezielles Autoren-Grundsystem, welches sich stark von den beiden oben erwähnten Grundsystemen unterscheidet. Die multimedialen Lernseiten bilden zusammen ein Hypermediadokument.

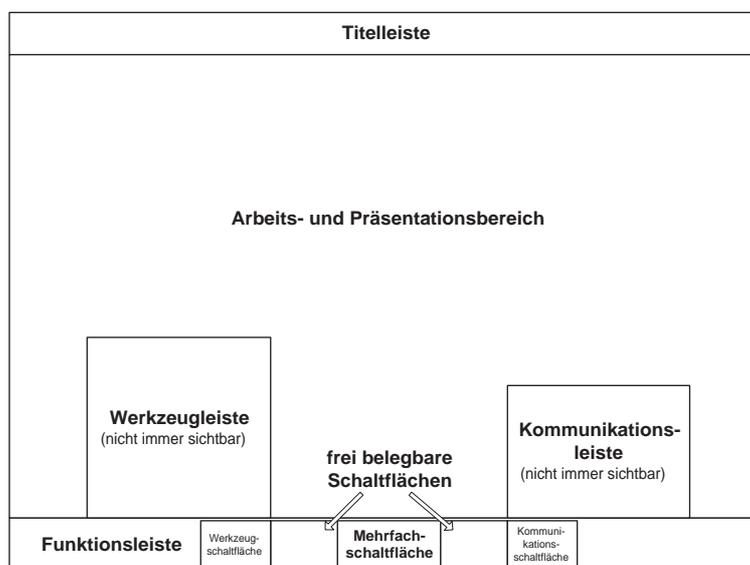


Abbildung 3.2: Schematische Ansicht der Oberfläche des Lernsystems

Das hypermediale System des Industriemeisters 2000 wurde als adaptives hypermediales System (vgl. Abschnitt 2.1.4) realisiert. Für jeden Teilnehmer wird zentral ein Benutzermodell gespeichert.

Als Plattform für das Lernsystem wurde Windows (95, 98 und NT) ausgewählt. Die einzelnen Kurse werden gesondert vertrieben und jeweils auf einer CD ausgeliefert. Das Hyperdokument liegt somit lokal vor und es wird nicht über ein Netzwerk gelesen.

### 3.1.2.1 Oberfläche

Die Oberfläche des rechnergestützten Lernsystems (siehe Abbildung 3.2) kann in drei horizontale Bereiche unterteilt werden. Der oberste Bereich, die *Titelleiste*, zeigt Informationen über die aktuell dargestellte multimediale Lerneinheit. Der unterste Bereich, die *Funktionsleiste*, ermöglicht den Zugriff auf die Dialoge der Hilfsmittel und der Kommunikationswerkzeuge (Email, Online-Konferenzen und Newsgroups), auf einen integrierten WWW-Browser sowie auf Dialoge zur individuellen Anpassung des Systems. Auf der Funktionsleiste befinden sich zwei frei belegbare Schaltflächen, die mit verschiedenen Funktionalitäten belegt werden können. Die meisten Hilfsmittel können über die *Werkzeugleiste* erreicht werden. Diese wird mit der *Werkzeugschaltfläche* links auf der Funktionsleiste ein- und ausgeblendet. Mit der *Kommunikationsschaltfläche* kann die *Kommunikationsleiste* ein- und ausgeblendet werden. Über diese Leiste kann der Benutzer Kommunikationsmöglichkeiten nutzen. Der mittlere Bereich dient der Darstellung der multimedialen Lerneinheiten. Er wird *Arbeits- und Präsentationsbereich* genannt.

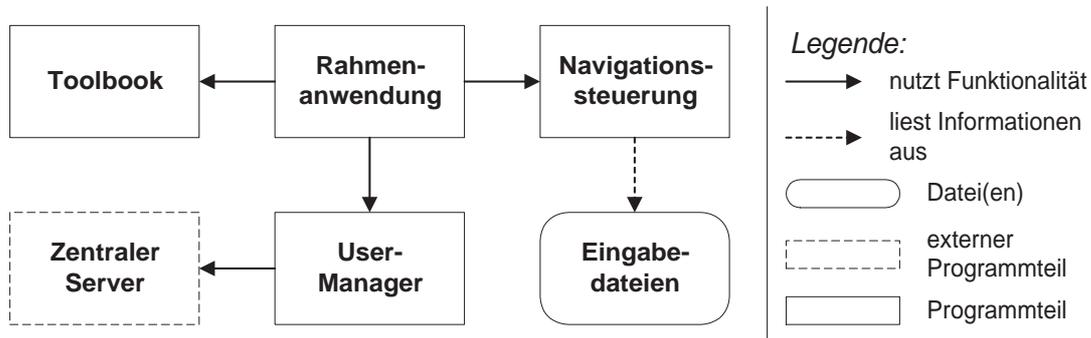


Abbildung 3.3: Die Architektur des adaptiven hypermedialen Teilsystems des Industriemeisters 2000

### 3.1.2.2 Architektur

Die multimedialen Lerneinheiten, also das Hypermediadokument, werden mit *Toolbook* [Toolbook97] erstellt. Toolbook ist ein Autorensystem der Firma Asymetrix. Es basiert auf der Buchmetapher, welche die Erstellung einer multimedialen Anwendung mit dem Erstellen von Seiten eines Buches vergleicht. Eine mit Toolbook erstellte Datenbasis besteht aus gleich großen *Seiten* („pages“) innerhalb einer größeren Zuordnungseinheit: dem *Buch* („book“).

Toolbook erlaubt die Erstellung und Darstellung von Seiten. Eine Seite umfaßt verschiedene Medien und interaktive Elemente wie Knöpfe und „Drag & Drop“. Mit Hilfe der Scriptsprache *Openscript* kann die Seite ein bestimmtes Verhalten erhalten. So kann auf Benutzeraktionen mit dem Abspielen von Tonaufnahmen oder mit neuen und geänderten Texten reagiert werden. Jede Toolbook-Seite befindet sich in einem Buch. Bücher umfassen im Allgemeinen mehrere Seiten. Seiten besitzen eine eindeutige Identifizierung innerhalb eines Buches. Bücher werden auf Dateiebene als einzelne Datei gespeichert. Toolbook ermöglicht die Darstellung von mit Toolbook erstellten und veränderten Seiten. Ein Wechsel der darzustellenden Seite erfolgt mit dem Openscript-Befehl: `go to page PAGENAME of book BOOKNAME`, wobei `PAGENAME` die Identifizierung der Seite und `BOOKNAME` der Dateiname des Buches ist. Toolbook arbeitet in zwei Modi. Der *Autorenmodus* („author-mode“) erlaubt das Erstellen und Löschen von Seiten sowie das Verändern von Seitenbeschreibungen, der *Lesermodus* („reader-mode“) nur Betrachtung und Interaktion mit einer Seite, nicht deren Veränderung.

Im folgenden werden die einzelnen Bestandteile der groben Architektur des Industriemeisters 2000 erläutert (siehe Abbildung 3.3).

Die *Navigationssteuerung* implementiert die Auflöserfunktion und die Laufzeitauflöserfunktion. Die dafür benötigte Datenbank besteht aus den *Eingabedateien*. Eine Eingabedatei beschreibt den Aufbau eines Kurses (vgl. Abschnitt 3.2).

Für die Verwaltung der verschiedenen Benutzermodelle jedes Teilnehmers ist der *zentrale Server* zuständig, der weltweit durch das Internet erreicht werden kann. Der *User-Manager*

verwaltet das jeweils aktuelle Benutzermodell und kümmert sich um die Verbindung zum zentralen Server. Die Benutzermodelle können lokal in einer Datei gespeichert werden und anhand eines Datenträgers zu einem anderen Rechner kopiert werden. So kann ein Benutzer sein Benutzermodell auch dann verwenden, wenn er sich an einem Rechner befindet, der keinen Zugang zu dem zentralen Server besitzt.

Die Verbindung zwischen Toolbox, dem User-Manager und der Navigationssteuerung erfolgt durch einen weiteren Bestandteil des Lernsystems: Die *Rahmenanwendung* nutzt Toolbox sowie die Navigationssteuerung und stellt eine Oberfläche zur Verfügung, mit der auf die Funktionalität des Systems zugegriffen werden kann. Die Rahmenanwendung erlaubt den Zugriff auf bestimmte Seiten mit Hilfe von Toolbox, verwaltet die Benutzerdaten und stellt Möglichkeiten zur Kommunikation zur Verfügung. Damit umfaßt die Rahmenanwendung auch Funktionalität, die nicht zu dem hypermedialen System gehört.

## 3.2 Die Datenbasis

Die Strukturen der Eingabedateien werden nun erläutert. Auf diesen Strukturen bauen das hypermediale System und viele Hilfsmittel auf.

Die Datenbasis wurde von der Benutzergruppe der Autoren in zwei Phasen erstellt. Das Ergebnis der ersten Phase war im wesentlichen ein *Drehbuch*, welches dann in einem zweiten Schritt mit Toolbox umgesetzt wurde. Der Inhalt der Datenbasis ist der Lehrgang Physik.

Grundlegend für die Strukturierung der Datenbasis sind Seiten, die den Knoten des Hyperdokumentes entsprechen. Jede Seite gehört (nicht transitiv gesehen) zu höchstens einem Kapitel und kann zu beliebig viele Geführten Touren gehören. Ein Kapitel muß mindestens eine Seite umfassen und kann Kapitel als Unterkapitel besitzen. Das Kapitel ist somit ein rekursiver Typ. Geführte Touren sind eine Menge von Seiten mit einer Reihenfolge.

Die Seiten und Unterkapitel eines Kapitels besitzen ebenfalls eine Reihenfolge, wodurch eine Ordnungsrelation induziert wird. Ein Kapitel wird als Aufzählung seiner Seiten und Unterkapitel beschrieben. Die Reihenfolge der Aufzählung entspricht der Ordnungsrelation. Seiten stehen innerhalb der Reihenfolge vor den Unterkapiteln. Für jeweils zwei Seiten, die direkt oder transitiv zu einem Kapitel  $Z$  gehören, existiert also eine Ordnung der Reihenfolge. Eine Seite  $A$  befindet sich vor einer Seite  $B$  im Sinne dieser Ordnungsrelation, wenn eine Tiefensuche durch alle Seiten und Unterkapitel (in dieser Reihenfolge) von  $Z$  die Seite  $A$  vor der Seite  $B$  besucht. Insbesondere hat jedes Kapitel eine erste und eine letzte Seite. Seiten, Kapitel und Geführte Touren besitzen jeweils einen Titel. Dieser Titel ist eine zusätzliche Darstellungsbeschreibung.

Die Datenbasis wurde in Kurse unterteilt. Ein Kurs ist ein Fachgebiet des Lehrgangs Physik, zum Beispiel der Kurs „Kinematik“. Der Benutzer des Systems kann an verschiedenen Kursen teilnehmen. Die Menge der Kurse, die einem Lernende somit zur Verfügung steht, ergibt sich aus der Schnittmenge der Kurse, an denen er teilnimmt und der auf dem Rechner lokal installierten. Ein Kurs ist in sich geschlossen, Verknüpfungen über Kursgrenzen hinweg erfolgen nur in den Glossar-Kurs hinein, welcher jedem Benutzer zur Verfügung

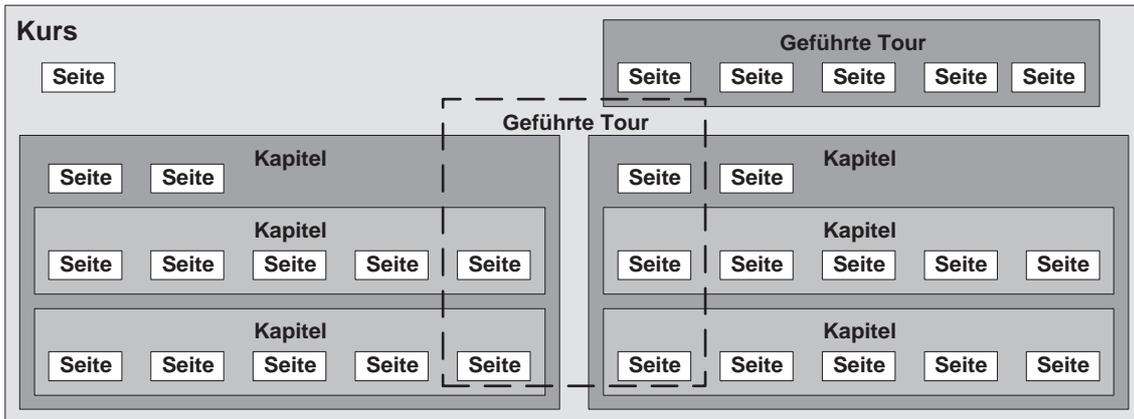


Abbildung 3.4: Schematische Darstellung einer Eingabedatei

steht. Ein Kurs ist intern ein Kapitel. Somit existiert zwischen den Seiten eines Kurses eine Ordnung der Reihenfolge. Es existiert also zu jeder Seite eine vorherige und nachfolgende, mit Ausnahme der ersten und letzten Seite eines Kurses.

Ein Kurs wird mitsamt seinen Seiten, Kapiteln und Geführten Touren in einer Eingabedatei deklariert. Dabei wird festgelegt, welche Seite in welchem Toolbook-Buch unter welcher Bezeichnung zu finden ist, welche Seite zu welchem Kapitel oder zu welcher Geführten Tour gehört, sowie Titel der Seiten und die Hierarchie der Unterkapitel. Abbildung 3.4 stellt eine Kursbeschreibung schematisch dar. Der Kurs umfaßt mehrere hierarchisch gegliederte Kapitel und zwei Geführte Touren. Davon besitzt eine Tour eigene Seiten und eine zweite Tour benutzt Seiten aus verschiedenen Kapiteln.

Die Dateien aller auf einem Rechner installierten Kurse bilden zusammen die Datenbank der Speicherschicht. Mit ihrer Hilfe arbeitet die Navigationssteuerung. Jede Seite entspricht einer Komponente und erhält einen innerhalb des Kurses eindeutigen Namen, die Komponenten-ID. Verknüpfungen sind Teil der Komponenteninhalte und werden nicht als eigene Komponenten verwaltet. Statt dessen ruft Toolbook eine Funktion der Rahmenanwendung auf, wenn eine dargestellte Ankerinstanz aktiviert wurde und übergibt als Parameter einen Komponentenverweis. Ein solcher Verweis besteht aus einer Komponenten-ID. Für einen Verweis in den Glossar-Kurs hinein wird eine zweite Funktion aufgerufen, die ebenfalls nur eine Komponenten-ID als Verweis übergeben bekommt.

Die meisten Seiten benutzen ein Konzept, das *Lernschritte* genannt wird. Ein Lernschritt ist eine Darstellungsbeschreibung einer Seite. Wird eine Seite zum erstenmal für einen Benutzer dargestellt, so befindet sie sich im Lernschritt Eins. Durch Interaktion des Benutzers mit der Seite kann diese Seite einen neuen Lernschritt anzeigen, womit sich die Darstellung einer Seite verändern kann. Auf einer Seite kann dem Lernende beispielsweise eine komplexe Aufgabe gestellt und immer, wenn der Benutzer ein Teilergebnis korrekt ermittelt hat, der nächste Aufgabenteil durch die Darstellung des nächsten Lernschrittes angezeigt werden. Ein Lernschritt ist zudem über die untere der beiden mittleren Schaltflächen der Mehrfachschaltfläche auslösbar (siehe Abbildung 3.11). Diese Schaltfläche besitzt einen

nach unten weisenden Pfeil als Symbol und kann nur betätigt werden, wenn die Seite noch nicht ihren letzten Lernschritt anzeigt.

### 3.3 Das hypermediale System

Aufbauend auf der beschriebenen Datenbasis erfolgt nun die Darstellung des hypermediale Systems des Industriemeisters 2000, welche mit dem Dexter Modell (siehe Abschnitt 2.2.1) und dem AHAM (siehe Abschnitt 2.6) erfolgt.

Das Dexter Modell (siehe Abschnitt 2.2.1) dient dem Vergleich von Hypertextsystemen. Das Hypermediasystem des Industriemeisters 2000 benutzt Toolbook zur Darstellung der Knoten. Toolbook verwaltet die Inhalte und die Struktur eines Knotens und synchronisiert zeitabhängige Medien. Dies entspricht dem „Ansatz der versteckten Struktur“ zur Beschreibung von Hypermedia (vgl. Abschnitt 2.2.4). Damit kann das System ohne spezielle hypermediale Gesichtspunkte beschrieben werden, und auf die Nutzung des komplexeren AHMs kann verzichtet werden.

Das AHAM wird nur für die speziellen Bereiche der Adaption genutzt. Danach wird sich auf das Dexter Modell als Referenzmodell beschränkt. Dies bedeutet zum Beispiel, daß auch weiterhin der Begriff Komponente genutzt wird und nicht der Begriff Konzept aus dem AHAM. Diese Vereinfachung ist möglich, da die Adaptivität nicht wesentlich das Verhalten des Systems beeinflusst.

Für die restliche Betrachtung wird somit das weniger komplexe Dexter Modell genutzt. Wenn im folgenden das AHAM genutzt wird, so wird dies explizit erwähnt werden.

#### 3.3.1 Adaption

Das Benutzermodell des Industriemeisters 2000 ist nicht sehr umfangreich. Nur wenige Informationen werden gesammelt und es existieren keine komplexen pädagogischen Regeln, da die Adaption als Teil der Komponenten realisiert wurde.

Die meisten Seiten besitzen mehrere Lernschritte (siehe Abschnitt 3.2). Welche Seite sich für einen bestimmten Benutzer in welchem Lernschritt befindet, ist Teil des Benutzermodells. Für jeden Benutzer wird weiterhin protokolliert, welche Seiten er bereits besucht und welche Seiten er bereits vollständig bearbeitet hat. Letzteres kann dadurch bestimmt werden, daß sich die Seite im letzten Lernschritt befindet. Weitere Daten des Benutzermodells sind Werte, die der Benutzer in dafür vorgesehene Eingabeflächen (Textfelder, Auswahlschaltflächen, u. s. w.) eingetragen hat und die auch beim nächsten Besuch der Seite wieder dargestellt werden sollen.

Das Lehrmodell umfaßt Regeln zur Aktualisierung des Benutzermodells. Die Darstellung der Seiten wird durch das Benutzermodell beeinflusst, da das Benutzermodell den aktuellen Lernschritt einer Seite verwaltet. Die pädagogischen Regeln des Industriemeisters 2000 wurden als Programmcode innerhalb der Rahmenanwendung niedergeschrieben. Auch sie werden in zwei Phasen bearbeitet: Vor dem Zugriff (Sichern der Benutzerwerte) und nach diesem (Anzeige des aktuellen Lernschrittes).

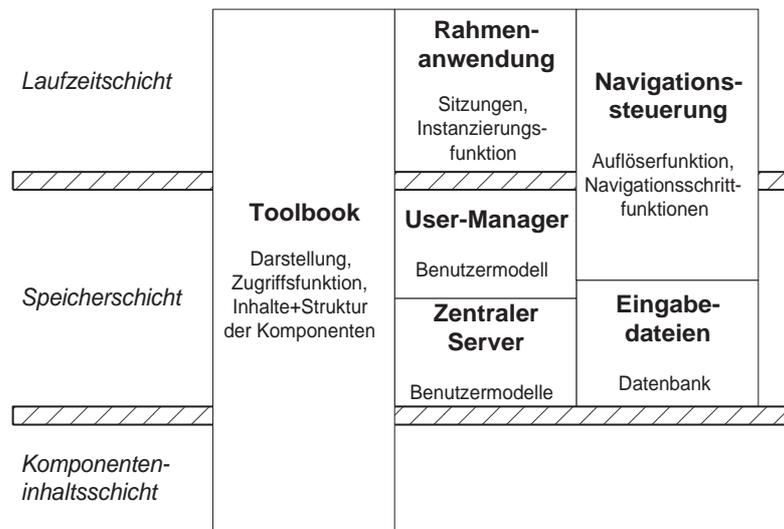


Abbildung 3.5: Das hypermediale System des Industriemeisters 2000 aufgeteilt auf die Schichten des Dexter Modells.

Die eigentliche Adaption erfolgt in der Komponenteneinhaltsschicht. Mit Openscript wird das Aussehen und Verhalten in Abhängigkeit der Lernschrittnummer adaptiert.

### 3.3.2 Die Realisierung der Schichten des Dexter Modells

Die Aufteilung der einzelnen Subsysteme auf die Schichten des Dexter Modells ist nicht eindeutig. Toolbox beispielsweise erfüllt Aufgaben aller drei Schichten. Es ist aber möglich, die Kernfunktionalitäten der jeweiligen Schichten auf die einzelnen Komponenten aufzuteilen.

**Komponenteneinhaltsschicht:** Die Komponenten werden vollständig mit Toolbox beschrieben und von Toolbox verwaltet. Dies schließt ihre Erstellung, Speicherung und ihre Darstellung mit ein. Toolbox besitzt intern die Ankerschnittstelle, mit deren Hilfe Startanker zur Darstellung an die Speicherschicht übergeben werden können. Diese wird ausschließlich von Toolbox genutzt und wird nicht näher beschrieben.

**Speicherschicht:** Die Eingabedateien entsprechen der Datenbank der Speicherschicht. Mit Hilfe dieser Datenbank kann die Auflöserfunktion der Navigationssteuerung arbeiten. Die Zugriffsfunktion wird von Toolbox realisiert. Der zentrale Server und der User-Manager gehören gemäß dem AHAM ebenfalls zur Speicherschicht.

**Laufzeitschicht:** Die Rahmenanwendung verwaltet die Sitzungen und stellt die Instanzierungsfunktion. Die Darstellung der Komponenten im Arbeits- und Präsentationsbereich erfolgt durch Toolbox.

### 3.3.3 Nicht unterstützte Merkmale des Dexter Modell

Folgende Merkmale des Dexter Modells werden von dem System nicht unterstützt:

**Kompositum Komponenten:** Es existieren nur Seiten als Knoten des Hyperdokumentes. Seiten sind aber immer atomare Komponenten, Kompositum-Komponenten werden nicht genutzt.

**Verknüpfungen als Komponenten:** Verknüpfungen werden nicht explizit deklariert, sondern innerhalb der Ankerbeschreibungen in den Komponenten verwaltet.

**Zielanker:** Jeder Knoten der multimedialen Lehrgänge besitzt eine feste Größe. Ein aktuell dargestellter Knoten ist immer vollständig sichtbar. Deswegen genügt es für Verknüpfungen auf Komponenten zu verweisen, Zielanker werden nicht benötigt.

**Verknüpfungen:** Verknüpfungen verweisen immer nur auf atomare Komponenten, nicht auf Verknüpfungskomponenten oder Kompositum-Komponenten.

**Mehrwertige Verknüpfungen:** Eine Verknüpfung des hypermedialen Industriemeister 2000 Systems besteht immer aus genau einem Startanker und genau einer Endpunktbeschreibung mit der Richtung „Ziel“.

**Instanzen:** Es existiert zu jeder Zeit maximal eine Instanz genau einer Komponente. Die gleichzeitige Betrachtung mehrerer Instanzen wird nicht unterstützt.

**Laufzeitauflöserfunktion:** Es wird keine Laufzeitauflöserfunktion implementiert, weil nur statische Verweise genutzt werden.

Auf die Einhaltung der Invarianten I1 (Umkehrbarkeit der Zugriffsfunktion) und I2 (Bildbereich der Auflöserfunktion umfaßt alle Komponenten-IDs) wurde während der Erstellung der Datenbasis geachtet. Da nur atomare Komponenten existieren, ist die Invariante I3 (keine Komponente ist Teil von sich selbst) immer erfüllt. Verknüpfungen sind keine Komponenten, sie werden innerhalb der Komponentenschicht verwaltet. Somit kann nicht im Rahmen des Dexter Modells kontrolliert werden, ob baumelnde Verweise existieren und damit, ob I5 erfüllt ist. Durch das Fehlen von Zielankern ist I4 immer dann erfüllt, wenn I5 erfüllt ist, da die Startanker auch die Verknüpfungen umfassen.

### 3.3.4 Navigationsschritte

Navigation wird auf technischen Niveau betrachtet, weshalb nun einige Implementierungsdetails betrachtet werden.

In der Klasse *CNav* der Navigationssteuerung existieren mehrere Funktionen, die einer Navigationsschrittfunktion entsprechen. Zum Beispiel nutzt die FolgeVerknüpfung-Funktion des Systems eine Funktion aus der Navigationssteuerung. Nach der Aktivierung einer Ankerinstanz wird die Verknüpfung von Toolbook ermittelt und der Verweis wird einer Funktion der Rahmenanwendung übergeben, welche wiederum eine Funktion der Navigationssteuerung aufruft. Weitere Funktionen der Klasse *CNav* sind spezielle Funktionen für



Abbildung 3.6: Die Werkzeugleiste

Hilfsmittel. Alle diese Funktionen werden von der Rahmenanwendung aufgerufen. Zusammen ergibt sich nachfolgender Aufbau für jede dieser Funktionen:

Zuerst wird der Zustand des Geführte Tour-Hilfsmittel abgefragt, da hierdurch die Navigation beeinflusst wird (vgl. Abschnitt 3.4.2). Danach wird das Historie-Hilfsmittel (siehe Abschnitt 3.4.4) über den Navigationsschritt informiert. Anschließend aktualisiert die Rahmenanwendung die restlichen Hilfsmittel über den Navigationsschritt und ruft die Toolbook-Zugriffsfunktion auf und versetzt die Seite in den richtigen Lernschritt (siehe Abschnitt 3.3.1). Der Zugriff auf die Seite gegen Ende eines Navigationsschrittes ergibt sich aus der Zeitspanne, welche von der Zugriffsfunktion benötigt wird. Somit erhält der Benutzer ein schnelles Feedback, indem beispielsweise die Titelleiste bereits den neuen Titel anzeigt, während die Seite noch nicht angezeigt wird.

## 3.4 Hilfsmittel der Navigation

Die Betrachtung der Hilfsmittel ist der Schwerpunkt dieses Kapitels. Hilfsmittel sind Teil des Systems und sind entweder direkt sichtbar, oder durch die geöffnete *Werkzeugleiste* (siehe Abbildung 3.6) zu erreichen. Die Hilfsmittel der Werkzeugleiste können mit den zwei frei belegbaren Schaltflächen verbunden werden und sind dann ebenfalls direkt über die Funktionsleiste erreichbar.

Die Hilfsmittel sind im einzelnen: *Geführte Touren*, *Glossar*, *Historie*, *Inhaltsverzeichnis*, *Lesezeichen*, *Reihenfolge*, *Startauswahl* und *Titelzeile*. Die Funktionalität dieser Hilfsmittel ist auf mehrere Teile des hypermedialen Systems aufgeteilt. Die Dialoge wurden innerhalb der Rahmenanwendung erstellt, die Navigationsmöglichkeiten wurden in die Navigationssteuerung integriert. Die Funktionalität der Hilfsmittel findet sich in der Schnittstelle der Navigationssteuerung wieder. Im folgenden werden die Hilfsmittel als Ganzes betrachtet, was bedeutet, daß die gesamte Funktionalität eines Hilfsmittels zusammengefaßt wird. Zunächst wird eine Beschreibungsform für Hilfsmittel vorgestellt.

### 3.4.1 Die Beschreibungsform für Hilfsmittel

Für die Beschreibung der Hilfsmittel wird in diesem und im folgenden Kapitel folgende Form genutzt:

**Name** Den identifizierten Hilfsmitteln werden im folgenden Namen gegeben, die als Überschrift des jeweiligen Abschnittes genannt werden.

**Zweck** Unter diesem Abschnitt wird der Verwendungszweck des Hilfsmittels beschrieben. Wenn eine konkrete Anforderungsdefinition für ein Hilfsmittel existiert, so wird diese unter diesem Abschnitt notiert.

**Gründe für die Auswahl** In beiden Projekten wurde die Auswahl der Hilfsmittel nicht immer begründet. Teilweise entstanden sie aufgrund von vergleichbaren Hilfsmitteln in anderen hypermedialen Systemen, die allen Beteiligten bekannt waren und die als wichtig für die Navigation eingestuft wurden. Dokumentierte Gründe für eine Auswahl existieren nur für wenige Hilfsmittel. Soweit Gründe dokumentiert wurden, werden sie unter diesem optionalen Punkt erläutert.

**Beschreibung der Umsetzung** Unter diesem Punkt wird die Umsetzung des Hilfsmittels beschrieben. Zu der Umsetzung gehört die Interaktion des Benutzers mit dem Hilfsmittel und eine Abbildung eines eventuell existierenden Dialoges.

**Genutzte Informationen** Den Hilfsmitteln ist gemeinsam, daß sie Informationen nutzen, um sie zu verarbeiten und dem Benutzer zu präsentieren. Diese Informationen werden für jedes Hilfsmittel genauer beschrieben.

**Interaktion** Betrachtet man die Hilfsmittel und die einzelnen Schichten des Dexter Modells als Objekte zur Ausführungszeit, so können Kontroll- und Informationsflüsse, also die Interaktion zwischen diesen Objekten, untersucht werden. Eine Beschreibung der jeweiligen Interaktion mit den anderen Objekten wird in diesem Abschnitt dokumentiert.

**Besonderheiten** Sollten während der Analyse des Hilfsmittels Besonderheiten identifiziert werden, so werden sie unter diesem optionalen Punkt zusammengefaßt.

### 3.4.2 Geführte Touren

**Zweck** Eine Geführte Tour ist eine Zusammenstellung von Seiten, die in einer linearen Reihenfolge durchlaufen wird. Eine Geführte Tour kombiniert Seiten, so daß sie einen gemeinsamen Kontext besitzen. So kann eine Geführte Tour beispielsweise einen ersten Überblick über ein komplexes Thema zusammenstellen.

Die Anforderungsdefinition [ICD98, Seite 53ff] beschreibt das Geführte Tour-Hilfsmittel folgendermaßen: „Der Lernende kann vordefinierte und selbsterstellte geführte Touren bearbeiten und zur Navigation einsetzen. Geführte Touren können in Lesezeichen umgewandelt werden.“ Weiterhin ist in der Anforderungsdefinition zu lesen: „Eine geführte Tour kann jederzeit verlassen werden, bspw. durch Verfolgen eines Querverweises, durch Start einer anderen geführten Tour oder durch das Beenden des Lernprogramms.“



Abbildung 3.7: Der Geführte Touren Dialog

**Gründe für die Auswahl** Tergan [Tergan97] schreibt über den Sinn Geführter Touren: „Lineare Strukturen im Sinne sog. Unterweisungen (engl. guided tours) sind z. B. geeignet, Benutzer in neue Sachverhalte einzuführen oder ihnen vorab strukturierte Informationen zu vermitteln.“ Da es sich bei dem Industriemeister 2000 um ein Lernprogramm handelt, und der Lernende in neue Sachverhalte eingeführt wird, wurden Geführte Touren in das System integriert.

**Beschreibung der Umsetzung** Der Dialog der Geführten Touren (siehe Abbildung 3.7) kann immer über die Werkzeugleiste aktiviert werden. Als Symbol für Geführte Touren wird ein Wegweiser verwendet. Der Dialog besitzt folgende Elemente: eine Auswahlliste der verfügbaren Touren, ein Textfeld zur Anzeige der aktuellen Tour, eine Startschaltfläche, eine Schaltfläche zum Beenden einer Geführten Tour und eine Schaltfläche zum Schließen des Dialoges.

Die Liste der verfügbaren Touren befindet sich im oberen Bereich des Dialoges. Es werden immer nur die Geführten Touren des aktuellen Kurses angezeigt. Mit einem einfachen Mausklick kann eine Geführte Tour ausgewählt werden, die dann farblich unterlegt wird. Mit einem Doppelklick kann eine geführte Tour gestartet werden.

Unterhalb der Liste befindet sich ein Textfeld, welches den Titel der aktuell *laufenden* (= gestarteten und noch nicht beendeten) Geführten Tour anzeigt. Läuft keine Tour, so steht

in diesem Feld „(Keine)“. Der Startzustand des Hilfsmittels ist somit, daß keine Geführte Tour läuft.

Unterhalb des Textfeldes befinden sich die beiden Schaltflächen zum Starten und Beenden einer Geführten Tour. Mit der linken Schaltfläche kann eine Tour gestartet werden. Die Schaltfläche kann nur dann betätigt werden, wenn zuvor eine Geführte Tour aus der Liste ausgewählt wurde. Wurde eine Geführte Tour ausgewählt, so ist das Symbol der Schaltfläche blau, ansonsten grau. Wenn sie betätigt wird, so wird eine Geführte Tour gestartet, ihr Titel wird in das Textfeld oberhalb der Schaltfläche eingetragen und es wird die erste Seite der Geführten Tour dargestellt. Eine laufende Tour kann mit der rechten Schaltfläche beendet werden. Die Schaltfläche kann nur betätigt werden, wenn eine Tour läuft und besitzt dann ein blaues Symbol, ansonsten ein graues. Mit der Schaltfläche in der rechten unteren Ecke des Dialoges kann man den Dialog schließen. Die aktuell dargestellte Seite ändert sich nicht durch das Beenden.

Während eine Geführte Tour läuft, kann der Benutzer mit dem Reihenfolge-Hilfsmittel (siehe Abschnitt 3.4.7) die Seiten der Geführten Tour besuchen. Wenn mit einem Hilfsmittel oder durch eine aktivierte Verknüpfung ein Navigationsschritt zu einer Seite außerhalb der Geführten Tour initiiert wird, so wird der Benutzer gefragt, ob er die Geführte Tour beenden und die Seite besuchen möchte, oder den Navigationsschritt abbrechen möchte.

Intern werden Geführte Touren auf Kapitel abgebildet. Eine Geführte Tour wird in der Navigationssteuerung als ein Kapitel behandelt, welches nicht im Inhaltsverzeichnis steht. Da ein Kapitel eine Reihenfolge besitzt, kann eine Geführte Tour mit dem Reihenfolge-Hilfsmittel (siehe Abschnitt 3.4.7) durchlaufen werden. Die Navigationssteuerung wird über den Beginn und das Ende einer neuen Tour durch dafür vorgesehene Funktionen benachrichtigt. Das Hilfsmittel wird somit also in der Rahmenanwendung und in der Navigationssteuerung realisiert.

**Genutzte Informationen** Geführte Touren nutzen die in den Eingabedateien vom Autor der Datenbasis erstellten Daten. Die notwendigen Informationen erhält das Hilfsmittel aus der Speicherschicht. Der Benutzer des Systems startet und beendet Geführte Touren.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Die Laufzeitschicht wird über den Start und das Ende einer Geführten Tour informiert. Diese Information wird dann von dem Reihenfolge-Hilfsmittel verarbeitet. Wenn der Benutzer einen Navigationsschritt während einer laufenden Tour mit einem anderen Hilfsmittel als der Reihenfolge anstößt, wird eine Anfrage an das Geführte Tour-Hilfsmittel gestellt, ob der Navigationsschritt „erlaubt“ ist. Das Hilfsmittel erlaubt den Schritt, wenn er innerhalb der Tour bleibt, oder interagiert mit dem Benutzer. Daraufhin wird der Navigationsschritt abgebrochen oder die Geführte Tour beendet. Der Start einer Geführten Tour löst einen Navigationsschritt aus.

**SPEICHERSCHICHT:** Aus der Speicherschicht werden die vorhandenen Geführten Touren und die Seiten der Touren ausgelesen.

**REIHENFOLGE:** Das Reihenfolge-Hilfsmittel nutzt den Namen einer eventuell laufenden Tour.

**TITELLEISTE:** Das Titelleiste-Hilfsmittel nutzt den Namen einer eventuell laufenden Tour.

**Besonderheiten** Das Hilfsmittel nutzt die strukturellen Besonderheiten der Datenbasis aus. Geführte Touren werden in den Eingabedateien definiert und sind somit Teil der Datenbasis.

#### 3.4.3 Glossar

**Zweck** Ein Glossar kann immer dann nützlich werden, wenn viele (Fach-)Begriffe in den Inhalten einer umfangreichen Datenbasis genutzt werden. Das hypermediale System des Industriemeisters 2000 besitzt als Datenbasis den Lehrgang Physik. Zum Nachschlagen wichtiger Fachbegriffe und Hintergrundinformationen dient das Glossar.

Die Anforderungsdefinition des Projektes [ICD98, Seite 66] beschreibt das Glossar folgendermaßen: „Das Glossar erscheint auf dem Bildschirm und listet dem Lernenden alle erklärten Begriffe alphabetisch sortiert auf. Er kann den gesuchten Begriff direkt eingeben oder die Liste nach ihm durchsuchen. Das Glossar liefert daraufhin eine Erklärung dieses Begriffes. Diese Erklärung ist in der Regel rein textuell, kann aber auch multimedial sein.“

**Gründe für die Auswahl** Das Glossar-Hilfsmittel dient dem Zugriff auf die *Glossarseiten*. Erklärungen nicht verstandener Fachbegriffe können mit diesem Hilfsmittel gefunden werden [ICD98]. Das Glossar-Hilfsmittel wurde somit wegen seiner inhaltlichen Funktion und den damit entstandenen zusätzlichen Navigationsmöglichkeiten ausgewählt.

**Beschreibung der Umsetzung** Das Glossar-Hilfsmittel besteht aus einem Dialog für die Funktionalität sowie aus Glossarseiten, welche die Einträge des Glossars erklären. Für jeden Begriff wurde eine eigene Seite erstellt, welche die wichtigsten Information zu dem Begriff darstellt. Handelt es sich etwa um den Namen eines Erfinders, so werden auf der Glossarseite Eckdaten seines Lebens vermittelt. Glossarseiten unterscheiden sich nur inhaltlich von den restlichen Seiten des Hypermediadokumentes. Sie können multimediale Inhalte besitzen.

Die Glossarseiten wurden als Kurs erstellt. Damit sind die Seiten des Glossars auch über das Inhaltsverzeichnis (siehe Abschnitt 3.4.5) zugänglich, wo sie nach Anfangsbuchstaben in Kapiteln gegliedert sind.

Die Glossarseiten können durch Verknüpfungen erreicht werden. Es existiert zusätzlich ein spezieller Dialog für das Glossar (siehe Abbildung 3.8). Als Symbol für das Glossar verwendet das System eine „A-Z“ Zeichenfolge. Der Dialog besitzt drei Bedienungselemente: ein Textfeld, eine Auswahlliste und eine Schaltfläche.

Die Auswahlliste befindet sich in der Mitte des Dialoges. Sie zeigt alle Glossareinträge in alphabetischer Reihenfolge an. Die Liste besitzt eine Bildlaufleiste zum Scrollen (verschieben) des Inhalts. Der Benutzer kann ein Element der Liste auswählen und die Glossarseite,



Abbildung 3.8: Der Dialog des Glossars

die zu dem Begriff gehört, durch einen Doppelklick darstellen lassen. Der Dialog schließt dadurch nicht.

Das Textfeld oberhalb der Auswahlliste dient der Eingabe eines Suchbegriffs. Sobald ein Zeichen in das Textfeld eingegeben oder gelöscht wird, wird in der Auswahlliste das erste Element angezeigt und farbig hinterlegt, welches mit derselben Buchstabenkombination beginnt wie die vom Benutzer eingegebene. Hierbei wird Groß- und Kleinschreibung vernachlässigt. Existiert kein Eintrag mit derselben Buchstabenkombination zu Beginn des Wortes, so wird ein Wort mit einer möglichst langen Übereinstimmung ausgewählt.

Mit der Schaltfläche unten rechts im Dialog wird dieser geschlossen. Die dargestellte Seite wird dadurch nicht verändert.

**Genutzte Informationen** Die Einträge der Liste werden aus der Eingabedatei des Kurses „Glossar“ entnommen. Die Einträge entsprechen den Seitentitel, während die Kapitelhierarchie ignoriert wird. Zu dem Titel werden Komponentenverweise gespeichert.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Das Glossar löst einen Navigationsschritt aus, wenn eine Glossarseite ausgewählt wurde.

**SPEICHERSCHICHT:** Die Titel und Komponentenverweise der Seiten des Glossar werden der Speicherschicht entnommen.

**Besonderheiten** Die Entscheidung, das Glossar als einen Kurs zu realisieren, für den ein spezielles Hilfsmittel zur Informationsfindung existiert, erlaubt eine einfache Integration der Glossarseiten in die Hilfsmittel Inhaltsverzeichnis und Historie.

#### 3.4.4 Historie

**Zweck** Mit der Historie können die einzelnen Navigationsschritte rückgängig gemacht werden. Die Anforderungsdefinition [ICD98, Seite 58] beschreibt die Historie folgendermaßen: „Die Historie enthält die vom Lernenden bereits besuchten Seiten inklusive der mit den Besuchen verbundenen Reihenfolge. Der Lernende kann damit auf bereits besuchte Seiten zugreifen.“

**Gründe für die Auswahl** Eine Historie-Funktion wird von fast allen modernen hypermedialen Systemen angeboten. Insofern kann ein solches Hilfsmittel als Standard angesehen werden.

**Beschreibung der Umsetzung** Die Historie ermöglicht die Zurückverfolgung der einzelnen Navigationsschritte bis zu dem Beginn der Sitzung. Während der Benutzer mit dem System arbeitet, werden bei jedem Navigationsschritt Komponentenverweise auf einem Kellerspeicher (Stack) abgelegt. Es wird nicht unterschieden, wie der Benutzer zu dieser Seite navigierte, d. h. ob durch Hilfsmittel oder durch eine Verknüpfung. Eine Zurückverfolgung bedeutet, daß der oberste Komponentenverweis auf dem Kellerspeicher aufgelöst und die dazu gehörende Seite angezeigt wird. Anschließend wird der oberste Komponentenverweis vom Kellerspeicher gelöscht. Ein Verweis auf die Seite, die dargestellt wurde, als der Benutzer eine Zurückverfolgung auslöste, wird nicht gespeichert. Jede weitere Zurückverfolgung führt dazu, daß der Benutzer die jeweils zuvor dargestellte Seite dargestellt erhält, bis zu der Seite, mit der die Sitzung gestartet wurde.

Die Historie besitzt keinen eigenen Dialog. Sie kann über die obere der beiden mittleren Schaltflächen der Mehrfachschaltfläche (siehe Abbildung 3.11) am unteren Rand des Lernsystems erreicht werden. Der Benutzer kann durch das Drücken der Schaltfläche auf die zuvor besuchte Seite gelangen. Die Schaltfläche trägt einen nach oben weisenden Pfeil als Symbol. Befindet sich ein Komponentenverweis auf dem Kellerspeicher, so ist das Symbol der Schaltfläche blau. Ohne Verweis auf dem Kellerspeicher ist das Symbol grau und die Schaltfläche kann nicht betätigt werden.

Der Startzustand des Historie-Hilfsmittel ist der leere Kellerspeicher. Dieser Startzustand wird zum Beginn einer Sitzung nach der Startseitenauswahl mit dem Startauswahl-Hilfsmittel eingenommen.

**Genutzte Informationen** Die Informationen zur Bereitstellung der Funktionalität dieses Hilfsmittels werden automatisch während der Navigation gesammelt.

**Interaktion** Das Hilfsmittel interagiert mit der

LAUFZEITSCHICHT: Die Laufzeitschicht informiert das Historie-Hilfsmittel über einen Navigationsschritt. Das Hilfsmittel kann einen Navigationsschritt auslösen.

### 3.4.5 Inhaltsverzeichnis

**Zweck** Das Inhaltsverzeichnis ist ein Hilfsmittel, das eine Übersicht über die Datenbasis des Industriemeisters 2000 bietet. Alle in Kapiteln gegliederten Seiten werden dem Benutzer zur Auswahl gestellt. Seiten, die nur in Geführten Touren stehen, werden nicht aufgezählt. Die Anforderungsdefinition beschreibt das Inhaltsverzeichnis als ein hierarchisch gegliedertes Hilfsmittel, mit dem die Seiten der Datenbasis per Mausklick angesteuert werden können. Ferner zeigt es an, welche Seiten bereits von einem Benutzer besucht, bzw. bearbeitet wurden [ICD98, Seite 132].

**Beschreibung der Umsetzung** Der Inhaltsverzeichnisdialog (siehe Abbildung 3.9) ist immer über die Werkzeugleiste erreichbar. Das Inhaltsverzeichnis ist standardmäßig die Belegung der linken frei belegbaren Schaltfläche. Als Symbol für das Inhaltsverzeichnis wird eine Kompaßrose verwendet.



Abbildung 3.9: Der Dialog des Inhaltsverzeichnisses

Der Inhaltsverzeichnisdialog besitzt folgende Elemente: Ein Kombinationsfeld zur Auswahl eines Kurses, eine Baumansicht der verfügbaren Seiten und eine Schaltfläche zum Verlassen des Dialoges.

Das Kombinationsfeld zur Auswahl eines Kurses befindet sich über der Baumansicht im oberen Bereich des Dialoges und ermöglicht die Auswahl eines Kurses, dessen Inhalt in der Baumansicht angezeigt werden soll. Der Benutzer des Systems kann entweder den Namen mit der Tastatur eingeben oder den Kurs aus einer Liste auswählen, die bei einem Mausklick auf die Schaltfläche am rechten Rand des Kombinationsfeldes erscheint. Die Liste enthält die Titel aller auswählbarer Kurse. Wenn der Dialog geöffnet wird, ist der aktuelle Kurs ausgewählt.

Die Baumansicht zeigt die Kapitel und Seiten des ausgewählten Kurses. Sie ermöglicht eine strukturierte und übersichtliche Darstellung von Hierarchien. In diesem Fall wird die Baumansicht genutzt, um die Hierarchie eines Kurses anzuzeigen. Die oberste dargestellte Hierarchiestufe ist die Seiten- und Kapitelaufzählung des Kurses. Kapitel können in der Baumansicht detaillierter betrachtet werden (der Kapiteleintrag wird „expandiert“) oder detaillierte Betrachtungen geschlossen werden (der Kapiteleintrag „kollabiert“). Eine detailliertere Betrachtung eines Kapitels K erfolgt durch die Darstellung der Seiten und der Unterkapitel, eingerückt unterhalb der Darstellung von K. Seiten und Kapitel werden in der Baumansicht durch ein links vom Kapitel- oder Seitennamen stehendes Symbol unterscheidbar gemacht. Wird eine Seite durch einen Doppelklick ausgewählt, so wird sie im Arbeits- und Präsentationsbereich dargestellt. Wird ein Kapitel durch einen Doppelklick ausgewählt, so wechselt die Darstellung zwischen detaillierter und nicht detaillierter Darstellungsweise. Die Baumansicht erlaubt nicht die Veränderung der Namen von Kapitel und Seiten.

In der Baumansicht werden verschieden farbige Symbole für Seiten die noch nicht, noch nicht vollständig und vollständig besucht wurden (vgl. Abschnitt 3.3), verwendet. Diese Informationen werden dem Benutzermodell entnommen. Seiten, die noch nie vom Benutzer besucht worden sind, besitzen einen roten Punkt im Symbol, die besuchten, aber noch nicht vollständig bearbeiteten einen gelben Punkt und die vollständig bearbeiteten einen grünen Punkt im Symbol. Das Inhaltsverzeichnis greift direkt auf das Benutzermodell zu und nutzt einen eigenen Adaptionsgestalter für die Adaption.

Wenn der Dialog geöffnet wird, so ist die aktuell dargestellte Seite farbig hinterlegt. Weiterhin sind alle Kapitel, zu denen die Seite transitiv gehört, detailliert dargestellt, alle anderen Kapitel nicht. Gehört die aktuelle Seite nur zu einer Geführten Tour und nicht zu einem Kapitel, so wird das Inhaltsverzeichnis auf der obersten Ebene angezeigt und keine Seite ausgewählt.

Die Schaltfläche rechts unten im Dialog schließt den Dialog. Die dargestellte Seite wird dadurch nicht verändert.

**Genutzte Informationen** Das Inhaltsverzeichnis nutzt die Hierarchien, die in den Eingabedateien von den Autoren der Datenbasis aufgestellt wurden. Ferner nutzt das Inhaltsverzeichnis die Titel der Kapitel und Seiten sowie Komponentenverweise auf die Seiten.

Aus dem Benutzermodell werden die Informationen über den Besuch-Status der Seiten ausgelesen.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Das Inhaltsverzeichnis kann einen Navigationsschritt auslösen.

**SPEICHERSCHICHT:** Die Hierarchien, verfügbaren Kurse und der Besuch-Status werden aus der Speicherschicht ausgelesen.

**Besonderheiten** Das Inhaltsverzeichnis bietet Informationen über den Fortschritt des Benutzers innerhalb des Hyperdokumentes und ermöglicht die direkte Navigation zu den meisten Seiten innerhalb desselben. Der Benutzer erhält Informationen darüber, welche Seiten er noch nicht oder nicht vollständig bearbeitet hat, d. h. wo noch Lücken in der Bearbeitung liegen. Er kann somit gezielt weiterarbeiten. Die Informationen hierzu werden aus dem Benutzermodell ausgelesen. Damit wird eine weitere Möglichkeit adaptiver hypermedialer Systeme aufgezeigt, die noch nicht in der Arbeit von [DeBra99] beachtet wurde: Neben Inhalts- und Verknüpfungsanpassungen können auch Hilfsmittel angepaßt werden.

Die detaillierte Betrachtung der näheren Umgebung der aktuellen Seite beim Aufruf des Hilfsmittels wird häufig auch als *Fischaugensicht* bezeichnet. Die nähere Umgebung wird detaillierter, die entferntere Umgebung nur am Rande gezeigt.

### 3.4.6 Lesezeichen

**Zweck** Lesezeichen sind leicht zu erreichende Verweise auf zuvor besuchte Seiten, die der Benutzer anlegt. Mit ihnen kann der Benutzer direkt zu wichtigen Seiten navigieren.

**Gründe für die Auswahl** In der Anforderungsdefinition [ICD98, Seite 116] wird über den Zweck von Lesezeichen gesagt: „Besonders umfangreiche Lernprogramme sollten die Möglichkeit der Zwischenspeicherung (elektronisches Lesezeichen) von Lernergebnissen und bereits bearbeiteten Kapiteln / Modulen bieten. So kann ein unterbrochener Lernprozeß zu einem späteren Zeitpunkt wieder an gleicher Stelle fortgesetzt werden.“ Lesezeichen werden von nahezu jedem modernen hypermedialen System unterstützt. Insofern kann ein solches Hilfsmittel als Standard angesehen werden.

**Beschreibung der Umsetzung** Der Lesezeichendialog (siehe Abbildung 3.10) kann durch die Werkzeugleiste ausgewählt werden. Als Symbol für das Lesezeichen-Hilfsmittel wird ein Pinnwandstecker verwendet. Der Lesezeichendialog besitzt folgende Elemente: eine Auswahlliste mit verfügbaren Lesezeichen, jeweils eine Schaltfläche zum Erstellen eines neuen Lesezeichens, zum Löschen eines Lesezeichens und zum Schließen des Dialoges.

Die Auswahlliste mit den verfügbaren Lesezeichen bietet dem Benutzer eine Liste bereits erstellter Lesezeichen. Sie füllt fast den gesamten Dialog aus. Wenn mehr Lesezeichen

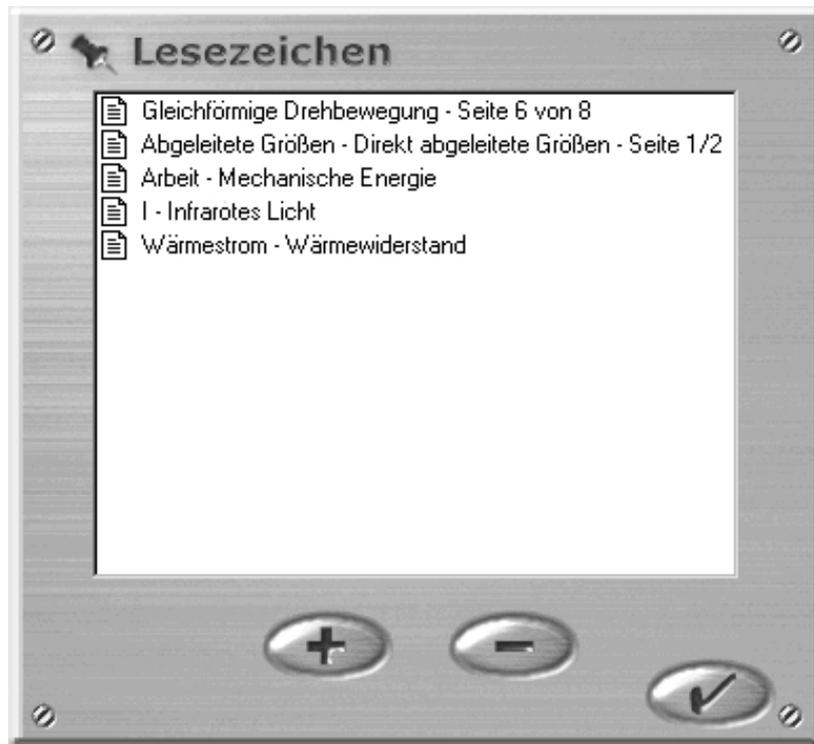


Abbildung 3.10: Der Lesezeichendialog

erstellt wurden als die Liste gleichzeitig anzeigen kann, wird am rechten Rand eine Bildlaufleiste eingeblendet, mit welcher der Inhalt der Liste gescrollt werden kann. Die Liste beinhaltet eine textuelle Beschreibung für jedes Lesezeichen in derselben Reihenfolge, in der sie angelegt wurden. Der Benutzer kann den Titel eines Lesezeichens, also den Eintrag in der Liste, ähnlich einem Windows-Textfeld, bearbeiten. Durch einen Doppelklick kann ein Lesezeichen aufgerufen werden. In diesem Fall wird der Knoten dargestellt, mit dem das Lesezeichen verbunden ist. Der Dialog schließt nicht, wenn ein Lesezeichen ausgewählt wurde.

Die Schaltfläche zum Erstellen eines neuen Lesezeichens befindet sich im linken unteren Bereich des Dialoges. Sie besitzt als Symbol ein Pluszeichen. Wenn sie schaltet, so wird ein Lesezeichen für die aktuell dargestellte Seite in die Liste der verfügbaren Lesezeichen eingetragen. Als Namen für das Lesezeichen wird standardmäßig der Text der Titelleiste (siehe Abschnitt 3.4.9) verwendet. Er besteht aus dem Namen des Kapitels, zu dem die Seite gehört, einem Verbindungsstrich und dem Titel der Seite. Im Falle einer laufenden Geführten Tour wird nicht der Name der Geführten Tour angezeigt, da durch ein Lesezeichen auch keine Geführte Tour gestartet werden kann. Da Geführte Touren auch auf Kapitel abgebildet werden, genügt es immer den Namen des Kapitels auszulesen. Es können mehrere Lesezeichen für eine Seite angelegt werden.

Die Schaltfläche zum Löschen eines Lesezeichens befindet sich im rechten unteren Bereich des Dialoges und besitzt ein Minuszeichen als Symbol. Wenn diese Schaltfläche betätigt

wird, so wird das aktuell ausgewählte Lesezeichen gelöscht. Wurde zuvor kein Lesezeichen ausgewählt, so ist diese Schaltfläche deaktiviert.

Jeder Benutzer besitzt seine eigenen Lesezeichen. Sie werden für jeden Benutzer einzeln verwaltet. Somit kann jeder Lehrgangsteilnehmer seine Lesezeichen selber erstellen, umbenennen und löschen. Die Reihenfolge, in der die Lesezeichen erstellt wurden und damit angezeigt werden, wird auch über mehrere Sitzungen hinweg erhalten.

Die Lesezeichen werden intern als Komponentenverweis im Benutzermodell gespeichert. Zwar sind die Lesezeichen keine adaptiven Daten (vgl. Abschnitt 2.1.4), aber auf diese Weise wird eine konsistente Datenhaltung ermöglicht, da nur ein Datensatz (das Benutzermodell) zu speichern ist.

Der Komponentenverweis erhält als weiteres Attribut einen Namen, der in der Liste angezeigt wird. Der Startzustand des Lesezeichen-Hilfsmittel ist die Liste der Lesezeichen, die dem letzten Zustand der letzten Sitzung entsprechen, oder die leere Liste, wenn der Benutzer seine erste Sitzung mit der Anwendung startet.

**Genutzte Informationen** Das Lesezeichen nutzt Informationen, die der Benutzer durch die Auswahl der Lesezeichen zur Verfügung stellt. Das Lesezeichen verwaltet Komponentenverweise auf bereits dargestellte Seiten und liest den Titel des Kapitels.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Ein Lesezeichen kann einen Navigationsschritt auslösen. Das Lesezeichen wird über die aktuell besuchte Seite informiert.

**SPEICHERSCHICHT:** Der Titel eines Lesezeichens, sowie der Titel des Kapitels wird aus der Speicherschicht ausgelesen.

**Besonderheiten** Obwohl das Hilfsmittel als ein Standardhilfsmittel angesehen werden kann, besitzt dieses Hilfsmittel nicht die sonst „übliche“ Einteilung in Lesezeichengruppen.

### 3.4.7 Reihenfolge

**Zweck** Wie im Abschnitt 3.2 beschrieben, besteht eine Ordnungsrelation (Vorgänger, Nachfolger) innerhalb eines Kurses und innerhalb einer Geführten Tour. Das nun beschriebene Hilfsmittel nutzt diese Relation.

Dazu steht in der Anforderungsdefinition des Projektes [ICD98, Seite 53]: „Der Lernende kann mit Hilfe der Knöpfe vorwärts und rückwärts den Inhalt eines Themas systematisch durchsehen. Dabei werden alle Knoten entsprechend ihrer Anordnung in der Hierarchie besucht und präsentiert.“



Abbildung 3.11: Das Reihenfolge-Hilfsmittel kann mit Hilfe der linken und der rechten Schaltfläche der Mehrfachschaltfläche erreicht werden.

**Gründe für die Auswahl** Das Reihenfolge-Hilfsmittel dient laut der Anforderungsdefinition der systematischen Bearbeitung der Datenbasis [ICD98]. Die Möglichkeit einer systematischen Bearbeitung ergab sich durch die prinzipiell linear aufgebauten Inhalte, die aus der multimedialen Aufbereitung von Lehrbüchern entstanden sind. Gleichzeitig dient das Reihenfolge-Hilfsmittel zur Navigation innerhalb einer Geführten Tour (siehe Abschnitt 3.4.2). Das Reihenfolge-Hilfsmittel baut, ebenso wie eine Geführte Tour, auf einer linearen Reihenfolge auf.

**Beschreibung der Umsetzung** Der Benutzer besitzt die Möglichkeit, mit Hilfe der Funktionsleiste zwischen Seiten zu navigieren. Die Mehrfachschaltfläche besitzt zwei Schaltflächen, die einen Pfeil nach links bzw. nach rechts als Symbol tragen (siehe Abbildung 3.11). Mit diesen Schaltflächen kann er zu der vorherigen bzw. nächsten Seite, durch das Betätigen der jeweiligen Schaltfläche, navigieren. So kann der Benutzer innerhalb eines Kurses alle Seiten des Kurses besuchen. Befindet sich der Lernende auf der ersten Seite eines Kurses, so besitzt die linke Schaltfläche ein Symbol in der Farbe Grau, ansonsten ist das Symbol mit der Farbe Blau gefüllt. Die rechte Schaltfläche erscheint mit einem grauen Symbol, wenn der Benutzer die letzte Seite des Kurses dargestellt bekommt. Schaltflächen mit grauen Symbolen können nicht schalten.

**Genutzte Informationen** Das Reihenfolge-Hilfsmittel nutzt Informationen über die Reihenfolge von Seiten innerhalb von Kapiteln oder Geführten Touren. Diese Informationen werden vom Autor in den Eingabedateien festgelegt. Weiterhin werden zur Laufzeit Informationen über eine eventuell laufende Geführte Tour und über den aktuell dargestellten Knoten verarbeitet.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Das Hilfsmittel wird über jeden Navigationsschritt in Kenntnis gesetzt.

**SPEICHERSCHICHT:** Das Reihenfolge-Hilfsmittel erhält die Reihenfolge der einzelnen Seiten aus der Speicherschicht.

**GEFÜHRTE TOUR:** Von dem Geführten Tour-Hilfsmittel erhält dies Reihenfolge-Hilfsmittel den Kontext der aktuellen Seite. Da eine Seite zu mehreren Geführten Touren und zu einem Kapitel gehören kann, hängt die Vorgänger- und die Nachfolgerseite von der eventuell laufenden Geführten Tour ab.

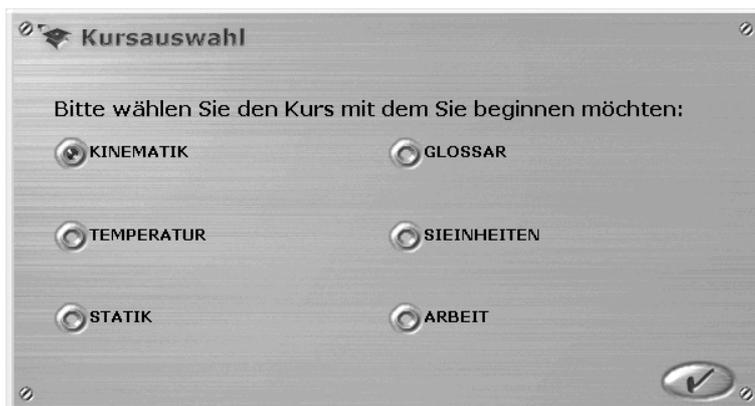


Abbildung 3.12: Startauswahl

**Besonderheiten** Daß eine Reihenfolge innerhalb der Seiten der Industriemeister 2000 Anwendung sinnvoll ist, entstammt aus der Aufarbeitung der Inhalte aus sequentiell aufgebauten Lehrbüchern.

### 3.4.8 Startauswahl

**Zweck** Dieses Hilfsmittel soll am Beginn einer neuen Sitzung dem Benutzer einen sinnvollen Einstiegspunkt zur Verfügung stellen. Die Anforderungsdefinition hatte dieses Hilfsmittel noch nicht vorgesehen. Es wurde realisiert, da keine herausragende Startseite in der Datenbasis existiert und der Benutzer eine Sitzung nicht in einem willkürlich ausgewählten Kurs starten sollte.

**Gründe für die Auswahl** Die Startauswahl wurde in das System integriert, da die Navigation in dem Hyperdokument eine Startseite benötigt, von wo aus der Benutzer weiter navigieren kann. Die Startauswahl setzt das System in einen Startzustand.

**Beschreibung der Umsetzung** Benutzt ein Teilnehmer zum ersten mal das System, so wird er beim Start einer Sitzung gefragt, mit welchem Kurs er beginnen möchte. Dazu wird ihm mit dem Startseitendialog (siehe Abbildung 3.12) eine Auswahl der ihm zur Verfügung stehenden Kurse geboten. Er kann sich einen Kurs auswählen und beginnt dann mit der ersten Seite des Kurses. Bei jeder nächsten Sitzung wird der Benutzer gefragt, ob er mit der zuletzt besuchten Seite der letzten Sitzung weiterarbeiten möchte. Die Information, auf welcher Seite sich der Benutzer befand, wird im Benutzermodell gespeichert. Möchte er nicht zu der letzten Seite navigieren, so wird ihm wieder der Startseitendialog angezeigt.

**Genutzte Informationen** Die Startseite verarbeitet eine eventuell vorhandene letzte Seite aus dem Benutzermodell und die dem Benutzer verfügbaren Kurse.



Abbildung 3.13: Die Titelleiste

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Es wird ein Navigationsschritt zu Beginn jeder Sitzung ausgelöst.

**SPEICHERSCHICHT:** Das Hilfsmittel liest aus der Speicherschicht die zuletzt besuchte Seite, sowie die dem Benutzer verfügbaren Kurse aus.

**Besonderheiten** Dieses Hilfsmittel ermöglicht dem Teilnehmer, an der Stelle weiter zu arbeiten, an der er zuletzt gearbeitet hatte, oder an einem herausragenden Punkt (*landmark*) innerhalb der Datenbasis mit der Navigation zu beginnen.

### 3.4.9 Titelleiste

**Zweck** Die Titelleiste dient der Orientierung innerhalb der Datenbasis. In der Anforderungsdefinition [ICD98, Seite 108] steht zu lesen: „Die Titelleiste befindet sich am oberen Bildschirmrand. Es sind dort der aktuelle Themenbereich, die aktuelle Kapitelüberschrift und der aktuelle Seitentitel ersichtlich.“

**Gründe für die Auswahl** Die Titelleiste stellt zu jeder Zeit Informationen über den Kurs und das Kapitel der aktuell dargestellten Seite dar. Somit dient dieses Hilfsmittel, das Problem der Desorientierung (siehe Abschnitt 2.4.2) zu vermeiden.

**Beschreibung der Umsetzung** Die Titelleiste besitzt zwei Bestandteile: eine Anzeige für den Kurs sowie eine textuelle Anzeige des Kapitels und des Seitennamens. Der Kurs wird in der linken Hälfte angezeigt. Hierfür existiert für jeden Kurs eine Bilddatei, welche in einem farbigen Schriftzug den Kursnamen enthält. Die unterschiedlichen Farben für verschiedene Kurse sollen eine schnelle Orientierung ermöglichen.

Im rechten Bereich wird ein Text in die Leiste eingeblendet, der den Titel des Kapitels angibt, in dem sich die aktuelle Seite befindet, gefolgt von einem Verbindungsstrich und dem Titel der Seite. Es handelt sich also um den gleichen Text wie er standardmäßig auch als Titel für Lesezeichen (siehe Abschnitt 3.4.6) vergeben wird. Er besteht aus dem Titel des aktuellen Kapitels bzw. der aktuellen Geführten Tour, einem Querstrich und dem Titel der Seite, wie in Abbildung 3.13 ersichtlich.

**Genutzte Informationen** Die Informationen werden vom Autor bereits für das Inhaltsverzeichnis zur Verfügung gestellt. Die Titelleiste kann diese Informationen wiederverwenden.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Die Titelleiste wird über jeden Navigationsschritt informiert.

**SPEICHERSCHICHT:** Die Titel der Seite, des Kapitels und des Kurses werden anhand des Komponentenverweises aus der Speicherschicht ausgelesen.

**GEFÜHRTE TOUR:** Der Titel einer Geführten Tour wird ausgelesen, um ihn eventuell anstatt des Kapiteltitels anzuzeigen.

**Besonderheiten** Die Titelleiste dient ausschließlich der Orientierung innerhalb der Datenbasis. Sie ist nicht Teil der Darstellung eines Knotens, weil die Informationen, die sie darstellt, nicht in der Komponente liegen, sondern in der Speicherschicht. Somit ist die Titelleiste kein Teil der Komponente.

---

## Kapitel 4

# Die hypermediale Altenberger Dom Anwendung

Dieses Kapitel umfaßt eine Beschreibung der hypermedialen Anwendung des Altenberger Doms. Die Darstellung des Projektes in Abschnitt 4.1 enthält einen Überblick über die Anwendung und dient als Kontext für die weitere Darstellung. Die Analyse der Datenbasis im Abschnitt 4.2 umfaßt eine Beschreibung ihrer Struktur und einen Vergleich mit dem Dexter Referenz Modell. Das hypermediale System wird im Abschnitt 4.3 abstrakt anhand einer virtuellen Maschine und dem Dexter Modell beschrieben. Die im Abschnitt 4.4 beschriebenen Hilfsmittel bauen auf diesen Beschreibungen auf, da sie Strukturen der Datenbasis nutzen und mit dem hypermedialen System zusammen arbeiten.

Die Ergebnisse dieses Kapitels werden in Kapitel 5 mit den Ergebnissen des Kapitels 3 verglichen. Damit ein solcher Vergleich möglich ist, werden Referenzmodelle genutzt.

### 4.1 Projektbeschreibung

In diesem Kapitel wird das Altenberger Dom Projekt beschrieben. Teilziel des Projektes ist die Erstellung einer hypermedialen Anwendung, auf die hier näher eingegangen werden wird. Weitere Angaben zu dem besagtem Projekt dienen nur der besseren Einordnung der Anwendung und der Begriffsbildung.

Das Projekt „Multimediatechnik in der Baugeschichte – Veranschaulichung eines gotischen Architektursystems am Beispiel des Altenberger Domes“, kurz Altenberger Dom, ist eine Kooperation der beiden Lehrstühle für Baugeschichte (Prof. Dr. Nußbaum, Fachbereich Bauwesen) und für Softwaretechnologie (Prof. Dr. Doberkat, Fachbereich Informatik) an der Universität Dortmund. Es begann am 1. September 1996, ein Endtermin wurde noch nicht gesetzt. Es wurde zwei Jahre lang durch den Universitätsbund Multimedia NRW gefördert.

Bislang sind weder die Datenbasis noch die Rahmenanwendung vollständig erstellt worden. Weiterhin existiert momentan noch keine Dokumentation des Projektes, welche die genauen Anforderungen oder Entwurfsentscheidungen dokumentiert.

### 4.1.1 Zielsetzung

Für das Projekt existieren drei verschiedene Zielsetzungen. Die Kooperation besitzt laut Projektantrag [AD97] folgende Zielsetzung: „Ziel ist die Erstellung einer interaktiv benutzbaren und mehrsprachigen CD-ROM, durch die wesentliche Inhalte der baugeschichtlichen Lehre unter Ausnutzung multimedialer Techniken anschaulich vermittelt werden. Dieses Produkt soll regelmäßig im Lehrbetrieb des Lehrstuhles für Baugeschichte eingesetzt und auch den bau- und kunsthistorischen Instituten und Lehrstühlen anderer Hochschulen zum Einsatz angeboten werden.“ Weiterhin möchte der Lehrstuhl für Baugeschichte [AD97] ein „exemplarisches Einüben der neuen multimedialen Techniken zur Darstellung kunsthistorischer Inhalte“ erreichen. Den Lehrstuhl für Softwaretechnologie interessiert weiterhin „die Möglichkeit der Interaktion mit geisteswissenschaftlich orientierten Nutzern, einem Anwenderkreis, der durch den Einsatz multimedialer Techniken als Benutzergruppe mit neuen, spezifischen Anforderungen gewonnen werden kann“ [Alfert98].

Die Erstellung einer interaktiv benutzbaren und mehrsprachigen CD-ROM umfaßt die Erstellung eines hypermedialen Systems (vgl. Abschnitt 4.3) und die Erstellung der Inhalte, d. h. der Datenbasis. Die Erstellung der Datenbasis ist Aufgabe des Lehrstuhls für Baugeschichte, die Entwicklung des hypermedialen Systems ist Aufgabe des Lehrstuhls für Softwaretechnologie.

### 4.1.2 Derzeitiger Entwicklungsstand

Die Erstellung der Hypermediaanwendung Altenberger Dom ist noch nicht abgeschlossen. Dennoch kann die Anwendung beschrieben werden, da sich die Mitarbeiter der beiden Lehrstühle bereits in den wichtigsten Punkten geeinigt haben und konkrete Umsetzungspläne existieren. Dies wird nun erläutert. Im Abschnitt 4.1.2.1 wird dann die Oberfläche und in Abschnitt 4.1.2.2 die Architektur der Umsetzung beschrieben.

Die Mitarbeiter des Lehrstuhles für Baugeschichte, also die Autoren der Datenbasis, erstellen die Inhalte der hypermedialen Anwendung. Dazu wurden zuerst das Layout und die Struktur der Datenbasis gemeinsam erarbeitet. Zur Beschreibung der Struktur wurde eine auf XML (siehe Abschnitt 2.3) basierende DTD mit dem Namen *Altenberger Dom Markup Language (ADML)* entwickelt. Das Layout wurde direkt im hypermedialen System umgesetzt.

Die Autoren erstellen ihre Inhalte direkt in ADML. Diese Beschreibung wird in einem automatisierten Prozeß in ein Format konvertiert, welches das Autorenwerkzeug *Director* (siehe [Director98]) der Firma Macromedia verarbeiten kann. Dieser Prozeß wird in [Alfert99] genauer beschrieben. Die fertige Anwendung wird eine mit Director erstellte hypermediale Anwendung sein. Die Konstrukte, die ADML bietet, werden mit Hilfe der Scriptsprache *Lingo* des Autorenwerkzeuges Director operationalisiert.

Im folgenden wird die Navigation auf der Ebene der Beschreibungssprache ADML erläutert und nicht auf der Ebene des Autorenwerkzeuges. Dies ist sinnvoll, weil die Navigation die Strukturen der Datenbasis nutzt, diese Strukturen aber nicht mit derselben Klarheit von den Konstrukten der Scriptsprache Lingo widergespiegelt werden.

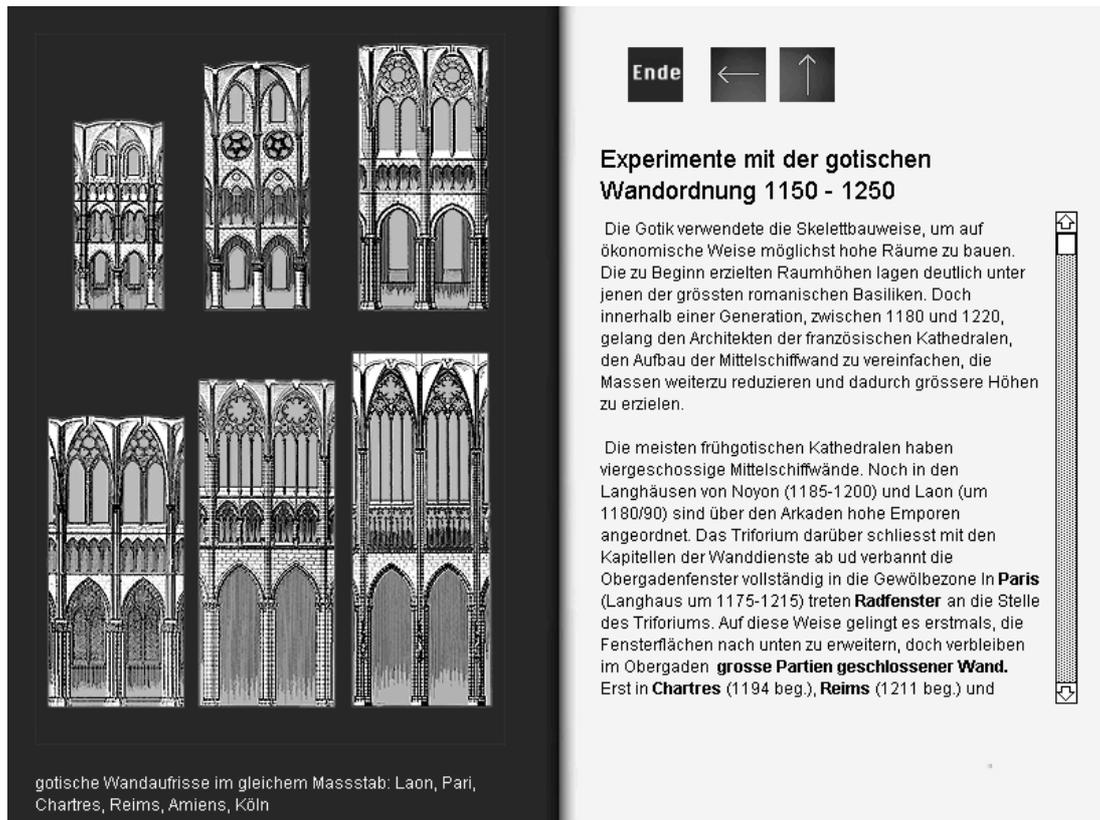


Abbildung 4.1: Die Oberfläche der hypermedialen Altenberger Dom Anwendung

#### 4.1.2.1 Oberfläche

Die Oberfläche des Altenberger Dom wurde in einem frühen Stadium der Projektentwicklung von den Mitarbeitern beider Lehrstühle entworfen und in fünf verschiedene Bereiche aufgeteilt.

Die linke Hälfte besteht aus zwei Bereichen. Der Großteil der zur Verfügung stehenden Fläche wird von dem *Medienpräsentationsbereich* ausgefüllt. In Abbildung 4.1 wird ein Bild im Medienpräsentationsbereich dargestellt (vgl. auch Abbildung 4.2 auf Seite 56). Unterhalb des Mediums befindet sich eine *Medienunterschrift*. Die rechte Hälfte ist in drei Bereiche aufgeteilt. Oben befindet sich eine *Navigationsleiste*, darunter befindet sich die *Themenüberschrift*. Hier steht der Titel des aktuellen Themas oder Unterthemas (vgl. Abschnitt 4.2). Unterhalb der Titelleiste werden die textuellen Informationen des aktuellen Themas im *Informationsbereich* angezeigt. Die Texte können eine beliebige Länge besitzen. Mit Hilfe der Bildlaufleiste (ganz rechts in der Abbildung) kann der sichtbare Bereich gescrollt werden.

Ein zweites Layout für Bibliothekseinträge (vgl. Abschnitt 4.2) war zum Zeitpunkt dieser Arbeit noch nicht vollständig spezifiziert.

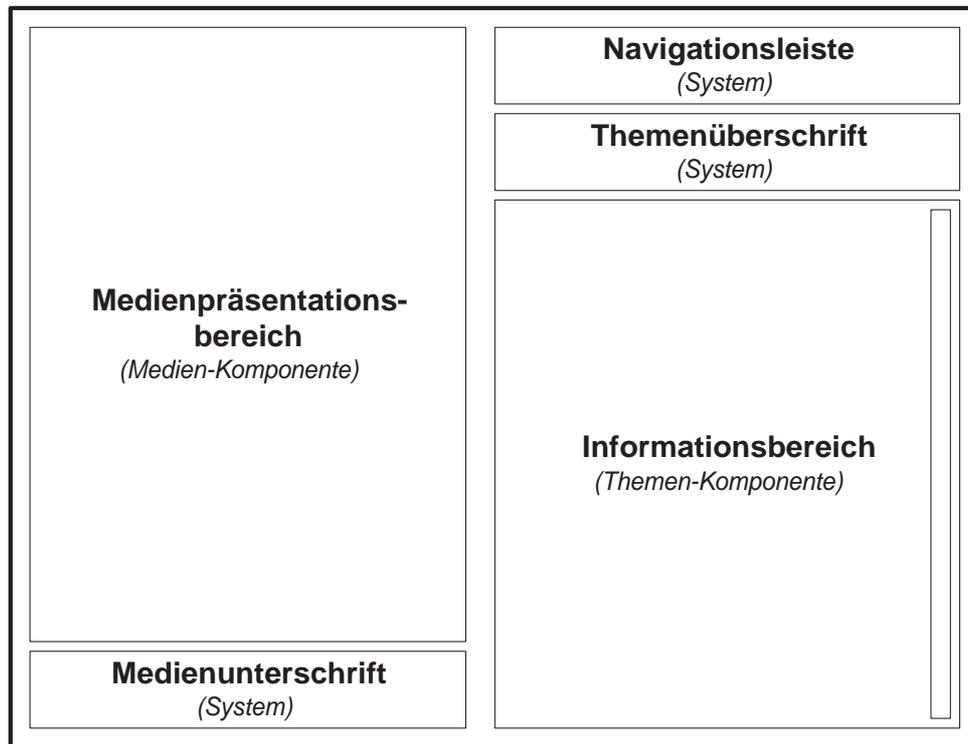


Abbildung 4.2: Schematische Übersicht der Oberfläche der Altenberger Dom Anwendung

#### 4.1.2.2 Architektur

Die Hypermediaanwendung Altenberger Dom wurde mit dem Autorenwerkzeug Macromedia Director erstellt. Director benutzt die Metapher der Filmproduktion. Um einen „Film“ zu erstellen, benutzt man „Darsteller“, „Besetzungen“, ein „Drehbuch“ und eine „Bühne“. Ein Director-Drehbuch, das sich von den Drehbüchern des Industriemeisters 2000 unterscheidet, beschreibt, welcher Darsteller zu welchem Zeitpunkt und für welchen Zeitraum die Bühne betritt. Die Zeitpunkte werden „Frames“ genannt und anhand der Zeitachse durchnummeriert. Die mit Director erstellte Hypermediaanwendung Altenberger Dom benutzt kein Drehbuch. Statt dessen werden die Drehbücher mit Hilfe der Scriptsprache dynamisch erstellt.

Die Datenbasis der Hypermediaanwendung wird mit Hilfe von ADML beschrieben. Jede neue Version der Datenbasis wird konvertiert, so daß die Anwendung schließlich vollständig mit Director erstellt werden kann. Dabei werden die Strukturen der Datenbasis auf Konstrukte des Autorenwerkzeuges Director abgebildet.

Eine nähere Betrachtung der Architektur wird nicht vorgenommen. Navigation beruht auf der Struktur der Datenbasis, in diesem Fall auf der von ADML. Die Altenberger Dom Anwendung basiert jedoch nicht mehr auf dieser Struktur, sondern auf einer konvertierten, die nicht der ursprünglichen entspricht und die Konzepte nicht mit derselben Klarheit

beschreibt. Im Abschnitt 4.3 wird eine virtuelle Maschine beschrieben, die direkt auf der Datenbasis arbeitet und sich damit von der tatsächlichen Realisierung unterscheidet.

## 4.2 Die Datenbasis

Die Datenbasis ist heterogen. Sie besteht aus einer Verbindung von ADML und einer Sammlung von Medien in bestimmten Formaten. In ADML wird sowohl die Struktur der Datenbasis beschrieben als auch Titel, Medienunterschriften und sämtliche Texte des Hyperdokumentes.

Die Datenbasis ist unterteilt in ca. 40 *Themen* mit jeweils mehreren *Unterthemen* sowie in ein zusätzliches *Lexikon*. Ein Thema entspricht inhaltlich einer Besonderheit der Architektur des Altenberger Domes (zum Beispiel „Der Aufbau der Mittelschiffwand“), ein Unterthema einer Verallgemeinerung des Themas (zum Beispiel „Experimente mit der gotischen Wandordnung 1150-1250“, ein Vergleich mit anderen gotischen Bauwerken). *Lexikoneinträge* stellen weitere Informationen zur Verfügung, die sich inhaltlich nicht mehr direkt mit dem Altenberger Dom befassen, sondern allgemeiner Natur sind. Hinter dieser Einteilung in Thema, Unterthema und Bibliothekseintrag steht der Gedanke, daß der Benutzer ausgehend von einer konkreten baulichen Besonderheit des Doms (Thema) über Vergleiche und Grundlagen (Unterthemen) weitergehende Hinweise (Lexikoneinträge) erhält. Themen, Unterthemen und Bibliothekseinträge entsprechen den Knoten des Hyperdokumentes.

Der Unterschied zwischen Themen und Unterthemen ist rein inhaltlicher Natur. Beide werden aus textuellen Informationen und den Medien zusammengesetzt. Eine Themenbeschreibung enthält einen Kopf- und einen Rumpfteil. Im Kopfteil werden die Medien eines Themas zusammen mit einer Medientypbeschreibung und einem innerhalb des Themas eindeutigen Bezeichner (*Medien-ID*) deklariert. Als Medientypen stehen Bilder, Videos, *Quicktime Virtual Reality* (ein Format der Firma Apple, siehe [Stern99]) und Audios zur Verfügung. Der *Rumpfteil* der Themenbeschreibung enthält textuelle Informationen und Verknüpfungen (siehe unten). Themen besitzen einen systemweit eindeutigen Bezeichner, die *Themen-ID* und eine Standard-Medienkomponente, welche automatisch im Medienpräsentationsbereich angezeigt wird, wenn die Themenkomponente dargestellt wird.

Ein Lexikoneintrag besitzt ebenfalls einen systemweit eindeutigen Bezeichner, die *Eintrags-ID*. Weiterhin umfaßt er einen Bibliothekstext mit eventuellen Verknüpfungen sowie einen Glossartext, d. h. eine textuelle Kurzfassung des Bibliothekstextes ohne Verknüpfungen. Der Bibliotheks- oder der Glossartext kann entfallen.

Startanker von Verknüpfungen liegen innerhalb von textuellen Informationen. Es existieren drei Arten von Verknüpfungen (vgl. Abbildung 4.3). Erstens existieren Verknüpfungen zwischen den Themen und Unterthemen, im folgenden *Themenreferenz* genannt. Der zweite Verknüpfungstyp, die *Medienreferenz*, verweist auf ein Medium innerhalb eines (Unter-)Themas. Die *Lexikonreferenz*, der dritte Typ, verweist auf Lexikoneinträge. Dabei wird mit einer zusätzlichen Information, der *Rolle*, spezifiziert, ob es sich um eine Referenz auf einen Glossartext, auf einen Bibliothekstext oder auf beides handelt.

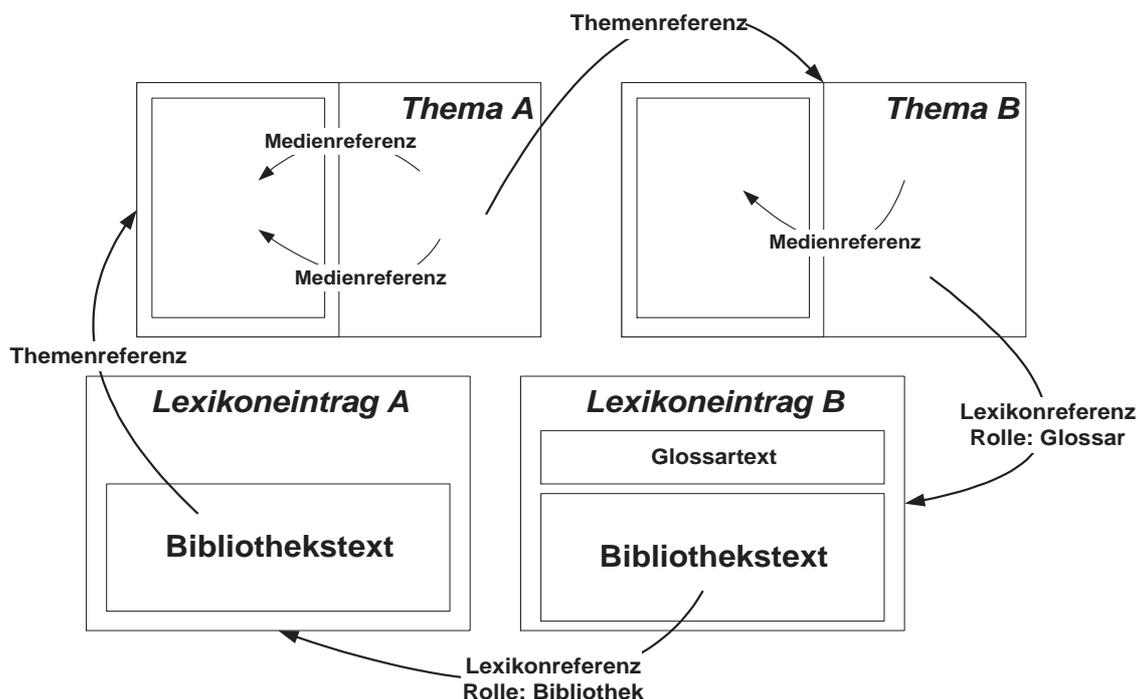


Abbildung 4.3: Schematischer, beispielhafter Ausschnitt aus der Datenbasis

Komponentenverweise haben in der statischen Beschreibungssprache ADML immer die Form einer Komponenten-ID, also einer Medien-ID, Themen-ID oder auch einer Eintrags-ID.

### 4.3 Das hypermediale System

Es wird nun eine *virtuelle ADML Maschine* beschrieben, die in der Lage ist, die ADML-Spezifikation zu nutzen, und entsprechend der Layoutvorgabe aus dem Projekt darzustellen. Dies ist abweichend von der realen Entwicklung der Hypermediaanwendung Altenberger Dom, welche die ADML-Beschreibung in eine Director-Anwendung konvertiert. Neben dieser Konvertierung wurde bereits eine (unvollständige) Konvertierung in ein HTML-Format verwirklicht. Sollten leistungsfähige XML-Browser verfügbar sein, könnte ein hypermediales System erstellt werden, daß die nun beschriebene virtuelle Maschine implementiert. Die weiter unten beschriebenen Hilfsmittel der Navigation könnten dann als „Plug-In“ in das System eingefügt werden. Die Beschreibung einer Maschine, die direkt auf ADML beruht, ist für die weitere Betrachtung hilfreich, da die Navigation auf der Struktur der Datenbasis aufbaut.

### 4.3.1 Die virtuelle ADML Maschine

Die virtuelle ADML Maschine (VAM) orientiert sich am Dexter Modell (vgl. Abschnitt 2.2.1). Die Verwendung des Dexter Modells ist hinreichend, da für die hypermedialen Inhalte der „Ansatz der versteckten Struktur“ (siehe Abschnitt 2.5) genutzt werden kann. Dies zeigt sich zum Beispiel darin, daß keine zeitlichen Informationen in ADML beschrieben werden können. Die Betrachtung anhand des Amsterdamer Hypermedia Modells ist somit nicht notwendig und er wird statt dessen das einfachere Dexter Modell genutzt. Da kein Benutzermodell existiert, ist auch eine Betrachtung anhand des AHAM unnötig.

Das Dexter Modell sieht auch die Erstellung eines Hypertextes mit einem hypermedialen Systems vor. Die VAM besitzt keine Funktionalität zur Erstellung der Datenbasis, somit vereinfacht sich der Aufbau der VAM, da sie nicht als Autorenwerkzeug genutzt wird.

Bevor die VAM für die Präsentation und Interaktion genutzt werden kann, wird in einer ersten Phase, der *Konvertierungsphase*, die in ADML beschriebene Datenbasis geparkt und die Struktur in eine interne, dem Dexter Modell entsprechende Datenstruktur konvertiert. Dies ist möglich, da die Beschreibung einer ADML-Datei endlich ist, und keine Verweise auf weitere ADML-Dateien spezifiziert sind. Für die Medien, Themen, Unterthemen und Lexikoneinträge werden Komponenten angelegt.

**Medien:** Für Medien werden atomare Komponenten angelegt. Sie erhalten aus Themen-ID und Medien-ID zusammengesetzte Komponenten-IDs, damit sind auch diese systemweit eindeutig identifizierbar und besitzen als Inhalt das Medium und als Eigenschaften die Medienunterschrift, ihren Typ „Medium“ und die Themen-ID des Themas, zu dem die Komponente gehört.

**Themen:** Für Themen und Unterthemen werden Kompositum-Komponenten angelegt. Als Komponenten-IDs erhalten sie die Themen-ID. Ihre Inhalte bestehen aus ihren textuellen Informationen und ihren Medienkomponenten. Ihr Titel wird als Eigenschaft der Komponente gespeichert, ebenso ihr Typ „Thema“ und ihre Standard-Medienkomponente. Da Startanker innerhalb eines Textes liegen, können Startanker anhand ihrer Position und Länge im Text bestimmt werden.

**Lexikoneinträge:** Für Lexikoneinträge werden atomare Komponenten angelegt. Sie erhalten die Eintrags-ID als Komponenten-ID. Als Inhalte besitzen sie die Bibliothekstexte. Ihre Eigenschaften sind die Glossartexte und ihr Typ „Lexikoneintrag“. Da Startanker, wie auch bei Themen, innerhalb eines Textes liegen, können sie anhand ihrer Position und Länge im Text bestimmt werden.

**Verknüpfungen:** Für Verknüpfungen werden Verknüpfungskomponenten erstellt. Verknüpfungen besitzen immer einen Startanker und einen Zielverweis, der auf eine Themen-, Medien- oder Lexikoneintragskomponente verweist. Der Typ der Verknüpfung (Themen-, Medien- oder Lexikonreferenz) wird als Eigenschaft gespeichert, ebenso die Rolle, wenn es sich um eine Lexikonreferenz handelt.

Für jede Komponente werden die Auflöser- und Zugriffsfunktion aktualisiert. Alle Komponenten-IDs sind nach Konstruktion her eindeutig.

Ebenfalls während der Konvertierungsphase wird die Eingabedatei auf ihre Korrektheit bzgl. der Hypertext-Invarianten aus dem Abschnitt 2.2.1.4 überprüft. Aufgrund der ADML-Struktur ist die Invariante I3 (Komponenten dürfen nicht Teil von sich selber sein) automatisch erfüllt. Da die Auflöser- und Zugriffsfunktionen während dieser Phase definiert werden, werden die Invarianten I2 (Bildbereich der Auflöserfunktion umfaßt alle Komponenten) und I5 (Komponentenverweise können aufgelöst werden) implizit erfüllt. Es verbleibt also die Überprüfung der Invariante I4 (In Komponentenverweisen genutzte Anker-IDs existieren) und I1 (Komponenten-IDs müssen eindeutig sein). Weiterhin werden die Medien auf ihren Typ (Video, Bild, Ton, QuicktimeVR) hin kontrolliert.

Werden Fehler während der Konvertierungsphase gefunden, so terminiert die VAM mit einer Fehlermeldung. Dieser Schritt ist in der tatsächlichen Realisierung der Hypermediaanwendung Altenberger Dom vorgeschaltet und mündet in der Director Anwendung.

In der Betriebsphase kann nun mit der VAM interagiert werden. Die VAM stellt immer genau eine Instanz einer Themenkomponente inklusive einer Medienkomponente oder einen Bibliothekseintrag dar. Sie nutzt hierfür das im Abschnitt 4.1 beschriebene Layout. Weiterhin wird immer genau eine Instanz einer atomare Komponente im Medienpräsentationsbereich dargestellt. Die im Informationsbereich stehenden textuellen Informationen enthalten Startanker. Eine Startankerinstanz bezieht sich immer auf einen Textabschnitt, der graphisch hervorgehoben wird. Eine Aktivierung einer Verknüpfung zu einem anderen Thema führt zu der Darstellung eines neuen zusammengesetzten Knotens, die Aktivierung einer Medienreferenz führt zu der Darstellung eines neuen Mediums im Medienpräsentationsbereich. Die Aktivierung einer Lexikonreferenz führt zu der Darstellung eines Glossartextes oder eines Bibliothekstextes (siehe Abschnitt 4.4.2).

### 4.3.2 Die Schichten des Dexter Modells

Da sich die VAM am Dexter Modell orientiert, kann im folgenden von einem Schichtenaufbau dieser ausgegangen werden. Für die Speicherung der Inhalte in der Komponenteninhaltsschicht werden Dateien genutzt, die Präsentation erfolgt mit externen Programmen, wie zum Beispiel Quicktime der Firma Apple. Sowohl die Datenbank der Speicherschicht, als auch die Zugriffs- und Auflöserfunktion werden in der Konvertierungsphase erstellt. Die Laufzeitschicht dient dann der Interaktion mit dem Benutzer.

### 4.3.3 Nicht unterstützte Merkmale des Dexter Modell

Aufgrund der Besonderheiten der Anwendung kann auf folgende Merkmale des Dexter Modells für die VAM verzichtet werden:

**Zielanker:** Jede Komponente ist zur Darstellungszeit vollständig sichtbar. Deswegen genügt es für Verknüpfungen auf Komponenten zu verweisen, Zielanker werden nicht benötigt.

**Mehrwertige Verknüpfungen:** Eine ADML-Verknüpfung besteht immer aus genau einem Startanker und genau einer Endpunktbeschreibung mit der Richtung „Ziel“.

**Instanzen:** Die VAM besitzt immer nur eine sichtbare Oberfläche. Das gleichzeitige Betrachten mehrerer Themen oder Bibliothekseinträge wird nicht unterstützt.

**Laufzeitauflöserfunktion:** Es wird keine Laufzeitauflöserfunktion implementiert, da nur statische Verweise genutzt werden.

#### 4.3.4 Navigationsschritte

Eine Beschreibung der konkreten Navigationsschritte im Altenberger Dom kann nicht erfolgen, weil die tatsächliche Realisierung nicht betrachtet wird. Der Aufbau eines Navigationsschrittes wird aber in den folgenden Kapiteln analysiert, und die im Abschnitt 6.1.5 beschriebene `Navigationsschritt`-Funktion erfüllt die Anforderungen an eine solche Funktion für den Altenberger Dom.

### 4.4 Hilfsmittel der Navigation

Die in diesem Abschnitt aufgeführten Hilfsmittel aus dem Bereich der Navigation sind in Gesprächen der Mitarbeiter beider Lehrstühle entwickelt worden. Es wurden noch keine Dokumentationen zu diesen Gesprächen verfaßt, so daß viele Details dieses Kapitels aus persönlichen Gesprächen des Autors mit den Mitarbeitern des Lehrstuhls für Softwaretechnologie entstanden sind.

Die hypermediale Anwendung Altenberger Dom stellt dem Benutzer die Hilfsmittel *Historie*, *Lexikonzugriff*, *Titelleiste* und *virtueller Dom* zur Verfügung. Geplant sind weiterhin noch die Hilfsmittel *Führer*, *Lexikonsuche*, *Lesezeichen* und *Notizblock*. Die geplanten Hilfsmittel werden nur kurz umrissen.

Die Funktionalität und das Layout der Lexikoneinträge sind noch nicht vollständig spezifiziert. Inwieweit die Hilfsmittel dem Lexikon noch angepaßt werden, ist ebenfalls noch nicht spezifiziert. Die nachfolgenden Hilfsmittel konzentrieren sich somit auf die Verarbeitung von Themen. Bekannte Wechselwirkungen mit den Lexikoneinträgen werden explizit erwähnt.

Die Hilfsmittel werden wiederum anhand der in Abschnitt 3.4.1 beschriebenen Form dokumentiert.

#### 4.4.1 Historie

**Zweck** Das Historie-Hilfsmittel dient der Rücknahme von Navigationsschritten. Dabei wird zwischen den verschiedenen Verknüpfungstypen unterschieden.

**Gründe für die Auswahl** Eine Historie-Funktionalität ist in nahezu jedem modernen hypermedialen System integriert und kann somit als Standard betrachtet werden.

**Beschreibung der Umsetzung** Das Historie-Hilfsmittel ist in der Navigationsleiste untergebracht. Es besteht aus zwei Schaltflächen (siehe Abbildung 4.4), von denen die linke Schaltfläche immer sichtbar ist und die rechte eingeblendet wird, sobald zu einem ersten Thema in der Sitzung navigiert wird. Wird dieser Navigationsschritt mit diesem Hilfsmittel zurückgenommen, so wird die Schaltfläche wieder ausgeblendet. Die linke Schaltfläche besitzt als Symbol einen nach links weisenden Pfeil, die rechte einen aufwärts zeigenden Pfeil. Mit Hilfe der linken Schaltfläche können unabhängig von dem Verknüpfungstyp einzelne Navigationsschritte rückgängig gemacht werden, insbesondere auch solche, die durch aktivierte Medienreferenzen entstanden sind. Mit der rechten werden nur die durch aktivierte Themenreferenzen entstandenen Navigationsschritte rückgängig gemacht.

Zur Realisierung dieses Hilfsmittel wird ein Kellerspeicher verwendet. Für jede aktivierte Verknüpfung wird ein Komponentenverweis, bestehend aus der Komponenten-ID, auf den Kellerspeicher gelegt zusammen mit einer Typbeschreibung der Verknüpfung. Das Betätigen der linken Schaltfläche führt dazu, daß das oberste Element von dem Kellerspeicher entnommen und danach der oberste Komponentenverweis unabhängig vom Typ der Verknüpfung aufgelöst und der zugehörige Knoten angezeigt wird. Das Betätigen der rechten Schaltfläche führt dazu, daß solange Elemente des Kellerspeichers entnommen werden, bis der oberste Verweis eine Themenreferenz auf ein anderes Thema als das aktuell dargestellte ist. Der Startzustand des Hilfsmittels ist ein leerer Kellerspeicher.

**Genutzte Informationen** Das Historie-Hilfsmittel arbeitet mit dynamisch erzeugten Informationen über die Navigation des Benutzers. Die einzelnen Navigationsschritte werden automatisch protokolliert.

**Interaktion** Das Hilfsmittel interagiert nur mit der

LAUFZEITSCHICHT: Das Hilfsmittel wird nach jedem Navigationsschritt über die aktuell dargestellte Komponente informiert. Weiterhin wurde eine Funktion für das Hilfsmittel bereitgestellt, um zu einer Komponente zu navigieren.

**Besonderheiten** Dieses Hilfsmittel nutzt eine Besonderheit der Datenbasis. Die Unterscheidung von Verknüpfungstypen erlaubt eine schnelle Rücknahme der Navigation. Die Zusammenarbeit mit Lexikonreferenzen ist noch nicht geklärt.



Abbildung 4.4: Das Historie-Hilfsmittel innerhalb der Navigationsleiste

#### 4.4.2 Lexikonzugriff

Der Lexikonzugriff ist zur Zeit in der Altenberger Dom Anwendung noch nicht umgesetzt worden. Sein Aufbau spiegelt sich aber in der ADML-Struktur wider und es existieren konkrete Umsetzungspläne. Daher kann das Hilfsmittel mit Hilfe der VAM beschrieben werden.

**Zweck** Der Altenberger Dom zeigt an einem konkreten Beispiel Prinzipien der gotischen Bauweise. Weiterhin wurden auch Informationen allgemeiner Natur mit in die Datenbasis aufgenommen. Das Lexikon enthält die weiterreichenden Informationen, die inhaltlich nicht direkt mit dem Altenberger Dom zusammenhängen. Der Zugriff auf diese Informationen aus einem normalen Text heraus wird mit Hilfe des Lexikonzugriffs verwirklicht.

**Beschreibung der geplanten Umsetzung** Das Lexikon besteht aus Einträgen, die wiederum aus Glossar- oder Bibliothekstexten bestehen. Auf die Einträge wird mit Hilfe von Lexikonreferenzen verwiesen. Der Referenz ist eine Rolle zugeordnet, welche die Möglichkeiten des Benutzers bestimmt. Es existieren genau drei verschiedene Rollen: *glossary*, *library* und *gloslib*. Geplant ist für aktivierte Referenzen der Rolle „glossary“ einen Tooltip des Glossartextes anzuzeigen und für aktivierte Referenzen der Rolle „library“ einen Navigationsschritt zu dem Bibliothekstext durchzuführen. Dies soll auf zwei verschiedene Aktivierungsweisen geschehen, denn für die dritte Rolle „gloslib“ ist die Aktivierungsart entscheidend. Je nachdem, wie die Ankerinstanz aktiviert wird, wird entweder ein Tooltip angezeigt oder ein Navigationsschritt durchgeführt.

Das Hilfsmittel nutzt die vom Dexter Modell nicht näher beschriebenen Darstellungsbeschreibungen der Instanzierungsfunktion. Für Lexikonreferenzen der Rollen „glossary“ und „gloslib“ übergibt das Hilfsmittel die Tooltip-Texte, also die Glossartexte, als Darstellungsbeschreibungen. Die „library“-Rolle kann ignoriert werden. Dazu wird der Lexikonzugriff vor jeden Navigationsschritt über die darzustellende Komponente benachrichtigt, in der Datenbank der Speicherschicht werden die von der Komponente ausgehenden Lexikonreferenzen ermittelt und die Texte ausgelesen. Diese werden dann der Instanzierungsfunktion übergeben, die daraufhin die Startanker mit der Tooltip-Funktionalität versieht.

**Genutzte Informationen** Die Informationen, mit denen dieses Hilfsmittel arbeitet, werden von dem Autor der Datenbasis explizit bereitgestellt. Der Autor muß sich für eine der drei Rollen entscheiden.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Die Laufzeitschicht informiert das Hilfsmittel vor jedem Navigationsschritt über die zu besuchende Komponente. Als Rückgabewert werden Darstellungsbeschreibungen zurückgegeben, die von der Instanzierungsfunktion als Tooltip-Texte genutzt werden.

**SPEICHERSCHICHT:** Das Hilfsmittel nutzt die Datenbank der Speicherschicht, um die benötigten Lexikonreferenzen zu ermitteln. Anschließend werden eventuell vorhandene Glossartexte, die als Komponenteneigenschaften gespeichert wurden, ausgelesen.

**Besonderheiten** Die explizite Einführung einer Rolle gibt dem Autor der Datenbasis Einfluß auf die Navigationsmöglichkeiten des Benutzers. Die Instanzierungsfunktion muß Tooltip-Darstellungsinformationen verarbeiten können und die Interaktion des Benutzers mit der Instanz muß eine Darstellung der Tooltips erlauben.

#### 4.4.3 Titelleisten

**Zweck** Die Titelleisten informieren den Benutzer über das aktuelle Thema und die aktuelle Medienkomponente, welche ihm vom System dargestellt werden. Es ist keine Navigation mit den Titelleisten möglich. Wie dieses Hilfsmittel mit dem Lexikoneinträgen zusammenarbeitet, ist noch nicht geklärt.

**Gründe für die Auswahl** Die Titelleisten stellen zu jeder Zeit Informationen über das Medium und das Thema dar. Somit dient dieses Hilfsmittel zur Vermeidung des Problems der Desorientierung (siehe Abschnitt 2.4.2).

**Beschreibung der Umsetzung** Die Themenüberschrift und die Medienunterschrift bilden zusammen das Titelleisten-Hilfsmittel des Altenberger Doms. Die Themenüberschrift befindet sich unterhalb der Navigationsleiste. Sie zeigt den Titel eines Themas oder Unterthemas an. Die VAM zeigt den Titel im Vergleich zu den restlichen textuellen Informationen gemäß den Layoutvorgaben mit einer größeren Schrift. Die Medienunterschrift befindet sich unterhalb des Medienpräsentationsbereiches und zeigt eine Anmerkung zu dem präsentierten Medium an.

**Genutzte Informationen** Der Titel eines Themas wird innerhalb von ADML spezifiziert. Die Autoren der Datenbasis vermerken für jedes Thema einen Titel innerhalb des Kopfteils der ADML Beschreibung. Medienunterschriften sind ebenfalls vom Autor geschriebene Texte, die für jedes Medium niedergeschrieben wurden.

**Interaktion** Das Hilfsmittel interagiert mit folgenden Objekten:

**LAUFZEITSCHICHT:** Das Hilfsmittel wird nach jedem Navigationsschritt über die aktuell dargestellte Komponente informiert.

**SPEICHERSCHICHT:** Es wurde eine Funktion bereitgestellt, die dem Hilfsmittel die Titel der Komponenten zurückliefert.

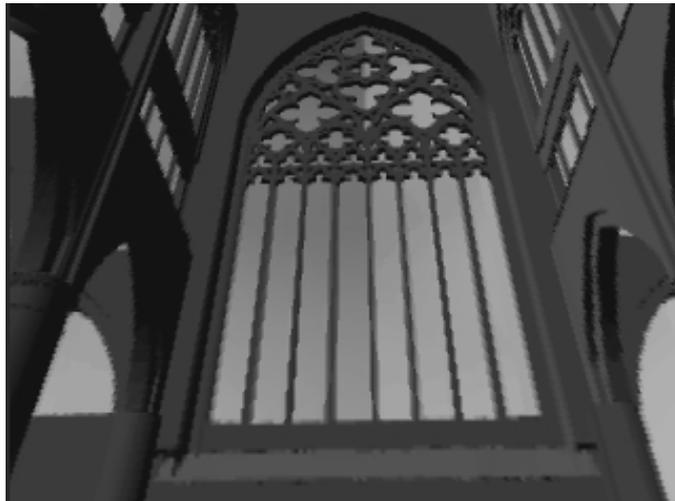


Abbildung 4.5: Eine Ansicht des virtuellen Doms-Hilfsmittels

#### 4.4.4 Virtueller Dom

**Zweck** Der virtuelle Dom dient der Navigation zu den einzelnen Themen der Datenbasis. Dazu wird dem Benutzer ein virtuelles Modell des Bauwerkes präsentiert, an dem er beachtenswerte bauliche Besonderheiten besichtigen kann. Es dient als Navigationshilfe zu den einzelnen Themen. Aus der Sicht des Benutzers ist der virtuelle Dom ein multimedial aufbereitetes Inhaltsverzeichnis.

**Beschreibung der Umsetzung** Vom Altenberger Dom wurde ein virtuelles Modell angefertigt, welches im Medienpräsentationsbereich der Startseite zu sehen ist. Das Modell erlaubt eine volle 360° Rundumsicht von einer Reihe von Standpunkten innerhalb und außerhalb des Domes aus. Die Standorte können innerhalb des Modells mit einem Mausklick oder direkt über eine einblendbare Übersichtskarte erreicht werden. Das virtuelle Modell wird mit Hilfe der Maus gesteuert. Mit gedrückter Maustaste kann der Blickwinkel durch Mausbewegungen geändert werden. Auch Bewegungen in der Vertikalen sind möglich. Mit der Umschalttaste auf der Tastatur kann in das Modell „hineingezoomt“, ein kleinerer Sichtbereich somit vergrößert angezeigt werden. Mit der Steuerungstaste kann wieder aus dem Modell „herausgezoomt“ werden. Der Mauszeiger verändert sein Aussehen, wenn er sich über einen Standpunkt bewegt: ein Pfeil in Richtung des Standpunktes wird angezeigt. Ein Mausklick verändert dann den Standpunkt innerhalb des Modells. Bewegt sich der Mauszeiger über eine Verknüpfung zu einem Thema, so verändert sich der Mauszeiger zu einer Hand. Ein Mausklick aktiviert dann die Verknüpfung. Der Startzustand ist ein besonderer Standpunkt, mit einer besonderen Blickrichtung. Zum derzeitigen Stand ist dies der Standpunkt am Haupteingang innerhalb des Doms, mit Blick in den Dom hinein. Dies kann sich aber noch ändern.

Das virtuelle Modell besitzt eine besondere Rolle in der ADML-Spezifikation. Es wird als spezieller Navigationsabschnitt außerhalb der restlichen Themenbeschreibungen und

Lexikoneinträgen notiert.

Das virtuelle Modell wird auf der Startseite im Medienpräsentationsbereich dargestellt. Wählt der Benutzer ein Thema aus, so wird ein Navigationsschritt ähnlich wie bei einer aktivierten Verknüpfung durch die Laufzeitschicht ausgelöst.

**Genutzte Informationen** Das Modell wurde von den Autoren der Datenbasis erstellt. Sie definierten die Standorte, von denen der Benutzer den Dom besichtigen kann, und verknüpften bestimmte Bereiche des Modells mit bestimmten Themen. Dabei wurden nur Themen und keine Unterthemen verknüpft. Alle diese Informationen wurden in dem Modell verarbeitet.

**Interaktion** Das Hilfsmittel interagiert mit der

LAUFZEITSCHICHT: Es wurde eine Funktion für das Hilfsmittel bereitgestellt, um zu einer Komponente zu navigieren.

**Besonderheiten** Intern kann der virtuelle Dom als ein eigenes hypermediales Subsystem betrachtet werden. Die einzelnen Standpunkte entsprechen unter diesem Gesichtspunkt Knoten, die Ankerinstanzen werden mit verändertem Mauszeiger gekennzeichnet. Aus Sicht des hypermedialen Systems des Altenberger Doms aber wird der virtuelle Dom als eine Komponente angesteuert. Die Komponenteninhaltsschicht stellt die Komponente dar und interagiert mit dem Benutzer. Der virtuelle Dom liegt als einziges der betrachteten Hilfsmittel innerhalb der Komponenteninhaltsschicht.

#### 4.4.5 Weitere geplante Hilfsmittel

Weitere Hilfsmittel sind geplant. Während die Aufgaben der Hilfsmittel bereits geklärt wurden, steht die Planung der konkreten Umsetzungen noch aus. Deswegen werden sie auch nicht genauer beschrieben. Teilweise existieren äquivalente Hilfsmittel im Hinblick auf Struktur, Navigation und Interaktion mit dem Benutzer im Industriemeister 2000 Projekt.

**Führer** Das Führerhilfsmittel entspricht dem Geführte Tour-Hilfsmittel des Industriemeisters 2000. Einzelne Stationen (Themen und Lexikoneinträge) werden nacheinander sequentiell durchlaufen. Der Name entstand aus der geplanten Umsetzung, einen Führer in der Bibliothek ansprechen und mit diesem eine Geführte Tour unternehmen zu können.

**Lexikonsuche** Innerhalb des Lexikons sollen Glossar- und Bibliothekseinträge durchsucht werden. Dieses Hilfsmittel könnte ähnlich wie das Glossar-Hilfsmittel des Industriemeisters 2000 umgesetzt werden.

**Lesezeichen** Lesezeichen sollen ähnlich wie das gleichnamige Hilfsmittel des Industriemeisters 2000 realisiert werden.

**Notizblock** Der Notizblock wird ein erweitertes Lesezeichen sein. Dieses Hilfsmittel wird es erlauben, Anmerkungen zu Themen zu notieren, ohne jedoch die eigentliche Datenbasis zu verändern. Der Benutzer kann textuelle Informationen zum Thema vermerken, die mit Verknüpfungen zu anderen Themen versehen werden können. Die Notizen werden für jeden Benutzer getrennt verwaltet werden.



---

## Kapitel 5

# Vergleich der beiden Anwendungen

In diesem Kapitel werden die beiden in den vorangehenden Kapiteln getrennt voneinander beschriebenen Anwendungen miteinander verglichen. Der Vergleich orientiert sich an dem Aufbau der beiden Beschreibungen und vergleicht die Datenbasis, die hypermedialen Systeme und die Hilfsmittel. Der Vergleich der Hilfsmittel erfolgt anhand eines Klassifikationsschemas, welches charakteristische Besonderheiten der Hilfsmittel aus Sicht der Navigation beschreibt und ebenfalls in diesem Kapitel vorgestellt wird. Das nächste Kapitel wird auf dem Vergleich der Hilfsmittel aufbauen.

### 5.1 Datenbasis

Beide untersuchten Systeme beruhen auf einer erweiterten Grundstruktur (siehe Abschnitt 2.3), die sich von dem einfachen Modell der Knoten und Verknüpfungen unterscheidet. Im Industriemeister 2000 wird eine (Vorgänger, Nachfolger)-Relation erstellt und im Altenberger Dom werden sowohl Knoten als auch Verknüpfungen verfeinert.

Die beiden Anwendungen unterscheiden sich deutlich in der Erstellung der Datenbasen. Im Altenberger Dom Projekt wird die Datenbasis direkt von den Autoren erstellt. Sie erzeugen die Medien und beschreiben die Themen, Unterthemen und Lexikoneinträge direkt mit der Beschreibungssprache ADML, wodurch sich dieses Projekt von dem Industriemeister 2000 unterscheidet. Im Industriemeister 2000 Projekt erstellten die Autoren der Datenbasis Drehbücher, welche dann „umgesetzt“ wurden. Dies bedeutete einen zusätzlichen Arbeitsschritt und erforderte zusätzliche Arbeitskräfte. Umgekehrt konnten sich die Autoren auf die inhaltliche Arbeit konzentrieren.

#### 5.1.1 Knoten

Die Datenbasis des Industriemeisters 2000 entstand durch eine multimediale Aufbereitung vorwiegend linear aufgebauter konventioneller Lehrbücher. Dieser Aufbau der Lehrbücher

wurde übertragen, wodurch eine Struktur aus Kapiteln, Unterkapiteln und Seiten entstand. Die Knoten des Hyperdokumentes sind ausschließlich Seiten. Kapitel können selbst nicht dargestellt werden, dienen aber der hierarchischen Darstellung im Inhaltsverzeichnis und als zusätzliche Information in der Titelleiste.

Die Altenberger Dom Anwendung unterteilt die Datenbasis in Themen, Unterthemen und Lexikoneinträge. Ursprünglich waren Themen sogar, ähnlich den Kapiteln des Industriemeisters 2000, eine rekursive Struktur. Auf die rekursive Struktur wurde aber zugunsten eines zweistufigen Aufbaus verzichtet, da die Autoren der Datenbasis sich für einen durchgängig zweistufigen inhaltlichen Aufbau entschieden und den Autoren so die Einhaltung der inhaltlichen Form erleichtert wird. Während Themen und Unterthemen von der Struktur her gleichwertig sind und sich nur inhaltlich und in der verschachtelten Notation unterscheiden, sind die Lexikoneinträge in Form und Inhalt von den Themen her verschieden. Somit wurde das inhaltliche Gestaltungskonzept, vom speziellen baulichen Merkmal über Verallgemeinerungen zu weiterreichenden allgemeinen Informationen hin, in Strukturen der Datenbasis umgesetzt.

### 5.1.2 Verknüpfungen

Die Trennung der Datenbasis in Kursen führte zu einer Sammlung von unabhängigen Datenmengen, die getrennt betrachtet werden können. Innerhalb eines Kurses existieren implizite Verknüpfungen, die eine (Vorgänger, Nachfolger)-Relation beschreiben.

ADML spezifiziert drei verschiedene Verknüpfungstypen. So werden zusätzlich zu den normalen Verknüpfungen (Themenreferenzen), Medienreferenzen und Lexikonreferenzen eingeführt. Die drei Typen besitzen eine unterschiedliche Semantik dadurch, daß sie Referenzen auf die drei verschiedenen Klassen von Knoten widerspiegeln. Da die Komponenten bereits getypt sind, sind die verschiedenen Verknüpfungstypen redundant. Lediglich die Rollenbeschreibung der Lexikonreferenz ist nicht redundant, da sie vom Autor spezifiziert werden muß. Die Redundanz kann wiederum den Autoren der Datenbasis die Arbeit erleichtern, da eine einfache Überprüfung der inhaltlichen Konzeption bereits während des Erstellens der Datenbasis überprüft werden kann.

### 5.1.3 Weitere Strukturen

In beiden Anwendungen wurden Strukturen für Geführte Touren erstellt. Geführte Touren sind somit ein Bestandteil der Datenbasis und müssen von den Autoren explizit während der Erstellung der Datenbasis berücksichtigt werden.

### 5.1.4 Zusammenfassung

Inhaltliche Strukturen hatten zur Folge, daß für die Datenbasis nicht nur „einfache“ Knoten und Verknüpfungen genutzt wurden, sondern diese verfeinert wurden. Drei Motivationen lassen sich hierfür identifizieren.

1. Durch die Strukturen lassen sich zusätzliche Informationen, beispielsweise Hierarchien und implizite Verknüpfungen, modellieren. Eine inhaltliche Semantik kann direkt auf den Strukturen spezifiziert werden. Eine Reihe von den oben beschriebenen Hilfsmitteln nutzen genau diese Informationen.
2. Strukturen können in Darstellungsbeschreibungen übertragen werden. Darstellungsbeschreibungen parametrisieren die Instanzierungsfunktion des Dexter Modells (siehe Abschnitt 2.2.1) und verändern damit die Präsentation der Komponente. Unterschiedliche Strukturen können verschieden dargestellt werden. Die Altenberger Dom Anwendung stellt Themen zusammen mit austauschbaren Medien dar und die Struktur „Thema“ wird anders dargestellt als die Struktur „Medium“.
3. Die Komplexität der Erstellung einer größeren Datenbasis kann durch die Strukturen in kleinere Teilprobleme aufgeteilt werden. Teilmengen können getrennt von anderen realisiert werden und die Einhaltung der inhaltlichen Konzepte kann überprüft werden. Dieser Punkt tritt besonders in der Altenberger Dom Anwendung in den Vordergrund. Hier wurde die Beschreibungssprache ADML den speziellen Bedürfnissen der inhaltlichen Konzeption angepaßt. Sie läßt wenig Spielraum für eine andere als die geplante Nutzung.

Somit bieten Strukturen viele Möglichkeiten der Anpassung des Systems, führen aber auch zu einer Spezialisierung des Systems auf spezielle Datenmengen. Eine spezielle Datenbeschreibungssprache, die für ein Anwendungsgebiet erstellt wurde, kann nicht unbedingt für andere Anwendungsbereiche wiederverwendet werden, eine zu allgemeine schränkt die Möglichkeiten des Systems ein. Im Altenberger Dom Projekt entschied man sich für eine sehr spezielle Beschreibungssprache, im Industriemeister 2000 Projekt hingegen für eine allgemeinere Beschreibungssprache die „lediglich“ eine sequentielle Bearbeitung der Inhalte nahelegt.

## 5.2 Die hypermedialen Systeme

Dieser Abschnitt betrachtet die Modellbeschreibungen, die Instanzierungs- und Navigationsschrittfunktionen sowie die Hilfsmittel der beiden hypermedialen Systeme näher.

Eine solche Betrachtung wird dadurch erschwert, daß sich die beiden Anwendungen deutlich in zwei Punkten unterscheiden:

1. Während die Industriemeister 2000 Anwendung bereits fertig erstellt wurde, befindet sich die Altenberger Dom Anwendung noch in der Entwicklung. Somit konnten nicht alle Wechselwirkungen innerhalb der Altenberger Dom Anwendung untersucht werden.
2. Das hypermediale System des Industriemeisters 2000 wurde anhand der realen Umsetzung beschrieben, für das Altenberger Dom System wurde eine virtuelle Maschine genutzt, mit der das System einfacher und zweckmäßiger beschrieben werden konnte, als es die reale Umsetzung erlaubt hätte.

Da der direkte Vergleich deshalb nicht sehr aussagekräftig ist, werden nur die wichtigsten Punkte herausgestellt.

### 5.2.1 Die genutzten Modelle

Für beide Systeme genügt die Betrachtung anhand des Dexter Modells. Obschon es sich um hypermediale Systeme handelt, ist eine Betrachtung anhand des komplexeren Amsterdamer Hypermediamodells nicht notwendig, da in beiden Fällen werden die hypermedialen Aspekte, Interaktion und zeitliche Synchronisation der Medien, von speziellen Autorenwerkzeugen übernommen. Im Industriemeister 2000 wird zeitliche Synchronisation innerhalb der Komponenteninhaltsschicht durch die Toolbook-Skriptsprache Openscript beschrieben, während im Altenberger Dom eine vollständige Beschreibung der Datenbasis ohne zeitliche Information in Form eines ADML-Dokumentes erstellt wurde. Zusätzlich zum Dexter Modell dient das AHAM zur Beschreibung der Adaptivität. Die Seiten werden durch die Darstellungsbeschreibung des Lernschrittes beeinflusst. Zusätzlich nutzt das System eine „Systemadaption“, in der Hilfsmittel durch einen eigenen Adaptionsgestalter beeinflusst werden. Die Systemadaption ist neben Inhalts- und Verweisadaption eine dritte Adaptionart, die allerdings noch nicht in [DeBra99] beschrieben wurde.

Die Aussage der Autoren, daß das Dexter Modell mächtiger sei, als notwendig wäre, hat sich auch für diese Arbeit gezeigt. Beide Anwendungen erfüllen nicht alle Merkmale des Systems. So verzichten beide Anwendungen auf eine gleichzeitige Präsentation mehrerer Instanzen, auf mehrwertige Verknüpfungen und Verknüpfungen auf Verknüpfungen. In beiden Anwendungen sind Zielanker unnötig gewesen, da alle Knoten eine dem Layout entsprechend feste und auf dem Bildschirm darstellbare Größe besaßen. Unterschieden haben sich die Systeme in der Verwendung von Kompositum-Komponenten und Verknüpfungen als Komponenten. Diese beiden Merkmale wurden nur von der VAM genutzt.

### 5.2.2 Instanzierungsfunktion

In beiden Anwendungen wird die Instanzierungsfunktion mit Darstellungsbeschreibungen parametrisiert. Im Industriemeister wird durch den Anpassungsgestalter die Seite in den richtigen Lernschritt versetzt und im Altenberger Dom können Tooltips spezifiziert werden.

### 5.2.3 Navigationsschritte

Das System des Industriemeisters 2000 besitzt nicht eine, sondern mehrere vergleichbare Funktionen, die mit weiteren Funktionen, die in verschiedenen Bestandteilen der Anwendung implementiert sind, zusammenarbeiten. Die VAM, eine virtuelle Maschine ohne Implementierung, besitzt demnach noch keine implementierte Navigationsschrittfunktion.

Beiden Anwendungen ist aber gemeinsam, daß Hilfsmittel einzelne Navigationsschritte auslösen können. Ein Navigationsschritt kann also nicht nur durch aktivierte Verknüpfungen aktiviert werden, sondern kann auch von einem Hilfsmittel ausgelöst werden, zum Beispiel durch Benutzerinteraktion mit dem Hilfsmittel.

## 5.3 Klassifikationsschema für Hilfsmittel

Im folgenden wird ein Klassifikationsschema beschrieben, welches den weiteren Betrachtungen zugrunde liegt. Es umfaßt die wesentlichen Besonderheiten aus den Dokumentationen der beiden vorangegangenen Kapiteln. Es dient zur Einordnung der Hilfsmittel im Abschnitt 5.4 und als Basis der Betrachtungen in Abschnitt 5.5, in dem die Eigenschaften der Hilfsmittel anhand der hier beschriebenen Klassifikationen verglichen werden.

Das Klassifikationsschema besteht aus mehreren Klassen in fünf Kategorien. Die Kategorien lauten *Darstellungsbeschreibungen*, *Zustand*, *Systeminteraktionen*, *Benutzerinteraktionen* und *Adaptivität*. Sie beschreiben wesentliche Merkmale der identifizierten Hilfsmittel.

Die Qualität des Schemas ist nicht überprüfbar. Es existiert keine zielgerichtete Herleitung für das Schema, deren Systematik man hätte beurteilen können, noch lassen sich bessere Schemas ausschließen. Einzige Indikatoren für die Qualität des Schemas sind die Unabhängigkeit der einzelnen Kategorien voneinander, die eindeutige Unterteilung in Klassen innerhalb der Kategorien und das sich für jede Klassen in jeder Kategorie Hilfsmittel finden lassen. Diese Kriterien werden anhand der Klassifikation im Abschnitt 5.4 überprüft.

Zu den einzelnen Begriffen werden Beispiele aus dem hypermedialen System des Industrie-meisters aufgeführt.

### 5.3.1 Darstellungsbeschreibungen

Einige Hilfsmittel beeinflussen die Darstellung der Komponenten, indem sie Darstellungsbeschreibungen für die Instanzierungsfunktion erzeugen.

**Darstellungsbeeinflussend:** Ein darstellungsbeeinflussendes Hilfsmittel erzeugt Darstellungsbeschreibungen, welche die Instanzierungsfunktion parametrisieren. Ein Beispiel für ein Hilfsmittel, welches die Darstellung beeinflusst, ist das Lexikonzugriff-Hilfsmittel des Altenberger Doms. Es erzeugt Tooltip-Informationen, die von der Instanzierungsfunktion verarbeitet werden.

**Darstellungsneutral:** Ein Hilfsmittel, welches keine Darstellungsbeschreibungen für die Instanzierungsfunktion erzeugt, wie das auf einen Navigationsschritt reagierende Tittleiste-Hilfsmittel, ist darstellungsneutral.

### 5.3.2 Zustand

Einige Hilfsmittel besitzen einen Zustand, der durch Navigation beeinflusst wird oder selbst die Navigation beeinflusst.

**Privater Zustand:** Ein privater Zustand eines Hilfsmittels beeinflusst die Navigation, oder wird durch Navigation beeinflusst. Er beeinflusst aber nicht die Funktionalität eines anderen Hilfsmittels. Ein Beispiel für ein Hilfsmittel mit einem privaten Zustand ist das Historie-Hilfsmittel.

**Öffentlicher Zustand:** Ein öffentlicher Zustand eines Hilfsmittels beeinflusst die Funktionalität eines anderen Hilfsmittels. Das Geführte Tour-Hilfsmittel besitzt als öffentlichen Zustand die eventuell laufende Tour.

**Externer Zustand:** Ein Hilfsmittel kann durch den Zustand eines anderen in seiner Funktionalität beeinflusst werden. Die Titelleiste zeigt in Abhängigkeit von dem Zustand des Geführten Tour-Hilfsmittels einen Titel entweder einer Geführten Tour oder eines Kapitels.

**Kein Zustand:** Ein Hilfsmittel, daß keinen solchen Zustand besitzt, ist aus Sicht der Navigation ohne Zustand. Ein Beispiel hierfür ist das Glossar-Hilfsmittel.

Hilfsmittel mit einem privaten oder einen öffentlichen Zustand werden im folgenden als *zustandsbasiert* bezeichnet. Zustandsbasierte Hilfsmittel besitzen einen Startzustand, in dem sie sich zu Beginn einer Sitzung befinden.

### 5.3.3 Systeminteraktion

Hilfsmittel interagieren mit dem hypermedialen System. Betrachtet man die Schichten des Dexter Modells als einzelne Objekte zur Laufzeit, so können Hilfsmittel anhand ihrer Interaktion mit den Schichten verglichen werden.

**Dynamische Systeminteraktion:** Hilfsmittel, die ausschließlich mit der Laufzeitschicht interagieren, besitzen eine dynamische Interaktion. Sie können einen Navigationsschritt auslösen und mit Darstellungsbeschreibungen parametrisieren sowie dynamisch anfallende Daten verarbeiten. Als Beispiel für ein Hilfsmittel, welches ausschließlich dynamische Interaktion nutzt, kann das Historie-Hilfsmittel angesehen werden.

**Statische Systeminteraktion:** Ein Hilfsmittel, welches nur mit der Speicherschicht, nicht aber mit der Laufzeitschicht interagiert, wurde in den beiden untersuchten Anwendungen nicht identifiziert. Ein solches Hilfsmittel könnte weder einen Navigationsschritt auslösen noch dynamische Informationen verarbeiten, wie zum Beispiel die Information des aktuell dargestellten Knotens. Es wäre aber eine Auswertung der statischen Datenbasis und des Benutzermodells möglich, um beispielsweise Statistiken zu erstellen und diese mit dem Benutzermodell zu vergleichen.

**Erweiterte Systeminteraktion:** Hilfsmittel, die mit der Laufzeitschicht und mit der Speicherschicht interagieren, besitzen eine erweiterte Interaktion. Das Geführte Tour-Hilfsmittel kann einen Navigationsschritt auslösen und liest Informationen über Geführte Touren aus der Datenbasis aus.

Interaktionen mit anderen Hilfsmitteln werden nicht betrachtet, da die in Abschnitt 6.1.5 vorgestellte Navigationsschrittfunktion eine direkte Interaktion zwischen Hilfsmitteln unnötig macht.

### 5.3.4 Benutzerinteraktion

Der Benutzer kann mit den Hilfsmitteln interagieren. Drei verschiedene Interaktionsarten konnten identifiziert werden:

**Zustandsverändernde Benutzerinteraktion:** Ein Hilfsmittel besitzt eine zustandsverändernde Benutzerinteraktion, wenn der Benutzer explizit den Zustand des Hilfsmittels ändern kann. Dies setzt voraus, daß das Hilfsmittel einen privaten oder öffentlichen Zustand besitzt. Das Geführte Tour-Hilfsmittel gehört zu dieser Klasse, da Geführte Touren gestartet und beendet werden können.

**Navigative Benutzerinteraktion:** Ein Hilfsmittel besitzt eine navigative Benutzerinteraktion, wenn der Benutzer einen Navigationsschritt mit dem Hilfsmittel auslösen kann. Das Historie-Hilfsmittel ist ein Beispiel dieser Klasse.

**Informative Benutzerinteraktion:** Eine informative Interaktion ermöglicht Benutzern, Informationen über die Datenbasis oder den Standort innerhalb der Datenbasis zu erhalten. Dies kann auch ohne Interaktion im eigentlichen Sinne geschehen, wie das Titelleiste-Hilfsmittel zeigt. Sie präsentiert dem Benutzer Informationen, kann aber nicht von ihm beeinflußt werden.

Diese drei Interaktionsarten schließen sich nicht gegenseitig aus, wie zum Beispiel das Inhaltsverzeichnis zeigt. Der Benutzer kann sich mit dem Inhaltsverzeichnis über seinen Fortschritt informieren und Navigationsschritte auslösen. Auf eine eigene Bezeichnung für jede mögliche Kombination aus Benutzerinteraktionen wird aus Gründen der Übersichtlichkeit abgesehen. Es erscheinen alle Kombination sinnvoll und möglich zu sein.

### 5.3.5 Adaptivität

Adaptive hypermediale Systeme verwalten ein Benutzermodell (vgl. Abschnitt 2.1.4), in dem Informationen über den Benutzer gespeichert werden. Durch pädagogischen Regeln können Inhalte und Verweise der Datenbasis und das hypermediale System selbst adaptiert werden.

**Adaptiv:** Solche Hilfsmittel, deren Erscheinungsform durch pädagogische Regeln verändert wird, sind adaptive Hilfsmittel. Ein Beispiel für ein adaptives Hilfsmittel ist das Inhaltsverzeichnis des Industriemeisters 2000. Hier werden dem Benutzer Informationen aus dem Benutzermodell bzgl. des Besucht-Status einer Seite präsentiert.

**Uniform:** Ein Hilfsmittel, das keine Informationen aus dem Benutzermodell nutzt, ist ein uniformes Hilfsmittel. Es erscheint jedem Benutzer gleich. Das Glossar-Hilfsmittel beispielsweise ist uniform, weil es nicht von dem Benutzermodell abhängt.

Adaptive Hilfsmittel sind nicht von DeBra et al. [DeBra99] beschrieben worden.

## 5.4 Klassifikation der Hilfsmittel

Anhand des oben beschriebenen Klassifikationsschemas können nun die in den Abschnitten 3.4 und 4.4 beschriebenen Hilfsmittel klassifiziert werden. Dies geschieht aus zwei Gründen. Zum einen sollen anhand des Schemas Besonderheiten aufgezeigt werden, also welche Klassen besonders häufig gewählt wurden. Zum anderen soll die Aussagekraft des Schemas gezeigt werden.

### 5.4.1 Die Hilfsmittel des Industriemeisters 2000

Zunächst werden die Hilfsmittel des Industriemeisters 2000 klassifiziert.

Das Geführte Tour-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt mit der Verwaltung laufender Touren einen **öffentlichen** Zustand,
- nutzt eine **erweiterte** Systeminteraktion, da es die Struktur Geführte Tour aus der Speicherschicht ausliest und Navigationsschritte abrechen kann,
- besitzt eine **zustandsverändernde** (aktuelle Tour) und eine **navigative** (Auslösen eines Navigationsschrittes beim Start einer Tour) Benutzerinteraktion und
- ist **uniform**.

Das Glossar-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt **keinen** Zustand,
- nutzt eine **erweiterte** Systeminteraktion, da es den Kurs Glossar aus der Speicherschicht ausliest und Navigationsschritte auslösen kann,
- besitzt eine **navigative** Benutzerinteraktion durch die Auswahl eines Glossarseite und
- ist **uniform**.

Das Historie-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt einen **privaten** Zustand (Kellerspeicher mit Komponentenverweisen),
- nutzt eine **dynamische** Systeminteraktion, da es nur Navigationsschritte protokolliert,
- besitzt eine **zustandsverändernde** und eine **navigative** Benutzerinteraktion, da das Auslösen eines Navigationsschrittes gleichzeitig auch den Zustand verändert und
- ist **uniform**.

Das Inhaltsverzeichnis-Hilfsmittel

- ist **darstellungsneutral**,

- besitzt **keinen** Zustand, da das Benutzermodell nicht Teil des Hilfsmittels ist,
- nutzt eine **erweiterte** Systeminteraktion, da es die Hierarchien aus der Speicherschicht ausliest und Navigationsschritte auslösen kann,
- besitzt eine **informative** (Besucht-Status der Seiten) und eine **navigative** (Auslösen eines Navigationsschrittes durch Auswahl) Benutzerinteraktion und
- ist **adaptiv**, da der Besucht-Status aus dem Benutzermodell ausgelesen wird.

Das Lesezeichen-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt einen **privaten** Zustand, der die vorhandenen Lesezeichen beschreibt,
- nutzt eine **erweiterte** Systeminteraktion, da es den Titel aus der Speicherschicht ausliest und Navigationsschritte auslösen kann,
- besitzt eine **zustandsverändernde** (Hinzufügen und Entfernen eines Lesezeichens) und eine **navigative** (Aktivierung eines Lesezeichens) Benutzerinteraktion und
- ist **uniform**.

Das Reihenfolge-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt einen **externen** Zustand, da eine aktuell laufende Tour die Funktionalität beeinflusst,
- nutzt eine **erweiterte** Systeminteraktion, da es die Reihenfolge aus der Speicherschicht ausliest und Navigationsschritte auslösen kann,
- besitzt eine **navigative** Benutzerinteraktion und
- ist **uniform**.

Das Startauswahl-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt **keinen** Zustand,
- nutzt eine **erweiterte** Systeminteraktion, da es die Struktur Kurs aus der Speicherschicht ausliest und einen Navigationsschritt auslöst,
- besitzt eine **navigative** Benutzerinteraktion, durch das Auslösen eines Navigationsschrittes zu Beginn einer Sitzung und
- ist **adaptiv**, da die zuletzt besuchte Seite eine Information aus dem Benutzermodell ist.

Das Titelleiste-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt einen **externen** Zustand (Geführte Tour),
- nutzt eine **erweiterte** Systeminteraktion, da es die Titel aus der Speicherschicht ausliest und auf Navigationsschritte reagiert,
- besitzt eine **informative** Benutzerinteraktion und
- ist **uniform**.

### 5.4.2 Die Hilfsmittel des Altenberger Doms

Nun werden die Hilfsmittel des Altenberger Doms analog zu den Hilfsmittel des Industriemeisters 2000 klassifiziert. Im Verhältnis zum obigen Abschnitt werden hier weniger Hilfsmittel aufgeführt, was mit dem unterschiedlichen Projektzustand zusammenhängt. Für das System des Altenberger Doms sind weitere vier Hilfsmittel geplant.

Das Historie-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt einen **privaten** Zustand, einen Kellerspeicher mit Komponentenverweisen,
- nutzt eine **erweiterte** Systeminteraktion, da es Navigationsschritte protokolliert und den Verknüpfungstyp aus der Speicherschicht ausliest,
- besitzt eine **zustandsverändernde** und eine **navigative** Benutzerinteraktion, da eine Rückverfolgung der Navigation gleichzeitig auch den Zustand ändert und
- ist **uniform**.

Das Lexikonzugriff-Hilfsmittel

- **beeinflusst** die Darstellung einer Komponente durch Tooltips,
- besitzt **keinen** Zustand,
- nutzt eine **erweiterte** Systeminteraktion, da es die Zielkomponente einer Lexikonreferenz aus der Speicherschicht ausliest und auf Navigationsschritte reagiert,
- besitzt eine **informative** Benutzerinteraktion und
- ist **uniform**.

Das Titelleisten-Hilfsmittel

- ist **darstellungsneutral**,
- besitzt **keinen** Zustand,
- nutzt eine **erweiterte** Systeminteraktion, da es die Titel aus der Speicherschicht ausliest und auf Navigationsschritte reagiert,
- besitzt eine **informative** Benutzerinteraktion und
- ist **uniform**.

Das virtueller Domhilfsmittel

- ist **darstellungsneutral**,
- besitzt einen **privaten** Zustand (der aktuelle Standpunkt im Modell),
- nutzt eine **dynamische** Systeminteraktion, da es nur Navigationsschritte auslösen kann,
- besitzt eine **zustandsverändernde** (Wechseln des Standpunktes) und eine **navigative** Benutzerinteraktion und
- ist **uniform**.

### 5.4.3 Bewertung des Klassifikationsschemas

Die Qualität des Schemas soll nun anhand der Hilfsmittel der beiden Anwendungen überprüft werden. Zur Übersicht wird die Klassifikation noch einmal in Tabelle 5.1 zusammengefaßt.

Die oben genannten Indikatoren für die Qualität des Schemas sind:

- Unabhängigkeit der einzelnen Kategorien voneinander
- Eindeutige Unterteilung in Klassen innerhalb der Kategorien
- Besetzung jeder Klasse in jeder Kategorie durch Hilfsmittel

Diese Kriterien sollten nun anhand der Tabelle 5.1 überprüft werden. Für eine statistische Unabhängigkeitsanalyse ist die Tabelle mit zwölf Zeilen jedoch nicht umfangreich genug, weshalb dieser Schritt ausgelassen wird. Eine inhaltliche Überprüfung der Kategorien läßt aber auf eine Unabhängigkeit schließen, weil sehr unterschiedliche Kategorien ausgewählt wurden.

Die eindeutige Unterteilung ist im Bereich der zustandsbeeinflussenden Benutzerinteraktion aufgrund der betrachteten Hilfsmittel nicht gegeben. Die zustandsbeeinflussende Benutzerinteraktion ging immer mit einer navigativen Benutzerinteraktion einher. In diesem Bereich könnte das Schema überarbeitet werden.

Die vollständige Besetzung wird mit zwei Ausnahmen erfüllt: Erstens existiert keine statische Systeminteraktion. Dies wurde bereits während der Vorstellung des Klassifikationsschemas erläutert und kann seine Ursache an den betrachteten Anwendungen haben. Zweitens ist die Benutzerinteraktion nicht vollständig belegt. Die Kategorie Benutzerinteraktionen erlaubt eine Kombination aus drei möglichen Benutzerinteraktionen, was eine Anzahl von sieben unterschiedlichen Klassen für diese Kategorie ergibt. Davon wurden aber nur vier Klassen belegt. Dies kann an der relativ geringen Anzahl von Hilfsmitteln liegen.

	Darstellung	Zustand	Systeminter.	Benutzerinter.	Adaptivität
Geführte Touren	neutral	öffentlich	erweitert	zustands. + nav.	uniform
Glossar	neutral	kein	erweitert	navigativ	uniform
Historie	neutral	privat	dynamische	zustands. + nav.	uniform
Inhaltsverzeichnis	neutral	kein	erweitert	info. + nav.	adaptiv
Lesezeichen	neutral	privat	erweitert	zustands. + nav.	uniform
Reihenfolge	neutral	extern	erweitert	navigativ	uniform
Startauswahl	neutral	kein	erweitert	navigativ	adaptiv
Titelleiste	neutral	extern	erweitert	informativ	uniform

Historie	neutral	privat	erweitert	zustands. + nav.	uniform
Lexikonzugriff	beeinflussend	kein	erweitert	informativ	uniform
Titelleisten	neutral	kein	erweitert	informativ	uniform
Virtueller Dom	neutral	privat	dynamisch	zustands. + nav.	uniform

Abbildung 5.1: Zusammenfassung der Klassifizierung

Zusammenfassend kann gesagt werden, daß das Schema den Kriterien weitestgehend entspricht und somit für die weiteren Betrachtungen ausreichend ist.

## 5.5 Vergleich der Hilfsmittel

Mit dem Klassifikationsschema aus dem Abschnitt 5.3 werden nun wesentliche Aspekte der Hilfsmittel diskutiert. Dabei wird versucht, Merkmale eines allgemeinen Hilfsmittels herauszustellen. Ein allgemeines Hilfsmittel abstrahiert von konkreter Funktionalität. Als beispielhafte Funktionalitäten werden die Hilfsmittel des Industriemeisters 2000 und des Altenberger Doms herangezogen. Es soll die Schnittstellen besitzen, mit denen alle konkreten Hilfsmittel der beiden betrachteten Anwendungen ihre Funktionalität verrichten können. Diese allgemeinen Betrachtungen werden die Basis für die Erweiterung des Dexter Modells im nächsten Kapitel sein.

### 5.5.1 Darstellungsbeschreibungen

Darstellungsbeschreibungen wurden im Dexter Modell nicht näher spezifiziert. Dies hängt mit den ebenfalls nicht spezifizierten Inhalten einer Komponente zusammen. Da die Medien innerhalb einer Komponente beliebig sein können, kann auch nicht beschrieben werden, wie ihre Darstellung beeinflusst werden kann. Das Dexter Modell nennt als Schnittstelle die Instanzierungsfunktion, die mit Darstellungsbeschreibungen parametrisiert werden kann, daraufhin die übergebenen Darstellungsbeschreibungen mit den Darstellungsbeschreibungen der Komponente kombiniert und anschließend die Komponente darstellt.

Die Betrachtungen des Lexikonzugriff-Hilfsmittel zeigten, daß ein Hilfsmittel mit Darstellungsbeschreibungen die Instanzierungsfunktion parametrisieren kann. Zwar wurde nur ein einziges solches Hilfsmittel in den beiden Anwendungen vorgefunden, aber das Prinzip kann verallgemeinert werden. Ein allgemeines Hilfsmittel kann Darstellungsbeschreibungen erzeugen und damit die Darstellung von Komponenten beeinflussen. Die dabei zu erfüllende Bedingung ist, daß die Instanzierungsfunktion die Darstellungsbeschreibungen auswerten kann. Ansonsten sind die Möglichkeiten der Darstellungsbeeinflussung zu umfangreich, um sie genauer spezifizieren zu können. Auch das AHAM benutzt Darstellungsbeschreibungen, um die Inhalte und die Verweise zu adaptieren (vgl. [DeBra99]). Die sich daraus ergebene Frage, ob man den Adaptionsgestalter ebenfalls als Hilfsmittel betrachten kann, läßt sich wahrscheinlich mit „Ja“ beantworten, sollte aber gesondert untersucht werden und wird deshalb hier nicht behandelt.

Ein allgemeines Hilfsmittel kann Darstellungsbeschreibungen erzeugen, die als Parameter der Instanzierungsfunktion übergeben werden.

### 5.5.2 Zustände

Zustandsbasierte Hilfsmittel in den beiden Anwendungen sind Geführte Tour, Historie und Lesezeichen bzw. Historie und virtueller Dom. Zusätzlich nutzen die Hilfsmittel Reihenfolge und Titelleiste den öffentlichen Zustand des Geführte Tour-Hilfsmittels. Der Zustand

eines Hilfsmittels kann die Navigation beeinflussen. So kann das Geführte Tour-Hilfsmittel in Abhängigkeit von seinem Zustand Navigationsschritte beeinflussen oder auch unterbinden. Des Weiteren wird der öffentliche Zustand des Hilfsmittels von anderen Hilfsmitteln genutzt.

In Abhängigkeit eines Zustandes kann ein zustandsbasiertes Hilfsmittel einen Navigationsschritt auslösen. Das Historie-Hilfsmittel des Industriemeisters 2000 nutzt einen Kellerspeicher. In Abhängigkeit von den bereits besuchten Seiten, also des Zustandes des Hilfsmittels, werden unterschiedliche Komponentenverweise beim Betätigen der Schaltfläche ausgelöst (vgl. Abschnitt 3.4.4). Allgemein kann keine genaue Beschreibung der Zustände eines Hilfsmittels erfolgen, da beliebige Informationen als Zustand gespeichert werden können. Eine Beschreibung der möglichen Zustände könnte die Funktionalität allenfalls einschränken.

Es ist möglich im Kontext eines Geführte Tour-Hilfsmittels eine Komponente um einen zusätzlichen Verweis auf die nächste Komponente der Tour zu erweitern (vgl. [Lyardet98]), oder den Titel der Geführten Tour in die Inhalte der Komponente einzublenden. In solchen Fällen würden Darstellungsbeschreibungen erzeugt werden, welche die Komponenten adaptieren würden (vgl. Abschnitt 5.5.1).

In den betrachteten Anwendungen wurde der öffentliche Zustand des Geführte Tour-Hilfsmittels nicht als Darstellungsbeschreibung für die Instanzierungsfunktion genutzt, sondern parametrisierte die Hilfsmittel und Reihenfolge und Titelleiste. Diese beiden Hilfsmittel reagieren beide auf einen neuen Navigationsschritt, indem sie Schaltflächen aktivieren oder deaktivieren bzw. den Titel der Seite anzeigen, jeweils in Abhängigkeit von dem Zustand des Geführte Tour-Hilfsmittels.

Allgemein gilt: beschreibt man einen öffentlichen Zustand als Darstellungsbeschreibungen, so kann der öffentliche Zustand sowohl der Instanzierungsfunktion als auch Hilfsmitteln mit einem externen Zustand als Parameter übergeben werden. Voraussetzung ist in beiden Fällen, daß Instanzierungsfunktion und Hilfsmittel die Darstellungsbeschreibungen auswerten können.

Besitzt ein Hilfsmittel einen Zustand, so besitzt es auch einen Startzustand. Ein Startzustand, eine konkrete Ausprägung des Zustandes, kann ebenfalls nicht genauer beschrieben werden. Der Startzustand kann von vorangegangenen Sitzungen abhängen, wie das Lesezeichen-Hilfsmittel zeigt.

### 5.5.3 Systeminteraktion eines allgemeinen Hilfsmittels

Im folgenden wird erneut angenommen, daß es sich bei den Hilfsmitteln und den Schichten des Dexter Modells um einzelne Objekte handelt, die miteinander interagieren.

#### 5.5.3.1 Interaktion mit der Laufzeitschicht

Die Beschreibungen der Hilfsmittel (siehe Abschnitt 3.4 und Abschnitt 4.4) zeigen, welche Funktionalität der einzelnen Schichten des Dexter Modells genutzt wurden. Betrachtet man die Interaktion mit der Laufzeitschicht, so kann man drei verschiedene Interaktionen unterscheiden:

1. Interaktion des Benutzers mit einem navigativen Hilfsmittel kann, wie auch die Aktivierung einer Verknüpfung zur Laufzeit, einen Navigationsschritt auslösen. Die Laufzeitschicht stellt den Hilfsmitteln eine Schnittstelle zur Verfügung, die also mindestens die Funktion `Navigationsschritt` umfaßt. Die Funktion `Navigationsschritt` nutzt die weiteren Funktionen des Dexter Modells.
2. Einige Hilfsmittel können die Navigation beeinflussen. Das Lexikonzugriff-Hilfsmittel beispielsweise beeinflußt die Darstellung von Komponenten durch Darstellungsbeschreibungen. Die in den vorhergehenden Betrachtungen identifizierten Darstellungsbeschreibungen sind Geführte Touren und Tooltip-Texte. Hilfsmittel generieren vor dem Aufruf der Instanzierungsfunktion Darstellungsbeschreibungen, geben dem Benutzer ein Feedback und können, wie das Geführte Tour-Hilfsmittel, sogar den Navigationsschritt verhindern. Ein allgemeines Hilfsmittel stellt der Laufzeitschicht also eine Schnittstelle zur Verfügung, welche die Ausführung von Anweisungen noch vor dem Aufruf der Instanzierungsfunktion ermöglicht.
3. Zustandsbasierte Hilfsmittel und Hilfsmittel mit informativer Benutzerinteraktion müssen über Navigationsschritte und externe Zustände informiert werden. Das Historie-Hilfsmittel protokolliert jeden durchgeführten Navigationsschritt. Da ein abgebrochener Navigationsschritt nicht protokolliert werden darf, muß eine Benachrichtigung über einen Navigationsschritt nach dem Aufruf der Instanzierungsfunktion durchgeführt werden.

Die Schnittstelle eines allgemeinen Hilfsmittels besitzt somit zwei Funktionen. Ein konkretes Hilfsmittel kann die übergebenen Parameter auswerten oder eine Funktion ohne Programmcode umsetzen.

**VorNavigation:** Diese Funktion eines Hilfsmittels erhält eine Komponenten-ID als Parameter. Sie kann mit dem Benutzer interagieren und hat als Rückgabewert das Tupel (Abbruch, Darstellung). Dabei ist Abbruch ein boole'scher Wert und Darstellung eine Darstellungsbeschreibung. Sie ermöglicht die Ausführung von Anweisungen vor dem Zugriff auf eine Komponente.

**NachNavigation:** Diese Funktion wird nach der Instanzierungsfunktion aufgerufen. Sie besitzt die Komponenten-ID der aktuell dargestellten Komponente und die übergebenen Darstellungsbeschreibungen als Parameter. Sie benachrichtigt das Hilfsmittel über den aktuell durchgeführten Navigationsschritt und erlaubt die Ausführung von Anweisungen nach dem Zugriff auf die Komponente.

Ein allgemeines Hilfsmittel nutzt folgende Schnittstelle der Laufzeitschicht:

**Navigationsschritt:** Diese Funktion der Laufzeitschicht kann von navigativen Hilfsmitteln aufgerufen werden. Als Parameter wird ein Komponentenverweis übergeben, der die darzustellende Komponente spezifiziert. Sie stellt eine neue Komponente dar und nutzt die Funktionen `VorNavigation` und `NachNavigation` der Hilfsmittel.

Weitere Funktionalität wird nicht benötigt. Die Laufzeitschicht verwaltet keinen eigenen Zustand. Die einzelnen Zustände werden innerhalb der Hilfsmittel verwaltet. Das Beispiel aus [Halasz94] für einen Zustand der Navigation, die Historie, wird durch das gleichnamige Hilfsmittel verwaltet. Durch eine Benachrichtigung über die aktuell dargestellte Komponente werden dem Hilfsmittel der beiden beschriebenen Anwendungen alle notwendigen Informationen mitgeteilt.

### 5.5.3.2 Interaktion mit der Speicherschicht

Wie im Abschnitt 5.4 beschrieben, interagieren einige Hilfsmittel zusätzlich mit der Speicherschicht. Die Interaktion mit der Speicherschicht dient dem Zugriff auf Komponenten- und anderen Strukturinformationen. Von den untersuchten Hilfsmitteln nutzen alle Hilfsmittel, die mit der Speicherschicht interagieren, strukturelle Besonderheiten der Datenbasis. Derartige Informationen werden von den Autoren der Datenbasis speziell für ein oder mehrere Hilfsmittel zur Verfügung gestellt. Von dem Titelleiste-Hilfsmittel des Industriemeisters 2000 beispielsweise wird die „Titel“-Eigenschaft einer Komponente ausgelesen, und das Reihenfolge-Hilfsmittel nutzt die Reihenfolgerelation, die durch die Kapitel definiert wurde. Damit hat die Auswahl eines Hilfsmittels direkte Auswirkungen auf die Informationen, die in der Datenbasis enthalten sein müssen.

Es findet keine Interaktion im eigentlichen Sinne statt, statt dessen erfolgt eine einseitige Nutzung der Dienste der Speicherschicht. Informationen über Komponenten und strukturelle Besonderheiten werden ausgelesen.

### 5.5.3.3 Interaktion mit der Komponenteninhaltsschicht

Eine Interaktion mit der Komponenteninhaltsschicht erfolgt von keinem der Hilfsmittel.

### 5.5.3.4 Interaktionen mit anderen Hilfsmitteln

Interaktion zwischen Hilfsmitteln erfolgt, um den öffentlichen Zustand eines Hilfsmittels auszulesen. Deshalb wird diese Betrachtung im Abschnitt 5.5.2 über Zustände geführt.

### 5.5.3.5 Zusammenfassung

Die Schnittstelle eines allgemeinen Hilfsmittels sieht zwei Funktionen vor, die den Hilfsmittel vor und nach der Instanzierung einer neuen Komponente ihre Aufgabe ermöglichen. Die Laufzeitschicht stellt den Hilfsmitteln eine Navigationsschrittfunktion zur Verfügung, während die Speicherschicht dem Hilfsmittel den Zugriff auf alle Komponenteninformationen ermöglicht.

#### 5.5.4 Benutzerinteraktion

Ein Hilfsmittel kann dem Benutzer vielfältige Interaktionsmöglichkeiten bieten. Dabei wurden drei Klassen vorgestellt. Die informativen Hilfsmittel stellen dem Benutzer Informationen zur Verfügung, die zum Beispiel das Problem der Desorientierung (siehe Abschnitt 2.4.2) zu vermeiden helfen. Für die weitere Betrachtung ist vor allem von Interesse, daß der Benutzer Navigationsschritte mit dem Hilfsmittel auslösen kann und daß er den Zustand eines Hilfsmittels verändern kann. Zum Beispiel kann der Benutzer eine Geführte Tour starten und damit den weiteren Verlauf der Navigation maßgeblich ändern.

Ein allgemeines Hilfsmittel kann dem Benutzer Informationen zur Verfügung stellen. Der Benutzer kann Navigationsschritte mit einem allgemeinen Hilfsmittel auslösen und seinen Zustand verändern.

#### 5.5.5 Adaptivität

Die im Industriemeister 2000 verwirklichte Möglichkeit, neben Inhalten und Verweisen auch Hilfsmittel zu adaptieren, wurde noch nicht in der Arbeit von De Bra et al. [DeBra99] beschrieben. Das Inhaltsverzeichnis greift direkt auf das Benutzermodell zu und nutzt einen eigenen Adaptionsgestalter für die Adaption des Hilfsmittels. Umgekehrt wäre es möglich, das Benutzermodell durch ein Hilfsmittel zu beeinflussen. Der Benutzer könnte zum Beispiel Seiten im Inhaltsverzeichnis als bereits bearbeitet markieren. Dies widerspricht aber dem Grundgedanken eines adaptiven Systems, da die Informationen implizit gesammelt werden sollten und nicht durch Benutzerinteraktion angepaßt werden sollten. Hier ist die Grenze zwischen Anpaßbarkeit und Adaptivität fließend. Die Altenberger Dom Anwendung besitzt kein Benutzermodell und deswegen können die Hilfsmittel nicht adaptiert werden.

Ein allgemeines Hilfsmittel besitzt einen eigenen Adaptionsgestalter und seine Darstellung kann mit pädagogischen Regeln adaptiert werden.

### 5.6 Bewertung des Vergleichs

Der Vergleich wurde dreigeteilt: Der Vergleich der Datenbasis ist ein Teilergebnis der Arbeit. Ein systemspezifischer Vergleich konnte aufgrund der verschiedenen Entwicklungszustände und der unterschiedlichen Betrachtungsansätze nicht ausführlich vorgenommen werden. Der Schwerpunkt dieses Kapitels lag auf dem Vergleich der Hilfsmittel. Allgemeine Eigenschaften eines beliebigen Hilfsmittels wurden herausgestellt, um hierauf im nächsten Kapitel aufzubauen.

Kritischer Punkt dieses Kapitels ist das Klassifikationsschema. Die Qualität des Schemas kann nicht überprüft werden, unter anderem, weil zwei Anwendungen hierfür nicht genug sind. Somit können auch die restlichen Ergebnisse der Arbeit nur so gut sein, wie das Schema es vorgibt. Die Kategorien und Klassen innerhalb der Kategorien wurden jedoch

sorgfältig ausgewählt und ständig weiterentwickelt, so daß zumindest von einer brauchbaren Qualität im folgenden ausgegangen werden kann. Eine Weiterentwicklung des Schemas anhand weiterer Anwendungen wäre dennoch langfristig wünschenswert.



---

## Kapitel 6

# Definition der Navigation

Im vorherigen Kapitel wurden die Hilfsmittel klassifiziert. Dabei wurden einige Eigenschaften herausgestellt, die sich auf die Navigation auswirken. Anhand derer wird nun im Abschnitt 6.1 eine Erweiterung des Dexter Modells für Hilfsmittel erstellt. Diese Erweiterung wird das Fundament für eine Definition der Navigation innerhalb der beiden betrachteten hypermedialen Systeme im Abschnitt 6.2 sein, die die Arbeitshypothese der Navigation aus dem Abschnitt 2.4.4 ersetzt.

### 6.1 Eine Erweiterung des Dexter Modells für Hilfsmittel

Im folgenden wird das Dexter Modell um Hilfsmittel erweitert. Schlüssel für die Erweiterung ist eine Analyse eines allgemeinen Hilfsmittels im Abschnitt 5.5. Mit dem überarbeiteten Modell kann die Modellierung der Hilfsmittel der beiden betrachteten hypermedialen Systeme erfolgen. Ob dieses Modell allgemein gültig ist, kann nicht aus den beiden betrachteten Anwendungen geschlossen werden. Das Modell sieht eine Einordnung der Hilfsmittel in die Schichtenarchitektur (siehe Abschnitt 6.1), eine genauere Beschreibung von Darstellungsbeschreibungen (siehe Abschnitt 6.1.3) und einen Ersatz der Laufzeitauflöserfunktion (siehe Abschnitt 6.1.4) sowie einen Ersatz für die **Komponentendarstellung**-Funktion (siehe Abschnitt 6.1.5) vor.

Auf der nun vorgestellten Erweiterung wird die abschließende Definition der Navigation aufbauen.

#### 6.1.1 Motivation

Das Dexter Modell dient unter anderem der Beschreibung eines hypermedialen Systems. Dabei wird, wie im Abschnitt 2.2.1.3 beschrieben, ein Navigationsschritt immer durch die Aktivierung einer Verknüpfung ausgelöst. Das Dexter Modell beschreibt das Zusammenspiel derjenigen Funktionen des Dexter Modells, die einen Navigationsschritt bewirken. Im wesentlichen ergab sich dieser durch zwei Schritte. In einem ersten Schritt wird die zur

Laufzeit visualisierte aktivierte Verknüpfung auf einen (oder mehrere) Zielkomponentenverweis(e) abgebildet. Ein so ermittelter Verweis wird in einem zweiten Schritt zusammen mit einer nicht näher spezifizierten Darstellungsbeschreibung an die Instanzierungsfunktion übergeben, welche eine Instanz der Komponente erzeugt.

Die Betrachtungen der vorhergehenden Kapitel zeigen, daß aktivierte Verknüpfungen nicht der einzige Startpunkt eines Navigationsschrittes sind. Ein Benutzer der beiden betrachteten hypermedialen Systeme kann mit navigativen Hilfsmitteln ebenfalls einen solchen initiieren. Des weiteren können Hilfsmittel, wie das Lexikonzugriff-Hilfsmittel, Darstellungsbeschreibungen für eine Instanzierung, also einer Darstellung einer Komponente, bereitstellen. Weiterhin müssen einige der betrachteten Hilfsmittel, wie beispielsweise das Titelleisten-Hilfsmittel, während jedes Navigationsschrittes über die aktuell dargestellte Komponente informiert werden. Das Geführte Tour-Hilfsmittel kann sogar einen Navigationsschritt unterbinden, dessen Ziel nicht innerhalb der Geführten Tour liegt.

Das Dexter Modell beschreibt das Zusammenspiel der vorhandenen Funktionen, um einen Navigationsschritt durchzuführen. Diese Beschreibung umfaßt keine Hilfsmittel der Navigation. Im folgenden wird unter anderen eine Funktion `Navigationsschritt` beschrieben, die die vorhandene Funktionalität des Dexter Modells nutzt und Hilfsmittel mit einbezieht, um einen Navigationsschritt durchzuführen. Diese Funktion ist der Kern der Erweiterung des Dexter Modells.

### 6.1.2 Die überarbeitete Schichtenarchitektur

Das Dexter Modell ist in drei Schichten aufgeteilt. Die Schichtenarchitektur sieht eine ausschließliche Nutzung der Funktionalitäten einer Schicht durch die nächsthöhere Schicht vor. Eine Integration der Hilfsmittel in die Schichtenarchitektur kann auf zwei Arten erfolgen: entweder durch eine Zuordnung der Hilfsmittel zu einer vorhandenen Schicht oder durch die Einführung einer neuen Hilfsmittelschicht.

Die Hilfsmittel müssen oberhalb der Speicherschicht modelliert werden, da sie Dienste der Speicherschicht nutzen und mit der Laufzeitschicht interagieren. Es würde somit gegen das Schichtenmodell verstoßen, wenn sie in der Speicherschicht oder Komponenteninhaltschicht liegen würden. Da sie aber Dienste der Speicherschicht nutzen und dem Benutzer Interaktionsmöglichkeiten bieten, können sie keine eigene Schicht oberhalb oder unterhalb der Laufzeitschicht bilden. Hilfsmittel sind also Teil der Laufzeitschicht. Abbildung 6.1 zeigt beispielhaft zwei Hilfsmittel innerhalb der Laufzeitschicht.

### 6.1.3 Darstellungsbeschreibungen

Das Dexter Modell geht davon aus, daß Darstellungsbeschreibungen notwendig für die Darstellung durch die Instanzierungsfunktion sind. Dabei werden statische Darstellungsbeschreibungen der Komponenten und dynamische Darstellungsbeschreibungen der Laufzeitschicht unterschieden. Darstellungsbeschreibungen werden nicht näher spezifiziert, da sie ähnlich den Inhalten einer Komponente zu allgemein sein können. Bemerkenswert ist

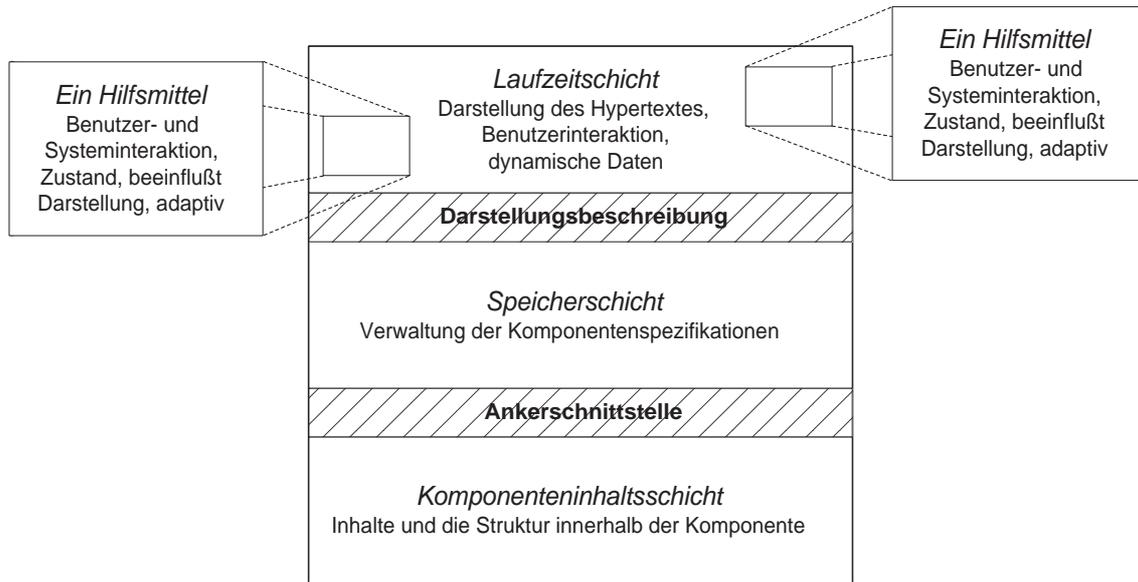


Abbildung 6.1: Das überarbeitete Dexter Modell. (Abbildung in Anlehnung an [Halasz94])

aber, daß dynamische Darstellungsbeschreibungen der Instanzierungsfunktion übergeben werden, ihre Herkunft aber nicht beschrieben wird.

Das überarbeitete Modell sieht eine Spezifizierung der Darstellungsbeschreibungen ebenfalls nicht vor. Aber ihre Herkunft kann nun beschrieben werden. Die Darstellungsbeschreibungen der beiden betrachteten Hypermediasysteme werden von Hilfsmitteln generiert. Die identifizierten Darstellungsbeschreibungen sind der Titel der aktuell laufenden Geführten Touren und Tooltips für Glossartexte. Beide Darstellungsinformationen wurden von Hilfsmitteln erzeugt. Teilweise nutzen weitere Hilfsmittel diese Darstellungsbeschreibungen, da sie externe Zustände beschreiben (vgl. Abschnitt 5.5.2).

Die der Instanzierungsfunktion übergebenen Darstellungsbeschreibungen werden in dem erweiterten Modell von den Hilfsmitteln einer Anwendung erzeugt. Im Abschnitt 6.1.5 wird beschrieben, wie die Darstellungsbeschreibungen der einzelnen Hilfsmittel der Instanzierungsfunktion übergeben werden.

#### 6.1.4 Laufzeitauflöserfunktion

Die Laufzeitauflöserfunktion des Dexter Modells kann dynamische Komponentenverweise anhand dynamischer Sitzungsdaten auflösen. Ein Beispiel für einen solchen dynamischen Verweis ist „die zuletzt dargestellte Komponente mit der Eigenschaft: Typ = Thema“. Auf die unzureichende Beschreibung dieser Funktion wurde bereits im Abschnitt 2.2.4 hingewiesen.

Dynamische Verweise werden in beiden Anwendungen nicht genutzt, alle Verweise sind statisch. Deshalb wird eine Laufzeitauflöserfunktion in diesen Systemen nicht benötigt.

```
procedure Navigationsschritt (Ziel: Komponentenverweis);  
var Ziel-ID : Komponenten-ID;  
    Abbruch : Boolean;  
    DB, DBNeu : Darstellungsbeschreibung;  
begin  
    Ziel-ID = Auflöserfunktion(Ziel);  
    for all Hilfsmittel in MengeAllerHilfsmittel do  
        (Abbruch, DBNeu) = Hilfsmittel.VorNavigation(Ziel-ID);  
        if Abbruch then exit procedure;  
        DB = DarstellungsKombination(DB, DBNeu);  
    end for;  
    Instanzierungsfunktion(Ziel-ID, DB);  
    for all Hilfsmittel in MengeAllerHilfsmittel do  
        Hilfsmittel.NachNavigation(Ziel-ID, DB);  
    end for;  
end;
```

Abbildung 6.2: Die Navigationsschritt-Funktion

Das erweiterte Modell benötigt ebenfalls die Laufzeitauflöserfunktion nicht, bietet aber mit Hilfsmittel und Adaption einen Ersatz für diese Funktion:

Das obige Beispiel beschreibt die Möglichkeit eines dynamischen Verweises. Obwohl solche nicht in den Anwendungen genutzt werden, beschreibt das Historie-Hilfsmittel des Altenberger Doms eine ähnliche Funktionalität. Allerdings wird sie nicht durch eine Verknüpfung, sondern durch ein Hilfsmittel angeboten. Das Hilfsmittel kann diesen „Verweis“ auflösen und einen Navigationsschritt initiieren. Es ist also nicht unbedingt ein dynamischer Verweis notwendig für derartige Funktionalität. Sollten dennoch dynamische Verweise erwünscht sein, so scheint eine Adaption der Inhalte unter Einhaltung der Hypertext-Invarianten eine Möglichkeit zu sein, die dem Gedanken des Dexter Modells entspricht und die Funktionalität der Laufzeitfunktion übernehmen kann. Diese Adaption könnte wiederum durch ein Hilfsmittel erfolgen, welches Darstellungsbeschreibungen erzeugt. Für eine Adaption sind Benutzer- und Lehrmodell nicht notwendig, es genügt eine flexible Instanzierungsfunktion, die die entsprechenden Darstellungsbeschreibungen verarbeiten kann.

Das überarbeitete Modell nutzt daher die Laufzeitauflöserfunktion des Dexter Modells nicht, sondern nur die Auflöserfunktion der Speicherschicht und gleicht dies durch die Hilfsmittel aus. Da die Laufzeitauflöserfunktion nicht näher beschrieben ist, kann nicht genau bestimmt werden, ob dieser Ersatz vollständig ist. Der Ersatz der Funktion durch Adaption und Hilfsmittel ist aber in sich konsistent und verstößt nicht gegen die Hypertext-Invarianten wie die Laufzeitauflöserfunktion (vgl. Abschnitt 2.2.4).

### 6.1.5 Die Navigationsschrittfunktion

Der Kern des überarbeiteten Modells ist die neue `Navigationsschritt`-Funktion. Sie wird in Abbildung 6.2 dargestellt und übernimmt die wesentlichen Aufgaben der `Komponentendarstellung`-Funktion des Dexter Modells (vgl. Abbildung 2.4).

Die `Navigationsschritt`-Funktion nutzt die allgemeine Schnittstelle eines Hilfsmittels wie in Abschnitt 5.5.3.5 beschrieben. Sie erzeugt Darstellungsbeschreibungen für die Instanzierungsfunktion und ermöglicht die Aktualisierung der Zustände.

Die `Navigationsschritt`-Funktion (siehe Abbildung 6.2) erhält als Parameter einen Komponentenverweis (`Ziel`). Dieser wird zu einer Komponenten-ID (`Ziel-ID`) durch die `Auflöserfunktion` aufgelöst. Daraufhin werden alle Hilfsmittel über einen bevorstehenden Navigationsschritt „informiert“, indem jeweils die `VorNavigation`-Funktion aufgerufen wird. Hilfsmittel können nun den Navigationsschritt unterbinden (`Abbruch = false`) oder durch Darstellungsbeschreibungen (`DBNeu`) für die Instanzierungsfunktion parametrisieren. Die einzelnen Darstellungsbeschreibungen der Hilfsmittel werden zu einer Darstellungsbeschreibung zusammengefaßt (`DB`). Nach dieser Vorbereitung wird die Instanzierungsfunktion mit der Komponenten-ID (`Ziel-ID`) und allen Darstellungsbeschreibungen (`DB`) aufgerufen. Sie erzeugt eine neue Instanz, die Komponente wird also dargestellt. Im Anschluß werden alle Hilfsmittel über den durchgeführten Navigationsschritt benachrichtigt, indem jeweils ihre `NachNavigation`-Funktion aufgerufen wird, der die Komponenten-ID der aktuellen Komponente (`Ziel-ID`) zusammen mit den Darstellungsbeschreibungen (`DB`) übergeben wird. Auf diese Art erhalten sowohl Instanzierungsfunktion als auch Hilfsmittel sämtliche Darstellungsbeschreibungen. Sie können dann diejenigen Informationen auswerten, die sie benötigen. Dieses Verfahren erlaubt es, neue Hilfsmittel und Darstellungsbeschreibungen hinzuzufügen, da keine Schnittstellen verändert werden müssen.

Dieses Verfahren ähnelt einem 2-Phasen-Commit-Protokoll (2PC-Protokoll) wie es zum Beispiel im Bereich verteilter Datenbanktransaktionen bekannt ist (vgl. [Dittrich97]). Im Unterschied zu dem 2PC-Protokoll wird hier die Funktion beendet, sobald ein Hilfsmittel einen Abbruch verlangt, eine Benachrichtigung über den Abbruch der Aktion findet nicht statt. Ein weiterer Unterschied besteht in der Rückgabe von Darstellungsbeschreibungen in dieser ersten Phase.

Eine Ähnlichkeit des Verfahrens existiert auch mit dem Observer-Muster von Gamma et al. [Gamma96]. Betrachtet man ein Hilfsmittel als einen Observer, so kann ein Hilfsmittel den Zustand (aktuelle Komponente + Zustände in Hilfsmittel) verändern (durch die `Navigationsschritt`-Funktion). Anschließend werden alle Hilfsmittel über den neuen Zustand informiert (durch die `NachNavigation`-Funktion). Die Unterschiede liegen in dem komplexeren Zustandswechsel, an dem alle Hilfsmittel beteiligt sind und an der expliziten Übergabe der Darstellungsbeschreibungen nach dem Zustandswechsel.

Diese `Navigationsschritt`-Funktion unterscheidet sich in drei wesentlichen Punkten von der `Komponentendarstellung`-Funktion aus Abschnitt 2.2.1.3:

1. Es hat sich herausgestellt, daß für die beiden betrachteten Anwendungen die Laufzeitauflöserfunktion nicht notwendig ist. Es werden nur statische Verweise genutzt, demnach wird die `Auflöserfunktion` genutzt (siehe Abschnitt 6.1.4).

2. Die in der Dexter Funktion nicht näher spezifizierte Herkunft der Darstellungsbeschreibungen kann nun als von Hilfsmitteln generierten Informationen beschrieben werden (siehe Abschnitt 6.1.3).
3. Die `Navigationsschritt`-Funktion interagiert mit Hilfsmitteln.

Die hier beschriebene `Navigationsschritt`-Funktion resultiert aus der Betrachtung der notwendigen Informations- und Kontrollflüsse eines allgemeinen Hilfsmittels und bietet die Schnittstelle für sämtliche Funktionalität der Hilfsmittel aus den beiden Anwendungen. Sie abstrahiert in wesentlichen Punkten die tatsächliche Realisierung im Industriemeister 2000 Projekt (vgl. Abschnitt 3.3.4). Dort werden allerdings mehrere ähnliche Funktionen genutzt, die alle die Reihenfolge der Zugriffsfunktion und der Aktualisierung der Hilfsmittel mit der `NachNavigation`-Funktion vertauschen, was aber keine weiteren Auswirkungen besitzt. Ein weiterer Unterschied liegt in der uneinheitlichen Schnittstelle der Hilfsmittel des Industriemeisters 2000, was sich auf einen prototypischen Ansatz in den ersten Projektphasen zurückführen läßt.

### 6.1.6 Zusammenfassung

Es wurde eine Erweiterung des Dexter Modells dargestellt. Hilfsmittel wurden als Bestandteil des Modells beschrieben, mit deren Hilfe die nicht näher spezifizierten Darstellungsbeschreibungen und die Laufzeitauflöserfunktion genauer beschrieben werden können. Kern der Erweiterung ist eine `Navigationsschritt`funktion, die mit Hilfsmitteln interagiert. Dafür wurde von konkreten Hilfsmitteln abstrahiert und eine allgemeine Schnittstelle für Hilfsmittel beschrieben. Öffentliche Zustände der Hilfsmittels werden allen anderen als Parameter übergeben, so daß die Hilfsmittel mit einem externen Zustand diese nutzen können und kein direkter Informationsfluß zwischen Hilfsmitteln notwendig ist. Die Laufzeitauflöserfunktion wird nicht mehr verwendet.

Es wurde mit den Hilfsmitteln ein Bereich identifiziert, der nicht durch das Dexter Modell beschrieben wird, obschon die meisten hypermedialen Anwendungen Hilfsmittel nutzen. Ein Historie-Hilfsmittel kann beispielsweise als „State of the Art“ im Bereich hypermedialer Anwendungen betrachtet werden. Das Dexter Modell sieht ebenfalls eine Historienfunktionalität vor, spezifiziert diese aber nicht genauer. Die Erweiterung spezifiziert ebenfalls keine Historienfunktionalität, beschreibt allerdings eine Möglichkeit diese und weitere Funktionalität in das System einzubinden.

Im Gegensatz zu den anderen vorgestellten Erweiterungen des Dexter Modells (AHM und AHAM) liegt der Schwerpunkt dieser Erweiterung auf der Laufzeitschicht und nicht auf der Speicherschicht. Insofern kann es sehr einfach mit den anderen beiden Erweiterungen kombiniert werden, d. h. anstelle auf dem Dexter Modell aufzubauen, könnte es auch auf einem der anderen beiden Modellen oder einer Kombination der beiden aufbauen.

Die Erweiterung des Dexter Modells erfolgte natürlichsprachlich. Eine genauere Spezifikation mit der Spezifikationsprache *Z* [Spivey98] wäre wünschenswert, da auch Dexter in *Z* spezifiziert wurde [Halasz90].

## 6.2 Definition

Für den Begriff der Navigation wurde im Abschnitt 2.4.4 nur eine Arbeitshypothese gegeben. Die Arbeitshypothese beruhte auf dem Dexter Modell, welches im vorangegangenen Abschnitt um die Modellierung von Hilfsmitteln erweitert wurde. Nun wird diese Erweiterung für eine abschließende Definition der Navigation in den beiden Systemen Industriemeister 2000 und Altenberger Dom genutzt. Das Dexter Modell eignet sich nicht, um dynamische Aspekte eines hypermedialen Systems zu betrachten. Damit wird die Arbeitshypothese verworfen. Die Definition der Navigation baut, ähnlich der Arbeitshypothese, auf Navigationsschritte auf.

### **Definition 6 (Navigationsschritt)**

*Ein Navigationsschritt ist der von Hilfsmitteln beeinflussbare Wechsel des aktuellen Knotens und der Veränderung der Zustände zustandsbasierter Hilfsmittel durch die in Abbildung 6.2 beschriebene Navigationsschritt-Funktion.*

Diese Funktion ist, wenn Hilfsmittel mit privaten oder öffentlichen Zuständen existieren, ebenfalls zustandsbasiert. Der Zustand einiger Hilfsmittel kann aber auch durch Benutzerinteraktion (siehe Abschnitt 5.5.2) verändert werden. Wird eine Geführte Tour gestartet, so hat diese Zustandsänderung Auswirkungen auf nachfolgende Navigationsschritte. Somit ergibt sich folgende abschließende Definition für Navigation in den beiden betrachteten Systemen:

### **Definition 7 (Navigation)**

*Navigation ist die sequentielle Abfolge der Navigationsschritte und der zustandsverändernden Benutzerinteraktionen mit Hilfsmitteln während einer Sitzung.*

Da Navigation zustandsabhängig ist, muß die Navigation einen Startzustand besitzen. Dieser ergibt sich aus den Startzuständen der einzelnen Hilfsmittel sowie aus der Startkomponente, d. h. aus der Komponente, welche beim Beginn einer Sitzung angezeigt wird. Der Startzustand kann durch das Benutzermodell beeinflusst werden, da im Benutzermodell sitzungsübergreifend Zustände gespeichert werden.

## 6.3 Bewertung der Definition

Welche Bedeutung hat diese Definition? Navigation sollte aus der Sicht der technischen Realisierung betrachtet werden. Ausgehend von einer technischen Sichtweise mit Hilfe des Dexter Modells, konnte diese Sichtweise verfeinert werden. Speziell die Hilfsmittel der Navigation, welche im Dexter Modell nicht betrachtet wurden, sind nun Teil der Definition. Sie beeinflussen die Präsentation, sie initiieren und unterbinden Navigationsschritte und verwalten den Zustand der Navigation. Die Vielseitigkeit eines Hilfsmittels macht sie zusammen mit dem Dexter Modell zu den Grundpfeilern der Definition.

Sind Hilfsmittel damit womöglich zu vielseitig? Sollte die Definition von Hilfsmitteln neu überdacht werden? Ihre Definition im Abschnitt 2.4.3 läßt einen großen Spielraum offen

und auch die beschriebenen Funktionalitäten differieren deutlich. Dennoch war es eher unproblematisch, Hilfsmittel in den Anwendungen zu identifizieren. Vor allem die externe Navigation von Tergan (siehe Abschnitt 2.4) bietet einen ersten Ansatzpunkt zum Auffinden von Hilfsmitteln. Andere Hilfsmittel, wie zum Beispiel das Lexikonzugriff-Hilfsmittel, sind weniger offensichtlich. Der hier vorgestellte Hilfsmittelbegriff ist zweckdienlich für diese Arbeit gewesen. Solange nicht weitere Anwendungen in Hinblick auf Hilfsmittel hin analysiert wurden, sollte er auch eher allgemein als zu speziell definiert werden, um interessante Bereiche nicht schon im Vorfeld auszuschließen.

Die Definition hat sich lediglich in technischen Details im Gegensatz zur Arbeitshypothese verändert. Die grobe Ausrichtung, eine Grundlage für den Informationszugriff, ist erhalten geblieben. Nicht betrachtet wurde die Motivation des Benutzers zur Navigation (vgl. Abschnitt 2.4), wie anfänglich gefordert. Damit war die Arbeitshypothese im Abschnitt 2.4.4 zweckdienlich gewählt.

Die technischen Details der Definition machen die Änderungen der Definition aus. Während das Dexter Modell nicht ausreichte, um Navigation in den beiden betrachteten Anwendungen zu beschreiben, konnte dies mit einer Erweiterung desselben erfolgen, welche die Spezifikation eines hypermedialen Systems verfeinerte.

---

## Kapitel 7

# Zusammenfassung und Ausblick

In diesem Kapitel werden die wichtigsten Ergebnisse der Diplomarbeit noch einmal zusammengefaßt und mögliche Fortsetzungen der Arbeit werden beschrieben.

### 7.1 Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde Navigation für die beiden Hypermediaanwendungen Industriemeister 2000 und Altenberger Dom anhand einer Erweiterung des Dexter Modells definiert.

Der Ausgangspunkt war eine Beschreibung und Analyse der beiden Anwendungen mit Hilfe des Dexter Modells. Anhand dieser Referenz konnten dann die beiden Anwendungen miteinander verglichen werden, sofern dieses möglich war. Der Schwerpunkt während der Analyse lag auf den Hilfsmitteln. Es konnten fünf wesentliche Eigenschaften von Hilfsmitteln herausgestellt werden: Erzeugung von Darstellungsbeschreibungen, Zustände, System- und Benutzerinteraktion sowie Adaptivität. Anhand dieser Eigenschaften wurde von konkreten Hilfsmitteln abstrahiert, und die Eigenschaften eines allgemeinen Hilfsmittels wurden beschrieben.

Mit dieser Vorarbeit konnte das Dexter Modell um Hilfsmittel erweitert werden. Hilfsmittel sind Teil der Laufzeitschicht, sie können die Instanzierungsfunktion mit Darstellungsbeschreibungen beeinflussen und die Funktion der Laufzeitauflöserfunktion übernehmen. Kern der Erweiterung ist eine Navigationsschrittfunktion, welche eine Interaktion mit den Hilfsmitteln beschreibt, die ähnlich dem 2-Phasen-Commit-Protokoll erfolgt.

Abschließend konnte das Ziel der Diplomarbeit, eine Definition der Navigation für die beiden betrachteten Anwendungen, mit Hilfe der Dexter-Erweiterung und den Eigenschaften der Hilfsmitteln erreicht werden.

## 7.2 Ausblick

Da als Basis für die Ergebnisse dieser Arbeit nur zwei Anwendungen genutzt wurden, kann noch nicht abschließend über die Allgemeingültigkeit der Ergebnisse diskutiert werden. Mit Hilfe des „Vorgehensmodells zur Analyse der Navigation“ (siehe Anhang B) sollten deshalb weitere Anwendungen unter besonderer Berücksichtigung der folgenden Punkte analysiert werden:

- Das Dexter Modell umschreibt eine Laufzeitauflöserfunktion. In den vorliegenden Anwendungen wurde sie durch den Einsatz von Hilfsmittel ersetzt. Kann ein solcher Schritt auch für andere Anwendungen erfolgen?
- Können die Hilfsmittel weiterer hypermedialer Anwendungen auch mit der Erweiterung des Dexter Modells abstrakt in ihrer Funktionalität beschrieben werden?
- Da Hilfsmittel, ähnlich den pädagogischen Regeln, vor und nach der Zugriffsfunktion aufgerufen werden, könnte die Ausführung der pädagogischen Regeln als Hilfsmittel realisiert werden. In diesem Fall wäre auch der Adaptionsgestalter als ein Hilfsmittel zu betrachten, welches Darstellungsbeschreibungen erzeugen würde. Kann also das AHAM in der Laufzeitschicht ebenfalls als ein Hilfsmittel modelliert werden?
- Bernstein [Bernstein98] beschreibt Hypertextmuster. Er untersuchte Hypertextdokumente auf häufig wiederkehrende Verknüpfungsschema, die er als Hypertextmuster bezeichnet. Ausgehend von identifizierten Mustern in Hyperdokumenten, könnte untersucht werden, ob eine Realisierung der Hypertextmuster durch Hilfsmittel erfolgen kann und sinnvoll ist.
- Werden Verknüpfungen dynamisch angezeigt oder verändert (auch ohne Benutzermodell wie zum Beispiel in [Doberkat96]), könnte die Erstellung eines Hilfsmittel „Verknüpfungen“ sinnvoll werden. Das Hilfsmittel würde, ähnlich dem Hilfsmittel Lexikonzugriff aus Abschnitt 4.4.2 und einem Anpassungsgestalter, die Darstellung einer Komponente mit Verknüpfungen ausstatten.
- In den untersuchten Anwendungen nutzen alle Hilfsmittel, die mit der Speicherschicht interagieren, strukturelle Besonderheiten der Datenbasis. Sind Zugriffe der Hilfsmittel auf die Speicherschicht immer nur notwendig, wenn strukturelle Besonderheiten ausgelesen werden sollen?

Wenn auf diese Art eine hinreichend große Sammlung von beschriebenen Hilfsmittel angelegt wurde, könnten diese auf gegenseitige Wechselwirkungen und Kombinierbarkeit hin untersucht werden. Welche Hilfsmittel passen besonders gut zusammen oder wurden besonders häufig miteinander kombiniert? Der Kriterienkatalog aus Abschnitt 5.3 könnte vielleicht ein Ansatzpunkt für die Kombination verschiedener Hilfsmittel darstellen, zum Beispiel könnte eine bestimmte Kombination von Hilfsmitteln mit verschiedenen Eigenschaften besonders gut zusammenpassen.

Weiterhin könnten die gewonnen Erkenntnisse als Muster einer Problemkategorie (siehe [Buschmann98]) beschrieben werden. Die Problemkategorie „Navigation in hypermedialen Systemen“ könnte dann Entwurfsmuster (siehe [Gamma96]) und Analysemuster (siehe [Fowler99]) umfassen. Die `Navigationsschritt`-Funktion des erweiterten Dexter Modells könnte, wenn sie auf weitere Anwendungen übertragbar ist, als ein Entwurfsmuster beschrieben werden. Die Hilfsmittel könnten als Analysemuster beschrieben werden. Analysemuster würden sich hierfür eignen, wenn der Schwerpunkt auf die Auswahl passender Hilfsmittel und ihre Kombination miteinander gelegt wird. Einen ähnlichen Weg sind bereits Lyardet et al. [Lyardet98] gegangen.

Das Dexter Modell besitzt eine formale Spezifizierung in der Z-Notation. Nachfolgende Arbeiten könnten das erweiterte Modell dieser Arbeit ebenfalls in eine formale Z-Spezifikation überführen.

In beiden Anwendungen wurden Entwurfsentscheidungen noch nicht dokumentiert, weshalb eine solche Dokumentation eine sinnvolle Fortsetzung dieser Arbeit wäre. Entwurfsentscheidungen bezüglich der Hilfsmittel würden den Hilfsmittel-Katalog inhaltlich aufwerten.



---

## Anhang A

# Die Terminologie des Dexter Modells

Das Dexter Modell besitzt eine eigene Terminologie, die noch keine allgemein anerkannte Übersetzung ins Deutsche besitzt. Da das Dexter Modell von zentraler Bedeutung für diese Arbeit ist, wird hier die Terminologie noch einmal zusammengefaßt.

**Anker** („anchor“): Möglicher Endpunkt einer Verknüpfung. Anker gehören zu den Komponenteninformationen. Sie bestehen aus einer Anker-ID und einem Ankerwert.

**Ankerwert** („anchor value“): Information für die Darstellung eines Ankers. Diese Information wird nicht von dem Modell ausgewertet.

**Anker-ID** („anchor id“): Eine eindeutige Bezeichnung für einen Anker innerhalb einer Komponente. Damit ist ein Anker systemweit eindeutig durch eine Kombination von Komponenten-ID und Anker-ID.

**Ankerinstanz** („link marker“): Ein instantierter Anker, also ein Teil einer Instanz zur Laufzeit.

**Ankerschnittstelle** („anchoring“): Die Schnittstelle der Komponenteninhaltsschicht. Die Speicherschicht greift über die Ankerschnittstelle auf die Komponenteninhaltsschicht zu.

**atomare Komponente** („atomic component“): Eine Komponente mit einem Knoten als Basiskomponente.

**Auflöserfunktion (Verweisauflöserfunktion)** („resolver function“): Diese Funktion der Laufzeitschicht liefert zu einem Komponentenverweis eine Komponenten-ID (sie „löst“ den Verweis „auf“). Wenn es sich bei dem Komponentenverweis um eine Komponenten-ID handelt, so reduziert sich diese Funktion auf die Identität.

**Basisinstanz** („base instantiation“): Eine dargestellte Basiskomponente.

**Basiskomponente** („base component“): Eine Basiskomponente ist entweder ein Knoten, eine Verknüpfung oder ein Kompositum. Die Basiskomponente bildet zusammen mit einer Komponenten-ID und den Komponenteninformationen eine Komponente.

**Benutzer** („user“): Ein Benutzer kann einen Hypertext betrachten und verändern, Sitzungen öffnen und entfernen. Ein Benutzer interagiert mit der Laufzeitschicht.

**darstellen** („presenting“): Das Sichtbarmachen von Komponenten oder Anker zur Laufzeit durch ein hypermediales System. Dies wird vom Modell nicht näher beschrieben.

**Darstellungsbeschreibung** („presentation specification“): Informationen, welche die Darstellung von Komponenten beeinflussen. Da Darstellungsbeschreibungen sehr stark auf die Strukturen innerhalb der Komponenten bezug nehmen können, werden sie ebenfalls nicht näher spezifiziert. Die Schnittstelle der Speicherschicht heißt ebenfalls Darstellungsbeschreibung.

**Eigenschaft** („attribute/value pair“): Eine beliebige Information, die eine Komponente besitzen kann. Sie wird als (Name, Wert)-Tupel gespeichert. Die Eigenschaften sind zusätzliche Informationen, auf die das Modell nicht näher eingeht.

**Endpunktbeschreibung** („specifier“, „end-point specification“): Teil einer Verknüpfung. Besteht aus einem Komponentenverweis, einer Anker-ID, einer Richtung und einer Darstellungsbeschreibung.

**entfernen** („unpresenting“): Eine Instanz kann entfernt werden. Dadurch wird die Instanz nicht länger angezeigt, und auf sie kann nicht mehr über die Instanz-ID zugegriffen werden. Die Komponente, also die statische Information der Speicherschicht wird dadurch nicht gelöscht. Sitzungen werden ebenfalls entfernt, wenn der Benutzer seine Sitzung beendet.

**FolgeVerknüpfung** („followLink“): Die **FolgeVerknüpfung**-Funktion realisiert einen Navigationsschritt im Dexter Modell. Ausgehend von einer aktivierten Ankerinstanz wird die zugehörige Verknüpfung bestimmt und die Zielkomponentenverweise zusammen mit nicht näher spezifizierten Darstellungsbeschreibungen der **Komponentendarstellung**-Funktion übergeben (siehe Abbildung 2.4).

**Hypertext-Invarianten** („invariants of hypertext“): Die Invarianten des Modells lauten:

**I1:** Die Zugriffsfunktion ist eine umkehrbare Funktion, die Komponenten-IDs auf Komponenten abbildet. Das impliziert, daß jede Komponente eine Komponenten-ID besitzt.

**I2:** Der Bildbereich der Auflöserfunktion umfaßt alle Komponenten-IDs.

**I3:** Keine Komponente ist (direkt oder transitiv) Teil von sich selbst.

**I4:** Die Anker-IDs der Endpunktbeschreibungen stimmen mit den Anker-IDs der Komponenten überein, zu denen die Komponentenverweise der Endpunktbeschreibungen aufgelöst werden können.

**I5:** Alle Komponentenverweise können zu Komponenten-IDs aufgelöst werden.

**Inhalt** („content“): Ein Knoten besitzt beliebige Inhalte. Die Inhalte sind in keiner Weise eingeschränkt. Es können beispielsweise Bilder, Animationen oder Texte sein. Die Inhalte werden nicht vom Dexter Modell ausgewertet.

**Instanz** („instantiation“): Eine Instanz ist eine Darstellung einer Komponente. Jede Instanz besitzt eine Instanz-ID. Wird eine Instanz erzeugt, so wird eine Basisinstanz sowie Ankerinstanzen erzeugt. Jede Instanz gehört zu einer Sitzung.

**Instanzierungsfunktion** („instantiator function“): Sie liefert anhand einer Komponenten-ID und einer Darstellungsbeschreibung eine Instanz der Komponente zurück.

**Instanz-ID** („instantiation identifier“, „IID“): Eindeutige Bezeichnung einer Instanz.

**Hypertext** („hypertext“): Der Hypertext ist die statische Beschreibung einer Menge von Komponenten. Er wird dem Benutzer über Instanzen dargestellt. Ein Hypertext muß nach diesem Modell bestimmte Invarianten erfüllen.

**Knoten** („atom“, „node“): Ein Knoten eines Hypertextes.

**Komponente** („component“): Eine Komponente ist die zentrale Zugriffseinheit eines Hypertextes. Eine Komponente besitzt eine Komponenten-ID, Komponenteninformationen und eine Basiskomponente.

**Komponentendarstellung-Funktion** („present component operation“): Die **Komponentendarstellung-Funktion** liefert anhand eines Komponentenverweises und einer Darstellungsspezifikation eine Instanz, indem sie den Komponentenverweis auflöst und anschließend die Instanzierungsfunktion benutzt (siehe Abbildung 2.4).

**Komponenteninhaltsschicht** („within-component layer“): Die „unterste“ Schicht des Dexter Modells. Sie stellt die Ankerschnittstelle zur Verfügung und wird nicht näher von dem Modell beschrieben.

**Komponenteninformationen** („component information“): Die Komponenteninformationen umfassen eine Liste der Anker, eine Darstellungsbeschreibung und eine Menge von Eigenschaften.

**Komponentenverweis** („component specification“): Der Komponentenverweis ist eine Beschreibung einer Komponente. Sie muß so gestaltet sein, daß sie durch die Auflöserfunktion zu einer Komponenten-ID aufgelöst werden kann.

**Komponenten-ID** („unique identifier“, „UID“): Eine systemweit eindeutige Bezeichnung für genau eine Komponente.

**Kompositum** („composite entity“): Ein Kompositum besteht aus mehreren Komponenten.

**Kompositum-Komponente** („composite component“): Eine Komponente mit einem Kompositum als Basiskomponente. Eine Komponente kann nicht Teil von sich selber sein.

**Laufzeitschicht** („run-time layer“): Die „oberste“ der drei Schichten des Dexter Modells. Sie verwaltet Instanzen und Sitzungen. Sie greift auf die Speicherschicht zu.

**Laufzeitauflöserfunktion** („run-time resolver function“): Diese Funktion ist das Laufzeit-Äquivalent zu der Auflöserfunktion der Speicherschicht. Ihr Definitionsbereich ist mindestens der der Auflöserfunktion, kann aber noch dynamische Informationen verarbeiten (zum Beispiel Historie-Funktion).

**Richtung** („direction“): Es existieren vier Richtungen für eine Endpunktbeschreibung: START („FROM“), ZIEL („TO“), BEIDES („BIDIRECT“) oder KEINES („NONE“). Die Richtung KEINE kann zum Beispiel für Funktionen außerhalb des Hypertext verwandt werden.

**löschen** („delete“): Wird eine Komponente (anhand einer Instanz) gelöscht, so werden alle Instanzen der Komponente entfernt und die Komponente sowie alle Verknüpfungen mit Endpunkten innerhalb der Komponente aus dem Hypertext gelöscht. Die Auflöserfunktion wird angepaßt.

**Sitzung** („session“): Jeder Zugriff auf einen Hypertext findet im Rahmen einer Sitzung statt. Instanzierungen sind Teil einer Sitzung.

**Speicherschicht** („storage layer“): Die „mittlere“ der drei Schichten des Dexter Modells. Auf ihr liegt der Schwerpunkt des Modells. Sie beschreibt die Struktur eines Hypertextes als eine endliche Menge von Komponenten zusammen mit der Zugriffs- und Auflöserfunktion.

**Speicherfunktion** („realizer function“): Wurde eine Instanz verändert, so können die Änderungen mit dieser Funktion gespeichert (realisiert) werden.

openSession	Erzeugt eine Sitzung für einen Hypertext. Es erfolgt keine Instanzierung.
openComponents	Erzeugt eine Menge von Instanzierungen für eine Menge von Komponenten.
presentComponent	Erzeugt eine Instanz anhand einer Endpunktbeschreibung und einer Darstellungsbeschreibung.
followLink	Benutzt openComponents um alle Komponenten darzustellen, die einen Zielpunkt einer ausgewählten Verknüpfung besitzen.
newComponent	Erzeugt eine Instanz einer neuen Komponente.
unPresent	Entfernt eine Instanz.
editInstantiation	Ermöglicht es, eine Instanz zu verändern.
realizeEdits	Speichert die Änderungen an einer Instanz.
deleteComponent	Löscht die Komponente einer gegebenen Instanz. Entfernt weiterhin alle Instanzen dieser Komponente.
closeSession	Entfernt eine Sitzung. Änderungen an Instanzen werden standardmäßig nicht gespeichert.

Tabelle A.1: Funktionen und Methoden der Laufzeitschicht

**speichern** („realizing“): Mit speichern ist das Zurückschreiben einer veränderten Instanz in die Speicherschicht gemeint. Die Komponente kann nur unter Erhaltung der Hypertext-Invarianten verändert werden.

**Startanker:** Ein Startanker ist ein Anker, auf den mit einer Endpunktbeschreibung der

CreateComponent	Erzeugt eine neue Komponente und fügt diese zum Hypertext hinzu. Weiterhin werden die Zugriffs- und Auflöserfunktion erweitert, damit die Komponenten-ID der Komponente genutzt werden kann.
CreateAtomicComponent	Benutzt CreateComponent, um anhand eines Knotens und einer Darstellungsbeschreibung eine neue atomare Komponente zu erzeugen.
CreateLinkComponent	Benutzt CreateComponent, um anhand einer Verknüpfung und einer Darstellungsbeschreibung eine Verknüpfungskomponente zu erzeugen.
CreateCompositeComponent	Benutzt CreateComponent, um anhand von mehreren Komponenten und einer Darstellungsbeschreibung eine neue Kompositum-Komponente zu erzeugen.
CreateNewComponent	Wird von der Laufzeitschicht aufgerufen. Ruft entweder CreateAtomicComponent, CreateLinkComponent oder CreateCompositeComponent auf.
DeleteComponent	Löscht eine Komponente und alle Verknüpfungen, deren Endpunktbeschreibungen auf die Komponente verweisen. Verändert wiederum Zugriffs- und Auflöserfunktion.
ModifyComponent	Erlaubt die Veränderung einer Komponente. Bewahrt den Typ der Komponente (atomar, Verknüpfung oder Kompositum), die Komponenteninformationen und die Konsistenz der Verknüpfungen (erlaubt also keine baumelnde Verweise).
GetComponent	Benutzt die Zugriffsfunktion, um anhand einer Komponenten-ID eine Komponente zurückzuliefern. Handelt es sich um eine Verknüpfungskomponente, so liefert GetComponent entweder eine Zielpunkt- oder eine Startpunktbeschreibung der Verknüpfung zurück.
AttributeValue	Liefert anhand einer Komponenten-ID und einem Eigenschaftsnamen einen Eigenschaftswert zurück.
SetAttributeValue	Setzt den Wert einer Eigenschaft.
AllAttributes	Liefert eine Menge aller Eigenschaftsnamen einer Komponente zurück.
LinksToAnchor	Liefert anhand eines Ankers und der dazugehörigen Komponente eine Menge von Verknüpfungen, die auf den Anker verweisen.
LinksTo	Liefert anhand einer Komponenten-ID die Komponenten-IDs der Verknüpfungen, die auf die Komponente verweisen.

Tabelle A.2: Funktionen und Methoden der Speicherschicht

Richtung „START“ oder der Richtung „BEIDES“ verwiesen wird. Ein Startanker ist der Ausgangspunkt einer Verknüpfung. Zur Laufzeit wird er als Ankerinstanz dargestellt, über die der Benutzer die Verknüpfung aktivieren kann.

**Verknüpfung** („link“): Eine Verknüpfung ist eine Basiskomponente. Sie besteht aus mindestens zwei Endpunktbeschreibungen. Ein Anker kann gleichzeitig Start- und Zielanker sein.

**Verknüpfungskomponente** („link component“): Eine Komponente mit einer Verknüpfung als Basiskomponente.

**Zielanker:** Ein Zielanker ist ein Anker, auf den mit einer Endpunktbeschreibung der Richtung „ZIEL“ oder der Richtung „BEIDES“ verwiesen wird. Ein Anker kann gleichzeitig Ziel- und Startanker sein.

**Zugriffsfunktion** („accessor function“): Die Zugriffsfunktion liefert eine Komponente anhand einer Komponenten-ID zurück.

Das Dexter Modell beschreibt verschiedene Funktionalitäten für die Laufzeit- und Speicherschicht. Diese Funktionen sollen den Kern der Funktionalität zur Benutzung von Hypertext widerspiegeln. Es werden bewußt keine Methoden für die Komponenteninhaltschicht aufgeführt, da diese Schicht nicht im Modell spezifiziert wird. Die Methoden der Laufzeit- und Speicherschicht, die nicht eingehender betrachtet werden, sind in den Tabellen A.1 und A.2 zusammengefaßt.

---

## Anhang B

# Ein Vorgehensmodell zur Analyse der Navigation

Im folgenden wird ein Vorgehensmodell zur Analyse der Navigation in hypermedialen Systemen beschrieben, welches dieser Arbeit zugrundeliegt. Es ist ein einfaches Modell, welches lediglich den Inhalt der jeweiligen Dokumente umschreibt, die erstellt werden sollen. Dieses Modell kann als Ansatzpunkt zur Analyse weiterer Hypermediaanwendungen dienen und könnte damit Grundlage sein für weitere Arbeiten, die auf dieser Arbeit aufbauen. Dem Vorgehensmodell liegt die Zielsetzung zugrunde, eine Wissensbasis über Hilfsmittel und Navigation mit Hilfsmitteln anzusammeln und damit das Dexter Modell und den Begriff der Navigation weiterzuentwickeln. Die Anwendung des Modells auf weitere Anwendungen soll die Wissensbasis verändern, erweitern oder bestätigen und das Modell und die Definition weiterentwickeln.

Das Vorgehensmodell sieht drei Phasen vor: Analyse, Vergleich und Weiterentwicklung. In der Analysephase wird eine Hypermediaanwendung getrennt von weiteren Anwendungen beschrieben und analysiert. In der Vergleichsphase werden die Ergebnisse der Analysephase mit Ergebnissen vorhergehenden Arbeiten verglichen. Die Weiterentwicklungsphase sieht dann eine Überarbeitung des Klassifikationsschemas und des Dexter Modells für Hilfsmittel vor. Die Phasen müssen mindestens einmal hintereinander durchlaufen werden. Innerhalb der Phasen gilt das gleiche für die Erstellung der einzelnen Dokumente. Die Phasen und ihre Dokumente sind im einzelnen:

### B.1 Dokumente der Analysephase

In der Analysephase wird eine einzelne hypermediale Anwendung beschrieben. Die Kapitel 3 und 4 dieser Arbeit entsprechen dieser Phase.

**Projektbeschreibung:** Eine Projektbeschreibung umfaßt eine Beschreibung der Zielsetzung und der Realisierung. Eine hypermediale Anwendung wird für einen bestimmten Zweck entworfen und umgesetzt. Die Zielsetzung könnte Auswirkungen

auf die gesamte Anwendung besitzen. Die Beschreibung sollte sich möglichst nah an der tatsächlichen Realisierung halten und sollte die notwendigen Informationen zum Verstehen der Anwendung umfassen.

**Beschreibung der Datenbasis:** Beschrieben werden sollten Inhalt und Strukturen mit dem Schwerpunkt auf den Strukturen. Der Inhalt ist von Interesse, wenn er den Kontext der Anwendung beschreibt oder Herkunft und Interpretationsmöglichkeiten der Strukturen erklärt.

**Beschreibung des hypermedialen Systems:** Mit der Hilfe des Dexter Modells kann nun, aufbauend auf der Datenbasis, das hypermediale System beschrieben werden. Verschiedene Erweiterungen des Modells können ebenfalls herangezogen werden. Dabei soll vor allem auf die Navigationsschrittfunktion des Systems und auf die Unterstützung der Merkmale des Dexter Modells eingegangen werden.

**Beschreibung der Hilfsmittel:** Dies ist der Schwerpunkt der Analysephase. Hilfsmittel sollten möglichst genau beschrieben werden, um auf nach mehreren Analysen aussagekräftige Sammlung von Hilfsmitteln zu erhalten. Beachtenswert sind auch die im Abschnitt 7.2 erwähnten offenen Fragestellungen im Zusammenhang mit Hilfsmitteln.

## B.2 Dokumente der Vergleichsphase

In der Vergleichsphase werden die Ergebnisse der Analysephase zu weiteren Anwendungen in Relation gesetzt. Das Kapitel 5 dieser Arbeit entspricht dieser Phase.

**Vergleich der Strukturen der Datenbasis:** Die Strukturen der Datenbasis sollten unabhängig von den Inhalten mit den Strukturen anderer Anwendungen in Hinblick auf Semantik und den sich daraus ergebenden Möglichkeiten für Hilfsmittel verglichen werden. Mehrfach vorhandene Strukturen sollten herausgestellt werden.

**Vergleich der hypermedialen Systeme:** Auch bei den hypermedialen Systemen liegt der Schwerpunkt des Vergleiches auf den Möglichkeiten, die den Hilfsmitteln zur Verfügung gestellt werden. So kann beispielsweise die Navigationsschrittfunktion verglichen und ihre Konformität zu der **Navigationsschritt**-Funktion dieser Arbeit gezeigt oder widerlegt werden. Wesentliche Eigenschaften der Hilfsmittel sollten herausgestellt werden.

**Klassifikationsschema:** Das Klassifikationsschema soll anhand der vorangegangenen Untersuchungen geändert, erweitert oder bestätigt werden. Diese Erstellung eines Klassifikationsschemas ist eine der wichtigsten und anspruchsvollsten Aufgaben des Modells, da keine äußeren Faktoren die Qualität bewerten können. Die einzelnen Kategorien sollten unabhängig voneinander sein und unterschiedliche Bereiche abdecken.

**Klassifikation der Hilfsmittel:** Anhand des überarbeiteten Schemas können dann alle bislang beschriebenen Hilfsmittel aller Anwendungen neu klassifiziert werden. Auf diese Weise erhält man einen Katalog von Hilfsmitteln.

**Vergleich der Hilfsmittel:** In diesem Dokument können nun Auffälligkeiten beschrieben werden. Schwerpunkt sollten aber die Gemeinsamkeiten der Hilfsmittel sein und wie man anhand eines allgemeinen Verhaltens alle Hilfsmittel beschreiben kann.

### B.3 Dokumente der Weiterentwicklungsphase

In der Weiterentwicklungsphase werden nun die Ergebnisse, die Erweiterung des Dexter Modells und die Definition der Navigation, weiterentwickelt. Das Kapitel 6 dieser Arbeit liefert die Grundlage für diese Phase.

**Erweiterung des Dexter Modells:** Das Dexter Modell legt den Schwerpunkt der Modellierung auf die Speicherschicht. Die zu entwickelnde Erweiterung des Dexter Modells soll vor allem die Laufzeitschicht eingehender betrachten und die Modellierung von Hilfsmitteln innerhalb des Dexter Modells beschreiben.

**Definition der Navigation:** Navigation wurde in dieser Arbeit auf der Basis des erweiterten Dexter Modells und den Hilfsmitteln beschrieben. Aufgrund neuer Erkenntnisse kann eine Änderung der Definition notwendig werden.

### B.4 Benutzung des Modells

Das Modell soll möglichst häufig auf verschiedene Hypermediaanwendungen angewendet werden. Ziel dieser Iteration ist langfristig eine aussagekräftige Definition der Navigation, eine umfassende Erweiterung des Dexter Modells für Hilfsmittel und ein möglichst vollständiger Katalog von Hilfsmitteln. Damit sind die Ergebnisse der vorliegenden Arbeit Hypothesen, die nach jeder Iteration entweder bestätigt oder verworfen werden. Eine verworfene Hypothesen wird durch eine neue ersetzt, die nicht im Widerspruch zu den bisherigen Erkenntnissen steht. Auf diese Weise sollten stetig verbesserte Hypothesen erzeugt werden, bis schließlich mehr über die allgemeine Anwendbarkeit und Gültigkeit der Hypothesen bekannt ist.



---

# Literaturverzeichnis

- [AD97] Lehrstuhl für Softwaretechnologie und Lehrstuhl für Baugeschichte (beide Universität Dortmund): *Antrag zum Kooperationsprojekt – Multimedialechnik in der baugeschichtlichen Lehre– Veranschaulichung eines gotischen Architektursystems am Beispiel des Altenberger Domes*, projektinternes Papier, 8 Seiten, 1997.
- [Alfert98] Klaus Alfert, Ernst-Erich Doberkat und Corinna Copka: *Der Altenberger Dom als multimediales Lernsystem: Ein Projekt zwischen zwei Disziplinen*, PowerPoint-Präsentation, erhältlich über den Lehrstuhl für Softwaretechnologie an der Universität Dortmund, 25 Folien, 1998.
- [Alfert99] Klaus Alfert: *Developing The Altenberger Dom Presentation – Integrating Content Providers And Software Developers*, Euromedia 99, Seiten 70-77, 1999.
- [Berk91] Emily Berk und Joseph Devlin (Herausgeber): *Hypertext / Hypermedia Handbook*, McGraw-Hill, New York, 583 Seiten, 1991.
- [Bernstein98] Mark Bernstein: *Patterns of Hypertext*, Proceedings of the 9th Conference on Hypertext and Hypermedia (Pittsburgh, Pennsylvania, 20.-24. Juni 1998) ACM, New York, Seiten 21-29, 1998.
- [Bosak99] Jon Bosak und Tim Bray: *Mehr Tempo auf der Datenautobahn - Neue Websprache XML*, Spektrum der Wissenschaft, Juli 1999, Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, 1999.
- [DeBra92] Paul De Bra, Geert-Jan Houben und Yoram Kornatzky: *An Extensible Data Model for Hyperdocuments*, Proceedings of the 4th Conference on Hypertext, Mailand, Seiten 222-231, 1992.
- [DeBra99] Paul De Bra, Geert-Jan Houben und Hongjing Wu: *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia*, Proceedings of the 10th ACM Conference on Hypertext and Hypermedia (Darmstadt, Deutschland, 21.-25. Februar 1999), ACM, New York, Seiten 147-156, 1999.
- [Brusilovsky96] P. Brusilovsky: *Methods and Techniques of Adaptive Hypermedia*, User Modeling and User-Adapted Hypermedia, Kluwer academic publishers, Band 6, Seiten 87-129, 1996.

- [Bulterman91] Dick C. A. Bulterman, Guido van Rossum und R. van Liere: *A structure for transportable, dynamic multimedia documents*, Proceedings of the Summer 1991 USENIX Conference (Nashville, Tennessee, Juni 1991), Seiten 137–155, 1991.
- [Bulterman93] Dick C. A. Bulterman: *Specifying and support of adaptable networked multimedia*, Multimedia Systems, Band 1, Nummer 2, Seiten 68-76, 1993.
- [Buschmann98] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad und Michael Stal: *Pattern-orientierte Softwarearchitektur*, Addison-Wesley, Bonn, 455 Seiten, 1998.
- [Bush45] Vannevar Bush: *As we may think*, Atlantic Monthly, Juli 1945, Seiten 101-108, 1945.
- [Director98] *Macromedia Director, Version 6.5*, Macromedia Incorporation, San Francisco, California, 1998.
- [Dittrich97] Klaus Dittrich: *Datenbanksysteme*, in [Rechenberg97], Seiten 745-774, 1997.
- [Doberkat96] Ernst-Erich Doberkat: *Browsing a Hyperdocument*, Memorandum Nr. 87 des Lehrstuhles für Softwaretechnologie, Universität Dortmund, 1996.
- [Duden96] *Deutsches Universalwörterbuch*, 3., völlig neu bearbeitete und erweiterte Auflage, Dudenverlag, Mannheim, 1996.
- [Faber93] Wolfgang Faber: *Hypermediale Lernsysteme*, Proceedings der Konferenz „Forschung für die Wirtschaft – Im Mittelpunkt: der Mensch“ (März 1993 in Wien), Service Verlag, Wien, 1993.
- [Fowler99] Martin Fowler: *Analysemuster – Wiederverwendbare Objektmodelle*, Addison-Wesley, Bonn, 386 Seiten, 1999.
- [Gamma96] Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides: *Entwurfsmuster*, Addison-Wesley, Bonn, 430 Seiten, 1996.
- [Gay91] Geri Gay und Joan Mazur: *Navigating in Hypermedia*, in [Berk91], Seiten 271-283, 1991.
- [Grønbæk97] Kaj Grønbæk, Niels Olof Bouvin und Lennert Sloth: *Designing Dexter-based hypermedia services for the World Wide Web*, Proceedings of the 8th ACM Conference on Hypertext (Pittsburg, Pennsylvania, 20.-24. Juni 1997), ACM, New York, Seiten 146-156, 1997.
- [Haack97] Johannes Haack: *Interaktivität als Kennzeichen von Multimedia und Hypermedia*, in [Issing97], Seiten 150-166, 1997.
- [Halasz90] Frank Halasz und Mayer Schwartz: *The Dexter Hypertext Reference Model*, Proceedings of the Hypertext Workshop (National Institute of Standards and Technology, Gaithersburg, Maryland, Januar 1990), NIST Special Publication 500-178, Seiten 95-133, März 1990.

- [Halasz94] Frank Halasz und Mayer Schwartz: *The Dexter Hypertext Reference Model*, Communications of the ACM, Band 37, Februar 1994, Seiten 30-39, 1994.
- [Hardman94] Linda Hardman, Dick C. A. Bulterman und Guido van Rossum: *The Amsterdam Hypermedia Modell: Adding Time And Context To The Dexter Modell*, Communications of the ACM, Band 37, Februar 1994, Seiten 50-62, 1994.
- [ICD98] Informatik Centrum Dortmund: *Gesamtkonzeption / Anforderungsdefinition für das Projekt „Industriemeister 2000“*, projektinternes Papier, 150 Seiten, 1998.
- [IHK96] R. Schulz und E. Koch: *Aufbau multimedialer Netze zur Entwicklung und Evaluation von Modulen im naturwissenschaftlichen Bereich für das DIHT/IHK-Konzept Industriemeister 2000*, projektinternes Papier, 1996.
- [Issing97] Ludwig J. Issing und Paul Klinsa (Herausgeber): *Information und Lernen mit Multimedia*, Psychologie Verlags Union, Weinheim, 2. überarbeitete Auflage, 494 Seiten, 1997.
- [Leggett94] John J. Leggett und John L. Schnase: *Viewing Dexter With Open Eyes*, Communications of the ACM, Band 37, Februar 1994, Seiten 76-86, 1994.
- [Lennon97] Jennifer A. Lennon: *Hypermedia Systems and Applications: World Wide Web and Beyond*, Springer Verlag, Heidelberg Berlin, 297 Seiten, 1997.
- [Lowe99] David Lowe und Wendy Hall: *Hypermedia & the Web: an engineering approach*, John Wiley & Sons Ltd., Chichester, 599 Seiten, 1999.
- [Lyardet98] Fernando Daniel Lyardet, Gustavo H. Rossi und Daniel Schwabe: *Using Design Patterns in Educational Multimedia Applications*, Proceedings of the ED-Media'98 Conference, Freiburg, 1998.
- [Mühlhäuser97a] Max Mühlhäuser: *Multimedia*, in [Rechenberg97], Seiten 729-744, 1997.
- [Mühlhäuser97b] Max Mühlhäuser: *Verteilte Systeme*, in [Rechenberg97], Seiten 557-588, 1997.
- [Nelson65] Theodor H. Nelson: *A file structure for the complex, the changing, and the indeterminate*, Proceedings of the 20th ACM National Conference, Seiten 84-10, 1965.
- [Newcomb91] Steven R. Newcomb, Neill A. Kipp und Victoria T. Newcomb: *The “HyTime“ Hypermedia/Time-based Document Structuring Language*, Communications of the ACM, Band 34, November 1991, Seiten 67-83, 1991.
- [Nielsen90] Jakob Nielsen: *The art of navigating in hypertext*, Communications of the ACM, Band 33, März 1990, Seiten 298-310, 1990.

- [Nürnberg 97] Peter J. Nürnberg, John J. Leggett und Erich R. Schneider: *As We Should Have Thought*, Proceedings of the 8th ACM Conference on Hypertext (Pittsburg, Pennsylvania, 20.-24. Juni 1997), ACM, New York, Seiten 96-101, 1997.
- [Pott97] Oliver Pott: *Netscape Communicator 4 im Internet*, Smart Books, Kilchberg, 294 Seiten, 1997.
- [Rechenberg97] Peter Rechenberg und Gustav Pomberger (Herausgeber): *Informatik-Handbuch*, Carl Hanser Verlag, München Wien, 961 Seiten, 1997.
- [Schulmeister96] Rolf Schulmeister *Grundlagen hypermedialer Lernsysteme: Theorie – Didaktik – Design*, Addison-Wesley, Bonn, 454 Seiten, 1996.
- [Steinmetz99] Ralf Steinmetz: *Multimediatechnologie: Grundlagen Komponenten und Systeme*, Springer Verlag, Berlin, 968 Seiten, 2. vollständig überarbeitete und erweiterte Auflage, 1999.
- [Stern99] Judith Stern und Robert Lettieri: *Quicktime Pro*, Peachpit Press, 288 Seiten, 1999.
- [Spivey98] J. Mike Spivey: *The Z Notation: a reference manual*, second edition, zu beziehen über den Autor (zuerst erschienen bei: Prentice Hall International, Hertfordshire, 1992), 158 Seiten, 1998.
- [Tergan97] Sigmar-Olaf Tergan: *Hypertext und Hypermedia: Konzeption, Lernmöglichkeiten, Lernprobleme*, in [Issing97], Seiten 122-137, 1997.
- [Tochtermann95] Klaus Tochtermann: *Ein Modell für Hypermedia: Beschreibung und integrierte Formalisierung wesentlicher Hypermediakonzepte*, Doktorarbeit an der Universität Dortmund, Verlag Shaker, Aachen, 289 Seiten, 1995.
- [Toolbook97] *Asymetrix Toolbook II, Version 6.0*, Asymetrix Corporation, Bellevue WA, 1997.
- [W3C98] World Wide Web Consortium: *Extensible markup language (XML) 1.0.*, Technical Report REC-xml-19980210, zu beziehen über <http://www.xml.org> (Stand: September 1999), 1998.